DEVELOPMENT OF A FAST RESPONSE DISPERSION MODEL FOR VIRTUAL URBAN ENVIRONMENTS

by

Balwinder Singh

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical Engineering

The University of Utah

May 2012

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **BALWINDER SINGH**

has been approved by the following supervisory committee members:

Eric R. Pardyjak	, Chair	**02/08/2010** Date Approved
Pete Willemsen	, Member	**02/08/2010** Date Approved
Patrick A. McMurtry	, Member	**02/08/2010** Date Approved
Michael J. Brown	, Member	**04/30/2010** Date Approved
Meredith M. Metzger	, Member	Date Approved

and by **Tim A. Ameel**, Chair of the Department of **Mechanical Engineering**

and by Charles A. Wight, Dean of The Graduate School.

ABSTRACT

According to a UN report, more than 50% of the total world's population resides in urban areas and this fraction is increasing. Urbanization has a wide range of potential environmental impacts, including those related to the dispersion of potentially dangerous substances emitted from activities such as combustion, industrial processing or from deliberate harmful releases. This research is primarily focused on the investigation of various factors which contribute to the dispersion of certain classes of materials in a complex urban environment and improving both of the fundamental components of a fast response dispersion modeling system – wind modeling and dispersion modeling. Specifically, new empirical parameterizations have been suggested for an existing fast response wind model for street canyon flow fields. These new parameterizations are shown to produce more favorable results when compared with the experimental data. It is also demonstrated that the use of Graphics Processing Unit (GPU) technology can enhance the efficiency of an urban Lagrangian dispersion model and can achieve near real-time particle advection. The GPU also enables real-time visualizations which can be used for creating virtual urban environments to aid emergency responders. The dispersion model based on the GPU architecture relies on the so-called "simplified Langevin equations (SLEs)" for particle advection. The full or generalized form of the Langevin equations (GLEs) is known for its stiffness which tends to generate unstable modes in particle trajectory, where a particle may travel significant distances in a small time step.

A fractional step methodology has been used to implement the GLEs into an existing Lagrangian random walk model to partially circumvent the stiffness associated with the GLEs. Dispersion estimates from the GLEs-based model have been compared with the SLEs-based model and available wind tunnel data. The GLEs-based model is more dispersive than the SLEs-based model in both the lateral and vertical directions. It is observed that for the present test case, the GLEs-based model performed relatively better than the SLEs-based model.

TABLE OF CONTENTS

1. INTRODUCTION

Recently, the world has seen rapid growth in urbanization (Rotach 2001). Rapid growth of urban areas can impact a myriad of activities, including those related to industry and transportation. This trend raises concerns about environmental deterioration. The dispersion of pollutants along with accidental discharges threatens urban air quality. Deliberate or accidental release of a harmful material in urban areas, where population density is high, is also of major concern. This research is primarily focused on the investigation of the various factors which contribute to the dispersion of certain classes of toxic material in a complex urban environment.

The threat of an accidental or deliberate release of a chemical or biological agent in densely populated urban areas has led to the development of a number of urban fast response transport and dispersion models. These models are characterized by short computational turnover times and account for the effects of buildings and other structures present in urban flow fields. Fast response models are particularly useful for emergency response scenarios where quick dispersion estimates are required to act appropriately, and for vulnerability assessments of urban areas where a wide range of model inputs must be investigated in a limited amount of time (Brown 2004).

Airborne releases of toxic gases and aerosols may occur in cities and cause great harm to the general population. Recent field experiments (Allwine et al. 2002; Allwine et

al. 2004; Hanna et al. 2004) and computational fluid dynamics modeling (DeCroix and Brown 2002; Coirier and Reich 2003; Pullen et al. 2005; Hanna et al. 2006) of tracer releases in built-up city centers indicate that buildings significantly alter transport and dispersion. Near street level, for example, the plume may travel several blocks in a direction opposing the prevailing wind and many blocks laterally. This topological dispersion can lead to secondary sources and significantly alter the rate of lateral dispersion (Belcher 2005). A ground-level source can rise several hundred meters in depth in less than a block when caught in the updraft just downwind of a tall building (Hanna et al. 2006). Buildings also alter the timing of the transport and dispersion, generally resulting in much longer residence times as compared to open terrain (Hanna 2006; Doran et al. 2006).

For applications where quick turnaround time is required (e.g., an emergency response to a chemical accident in a city) or where thousands of simulations must be performed in a few days or less (e.g., a vulnerability assessment of a particular urban site), computational fluid dynamics (CFD) modeling is currently not fast enough. While "on demand" CFD calculations are not practical for these applications, there are a number of research groups investigating the use of computational fluid dynamics (CFD) models for fast response applications. Ideas range from coarse resolution simulations using drag (Lim et al. 2001; Chan et al. 2004) to library approaches where a large number of cases are precomputed and results for specific cases are interpolated from the library (Smith and Brown 2002). Another approach to speed up a CFD model is to run a simulation

where the pressure Poison equation is not solved for complete convergence (Gowardhan et al. 2010).

For many years, "urbanized" Gaussian plume models have been used for rapid turnaround applications using urban-specific vertical and lateral plume spread parameters (McElroy 1969). Evaluation studies have shown that at distances greater than about one kilometer from the source and/or for low density urban areas, Gaussian models perform fairly well when comparing maximum concentrations that are unpaired in space (Hanna et al. 2003; Venkatram et al. 2004). Since Gaussian plume models only use a single averaged wind speed and wind direction as input, they cannot represent the complex three-dimensional wind and concentration fields that develop within the urban core around buildings. To better account for the effects of buildings on near source transport and dispersion, a number of researchers have developed simple fast-running models to account for the lateral displacement of a plume centerline due to off-axis channeling (Theurer et al. 1996) and to compute the concentration fields around a single isolated building (Genikhovich and Snyder 1994; Wilson and Chui 1994; Ramsdell and Fosmire 1995; Schulman et al. 2000) and within a street canyon (Dabberdt et al. 1973; Yamartino et al. 1989; Eerens et al. 1993; Berkowicz 2000). These models are capable of predicting dispersion for isolated buildings, two building street canyons, or for idealized building arrays, but not for the complex arrangements and shapes of buildings that occur in real cities.

Over the past ten to fifteen years, there has been a considerable amount of effort placed in developing urban transport and dispersion models that run relatively fast but

account for the effects of groups of arbitrary shapes and arrangements of buildings in an approximated way. Hall et al. (2000) describe a Gaussian puff model called the Urban Dispersion Model (UDM) for use on neighborhood to city scales (~10 m to 10 km). The model accounts for building wake cavity mixing and some along street channeling, but does not compute a 3D flow field around the buildings. When a puff intercepts a building, it is instantaneously placed in the lee of the building in the cavity and puffs are then emitted over time from the cavity. Brook et al. (2003) have evaluated the model against idealized building arrays in the lab and field as well as against outdoor urban field experiment data. Another well known, but non-peer-reviewed modeling system, called MIDAS-AT, computes 3D wind fields around building complexes using potential flow theory with dispersion modeled using a traditional three-term boundary-layer random-walk model (http://www.absconsulting.com/midas/). The potential flow approach allows for channeling of the flow down streets, but does not allow for important rotational flow phenomena such as street canyon vortices that form between buildings or recirculating cavities that develop downwind of an isolated building.

Röckle (1990) derived a unique model that computes flow around buildings using empirical equations and mass conservation. Röckle's methodology was incorporated into the ABC and ASMUS models which were intended for dispersion applications at industrial sites and have undergone several evaluation studies (Gross et al. 1994; Gross 1997). The ABC and ASMUS models accomplished transport and dispersion through a K-theory Eulerian diffusion model. A small number of urban wind models have been developed based on the Röckle approach which have all utilized Lagrangian random-

walk models to accomplish transport and dispersion (Kaplan and Dinar 1996; Moussafir et al. 2004; Wang et al. 2005). The Los Alamos National Laboratory (LANL) in collaboration with various universities has also utilized the Röckle concept. Over the past seven years, a dedicated team of researchers has worked to carefully evaluate the model, improve the original flow algorithms and implement new algorithms. The wind model has been modified to work with complex arrangements of buildings, including the ability to stack buildings on top of one another to create semirealistic city center layouts. The wind model, QUIC-URB (Singh et al. 2008; Gowardhan et al. 2010), is part of the Quick Urban & Industrial Complex (QUIC) dispersion modeling system which contains an "urbanized" random-walk model called QUIC-PLUME (Williams et al. 2004) and a graphical user interface called QUIC-GUI (Nelson et al. 2006). QUIC-PLUME is unique in that it contains a nonlocal mixing scheme and more drift terms than the traditional random-walk model in order to account for the inhomogeneous turbulence associated with urban flows (Williams et al. 2004). QUIC has been applied to neighborhood-scale problems in such places as New York City, Washington DC, Chicago, Oklahoma City and Salt Lake City (see for example: http://www.lanl.gov/orgs/d/d4/atmosphere/chbio .shtml).

This dissertation attempts to further improve/advance the two important components of urban fast response dispersion modeling - wind modeling and dispersion modeling. Improvements have been suggested for the QUIC-URB wind model's empirical parameterizations for complex urban building configurations. New street-canyon parameterizations have been suggested which help in better mimicking the mean

flow field in presence of complex building configurations. An attempt has been made to further advance the current state of fast response dispersion modeling by utilizing commodity graphics hardware, in what is called general purpose computation on graphics processing units (GPU) to achieve real-time simulation and visualization of large numbers of particles simulated using a so-called "simplified" Lagrangian random-walk model (simplified to horizontally homogenous flows). This research investigates the feasibility of a novel application of an existing fast response Lagrangian dispersion modeling system to achieve real-time simulation and visualization of an urban plume that a user can interact with in a virtual environment (VE) through the utilization of commodity graphics hardware. This GPU-based fast response dispersion model (GPU Plume) can also be used for virtual reality applications and emergency training purposes. The simplified Lagrangian random-walk model, employed in the GPU Plume, does not solve the full or generalized form of Langevin equations (GLEs) (Rodean 1996; Yee and Wilson 2007) for advecting particles due to the stiffness associated with the GLEs and large number of terms present in the GLEs. The simplified form of Langevin equations (SLEs) solves the equations for an assumed horizontally homogenous flow field, which drastically reduces the number of terms in the equations. This dissertation attempts to incorporate and validate the GLEs into an existing Lagrangian random-walk model using a fraction step method (Yee and Wilson 2007) to partially circumvent the stiffness associated with the GLEs.

As part of this work, a Lagrangian dispersion model based on the GPU architecture has been developed and evaluated against the available analytical solutions,

wind-tunnel data and an existing Lagrangian dispersion model, QUIC Plume, for its performance and accuracy of results. A Lagrangian random-walk model using the GLEs has also been developed and validated against the available analytical solutions, wind-tunnel data and QUIC-Plume. This research can be divided into three components: (1) development and evaluation of a new street canyon empirical parameterization for the wind model, QUIC-URB; (2) development and evaluation of a Lagrangian dispersion model based on the GPU-based architecture; (3) development and evaluation of a Lagrangian dispersion model based on the GLEs using a fractional step method (Yee and Wilson 2007).

1.1 QUIC-URB - an improved street canyon parameterization

The QUIC-URB diagnostic wind model generates a mass consistent spatially explicit mean wind field in urban areas with complex building configurations. Based on the work done by Röckle (1990) and Kaplan and Dinar (1996), QUIC-URB uses empirical parameterizations to specify the flow field physics on a gridded domain and enforces mass consistency on the initial flow field to obtain a final flow field (Singh et al. 2008; Gowardhan et al. 2010).

Chapter 2 describes the QUIC-URB fast response urban diagnostic wind modeling tool and evaluates it against wind-tunnel data for a 7 x 11 cubical building array (Brown et al. 2001) and a wide building street canyon (Kastner-Klein and Plate 1999). Röckle-type wind models, such as QUIC-URB, do not solve transport equations for momentum or energy; rather, they rely heavily on empirical parameterizations and

mass conservation. In the model-experiment comparisons, two empirical building flow parameterizations are tested within the QUIC-URB model: our team's implementation of the standard Röckle (SR) algorithms and a set of modified Röckle (MR) algorithms. The MR model attempts to build on the strengths of the SR model and introduces additional physically-based but simple parameterizations that significantly improve the results in most regions of the flow for two wind-tunnel test cases considered for its evaluation. The MR model produces vortices in front of buildings, on rooftops and within street canyons that have velocities that compare much more favorably to the experimental results.

1.2 GPU-based Lagrangian dispersion model-GPU Plume

There are a number of existing fast response dispersion models using various approaches to achieve the fast response goals (Gross et al. 1994; Kaplan and Dinar 1996; Gross 1997; Hall et al. 2000; Moussafir et al. 2004; Wang et al. 2005; Singh et al. 2008). These approaches can be broadly classified into Gaussian plume models (Robins 2001), Gaussian puff models (Sykes and Gabruk 1997), Eulerian models (Baik et al. 2003) and Lagrangian models (Kaplan and Dinar 1996; Wilson and Sawford 1996; Williams et al. 2004). Gaussian plume models (or "urbanized" Gaussian plume models) are used in urban areas by modifying standard input parameters such as the vertical and lateral dispersion functions to account for the effects of buildings (Robins 2001). Gaussian models are simple and generally produce results very quickly. They utilize a single wind direction and wind speed to estimate dispersion and transport of a contaminant. While these models have been shown to predict certain statistics such as maximum normalized

concentrations well (Hanna et al. 2003), complicated flow features encountered in urban areas such as varying wind directions and wind speeds limit the range of applicability of "urbanized" Gaussian plume models. Gaussian puff models (Sykes and Gabruk 1997), on the other hand, take temporally varying wind direction and variable wind speeds into account for describing dispersion and transport of a contaminant. These models are limited by their inability to capture the localized complex flow features encountered in the urban areas due to the presence of buildings and other structures (Coirier et al. 2005). In Eulerian transport and dispersion models, the conservation equations of mass and momentum along with an appropriate form of the scalar transport equation are solved to obtain dispersion estimates. These models are computationally intensive and are limited by their long turnaround times and nonclosure issues due to their nonlinear advection terms (Deardorff 1978; Wilson and Sawford 1996).

Lagrangian dispersion models, on the other hand, solve a much simpler form of mass conservation equation (Wilson and Sawford 1996). Particles with finite mass are advected under the influence of the instantaneous winds (Williams et al. 2004). The time derivative following the "marked" particles implicitly includes the nonlinear advection terms, without any approximations. The approximations which are required in the Lagrangian framework are only associated with the velocity field (e.g., turbulent stresses, velocity probability density function, see Thomson (1987)). Concentration evolution is a completely separate and exact process (Wilson and Sawford 1996). The concentration field is altered only due to the redistribution of the "marked" particles. The Lagrangian

approach is also very flexible as it is grid free and allows the user to run models both forward and backward in time (Yee and Wilson 2007).

Lagrangian particle dispersion models, coupled with a diagnostic wind model (such as QUIC-URB), have proved to be very successful for quickly describing mean concentration profiles around built-up urban areas (Kaplan and Dinar 1996; Bowker et al. 2007; Singh et al. 2008; Gowardhan et al. 2010). These models have a quick turnaround time and are therefore suitable for the emergency response applications. For applications such as modeling environmental flows in virtual environments, fast response models based on the Lagrangian approach are particularly useful. The Lagrangian framework provides a more natural approach to simulate particle dispersion (Thomson 1987) and therefore is preferred over high-order closure models owing to its ease and range of applicability. This approach can be easily extended to various real-life scenarios, such as simulating emission due to traffic, multiple sources of emission, variable particle size etc. which make it a preferred choice for virtual environment applications.

Effective visualizations and real-time computations are an integral part of a virtual reality system. To accomplish these goals, GPU Plume is developed on GPU-based architecture. GPUs have quickly developed from video game technology to open up new avenues for enhancing simulation performance and visualization of engineering and science applications. GPUs provide highly parallel and inexpensive data paths for processing geometry and pixels. In this research, a GPU-based Lagrangian dispersion model, GPU Plume, is developed and tested against an analytical solution, a CPU implementation of the Lagrangian dispersion model (QUIC-Plume) and wind-tunnel data

for dispersion around a single hi-rise building. Chapter 3 discusses this unique implementation and GPU Plume is shown to provide results that are similar in accuracy to the CPU model, but with computation times that are up to two orders of magnitude smaller. In addition, the challenges associated with the implementation of Lagrangian dispersion models onto the GPU architecture are also discussed.

1.3 Solving the generalized form of the Langevin equations (GLEs) using a fractional step method

Due to a large number of terms and stiffness (due to varied rate and time constants in the simulation) associated with the GLEs, advection in the GPU Plume model is accomplished by utilizing the Langevin equations simplified for the horizontally homogenous flows (Rodean 1996; Williams et al. 2004). That is, the Langevin equations are substantially reduced for flows having velocity gradients only in the vertical direction (e.g., atmospheric boundary layer). For the complex flow field found in the urban areas, the coordinate system is aligned with the mean velocity vector at each grid cell, using coordinate transformation (see Chapter 4 for details) to partially accommodate the horizontally homogenous assumptions (Williams et al. 2004). However, we hypothesize that these assumptions may not be valid for complex flow field found in the urban areas. Therefore, the full or generalized form of Langevin equations, GLEs without the horizontally homogenous assumptions, may be required for complex flow field encountered in the urban areas. The GLEs are considered to be stiff (Yee and Wilson 2007). The stiffness in the GLEs can be partially circumvented by using a fractional step

method (Yee and Wilson 2007), which is described in Chapter 4. Chapter 4 outlines the three steps associated with the fractional step method in detail and discusses how the fractional step method helps in revealing the unstable modes present in a particle's trajectory before the advection process begins. The challenges associated with the implementation of the GLEs are also discussed in Chapter 4. Chapter 4 details the current implementation of the GLEs and evaluates its results against available analytical solutions and a wind-tunnel test case for a 7 x 11 cubical array of buildings (Snyder and Lawson 1996). The concentration estimates from the GLEs-based model match the experiment data favorably in comparison with the SLEs-based model for this test case. The GLEs-based model seems to be more dispersive than the SLEs-based model. The increased lateral and vertical dispersion may be partially attributed to the presence of a large number of terms in the GLEs-based model.

1.4 References

Allwine, K. J., J. H. Shinn, G. E. Streit, K. L. Clawson, and M. Brown, 2002: Overview of URBAN 2000: A Multiscale Field Study of Dispersion through an Urban Environment. *Bulletin of the American Meteorological Society*, **83,** 521-536.

Allwine, K. J., M. L. Leach, L. W. Stockham, J. S. Shinn, R. P. Hosker, J. F. Bowers, and J. C. Pace, 2004: Overview of Joint Urban 2003-An atmospheric dispersion study in Oklahoma City. *Symp. on Planning, Nowcasting, and Forecasting in the Urban Zone*, Seattle, WA, Amer. Meteor. Soc., CD-ROM J7.1.

Baik, J.-J., J.-J. Kim, and H. J. S. Fernando, 2003: A CFD Model for Simulating Urban Flow and Dispersion. *Journal of Applied Meteorology*, **42,** 1636-1648.

Belcher, S. E., 2005: Mixing and transport in urban areas. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **363,** 2947-2968.

Berkowicz, R., 2000: A parameterized street pollution model. *Environmental Monitoring and Assessment*, **65,** 323-331.

Bowker, G. E., R. Baldauf, V. Isakov, A. Khlystov, and W. Petersen, 2007: The effects of roadside structures on the transport and dispersion of ultrafine particles from highways. *Atmospheric Environment*, **41,** 8128-8139.

Brook, D. R., and Coauthors, 2003: Validation of the Urban Dispersion Model (UDM). *International Journal of Environment and Pollution*, **20,** 11-21.

Brown, M., R. Lawson, D. DeCroix, and R. Lee, 2001: Comparison of centerline velocity measurements obtained around 2D and 3D building arrays in a wind tunnel. *Int. Soc. Environ. Hydraulics*, Tempe, AZ, 6.

Brown, M. J., 2004: Urban dispersion : challenges for fast response modeling. *Fifth AMS Symp. Urban Env*, Vancouver, BC, CD-ROM J5.1.

Chan, S., T. Humphries, and R. Lee, 2004: A simplified CFD approach for modeling urban Dispersion. *Symp. on Planning, Nowcasting, and Forecasting in the Urban Zone*, Seattle, WA, Amer. Meteor. Soc., CDROM 6.4.

Coirier, W., D. Fricker, M. Furmanczyk, and S. Kim, 2005: A Computational Fluid Dynamics Approach for Urban Area Transport and Dispersion Modeling. *Environmental Fluid Mechanics*, **5,** 443-479.

Coirier, W. J., and A. J. Reich, 2003: Oklahoma City high-resolution dispersion simulation. CFDRC Report 8520/1, 23 pp.

Dabberdt, W. F., F. L. Ludwig, and W. B. Johnson jr, 1973: Validation and applications of an urban diffusion model for vehicular pollutants. *Atmospheric Environment (1967)*, **7,** 603-618.

Deardorff, J. W., 1978: Closure of second- and third-moment rate equations for diffusion in homogeneous turbulence. *Physics of Fluids*, **21,** 525-530.

DeCroix, D., and M. J. Brown, 2002: Report on CFD model evaluation using URBAN 2000 field experiment data: IOP 10, release 1, 26 October, 2000. LA-UR-02-4755, 80 pp.

Doran, J. C., K. J. Allwine, K. L. Clawson, and R. G. Carter, 2006 Retention of tracer gas from instantaneous releases of SF6 in an urban environment. *14th Joint Conference on the Applications of Air Pollution Meteorology with the Air and Waste Management Assoc.*, Atlanta, GA, Amer. Meteor. Soc., CDROM 6.2.

Eerens, H. C., C. J. Sliggers, and K. D. van den Hout, 1993: The CAR model: The Dutch method to determine city street air quality. *Atmospheric Environment. Part B. Urban Atmosphere*, **27,** 389-399.

Genikhovich, E. L., and W. H. Snyder, 1994: A new mathematical model of pollutant dispersion near a building. *8th AMS/AWMA Conf. on Appl. Air. Pollut. Meteorol.*, Nashville, TN, 254-261.

Gowardhan, A., M. Brown, and E. Pardyjak, 2010: Evaluation of a fast response pressure solver for flow around an isolated cube. *Environmental Fluid Mechanics*, **10,** 311-328.

Gross, G., 1997: ASMUS-Ein numerisches model zur berechnung der strömung und der schadstoffverteilung im bereich einzelner gebaude. *II Schadstoffausbreitung und anwendung Meteorol Zeitschrift*, **6,** 130-136.

Gross, G., R. Röckle, and U. Janssen, 1994: ASMUS-Ein numerisches model zur berechnung der strömung und der schadstoffverteilung im bereich einzelner gebäude. *I Das strömungsfeld Meteorol Zeitschrift*, **3,** 267-274.

Hall, D., A. Spanton, I. Griffiths, M. Hargrave, and S. Walker, 2000: The UDM: A model for estimating dispersion in urban areas. Tech Report No. 03/00 (DERA-PTN-DOWN).

Hanna, S. R., R. Britter, and P. Franzese, 2003: A baseline urban dispersion model evaluated with Salt Lake City and Los Angeles tracer data. *Atmospheric Environment*, **37,** 5069-5082.

Hanna, S. R., R. M. Reynolds, J. Heiser, and R. Bornstein, 2004: Plans for MSG04 tracer experiment in Manhattan. *5th AMS Symp. Urban Env.*, Vancouver, BC.

Hanna, S. R., and Coauthors, 2006: Detailed Simulations of Atmospheric Flow and Dispersion in Downtown Manhattan: An Application of Five Computational Fluid Dynamics Models. *Bulletin of the American Meteorological Society*, **87,** 1713-1726.

Hanna, S. R., J. White, Y. Zhou, and A. Kosheleva, 2006: Analysis of JU2003 and MSG05 meteorological and tracer data. *6th AMS Symp. Urban Env.*, Atlanta, GA.

Kaplan, H., and N. Dinar, 1996: A lagrangian dispersion model for calculating concentration distribution within a built-up domain. *Atmospheric Environment*, **30,** 4197-4207.

Kastner-Klein, P., and E. J. Plate, 1999: Wind-tunnel study of concentration fields in street canyons. *Atmospheric Environment*, **33,** 3973-3979.

Lim, D. W., D. S. Henn, and P. A. Hookham, 2001: Preliminary assessment of linked CFD operational dispersion models on the urban scale. *3rd Int. Symp. on Environmental Hydraulics*, Tempe, AZ.

McElroy, J. L., 1969: A Comparative Study of Urban and Rural Dispersiont. *Journal of Applied Meteorology*, **8,** 19-31.

Moussafir, J., O. Oldrini, G. Tinarelli, J. Sontowski, and M. Dougherty, 2004: A new operational approach to deal with dispersion around obstacles: the MSS (Micro Swift Spray) software system. *9th Int. Conf. Harm. Atm. Disp. Mod. Reg. Purp.*, Garmisch, Germany.

Nelson, M. A., B. Addepalli, D. Boswell, and M. J. Brown, 2006: The QUIC v. 4.5 Start Guide. LA-UR-07-2799.

Pullen, J., J. P. Boris, T. Young, G. Patnaik, and J. Iselin, 2005: A comparison of contaminant plume statistics from a Gaussian puff and urban CFD model for two large cities. *Atmospheric Environment*, **39,** 1049-1068.

Ramsdell, J. V., and C. J. Fosmire, 1995: Atmospheric dispersion estimates in the vicinity of buildings. Report no PNL-10286, Pacific Northwest National Laboratory.

Robins, A., 2001: Review of flow and dispersion in the vicinity of group of buildings. University of Surrey.

Röckle, R., 1990: Bestimmung der Stromungsverhaltnisse im Bereich komplexer Bebauungsstrukturen . Ph.D. dissertation, Vom Fachbereich Mechanik, der Technischen Hochschule Darmstadt, Germany.

Rodean, H. C., 1996: *Stochastic Lagrangian models of turbulent diffusion.* The American Meteorological Society, 82 pp.

Rotach, M. W., 2001: Simulation Of Urban-Scale Dispersion Using A Lagrangian Stochastic Dispersion Model. *Boundary-Layer Meteorology*, **99,** 379-410.

Schulman, L. L., D. G. Strimaitis, and J. S. Scire, 2000: *Development and evaluation of the PRIME plume rise and building downwash model.* Vol. 50, Air & Waste Management Association.

Singh, B., B. Hansen, M. Brown, and E. Pardyjak, 2008: Evaluation of the QUIC-URB fast response urban wind model for a cubical building array and wide building street canyon. *Environmental Fluid Mechanics*, **8,** 281-312.

Smith, W. S., and M. J. Brown, 2002: A CFD generated wind field library feasibility study: maximum wind direction interval. *4th AMS Symp. Urban Env.*, Norfolk, VA.

Snyder, W. H., and R. Lawson, 1996: Wind-tunnel measurements of flow fields in the vicinity of buildings. *8th Conf. on Air Poll. Met.*, Amer. Met. Soc.

Sykes, R. I., and R. S. Gabruk, 1997: A second-order closure model for the effect of averaging time on turbulent plume dispersion. *J. Appl. Meteor.*, **36,** 1038-1045.

Theurer, W., E. J. Plate, and K. Hoeschele, 1996: Semi-empirical models as a combination of wind tunnel and numerical dispersion modelling. *Atmospheric Environment*, **30,** 3583-3597.

Thomson, D. J., 1987: Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *Journal of Fluid Mechanics*, **180,** 529-556.

Venkatram, A., V. Isakov, D. Pankratz, J. Heumann, and J. Yuan, 2004: The analysis of data from an urban dispersion experiment, **38,** 3647-3659.

Wang, Y., C. Williamson, D. Garvey, S. Chang, and J. Cogan, 2005: Application of a Multigrid Method to a Mass-Consistent Diagnostic Wind Model. *Journal of Applied Meteorology*, **44,** 1078-1089.

Williams, M. D., M. J. Brown, D. Boswell, B. Singh, and E. M. Pardyjak, 2004: Testing of the QUIC-plume model with wind-tunnel measurements for a high-rise building. *5th AMS Urban Env. Conf.*, Vancouver, BC.

Wilson, D. J., and E. H. Chui, 1994: Influence of building size on rooftop dispersion of exhaust gas. *Atmospheric Environment*, **28,** 2325-2334.

Wilson, J. D., and B. L. Sawford, 1996: Review of Lagrangian stochastic models for trajectories in the turbulent atmosphere. *Boundary-Layer Meteorology*, **78,** 191-210.

Yamartino, R., D. Strimaitis, and T. Messier, 1989: Modification of highway air pollution models for complex site geometries. Data analyses and development of the CPB-3 model. FHWA-RD-89-112.

Yee, E., and J. Wilson, 2007: Instability in Lagrangian stochastic trajectory models, and a method for its cure. *Boundary-Layer Meteorology*, **122,** 243-261.

2. QUIC-URB: FAST RESPONSE URBAN WIND MODEL

2.1 Introduction

QUIC-URB is a diagnostic wind model which generates a mass consistent spatially explicit mean wind field in urban areas with complex building configurations. Based on the work done by Röckle (1990) and Kaplan and Dinar (1996), QUIC-URB uses empirical parameterizations to specify the flow field physics on a gridded domain and enforces mass consistency on the initial flow field to obtain a final flow field (Singh et al. 2008; Gowardhan et al. 2010).

In this chapter, the QUIC-URB wind model is described in detail, including the standard Röckle (1990) empirical parameterizations used for describing the flow physics around buildings. The standard Röckle empirical parameterizations are then evaluated against mean velocity measurements obtained from a wind-tunnel experiment of flow around a 7 x 11 array of cubes (Snyder and Lawson 1996). The shortcomings of the standard Röckle parameterizations are highlighted and a new street canyon algorithm for flow normal to street canyons is presented and compared against the standard Röckle model and the wind-tunnel experimental data. The new street canyon parameterization is also evaluated against wind-tunnel data obtained for a street canyon region formed between two wide buildings (Kastner-Klein and Plate 1999).

Although a number of evaluation studies have been performed looking at the performance of Röckle-style modeling systems using concentration measurements from tracer experiments, relatively few detailed evaluations have included high resolution wind measurements. This chapter evaluates the new empirical parameterizations implemented in the wind model for upwind cavity, rooftop recirculation and street canyon flow fields against wind-tunnel data at fairly high spatial resolution for an incident flow normal to a 7 x 11 array of cubes. This section is followed by detailed description of the QUIC-URB model including the standard Röckle empirical parameterizations. The results section presents a comparison between the QUIC-URB model and the velocity measurements of 7 x 11 cubic array of buildings as well as a wide buildings street canyon data set using both the standard Röckle algorithms and our new empirical parameterizations.

2.2 QUIC-URB model description

The modeling strategy adopted in QUIC-URB was originally developed by Röckle (1990) and uses a 3D mass consistent wind model to explicitly resolve time-averaged wind fields around buildings. The mass consistent technique is based on Sherman (1978) 3D complex terrain diagnostic wind model. The basic methodology involves generating an initial wind field ($\vec{V}^o = u^o\hat{i} + v^o\hat{j} + w^o\hat{k}$) that includes various empirical parameterizations to account for the physics of flow around buildings and then forcing this velocity field to be divergence free subject to the weak constraint such that the variance of the difference between the initial velocity field and mass consistent final velocity field ($\vec{V} = u\hat{i} + v\hat{j} + w\hat{k}$) is minimized. This is done using a general variational

analysis formalism originally developed by Sasaki (1958, 1970a,b) in which Eq. (2.1) is minimized.

$$E(u,v,w,\lambda)=\int_V\left[\alpha_1^2\left(u-u^o\right)^2+\alpha_1^2\left(v-v^o\right)^2+\alpha_2^2\left(w-w^o\right)^2+\lambda\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)\right]dxdydz \quad (2.1)$$

In Eq. (2.1), λ are Lagrange multipliers (with units of inverse time) and α_i are Gaussian precision moduli (weighting factors with units of inverse velocity). As noted by Kaplan and Dinar (1996), α_i are variables that enhance or restrict the correction of the wind components with respect to each other. In QUIC-URB, single building tests (Bagal 2005) confirmed that for most cases, $\alpha_i = 1$ produces optimal results in neutral stability flows.

The velocity field is updated using the Euler-Lagrange equations whose solution minimizes Eq. (2.1). Namely,

$$u = u^o + \frac{1}{2\alpha_1^2}\frac{\partial\lambda}{\partial x} \quad (2.2a)$$

$$v = v^o + \frac{1}{2\alpha_1^2}\frac{\partial\lambda}{\partial y} \quad (2.2b)$$

$$w = w^o + \frac{1}{2\alpha_2^2}\frac{\partial\lambda}{\partial z} \quad (2.2c)$$

These equations are subject to the boundary conditions $\partial\lambda/\partial n = 0$ at a solid boundary (where n is the outward normal direction) and $\lambda = 0$ at inflow/outflow boundaries. An equation for λ is obtained by differentiating Eqs. (2.2) and substituting

the result into the continuity equation for the final velocity field, $\nabla \cdot \vec{V} = 0$ (Sherman 1978). This procedure results in the following Poisson equation that can easily be solved for λ using the specified boundary conditions:

$$\frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial^2 \lambda}{\partial y^2} + \left(\frac{\alpha_1}{\alpha_2}\right)^2 \frac{\partial^2 \lambda}{\partial z^2} = -2\alpha_1^2 \left(\frac{\partial u^o}{\partial x} + \frac{\partial v^o}{\partial y} + \frac{\partial w^o}{\partial z}\right). \tag{2.3}$$

In QUIC-URB, Eq. (2.3) is solved using a simple iterative successive over-relaxation or SOR solver (Press et al. 2007) on a nonuniform staggered grid where velocities are face center values and Lagrange multipliers are cell-centered quantities.

The ability of the QUIC-URB model to produce accurate wind fields around buildings is dependent on the empirical wind parameterizations. These parameterizations introduce rotation into the flow field and without these parameterizations, the method is essentially a potential flow solver. The upwind boundary-layer profile may be specified as a power-law, log-law, urban canopy or user-specified profile. This profile is applied uniformly in portions of the domain that are not affected by building flow features. For problems in which the flow varies spatially outside of the urban area (e.g., complex terrain), multiple data profiles or point measurements can be assimilated to produce a spatially-varying wind field (Booth and Pardyjak 2006). As described by Gowardhan et al. (2010), for isolated buildings, QUIC-URB utilizes a number of empirical building algorithms for determining the initial wind fields of the vortex regions associated with the building rooftop (Bagal et al. 2004b; Pol et al. 2006), the upstream recirculation zone (Bagal et al. 2004a) and the downwind recirculation cavity and the velocity deficit wake

(Röckle 1990). Details of these models are given in Gowardhan et al. (2010), but here few points relevant to this test case are highlighted. The standard Röckle upstream recirculation zone model is an elliptical volume with all initial velocity field components specified to be zero. For flow normal to a building face, the improved QUIC-URB model separates the region upstream of the building into two elliptical regions: a displacement zone where the velocities are reduced and a recirculation zone where the velocities are specified to form a vortex. The original Röckle formulation did not contain a rooftop recirculation parameterization to account for separation at the leading edge of a building. The complete details of the improved QUIC-model are described in Gowardhan et al. (2010) for both normal and off-angle winds. For normal incident winds, a rooftop vortex region is specified following Wilson (1979). The velocity field in the vortex is applied uniformly across the width of the rooftop, and takes on a maximum negative value just above the building rooftop and then increases monotonically to the upstream boundary layer velocity at the top of the cavity zone. The improved QUIC-model includes a rooftop recirculation region with logic to determine when a rooftop recirculation cavity is necessary. This is quite important for groups of buildings. For example, in the 7 x 11 test case presented here, only the buildings in the first row have a rooftop recirculation zone. For flow normal to a building face, the rooftop algorithm logic always applies a rooftop cavity if the building is isolated and far from other buildings or if the building is greater in height than the nearby upwind building.

2.2.1 Standard Röckle street canyon model

In this section, our implementation of the Röckle (1990) street canyon (SC) algorithm for urban flows with multiple buildings (hereafter referred to as SR for standard Röckle model) is described. The empirical SC algorithm accounts for the generation of a classical SC vortex that forms between two closely spaced buildings (Oke 1987). While 3D SC flow is extremely complicated, it is convenient (and conventional) to break the flow into three phenomenological flow regimes based on the spacing between the buildings: isolated roughness flow, wake interference flow and skimming flow (Hussain and Lee 1980). Following the notation in Fig. 2.1, these regimes roughly correspond to cubical building ratios of spacing (S) to building height (H) of about: $S/H > 2.5$, $1.4 < S/H < 2.5$ and $S/H < 1.4$, respectively (Oke 1987). Because the wake interference flow regime is unsteady and difficult to parameterize, Röckle assumed that the canyon flow could be simply modeled by two flow regimes: skimming and isolated flow.

The decision criteria used to determine which flow regime to implement is based on a nondimensional building spacing parameter that is a function of street canyon and building geometries. In our implementation, canyon flow is parameterized by two flow regimes similar to Kaplan and Dinar (1996): skimming (when $S/H < 1.25 + 0.15(W/H)$ for $W/H < 2$ and $S/H < 1.55$ for $W/H \geq 2$) and isolated flow $S/H > 1.25 + 0.15(W/H)$, where W is the crosswind width of the building. In the isolated flow regime, parameterizations for the upwind, rooftop and wake cavities are applied in the same manner as for the case when there are no other buildings in the domain. In the skimming regime, a reverse flow is imposed between the buildings below

roof level (see Fig. 2.1a and 2.1b). The imposed reverse flow interacts with the boundary layer flow at the sides of the street canyon to form two counter-rotating vortices. The streamwise and wall normal initial velocity specification for the reverse flow within the canyon is given by

$$\frac{u^o(x,y,z)}{u^o(H)} = -\frac{x_{can}}{(0.5S)}\left(\frac{S-x_{can}}{0.5S}\right), \tag{2.4}$$

and the vertical component is given by

$$\frac{w^o(x,y,z)}{u^o(H)} = -\left|\frac{1}{2}\left(1-\frac{x_{can}}{0.5S}\right)\right|\left(1-\frac{S-x_{can}}{0.5S}\right). \tag{2.5}$$

Here, x_{can} is the distance from the backwall of the upwind building, and the other parameters are defined in Fig. 2.2.

For flow that is not normal to the SC, the velocity component normal to the axis of the SC is specified by Eq. (2.4) and the component parallel to the SC is left unchanged, resulting in a "channeling velocity." The delineation of the SC zone is shown in Fig. 2.1c. The next section describes the SR model evaluation against the wind-tunnel data and highlights the various shortcomings present in the SR model.

2.3 Evaluation of the standard Röckle model against a 7 x 11 building array wind-tunnel data set

2.3.1 Wind-tunnel experiment description

The wind-tunnel data used here for validation are from the work of Brown et al. (2001). The experiments were carried out in a 3.7 m wide, 2.1 m high and 18.3 m long open-return meteorological wind-tunnel at the U.S. Environmental Protection Agency's Fluid Modeling Facility (Snyder 1979). The cubical building array examined in this study consisted of eleven rows of blocks in the streamwise direction and seven columns of blocks in the crosswind direction (Fig. 2.3). The building array was oriented perpendicular to the inflow wind. The blocks were of equal height, width and length ($H = W = L = 150$ mm) and were spaced $S = H$ apart in the along-wind and crosswind directions. As discussed above, with a space-to-height (S/H) ratio of unity, the 7 x 11 array of cubes should be in the skimming flow regime (Oke 1987). The building models were immersed in a simulated 1.8 m deep neutral atmospheric boundary layer which was created using spires near the tunnel entrance (Irwin 1981) and floor roughness elements. Using a length scale equal to H and a reference velocity of 3 ms^{-1} at $z = H$, the Reynolds number was approximately 30,000, well above the critical value required for Reynolds number independence (Castro and Robins 1975; Snyder 1981). The building height was less than 10% of the boundary-layer depth, similar to the ratio in real downtown areas. While no specific scale ratio was chosen, a representative value would be 250:1; hence, the building models would correspond to full-scale buildings on the order of 30 to 40 m in height.

A hot-wire anemometer with an X-array sensor was used to measure the mean velocity and turbulence intensity profiles of the approach flow in the absence of any buildings. To account for reversed flow and high turbulence intensity within the building array, measurements were made with a pulsed-wire anemometer (PWA, Bradbury and Castro (1971)). All PWA measurements were obtained using a pulsing rate of 10Hz and an averaging time of 120 seconds at each measurement location. More information on the experiment can be found in Lawson et al. (2000) and Brown et al. (2001).

2.3.2 Description of the model test case

In the section that follows, the standard Röckle (SR) model is compared to the wind-tunnel data described in Section 2.3.1. Matching the inlet profile proved to be somewhat difficult because a 500 mm smooth wall gap existed between the upstream roughness elements and the start of the 7 x 11 array in the wind-tunnel. In this region, the experimental data showed the development of an internal boundary layer that QUIC-URB is unable to simulate. Hence, the inlet profile was specified at x/H = -3.3 such that the solution matched the experimental data at x/H = -1.5 as closely as possible. To match the data, the inlet velocity profile was specified to be logarithmic with a roughness length of 2 mm and a reference velocity U_{ref} = 2.82 ms^{-1} at the building height (H = 0.15 m). Figure 2.4 shows the comparison of the velocity profile from the model and the experimental data at x/H = -1.5. The profile is described well with the logarithmic fit shown (2.2% RMS error). The boundary conditions on the velocity at the inlet, outlet and along the top of the domain are Dirchlet and specified by the initial logarithmic profile. While QUIC has a variable grid resolution capability, the simulations were run with a

uniform grid resolution of 0.015 m such that the buildings were resolved with 10 cells in each direction. Simulations were also run at double and half of this resolution and the results for the finer grid were quite similar (< 1% difference in RMS error) to the 0.015 m grid.

The experimental measurements were made in a subset of the 7 x 11 array. The region where measurement comparisons have been made to the simulations is highlighted in Fig. 2.3a. Experience with the type of diagnostic urban wind model discussed here indicates that the last column of buildings on either side of the array does not affect the simulated centerline velocities. Hence, these buildings have been omitted from the simulations and the building array of 5 x 5 (25 buildings) shown in Fig. 2.3b was used for all of the simulations. The domain size used for the simulations was 2.295 m x 1.650 m x 0.450 m (153 x 110 x 30 in grid cell units). The test case presented here took ~26 seconds to run on a 2.4 Ghz Intel Core 2 Duo Processor with 2 GB of random access memory (RAM).

2.3.3 Standard Röckle model evaluation – 7 x 11 array

In this section, the performance of the SR model is evaluated qualitatively by comparing general flow features. Figure 2.5a shows a velocity vector comparison between the experimental data and the SR model in the vertical plane along the centerline at the beginning of the building array. The SR model does a reasonably good job of predicting the location of the stagnation point on the upwind face of the first-row building ($z/H \sim 0.7$). The experimental data reveal a small, but well-defined recirculation zone

upstream of the first building, whereas the SR model results in a large unorganized cavity with no well-defined features (Fig. 2.5a and Fig. 2.5b).

The experimental data suggest that a recirculation zone may exist above the rooftop of the first building (Fig. 2.5a). As expected, the SR model - which does not contain a rooftop recirculation scheme - overestimates the streamwise velocity above the rooftop.

Looking at the flow in the first street canyon (Fig. 2.5a), the SR model produces somewhat stronger downdrafts and backflow as compared to the experimental data. Moreover, the center of the SC vortex simulated by our implementation of the SR model is raised well above the height yielded in the experimental data. The SR scheme results in winds that too quickly revert to the purely horizontal flow above the canyon, whereas the measurements show a significant downward component at the midpoint of the canyon. Figure 2.5c depicts the wind patterns in the first canyon near ground level at $z/H = 0.2$. The SR scheme generates counter-rotating vortices in agreement with the measurements; however, the modeled winds are too strong and the vortex centers are too close to the street canyon ends as compared to the experimental data.

2.4 Modified Röckle street canyon model

To address the various shortcomings present in our implementation of the SR model, a modified Röckle street canyon model (hereafter referred to as MR for modified Röckle model) has been developed and evaluated prior to incorporating it into the operational version of the QUIC modeling system. Our implementation of the SR model does not account for diffusion of streamwise momentum into the SC from aloft, resulting

in a sharp transition between the street canyon velocity and the air aloft. It also tends to overpredict the velocities within the central part of the canyon. The modified SC algorithm for the skimming flow regime suggested here extends the Röckle model to include parameterizations that more effectively approximate the physics observed in field and wind-tunnel data. As shown in Fig. 2.2 and 2.6, the SC is broken up into three physically-based regimes: (i) a central canyon region dominated by the classical SC vortex, (ii) a vertical turbulent diffusion region associated with the transport of momentum into and out of the canyon from above, and (iii) a horizontal turbulent diffusion region associated with the lateral transport of momentum into and out the canyon. The vertical and horizontal diffusion regions are defined by triangular prisms or wedges that extend from the leeward edge of the upstream building. Conceptually, the flow within each of these wedges is modeled as a single stream shear layer (or mixing layer). The width of the mixing region within plane mixing layers is well known to grow linearly with distance downstream of the start of the layer and to have a velocity profile that takes on a hyperbolic tangent shape (Pope 2000). We use these physically-based concepts to develop a model for the mixing region at the edges of the SC.

As illustrated in Fig. 2.2, the vertical wedge is a right-angled wedge with length *S*, and maximum height, $\delta_{vw}\big|_{x_{can}=S} = 0.2S$ (Pope 2000). The vertical wedge extends across the entire width *W* of the street canyon. Within the wedge, the velocity is specified to behave similar to a classical single-stream shear layer and is given by the following hyperbolic tangent model (Morris and Foss 2003):

$$u^o(x,y,z) = U_{roof}\,\frac{\tanh\left[\left(\delta_{vw}(x_{can}) - z_{can}\right)/\,\delta_{vw}(x_{can})\right]}{\tanh(1.0)} \tag{2.6}$$

Here, the vertical wedge depth $\delta_{vw}(x_{can}) = 0.2x_{can}$ is a linear approximation to the depth of the shear layer. U_{roof} is a reference velocity in the streamwise direction obtained by generating a displaced logarithmic profile over the canyon. The displacement height (*d*) is taken as the height of the shortest building making up the street canyon and the reference wind direction is given by the local wind direction at the center of the canyon at rooftop level of the shortest building. We recognize that typical measured values of the displacement height are closer to *d/H* ~ 0.7 (Jackson 1981); however, for modeling simplicity, we use *d/H* = 1. Within the wedge, the vertical winds are specified to be zero. Currently, for flow that is not normal to the SC, the MR model is identical to the SR model described above.

Similarly, the lateral wedge is also a right-angled wedge with maximum width $\delta_{lw}\big|_{x_{can}=S} = 0.2S$. The streamwise velocity in the shear layer is specified using the following hyperbolic tangent model:

$$\frac{u^o(x,y,z)}{u^o_{bl}(z)} = \gamma\left[\frac{\tanh\left[Y_{lw}/\delta_{lw}(x_{can})\right]}{\tanh(1.0)}\right] \tag{2.7}$$

In Eq. (2.7), $u^o_{bl}(z)$ is the upstream boundary layer velocity that is unaffected by buildings and $\gamma = 0.3$ is a velocity reduction correction factor that has been empirically determined with the present 7 x 11 data set to account for the SR model's overprediction of the strength of the velocities in the canyon. $Y_{lw} = \left[\left|y_{can}\right| - \delta_{SC}(x_{can})/2\right]$,

$\delta_{SC}(x_{can}) = W - 2\delta_{lw}(x_{can})$ and $\delta_{lw}(x_{can}) = 0.2x_{can}$, where $\delta_{lw}(x_{can})$ is the width of the lateral wedge. As in the vertical wedge, the vertical component of the velocity is set to zero.

As shown in Eqs. (2.4) and (2.5), the original SR model does not explicitly parameterize a lateral variation in the wind speed within the street canyon. Including a lateral diffusion wedge provides a smooth transition of the flow between the wedge and the street canyon interior (see cross hatched region in Fig. 2.2) that approximates the momentum diffusion process. This lateral diffusion is accomplished by modeling the along-wind component of the velocity within the street canyon core to incorporate lateral variation:

$$\frac{u^o(x,y,z)}{u^o(H)} = -\gamma \frac{x_{can}}{(0.5S)}\left(\frac{S - x_{can}}{0.5S}\right) F_{SC}(y_{can}) \tag{2.8}$$

In Eq. (2.8), γ is the same velocity reduction factor from Eq. (2.7) and is applied throughout the canyon. F_{SC} is a continuously-varying function that reduces the velocity in the lateral direction with distance from the center of the canyon and is given by:

$$F_{SC}(y_{can}) = \left[1 - \frac{|y_{can}|}{\delta_{SC}(x_{can})/2}\right]^p \tag{2.9}$$

The empirical coefficient p is a parameter that was adjusted to improve the final comparison with the 7 x 11 data set. The best match to the experimental data was obtained with the exponent p set to 0.25. Note that the algorithms are applied in

sequential order: the central SC is calculated first, then the lateral wedge and finally the vertical wedge; in this process, the velocities computed with the later algorithms overwrite the earlier ones in regions of overlap.

Another change that we have made to our implementation of the SR algorithm is a modification to the criteria to determine the existence of a street canyon; it is now based on the single building wake recirculation cavity length formula of Fackrell (1984), namely

$$\frac{S^*}{H} = \frac{1.8\frac{W}{H}}{\left(\frac{L}{H}\right)^{0.3}\left(1+0.24\frac{W}{H}\right)}. \tag{2.10}$$

Here, S^* is the length of recirculation cavity in the wake of an isolated building, L is the streamwise length of the upwind building in the lateral direction and W is the width of the building in the crosswind direction. If $S < S^*$, then a street canyon vortex flow parameterization is implemented; otherwise, the building is assumed to be isolated and cavity and wake algorithms are utilized (Kaplan and Dinar 1996).

2.4.1 Modified Röckle model evaluation – 7 x 11 array

In this section, the performance of the MR model is evaluated both qualitatively by comparing general flow features and quantitatively through point-by-point mean velocity comparisons. This write-up focuses on specific regions of the flow field beginning with the region just upwind of the first row of the 7 x 11 cube array, followed

by the rooftop zone, and ending with the flow in the first street canyon. The next section begins by looking at the overall flow patterns around the first two rows of buildings.

2.4.1.1 General comparison of the mean flow field

Figure 2.7a shows a velocity vector comparison between the experimental data and the MR model in the vertical plane along the centerline at the beginning of the building array. Similar to the case of the SR model, the MR model does a reasonably good job of predicting the location of the stagnation point on the upwind face of the first-row building ($z/H \sim 0.7$). Upwind of the first building, the MR upwind cavity algorithm significantly improved the results by producing a smaller recirculation cavity, which is in agreement with the experimental data. The plan view in Fig. 2.7b illustrates the flow structure improvement of the MR model over the SR model (Fig. 2.5b). The MR scheme agrees well with the strength and direction of the winds that were measured on the front side near ground level at $z/H = 0.2$. The SR scheme (Fig. 2.5b), however, shows an overly large region of near-zero winds upwind of the building that disagrees with the measurements.

While the experimental data do not show the recirculation region expected along the rooftop of the first building (possibly due to a lack of spatial resolution of the measurements), the velocities decay substantially and show vertical velocity components. As compared to the SR model (Fig. 2.5a), the MR model (Fig. 2.7a) improves the results by producing a rooftop recirculation zone and more realistic updrafts and downdrafts. Figure 2.7a shows that the rooftop recirculation scheme is correctly turned off on row 2 (and beyond) through logic integrated into the MR model.

As compared with the SR model (Fig. 2.5a), the SC vortex computed by the MR model (Figure 2.7a) matches the wind-tunnel data better by reducing the SC vortex strength. The MR street canyon model predicts the center of the vortex to be slightly closer to the experimental data due to the wedge scheme that mimics diffusion of winds from aloft, although it is still shifted to the right compared to the data. Both the SR as well as the MR schemes produce winds that revert to purely horizontal flow too quickly, while the measurements show a significant downward component at the midpoint of the canyon. Figures 2.5c and 2.7c depict the wind patterns in the first canyon near ground level at $z/H = 0.2$. Both the SR and MR schemes generate counter-rotating vortices in agreement with the measurements. As compared to the SR scheme (Fig. 2.5c), the MR scheme (Fig. 2.7c) matches the strength of the wind and the vortex location better, in part due to the lateral diffusion wedges and the lateral velocity gradient described above.

2.4.1.2 Upstream flow field

Figure 2.8 shows a velocity profile comparison of the normalized a) streamwise and b) vertical velocities upstream of the building along the centerline of the domain. The streamwise and vertical velocity measurements show that the upwind recirculation zone starts at about $x/H \sim -0.5$. As shown in the Fig. 2.7 vector plot, the MR model predicts the upwind extent to be $x/H \sim -0.7$. It is clear from Fig. 2.8 that the upwind cavity zone computed by the SR model is apparent much further upstream (x/H = -1.0), where near zero velocities are found between the ground and $z/H \sim 0.25$. At this upwind distance, both the experimental data and the MR model show a logarithmic behavior in the streamwise velocity.

At $x/H = -0.5$, the SR model predicts a near-zero streamwise velocity between the ground and $z/H \sim 0.5$ (Fig. 2.8a). The experimental data and the MR model, however, continue to show a positive velocity gradient in this region, although the MR model shows more reduction in wind speed compared to the measurements. Above $z/H \sim 0.5$, the experimental data, the SR model and the MR model are all in agreement. At $x/H = -0.25$ and -0.1, the SR model continues to produce near-zero streamwise velocities below *z/H* = 0.6. In contrast, the experimental data and the MR model show reverse flow near the ground. At $x/H = -0.1$, significant reverse flow is still not apparent near the ground in the SR model results. It is interesting to note that at this distance, the MR model has a smoother transition into the boundary-layer flow above the building height, likely due to upstream propagation of the effect of the rooftop recirculation found in the MR model. Unphysical kinks in both the SR and MR model-produced streamwise velocity profiles at *x/H* = - 0.1 and -0.5 are found at the transition between the upwind cavity zone and ambient flow due to a lack of momentum diffusion in Röckle-style models.

The upstream vertical velocity profiles at $x/H = -1$ demonstrate that the SR model slightly overestimates the vertical velocity from $0.3 < z/H < 1$, whereas the MR model and the experimental data are near-zero and in much better agreement. A small kink in the vertical velocity profiles (also seen in the *u* velocity in Fig 2.8a) is observed in both the SR and MR model near *z/H*~0.6 at $x/H = -0.5$ as a result of insufficient model diffusion between the interface of the recirculation zone and the boundary layer. Both the SR and MR models underestimate the magnitude of the vertical velocities near the ground at $x/H = -0.25$ and -0.1, but both models are in general agreement with the

experimental results above the stagnation point location, where a strong updraft is observed.

Table 2.1 summarizes the RMS error difference between the MR and SR models for the available experimental data upwind of the first building. The average error in the MR model is approximately 60% of the SR model.

2.4.1.3 Rooftop flow field

Figure 2.9 shows model-produced and measured vertical profiles of the normalized a) streamwise and b) vertical velocities at the rooftop of the first building along the centerline of the domain. As the upwind flow strikes the front building, the flow separates from the rooftop. The separated flow reattaches near the end of the rooftop forming a recirculation region. As noted earlier, the SR model does not simulate a recirculation region on the rooftop. Hence, the streamwise velocity profile in Fig. 2.9a confirms that the SR model overestimates the streamwise velocities above the rooftop, while MR model produces better agreement with the experimental data for the u velocity at all streamwise locations due to the rooftop recirculation algorithm. At x/H = 0.3, 0.5 and 0.7, the MR model shows a slight increase in the streamwise velocity at about $z/H \sim$ 1.25 above the rooftop compared to the SR model, but there is not enough experimental data to corroborate this feature. Further downstream (x/H = 0.7 and 0.9) just above the rooftop (z/H = 1.1), the MR model overpredicts the strength of the recirculation. It is interesting to note that at this resolution, the experimental data do not actually show any reverse flow.

Vertical velocities along the rooftop are shown in Fig. 2.9b. The SR model underestimates the vertical velocities (updraft strength) above the rooftop at x/H = 0.1 and 0.3, whereas the experimental data and the MR model are in fair agreement. At x/H = 0.5, 0.7 and 0.9 the SR model predicts slightly positive vertical velocities, while the data show a downdraft. The MR model produces a very small slightly positive vertical velocity at x/H = 0.5, while the measurements show a very small negative velocity near the rooftop. The MR scheme predicts the vertically velocity very well at x/H = 0.7, but overpredicts the strength of the downdraft at the end of the building (x/H = 0.9).

Figure 2.10 shows a comparison of vertical profiles of normalized streamwise velocity above the rooftop of the second building along the centerline of the domain. Due to the shielding effect and advection from the rooftop of the first building, the flow striking the second building does not form a recirculation region above the rooftop. The flow forms a wall-normal logarithmic layer on the rooftop of the second building. The streamwise velocity comparison in Fig. 2.10 shows that the SR model slightly overestimates the velocities above the rooftop, while the MR model generates velocities that agree better with the experimental data. As shown in Table 2.1, the average RMS error of the MR model is roughly half that of the SR model for the available profiles.

2.4.1.4 Street canyon flow field

Figure 2.11 shows a comparison of vertical profiles of normalized a) streamwise and b) vertical velocities in the first SC along the centerline of the domain. As demonstrated earlier in Fig. 2.5a, our implementation of the SR model produces a SC vortex with an elevated core, a region of streamwise flow above the vortex core that is

too small and a region of backflow below the vortex core that is too large compared to the measurements. As shown in Fig. 2.11a, the SR model is in reasonably good agreement with the experimentally measured streamwise velocity within the SC below $z/H \sim 0.6$ at all locations, although at $x/H = 1.25$ and 1.5, the SR model slightly overestimates the magnitude of backflow in the canyon. Near z/H~1, the SR model results in a strong shear in streamwise velocity due to the lack of downward diffusion of streamwise momentum from aloft. The MR model is in excellent agreement with the streamwise velocity data in the canyon and is in better agreement than the SR model near the building height (z/H~1). The streamwise velocities calculated using the MR model yield a smoother transition from the canyon flow to the boundary-layer flow aloft due to the incorporation of the wedge diffusion parameterization.

The vertical profiles of the vertical velocities shown in Fig. 2.11b at $x/H = 1.1$ and 1.25 indicate that both models overestimate the vertical velocities, although the MR model performs better in the vicinity of the canyon top. Near $z/H \sim 1$, both models give rise to vertical velocities of opposite sign compared to the experimental data. This is a result of an overprediction of the height of the center of the canyon vortex as illustrated in the vector plots in Figs. 2.5 and 2.7. At $x/H = 1.5$, the SR and MR models generate near zero w velocities as opposed to the negative velocities obtained in the experiment. This is a result of the models predicting a more symmetric vortex about the canyon center, while the experimental data indicate the center of the vortex is shifted slightly upstream. Both the SR and MR models overestimate the w velocity within the canyon at $x/H = 1.75$ and 1.9. A significant improvement is seen in the results of the MR model at $x/H = 1.75$ and

1.9 compared to the SR model, where the *w* velocity follows the experimental results more closely between *z/H* ~ 0.6 and *z/H* ~ 1.6.

Figures 2.5c, 2.7c and 2.12-2.15 show velocity vector and profile plots within and just outside of the first street canyon for three different horizontal planes above the ground (*z/H* = 0.2, 0.5 and 0.8). Figure 2.5c and 2.7c show the plan view of the model-computed and measured velocity vectors at *z/H* = 0.2 for the SR and MR models, respectively. The SR model's SC parameterization significantly overestimates the magnitude of the backflow in the SC near the ground (Fig. 2.5c). This result stems partly from the strong lateral flux of momentum into the street canyon from the street intersection ($|y|/H > 1$) and partly from the lack of a laterally-varying velocity reduction factor in the SR model. The SC parameterization in the MR model accounts for the advection and diffusion from the intersections as well as from above the SC; this shifts the center of the vortices well inside the SC (see Fig. 2.7c). The MR model shows a significant improvement in simulating the strength and direction of the mean flow measured in the canyon.

Figure 2.12 shows a comparison of lateral velocity profiles of normalized a) streamwise and b) crosswind velocities in the first SC at *z/H* = 0.2. As shown in Fig. 2.12a, the SR model creates a strong streamwise velocity gradient near the sides of the SC, while the lateral diffusion associated with the MR model produces a smoother profile that is in better agreement with the experimental data in the interior of the canyon and outwards into the intersection ($|y|/H > 0.5$). The SR model slightly underestimates the streamwise velocities near the center of the canyon at *x/H* = 1.25 and 1.5, while at *x/H* = 1.1, 1.75 and 1.9, the SR model is in agreement with the experimental data. However,

the curvature of the SR velocity profile is opposite to the MR model and the experimental data in the canyon.

The crosswind velocity profiles in Fig. 2.12b show that the MR model captures the lateral flow better than the SR model at *z/H* = 0.2. As explained earlier, the SR model significantly overestimates the magnitude of the crosswind velocity from $0.2 < |y| / H < 0.8$ at *x/H* = 1.1 and 1.25. Downstream of the canyon center, an opposite trend in the crosswind velocity profile is observed in the SR model at *x/H* = 1.75 and 1.9. The SR model agrees with the experimental data in the interior of the canyon ($|y| / H < 0.2$) at *x/H* = 1.1 and 1.25. The magnitude of the lateral velocity computed by the MR model is in better agreement with the measurements at all the streamwise locations, but of opposite sign from $1.1 < x / H < 1.75$.

Figures 2.13 and 2.14 show plan view comparisons of the velocity vectors at *z/H* = 0.5 and *z/H* = 0.8, respectively, from the (a) SR and (b) MR models with the experimental data. The plots clearly illustrate the behavior of the wall normal vortices formed in the SC. Similar to the *z/H* = 0.2 case, the SR model significantly overpredicts the magnitude of the velocities within the SC and poorly predicts both the streamwise and lateral locations of the center of the wall normal vortices. Again, due to the lack of lateral diffusion into the street canyon associated with the SR model, the simulated velocity vectors near the end of the SC do not match the experimental results in direction or magnitude. The MR model, however, significantly improves the direction as well as magnitude of velocity vectors at the lateral ends of the SC compared to SR simulation. The downstream centers of the lateral vortices in the experimental data show a systematic shift upstream with height above the canyon floor. This behavior is similar to the

"rainbow vortex" identified in the wake of an isolated cube by Hunt et al. (1978). This phenomena, which is associated with the change in flow direction near x/H = 1.5 (in the z/H=0.8 slice), is a result of downward diffusion of momentum that is better captured in the MR model than in the SR model.

Figure 2.15 shows a comparison of the lateral velocity profiles of the normalized streamwise velocities in the first SC at (a) z/H = 0.5 and (b) z/H = 0.8. The comparison of the models with the experimental data is quite similar to the descriptions given above for z/H = 0.2 with overly strong shear at the lateral edges of the canyon and incorrect prediction of the curvature of the velocity profile. However, the underestimation of the u velocity by the SR model within the SC becomes much more pronounced higher up in the canyon and further downstream, while the MR model agrees quite well with the experimental data. As shown in Table 2.1, the average RMS error in the SC for the MR model is slightly less than half that of the SR model for the available profiles.

2.4.2 Wide building test case

2.4.2.1 Wind-tunnel experiment

As a second evaluation test case, a wide building street canyon experimental data set was utilized. The data were obtained by researchers in the Institute of Hydromechanics at University of Karlsruhe (for details, see Kastner-Klein and Plate (1999); Kastner-Klein (1999)). The researchers used a 2 m wide by 1 m high test section in a neutrally stratified atmospheric boundary layer wind-tunnel. Two building rows were mounted on the floor of the test section surrounded by homogenously distributed roughness elements. The street canyon was oriented normal to the incident wind. The

buildings were wide with widths ten times the building height (i.e., W=10H; H=L), and the distance between the buildings was S=H (see Fig. 2.16). The boundary layer flow was generated by employing vortex generators at the tunnel entrance and roughness elements on the floor. According to Kastner-Klein and Plate (1999), the inlet mean velocity profile can be described by a power law with an exponent of 0.23. The Reynolds number for this flow was approximately 56,000 with a reference velocity of 7 ms^{-1} at a reference height (H) of 0.12 m. The measurements were taken using Laser-Doppler velocimetry (LDV) and a single hotwire.

2.4.2.2 Description of the test case

Following Kastner-Klein and Plate (1999), a power law profile with an exponent of 0.23 was specified as the inlet velocity profile for running the SR and MR models. Similar to the 7 x 11 building array test case, the velocity boundary conditions at the inlet, outlet and along the top of the domain were Dirchlet and specified by the initial power law profile. The simulations were run with a grid resolution 0.012 m such that the buildings were resolved with 10 cells in each direction. The domain size used for the simulations was 0.960 m x 1.680 m x 0.240 m (80 x 140 x 20 in grid cell units).

2.4.2.3 Model-measurement comparison

Unfortunately, this wide building data set did not included measurements immediately upstream or along the rooftop of the buildings; hence, this section only discusses the comparison between the results obtained from the SR model, MR model

and the wind-tunnel experiment in the street canyon region between the two wide buildings.

Figure 2.17 shows a velocity vector comparison between the experimental data, the a) SR and b) MR models in the vertical plane along the centerline of the domain. Like the 7 x 11 array test case, the SR model produces stronger downdrafts and backflow as compared to the experimental data for the wide building case. The center of the SC vortex computed using the SR model is raised well above the height of the vortex center indicated by the experimental data. The MR model predicts the center of the vortex to be slightly closer to the experimental data due to the wedge scheme that mimics the diffusion of winds from aloft, although it is still higher than the experimental data.

Figure 2.18 shows a comparison of vertical profiles of normalized a) streamwise and b) vertical velocities in the SC along the centerline of the domain. As in the 7 x 11 test case, the SR model produces a SC vortex with an elevated core and a region of streamwise flow above the vortex core that does not penetrate far enough down into the canyon. As shown in Fig. 2.18a, the SR model is in reasonably good agreement with the experimentally-measured streamwise velocity within the SC below $z/H \sim 0.6$ at all locations. Near $z/H \sim 1$, the SR model produces large streamwise velocity gradients due to the lack of downward diffusion of streamwise momentum from aloft. Similar to the SR model, the MR model is in agreement with the streamwise velocity data within the canyon. The MR model, however, is in better agreement than the SR model near the building height ($z/H \sim 1$). The MR model yields a smoother transition of streamwise velocities from the canyon to the boundary-layer flow aloft due to the incorporation of the wedge diffusion parameterization.

The vertical profiles of the vertical velocities shown in Fig. 2.18b at x = -0.25 indicate that both models slightly underestimate the vertical velocities below z/H~0.4. The MR model performs slightly better than the SR model from $0.6<z/H<1.4$. In addition, the SR model overpredicts the updraft close to the height of the building (z/H~1.0). At x/H = 0, both the models underestimate the vertical velocity within the canyon, yielding similar results; however, the SR model performs better than the MR model above the street canyon region. This is likely the result of an overly intense rooftop recirculation region on the upwind building. At x/H = 0.25, both models slightly overestimate the vertical velocities below z/H~0.4. The MR model performs better than the SR model from $0.6<z/H<1.4$, where the SR model overpredicts the downdraft velocities.

Figure 2.19 shows a comparison of lateral profiles of normalized streamwise velocities over half of the SC at z/H = 0.25. As shown in Fig. 2.19, the SR model creates a strong streamwise velocity gradient near the sides of the SC due to the absence of lateral diffusion, while the MR model produces a smoother profile that is in better agreement with the experimental data at the sides of the canyon (y/H~-5.0). Both models are in agreement with the experimental data within the SC; however, they overestimate the streamwise velocity outside of the canyon (y/H<-5.0). The MR model performs slightly better than the SR model outside of the canyon. Similar to the 7 x 11 array case, the curvature of the SR velocity profile is opposite to the MR model and the experimental data in the canyon. As summarized in Table 2.2, the RMS error of the MR model is about ~30% less than the error associated with the SR model. While this is not quite as good as the results from the 7 x 11 array, it represents a substantial improvement.

2.5 Conclusions

Transport and dispersion in urban environments is extremely complicated. Buildings alter the flow fields and deflect the wind, causing updrafts and downdrafts, channeling between buildings, areas of calm winds adjacent to strong winds, and horizontally and vertically rotating-eddies between buildings, at street corners and other places within the urban canopy (see review by Hosker (1984)). This makes it very difficult to devise fast response urban dispersion models that will work at the street canyon to neighborhood scales.

(Röckle 1990) developed a methodology for quickly computing 3D wind fields around buildings using an empirical-diagnostic approach. The Röckle modeling strategy is a unique and potentially powerful tool because it rapidly produces spatially-resolved wind fields in urban areas that can be used to drive urban dispersion models. Röckle-type models do not solve transport equations for momentum or energy; rather, they rely heavily on empirical parameterizations and mass conservation. In this chapter, we evaluate a fast-running wind model that is based on the Röckle formulism called QUIC-URB using wind measurements from two wind-tunnel data sets: an idealized 7 x 11 cubical building array and a wide building street canyon. In the model-experiment comparison, we test two empirical building flow parameterizations within the QUIC-URB model: the standard Röckle (SR) algorithms and the modified Röckle (MR) algorithms.

To our knowledge, this is the most rigorous comparison of a Röckle-type wind model in the literature. The results indicate that our implementation of the SR model produces wind fields that are in reasonable agreement with experimental data within the

urban street canyon; however, the velocities are generally too strong and the location of vortex centers in various planes are shifted toward the edges of the street canyon where the wind shear is highest. Upstream of the first building, the SR model produces a recirculation cavity that is larger than experimentally expected and poorly predicts the velocities in this region. Above the buildings along the rooftop, the SR does not account for the potential of rooftop recirculation.

The MR model attempts to build on the strengths of the SR model and introduces additional physically-based but simple parameterizations that significantly improve the results in most regions of the flow in the 7 x 11 array and wide street canyon. The MR model produces vortices within street canyons that have velocities that compare much more favorably to the experimental results with the vortices shifted inward away from the edges of the street canyon. This is largely accomplished by modeling the effect of advection and momentum diffusion from outside the street canyon into the street canyon on the sides and from aloft. Upstream of the first building, a reduced velocity displacement zone and simple trigonometric vortex parameterization produce greatly improved results. Above the first rooftop, the rooftop recirculation zone improves the results. In addition, logic that removes the recirculation zone from downstream buildings produces physically realistic results in groups of buildings.

We expect that these improvements in the wind field will result in improved dispersion calculations in built environments. As a final note, we stress the importance of testing multibuilding parameterizations under a wide range of nonidealized conditions. Since it is quite rare that buildings in real cities take on the form that the original parameterizations were developed from, the model may not yield physically reasonable

results when generalized. Hence, it is imperative to rigorously evaluate the model for a wide range of scenarios. This is one of the greatest challenges in utilizing Röckle type wind models.

2.6 References

Bagal, N., E. R. Pardyjak, and M. J. Brown, 2004a: Improved upwind cavity parameterization for a fast response urban wind model. *Symp. on Planning, Nowcasting, and Forecasting in the Urban Zone*, Seattle, WA, Amer. Meteor. Soc., CD-ROM P1.13.

Bagal, N., B. Singh, E. R. Pardyjak, and M. J. Brown, 2004b: Implementation of rooftop reciculation parameterization into the QUIC fast response urban wind model. *Fifth AMS Symp. Urban Env.*, Vancouver, BC, Amer. Meteor. Soc., CD-ROM 6.10.

Bagal, N. L., 2005: Development and testing of empirical parameterization for QUIC urban and industrial complex model. M.S. Thesis, Dept. of Mechanical Engineering, University of Utah, 92 pp.

Booth, T. M., and E. R. Pardyjak, 2006: Validation of a data assimilation technique for an urban wind model. *Fifth AMS symposium urban environment*, Atlanta, GA, Amer. Meteorol. Soc., CDROM J4.9.

Bradbury, L. J. S., and I. P. Castro, 1971: A pulsed-wire technique for velocity measurements in highly turbulent flows. *Journal of Fluid Mechanics*, **49,** 657-691.

Brown, M., R. Lawson, D. DeCroix, and R. Lee, 2001: Comparison of centerline velocity measurements obtained around 2D and 3D building arrays in a wind tunnel. *Int. Soc. Environ. Hydraulics*, Tempe, AZ, 6.

Castro, I. P., and A. G. Robins, 1975: *The effect of a thick incident boundary layer on the flow around a small surface mounted cube.* Central Electricity Generating Board, Research Dep., Marchwood Engineering Laboratories.

Fackrell, J. E., 1984: Parameters characterising dispersion in the near wake of buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, **16,** 97-118.

Gowardhan, A., M. Brown, and E. Pardyjak, 2010: Evaluation of a fast response pressure solver for flow around an isolated cube. *Environmental Fluid Mechanics*, **10,** 311-328.

Hosker, R. P., 1984: Flow and diffusion near obstacles. In Atmospheric Science and Power Production (edited by D. Randerson), 241-326. Publication DOE/TIC-27601, Technical Information Centre, U. S. Department of Energy, Washington, D.C.

Hunt, J. C. R., C. J. Abell, J. A. Peterka, and H. Woo, 1978: Kinematical studies of the flows around free or surface-mounted obstacles; applying topology to flow visualization. *Journal of Fluid Mechanics*, **86,** 179-200.

Hussain, M., and B. E. Lee, 1980: A wind tunnel study of the mean pressure forces acting on large groups of low-rise buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, **6,** 207-225.

Irwin, H., 1981: The design of spires for wind simulation. *Journal of Wind Engineering and Industrial Aerodynamics*, **7,** 361-366.

Jackson, P. S., 1981: On the displacement height in the logarithmic velocity profile. *Journal of Fluid Mechanics*, **111,** 15-25.

Kaplan, H., and N. Dinar, 1996: A lagrangian dispersion model for calculating concentration distribution within a built-up domain. *Atmospheric Environment*, **30,** 4197-4207.

Kastner-Klein, P., 1999: Experimentelle Untersuchung der strömungsmechanischen Transportvorgänge in Straßenschluchten. Ph.D. dissertation, Diss. Fakultät für Bauingeneur- und Vermessungswesen, der Universität Karlsruhe (TH).

Kastner-Klein, P., and E. J. Plate, 1999: Wind-tunnel study of concentration fields in street canyons. *Atmospheric Environment*, **33,** 3973-3979.

Lawson, R., S. Perry, and R. Thompson, 2000: Measurement of velocity and concentration fields in arrays of 2-dimensional and 3-dimensional buildings in a simulated neutrally-buoyant atmospheric boundary layer. U.S. Environmental Protection Agency FMF Internal Data Rep., 55 pp.

Morris, S. C., and J. F. Foss, 2003: Turbulent boundary layer to single-stream shear layer: the transition region. *Journal of Fluid Mechanics*, **494,** 187-221.

Oke, T. R., 1987: *Boundary layer climates.* 2nd ed. Routledge, London, 435 pp.

Pol, S. U., N. L. Bagal, B. Singh, M. J. Brown, and E. R. Pardyjak, 2006: Implementation of a new rooftop recirculation parameterization into the QUIC fast response urban wind model. *6th AMS Symp Urban Env*, Atlanta, GA.

Pope, S. B., 2000: *Turbulent Flows.* Cambridge University Press, 771 pp.

Press, W. H., S. A. Teukolskay, W. T. Vetterling, and B. P. Flannery, 2007: *Numerical Recipes: The Art of Scientific Computing.* 3rd ed. Cambridge Univ. Press, 1235 pp.

Röckle, R., 1990: Bestimmung der Stromungsverhaltnisse im Bereich komplexer Bebauungsstrukturen . Ph.D. dissertation, Vom Fachbereich Mechanik, der Technischen Hochschule Darmstadt, Germany.

Sasaki, Y., 1958: An objective analysis based on the variational method. *Journal of the Meteorological Society of Japan. Ser. II*, **36,** 77.

Sasaki, Y., 1970a: Numerical variational analysis formulated under the constraints as determined by longwave equations and a low-pass filter. *Monthly Weather Review*, **98,** 884-898.

——, 1970b: Some basic formalisms in numerical variational analysis. *Monthly Weather Review*, **98,** 875-883.

Sherman, C. A., 1978: A Mass-Consistent Model for Wind Fields over Complex Terrain. *Journal of Applied Meteorology*, **17,** 312.

Singh, B., B. Hansen, M. Brown, and E. Pardyjak, 2008: Evaluation of the QUIC-URB fast response urban wind model for a cubical building array and wide building street canyon. *Environmental Fluid Mechanics*, **8,** 281-312.

Snyder, W. H., 1979: The EPA meteorological wind tunnel: its design, construction and operating characteristics. U.S. Environmental Protection Agency, Office of Research and Development, Environmental Sciences Research Laboratory.

Snyder, W. H., 1981: Guideline for fluid modeling of atmospheric diffusion. Report no EPA-600/8-81-009, Environmental Protection Agency, Research Triangle Park, NC. 200.

Snyder, W. H., and R. Lawson, 1996: Wind-tunnel measurements of flow fields in the vicinity of buildings. *8th Conf. on Air Poll. Met.*, Amer. Met. Soc.

Wilson, D. J., 1979: Flow patterns over flat-roofed buildings and application to exhaust stack design. *ASHRAE Transactions*, **84,** 284--295.

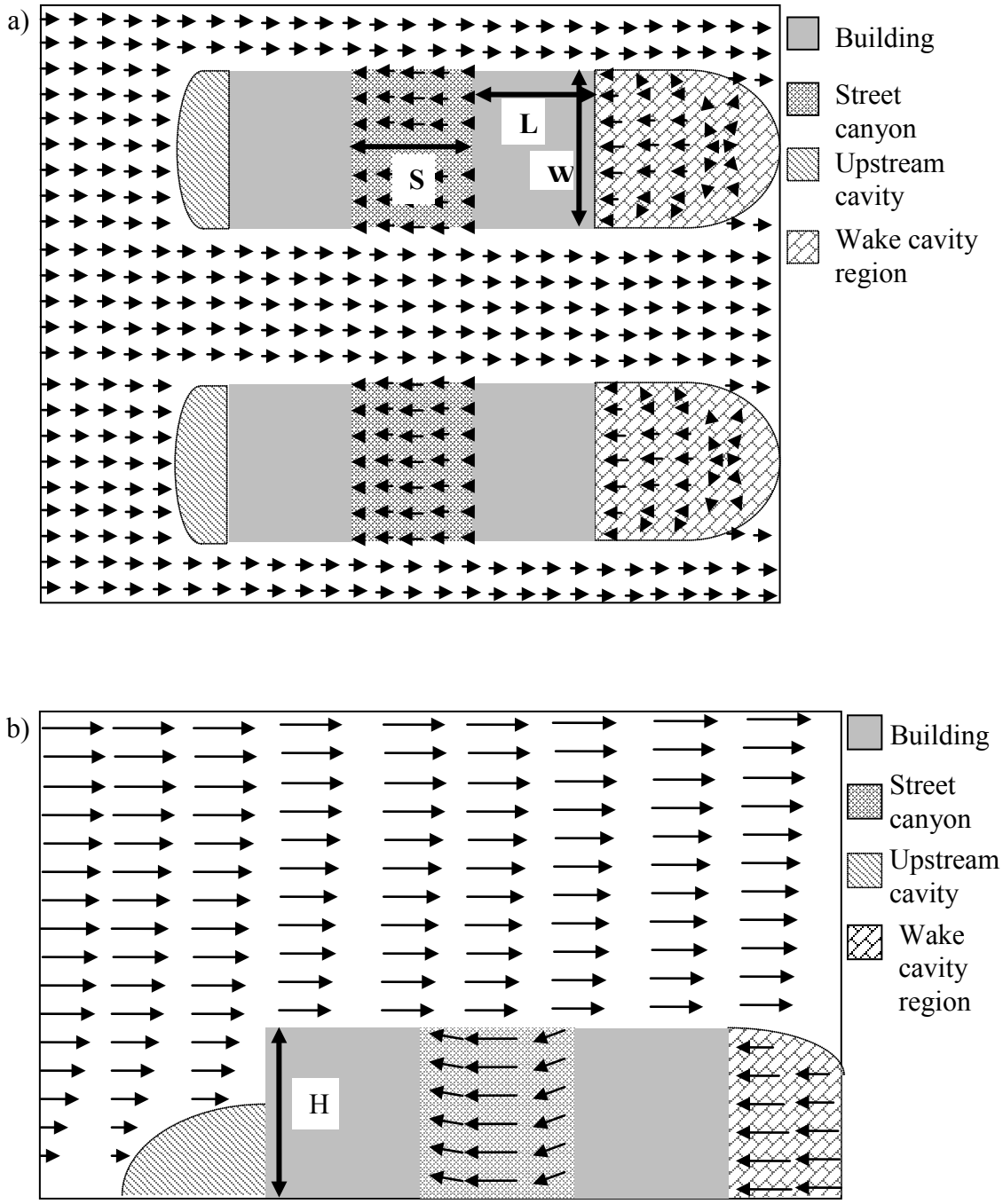


Figure 2.1: Schematic showing the various flow regions and initial velocity field (prior to mass conservation) associated with the Röckle parameterizations in the a) horizontal plane for normal flow b) vertical plane for normal flow c) horizontal plane for off angle flow.

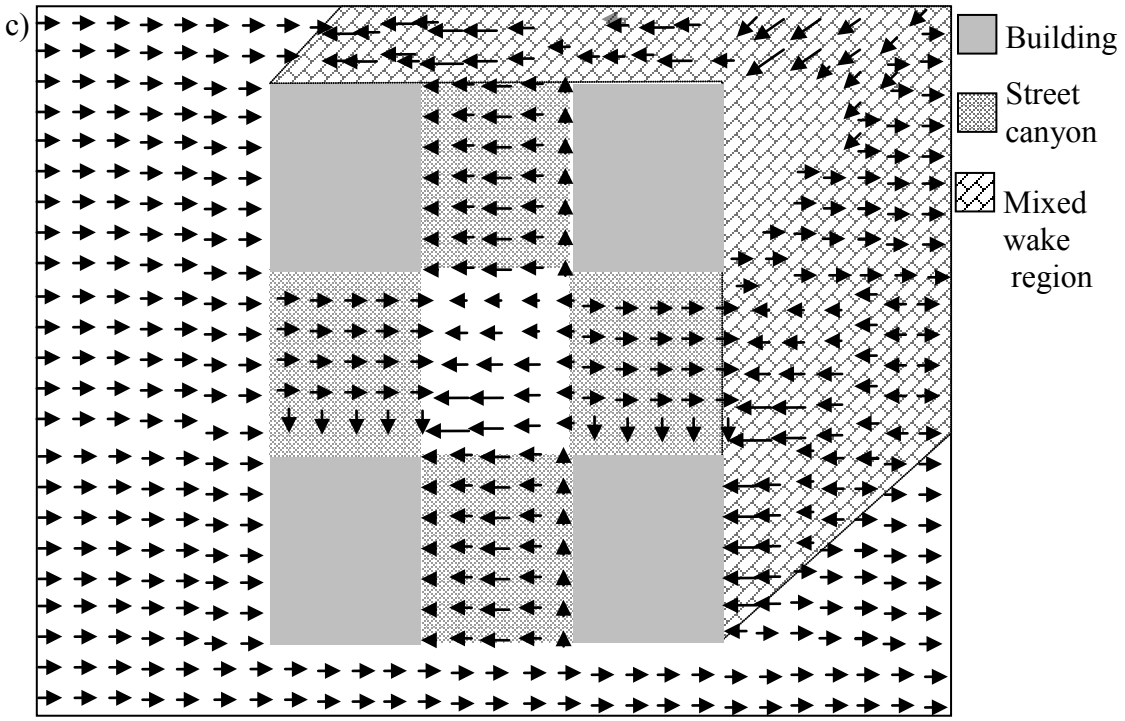


Figure 2.1: Continued.

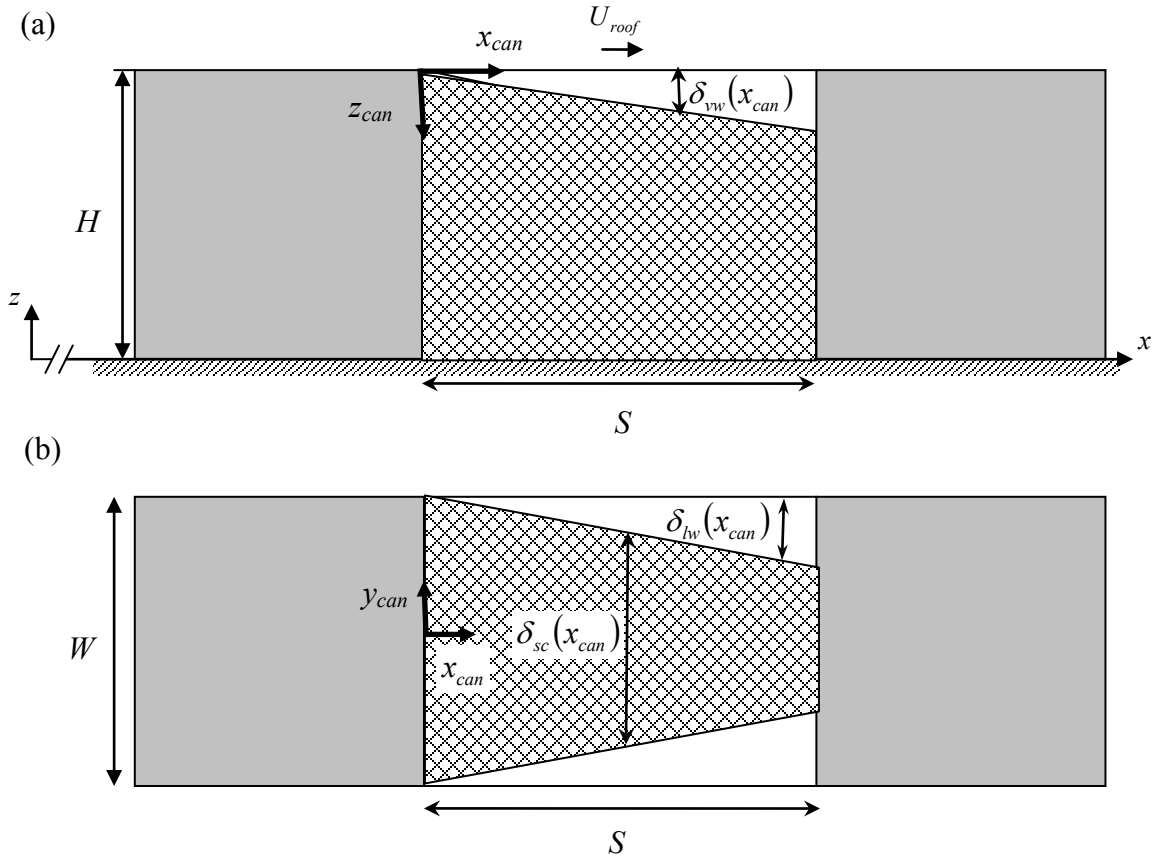


Figure 2.2: Schematic showing the notation and local coordinate system for the modified Röckle (MR) street canyon model (a) side view (b) plan view.

a)

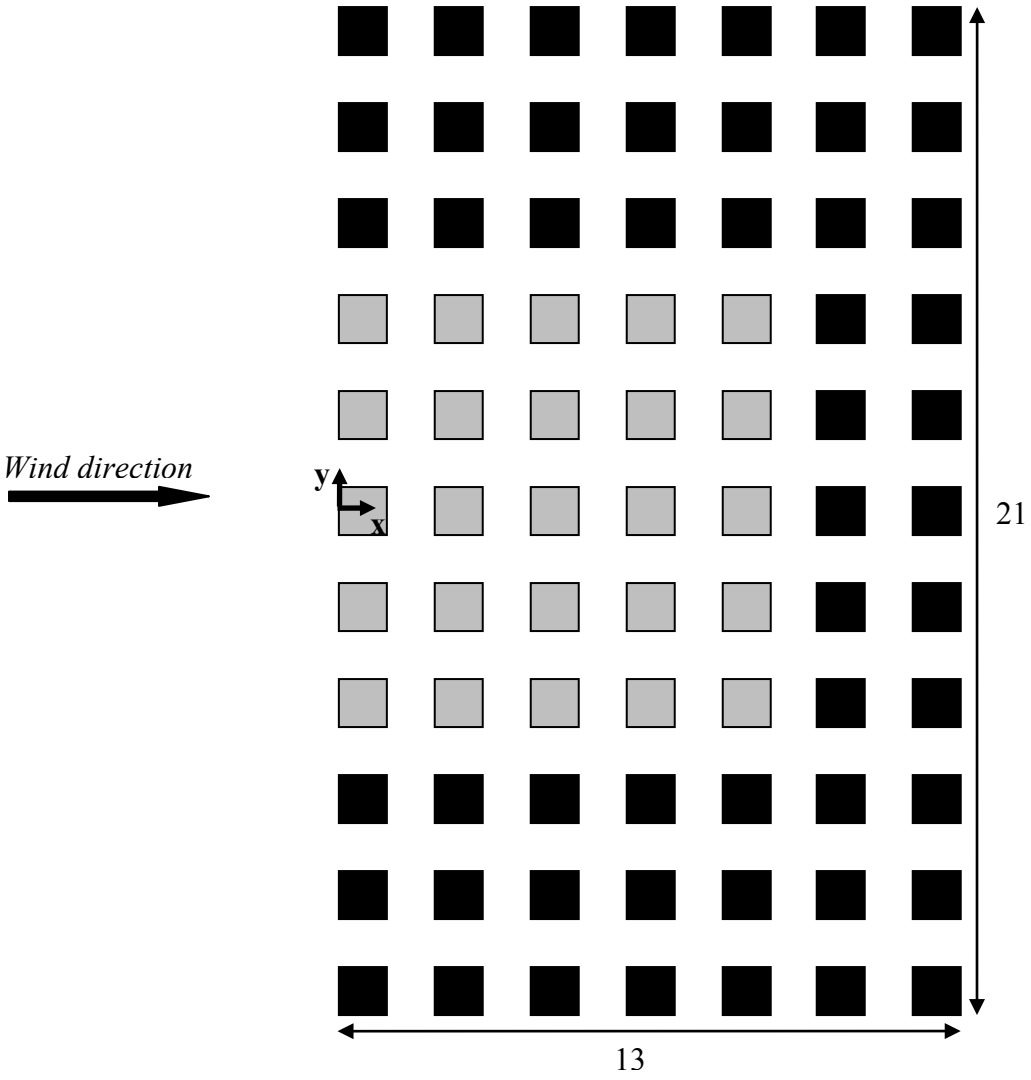


Figure 2.3: Schematic of the 7 x 11 building array used in the wind tunnel study. a) The shaded area represents the region where the simulations have been compared to the experimental data in the text. b) Schematic of the 5 x 5 building array used in the QUIC-URB simulations.

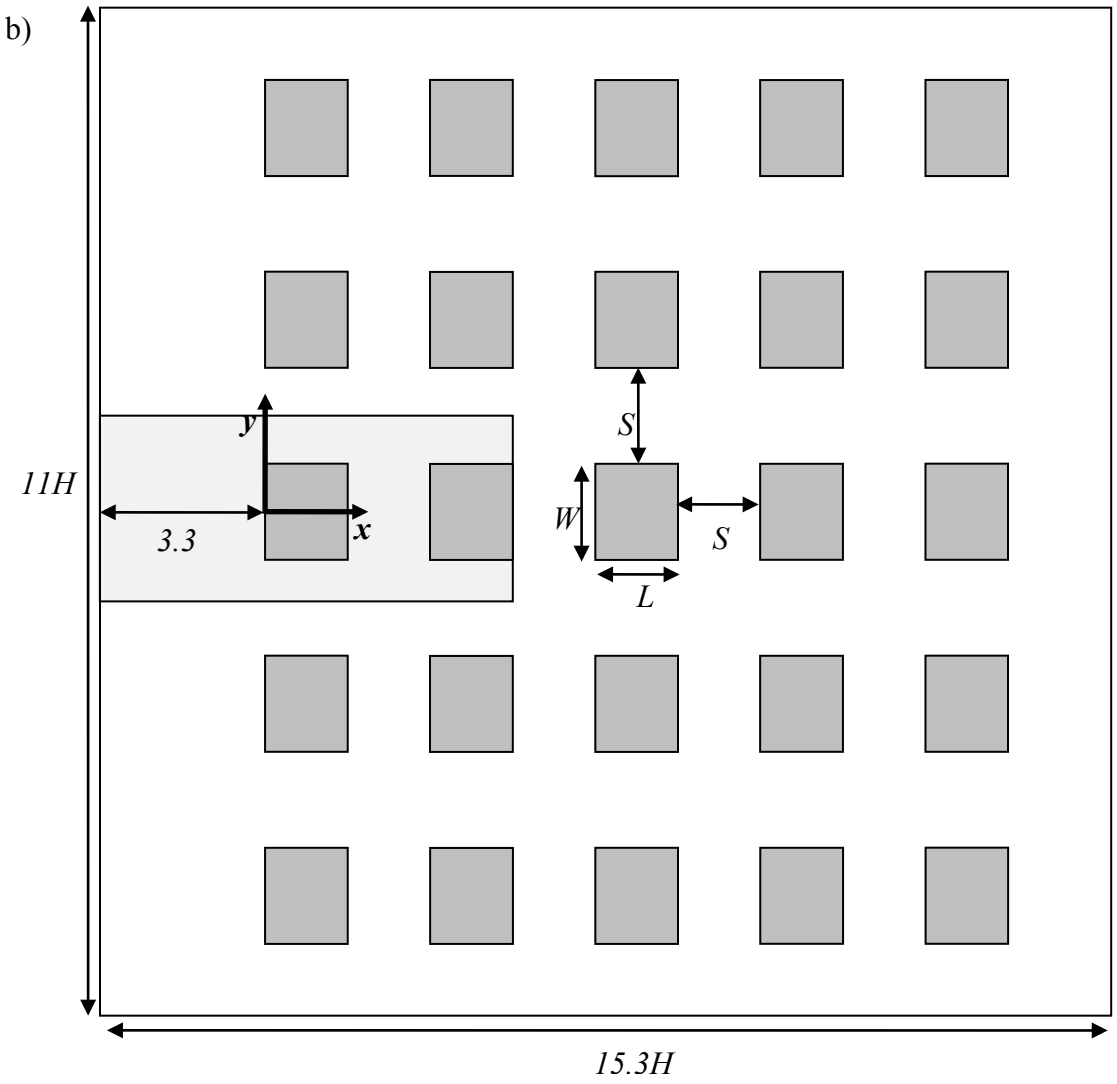


Figure 2.3: Continued.

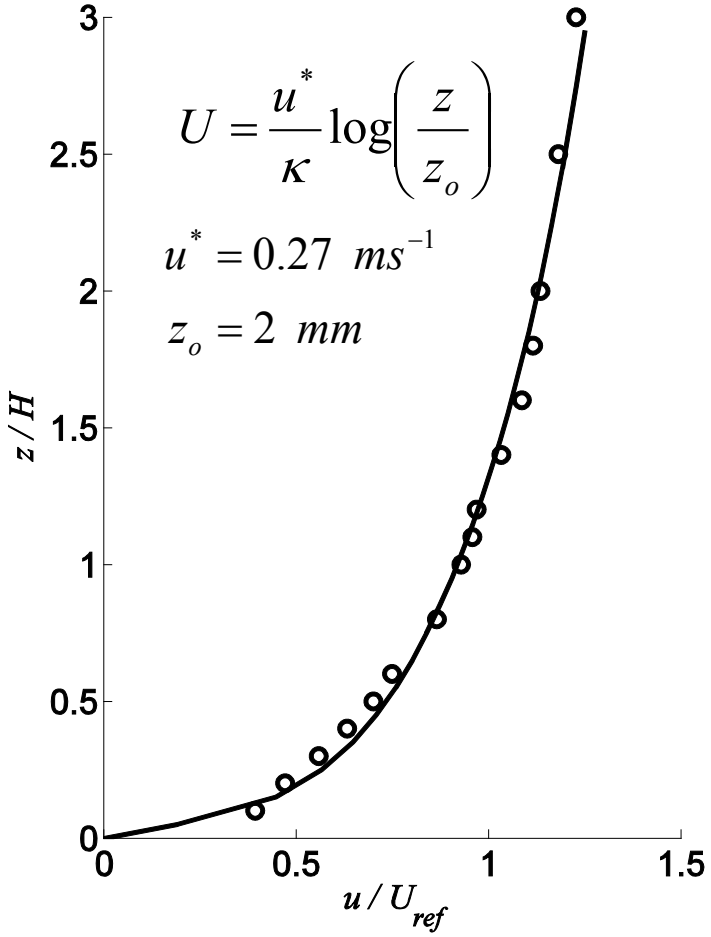


Figure 2.4: Inflow profile comparison between the experimental data and the logarithmic velocity profile used to initialize QUIC-URB at x/H = -1.5.

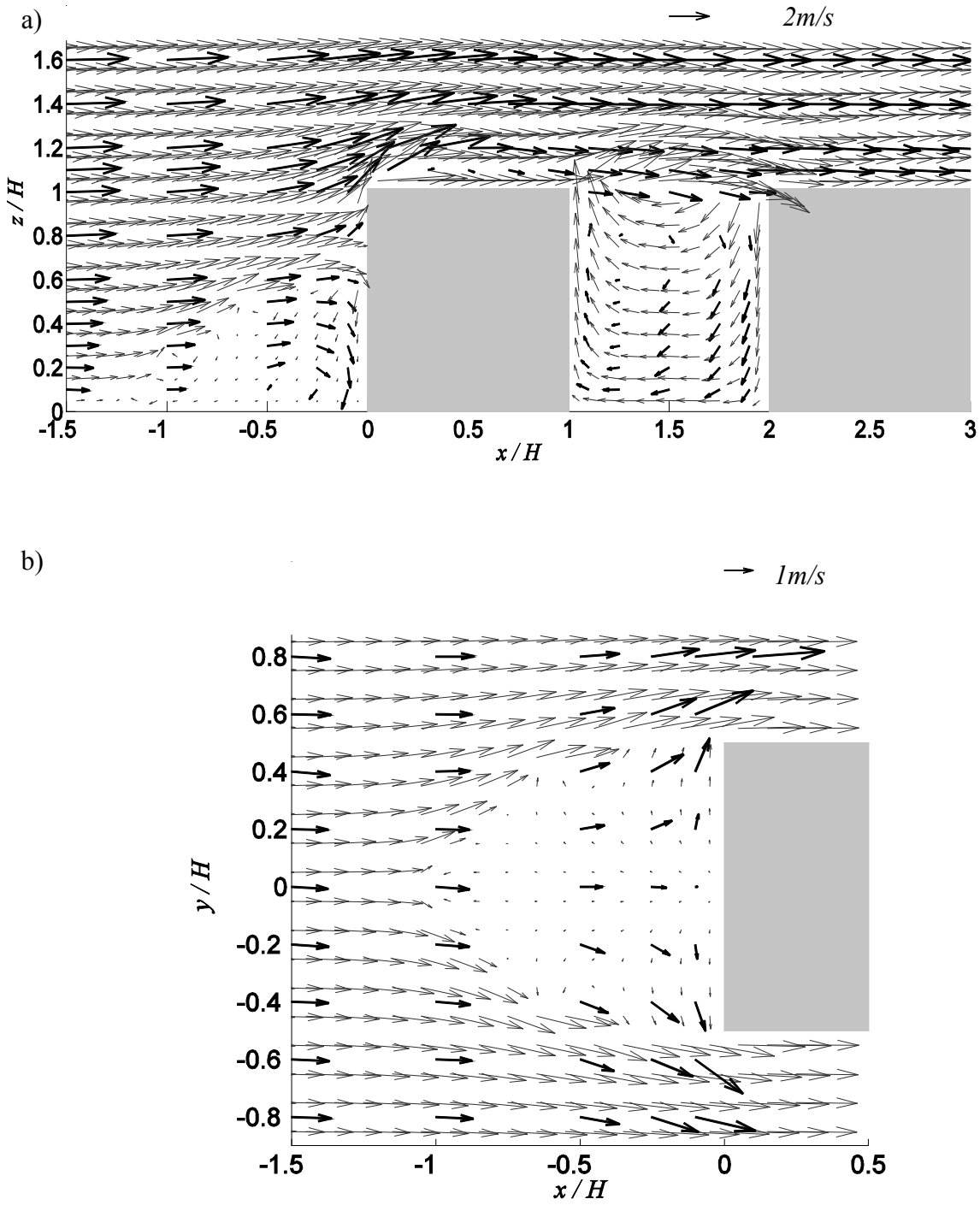


Figure 2.5: The SR model (gray) velocity vector comparison at the a) centerline b) upstream c) street canyon region with the experimental data (black).

c)

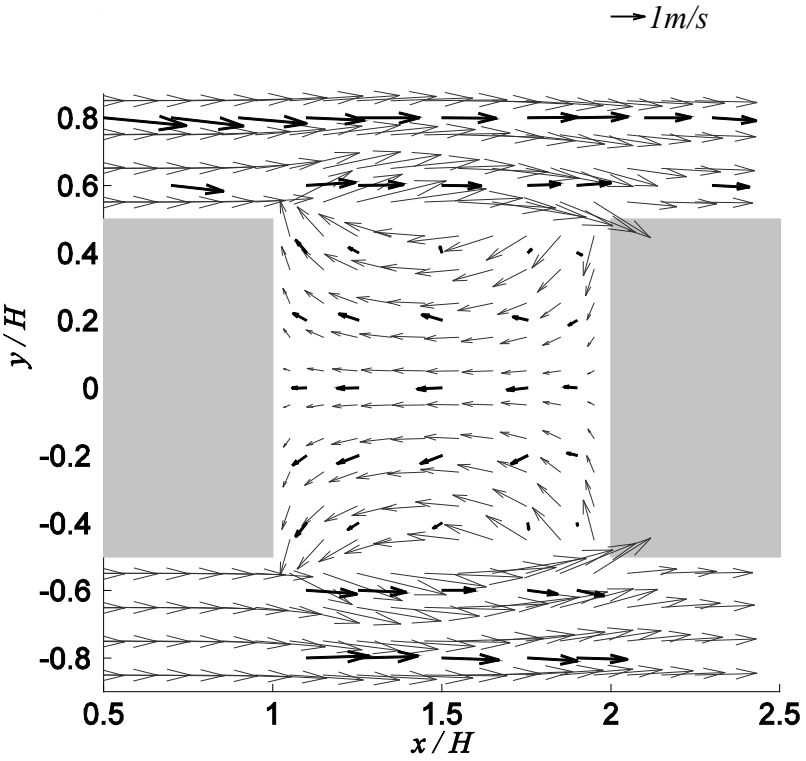


Figure 2.5: Continued.

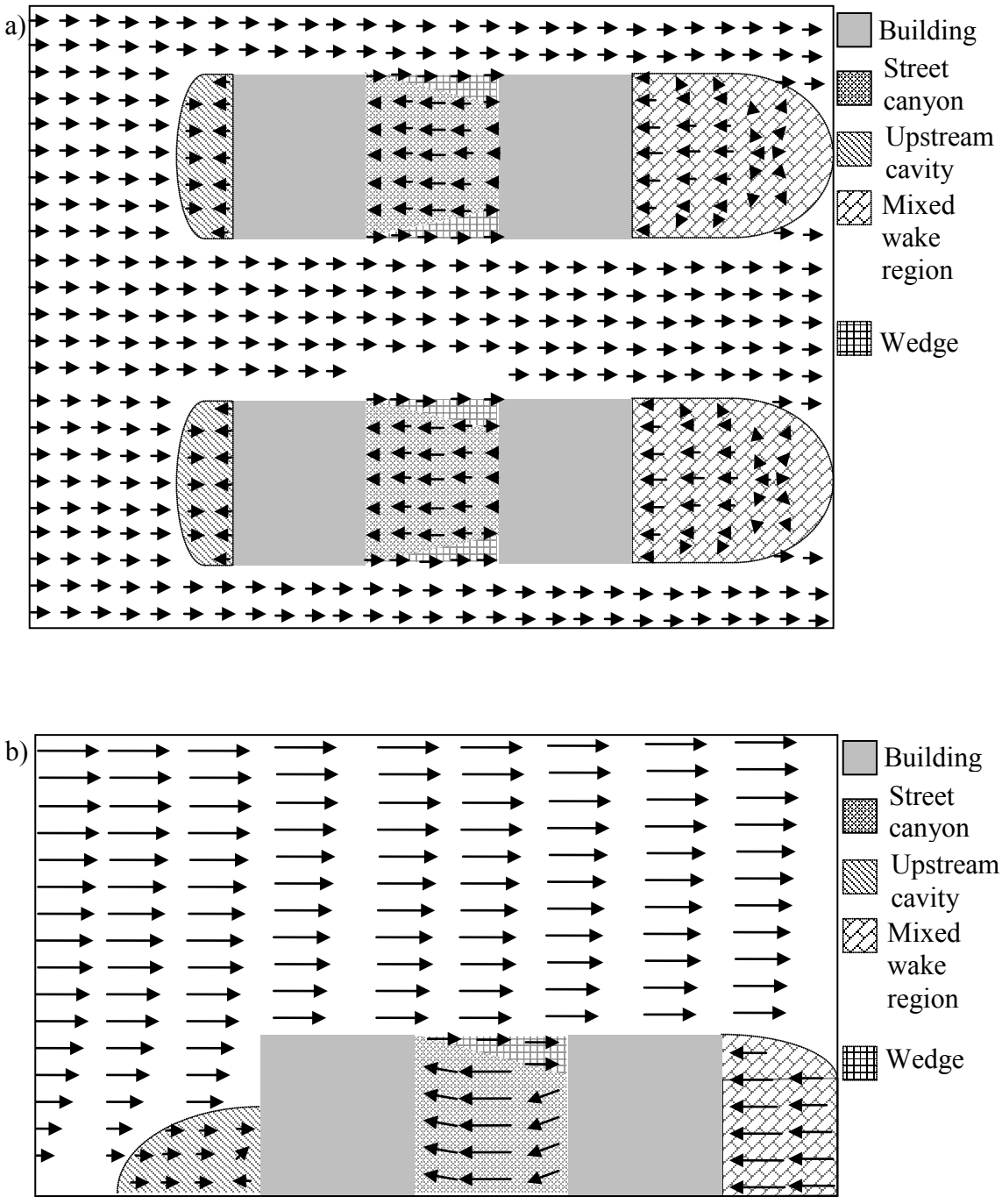


Figure 2.6: Schematic illustrating the flow regions and initial velocity fields associated with the MR street canyon parameterization in the a) horizontal b) vertical plane.

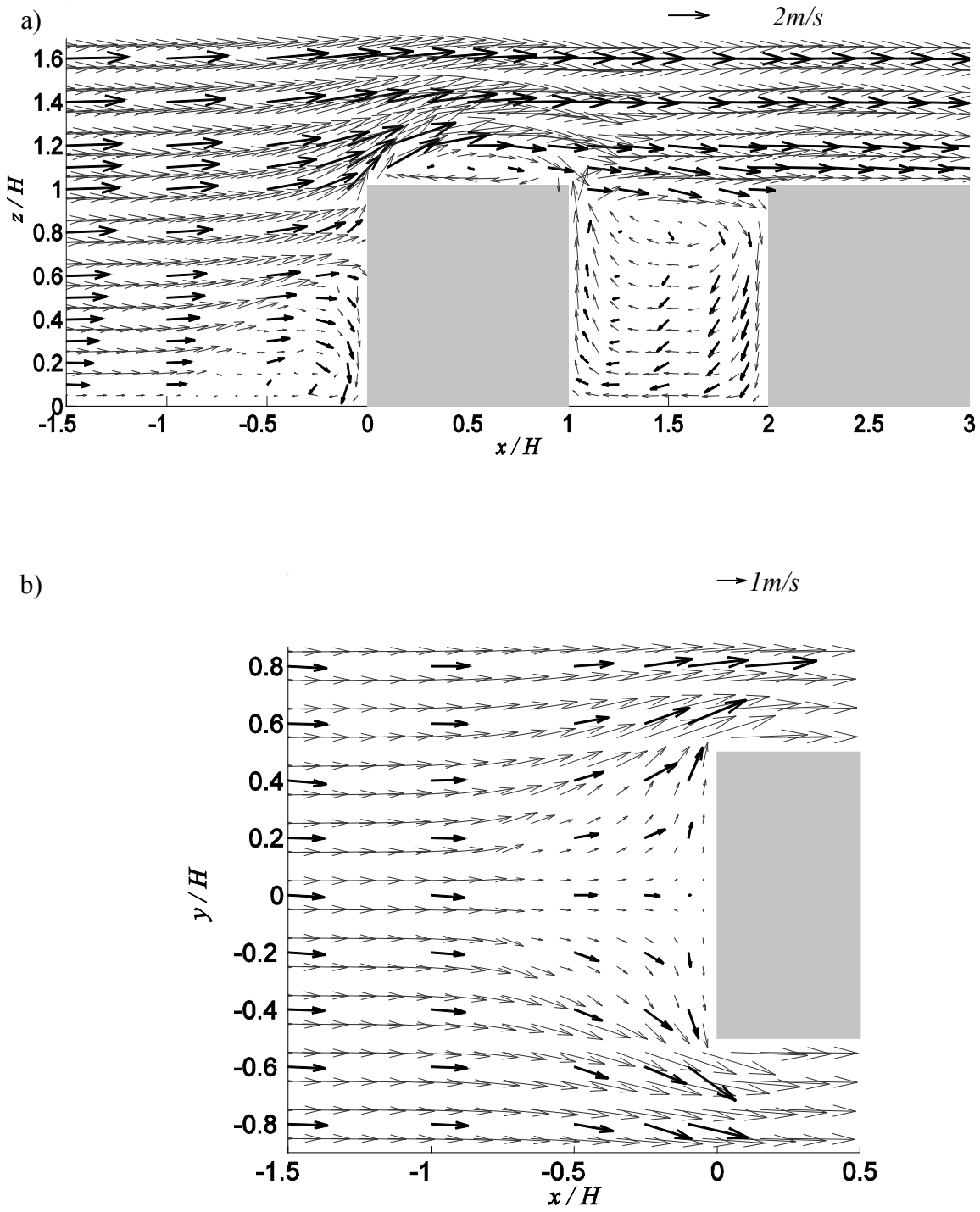


Figure 2.7: The MR model (gray) velocity vector comparison at the a) centerline b) upstream c) street canyon region with the experimental data (black).

c)

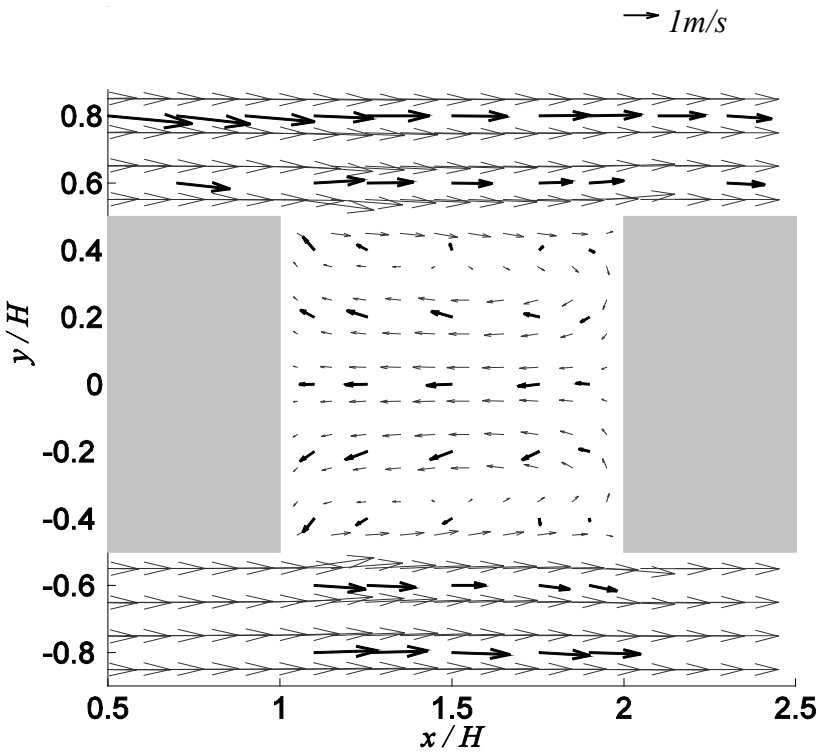


Figure 2.7: Continued.

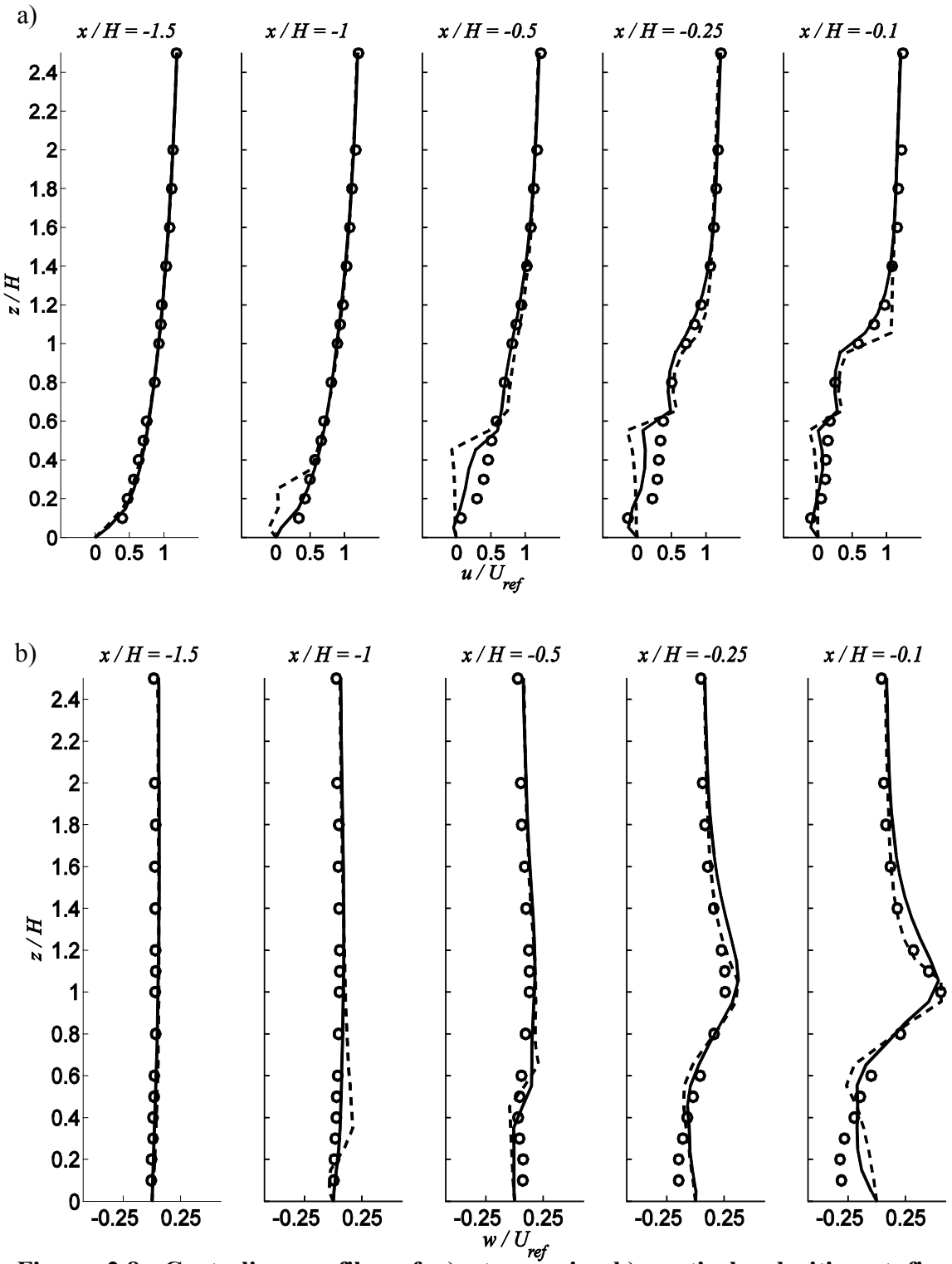


Figure 2.8: Centerline profiles of a) streamwise b) vertical velocities at five streamwise locations upstream of the building - experimental data (open circles), SR (dashed line) and MR (solid line).

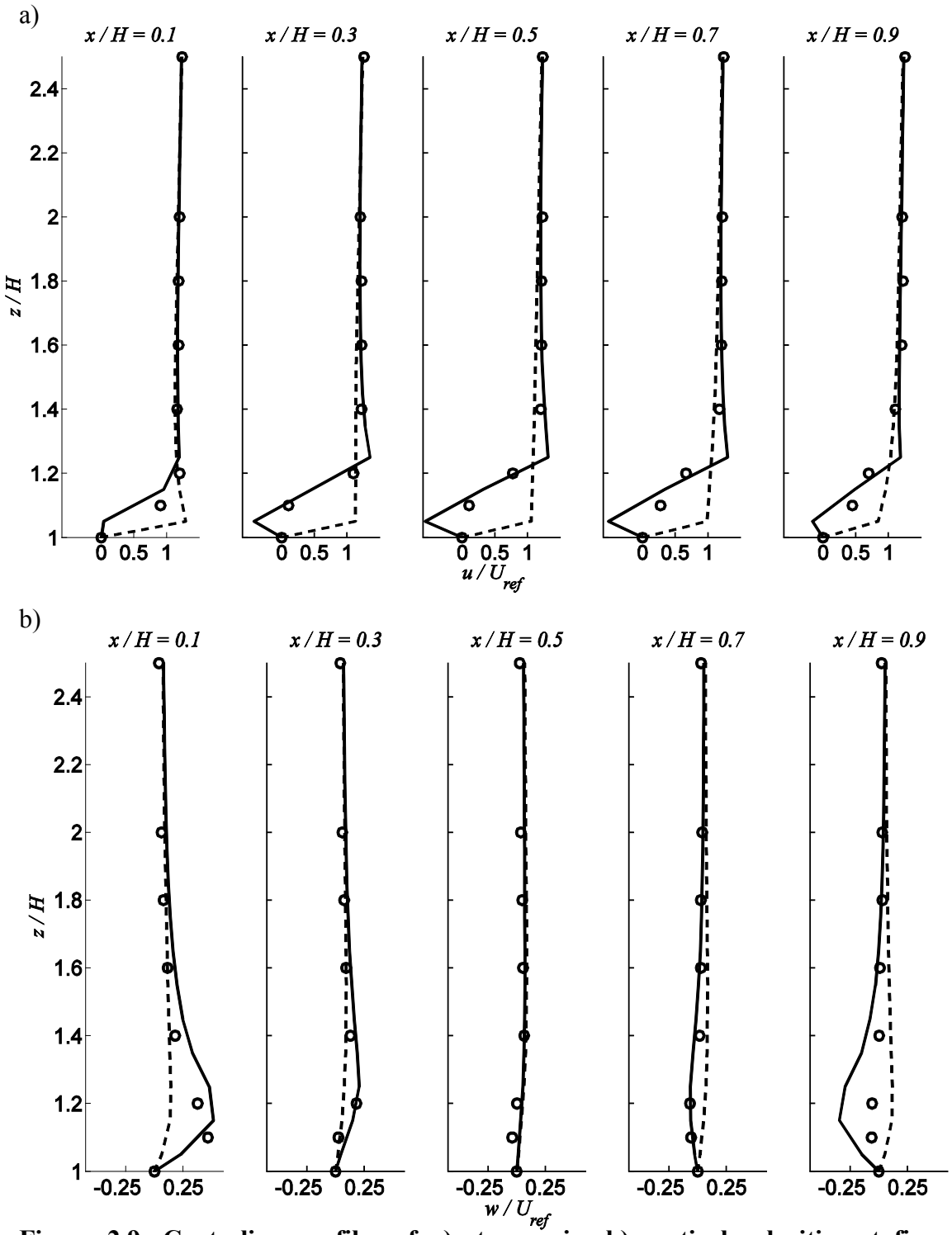


Figure 2.9: Centerline profiles of a) streamwise b) vertical velocities at five streamwise locations on the rooftop of the first building - experimental data (open circles), SR (dashed line) and MR (solid line).

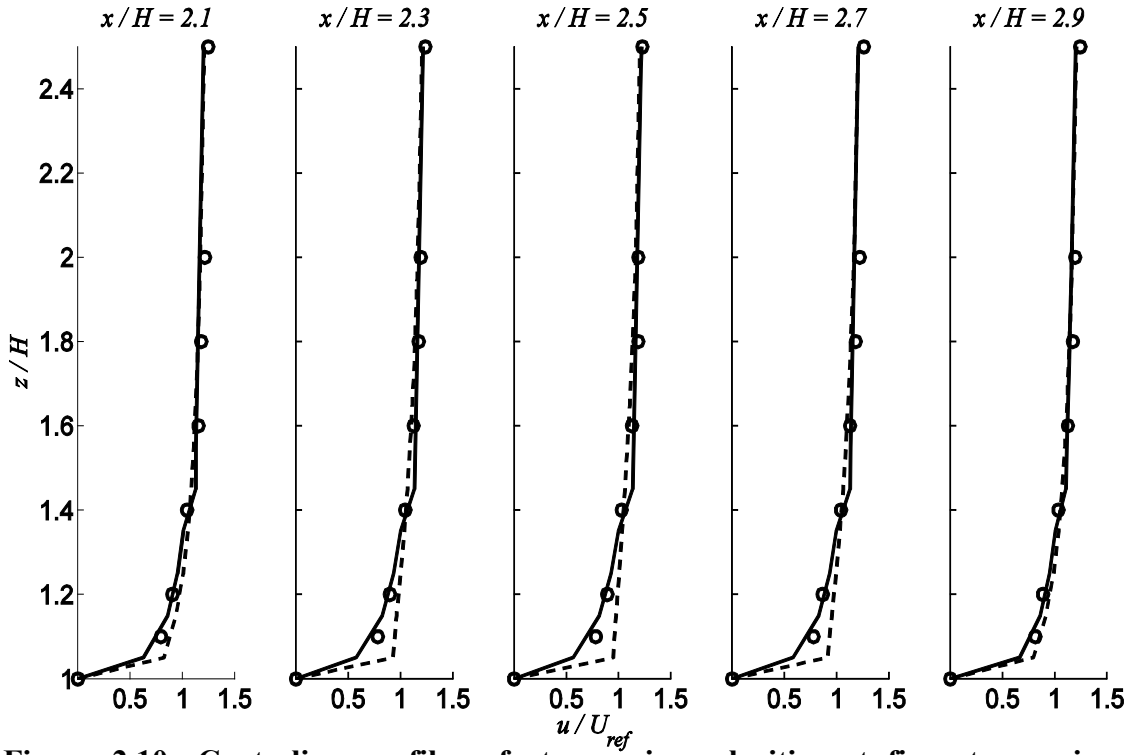


Figure 2.10: Centerline profiles of streamwise velocities at five streamwise locations on the rooftop of the second building - experimental data (open circles), SR (dashed line) and MR (solid line).

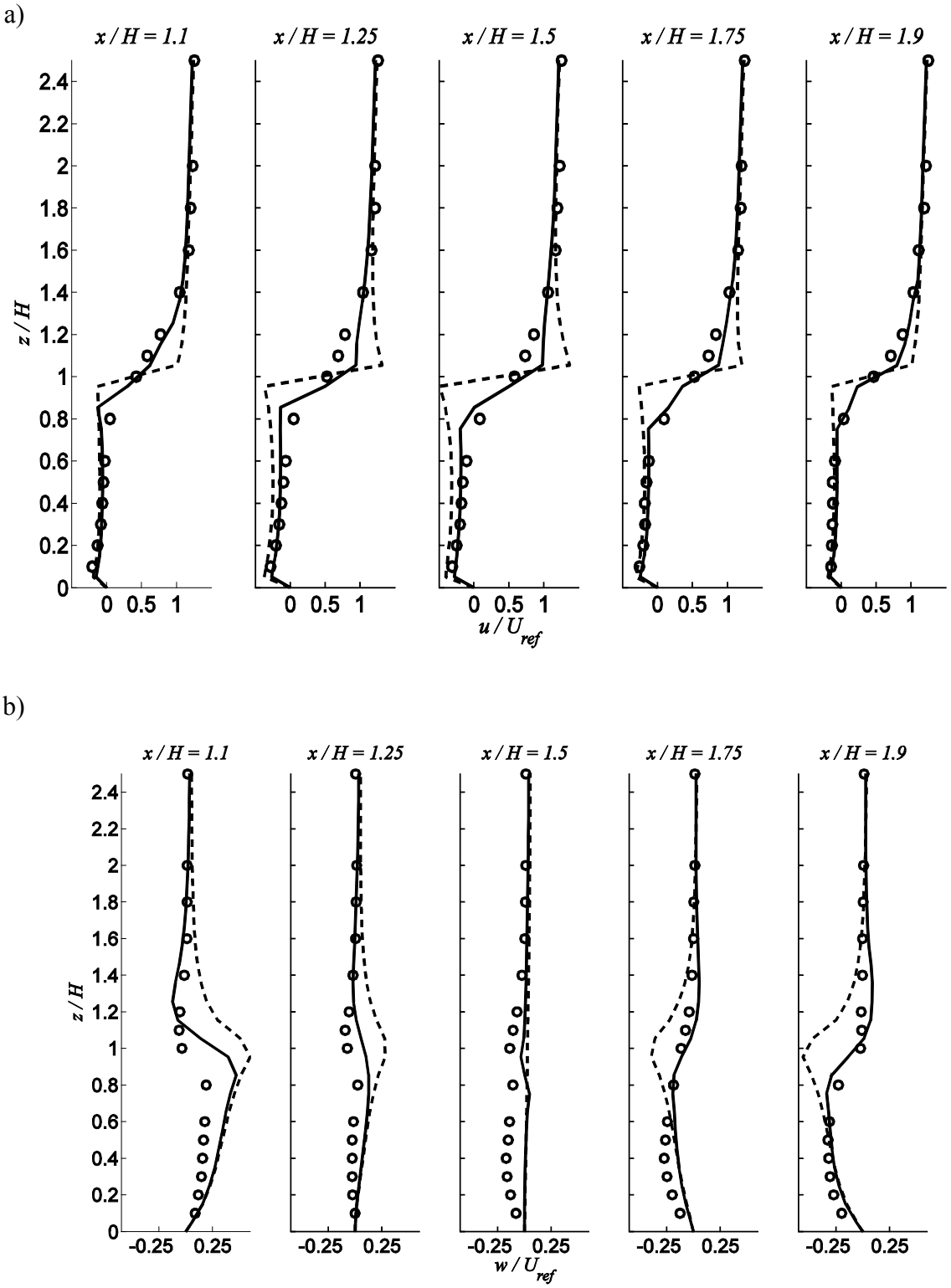


Figure 2.11: Centerline profiles of a) streamwise b) vertical velocities at five streamwise locations in the first street canyon - experimental data (open circles), SR (dashed line) and MR (solid line).

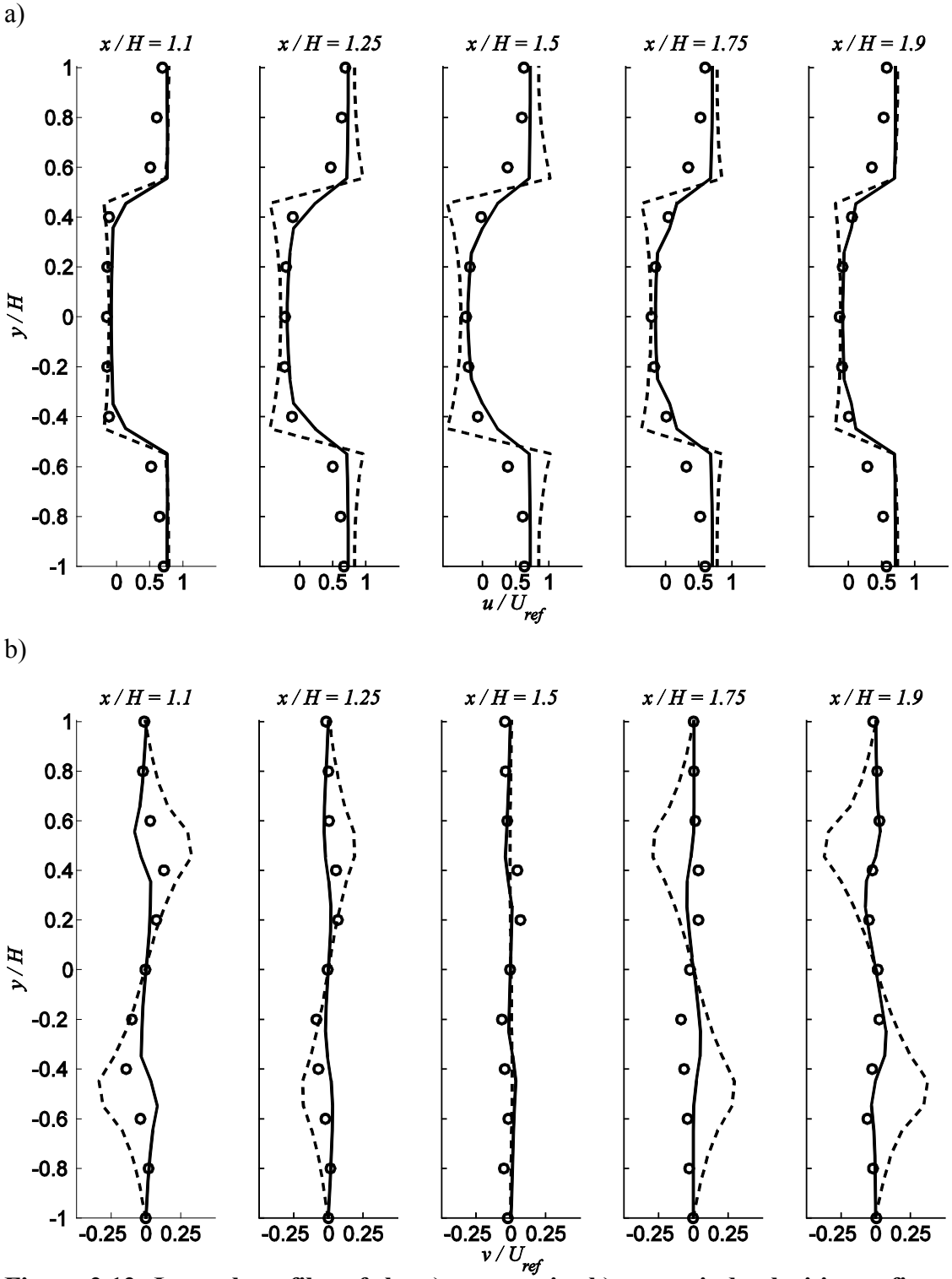
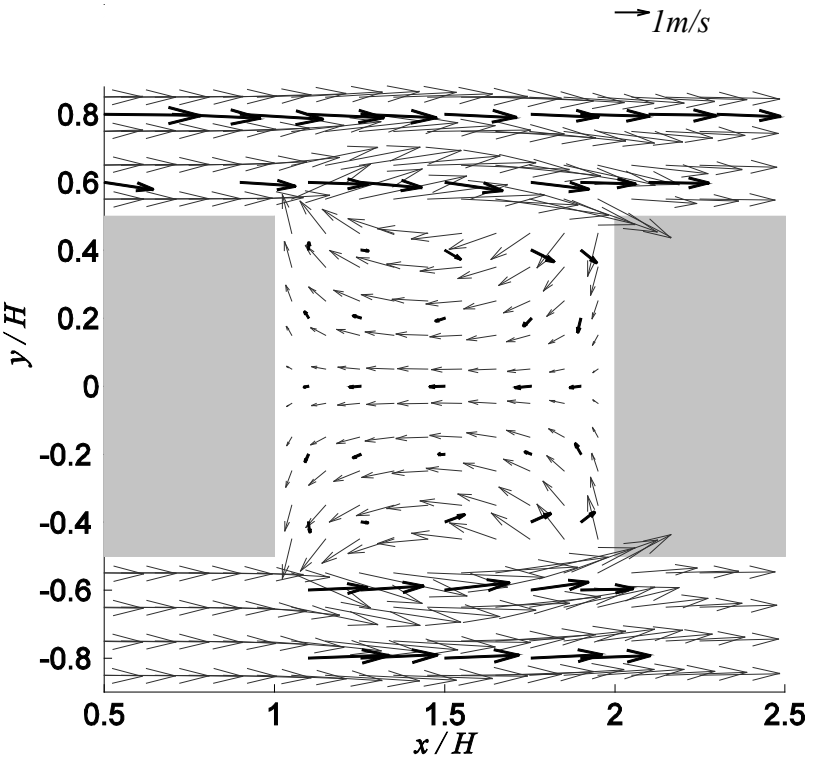


Figure 2.12: Lateral profiles of the a) streamwise b) crosswind velocities at five streamwise locations at z/H=0.2 in the first street canyon experimental data (open circles), SR (dashed line) and MR (solid line).

a)



b)

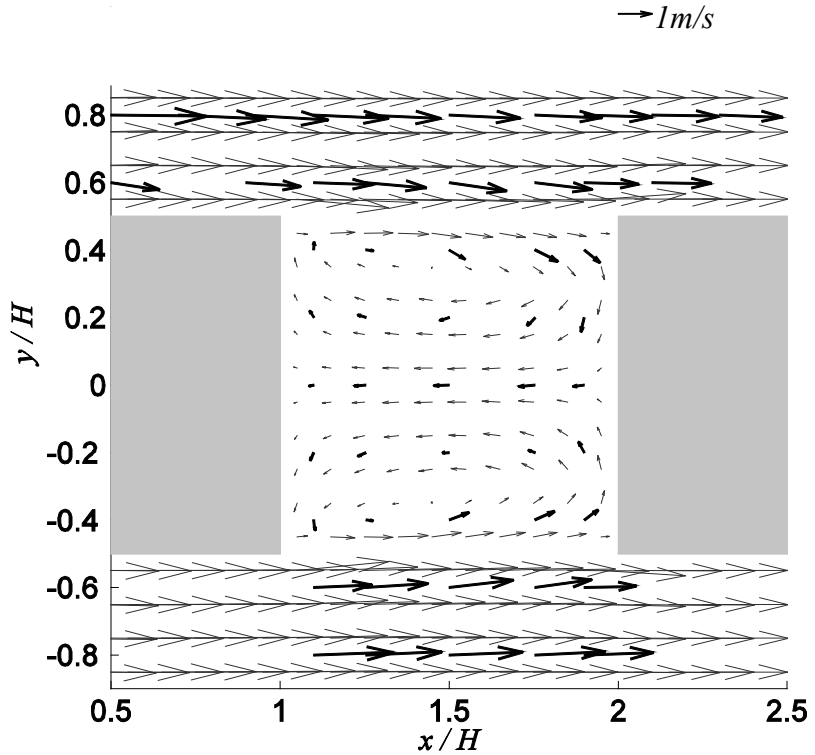
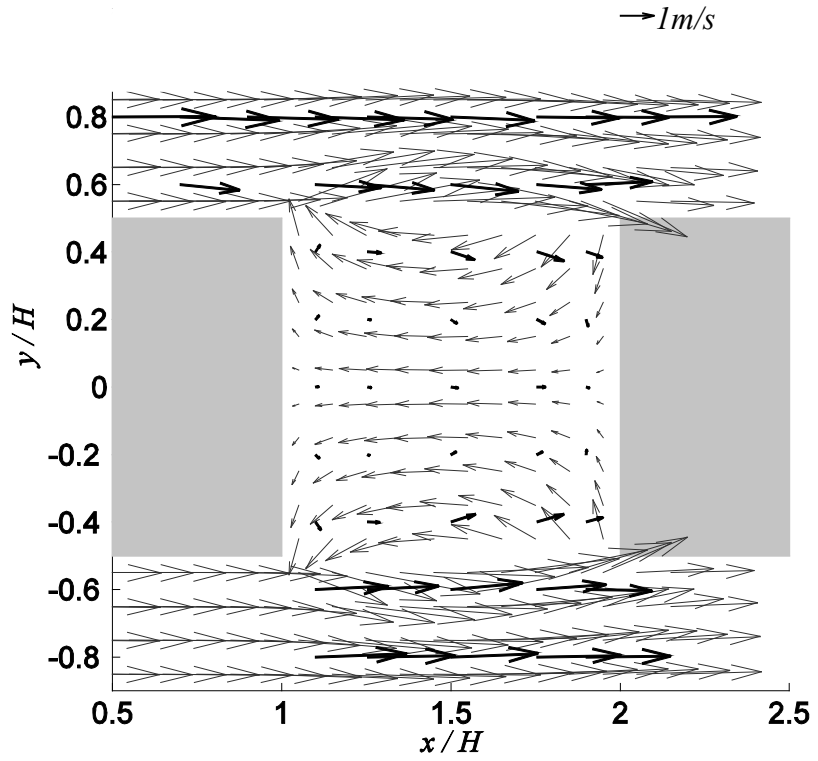


Figure 2.13: Plan view velocity vector comparison of the a) SR (gray) b) MR (gray) with the experimental data (black) at z/H=0.5 in the first street canyon.

a)



b)

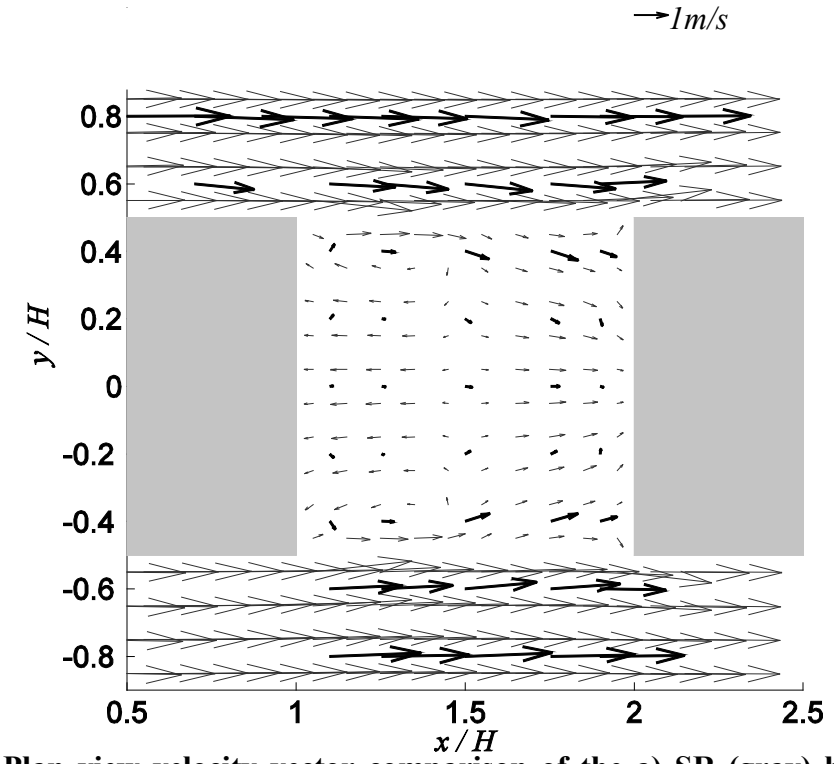


Figure 2.14: Plan view velocity vector comparison of the a) SR (gray) b) MR (gray) with the experimental data (black) at z/H=0.8 in the first street canyon.

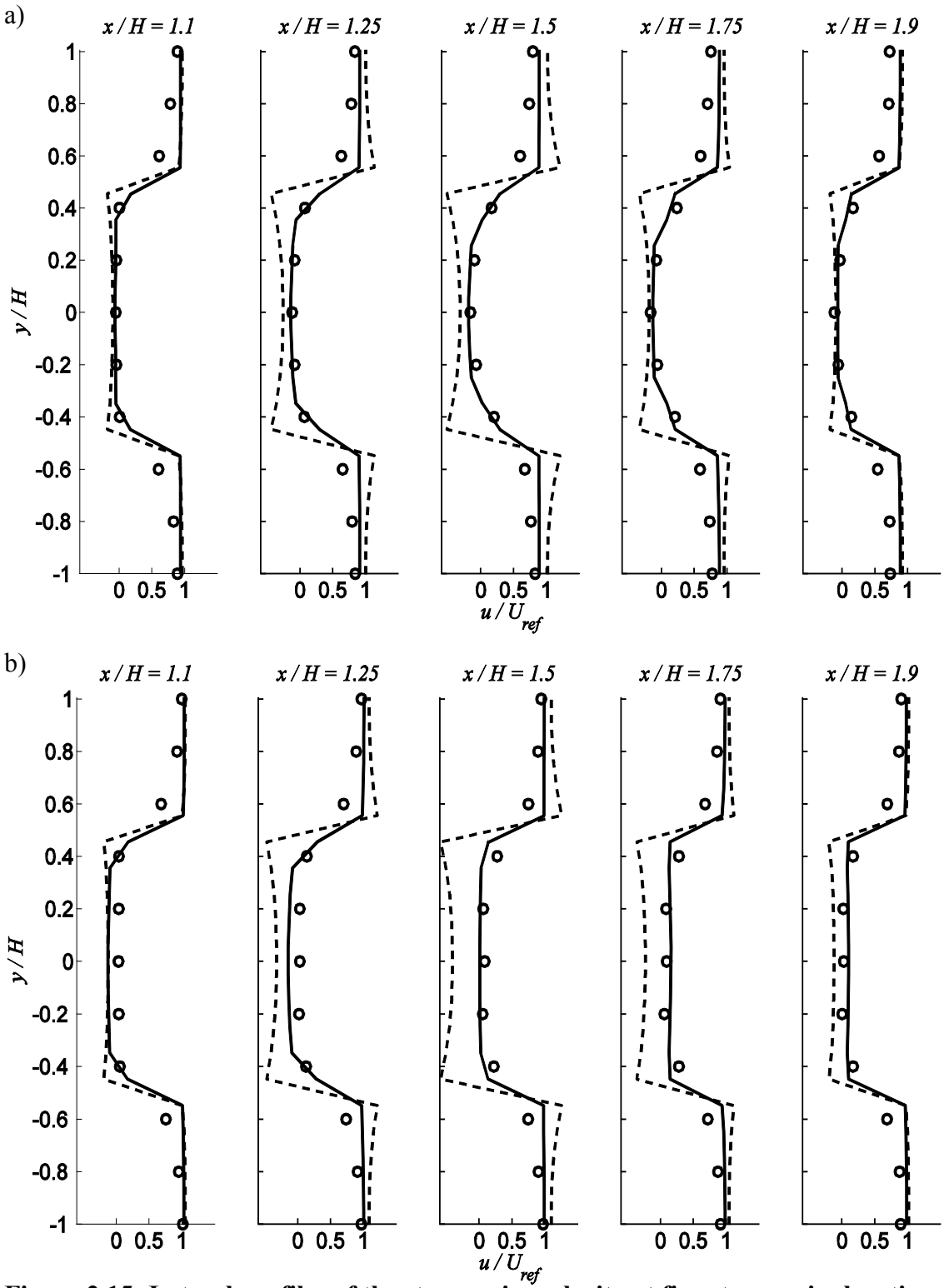


Figure 2.15: Lateral profiles of the streamwise velocity at five streamwise locations at a) z/H=0.5 b) z/H=0.8 in the first street canyon - experimental data (open circles), SR (dashed line) and MR (solid line).

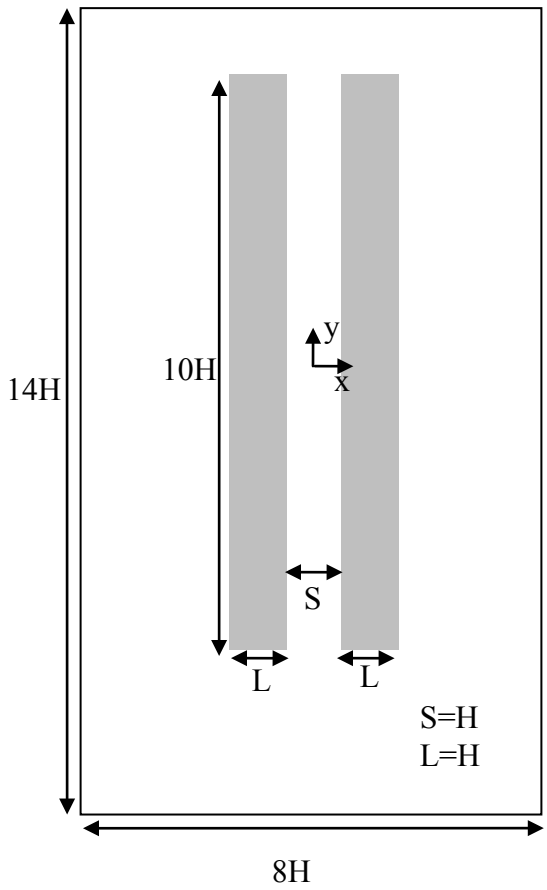


Figure 2.16: Schematic of the wide building street canyon used in the QUIC-URB simulations.

a)

b)

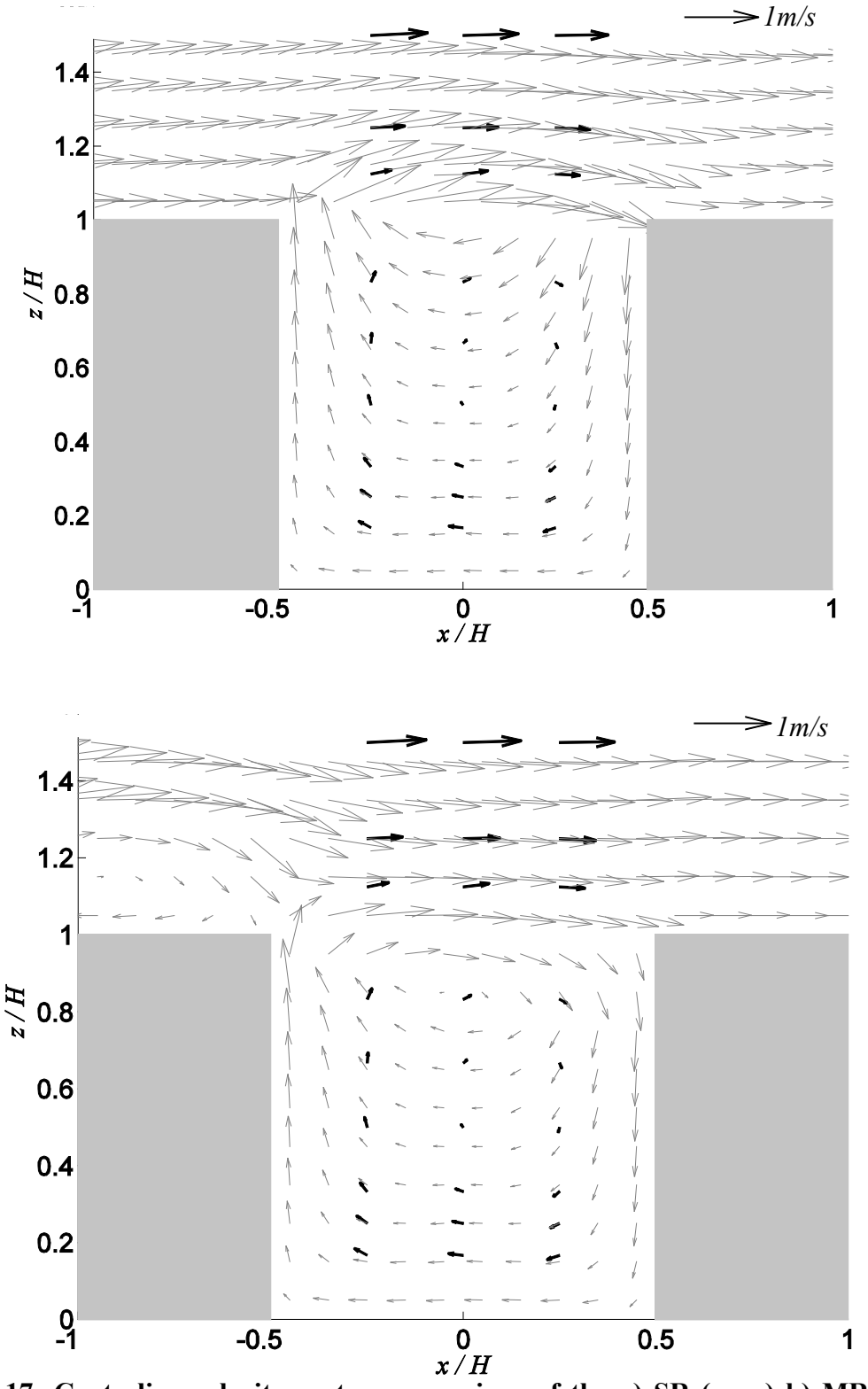


Figure 2.17: Centerline velocity vector comparison of the a) SR (gray) b) MR (gray) model with the experimental data (black).

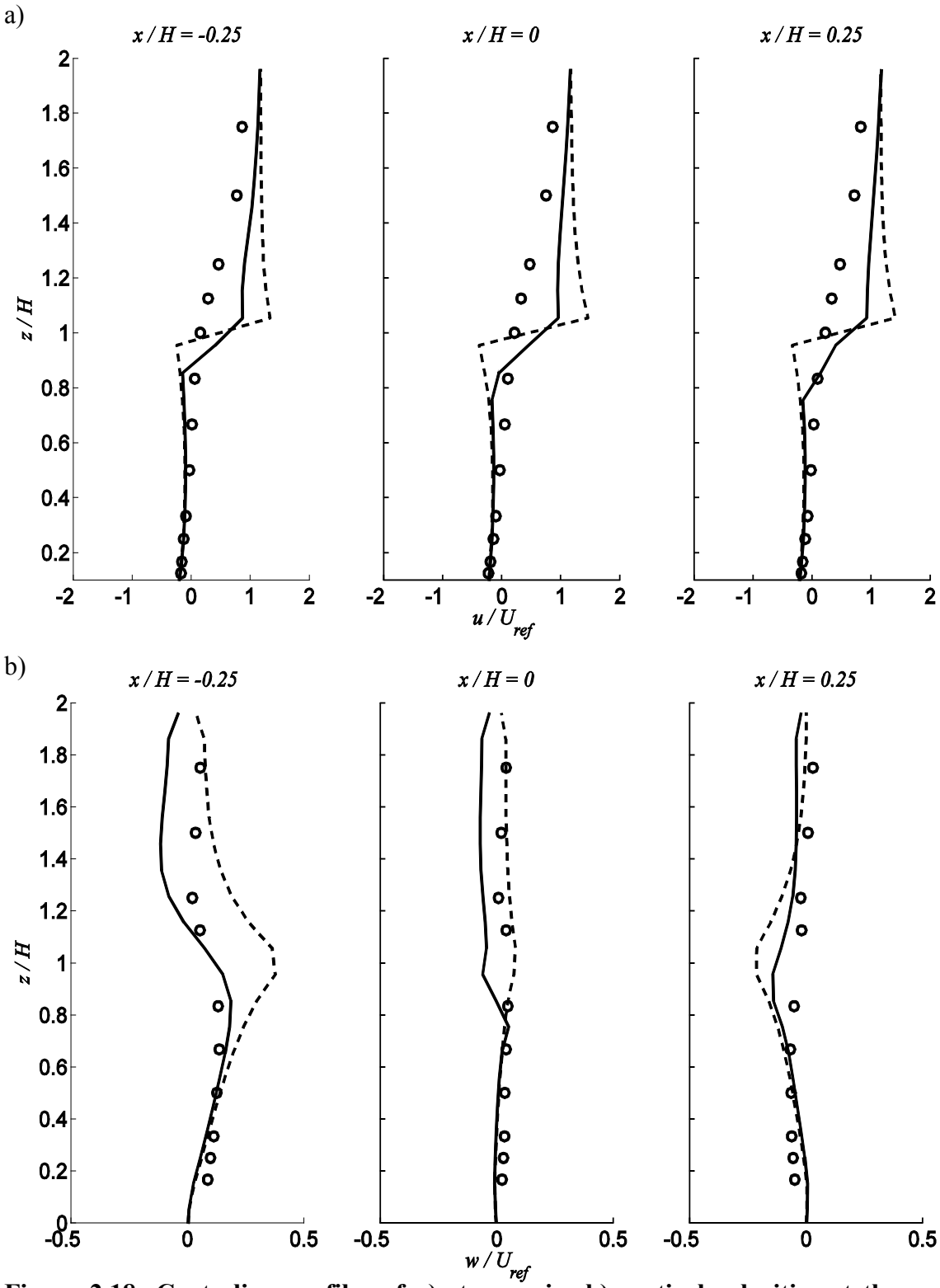


Figure 2.18: Centerline profiles of a) streamwise b) vertical velocities at three streamwise locations in the wide street canyon - experimental data (open circles), SR (dashed line) and MR (solid line).

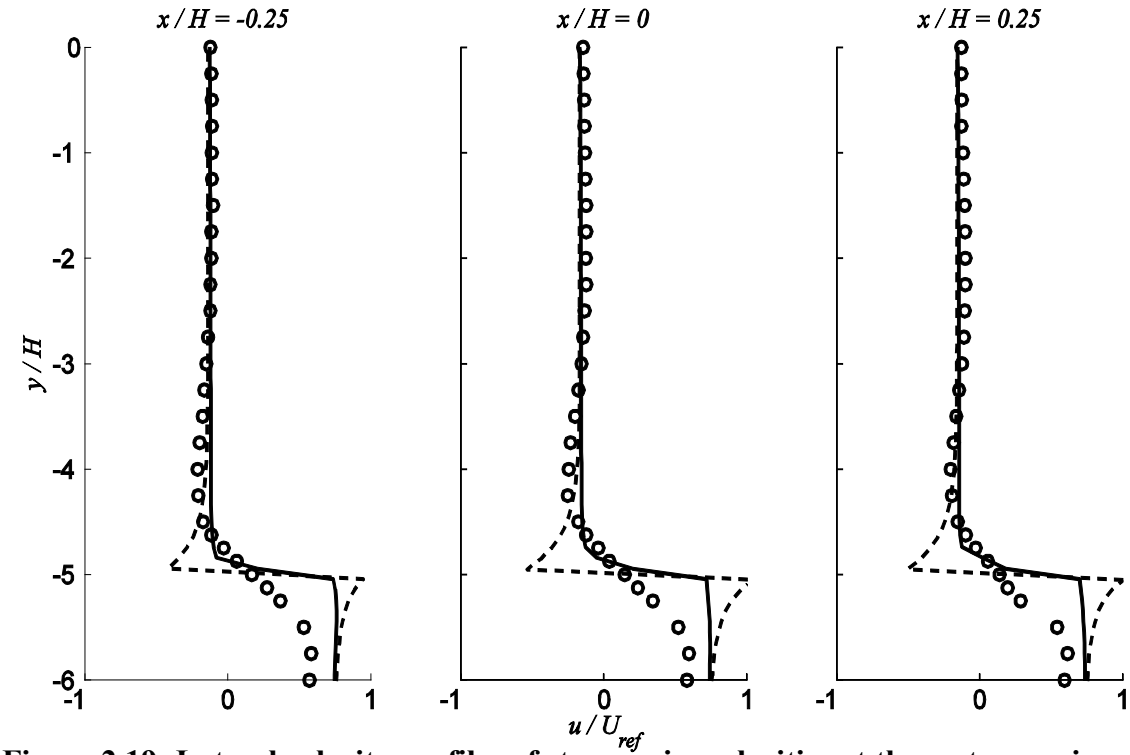


Figure 2.19: Lateral velocity profiles of streamwise velocities at three streamwise locations in the wide street canyon - experimental data (open circles), SR (dashed line) and MR (solid line).

Table 2.1: Cumulative average RMS error for the two models compared to the 7 x 11 experimental data.

%Error (RMS)				
	Upstream	Rooftop	Street Canyon	**Total**
Standard Röckle	10.4	13.7	20.2	17.2
Modified Röckle	6.1	7.2	9.5	8.4

Table 2.2: Cumulative average RMS error for the two models compared to the wide building experimental data.

%Error (RMS)	
	Street Canyon
Standard Röckle	23.9
Modified Röckle	16.9

3. SPEEDING UP URBAN FAST RESPONSE LAGRANGIAN DISPERSION SIMULATIONS USING VIDEO GAME TECHNOLOGY

3.1 Dispersion modeling for virtual urban environments on Graphics Processing Units

This chapter explains the implementation of a Graphics Processing Unit (GPU)-based fast response Lagrangian dispersion model which can also be used for virtual reality applications. For simulating real-world dispersion scenarios in virtual environments, it is imperative for the modeling system to run in near real-time. Generally, performance of a Lagrangian dispersion model is dependent upon the number of particles released in the simulation, domain size and resolution. The Lagrangian dispersion model employed for virtual environment applications must be computationally efficient and capable of advecting large number of particles to ensure stationary statistics. To completely immerse the user into a virtual environment, the user must be provided with a visual outlet that depicts interactive images of a 3D urban landscape along with the evolving contaminant field. Hence, real-time dispersion calculations must take place in the background while the visual outlet displays changes in the visible particles within the domain. This is different from more typical dispersion model computations where

variables (such as particle position data, concentration estimates, etc.) are available to the user after a set number of iterations or even after the simulation has completed.

In order to meet the computational and visualization goals of this project, the GPU on commodity graphics hardware has been utilized. GPUs typically take care of the processing needed to render images onto the computer screen. The faster this is done, the better the graphics are that can be displayed. The video game industry has been driving the performance of graphics cards in order to display better and better graphics. GPUs are vector processors that contain highly parallel stream processors, used to display real-time 3D graphics. These stream processors have been designed to work well with computations that are SIMD (single-instruction, multiple-data) computations. They are normally used for the SIMD operation of rendering images by coloring the pixels of the screen in parallel to display the image. On the most current GPUs, there are up to 240 stream processors (and rapidly increasing) capable of calculating 64-bit floating-point operations.

Researchers have taken notice of the GPU and started taking advantage of their computational power in order to solve problems that are not necessarily related to computer graphics. Because of this, General Purpose computation on Graphics Processing Units (GPGPU) has started, which is a way of using the GPU to solve non-graphics problems. Many scientific problems and simulation applications have already been solved and developed using the parallel stream processors of the GPU. Some examples of these are numerically solving the Navier−Stokes equations (Scheidegger et al. 2005), solving multigrid problems (Goodnight et al. 2005), solving dense linear systems (Galoppo et al. 2005) and cloud dynamics (Harris et al. 2003). A particle system

has been designed to run entirely on the GPU, simulating a million particles at interactive rates (Kipfer et al. 2004). This particle system is similar to our GPU particle system, with the difference being that our GPU particle system involves more complicated physics and has been developed to serve as a tool for engineers. Another particle dispersion model has been developed that visualizes dispersion in urban environments using a Lattice Boltzmann Model (Feng et al. 2004). It uses the GPU to accelerate the simulation; however, it does not run in real-time.

When solving a problem using a graphics card, the approach has to be designed to fit into the framework of the GPU architecture. Flexible functionality is provided to the GPU architecture through the use of vertex, geometry and fragment processing units called shaders. These shaders are low-level programs written using specialized languages similar to C, such as the OpenGL Shading Language or Cg, to overwrite stages of the graphics-processing pipeline. They were originally designed for graphics programmers to have more control over their applications to create better graphics and increase performance. Detailed explanations of shaders and the graphics processing pipeline can be found in Shreiner et al. (2007) and Rost (2006).

The main memory structure on the GPU is a texture, which are available in one, two and three dimensions. A texture is an array of vectors, where each vector, called a texel, is a color defined by a red, green, blue and alpha value, which is normally used as the colors for texturing geometry in graphics (Shirley and Marschner 2009). When thinking of textures as just an array of vectors, there is no reason why the values for the vectors may not be represented as information other than a color, such as the position of a particle. A texture can then be used to store data, where a fragment shader can then be

programmed to operate on the values of the texture in parallel. This is done by drawing geometry associated with the texture that stores the data to be operated on. The drawing can be done with the graphics state set so that there is a one to one mapping between a pixel and a texel. This makes it so that the code in the fragment shader is applied to each texel of the texture. Then instead of letting the output values of the fragment shader be displayed as pixel colors on a screen, they are stored as output values into the texels of another texture.

Solving problems on the GPU does not mean that data and operations occur solely on the GPU. In graphics programming, the data representation of graphics is created on the CPU and sent to the GPU, which then stores and processes the data to display on the screen. GPGPU also works in this same way. However, it is a good idea to limit the amount of data transfers between the CPU and GPU. The reason for this is *locality* of data. Processors work faster if data are stored locally, avoiding extra time taken to fetch data. Data are transferred between the CPU and GPU across the GPU "bus". The bus will act as a bottleneck if large amounts of data are being sent across it, because of insufficient bus width. This means that large amounts of data are unable to go across at the same time.

In this chapter, an implementation of a Lagrangian dispersion model onto the GPU is discussed. The goals of this work are: (i) to implement an urbanized Lagrangian dispersion model onto the GPU framework, (ii) to validate the model against the several simple analytical solutions, the original CPU implementation and wind-tunnel data for a single building test case, (iii) to understand the issues and challenges associated with this

type of GPU implementation and (iv) to understand and exploit the performance gains that can be realized through such an implementation.

3.2 Methodology

As discussed earlier (Chapter 1), the Lagrangian approach presents a logical way to describe the atmospheric dispersion phenomena for virtual environment applications. Therefore, as part of this work, a Lagrangian dispersion model, GPU Plume, based on the GPU architecture has been developed. "GPU Plume" is based on QUIC-Plume (Williams et al. 2004), the dispersion model of the Quick Urban and Industrial Complex (QUIC) dispersion modeling system. The QUIC dispersion modeling system was primarily developed to address accident releases and to respond to the threat of a terrorist releasing dangerous agents in a city. It is comprised of a diagnostic wind model - QUIC-URB (Singh et al. 2008), a Lagrangian dispersion model - QUIC-Plume (Williams et al. 2004) and a visualization tool, QUIC-GUI. The QUIC-URB wind model generates a mass consistent spatially explicit mean wind field in an urban area with complex building configurations. QUIC-URB (Singh et al. 2008) is based on the work done by Röckle (1990) and Kaplan and Dinar (1996). Empirical parameterizations are used to specify the wind field on a gridded domain and mass consistency is enforced on the flow field around the buildings (Singh et al. 2008).

GPU Plume requires a mean wind field and fluctuating components of the velocity to accomplish the particle advection process. The mean wind field is obtained from the diagnostic wind model, QUIC-URB. The fluctuating component of the velocity is obtained by solving the three-dimensional (3D) Langevin equations (Rodean 1996;

Wilson and Sawford 1996). The Langevin equations are a function of turbulence intensity at the location of the particle, drag experienced by the particle and random forcing due to the small-scale turbulent eddies on the particle equations (Rodean 1996; Wilson and Sawford 1996). To solve the Langevin equations, turbulent stresses associated with the flow field are required. The turbulence quantities are estimated from the mean wind field by employing a velocity gradient-driven mixing length turbulence model (Williams et al. 2004).

Before describing the details of the GPU implementation, it is useful to conceptualize the basic methodology. Figure 3.1 illustrates the difference in particle advection between the GPU and CPU implementations of QUIC-Plume. For the GPU implementation, the particle positions are represented in 3D space using 3D coordinates, which are stored in a 2D texture. The GPU is programmed to perform the advection of multiple particles at once. The number of particles that it can process at the same time increases with the amount of stream processors available in the hardware. On the most current hardware, there are up to 240 stream processors available. (However, this does not ensure the GPU will operate on 240 particles at once.) For the CPU implementation, the particles are stored in a 2D array. A CPU can only calculate the advection of one particle at a time (assuming it is a single core processor).

Figure 3.2 is an example of how a 2D texture is represented on the GPU. Each colored square is an individual texel, which represents an attribute for one particle, storing up to four values. This figure shows that there are 16 particles being represented in the simulation using 4 x 4 2D textures. The size and dimension of these textures will increase when the number of particles being simulated is more (i.e., one million particles

could be represented using a 1000 x 1000 2D texture). The order of the numbering shown here is important, because it shows how the texture is created from a set of data (which is stored in a 1D array on the CPU). To create one of these textures, a 1D array is used that stores all the values for the texture in a "line". Then the dimensions specified determine the layout for the texture. In Figure 3.2, the 1D array used to create a texture is of size 64, (16 texels x 4 values per texel) with the values, for each of the texels, placed in the array in the numbering shown. To clarify, the first four values for texel 0 are placed one after another in the array, the four values for texel 1 are placed one after another following the four values of texel 0 and this process continues for each texel. Then by specifying a width and height value of four, the texture is created with the layout for the texels shown in Figure 3.2. This layout is important in order to access the values of a particular texel using what are called texture coordinates. The texture coordinates give the location for a specific texel in a texture similar to the way values of an array are accessed using indices. In Figure 3.2, the texels of these 2D textures can be accessed using two texture coordinates: one being the row number and the other the column number.

The movement of particles in a Lagrangian dispersion model is largely based on domain data variables that are stored on a 3D grid. The wind field is a domain data variable in which each grid cell stores a velocity vector. Domain data, such as the wind field, could be easily loaded into a 3D texture making it trivial to determine which cell of the domain the particle is in. However, using a 2D texture to store the domain data provides better performance since GPUs tend to be optimized for working with 2D textures. A simple mapping can determine the cell location in the 2D texture using the 3D coordinates of a particle. Figure 3.3 illustrates how the three-dimensional domain data

can be stored into a 2D texture. Each colored block represents one layer of the domain. The domain is split up into layers at each height level of the domain grid. Each layer is made up of *nx* by *ny* cells. The variables *nx*, *ny* and *nz* are the domain dimensions in the *x*, *y* and *z* directions, respectively. The width and height of the 2D texture are even integers made to be equal and are determined by trying to minimize the dimension, while being able to fit all the layers. This is done to minimize the amount of space used for the texture. Each layer of the domain is taken and placed side by side in the 2D texture. The order starts at the bottom of the domain and places each layer from left to right. When room is run out of in one row of the 2D texture, the next layer is placed on top of that row on the left. Particles must then be mapped to a texel in the 2D texture. Details of the process are described in Norgren (2008).

3.3 Implementation of the GPU Plume dispersion model

3.3.1 Data management for GPU Plume

Table 3.1 summarizes the data storage for GPU Plume. As described above, the 3D wind field generated by the QUIC-URB wind model is converted into a 2D field that is compatible with the GPU architecture. This 2D field is stored in a texture (*wind* texture). Each node of this texture stores a *U, V* and *W* component of the mean velocity as three of the four values to be stored on each node of texture. The turbulence field estimated from the mean wind field is also stored in three *turbulence* textures.

The mean wind field and turbulence stresses describe the conditions prevailing in the domain that drive particle dispersion. To solve for the particle motion, the initial conditions are also specified. The initial positions of the particles to be released in the

simulation are distributed randomly within the source. For example, for a spherical source, the initial positions of the particles are scattered randomly within the radius of the sphere. The initial positions of all the particles are stored in one *particle position* texture. Because the GPU does not allow for new positions (after advection) to be stored back into the same textures as the previous positions, a second *particle position* texture is required. Each node of the *particle position* texture stores the *x, y* and *z* location of each particle as three of its four values and the state of the particle (active or inactive) as its fourth value.

GPU Plume also requires initialized fluctuating components of velocity associated with each particle. That is, the fluctuating component of velocity (*u', v'* and *w'*) should be initialized for each particle before the very first advection step. This is accomplished by employing the following equations (Williams et al. 2004):

$$u' = \sigma_u X_{ran}, \tag{3.1}$$

$$v' = \sigma_v X_{ran}, \tag{3.2}$$

$$w' = \sigma_w X_{ran}, \tag{3.3}$$

where, σ_u, σ_v and σ_w are the standard deviations of wind velocity in the *x, y* and *z* direction, respectively. The standard deviation of wind in the respective directions is obtained from the turbulence model. X_{ran} is a normally distributed random number with zero mean and a standard deviation of unity. The *u', v'* and *w'* are also stored in the *prime* textures. Similar to the particle positions, two textures are needed for the velocity fluctuations: one for the previous time step and one for the current time step. To estimate

the random forcing on the particles, Langevin equations also require normally distributed random numbers for the advection process; therefore, the generated normal random numbers (with zero mean and standard deviation of one) are stored in the *random* texture.

3.3.2 Utilizing stream processors for advection on the GPU

3.3.2.1 Mathematics behind the advection process -

The Langevin equations

To advect particles with the mean wind along with their respective fluctuating components, the following equations are employed (Williams et al. 2004):

$$x = x_p + U\Delta t + \frac{u'_p + u'}{2}\Delta t, \tag{3.4}$$

$$y = y_p + V\Delta t + \frac{v'_p + v'}{2}\Delta t, \tag{3.5}$$

$$z = z_p + W\Delta t + \frac{w'_p + w'}{2}\Delta t. \tag{3.6}$$

where *x, y* and *z* represent the current position of the particle and the subscript '*p'* refers to the previous positions. *U, V* and *W* are the mean winds in *x, y* and *z* direction, respectively, and u', v' and w' are the fluctuating components of the instantaneous wind. The time step is represented by Δt.

The fluctuating components of the wind are obtained from the following equations (Williams et al. 2004):

$$u' = u'_p + du, \tag{3.7}$$

$$v' = v'_p + dv, \tag{3.8}$$

$$w' = w'_p + dw. \tag{3.9}$$

where the subscript '*p'* refers to the previous fluctuating component of the velocity. As described earlier, the fluctuating components of the velocities are stored in a 2D texture. After each advection step, *u'*, *v'* and *w'* are updated with their new values in the *prime* texture.

The small incremental changes in the fluctuating component of the velocity are denoted by *du, dv* and *dw*, respectively, and are obtained by solving the 3D Langevin equations. As pointed out earlier, the Langevin equations are a function of turbulence intensity at the location of the particle, drag experienced by the particle and random forcing due to the small-scale isotropic turbulent motion of the particle (Wilson and Sawford 1996). Mathematically, the Langevin equations are represented as follows (Rodean 1996):

$$du_i = -\left(\frac{C_o\varepsilon}{2}\right)\lambda_{ik}u'_k dt + \left(U_j\frac{\partial U_i}{\partial x_j} + \frac{\partial U_i}{\partial x_j}u'_j + \frac{1}{2}\frac{\partial \tau_{ij}}{\partial x_j} + \frac{1}{2}U_m\frac{\partial \tau_{il}}{\partial x_m}\lambda_{lj}u'_j + \frac{1}{2}\frac{\partial \tau_{il}}{\partial x_k}\lambda_{lj}u'_j u'_k\right)dt + \left(C_o\varepsilon dt\right)^{1/2} dW_i(t) \tag{3.10}$$

In the above equation, C_o is a Universal constant (~5.7, Pope (2000)), ε is the mean rate of kinetic energy dissipation, the tensor $\lambda_{ij} = Adj(\tau_{ij})/\det(\tau_{ij}) = \tau_{ij}^{-1}$ is the

inverse matrix of the symmetric Reynolds stress tensor τ_{ij} and $dW_i(t)$ is uncorrelated, normally distributed variable with means of zero and standard deviations of one. The total velocity is denoted by u_i, whereas U_i and u_i' represent the mean velocity and the fluctuating components of the total velocity, respectively.

3.3.2.2 Concentration estimation

For concentration estimates, the volume of interest in the domain is divided into cuboids known as sampling boxes. Concentrations are estimated by dividing the total mass (in grams) of particles present in the sampling box by volume of the sampling box. If the total mass released in the entire simulation is *m* grams and total number of particles released is N_{tot}, then mass of each particle is m/N_{tot} grams. The concentration at any time step is given by,

$$C = \frac{total\ mass\ of\ \ particles\ in\ a\ sampling\ box}{Volume\ of\ \ the\ box} = \frac{n}{\forall}\frac{m}{N_{tot}}, \tag{3.11}$$

where, *n* is the total number of particles present in the sampling box and $\forall$ is the volume of the box.

For an averaging time of T seconds, the average concentration in a sampling box (*i, j, k*) is given by,

$$C_{avg(i,j,k)} = \frac{1}{T}\int_0^T C(t)dt,$$

where, dt is the sampling time after which particles are summed up in a sampling box. Substituting Eq. (3.11) in the above equation yields,

$$C_{avg(i,j,k)} = \frac{1}{T}\int_0^T \frac{nm(t)}{\forall N_{tot}} dt \tag{3.12}$$

For discrete data, the integral is replaced by summation and Eq. (3.12) becomes,

$$C_{avg(i,j,k)} = \frac{1}{T\forall}\sum \frac{nm(t)}{N_{tot}} dt \,.$$

Since T and $\forall$ are constant, the above equation reduces to simply,

$$C_{avg(i,j,k)} = \sum \frac{nm(t)dt}{T\forall N_{tot}} \tag{3.13}$$

where concentration is given in the units of g/m^3 .

3.3.2.3 GPGPU advection implementation

The simulation was created using the OpenGL API (Shreiner et al. 2007) along with the C++ programming language, and OpenGL's Shading Language (Rost 2006) (used to program the GPU). For the purposes of GPU Plume, our shader programs perform the advection computation, operating on 2D input textures to produce updated particle

position textures. For example, a Lagrangian dispersion model updates the position of the particles at each time step; therefore, at each time step, the shader program (for advection) acts on the *particle position* texture to update the particle positions stored in the *particle position* texture.

The Langevin equations are solved in the shader program to accomplish the advection steps. All of the textures (*wind* texture*, prime* texture*, particle position* texture*, random* texture*, turbulence field* textures) along with other simulation parameters (source parameters, building parameters, dispersion parameters) are made available to the shader programs through OpenGL commands. To accomplish particle advection, the velocity and turbulence fields local to the particle position are obtained through a GPU texture lookup function. That is, the velocity field (*U, V* and *W*) closest to the particle position is obtained by indexing a particle's position. A similar procedure is used to obtain the turbulence field around the particle. The shader program explicitly solves the 3D Langevin equations for the small incremental changes (*du*, *dv* and *dw*) to the fluctuating component of the velocity (*u'*, *v'* and *w'*), respectively. By employing the 3D Langevin equations along with the Eqs. (3.1) through (3.9), the new position of each particle is obtained for each simulation time step.

3.3.2.4 Treatment of particle reflection by building walls

and ground surface

Reflection is required when the estimated next position of a particle is either inside a building or under the ground surface. A billiard ball type reflection approach has been utilized for walls as well as ground reflection of a particle. As reflection is

extremely common in graphics operations, GPUs inherently support reflection calculations with built-in functions such as *normalize*, *distance* and *reflect. normalize* returns a vector of unit length in the same direction as the input vector, *distance* automatically calculates the length of a vector between two specified points and *reflect* returns the reflection direction for a vector coming into a surface with a specified outward point normal. To implement the reflection encoding, a 2D texture (*celltype* texture) containing information about building and ground cells in the domain is loaded on to the GPU memory. Using the current positions of the particle, the *celltype* texture provides run-time information to determine if the particle requires reflection or not by examining if the particle is within a building (or below ground). If a particle requires reflection, the particle is reflected appropriately by employing plane and line intersection equations (Norgren 2008).

3.3.2.5 Random number generator

Random numbers with a normal distribution are required at each advection step for solving the Langevin equations. The Box-Muller algorithm has been used to generate the normal random numbers from the uniformly distributed pseudo-random numbers available in the standard C++ (or C) library function (Press et al. 2007). The normally distributed random numbers (with values between 0 and 1 with a standard deviation of unity) are then stored in the *random* texture that is the same size as the *particle position* texture and is available to the shader program for particle advection. At each advection step, the required random numbers are obtained by sampling the normally distributed *random* texture. The sampling or indexing of the texture is done by randomly selecting

two unique offset indices (using the standard C++ uniform random variable library function) that define a location in the *random* texture.

3.4 Model evaluation

The performance of the GPU Plume model has been evaluated against three test cases. The primary goal of these validations was to test GPU Plume against its CPU implementation for its efficiency and the accuracy of the results.

3.4.1 Test case-I: Continuous release in uniform flow

In this test case, the performance of GPU Plume has been tested against QUIC-Plume (single precision CPU implementation) and an existing analytical solution for an elevated continuous point source release in a uniform flow. The normalized concentration profiles from the GPU Plume and QUIC Plume calculations have been compared against the classical Gaussian solution (Seinfeld and Pandis 1998) for a steady state, horizontally homogenous, neutral atmospheric stability, constant wind speed and constant eddy diffusivity (see Singh et al. (2004) for details). For this test case, the Langevin equation model was simplified for the horizontally homogenous and constant eddy diffusivity conditions (Willemsen et al. 2007),

$$du_i = -\left(\frac{C_o \varepsilon}{2}\right)\lambda_{ik} u_k' dt + \left(C_o \varepsilon dt\right)^{1/2} dW_i(t) \tag{3.14}$$

The plume parameters required to run the test case have been described in detail by Singh et al. (2004).

To obtain near statistically stationary concentration estimates, 100,000 particles were continuously release from a spherical source (0.2 m diameter) at a height H= 70 m. The rate of emission was 100 particles per second with a time step of 1 second (dt = 1 second) for a duration of 1000 seconds. The uniform wind speed was U=2 ms^{-1}, with a friction velocity of u_*=0.18 ms^{-1}. The concentration was averaged over 800 seconds with a starting time of 200 seconds after the beginning of the release. Number of sampling boxes in x, y and z directions were 20, 50 and 50, respectively, over a domain size of 100 m x 100 m x 100 m. The source was specified to be at x=20 m, y=50 m and z=H=70 m.

Figure 3.4 shows the lateral concentration profiles at two streamwise locations (x/H=0.964 and x/H=1.179). The concentration is normalized ($C^* = CUh^2/Q$) for the comparison purposes. The lateral profiles are in agreement with the analytical solution. Our objective was to compare the GPU Plume with the QUIC Plume (CPU implementation). The GPU Plume solution is nearly identical to the QUIC Plume solution with small statistical variations resulting from different random number generators. Similar trends have been observed in the vertical profiles (Figure 3.5).

In addition to the validation studies, simulations were run to analyze the performance benefit realized by using the GPU for the advection process. Figure 3.6 shows the comparison between the average time taken by GPU Plume and QUIC Plume for advecting N particles (averaged over 1000 advection steps). For all of the results reported, the simulations were run on a 2.4 Ghz Intel Core 2 Duo Processor with an NVIDIA GeForce 8800 GTS video card with 128 stream processors and 32-bit precision. GPU Plume shows a significant improvement in performance as compared to QUIC Plume. GPU Plume shows a speed up of two orders of magnitude when 100,000 particles

are released in the simulation (~250 times faster). Using the GPU, for the present test case, 3.3M particles can be simulated in real-time. However, with visuals turned on, a little less than 1M particles can be simulated in real-time.

3.4.2 Test case-2: Dispersion around an isolated cube

For this test case, an idealized continuous point source upstream of an isolated cubic building is tested (Figure 3.7). The concentration estimates were compared against the QUIC Plume dispersion model.

A spherical source of radius 0.1 m was placed 20 m upstream of an isolated cubical building (H=10 m). The source was located at x=5 m, y=25 m and z=3 m in a domain of 100 m x 50 m x 20 m in x, y and z directions, respectively. The wind field was generated by the wind model of the QUIC dispersion modeling system (QUIC-URB) with a logarithmic velocity profile upstream of the cube. The reference velocity for the velocity profile was specified to be 3 ms^{-1} at 10 m above the ground with a roughness length of 0.1 m. A total of 50,000 particles were released for 1000 seconds for this test case. The concentration averaging time was 1000 seconds. In order to assure that both CPU and GPU simulations had similar turbulence fields, GPU Plume utilized the turbulence parameters obtained from the QUIC Plume model to run this simulation.

The Langevin equations for 3D inhomogeneous turbulent flow have 67 terms for each component of the fluctuating velocity (assuming stationary conditions). The number of terms for each component of the fluctuating velocity can be drastically reduced by making boundary layer (BL) assumptions of horizontal homogeneity ($\partial/\partial x = \partial/\partial y = 0$)

and zero mean lateral and vertical flow ($V = W = 0$). Therefore, Eq. (3.10) with BL assumptions becomes,

$$du = -\left(\frac{C_o\varepsilon}{2}\right)\left[\lambda_{11}u' + \lambda_{13}w'\right]dt + \left(\frac{\partial U}{\partial z}w' + \frac{1}{2}\frac{\partial \tau_{13}}{\partial z} + \frac{w'}{2}\left[\begin{array}{l}\frac{\partial \tau_{11}}{\partial z}\left[\lambda_{11}u' + \lambda_{13}w'\right] \\ +\frac{\partial \tau_{13}}{\partial z}\left[\lambda_{13}u' + \lambda_{33}w'\right]\end{array}\right]\right)dt \quad (3.15)$$
$$+\left(C_o\varepsilon dt\right)^{1/2} dW_1(t)$$

$$dv = -\left(\frac{C_o\varepsilon}{2}\right)\left[\lambda_{22}v'\right]dt + \left(\frac{1}{2}\frac{\partial \tau_{22}}{\partial z}\lambda_{22}v'w'\right)dt + \left(C_o\varepsilon dt\right)^{1/2} dW_2(t) \quad (3.16)$$

$$dw = -\left(\frac{C_o\varepsilon}{2}\right)\left[\lambda_{13}u' + \lambda_{33}w'\right]dt + \left(\frac{1}{2}\frac{\partial \tau_{33}}{\partial z} + \frac{w'}{2}\left[\begin{array}{l}\frac{\partial \tau_{13}}{\partial z}\left[\lambda_{11}u' + \lambda_{13}w'\right] \\ +\frac{\partial \tau_{33}}{\partial z}\left[\lambda_{13}u' + \lambda_{33}w'\right]\end{array}\right]\right)dt \quad (3.17)$$
$$+\left(C_o\varepsilon dt\right)^{1/2} dW_3(t)$$

For the present test case, the Langevin equations in GPU Plume were simplified by making BL assumptions. The QUIC Plume dispersion model also utilizes the simplified equations with the BL assumption. However, to partially accommodate the horizontal homogeneity, the QUIC Plume dispersion model utilizes local streamline coordinate transformation in the direction of the total mean wind (Näslund et al. 1994; Williams et al. 2004). The local streamline coordinate transformation is not implemented in the GPU Plume model. Due to this difference in the implementation of the GPU Plume and QUIC Plume, the results from the GPU Plume and QUIC Plume do not match exactly; however, the concentration profiles follow each other quite well.

Figure 3.8a shows the lateral concentration profiles at three streamwise locations. At x=9.75 m and z=2.5 m (Figure 3.8a), the GPU Plume is in very good agreement with the QUIC Plume. Both GPU Plume and QUIC Plume show the peak concentration at y/H=2.5, inline with the source location.

Figure 3.8b shows the lateral concentration profile at x=57.25 m and z=2.5 m. The entrainment of the plume behind the building resulted in two peaks at y/H~1.5 and 3.5. A concentration peak is also observed at the centerline of the building in the lateral direction. The GPU Plume and QUIC Plume concentration profiles match well; however, GPU Plume shows less lateral dispersion. We believe that the lack of dispersion is due to the slightly different implementations of the GPU Plume and QUIC Plume. Specifically, QUIC Plume utilizes local streamline coordination transformation (Williams et al. 2004) which is not implemented in the current version of GPU Plume.

Figure 3.8c shows lateral concentration comparison at x=85.75 m and z=2.5 m. The concentration profiles are in agreement with each other. Figure 3.9a & b shows vertical concentration profiles at two streamwise locations downstream of the building (x=57.25 m and x=85.75 m with y=24.01 m). The concentration profiles (Figures 3.8c, 3.9a and 3.9b) show similar results except for some differences due to the differences in the implementations of the GPU Plume and QUIC Plume along with the differences in random number generators.

Figure 3.10 shows the performance analysis of the GPU Plume with respect to the QUIC Plume (single processor CPU implementation). Figure 3.10 shows a comparison between the average time taken by GPU Plume and QUIC Plume for advecting N particles (averaged over 1000 advection steps). GPU Plume again shows two orders of

magnitude speed up in the performance. GPU Plume is in fact 180 times faster than QUIC Plume for a release of 200,000 particles. The uniform flow test case (see Section 3.4.1) showed that GPU Plume is ~250 times faster than QUIC Plume. The added textures to store 3D turbulence data and the reflection algorithm integrated in GPU Plume for this test case resulted in the performance penalty of less than a factor of two. Here again, the simulations were run on a 2.4 Ghz Intel Core 2 Duo Processor with an NVIDIA GeForce 8800 GTS video card.

3.4.3 Test case 3: Dispersion in the lee-side of a Hi-Rise building

For this test case, GPU Plume was validated against a wind-tunnel experiment of dispersion in the lee side of a tall building (*Hi-Rise*). The experiment was conducted at a USEPA meteorological wind-tunnel by Ohba et al. (1993). A small spherical source was placed near the surface on the lee side of the building as shown in Figures 3.11 and 3.12. The concentration and wind measurements were made in the *x*-*z* plane on the downwind side of the building. The height of the building was 3 times the length and width of the building (L=W=0.2 m, H=0.6 m). The mean wind field was obtained from the QUIC-URB wind model. The inlet/initial profile was specified to be a power law profile with an exponent of 0.295 and a reference velocity of 3.5 ms^{-1} at the height of the building. For further details, see Williams et al. (2004) and Ohba et al. (1993). The turbulence parameters required to run the GPU Plume model were obtained from the QUIC Plume model.

Figure 3.13a shows vertical concentration profiles at the centerline of the building at x/H=0.042. Both GPU Plume and QUIC Plume overestimate the ground level concentrations; however, GPU Plume is slightly closer to the experimental data (black open circles). As pointed out earlier, the differences between GPU Plume and QUIC Plume are due to the different implementation of the model. At x/H=0.17 (Figure 3.13b), both models produced similar results and agree well with the data. Both models show a peak normalized concentration of 700; however, data are unavailable at that height. Figure 3.13c shows the normalized concentration profile at x/H=0.42. Both models underestimate the ground level concentration; however, GPU Plume performs slightly better than QUIC Plume as the distance from the ground increases. At x/H=0.58 (Figure 3.13d), QUIC Plume slightly underestimates the ground level concentration; however, GPU Plume overestimates the ground level concentration. GPU Plume follows the experimental data well as the distance from the ground increases; however, QUIC Plume underestimates the normalized concentration as distance from the ground increases.

3.5 Discussion and summary

Fast response atmospheric dispersion models are valuable tools that can aid first responders in making decisions regarding accidental or deliberate releases of chemical or biological agents in complex urban environments. When implemented on the GPU, the real-time performance of these models also makes them suitable for applications involving the display of dispersion phenomena in urban virtual environments. To our knowledge, this is the first attempt to integrate a fast response dispersion model into an urban virtual environment running in real-time. The results indicate that by using the

GPU, a substantial performance benefit can be obtained in the advection process of the dispersion model. The performance of GPU Plume is two orders of magnitude faster than its CPU implementation, while preserving computational accuracy. The concentration profiles obtained from GPU Plume are in good agreement with the CPU implementation with small variations attributed to the differences in the GPU and CPU implementations. The single building algorithm does show a small performance penalty over the uniform flow case and is associated with the added memory textures and reflection algorithm. While not shown here, tests of up to 35 buildings in a domain indicate the penalty for added buildings is less than that of the CPU implementation. For example, for a 500,000 particle simulation, the CPU simulation takes about 34% longer to run a 35 building simulation compared to a single building simulation, while the GPU only takes about 12% longer (Norgren 2008). This improved performance may be a result of the efficient built-in reflection algorithms on the GPU.

The immediate benefit of real-time visualization, obtained by rendering the dispersion data on the screen, is a novel approach for probing the evolving dispersion field, which further enables one to qualitatively understand the dispersion phenomena better. The visualization aspect of GPU Plume is a powerful tool that also enables the user to view the turbulence contours and the evolving concentration field in real-time. Figure 3.14 shows an example of such a visual where the user's head and hand positions and orientations are tracked to provide interaction with the particle simulation data. In Figure 3.14, the user is rotating a contour plane of a turbulence variable in real-time during the simulation. Additional details regarding the integration of the simulation into a virtual environment can be found in Willemsen (2008) and Norgren (2008). An

additional exploitation of the benefits of real-time visualization of GPU Plume is modification of simulation parameters and models during the simulation. This will allow users to receive immediate feedback regarding changes in material properties, building positions and physical submodels.

Some of the memory limitations associated with the GPU Plume simulations deserve more explanation. The number of particles available for simulation is limited by the size of 2D textures and the amount of memory on the graphics card. The maximum size of a 2D texture is 8192 x 8192 on an NVIDIA Geforce 8800, theoretically meaning that the amount of particles that can be simulated is about 64 million. However, in practice, the number of particles is also limited by the amount of memory available on the graphics card. For example, if the number of particles being simulated is 1 million, there will be five 1000 x 1000 2D textures (previous_positions, new_positions, previous_velocities, new_velocities and the random_values texture). Each of these textures store 32 - bit floats and are 15.25 MB ($1000*4*32/(8*1024^2)$). These five 71 textures together hold 76.3 MB. Then there will also be five textures holding domain data (new and old particle positions, new and old velocity data and random numbers). If the domain is 100 x 100 x 50, then each of these textures holds approximately 7.63 MB, combining to 38.1 MB. These ten textures are the textures required to run the core simulation, which requires 114.4 MB. However, there are additional textures required to color the particles, store building information, draw the isosurfaces, etc., along with additional memory used for displaying the visualization. If the amount of particles to simulate increases to 4 million using the same domain, 343.28 MB would be required. The present GPU simulations are stable with 4 million particles on an NVIDIA GeForce

8800 GTS with 640 MB of memory. 16 million particles would require 1221 MB, which is well over the size of the NVIDIA GeForce 8800 Ultra that has the largest amount of memory available at 768 MB. If the amount of memory available was not an issue, textures of size 8192 x 8192 could be used to simulate over 16 million, and additional textures could also be used to simulate even more particles by operating on each texture individually each time step.

Programming the GPU is currently a nontrivial task, but does provide increased performance over CPU implementations. Programming GPUs is likely to become easier as higher level languages are developed to access the graphics hardware. Graphics card manufacturers, such as NVIDIA, are developing additional tools to help program these cards using C APIs (Application Programming Interfaces) to access the hardware. As part of our continued work, we are now investigating how NVIDIA's CUDA (Compute Unified Device Architecture) framework might compare in performance and functionality to our current implementation. For our future work, we plan to optimize our system for increased functionality, including support for generalized building structures, greater than 10 million particles, and multi-GPU configurations. Multi-GPU arrangements may help with speeding up the advection of an increased number of particles since the work could be spread across the set of GPUs.

3.6 References

Feng, Q., and Coauthors, 2004: Dispersion simulation and visualization for urban security. *Visualization, 2004. IEEE*, 553-560.

Galoppo, N., N. K. Govindaraju, M. Henson, and D. Manocha, 2005: LU-GPU: Efficient Algorithms for Solving Dense Linear Systems on Graphics Hardware. *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, IEEE Computer Society, 3.

Goodnight, N., C. Woolley, G. Lewin, D. Luebke, and G. Humphreys, 2005: A multigrid solver for boundary value problems using programmable graphics hardware. *ACM SIGGRAPH 2005 Courses*, ACM, 193.

Harris, M. J., W. V. Baxter, T. Scheuermann, and A. Lastra, 2003: Simulation of cloud dynamics on graphics hardware. *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, Eurographics Association, 92-101.

Kaplan, H., and N. Dinar, 1996: A lagrangian dispersion model for calculating concentration distribution within a built-up domain. *Atmospheric Environment*, **30,** 4197-4207.

Kipfer, P., M. Segal, and R. U. Westermann, 2004: UberFlow: a GPU-based particle engine. *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, Grenoble, France, ACM, 115-122.

Näslund, E., H. C. Rodean, and J. S. Nasstrom, 1994: A comparison between two stochastic diffusion models in a complex three-dimensional flow. *Boundary-Layer Meteorology*, **67,** 369-384.

Norgren, A. P., 2008: GPU based particle dispersion modeling with interactive visualization support for real-time simulation. M.S. thesis, Dept. of Computer Science, University of Minnesota, Duluth.

Ohba, M., W. H. Snyder, and R. E. Lawson, 1993: Study on prediction of gas concentrations around twin high-rise buildings using windtunnel techniques. USEPA.

Pope, S. B., 2000: *Turbulent Flows.* Cambridge University Press, 771 pp.

Press, W. H., S. A. Teukolskay, W. T. Vetterling, and B. P. Flannery, 2007: *Numerical Recipes: The Art of Scientific Computing.* 3rd ed. Cambridge Univ. Press, 1235 pp.

Röckle, R., 1990: Bestimmung der Stromungsverhaltnisse im Bereich komplexer Bebauungsstrukturen . Ph.D. dissertation, Vom Fachbereich Mechanik, der Technischen Hochschule Darmstadt, Germany.

Rodean, H. C., 1996: *Stochastic Lagrangian models of turbulent diffusion.* The American Meteorological Society, 82 pp.

Rost, R. J., 2006: *OpenGL Shading Language.* 2nd ed. Addison-Wesley Professional.

Scheidegger, C. E., J. L. D. Comba, and R. D. Da Cunha, 2005: Practical CFD Simulations on Programmable Graphics Hardware using SMAC. *Computer Graphics Forum*, **24,** 715-728.

Seinfeld, J. H., and S. N. Pandis, 1998: *Atmospheric chemistry and physics: from air pollution to climate change.* John Wiley and Sons, Inc., 1326 pp.

Shirley, P., and S. Marschner, 2009: *Fundamentals of computer graphics.* A. K. Peters, 392 pp.

Shreiner, D., M. Woo, J. Neider, and T. Davis, 2007: *OpenGL Programming Guide.* 6th ed. Addison-Wesley Professional, 928 pp.

Singh, B., M. D. Williams, E. R. Pardyjak, and M. J. Brown, 2004: Testing of an urban Lagrangian dispersion model using Gaussian and non-Gaussian solutions. *4th AMS Symp. Urban Env.*, Norfolk, VA.

Singh, B., B. Hansen, M. Brown, and E. Pardyjak, 2008: Evaluation of the QUIC-URB fast response urban wind model for a cubical building array and wide building street canyon. *Environmental Fluid Mechanics*, **8,** 281-312.

Willemsen, P., A. Norgren, B. Singh, and E. R. Pardyjak, 2007: Development of a new methodology for improving urban fast response Lagrangian dispersion simulation via parallelism on the graphics processing unit. *In Proceedings of the 11th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes*, Queen's College, University of Cambridge, United Kingdom.

Willemsen, P., A. Norgren , B. Singh, E. R. Pardyjak, 2008: Integrating particle dispersion models into real-time virtual environments. *In Proceedings of the 14th Eurographics Symposium on Virtual Environments*, Eindhoven, Netherlands, 57-63.

Williams, M. D., M. J. Brown, D. Boswell, B. Singh, and E. M. Pardyjak, 2004: Testing of the QUIC-plume model with wind-tunnel measurements for a high-rise building. *5th AMS Urban Env. Conf.*, Vancouver, BC.

Wilson, J. D., and B. L. Sawford, 1996: Review of Lagrangian stochastic models for trajectories in the turbulent atmosphere. *Boundary-Layer Meteorology*, **78,** 191-210.

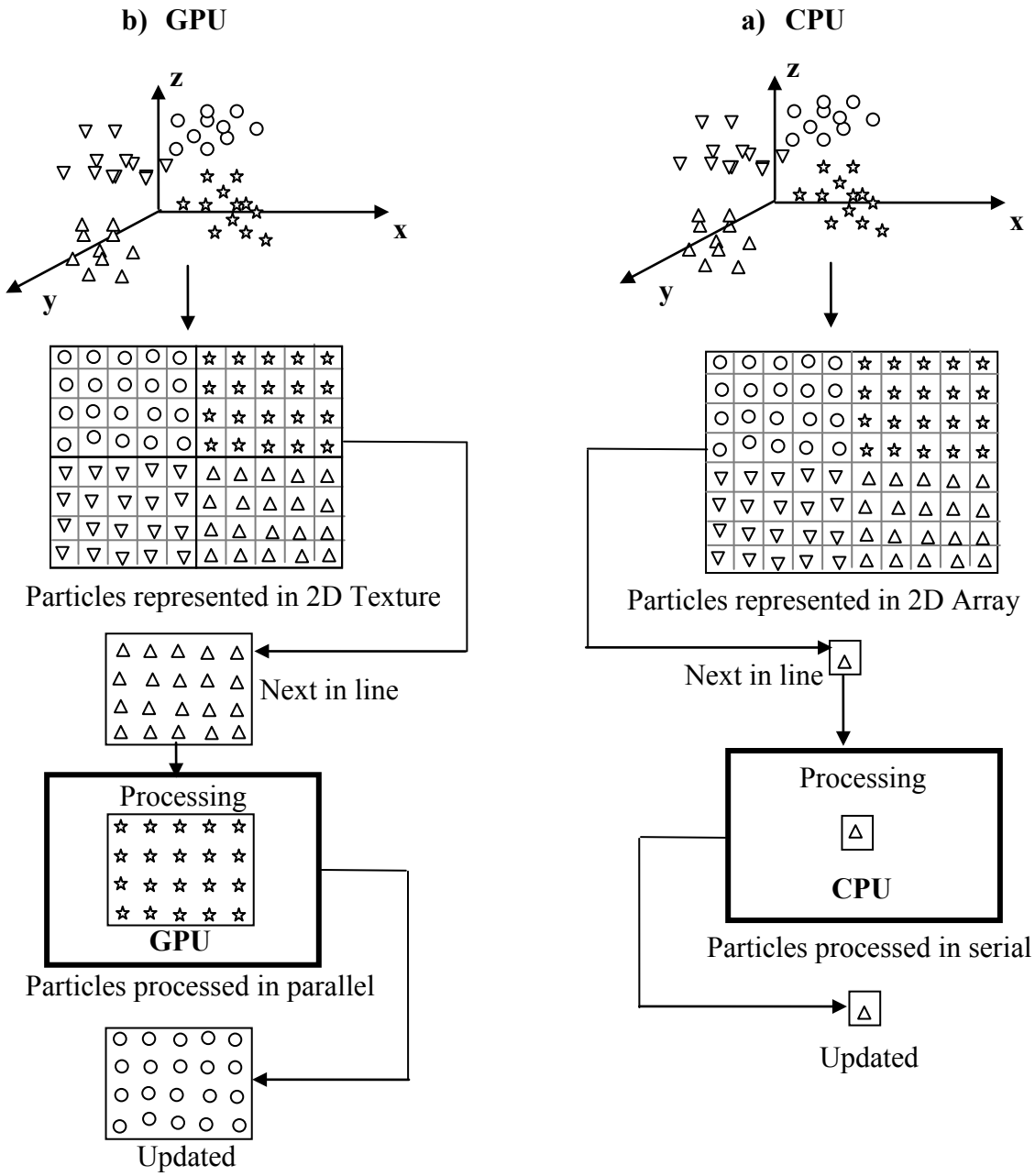


Figure 3.1: Illustration contrasting a Lagrangian particle model's advection calculation on (a) a GPU and (b) a standard CPU.

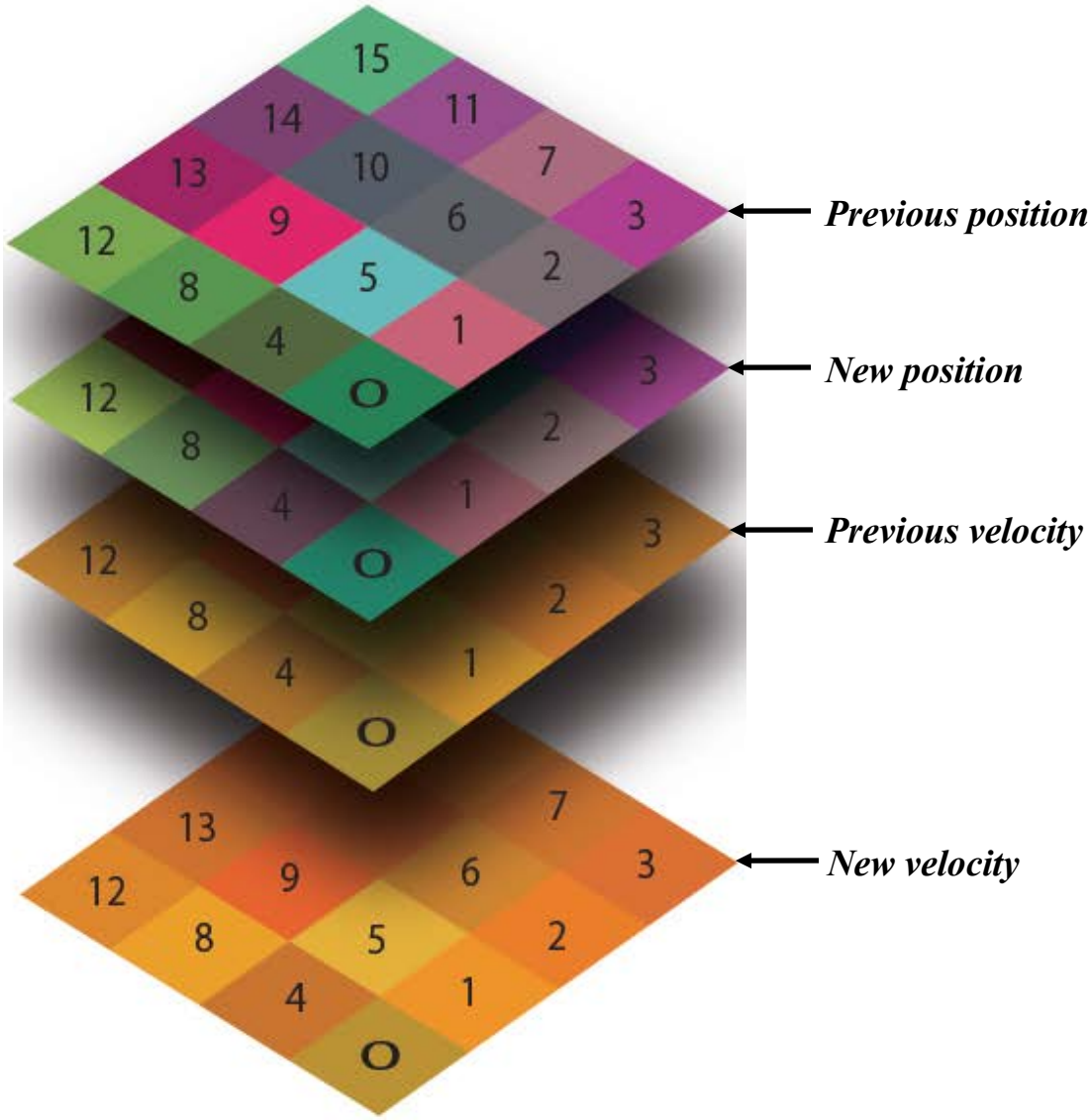


Figure 3.2: Rectangular 2D textures that store particle position and velocity data. Each colored square can hold up to four attributes such as the *x*, *y* and *z* locations of a particle and a flag that specifies weather the particles is in the computational domain. Parallelism is achieved by programming the GPU to operate on the 2D textures.

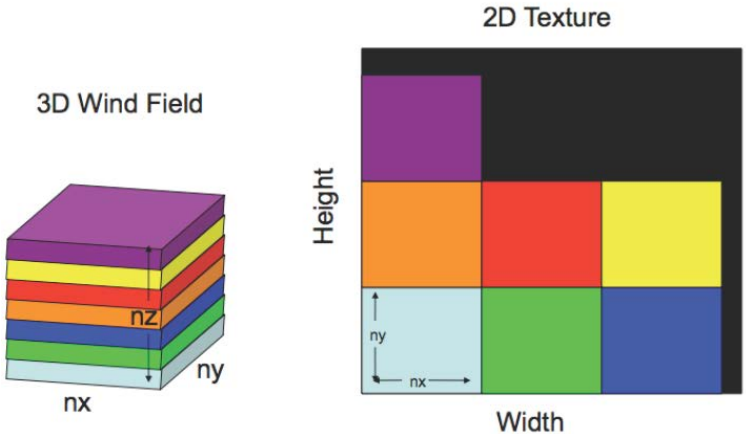


Figure 3.3: Storage of 3D domain data in a 2D texture.

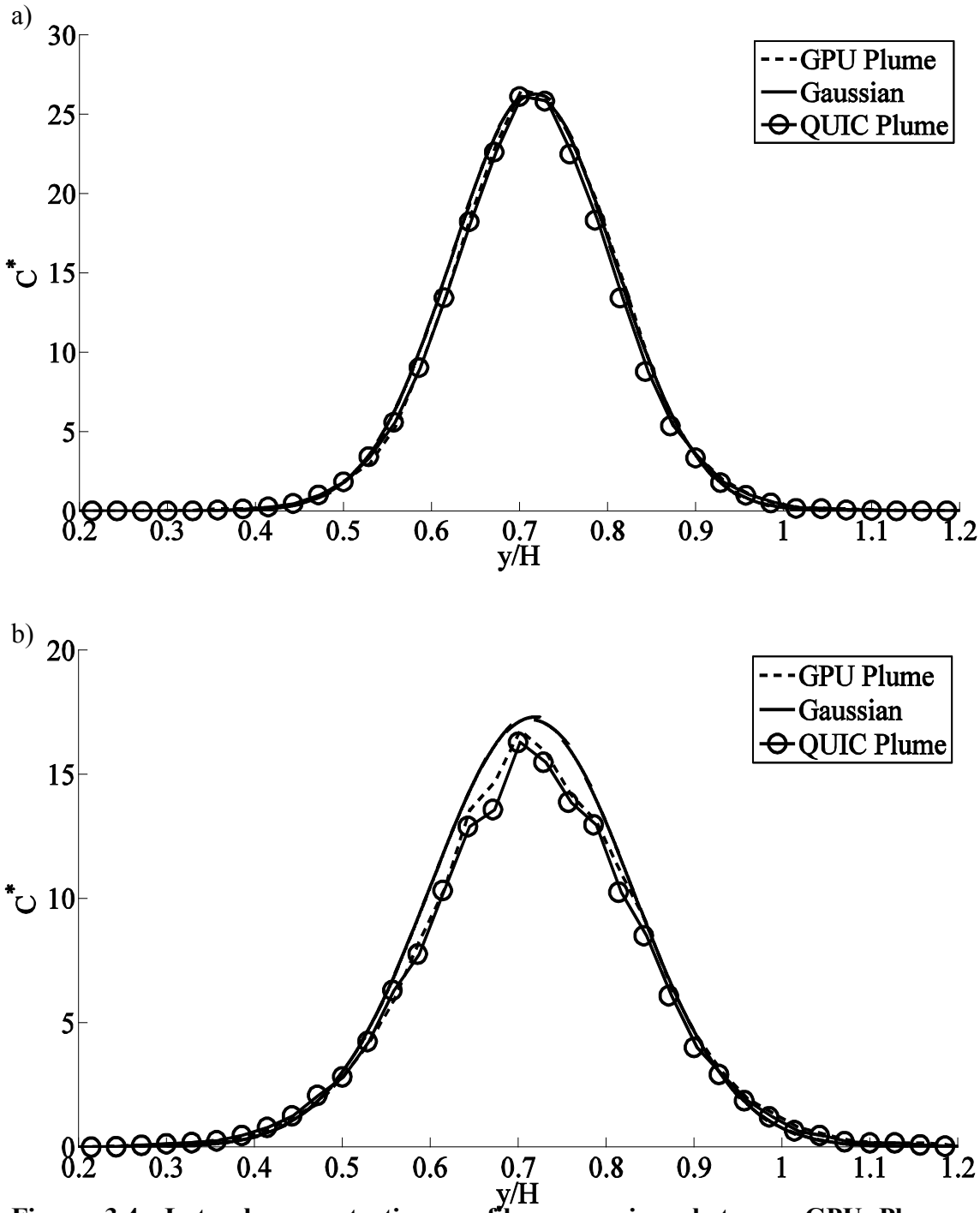


Figure 3.4: Lateral concentration profile comparison between GPU Plume, Gaussian Solution and QUIC Plume at a) x/H=0.625 b) x/H=0.982.

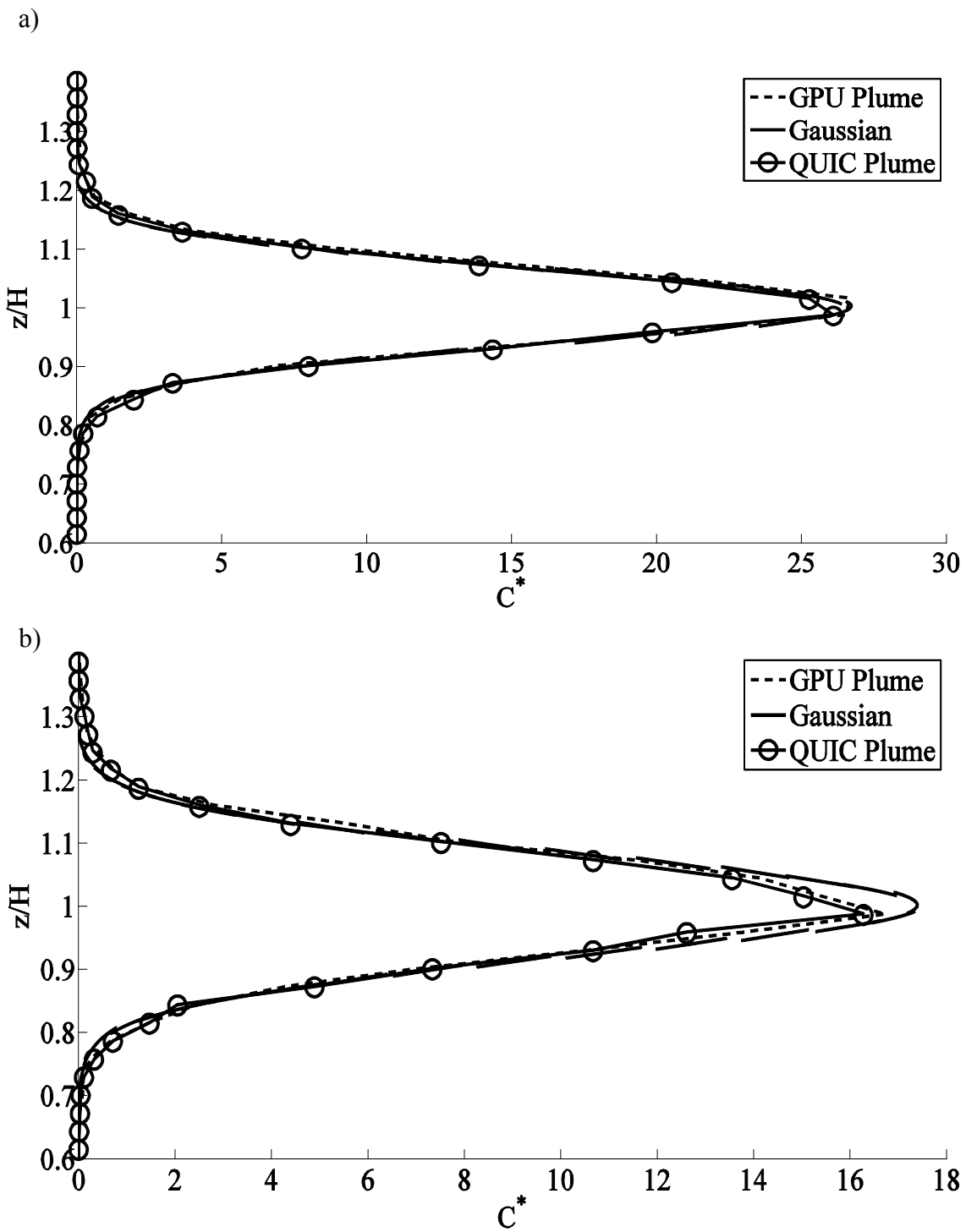


Figure 3.5: Vertical concentration profile comparison between GPU Plume, Gaussian Solution and QUIC Plume at a) x/H = 0.625 b) x/H=0.982.

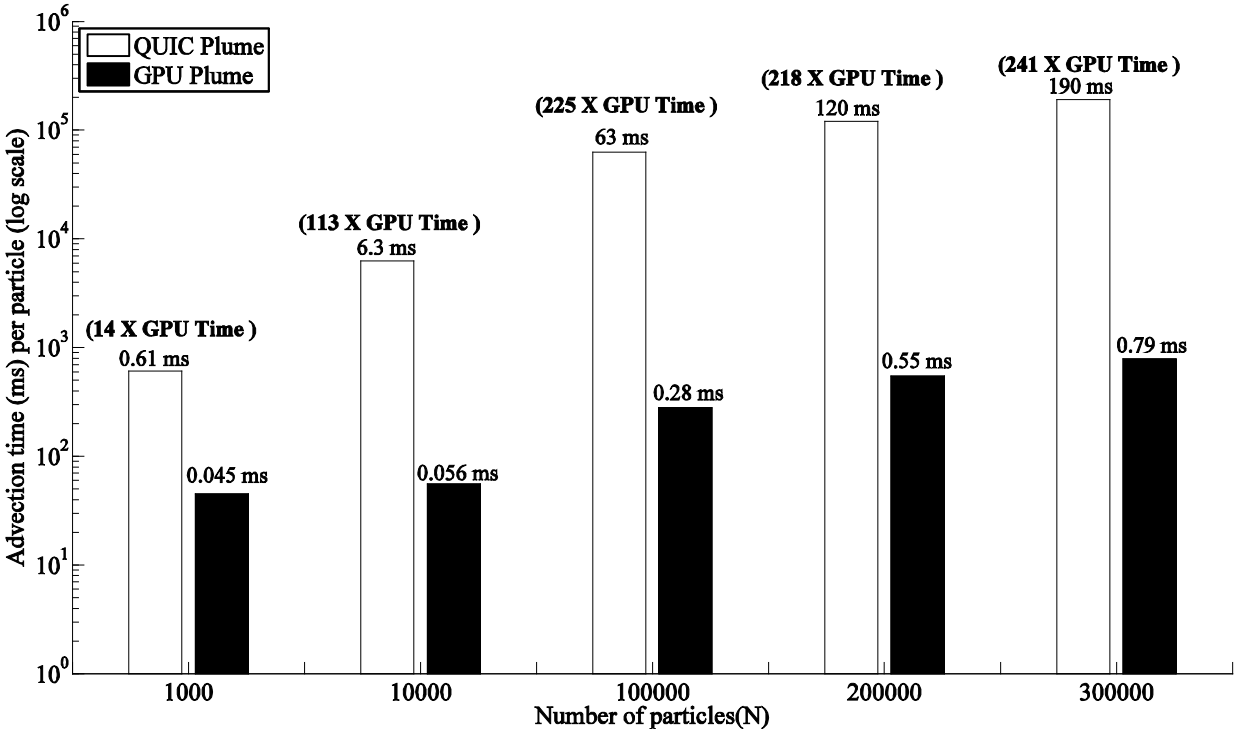


Figure 3.6: Comparison of the average time required to advect N particles in GPU Plume and QUIC Plume without visualization.

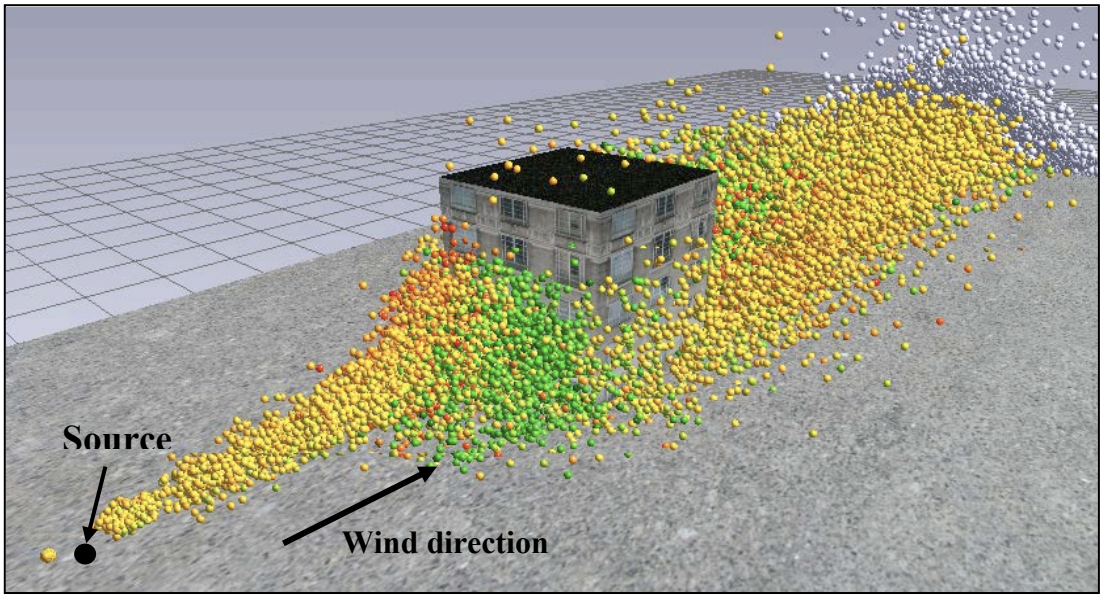


Figure 3.7: Screen-shot of a continuous point source release upstream of an isolated cube.

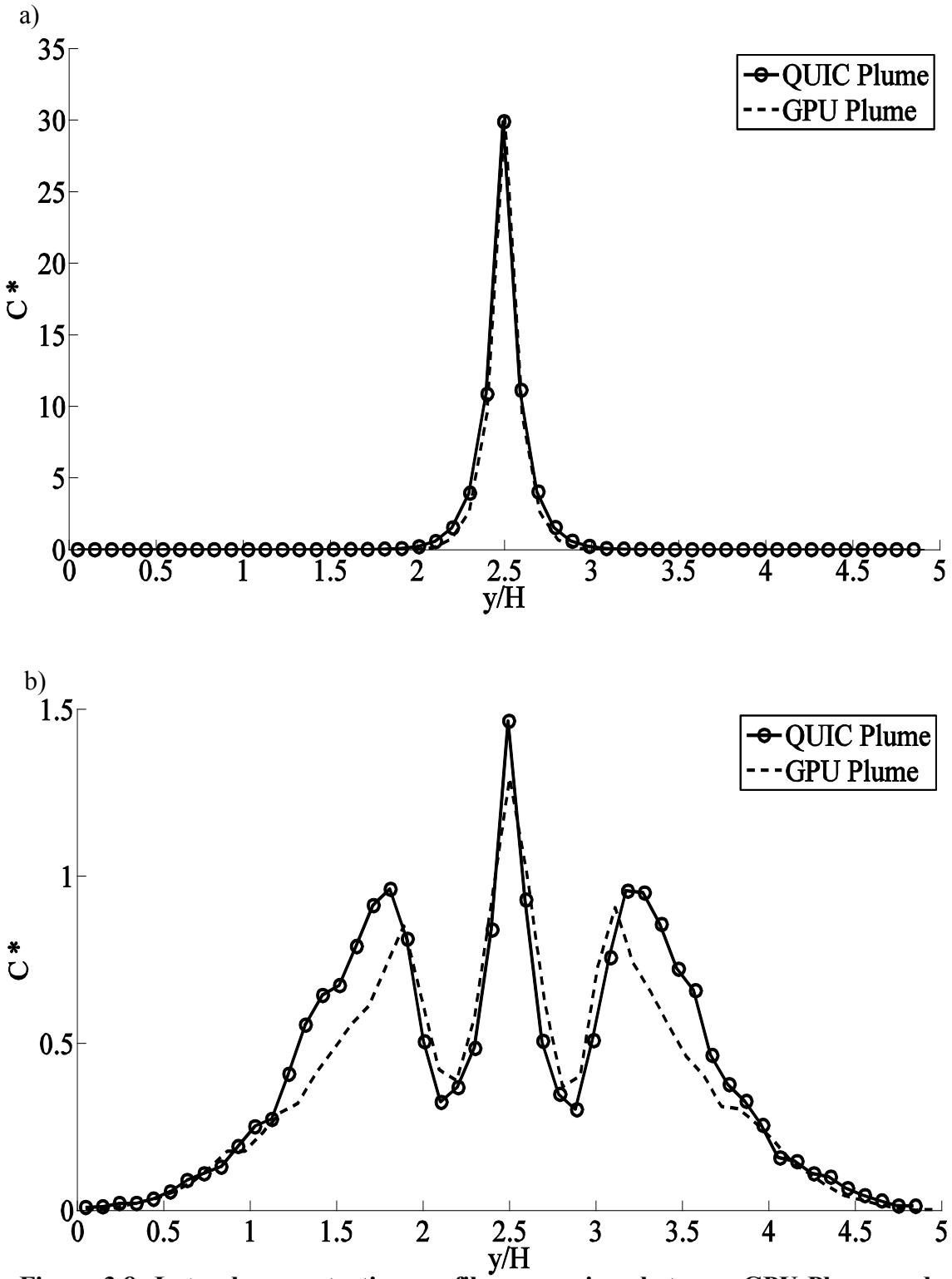


Figure 3.8: Lateral concentration profile comparison between GPU Plume and QUIC Plume at a) x=9.75 m b) x=57.25 m c) x=87.75 and z=2.5 m.

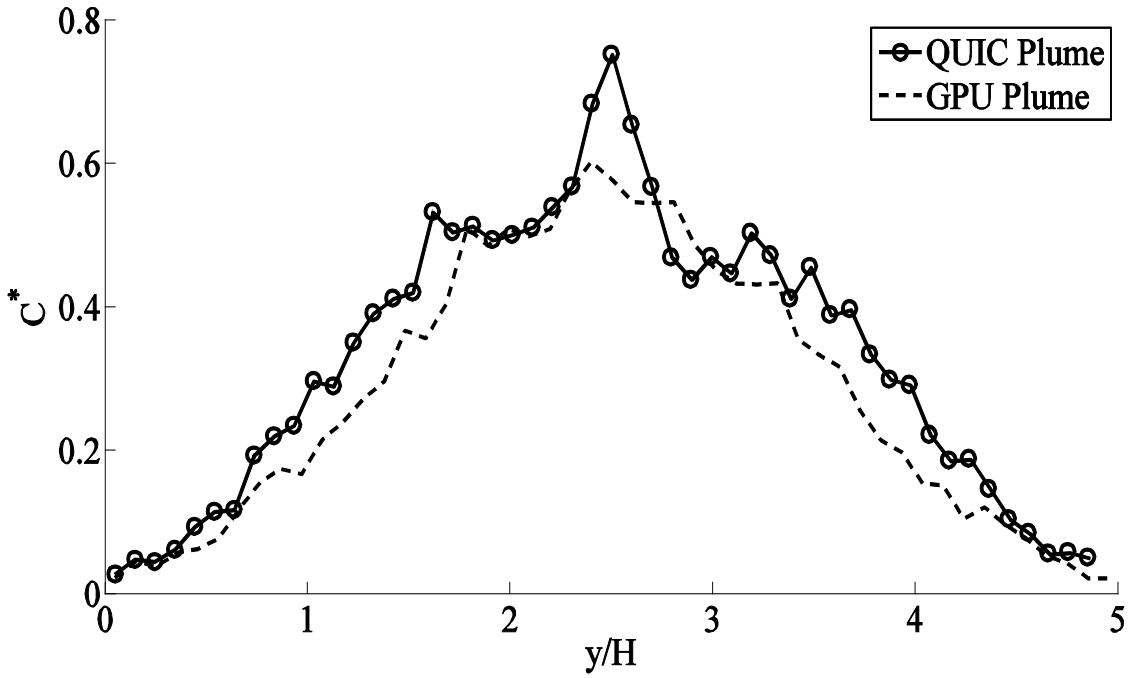


Figure 3.8: Continued.

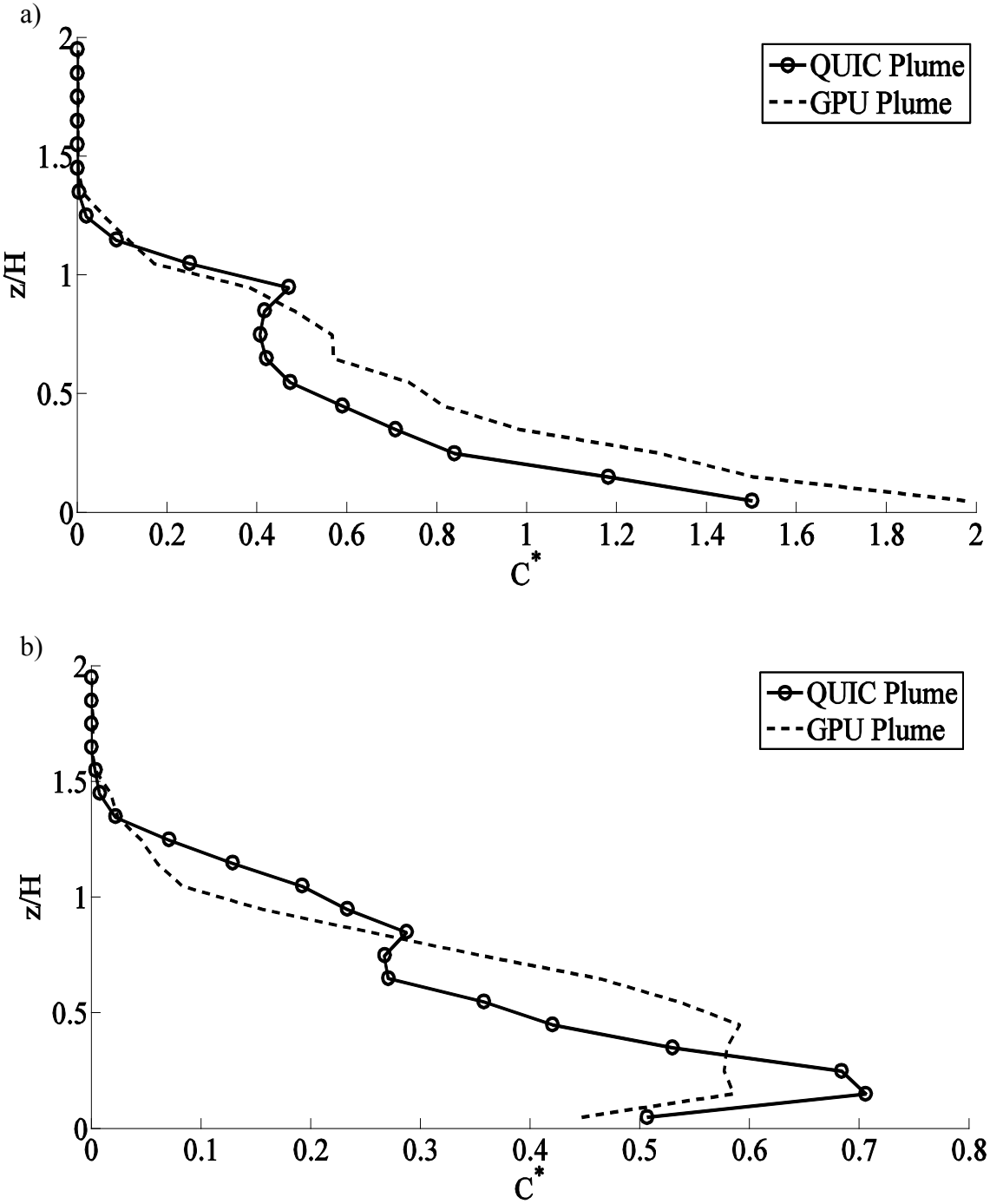


Figure 3.9: Vertical concentration profile comparison between GPU Plume and QUIC Plume at a) x=57.25 m b) x=85.75 m and z=2.5 m.

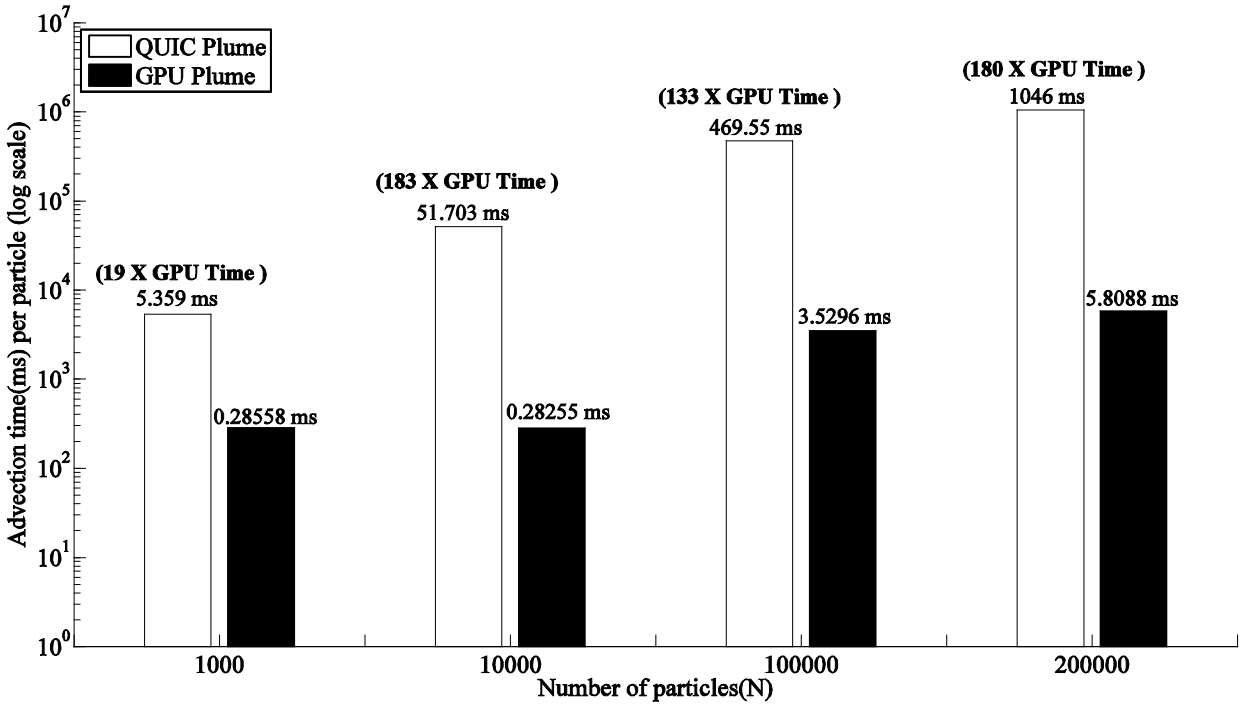


Figure 3.10: Comparison of the average time required to advect N particles in GPU Plume and QUIC Plume without visualization.

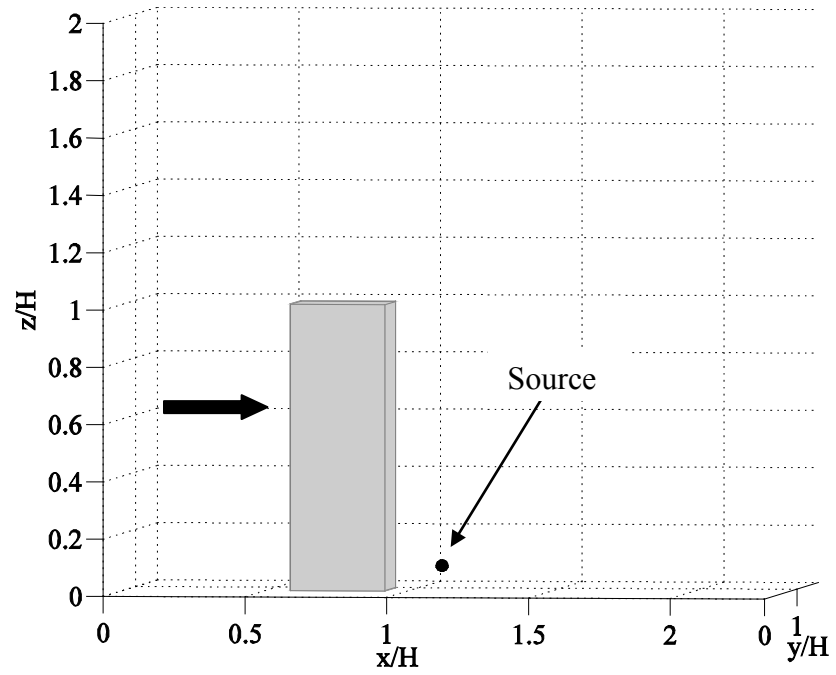


Figure 3.11: Schematic of the Hi-Rise building test case. Arrow indicates the inflow wind direction.

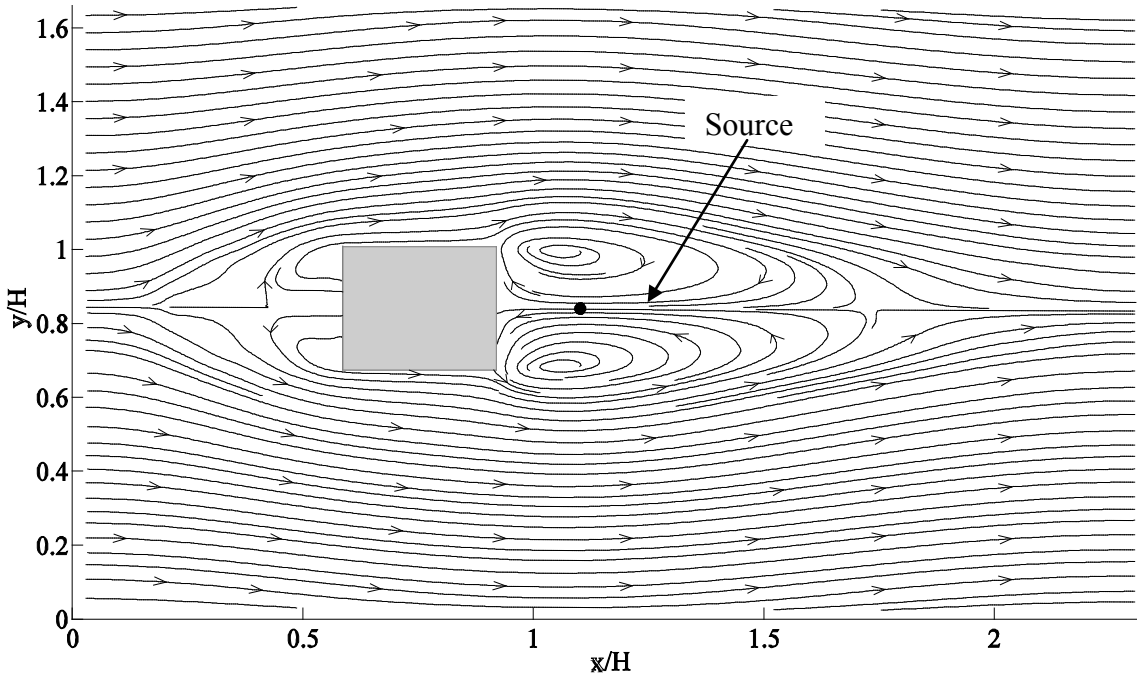


Figure 3.12: Schematic showing source immersed in the streamline pattern in a plan view of the Hi-Rise building test case.

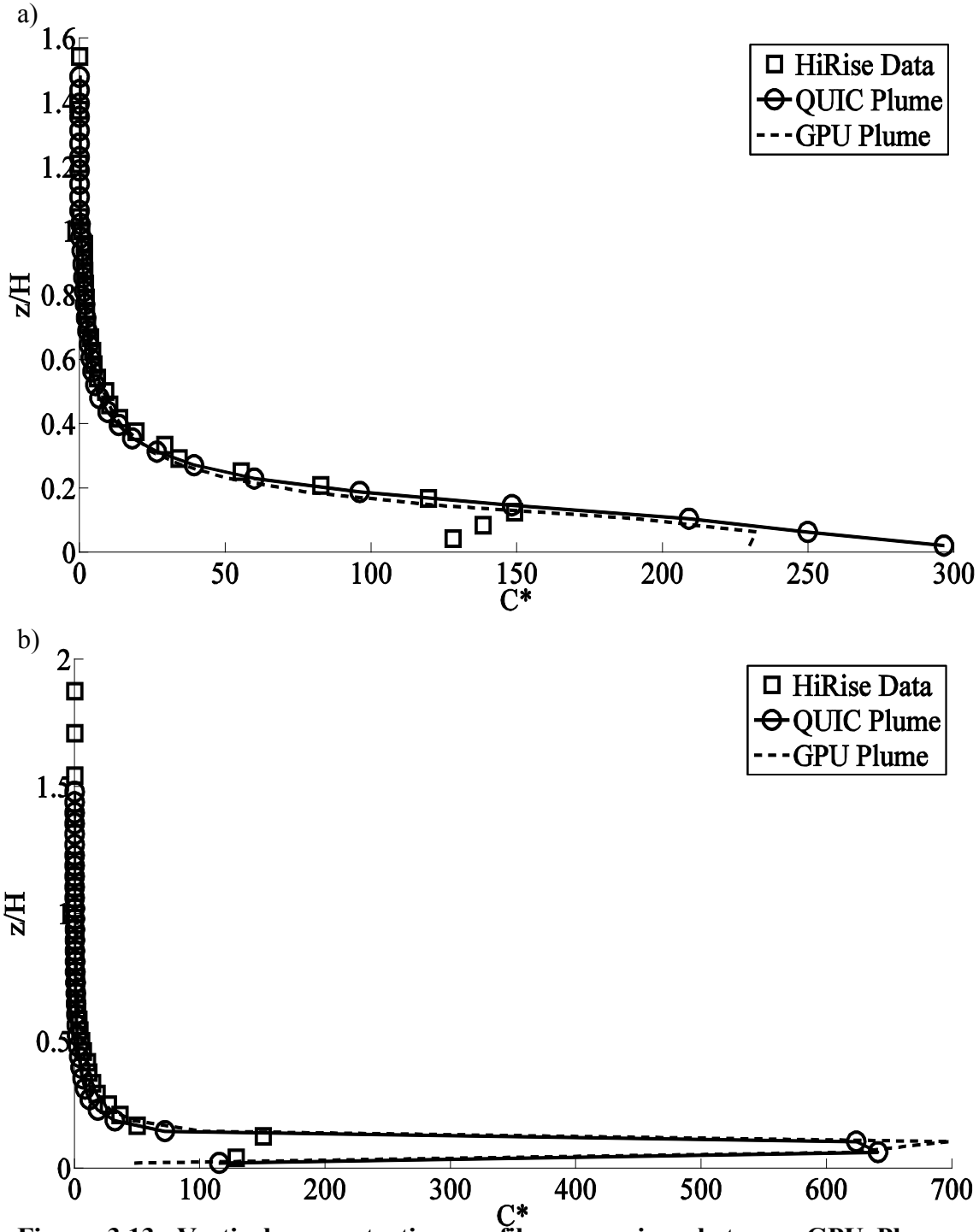


Figure 3.13: Vertical concentration profile comparison between GPU Plume, QUIC Plume and Hi-Rise experimental data at a) x/H=0.04 b) x/H=0.17 c) x/H=0.42 d) x/H=0.58 and centerline of the building.

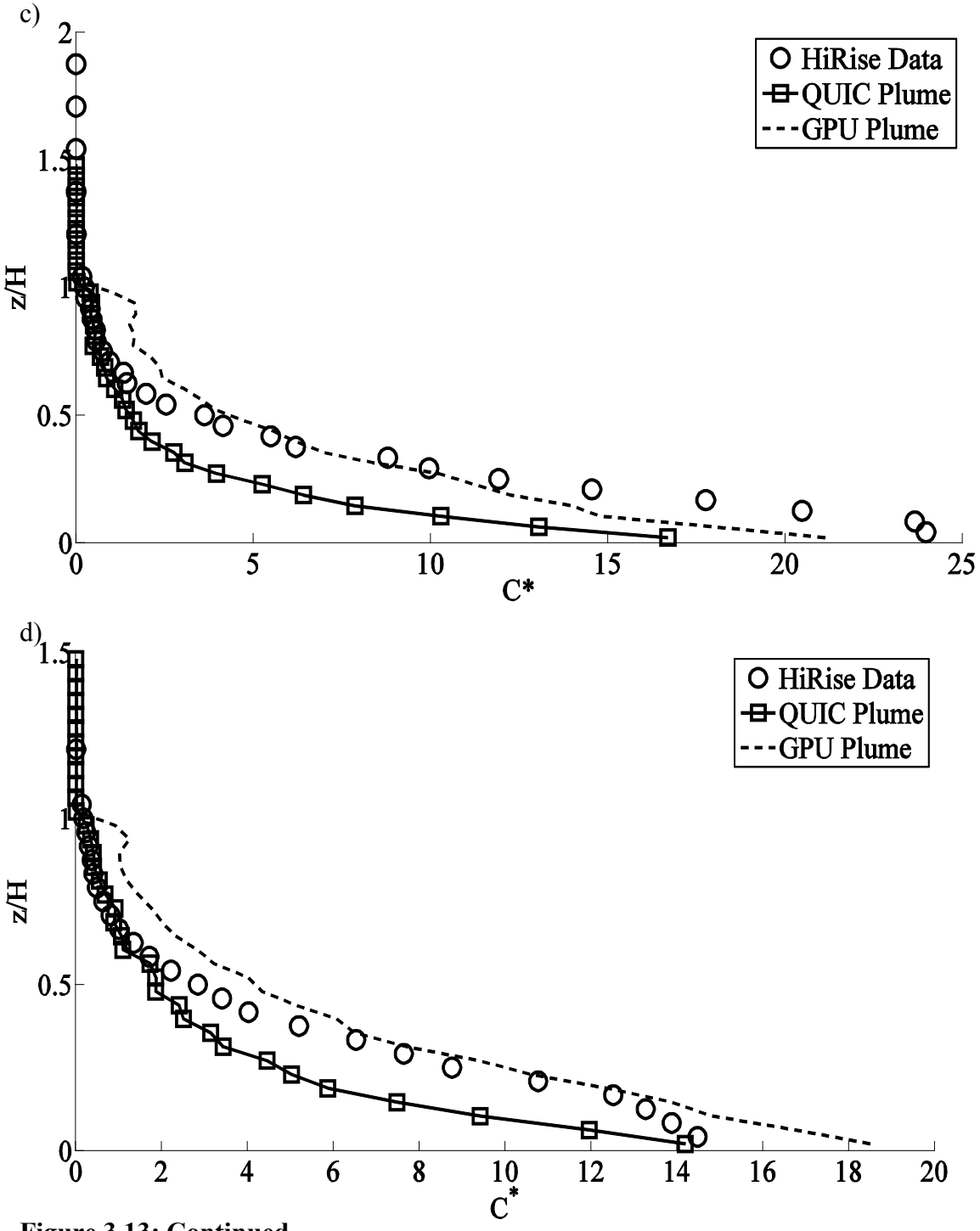


Figure 3.13: Continued.

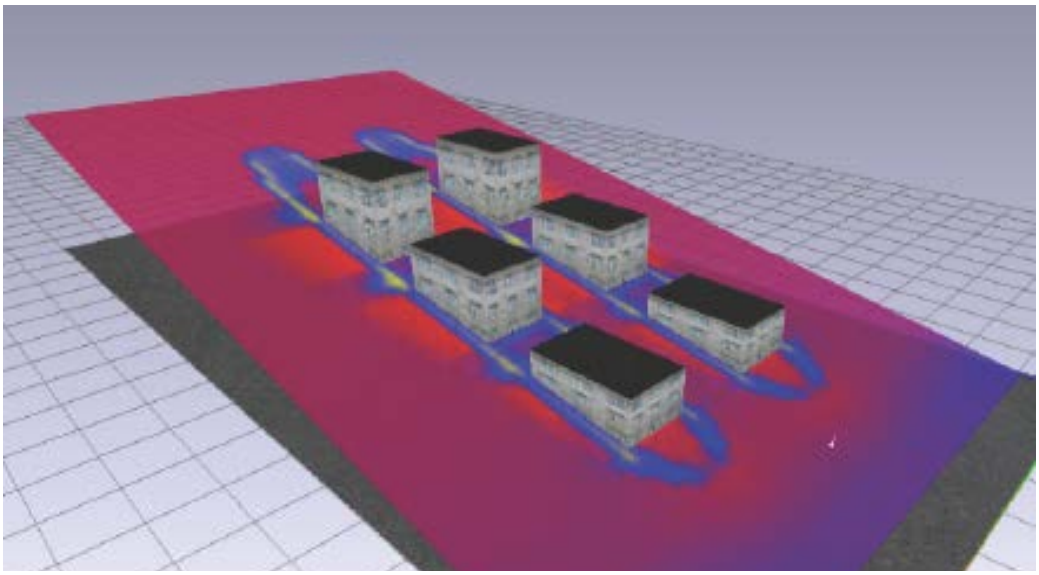


Figure 3.14: Screen-shot showing user's view while interacting with a turbulence contour in real-time.

Table 3.1*:* Textures used in the GPU Plume along with the four values stored at each node. Note that there are two position and prime textures that are for the current and previous time steps.
**Random number*

Textures	**R_{Value}**	**G_{Value}**	**B_{Value}**	**Alpha**
Wind	u	v	W	Dissipation(ε)
turbulence (1)	$\partial\tau_{11}/\partial z$	$\partial\tau_{22}/\partial z$	$\partial\tau_{33}/\partial z$	$\partial\tau_{13}/\partial z$
turbulence (2)	$\partial u/\partial z$	u_*	-	-
turbulence (3)	λ_{11}	λ_{22}	λ_{33}	λ_{13}
particle position	x	y	Z	Active/inactive
prime	u'	v'	w'	-
random	1^{st} RN*	2^{nd} RN*	3^{rd} RN*	-
celltype	x	y	Z	Free/surface cell

4. INTEGRATING THE GENERALIZED FORM OF LANGEVIN EQUATIONS INTO AN URBAN LAGRANGIAN DISPERSION MODEL

4.1 Introduction

In the previous chapter, an emergency response dispersion modeling system based on a semi-empirical approach on a Graphics Processing Unit (GPU)-based architecture was discussed. The GPU-based architecture helps in enhancing simulation efficiency in terms of computational time while maintaining the accuracy of the results. Another important advantage of utilizing GPU-based architecture is the ability to achieve real-time visualization. The current GPU Plume model utilizes the simplified Langevin equations (SLEs) (Rodean 1996) based on the assumptions of a horizontally homogenous flow (i.e., gradients of velocity are only in the vertical direction). The SLEs drastically reduce the number of terms in the Langevin equations for each component of the fluctuating velocity, and hence are relatively easy to compute. To apply these SLEs to a complex flow field encountered in urban areas, the coordinate system must be aligned in the direction of the streamlines to partially accommodate the horizontally homogenous flow assumptions (Williams et al. 2004). Figure 4.1 illustrates the procedure to align the coordinate system in the direction of the streamlines.

The x-y-z coordinate system (original coordinate system) is first rotated in the x-

y plane by an angle of ψ in the direction of the mean wind and the new coordinate system formed is identified as x'-y'-z'. The x'-z' plane is then rotated to x''-z'' by an angle ϕ such that the x'' direction is aligned with the mean velocity vector $\vec{U}$. The new coordinate system x''-y''-z'' is the final coordinate system where the velocity gradients are assumed to be only in the vertical direction (z''). The positive z'' direction in the x''-y''-z'' coordinate system is the direction of maximum mean velocity gradient, $\frac{\partial \vec{U}}{\partial z''}$. This approach renders the gradients in the crosswind and streamwise direction to be zero; therefore, the number of terms in the full or generalized form of Langevin equations (GLEs) can be drastically reduced (Rodean 1996).

The GLEs are considered stiff (i.e., existence of unstable modes where a particle may travel significant distances over a small time step) due to the presence of a wide range of rate constants or time scales in the simulation (Yee and Wilson 2007). The SLEs drastically reduce the number of terms in the GLEs; therefore, the SLEs are considered less unstable; however, unstable modes still exist in the SLEs (Yee and Wilson 2007).

The above approach of simplifying the GLEs may not be valid for the complex flow field present in urban areas. We hypothesize that to fully describe the dispersion in urban areas, the 3D generalized Langevin equations (3D GLEs) (Thomson 1987; Rodean 1996; Wilson and Sawford 1996; Yee and Wilson 2007) must be utilized without the horizontally homogenous assumptions.

This chapter discusses challenges associated with the implementation of the GLEs into an existing Lagrangian random-walk dispersion model. The preliminary results obtained from two idealized baseline test cases to validate the performance of the 3D GLEs Lagrangian dispersion model against available analytical solutions for simple

flows are presented. A third test case is presented for a more realistic urban setup where dispersion estimates from the 3D GLEs model are compared with a wind-tunnel experiment for a 7 x 11 cubical array of buildings (Brown et al. 2001).

4.2 Methodology

As discussed earlier in Chapter 3, the Langevin equations require a mean wind field and a mean turbulence field to make dispersion estimates. The mean wind field for this model was obtained from the QUIC-URB wind model and mean turbulence field was obtained by utilizing a turbulence model based on the mixing length theory (see Williams et al. (2004)).

To partially circumvent the stiffness in the GLEs, the 3D GLEs Lagrangian dispersion model utilizes a fractional step methodology (FSM) outlined by Yee and Wilson (2007). The following section outlines the basic methodology for solving the GLEs using the FSM. For further details on the solution method, please refer to Yee and Wilson (2007).

4.3 Fractional step method

This section is included for the sake of completeness. For complete derivation and further details on the FSM, please refer to Yee and Wilson (2007). The notation used by Yee and Wilson (2007) is followed here to explain the FSM. The matrices are denoted by bold upper case symbols with the only exception of $\mathbf{U}$, which is used to denote Lagrangian fluctuating velocity vector. The bold lower case symbols represent the vectors. Einstein notation is also being used where i, j, k can take values of 1, 2 or 3.

The GLEs can be written in the index (Einstein) notation as follows(Yee and Wilson 2007),

$$dU_i = \left(T_i^{(0)} + T_{ij}^{(1)}U_j + T_{ijk}^{(2)}U_jU_k\right)dt + \left(C_0\varepsilon\right)^{1/2} d\xi_i \tag{4.1}$$

Eq. (4.1) is written in terms of three expressions which are classified in terms of their relationship to the fluctuating component of velocity, represented by U_i. The first expression, $T_i^{(0)}$ (ms^{-2}, with a tensor rank of 1), is considered as a constant term in U_i. The second expression, $T_{ij}^{(1)}$ (s^{-1}, with a tensor rank of 2), is linear in terms of U_i and the third term $T_{ijk}^{(2)}$ (m^{-1}, with a tensor rank of 3) is nonlinear in terms of U_i. The last expression, $\left(C_0\varepsilon\right)^{1/2} d\xi_i$, represents the contribution of random fluctuation to the motion of a particle. The random Gaussian variate with mean equal to zero and variance equal to $\sqrt{dt}$ is represented by $d\xi_i(t)$.

The above expressions can be further written in terms of the turbulence stresses as,

$$T_i^{(0)} = \frac{1}{2}\frac{\partial R_{il}}{\partial x_l} \tag{4.2}$$

$$T_{ij}^{(1)} = -\frac{1}{2}\left(C_0\varepsilon\right)R_{ij}^{-1} + \frac{1}{2}R_{lj}^{-1}\frac{\partial R_{il}}{\partial x_k}\overline{u}_k \tag{4.3}$$

$$T_{ijk}^{(2)} = \frac{1}{2}R_{lj}^{-1}\frac{\partial R_{il}}{\partial x_k} \tag{4.4}$$

The turbulent stress tensor is denoted by $R_{ij}\,(m^2 s^{-2})$ in the above equations. The dissipation of turbulent kinetic energy is represented by ε and C_o is a universal constant for Lagrangian structure function (Rodean 1996; Pope 2000). The matrix inverse of the stress tensor R_{ij} is represented by R_{lj}^{-1} and $\overline{u}_k$ is the mean velocity.

In the 3D GLEs model, the particles are advected at each time step (Δt) by using the mean wind velocity and the fluctuating components of velocity in the respective directions. The advection is accomplished by employing the following equation,

$$x_i(t+\Delta t) = x_i(t) + [\overline{u}_i + U_i]\Delta t \tag{4.5}$$

In the above equation, the mean wind velocity ($\overline{u}_i$) is obtained at position $x_i(t)$. The fluctuating components of the velocity (U_i) are obtained by solving the GLEs.

The FSM involves three steps. In each step, an intermediate fluctuating component of the velocity is evaluated which is used as an input for the next step. These three steps can be symbolically expressed as,

$$U^* = \Im_1(U(t), \Delta t) \tag{4.6}$$

$$U^{**} = \Im_2(U^*, \Delta t) \tag{4.7}$$

$$U(t+\Delta t) = \Im_3(U^{**}, \Delta t) \tag{4.8}$$

An intermediate fluctuating component of the velocity is obtained from the first step (U^*) and is used as an input to the second step. The result from the second step

(U^{**}) is used to evaluate the final step to obtain the fluctuating component of the velocity (U) at the time step $(t+\Delta t)$.

The first two steps are obtained analytically for a finite time step Δt (see Yee and Wilson (2007) for details). That is, there is an analytical mapping available for the functions $\Im_1$ and $\Im_2$ (discussed later). The last step employs the simple forward Euler method to evaluate the fluctuating component of the velocity.

The FSM is applied by "splitting" the fluctuating component of velocity increments as follows:

$$dU_i = dU_i^{(1)} + dU_i^{(2)} + dU_i^{(3)} \tag{4.9}$$

where,

$$dU_i^{(1)} = \left(T_i^{(0)} + T_{ij}^{(1)} U_j\right) dt + \left(C_o \varepsilon\right)^{1/2} d\xi_i \tag{4.10}$$

$$dU_i^{(2)} = T_{(i)(i)(i)}^{(2)} U_{(i)} U_{(i)} dt \tag{4.11}$$

$$dU_i^{(2)} = T_{ijk}^{(2)} U_j U_k dt \text{ for } [i,j,k]/\{i=j=k\} \tag{4.12}$$

In Eq. (4.10), $dU_i^{(1)}$ is evaluated from the constant and linear terms ($T_i^{(0)}$ and $T_{ij}^{(1)} U_j$) along with the nondeterministic random fluctuation term. The $dU_i^{(1)}$ combines the complete memory and random terms of the GLEs and a part of drift terms (Rodean 1996) which is constant and linear in terms of U_i. Eq. (4.10) evaluates $dU_i^{(2)}$ from the

nonlinear drift term ($T^{(2)}_{(i)(i)(i)}U_{(i)}U_{(i)}$) where the bracket (i) represents that there is no summation over the enclosed indices. That is, Eq. (4.11) only evaluates the nonlinear drift terms for $\{i = j = k\}$ (diagonal terms). For all other cases (non-diagonal drift terms), Eq. (4.12) evaluates the nonlinear terms in U_i.

The above decomposition into three steps is performed since analytical mapping for the first two steps (Eqs. (4.10) and (4.11)) is available which helps in circumventing the unstable modes in the GLEs and hence solves the stiffness problem encountered in solving the GLEs. No analytical mapping is available for the last step (Eq. (4.12)); therefore, this step does not guarantee the circumvention of the existing unstable modes. Hence, the FSM only partially addresses the stiffness problem encountered in the 3D GLEs.

4.3.1 First step: Integration of constant and linear terms

In the first step, the contributions from the constant and linear terms in U_i are obtained by an analytical mapping function $\Im_1$. Eq. (4.10) can be rewritten as (using matrix notation),

$$d\mathbf{U} = \mathbf{a_0}dt + \mathbf{A_1}\mathbf{U}dt + \mathbf{B}d\mathbf{w} \tag{4.13}$$

where,

$$\mathbf{a_0} \equiv \frac{1}{2}\frac{\partial R_{ij}}{\partial x_j} \tag{4.14}$$

$$\mathbf{A_1} \equiv -\frac{1}{2}\left(C_o\varepsilon\right)R_{ij}^{-1} + \frac{1}{2}R_{lj}^{-1}\frac{\partial R_{il}}{\partial x_k}\overline{u}_k \tag{4.15}$$

$$\mathbf{B} \equiv \left(C_o\varepsilon\right)^{1/2}\delta_{ij} \tag{4.16}$$

and

$$d\mathbf{w} \equiv \left(d\xi_1, d\xi_2, d\xi_3\right)$$

For employing the analytical mapping function $\Im_1$, the eigenvalues of matrix $\mathbf{A_1}$ are obtained by eigenvalue-eigenvector decomposition as,

$$\mathbf{A_1} = \mathbf{S}\Lambda\mathbf{S}$$

where,

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \tag{4.17}$$

and $\mathbf{S}$ is the corresponding eigenvector matrix. The eigenvectors in the $\mathbf{S}$ matrix are normalized to have a unit length.

According to Yee and Wilson (2007), for general cases, the $\mathbf{A_1}$ matrix is not always symmetric, i.e., if the second term ($\frac{1}{2}R_{lj}^{-1}\frac{\partial R_{il}}{\partial x_k}\overline{u}_k$) in $\mathbf{A_1}$ is non-zero, then the $\mathbf{A_1}$

matrix is nonsymmetric. For a nonsymmetric matrix, the presence of real eigenvalues is not always guaranteed. There may be some cases where rather than having a single real eigenvalue, we may get two complex eigenvalues occurring as a complex conjugate pair. The complex eigenvalues for the $\mathbf{A_1}$ matrix indicate the presence of resonant or oscillatory modes in the dynamical system (Yee and Wilson 2007). Also, the presence of positive eigenvalues (signifying that a particle's velocity is not relaxing towards its mean velocity) indicates an unstable system where the particles may experience unstable modes during their trajectory.

Experience with a number of test cases suggests that the cases where the $\mathbf{A_1}$ matrix has complex eigenvalues are extremely rare (Yee and Wilson 2007). Complex eigenvalues may occur for the cases where turbulence field is not realizable (Vreman 1994). In some rare cases where the turbulence field is realizable and still the eigenvalues of the $\mathbf{A_1}$ matrix are complex, the second term of the $\mathbf{A_1}$ matrix can be dropped to yield a symmetric $\mathbf{A_1}$ matrix which guarantees real eigenvalues (although this is an extreme step and would certainly violate the well-mixed criterion for the 3D GLEs (Thomson 1987) as few drift terms are neglected – drift terms are essential for maintaining well-mixed criterion).

The presence of positive eigenvalues may also cause unstable modes in the trajectory of a particle. Due to these unstable modes, a particle may change its position significantly over a small time step. For simple test cases, where velocity gradient is in a different direction than the mean flow direction (e.g., boundary layer flows, where velocity gradients are in vertical direction and mean flow is in horizontal direction),

unstable modes due to this first step of the FSM are not present. In such cases, the second term ($\frac{1}{2} R_{lj}^{-1} \frac{\partial R_{il}}{\partial x_k} \overline{u}_k$) will always be zero, rendering the $\mathbf{A_1}$ matrix to be symmetric.

To further investigate the potential reasons for the presence of positive eigenvalues, another simple test case of a spatially developing shear layer was performed. In this test case, the velocity gradients were present in the crosswind direction due to the presence of a shear layer. The shear layer "grows" with increase in the streamwise distance relaxing the cross-stream shear layer velocity gradients. This arrangement ensures that the velocity gradients are present in both streamwise and cross-stream directions (Bell and Mehta 1990; Pope 2000). As the velocity gradients (and hence turbulence stress gradients) are present in the direction of mean flow (streamwise direction), the second term ($\frac{1}{2} R_{lj}^{-1} \frac{\partial R_{il}}{\partial x_k} \overline{u}_k$) of the $\mathbf{A_1}$ matrix is non-zero, rendering $\mathbf{A_1}$ to be a nonsymmetric matrix.

This plane mixing layer grows with the streamwise distance; therefore, the velocity profile varies as we move in the streamwise direction. In our test case, initially, the ratio between the two velocities that drove the shear layer was 0.6 (U_1=9 ms^{-1}, U_2=15 ms^{-1} with $\frac{U_1}{U_2} = 0.6$). Figure 4.2 shows the velocity profile comparison of the test data with the velocity profile used as an input for the 3D GLEs model at a streamwise distance (velocity profile remains similar for every streamwise distance as the data are plotted with similarity parameters). The normalized ordinate and abscissa for Figure 4.2 are defined by the following similarity parameters, respectively,

$$\eta = \frac{y - y_0}{\delta}$$

$$U^{\#} = \frac{U - U_1}{U_2 - U_1}$$

In the above equations, y_0 is the centerline of the mixing layer and δ is the mixing layer thickness which grows as the streamwise distance increases.

Figure 4.3 shows the mixing length turbulence model's output against the test data for this test case. The results indicate that the turbulence model is in good agreement with the test data.

The $\mathbf{A_1}$ matrix, for this test case, produced negative eigenvalues for the most part except at some locations at $\eta < -2$ and $\eta > 2$ where some of the eigenvalues were positive. Further investigation revealed that, in regions where eigenvalues are positive, the product of kinetic energy dissipation and the universal constant ($C_0\varepsilon$) is less than the gradient of the turbulent ($\frac{\partial R_{11}}{\partial x}$) stresses.

Figure 4.4 shows the comparison of the gradient of a turbulent stress ($\frac{\partial R_{11}}{\partial x}$) in streamwise direction with the product of kinetic energy dissipation and the universal constant ($C_0\varepsilon$). For the most part, $C_0\varepsilon$ is greater in magnitude than the turbulence stress in the streamwise direction, rendering negative eigenvalues at these locations. However, at some regions near $\eta < -2$ and $\eta > 2$, the gradient of turbulent stress is more than the $C_0\varepsilon$, rendering positive eigenvalues for the $\mathbf{A_1}$ matrix (see Figure 4.5). The magnitude of the positive eigenvalues is found to be close to zero for the most part for this test case.

As discussed earlier, the positive eigenvalues can be avoided if we neglect the second term of the $\mathbf{A_1}$ matrix. As the second term of $\mathbf{A_1}$ is a drift term and drift terms are imperative for maintaining well-mixed criteria (Thomson 1987; Rodean 1996), this extreme step certainly leads to the violation of well-mixed criterion. In the present model, the positive eigenvalues are accommodated by forcing the fluctuating component of the velocity during the advection process to be less than some multiple of standard deviation of velocity at that location. This technique has been used by various dispersion modelers to circumvent a significant (and unphysical) change of position of a particle during a small time step to avoid instabilities in a particle's trajectory (Yee and Wilson 2007).

For calculating the first intermediate fluctuating component of the velocity (U^*, see Eqs. (4.6)) through the mapping function $\Im_1$, the following transformation is performed,

$$\mathbf{U}^{rot} = \mathbf{S}^{-1}\mathbf{U}$$

with a reverse transformation, $\mathbf{U} = \mathbf{S}\mathbf{U}^{rot}$, Eq. (4.13) can be rewritten as,

$$d\mathbf{U}^{rot} = \Lambda\mathbf{U}^{rot}dt + \mathbf{S}^{-1}\mathbf{a_0}dt + \mathbf{S}^{-1}\mathbf{B}d\mathbf{w} \tag{4.18}$$

Eq. (4.18) can be integrated exactly over a time step Δt, to yield (see Yee and Wilson (2007) for details),

$$U_i^{rot}(t) = U_i^{rot}(t_0 + \Delta t)$$

$$= U_{i,0}^{rot} \exp(\lambda_i \Delta t) - \frac{k_i}{\lambda_i}(1 - \exp(\lambda_i \Delta t)) + \gamma_i(t) \tag{4.19}$$

In Eq. (4.19), $U_{i,0}^{rot}$ is the velocity at time t_0 which is obtained by $\mathbf{U}^{rot} = \mathbf{S}^{-1}\mathbf{U}$ transformation and $\mathbf{k} \equiv k_i = \mathbf{S}^{-1}\mathbf{a_0}$. The stochastic forcing term $\gamma_i(t)$ has the following form,

$$\gamma_i(t) = (C_0 \varepsilon)^{1/2} \sum_j S_{ij}^{-1} \exp(\lambda_i \Delta t) \int_{t_0}^{t} \exp(-\lambda_i s) d\xi_j(s) \tag{4.20}$$

The stochastic random forcing term can be obtained by generating a Gaussian random number with mean zero and with a variance defined as,

$$< \gamma_i(t) \gamma_j(t) >= \frac{C_0 \varepsilon}{(\lambda_i + \lambda_j)} (\exp((\lambda_i + \lambda_j)\Delta t) - 1) \delta_{ij} \tag{4.21}$$

where δ_{ij} is the Kronecker delta function.

The intermediate velocity vector (U^*) is finally obtained by rotating Eq. (4.19) by $\mathbf{S}$ ($\mathbf{U} = \mathbf{S}\mathbf{U}^{rot}$).

The steps outlined above enable us to integrate the first (linear and constant terms in U) part of the GLEs exactly. If eigenvalues of the $\mathbf{A_1}$ matrix are negative, then there will not be any unstable mode in the particle trajectories. The analytical mapping of the first part also ensures that there will not be any numerical instability as solution is

obtained from an analytical expression without employing any finite difference scheme such as Euler scheme. It should be noted that the time step (Δt) should be smaller than the characteristic time scale (Yee and Wilson 2007) of the problem. Otherwise, the particle may travel a significant distance in a single time step.

4.3.2 Second step: Diagonal part of the nonlinear terms

The intermediate velocity vector obtained from the first step (U^*) is used as an input for the second step. In the second step, the diagonal part of the nonlinear terms is integrated using the following analytical expression,

$$U_i(t) \equiv U_i(t_0 + \Delta t) = \frac{U_i^{0*}}{1 - g_i U_i^{0*} \Delta t} \tag{4.22}$$

where $g_i = \frac{1}{2} R_{l(i)}^{-1} \frac{\partial R_{(i)l}}{\partial x_{(i)}}$

Please note that the indices in bracket (i) indicate that there is no summation over these indices. Eq. (4.22) defines the second mapping function $\Im_2$ to update the intermediate fluctuating component of the velocity from U^* to U^{**}.

It should be noted that Eq. (4.22) may cause unstable modes if $\mathrm{sgn}(g_i) = \mathrm{sgn}(U_i^{0*})$. That is, the velocity vector tends to increase in magnitude to potentially cause unphysical changes in the particle position (Yee and Wilson 2007). If $\mathrm{sgn}(g_i) = -\mathrm{sgn}(U_i^{0*})$, the particle's velocity contracts towards zero, enabling the particle velocity to relax towards the mean flow thereby circumventing the unphysical jumps in

particle trajectory. To avoid these unstable modes, the time step can be restricted such that the velocity update falls in a preset range. If the time step is chosen such that, $\Delta t = f \min\left(\frac{1}{g_1 U_1^{0*}}, \frac{1}{g_2 U_2^{0*}}, \frac{1}{g_3 U_3^{0*}}\right)$, where f is a constant having value between 0 and 1, then the amplification from U^* to U^{**} can be limited by the factor $\frac{1}{1-f}$.

4.3.3 Third step: Off-diagonal part of the nonlinear terms

The off-diagonal part of the nonlinear terms is evaluated in the third step. There is no analytical solution for this step; therefore, the simple Euler forward method (similar to QUIC-Plume's SLEs model (Williams et al. 2004)) is utilized for obtaining the fluctuating component of the velocity using U^{**} obtained from the second step.

4.4 Model evaluation

The performance of the 3D GLEs model has been evaluated against two idealized test cases and a wind-tunnel test case for a 7 x 11 cubical array of buildings described by Brown et al. (2001). The primary goal of these validations was to ensure that the performance of the 3D GLEs model was acceptable with the newly implemented GLEs when compared with the available analytical solutions and the test data.

4.4.1 Test case-I: Continuous release in uniform flow

In this test case, the performance of the 3D GLEs model has been tested against an existing analytical solution for an elevated continuous point source release in a uniform flow and the QUIC Plume model. The normalized concentration profiles from

the 3D GLEs model computations have been compared against the classical Gaussian solution (Seinfeld and Pandis 1998) for a steady state, horizontally homogenous, neutral atmospheric stability, constant wind speed and constant eddy diffusivity (see Singh et al. (2004) for details). For this test case, the turbulence model of 3D GLEs was simplified for the horizontally homogenous and constant eddy diffusivity conditions (Willemsen et al. 2007). The plume parameters required to run the test case have been described in detail by Singh et al. (2004).

To obtain near statistically stationary concentration estimates, 100,000 particles were continuously released from a spherical source (0.2 m diameter) at a height, H=70 m. The rate of emission was 100 particles per second with a time step of 1 second (dt = 1 second) for a duration of 1000 seconds.

The uniform wind speed was, U=2 ms^{-1}, with a friction velocity of, u_*=0.18 ms^{-1}. The concentration was averaged over 800 seconds with a starting time of 200 seconds after the beginning of the release. The physical domain was broken up into 20, 50 and 70 sampling boxes in x, y and z directions, respectively, over a domain size of 100 m x 100 m x 140 m. The source was specified to be at x=20 m, y=50 m and z=H=70 m.

Figures 4.6 and 4.7 show the lateral concentration profiles at two streamwise locations (x/H=0.393 and x/H=0.464) for the 3D GLEs model, QUIC Plume and the Gaussian analytical solution. The concentration has been normalized ($C^* = CUH^2/Q$) for comparison purposes. The lateral profiles from both the 3D GLEs model and QUIC Plume model are in good agreement with the analytical solution. Similar trends have been observed in the vertical profiles (Figures 4.8 and 4.9).

For this test case, the second term of the $\mathbf{A_1}$ matrix in the 3D GLEs model was always zero as the turbulence stresses were uniform throughout the domain. As $\mathbf{A_1}$ was a symmetric matrix for this test case, the eigenvalues for the $\mathbf{A_1}$ matrix were always negative, implying the absence of the unstable modes for the first step in the calculation process. The updates to the fluctuating component of velocity from the second and third step were zero due to the absence of the gradients in the turbulence stresses.

4.4.2 Test case-II: Continuous release in a power-law boundary layer flow

In this test case, the performance of the 3D GLEs model has been tested against an existing analytical solution for a continuous point source release in a boundary layer flow and the QUIC Plume model. The emission source was relatively close to the ground (H=4 m) to allow reflection of the emitted particles off of the ground.

The normalized concentration profiles from the 3D GLEs model computations have been compared against the classical non-Gaussian solution (Brown et al. 1993) and the QUIC Plume model for a steady state, horizontally homogenous, neutral atmospheric stability, power law wind profile and power law eddy diffusivity (see Singh et al. (2004) for details). The plume parameters required to run the test case have been described in detail by Singh et al. (2004).

To obtain near statistically stationary concentration estimates, 100,000 particles were continuously release from a point source. The rate of emission was 100 particles per second with a time step of 1 second (dt = 1 second) for a duration of 1000 seconds. The power law exponent for the velocity profile was 0.15 with a reference velocity, U=5.90

ms^{-1} at a reference height, H=4 m. The concentration was averaged over 800 seconds with a starting time of 200 seconds after the beginning of the release. The number of sampling boxes in the *x*, *y* and *z* directions were 18, 51 and 20, respectively, over a domain size of 100 m x 100 m x 20 m. The source was specified to be at *x*=20 m, *y*=50 m and *z*=*H*=4 m.

Figures 4.10 and 4.11 show the lateral concentration profiles at two streamwise locations (*x/H*=10.97 and *x/H*=19.31) for the 3D GLEs model, the QUIC Plume model and the non-Gaussian analytical solution. The concentration is normalized ($C^* = CUH^2/Q$) for comparison purposes. The lateral profiles from both the 3D GLEs model and the QUIC Plume model are in agreement with the analytical solution at both the streamwise locations. However, the QUIC Plume model slightly overpredicts the concentration at the centerline of the profiles. Similar trends have been observed in the vertical profiles (Figures 4.12 and 4.13). The QUIC Plume model overpredicts the peak concentration value (*z*/*H*=1) and the ground level concentration at both the streamwise locations.

In this test case, the turbulence stress gradients were present due to the presence of velocity gradients in the vertical direction. The mean flow was in the horizontal direction; therefore, the second term of the $\mathbf{A_1}$ matrix in the 3D GLEs model was zero, yielding a symmetric matrix. The eigenvalues for the symmetric $\mathbf{A_1}$ matrix were always negative, implying the absence of unstable modes in particle trajectory. The second and third steps may yield unstable modes for a few particles for this test case; however, no unstable mode was detected during the simulation.

4.4.3 Test case-III: Dispersion estimates in street canyons of a cubical array of buildings

For this test case, dispersion data from a wind-tunnel experiment for a 7 x 11 cubical array of buildings is compared against the 3D GLEs model and the QUIC Plume model for dispersion estimates within the street canyons. The wind-tunnel experiment has been explained in Chapter 2 and further details can be obtained from Brown et al. (2001). Please refer to Figures 2.3a and 2.3b (Chapter 2) for details on the placement of the cubic buildings within the domain.

The wind-tunnel experiment was conducted in a USEPA Meteorological Wind Tunnel. For dispersion estimates, a high purity ethane (C_2H_6; CP grade; minimum purity 99.5 mole percent) tracer was used in the experiment (Lawson et al. 2000), which was slightly heavier than the air (molecular weight 30). This tracer may be regarded as neutrally buoyant owing to the high turbulence level and the release rate of the tracer (Lawson et al. 2000). A perforated plastic sphere of 10 mm diameter was used for continuously releasing the tracer at ground level behind the first centerline building of the 7 x 11 cubical building array.

As explained in Chapter 2, various shortcomings in the street canyon parameterizations of the diagnostic wind model, QUIC-URB, were addressed by using the so-called “modified” Röckle model. The mean wind field generated by the modified Röckle model has been used for driving both the 3D GLEs and the QUIC Plume model for this test case. The coordinate system, grid resolution, inflow profile and the boundary conditions used for driving the wind model has been explained in Chapter 2.

A local mixing length turbulence model was used to drive the 3D GLEs dispersion model. The local mixing length model is based on the classical Reynolds averaged Navier-Stokes equation modeling given by (e.g., Wilcox (2006)):

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{\partial}{\partial x_j}\left(2\nu S_{ij} - \overline{u_i u_j}\right) \tag{4.23}$$

where,

U= mean velocity (ms^{-1}),

P= mean pressure (Nm^{-2}),

ρ = density (kgm^{-3}),

ν = viscosity (m^2s^{-1}),

$\overline{u_i u_j}$ = Reynolds stress,

$\overline{S_{ij}} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)$= strain rate tensor.

Eq. (4.23) represents the classic "closure" problem. There are 10 unknowns (3 velocity components, pressure and 6 Reynolds stresses) in this equation; however, there are only 4 equations (including continuity equation). To obtain closure of Eq. (4.23), the Reynolds stresses ($\overline{u_i u_j}$) must be modeled. Employing the turbulent viscosity hypothesis by Boussinesq (1877), we obtain (Pope 2000),

$$\overline{u_i u_j} = \frac{2}{3}k\delta_{ij} - 2\nu_T \overline{S}_{ij}$$

where,

k = kinetic energy (m^2s^{-2}),

δ_{ij} = Kronecker delta,

ν_T = turbulent viscosity (m^2s^{-1}).

For high Reynolds number flows, the molecular viscosity, ν, is neglected and the turbulent viscosity can be specified using Prandtl's mixing length hypothesis as (Pope 2000),

$$\nu_T = \ell_m^2 \left(2\overline{S_{ij}}\,\overline{S_{ij}}\right)^{1/2}$$

and,

$$k = \left(\frac{\nu_T}{c\ell_m}\right)^2$$

where, ℓ_m is the mixing length (m) and c is a constant ≈ 0.55 (Pope 2000). In classical wall bounded flow problems, the mixing length, ℓ_m, is related to the distance to the wall. For building flow, ℓ_m is obtained from the distance to the closest surface, i.e., (Williams et al. 2004),

$$\ell_m = c_l L_{\min}.$$

Here, c_l is an unknown model constant and $L_{\min}$ is the minimum distance to a surface. The value of c_l is specified to be 0.4 (Pope 2000). The minimum distance to the surface, $L_{\min}$, is obtained by solving the point-plane distance equations (Shirley and Marschner 2009).

The above formulations suggest that the normal stresses (τ_{11}, τ_{22} and τ_{33}) are isotropic, therefore $\tau_{11}=\tau_{22}=\tau_{33}$. However, the experimental data (Castelli et al. 2001) suggest that for a normal boundary layer flow (i.e., far away from buildings), τ_{11} is greater than τ_{22} and τ_{33}. To accomplish $\tau_{11}>\tau_{22}>\tau_{33}$, constants $(2.5)^2$, $(1.6)^2$ and $(1.3)^2$ are multiplied with τ_{11}, τ_{22} and τ_{33}, respectively, as follows:

$$\tau_{11} = (2.5)^2 \tau_{11}$$

$$\tau_{22} = (1.6)^2 \tau_{22}$$

$$\tau_{33} = (1.3)^2 \tau_{33}$$

These constant values are obtained from the test case II for the boundary layer flow. Other researchers have reported these constants to be close to the above stated values; for example, Castelli et al. (2001) reported these constants to be $(2.5)^2$, $(1.8)^2$ and $(1.3)^2$ for τ_{11}, τ_{22} and τ_{33}, respectively.

Since the turbulence model is a local mixing model, it relies heavily on the magnitude of the local velocity gradients for turbulence estimates. This is problematic in the core of a street canyon where velocity gradients can be small (see Figure 2.7, Chapter 2) and the model predicts negligible turbulence. With this model, particles tend to closely

follow the mean wind near the source until they are ejected out of the first street canyon without much mixing within the canyon. To enhance the mixing, a nonlocal turbulence mixing coefficient C_{sc}~0.3 was added to the diagonal elements of the stress tensor (τ_{11}, τ_{22} and τ_{33}) to account for processes such as turbulent advection of turbulence into the central portion of the street canyon. The result of this modification was to augment the magnitude of the turbulence within the street canyons (and match the experimental data within the canyons better) while maintaining the influence of the velocity gradients in the stress tensor. The nonlocal mixing coefficient, C_{sc}, may not be valid for general applicability as this may yield unreasonable results unless verified extensively with more experimental data. Figure 4.14 compares the 3D GLEs model, the QUIC Plume model and the wind-tunnel data for the normalized $\sigma_U (= \sqrt{\tau_{11}})$ (normalized by reference velocity, U=2.82 ms^{-1}, i.e., $\sigma_U^* = \sigma_U / U$) profiles at x/H=1.15, x/H=1.55 and x/H=1.95 at the centerline (y/H =0) within the first street canyon. At x/H = 1.15, both the QUIC Plume and the 3D GLEs models overestimate the σ_U^* within the canyon (~z/H<1). The 3D GLEs model overestimates σ_U^* above the street canyon also (~z/H>1) but stays much closer to the test data (Root Mean Squared (RMS) error =0.16) as compared with the QUIC Plume model. The QUIC Plume model overestimates σ_U^* at ~z/H<1.7 (possibly due to QUIC-Plume's nonlocal mixing model) and then underestimates σ_U^* for ~z/H>1.7 (RMS error =0.23). At x/H=1.55, the σ_U^* for the 3D GLEs model stay closer to the wind-tunnel data within the canyon; however, it exhibits a sharp increase in σ_U^* at ~z/H=0.95 (RMS error =0.43). The QUIC Plume model slightly overestimates σ_U^* within the

canyon and exhibits an increase in σ_U^* at ~*z/H*=1.05 (RMS error = 0.16). This sharp increase in σ_U^* for both the models may be related to the presence of the vortex center close to *x/H*=1.55. At *x/H*=1.95, the 3D GLEs model (RMS error = 0.06) predicted the σ_U^* profile stay closer to the wind-tunnel data at almost all the vertical heights as compared with the QUIC Plume model (RMS error = 0.2).

Figure 4.15 compares the 3D GLEs model, the QUIC Plume model and the wind-tunnel data for the σ_U^* profiles at *x/H*=2.15, *x/H*=2.35, *x/H*=2.55 and *x/H*=2.75 at the centerline (*y/H*=0) over the rooftop of the second centerline building. All the profiles show that the 3D GLEs model is in better agreement with the wind-tunnel data as compared with the QUIC Plume model. The RMS errors for the 3D GLEs model are 0.051, 0.052, 0.053 and 0.056 for *x/H*=2.15, 2.35, 2.55 and 2.75, respectively. For the QUIC Plume model, the RMS errors are 0.26, 0.28, 0.28 and 0.26, respectively.

Figure 4.16 compares the 3D GLEs model, the QUIC Plume model and the wind-tunnel data for the σ_U^* profiles at *x/H*=1.15, *x/H*=1.55 and *x/H*=1.95 at z/*H*=0.2 within the first street canyon. At *x/H*=1.15, the 3D GLEs model and the QUIC Plume model slightly overestimate the σ_U^*. At the edges of the street canyon (*y/H*~=$\pm$0.5), the 3D GLEs model shows sharp increase in the σ_U^* due to the presence of strong velocity gradients (RMS error = 0.83). The QUIC Plume model stays closer to the wind-tunnel data and shows a relatively mild increase in σ_U^* at the edges of the canyon (RMS error = 0.3). Similar trends in σ_U^* profiles are observed at *x/H*=1.55 with the 3D GLEs model (RMS error = 1.5) showing sharp increase in σ_U^* at the edges of the canyon. The QUIC Plume model (RMS error = 0.3) stays closer to the wind-tunnel data at this streamwise

distance. At *x/H*=1.95, the 3D GLEs model stays closer to wind-tunnel data for the most part within the canyon and does not overestimate σ_U^* close to the edges of the canyon (RMS error = 0.09). The QUIC Plume model overestimates σ_U^* at the edges of the canyon and slightly overestimates σ_U^* within the canyon (RMS error = 0.3). Both the models underestimate σ_U^* within the street channel.

A total of 100,000 particles were released continuously for 1000 seconds (i.e., *Q* = 100 particles/sec) from a spherical source with 10 mm diameter to obtain near statistically stationary concentration estimates for both the 3D GLEs model and the QUIC Plume model. A release of double the number of particle (200,000) did not show appreciable change in concentration field from a release of 100,000 particles in both the QUIC Plume and the 3D GLEs models. The concentration was averaged over 900 seconds with a starting time of 100 seconds after beginning the release. The number of sampling boxes in x, y and z directions was 60, 55 and 25, respectively, over a domain size of 153 m x 110 m x 30 m. The source was placed behind the first centerline building of the 7 x 11 cubic building array at *x/H* = 0.067, *y/H* = 0 and *z/H* = 0.067 to be in agreement with the wind-tunnel experiments.

Figures 4.17 and 4.18 show the horizontal concentration contours (normalized on log scale) for both the 3D GLEs model and the QUIC Plume model respectively at *z/H*=0.3. The contour plots show that the 3D GLEs model exhibit more lateral dispersion compared to the QUIC-Plume model at this height with almost double the downstream plume width. The wake of the last building along the center row (downstream) shows lack of mixing (very low concentration) in the QUIC Plume model as compared with the 3D GLEs model. At planes located further above the ground (*z/H*=0.54 and 0.78 in

Figures 4.19, 4.20, 4.21 and 4.22) similar trends are observed. At *z/H*=1.02 (Figures 4.23 and 4.24), the 3D GLEs model shows slightly higher concentrations at the roof top of the first centerline line building (upstream of the source) as compared with the QUIC Plume model. Similar trends are observed at *z/H*=1.26 (Figures 4.25 and 4.26). QUIC Plume shows higher concentrations at the upstream walls of all the centerline buildings at *z/H*=0.78 (see Figure 4.27). The vertical concentration profiles at the upstream walls of the four (downstream) buildings (*x/H*=1.9, 3.9, 5.9 and 7.9) indicate that the QUIC Plume model shows higher concentrations at ~*z/H* = 1 as compared with the 3D GLEs model (Figure 4.27).

Figures 4.28 and 4.29 show centerline vertical slices of plume concentration (normalized and log scaled) at *y/H*=0 for the 3D GLEs model and QUIC Plume, respectively. The 3D GLEs model shows more vertical dispersion compared to the QUIC Plume model along the centerline. As also observed in the horizontal contours, QUIC Plume does not show enough mixing in the wake of the last centerline (downstream) building as compared with the 3D GLEs model. The 3D GLEs model tends to predict higher concentrations near the center of the downstream street canyons (second, third and fourth street canyons); however, the QUIC Plume model tends to predict higher concentrations close to the upstream of the roof top of the downstream street canyons (see Figure 4.27). The 3D GLEs model shows higher concentration at the rooftop of the first centerline (upstream of the source) building as compared with the QUIC Plume model. Similar trends are observed at the midpoint between the centerline and an edge of the street canyon at *y/H*=0.2 as shown in Figures 4.30 and 4.31. Figure 4.32 and 4.33 show vertical slices of the concentration contours at the center of the street channel

(y/H=1) for the 3D GLEs model and QUIC Plume, respectively. The 3D GLEs model shows enhanced mixing and more vertical dispersion of the plume as compared with the QUIC Plume model in the street channel. It is also evident from the horizontal contour slices (Figures 4.18 and 4.20) that the QUIC Plume model does not exhibit much dispersion laterally in the adjoining street channels and restricts the plume in the lateral direction to the centerline buildings. Figure 4.32 show that the 3D GLEs model mixes the plume well within the street channel. Figure 4.33 shows lack of mixing in the street channel in the QUIC Plume model.

The dispersion estimates from the 3D GLEs model are compared with the wind-tunnel data at the center for the first three centerline street canyons (see Figure 2.3). The dispersion estimates from the QUIC Plume model with nonlocal mixing (Williams et al. 2004) are also included for the comparison.

Figure 4.34 shows vertical concentration profile comparison of wind-tunnel data with the 3D GLEs model and the QUIC Plume dispersion model at the center of the first centerline street canyon (x/H =1.5). The concentration profiles from the QUIC Plume model and the 3D GLEs model are normalized using the following equation:

$$C^* = \frac{CUH^2}{Q}$$

In the above equation, C^* is the normalized concentration with U (=2.82 ms^{-1}) as the reference velocity at the building height of H =10 m. The concentration is represented by C and Q is the release rate (100 particles/sec).

In Figure 4.34, the normalized concentration of the test data increases from a minimum at the ground level to a maximum near the building height (z/H = 0.867) and then decreases with increase in the height. The 3D GLEs model slightly underpredicts the concentration at the ground level but shows an increase in the concentration with a maximum concentration at ~z/H=0.78. The 3D GLEs model shows a dip in the concentration profile at ~z/H=0.9 and then an increase in the concentration at ~z/H=1.02. The concentration then decreases with increase in height. The QUIC Plume model overestimates the concentration at the ground level and shows two maxima at ~z/H=0.3 and ~z/H=1.14. A minimum concentration was predicted at ~z/H=0.9 by the QUIC Plume model. The concentration decreases with height after the second maximum at ~z/H=1.14 in the QUIC Plume model. Both the 3D GLEs and the QUIC Plume models show a concentration dip at ~z/H=0.9 with two maxima. The presence of the vortex center close to ~z/H=0.9 may have resulted in this concentration dip. Overall, the 3D GLEs model performs better than the QUIC Plume model for this concentration profile with an RMS error 1.47 as compared to the RMS error of 1.74 in the case of the QUIC Plume model.

Figure 4.35 shows the concentration profile along the centerline of the second street canyon (i.e., $y/H=0$) at x/H=3.5. Both the 3D GLEs and the QUIC Plume models underestimate the concentration at the ground level; however, the 3D GLEs model predicts the concentrations to be closer to the wind-tunnel concentrations. The QUIC Plume model underpredicts the concentration up to a height of ~z/H=1.02 and then shows a sharp increase at ~z/H=1.14. The concentration profile of the 3D GLEs model stays closer to the experimental data below ~z/H=0.5 compared with QUIC Plume. The 3D GLEs model shows a sharp increase in the concentration profile at ~z/H=0.78 and then a

dip which bring the concentration profile closer to the test data. A sharp increase in the concentration profile of the 3D GLEs model is again seen close to ~z/H=1.02. Overall, the 3D GLEs model performed better than the QUIC Plume model for this concentration profile with an RMS error of 0.27 as compared with the RMS error of 0.32 in the case of the QUIC Plume model.

Figure 4.36 shows a concentration profile at the center of the third street canyon at x/H=5.5. Both the 3D GLEs and the QUIC Plume model underestimate the concentration at the ground level. The QUIC Plume model underestimates the concentration below ~z/H=1.02 and exhibit a sharp increase in the concentration at ~z/H=1.14. The concentration profile of the 3D GLEs model exhibits two sharp increases in the concentration at ~z/H=0.78 and ~z/H=1.02. Overall, the 3D GLEs model performed comparable to the QUIC Plume model for this concentration profile with an RMS error of 0.173 as compared with an RMS error of 0.186 in the case of the QUIC Plume model.

Figure 4.37 shows a paired scatter plot of the above vertical concentration profiles at x/H=1.5, 3.5 and 5.5. The wind-tunnel concentration data are represented on the abscissa and the predicted concentration by the QUIC Plume model (open circles - o) and 3D GLEs model (filled circles - ●) is represented on the ordinate of the scatter plot.

The scatter plot shows that both the models underestimate the concentration values for the most part. Most of the predicted concentrations by both the models fall within a factor of 2. The QUIC Plume model shows more scatter in both underestimation and overestimation of the concentrations as compared with the 3D GLEs model.

Figure 4.38 shows the normalized concentration profile of the test data compared against the 3D GLEs model and the QUIC Plume model within the street channel at y/H=1 and x/H=1.5. Both the models underestimate the concentration, predicting almost zero concentrations at all heights, signifying the lack of lateral dispersion into the channel by both the models at x/H=1.5. At x/H=2.5 and y/H=1 (Figure 4.39), both the models still underestimate the concentrations; however, the 3D GLEs model shows slightly more lateral dispersion into the street channel as compared with the QUIC Plume model.

Figure 4.40 shows the normalized concentration comparison at x/H=3.5 and y/H=1. At this streamwise distance, the 3D GLEs model shows significant lateral dispersion in comparison with the QUIC Plume model. The 3D GLEs model is in good agreement with the test data above $\sim z/H$>1.02 for the vertical dispersion; however, it underestimates the concentration below $\sim z/H$<1. The QUIC Plume model underestimates the concentrations at all heights.

Figure 4.41 shows the normalized concentration profiles comparison at x/H=5.5 and y/H=1. The 3D GLEs model shows significant lateral and vertical dispersion as compared with the QUIC Plume model. The QUIC Plume model underestimates the concentrations at all heights. The 3D GLEs model overestimates the concentrations above $\sim z/H$ >1.35 and underestimates it below $\sim z/H$<1.35.

4.5 Discussion and summary

Due to the presence of a large number of terms in the GLEs, a simplified version of the Langevin equations (SLEs) is employed in most of the main stream dispersion models such as the QUIC Plume dispersion model (Williams et al. 2004) discussed in this

chapter. The presence of unstable modes due to the stiffness of the GLEs has also been a problem for the direct implementation of the GLEs into a Lagrangian dispersion model. Although unstable modes also exist in the SLEs, the SLEs are considered slightly stable due to the drastic reduction in the number of terms in the SLEs. This chapter discussed the implementation of the GLEs with a fractional step method (FSM) which partially solves the stiffness problem exhibited by the GLEs (Yee and Wilson 2007). The FSM decomposes the GLEs into three steps and the first two steps are solved analytically, circumventing the unstable modes. The third step is performed using the Euler forward method, which does not guarantee the absence of the unstable modes as no analytical mapping is available for the third step.

The analytical mapping functions used in the first two steps of the FSM helps in revealing the existence of potential unstable modes in the particle trajectory before the advection process even begins in the dispersion model. For example, in the first step of the FSM, the eigenvalues of the $\mathbf{A}_1$ matrix are computed. The presence of positive eigenvalues indicates the existence of the unstable modes in the particle trajectory. In the dispersion model, the eigenvalues of the $\mathbf{A}_1$ matrix are computed before the particle advection process starts. Therefore, if the eigenvalues are positive, the dispersion modeler can further investigate the potential weaknesses in the turbulence model and the background wind field. One the other hand, in the Euler forward method (unlike analytical integration functions), no such information is available to the dispersion modeler beforehand to identify the potential unstable modes. Therefore, it is recommended to use the analytical mapping functions for solving the SLEs also rather than explicitly using the Euler forward method to integrate the SLEs (in conjugation with

coordinate rotation in the mean wind direction for accommodating horizontally homogenous assumptions).

The Gaussian and non-Gaussian test case results demonstrated that the 3D GLEs implementation with the FSM works well for these simple test cases. For the Gaussian test case, the second and third steps of the FSM yield zero due to the absence of the velocity as well as turbulence stress gradients. Also, the first step of the FSM yielded a symmetric $\mathbf{A_1}$ matrix due to the absence of the drift terms (as no velocity and turbulence stress gradients exist). Therefore, there was no possibility of the existence of the unstable modes in the Gaussian test case. This information is available to the dispersion modeler before the actual advection of the particle starts in the dispersion model. In the non-Gaussian test case, the second and third step did not yield zero values due to the presence of the velocity and turbulence stress gradients. The $\mathbf{A_1}$ matrix for the non-Gaussian test case was symmetric, as the second term of the $\mathbf{A_1}$ matrix was zero (see Eqs. (4.15)) for this case. The second and third steps did not yield unstable modes for this test case possibly due to the simplicity of this case.

Another simple test case of a spatially varying shear layer was used to investigate possible unstable modes associated with the positive eigenvalues of the $\mathbf{A_1}$ matrix. The turbulence model used for this test case yielded favorable results when compared with the experimental data. This test case demonstrated that whenever the model computed product of C_0 and dissipation (ε) is less than the gradient of the turbulence stress, the unstable modes may exist during the advection process. These unstable modes may yield rogue trajectories where a particle may travel significant distance within a small time step. To prevent these rogue trajectories, the dispersion modelers normally limit the size

of the fluctuating component of the velocity by some multiple of the standard deviation of the wind in the respective directions (Yee and Wilson 2007). The QUIC Plume model uses 4.0 as a limiting factor and the 3D GLEs model uses 2.5 as the limiting factor to limit the unphysical rogue trajectories. The last test case of a more realistic urban setup (7 x 11 cubical array of buildings) yielded positive eigenvalues for the $\mathbf{A}_1$ matrix at some regions close to the cubical buildings. The values of the positive eigenvalues were however close to zero for most part. Part of the reason for the existence of the positive eigenvalues may be the background wind field and a simple turbulence model. Personal communication with Dr. Yee confirmed the existence of positive eigenvalues even for the wind fields obtained from CFD (Computational Fluid Dynamics) models, which are considered more accurate than the diagnostic wind models like QUIC-URB.

All the test cases considered for validation of the 3D GLEs model yielded good results in comparison with the test data as well as the SLEs-based model QUIC Plume. The improved wind model used to drive the 7 x 11 test case in both the 3D GLEs and QUIC Plume model was in good agreement with the test data (see Chapter 2). The turbulence model used to drive the 3D GLEs was a local mixing length model with an addition of a constant 0.3 to enhance the turbulence within the street canyons. The constant 0.3 was added to match the turbulence level within the street canyons with the test data. Although this addition raised the turbulence level at the edges of the street canyons, the turbulence levels within the street canyons matched the test data well. The vertical turbulence profiles also showed a good match with the test data. The QUIC Plume model uses its nonlocal mixing length model (Williams et al. 2004) for driving the 7 x 11 test case. The concentration contours showed that the 3D GLEs model exhibits

more lateral and vertical dispersion as compared with the QUIC Plume model. The plume in the street channels (adjoining centerline buildings) was well-mixed in the case of the 3D GLEs model, indicating enhanced lateral dispersion exhibited by the 3D GLEs model in comparison with the QUIC Plume model. In emergency scenarios, underestimating the concentrations would be worse than overestimating as people may not be evacuated from the "unsafe" places which the model declares "safe" due to the underestimation of the concentrations. The differences in the lateral and vertical dispersion of the 3D GLEs model and the QUIC Plume model may be partially attributed to the increased number of terms present in the 3D GLEs model. The different turbulence models used to drive the models may also lead to different lateral and vertical dispersion. The concentration profiles showed that the 3D GLEs model performs relatively better than the QUIC Plume model for the most part. The paired scatter plot also indicates that the 3D GLEs model was in good agreement with the test data.

The future work will include more validation studies of the 3D GLEs model and implementation of the 3D GLEs model on a GPU-based architecture to realize real-time run times. As the FSM is capable of providing insights into the unstable modes before the actual advection of the particles, these insights may be utilized to further improve the diagnostic wind model and turbulence models to better suit the needs of a Lagrangian dispersion model based on Langevin equations.

4.6 References

Bell, J. H., and R. D. Mehta, 1990: Development of a two-stream mixing layer from tripped and untripped boundary-layers. *Aiaa J.*, **28,** 2034-2042.

Boussinesq, J., 1877: *Essai sur la théorie des eaux courantes.* Imprimerie Nationale.

Brown, M., R. Lawson, D. DeCroix, and R. Lee, 2001: Comparison of centerline velocity measurements obtained around 2D and 3D building arrays in a wind tunnel. *Int. Soc. Environ. Hydraulics*, Tempe, AZ, 6.

Brown, M. J., S. P. Arya, and W. H. Snyder, 1993: Vertical Dispersion from Surface and Elevated Releases: An Investigation of a Non-Gaussian Plume Model. *Journal of Applied Meteorology*, **32,** 490-505.

Castelli, S., E. Ferrero, and D. Anfossi, 2001: Turbulence Closures In Neutral Boundary Layer Over Complex Terrain. *Boundary-Layer Meteorology*, **100,** 405-419.

Lawson, R., S. Perry, and R. Thompson, 2000: Measurement of velocity and concentration fields in arrays of 2-dimensional and 3-dimensional buildings in a simulated neutrally-buoyant atmospheric boundary layer. U.S. Environmental Protection Agency FMF Internal Data Rep., 55 pp.

Pope, S. B., 2000: *Turbulent Flows.* Cambridge University Press, 771 pp.

Rodean, H. C., 1996: *Stochastic Lagrangian models of turbulent diffusion.* The American Meteorological Society, 82 pp.

Seinfeld, J. H., and S. N. Pandis, 1998: *Atmospheric chemistry and physics: from air pollution to climate change.* John Wiley and Sons, Inc., 1326 pp.

Shirley, P., and S. Marschner, 2009: *Fundamentals of computer graphics.* A. K. Peters, 392 pp.

Singh, B., M. D. Williams, E. R. Pardyjak, and M. J. Brown, 2004: Testing of an urban Lagrangian dispersion model using Gaussian and non-Gaussian solutions. *4th AMS Symp. Urban Env.*, Norfolk, VA.

Thomson, D. J., 1987: Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *Journal of Fluid Mechanics*, **180,** 529-556.

Vreman, B., 1994: Realizability conditions for the turbulent stress tensor in large-eddy simulation. *Journal of Fluid Mechanics*, **278,** 351.

Wilcox, D. C., 2006: *Turbulence modeling for CFD.* DCW Industries.

Willemsen, P., A. Norgren, B. Singh, and E. R. Pardyjak, 2007: Development of a new methodology for improving urban fast response Lagrangian dispersion simulation via parallelism on the graphics processing unit. *In Proceedings of the 11th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes*, Queen's College, University of Cambridge, United Kingdom.

Williams, M. D., M. J. Brown, D. Boswell, B. Singh, and E. M. Pardyjak, 2004: Testing of the QUIC-plume model with wind-tunnel measurements for a high-rise building. *5th AMS Urban Env. Conf.*, Vancouver, BC.

Wilson, J. D., and B. L. Sawford, 1996: Review of Lagrangian stochastic models for trajectories in the turbulent atmosphere. *Boundary-Layer Meteorology*, **78,** 191-210.

Yee, E., and J. Wilson, 2007: Instability in Lagrangian stochastic trajectory models, and a method for its cure. *Boundary-Layer Meteorology*, **122,** 243-261.

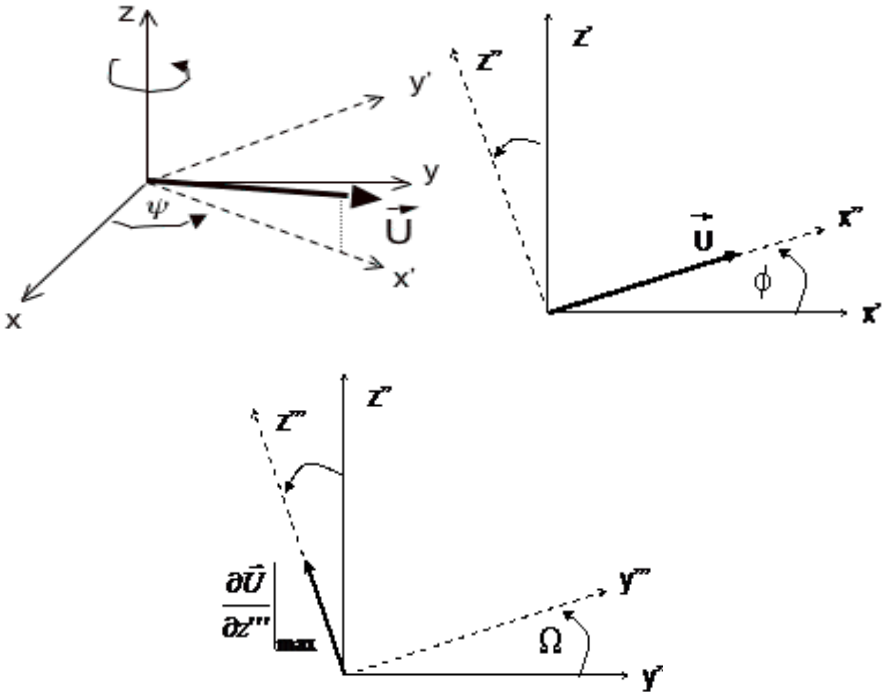


Figure 4.1: Schematic outlining the procedure to rotate the coordinate system in the direction of the mean wind (adapted from Williams et al. (2004)).

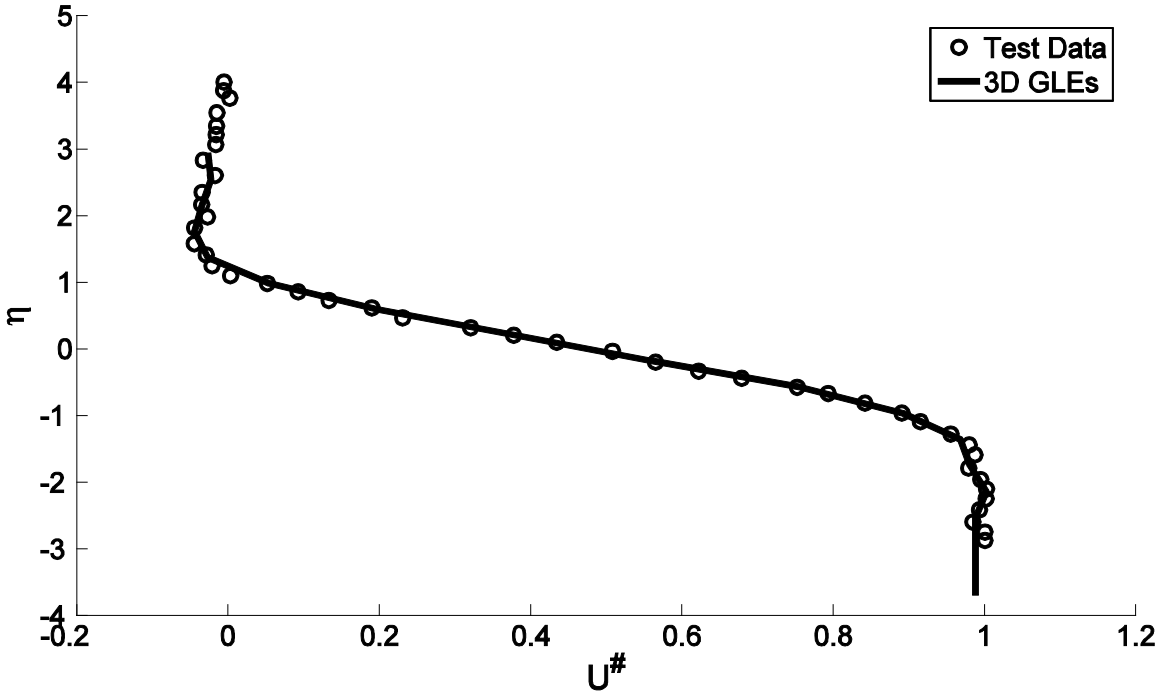


Figure 4.2: Velocity profile comparison between the test data and the velocity profile used as an input for the 3D GLEs model.

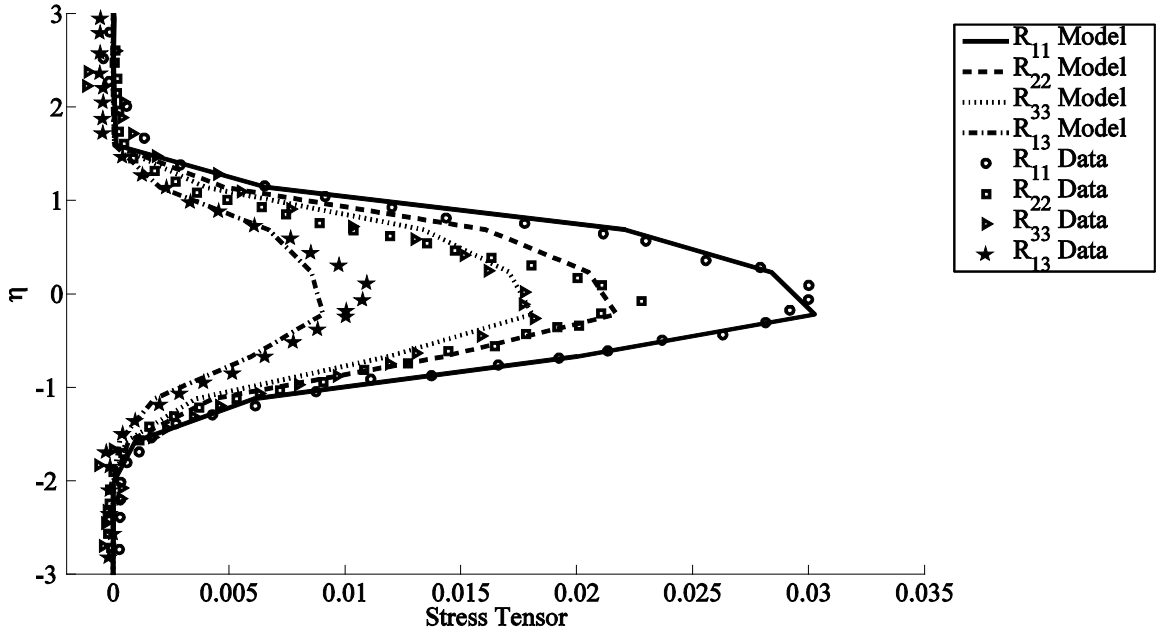


Figure 4.3: Comparison between the turbulence stresses from the 3D GLEs mixing length model with the test data.

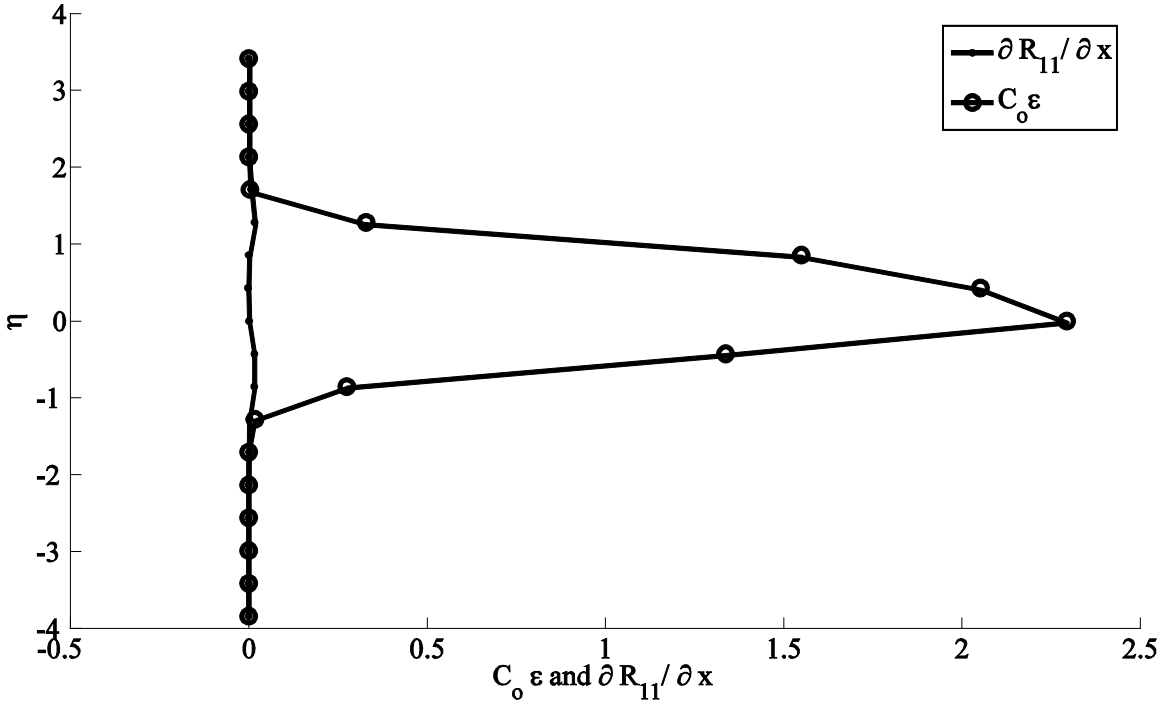


Figure 4.4: Comparison between the relative magnitude of turbulence stress gradient in streamwise direction and the magnitude of $C_0\varepsilon$.

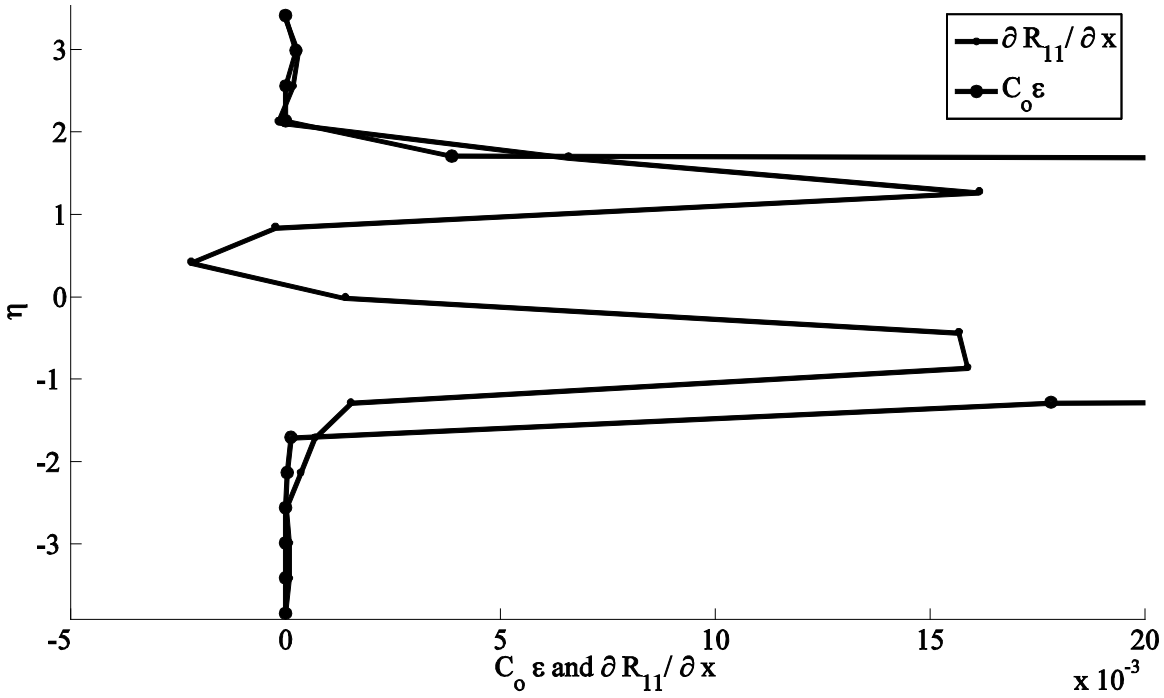


Figure 4.5: Comparison between the relative magnitude of turbulence stress gradient in streamwise direction and the magnitude of $C_0\varepsilon$ (zoomed in).

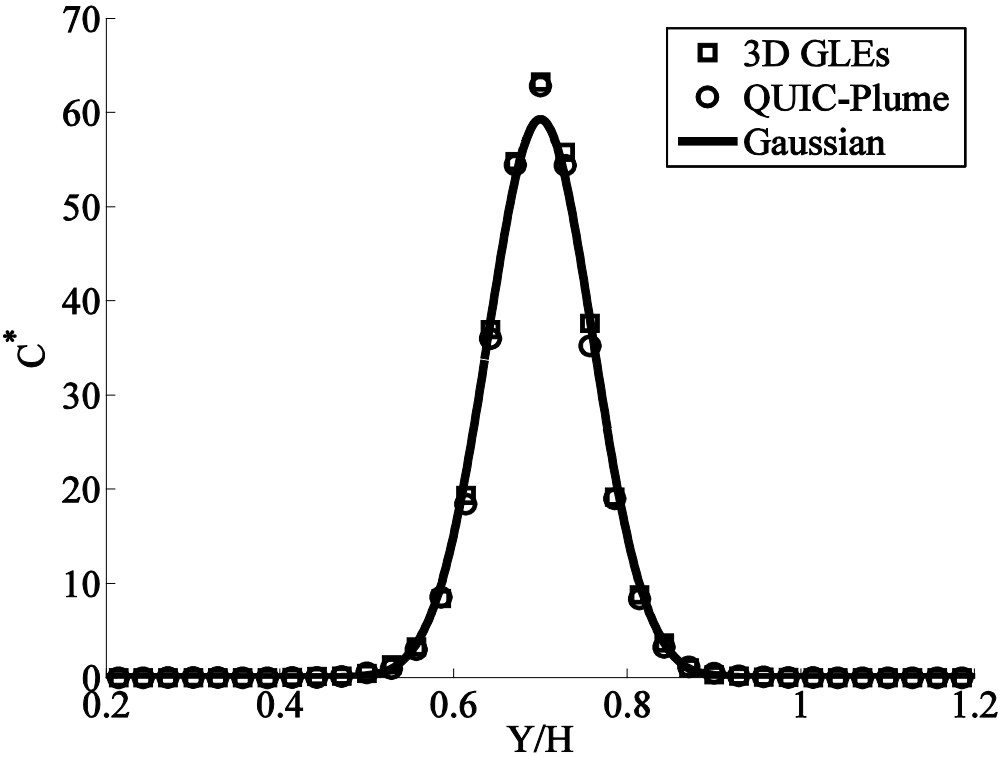


Figure 4.6: Lateral normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the Gaussian Solution at *x/H*=0.393.

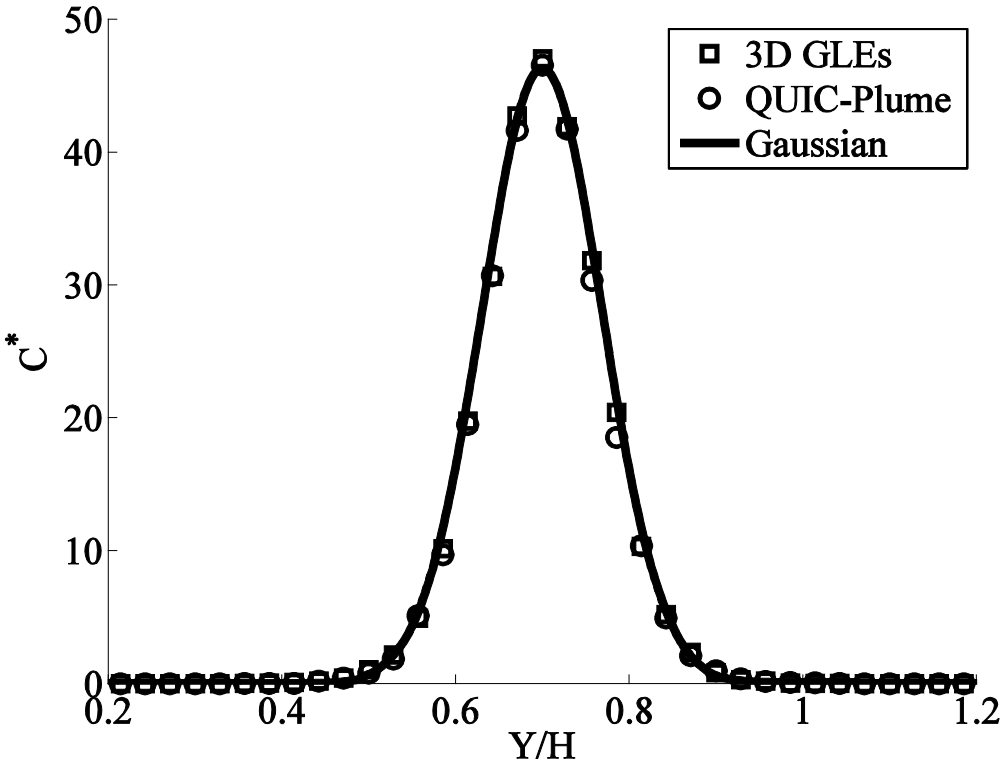


Figure 4.7: Lateral normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the Gaussian Solution at *x/H*=0.464.

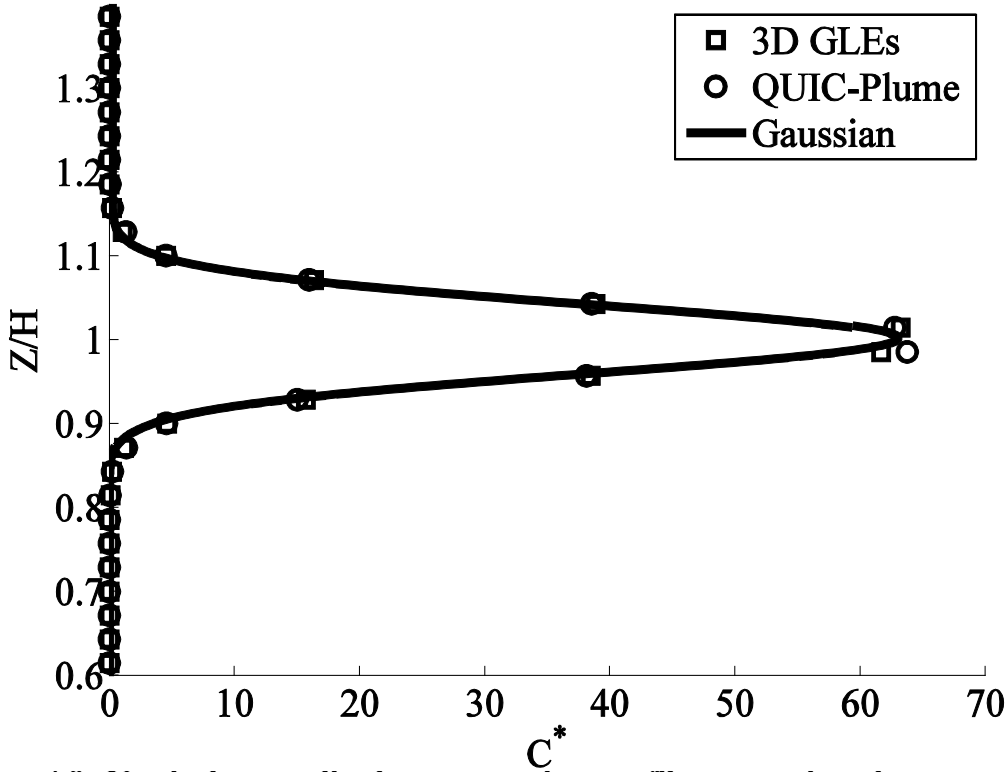


Figure 4.8: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the Gaussian Solution at *x/H*=0.393.

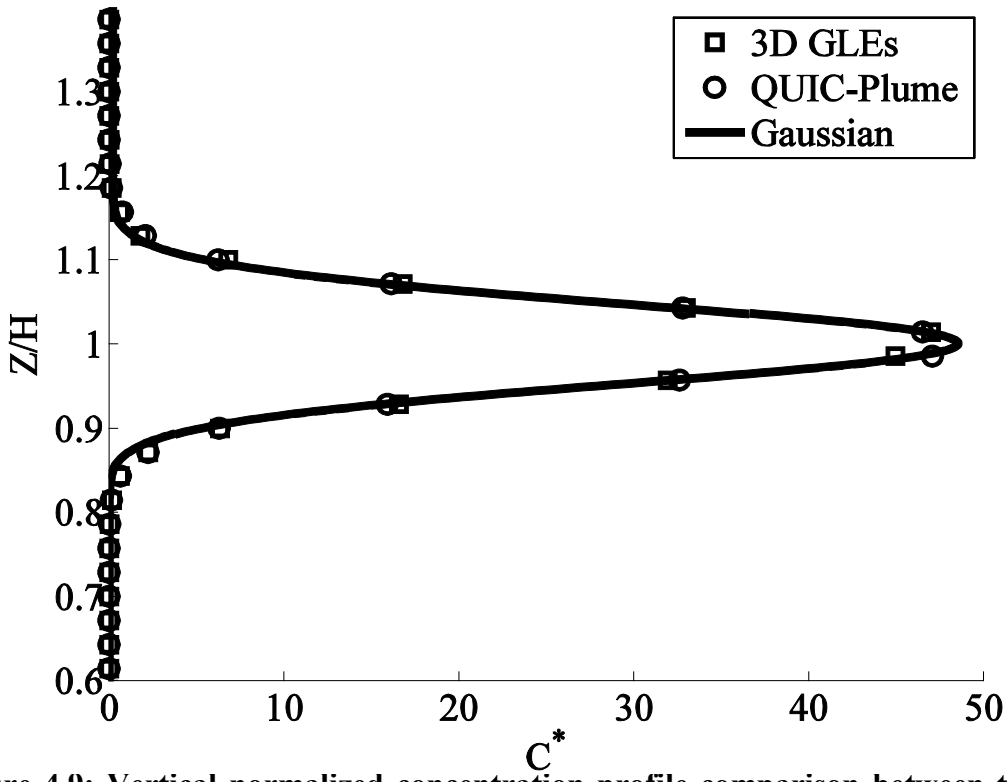


Figure 4.9: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the Gaussian Solution at *x/H*=0.464.

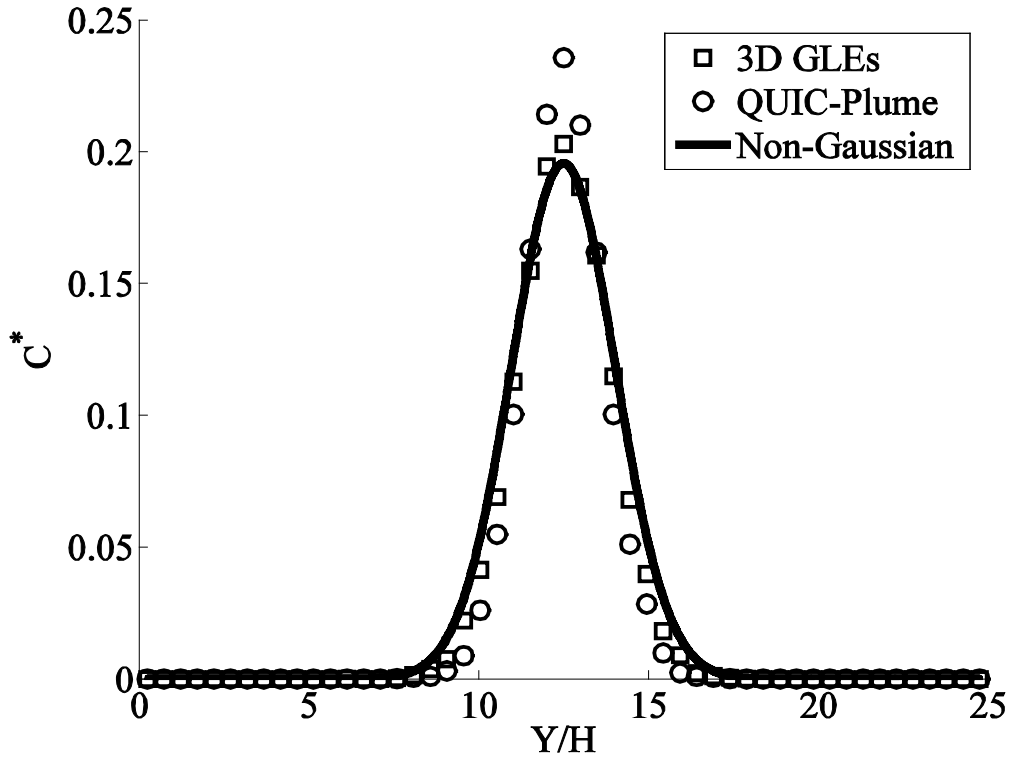


Figure 4.10: Lateral normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the non-Gaussian Solution at *x/H*=10.97.

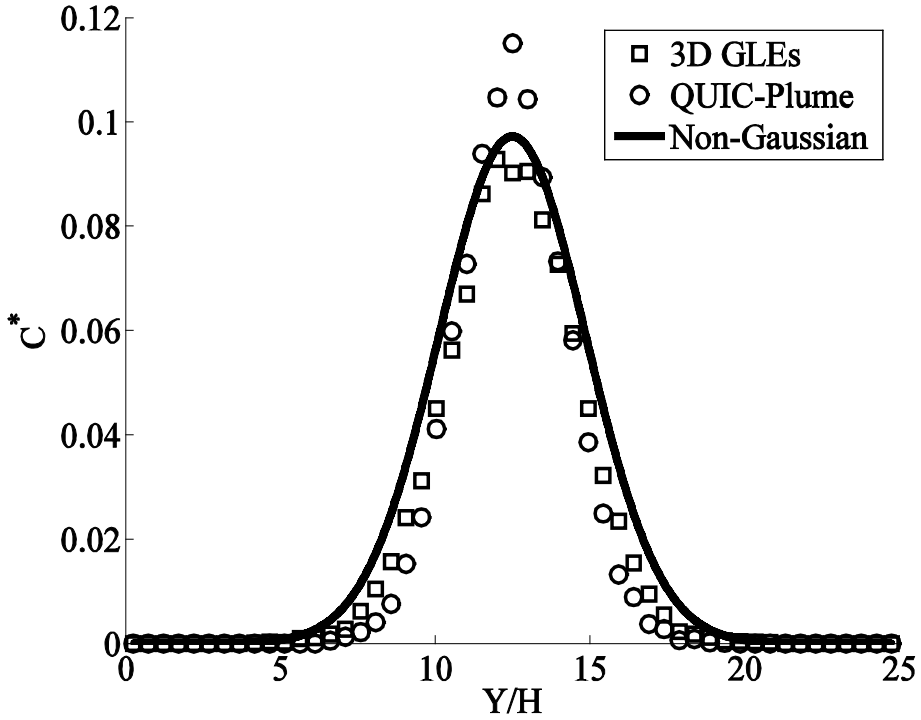


Figure 4.11: Lateral normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the non-Gaussian Solution at *x/H*=19.31.

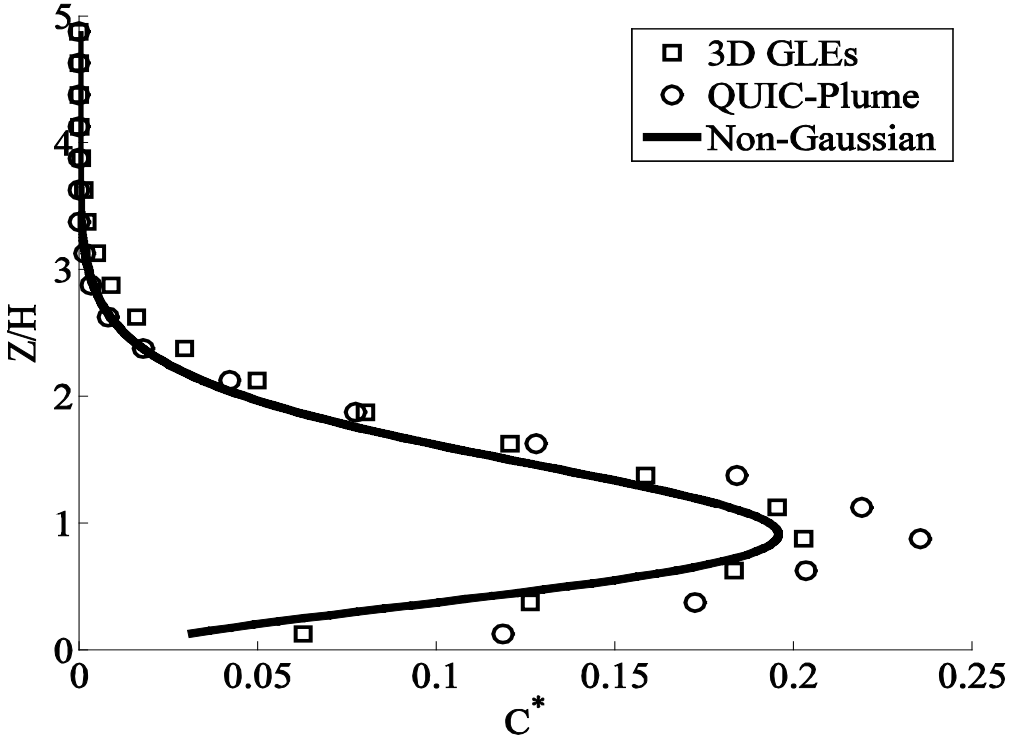


Figure 4.12: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the non-Gaussian Solution at *x/H*=10.97.

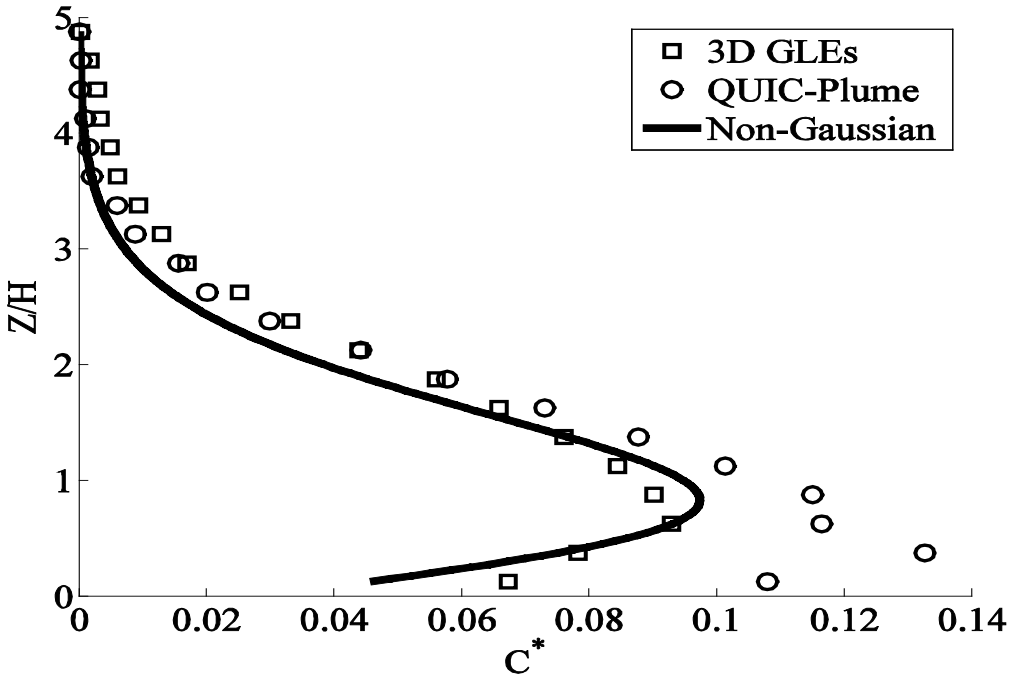


Figure 4.13: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the non-Gaussian Solution at *x/H*=19.31.

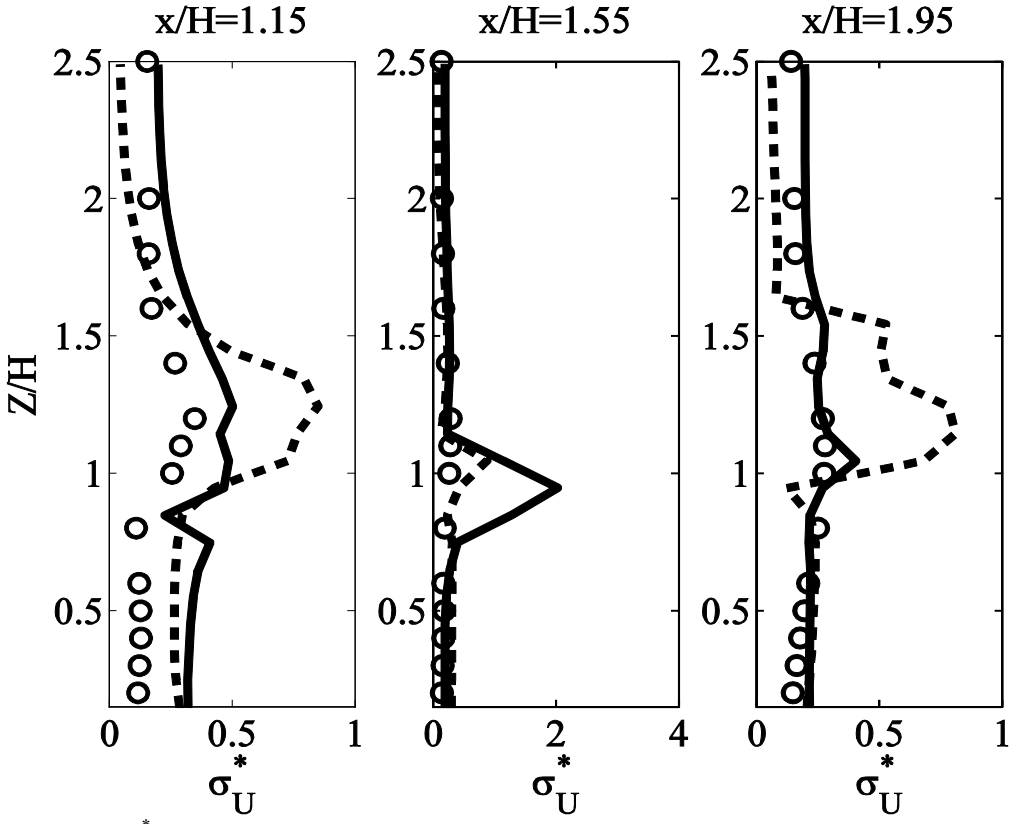


Figure 4.14: σ_U^* vertical profile comparison between the 3D GLEs model (-), the QUIC Plume model (- -) and the wind-tunnel data (o) at *x/H*=1.15, 1.55 and 1.95 and centerline (*y/H*=0) within the first street canyon.

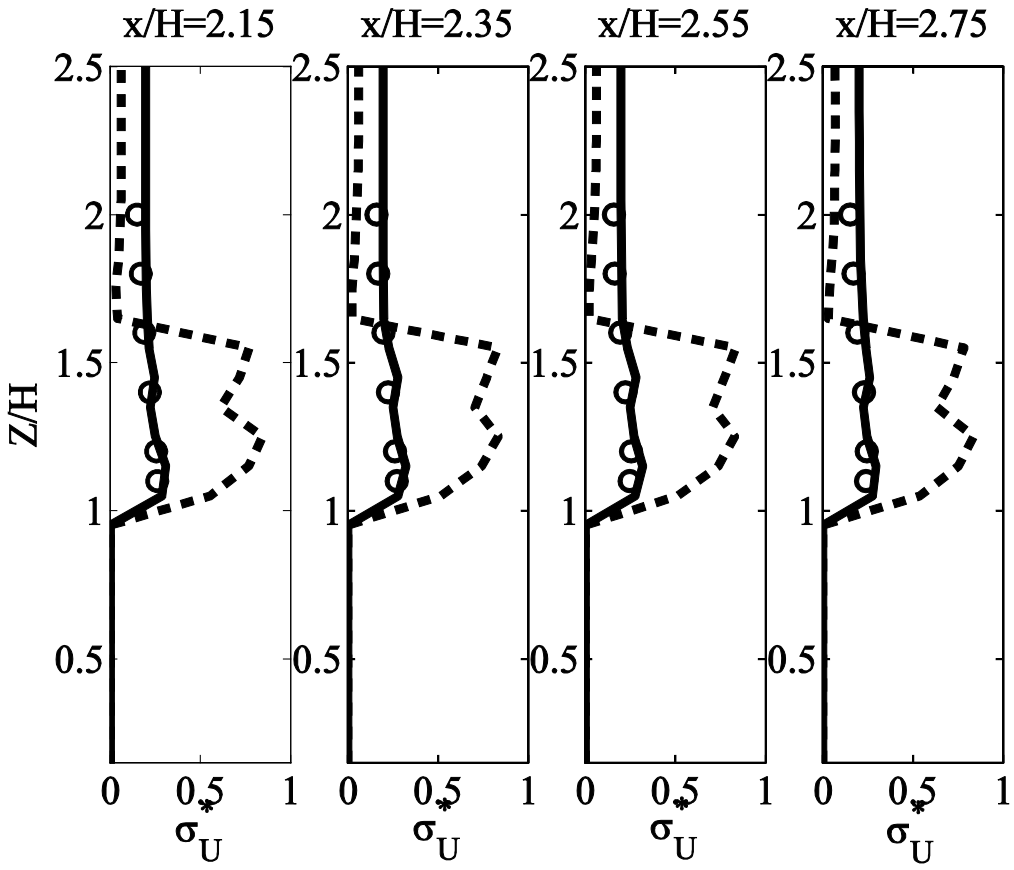


Figure 4.15: σ_U^* **vertical profile comparison between the 3D GLEs model (-), the QUIC Plume model (- -) and the wind-tunnel data (o) at *x/H*=2.15, 2.35, 2.55 and 2.75 and centerline (*y/H*=0) at the rooftop of the second centerline building.**

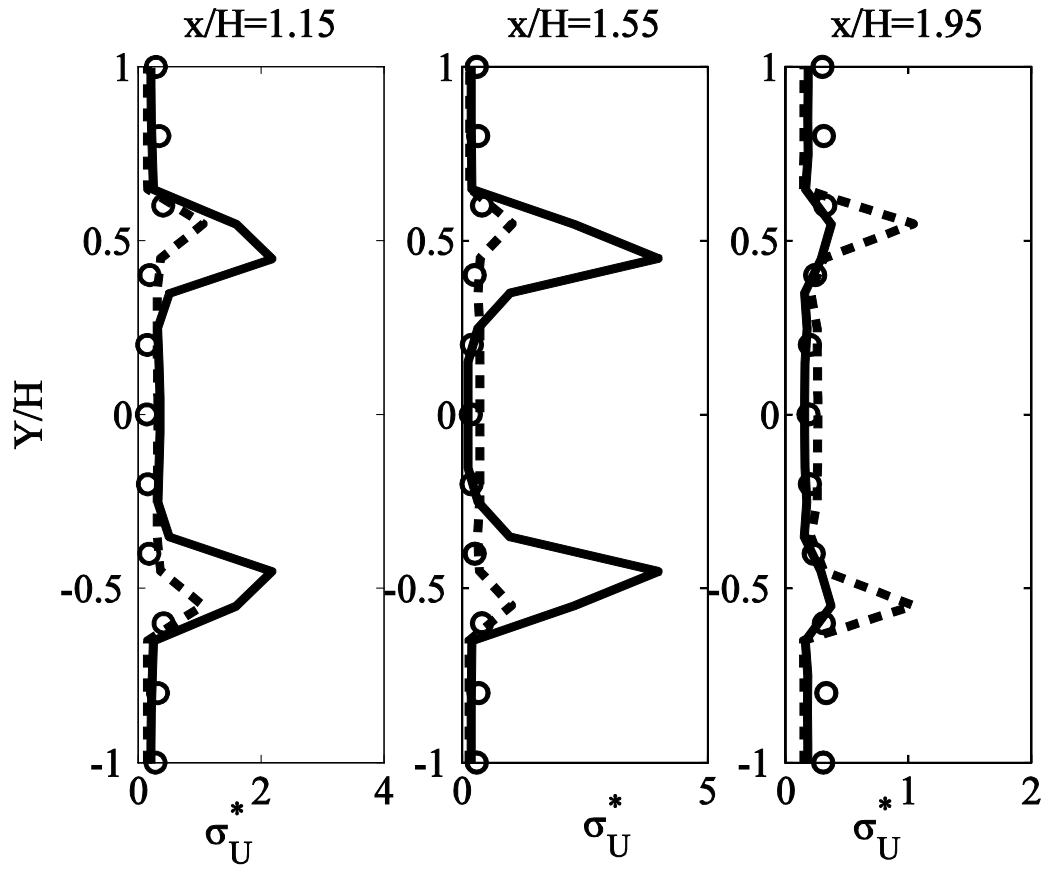


Figure 4.16: σ_U^* lateral profile comparison between the 3D GLEs model (-), the QUIC Plume model (- -) and the wind-tunnel data (o) at *x/H*=1.15, 1.55 and 1.95 at *z/H*=0.2 within the first street canyon.

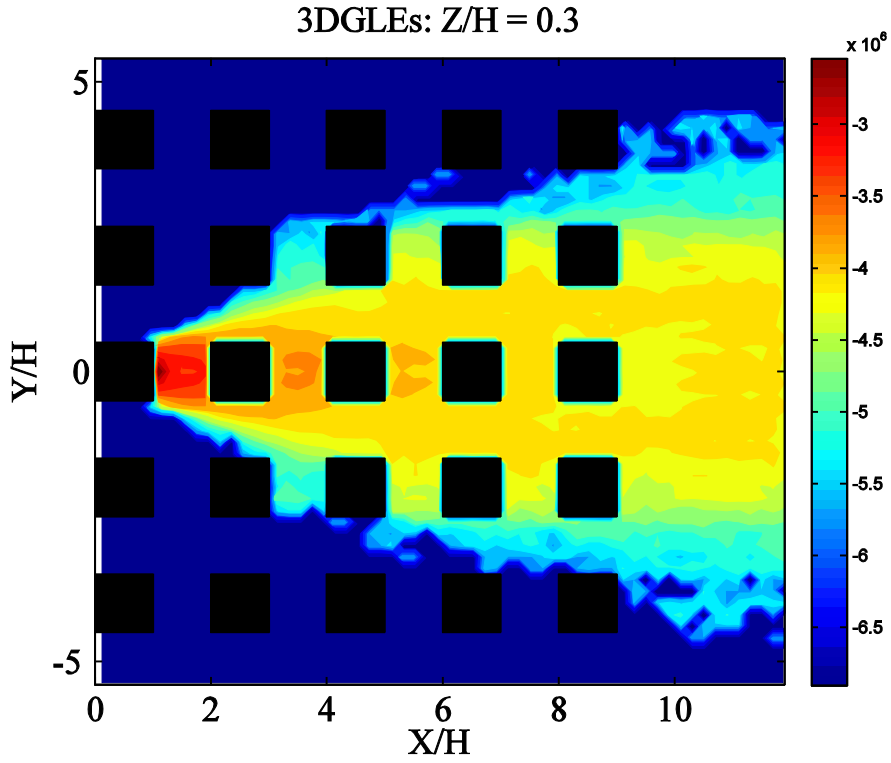


Figure 4.17: Horizontal concentration contour at *z/H*=0.3 for the 3D GLEs model.

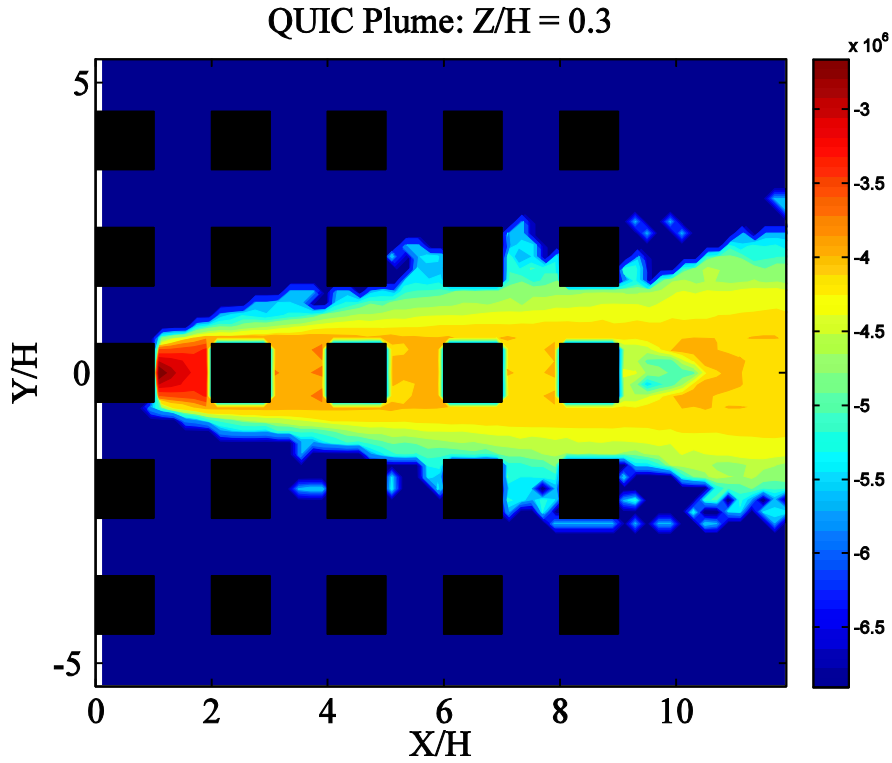


Figure 4.18: Horizontal concentration contour at *z/H*=0.3 for the QUIC Plume model.

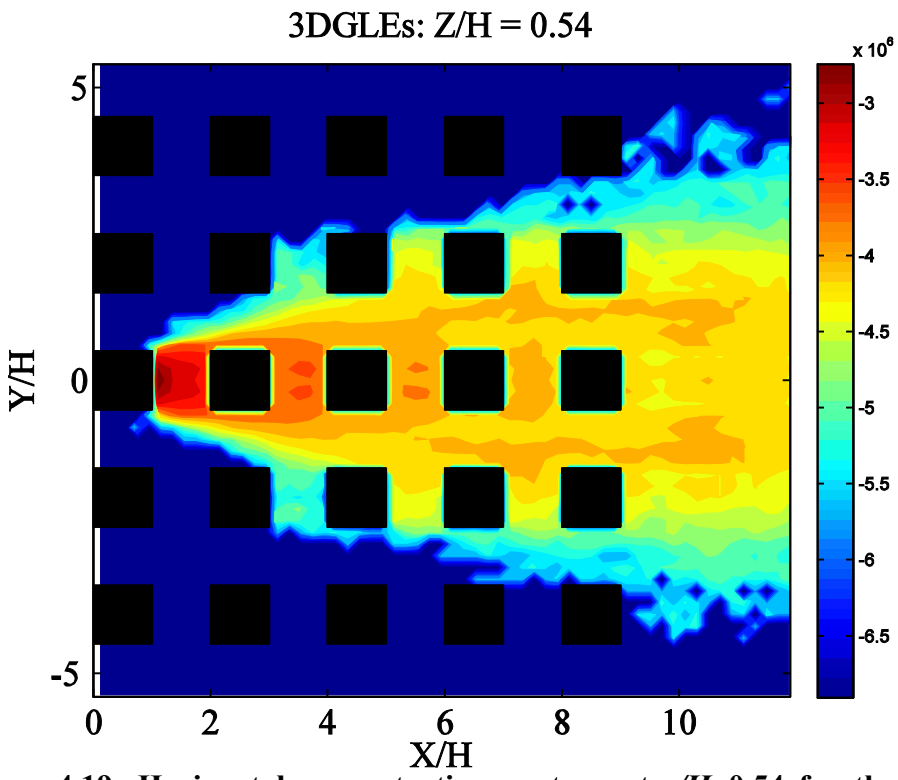


Figure 4.19: Horizontal concentration contour at *z/H*=0.54 for the 3D GLEs model.

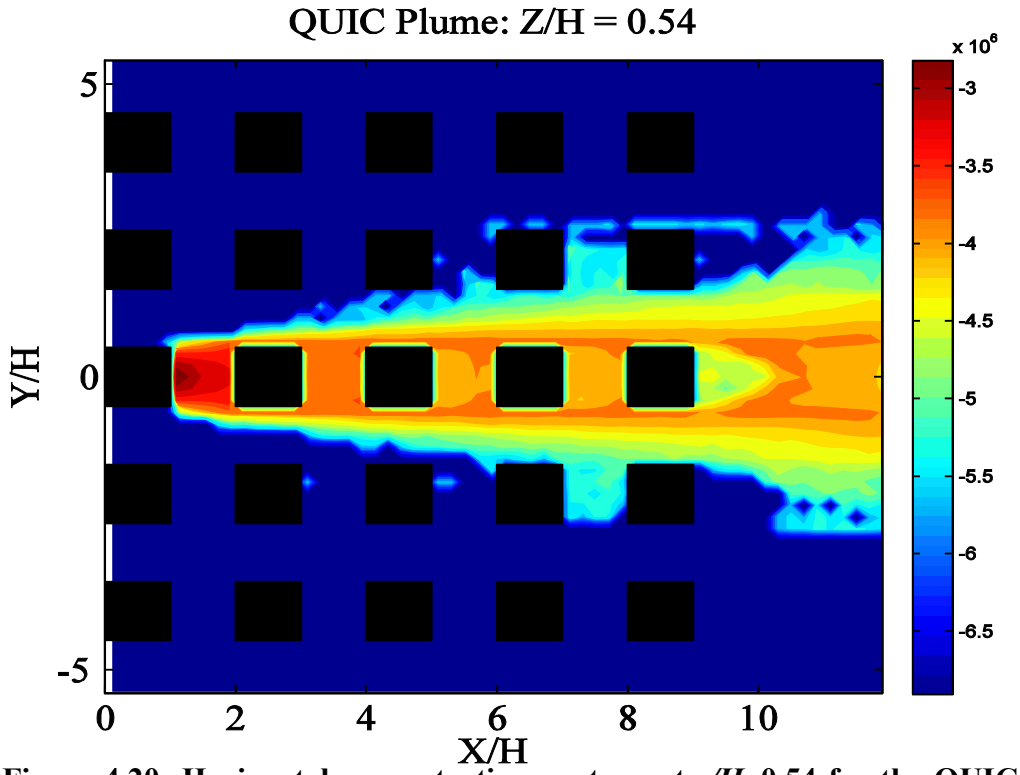


Figure 4.20: Horizontal concentration contour at *z/H*=0.54 for the QUIC Plume model.

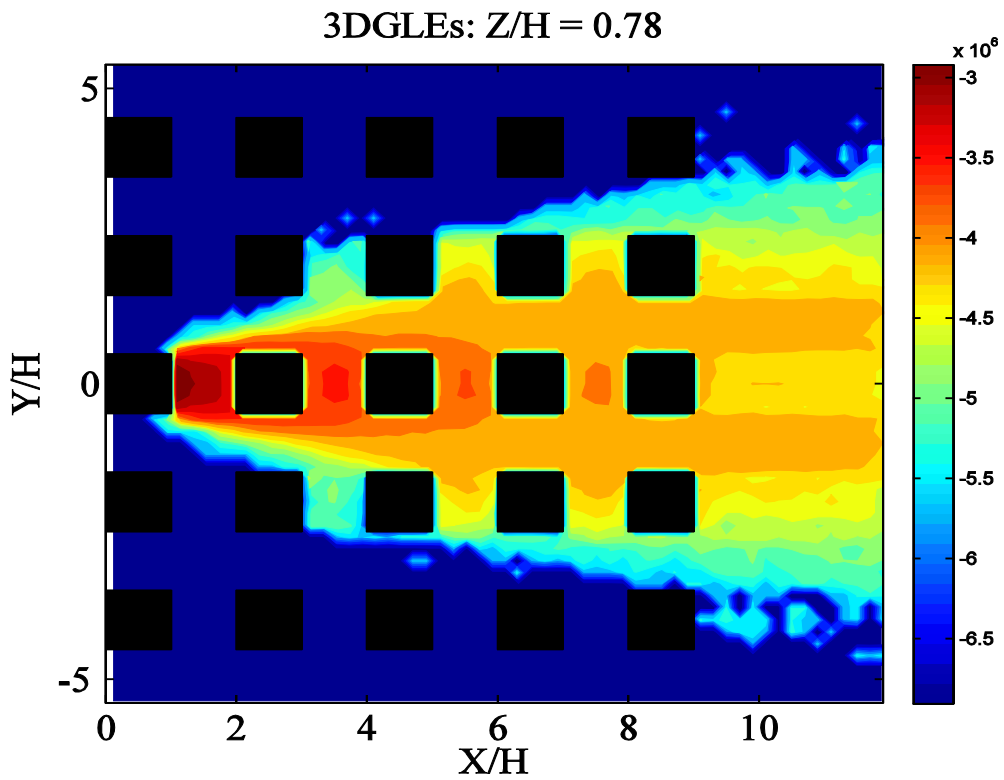


Figure 4.21: Horizontal concentration contour at *z/H*=0.78 for the 3D GLEs model.

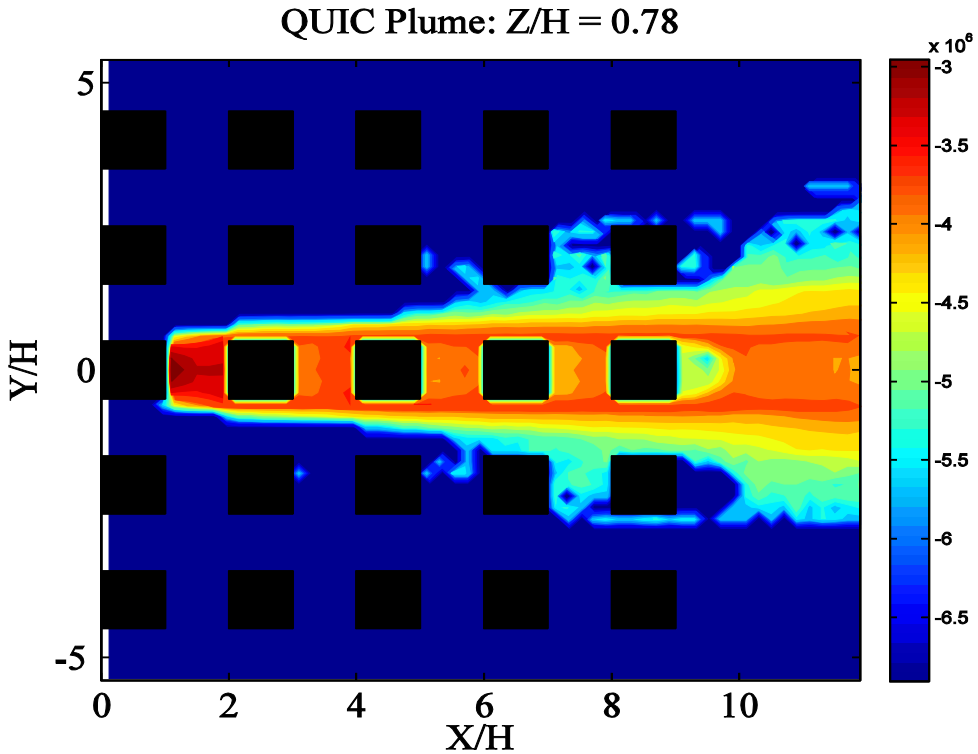


Figure 4.22: Horizontal concentration contour at *z/H*=0.78 for the QUIC Plume model.

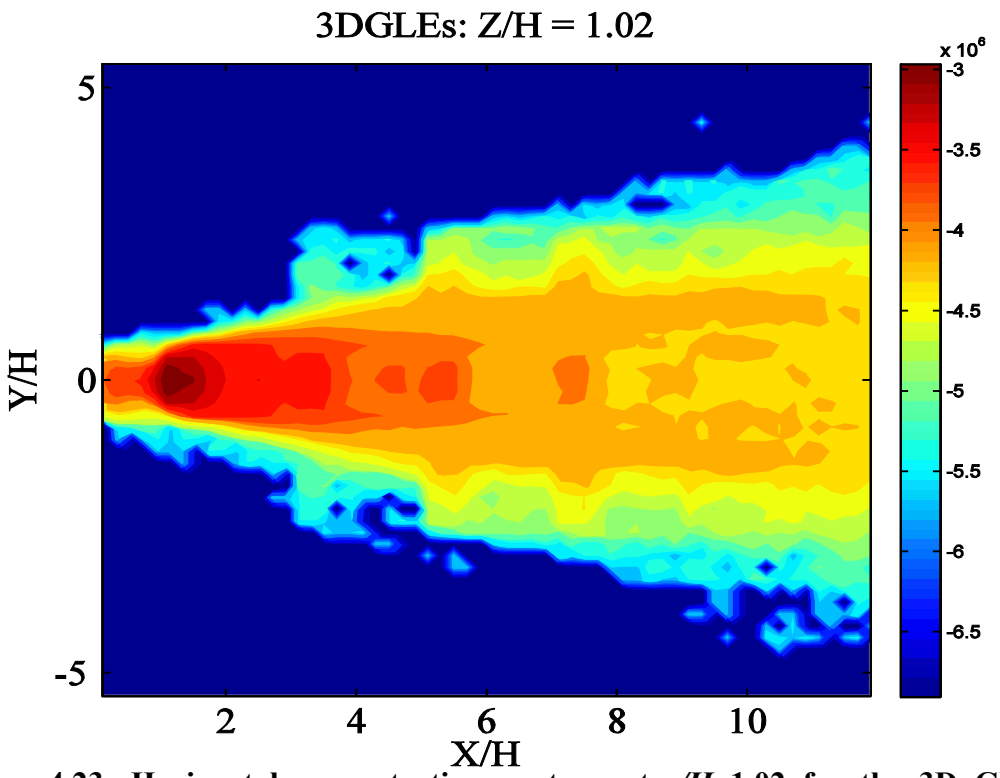


Figure 4.23: Horizontal concentration contour at *z/H*=1.02 for the 3D GLEs model.

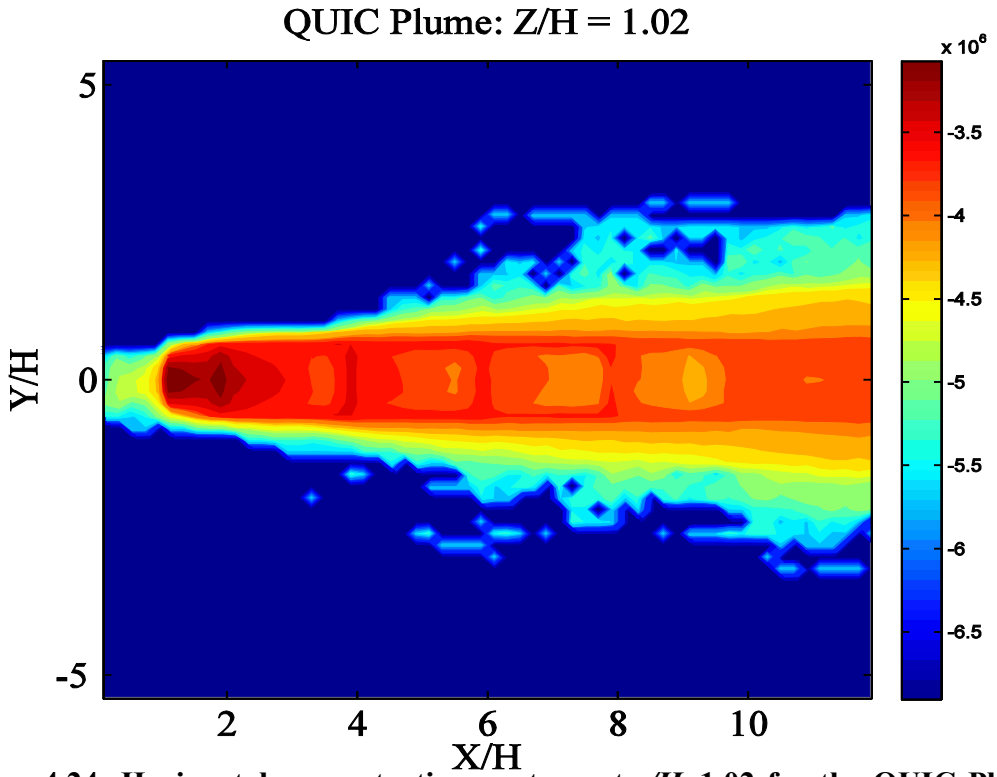


Figure 4.24: Horizontal concentration contour at *z/H*=1.02 for the QUIC Plume model.

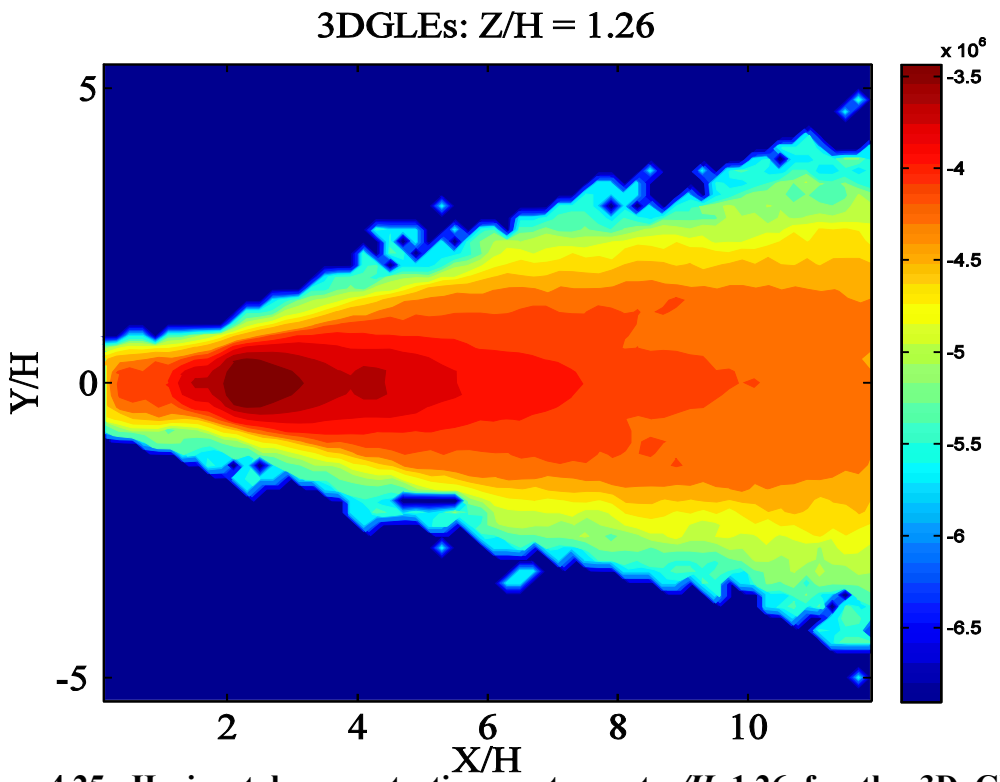


Figure 4.25: Horizontal concentration contour at *z/H*=1.26 for the 3D GLEs model.

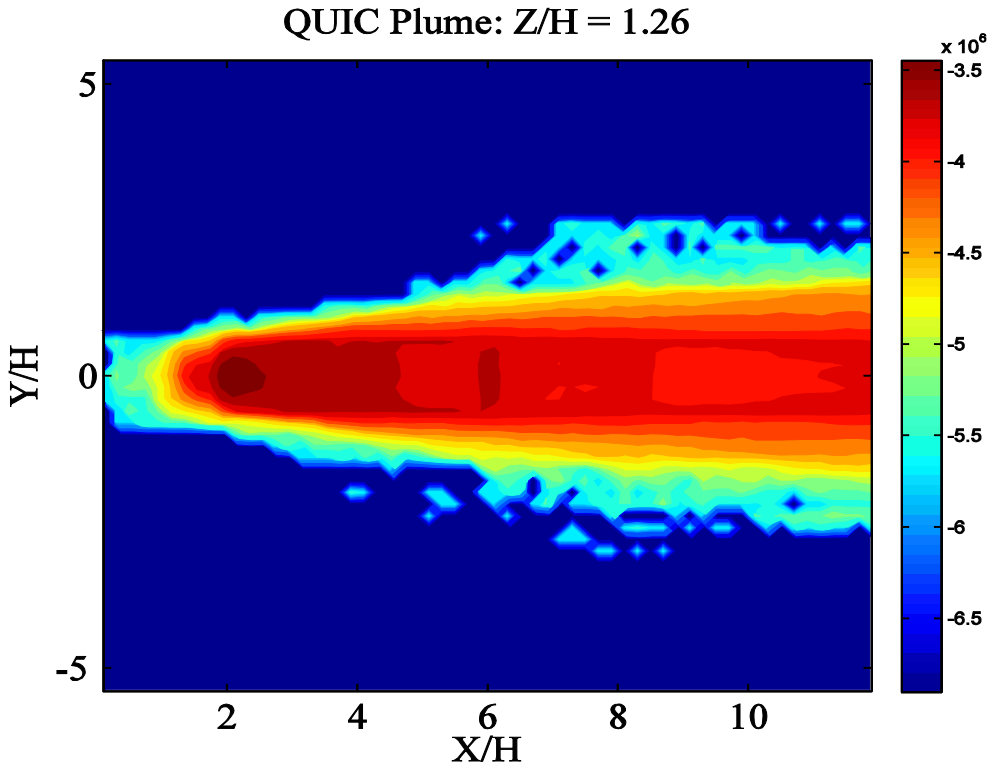


Figure 4.26: Horizontal concentration contour at *z/H*=1.26 for the QUIC Plume model.

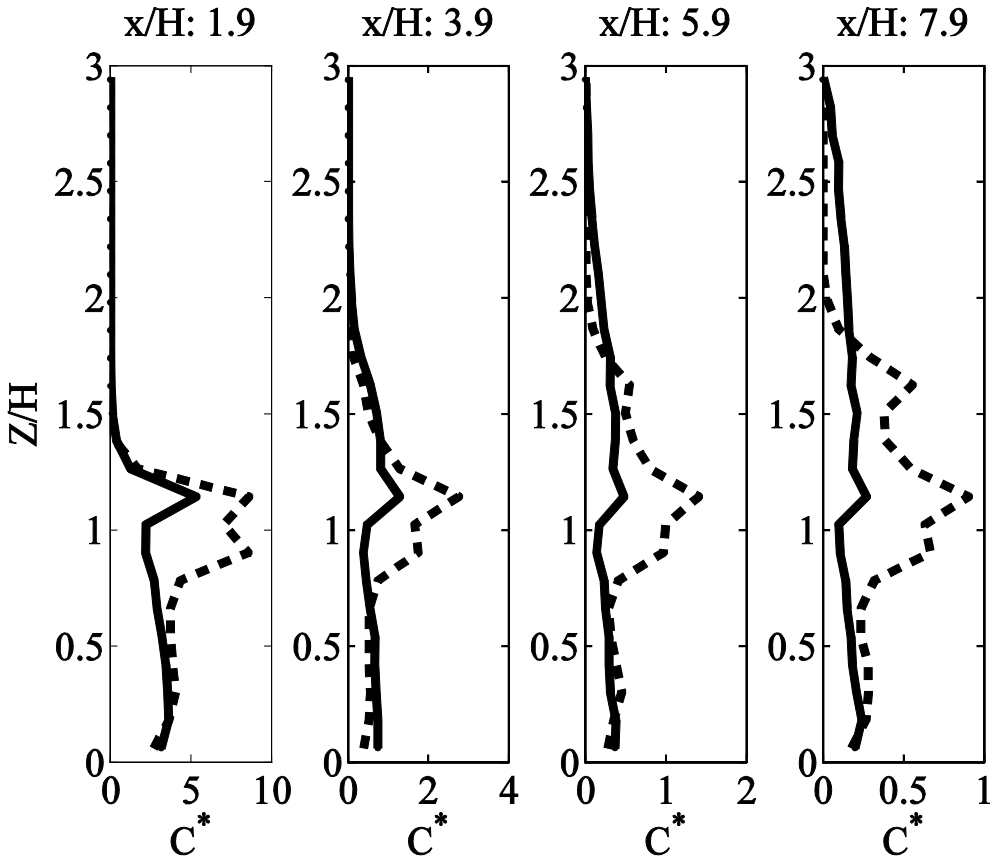


Figure 4.27: Vertical concentration profile comparison of the 3D GLEs model (-) and the QUIC Plume model (- -) close to the rooftops of the downstream street canyons at *y/H*=0.

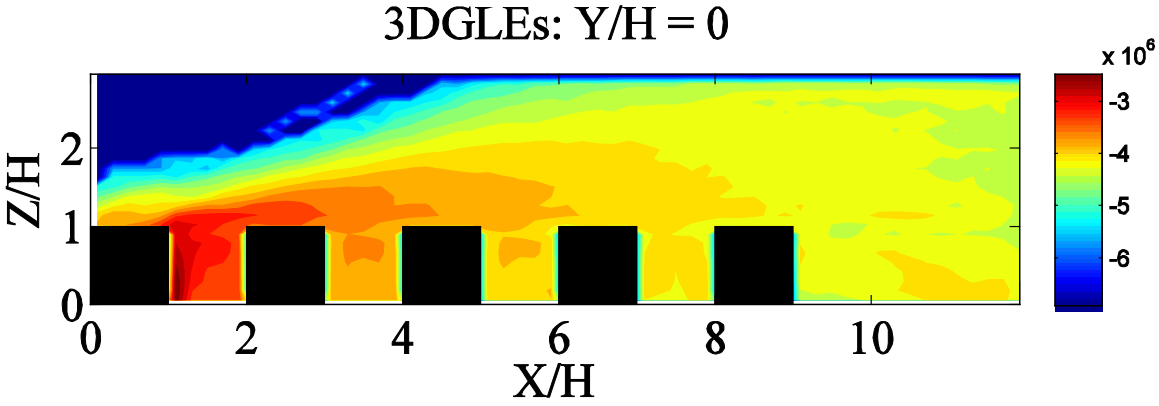


Figure 4.28: Vertical concentration contour at *y/H*=0 for the 3D GLEs model.

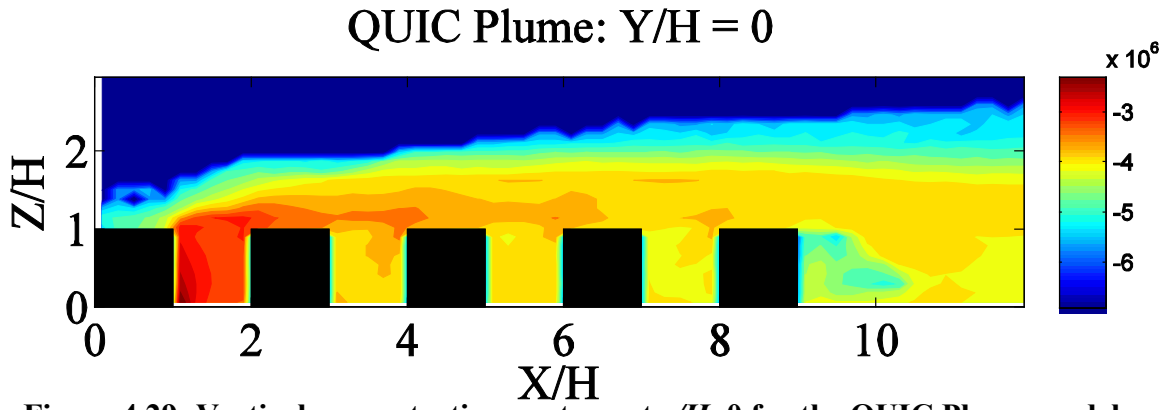


Figure 4.29: Vertical concentration contour at *y/H*=0 for the QUIC Plume model.

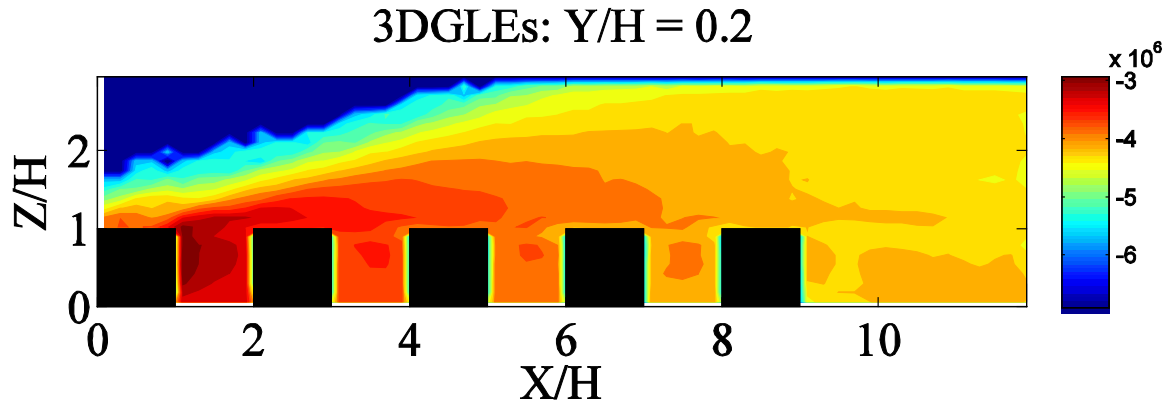


Figure 4.30: Vertical concentration contour at *y/H*=0.2 for the 3D GLEs model.

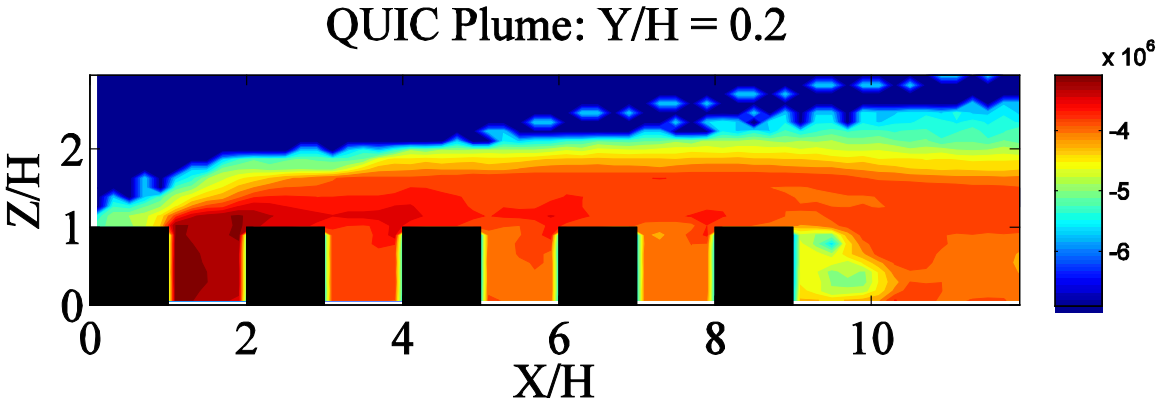


Figure 4.31: Vertical concentration contour at *y/H*=0.2 for the QUIC Plume model.

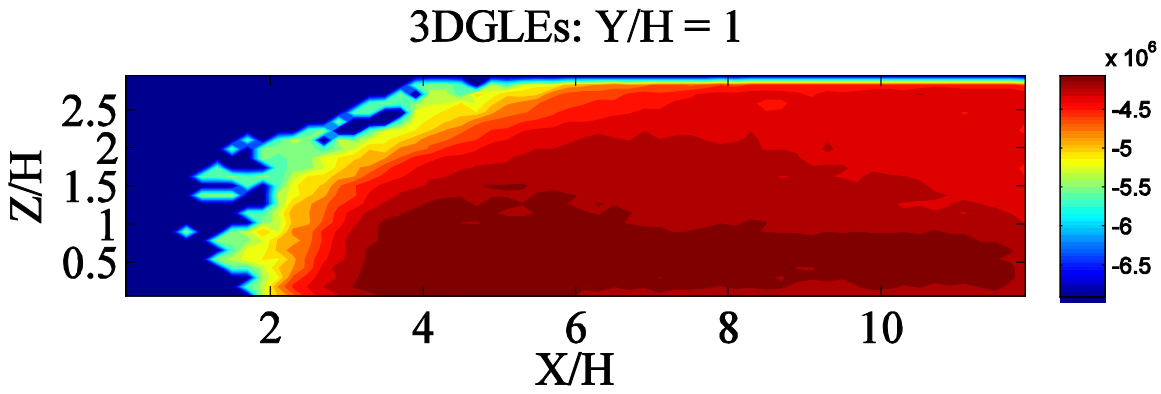


Figure 4.32: Vertical concentration contour at *y/H*=1 for the 3D GLEs model.

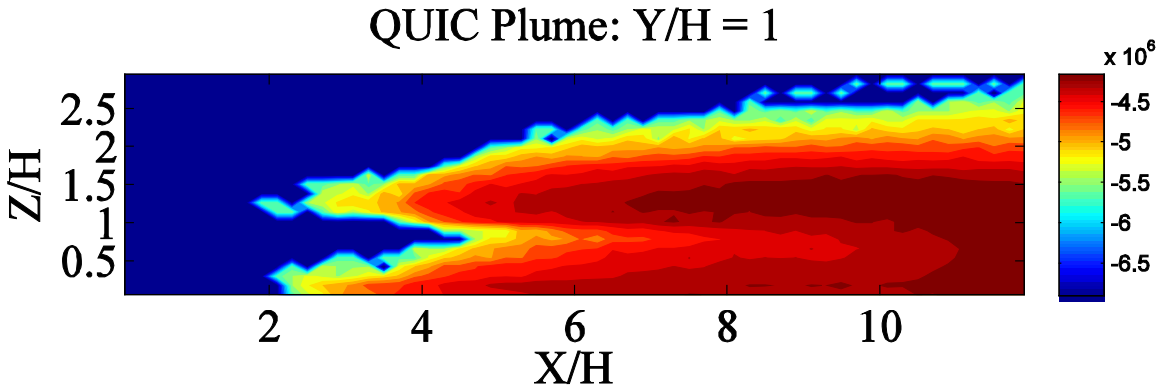


Figure 4.33: Vertical concentration contour at *y/H*=1 for the QUIC Plume model.

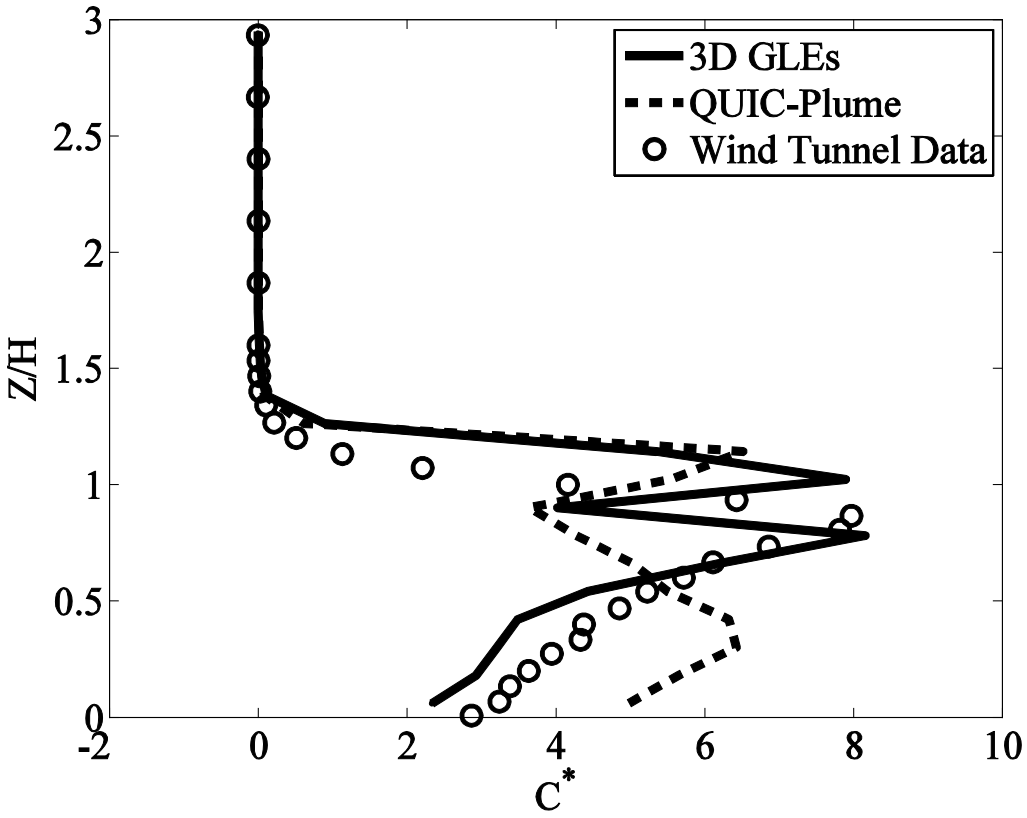


Figure 4.34: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the test data at the center of the first street canyon (*x/H*=1.5) and the centerline (*y/H*=0).

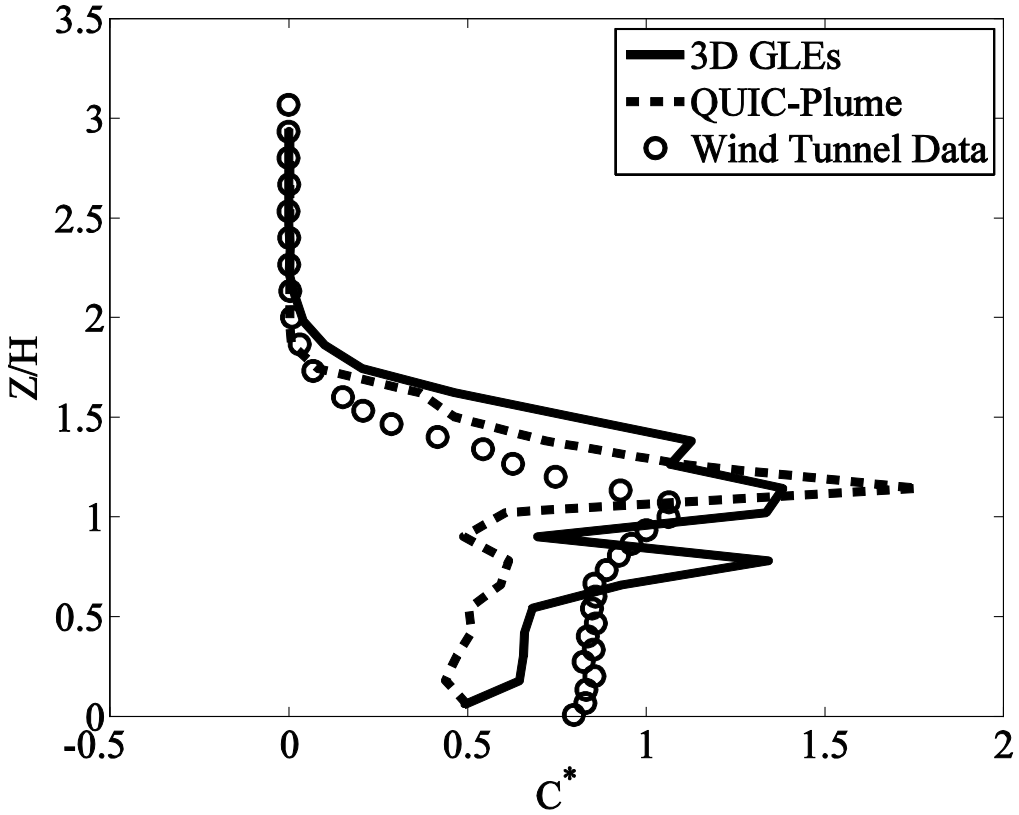


Figure 4.35: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the test data at the center of the second street canyon (*x/H*=3.5) and centerline (*y/H*=0).

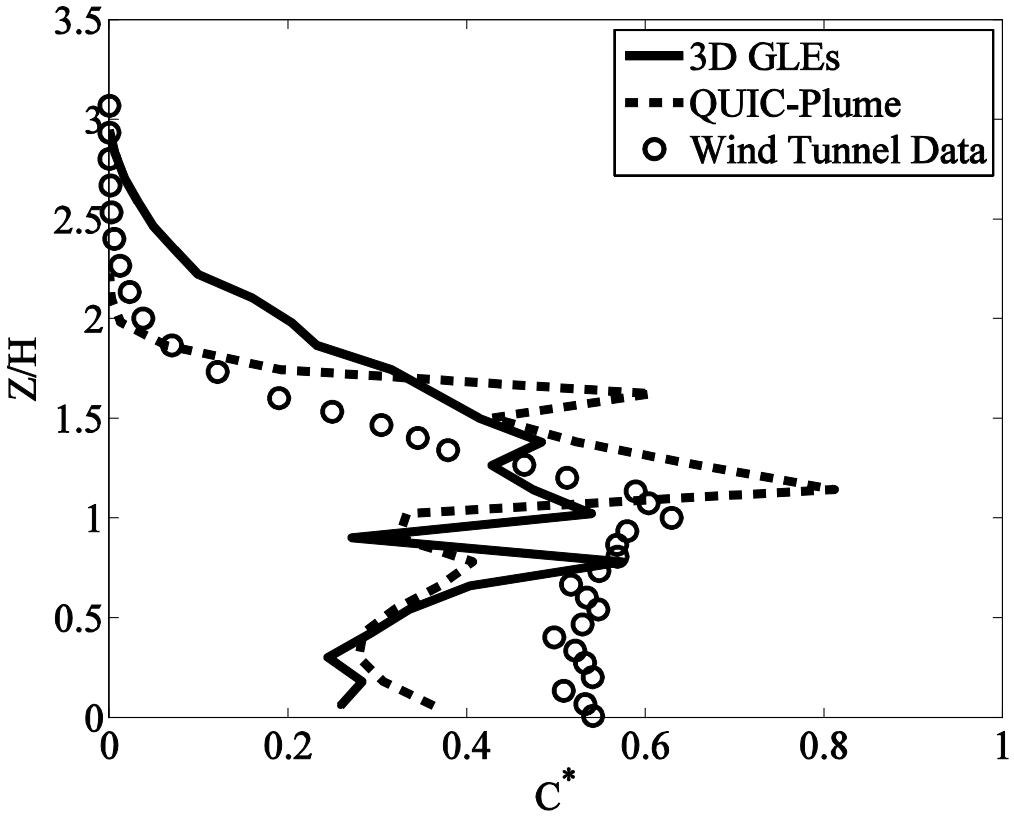


Figure 4.36: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the test data at the center of the first street canyon (*x/H*=5.5) and centerline (*y/H*=0).

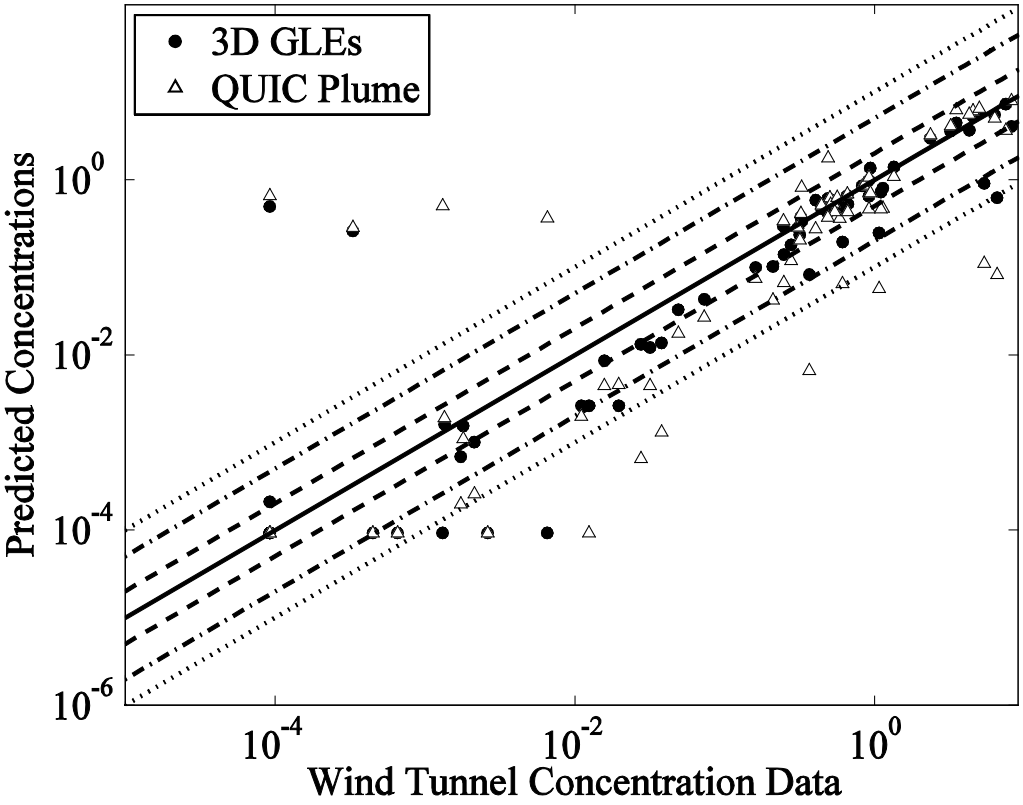


Figure 4.37: Paired scatter plot of the wind-tunnel concentration data and the predicted concentrations by the 3D GLEs model and the QUIC Plume model. A factor of 2 is represented by (- -), a factor of 5 is represented by (-.) and a factor of 10 is represented by (...).

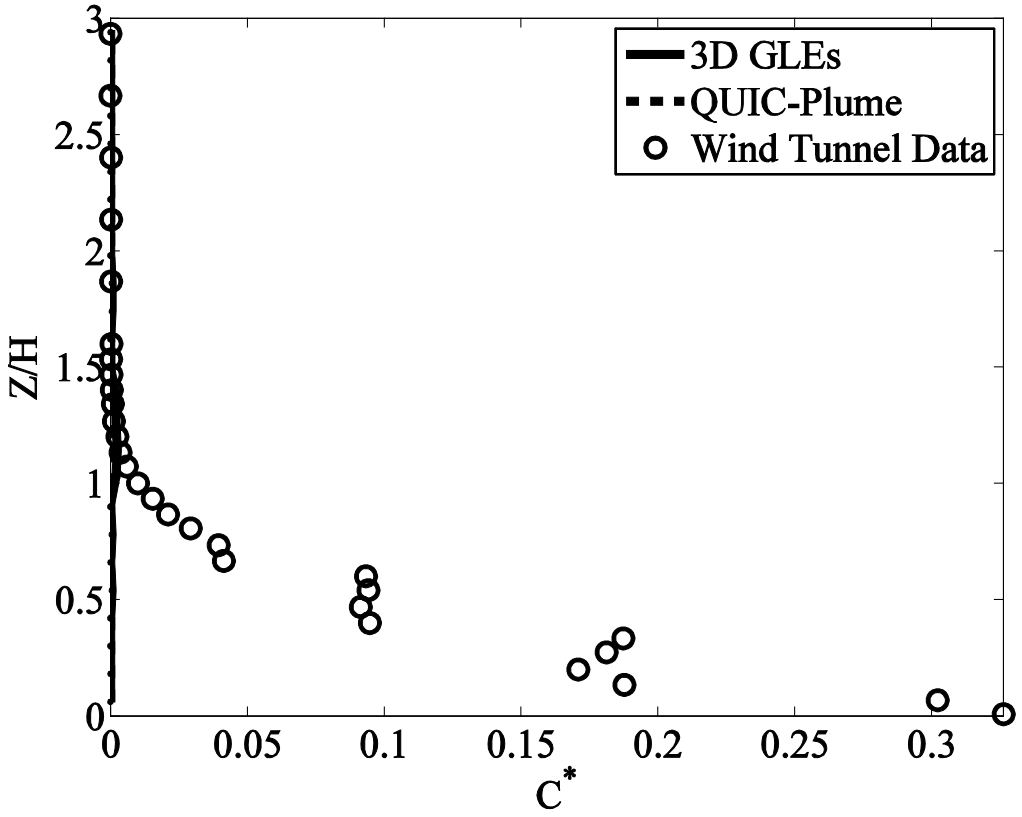


Figure 4.38: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the test data at *x/H*=1.5 and *y/H*=1 (center of the street channel).

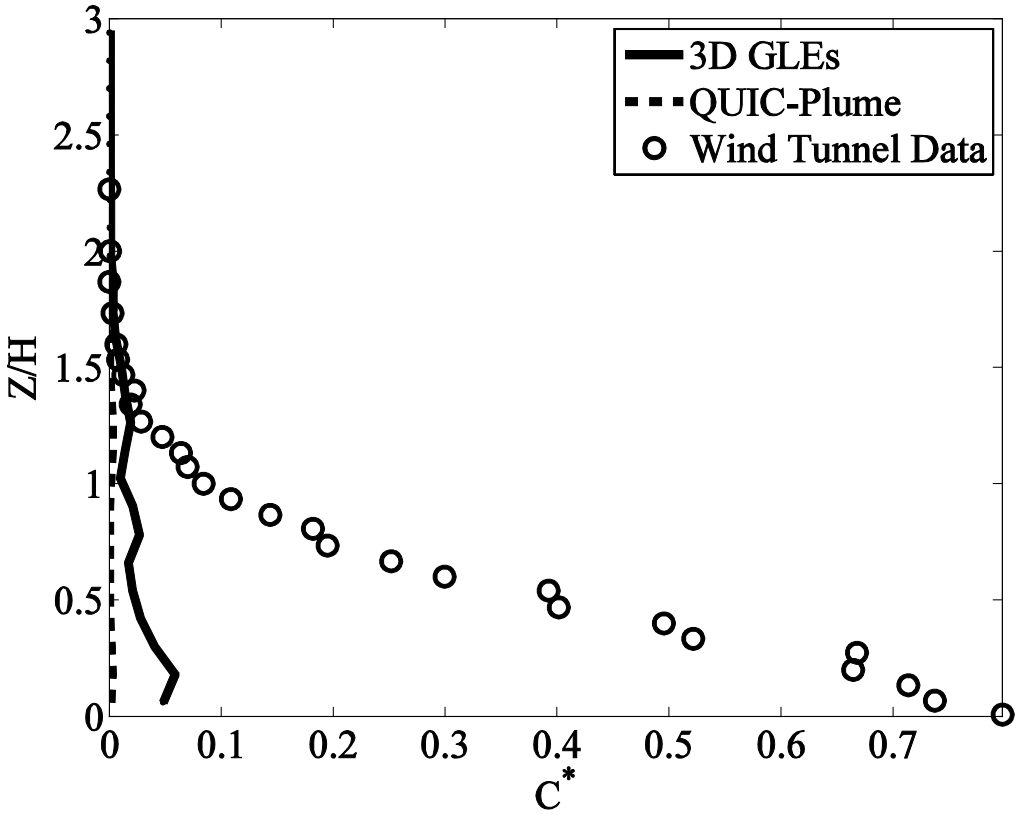


Figure 4.39: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the test data at *x/H*=2.5 and *y/H*=1 (center of the street channel).

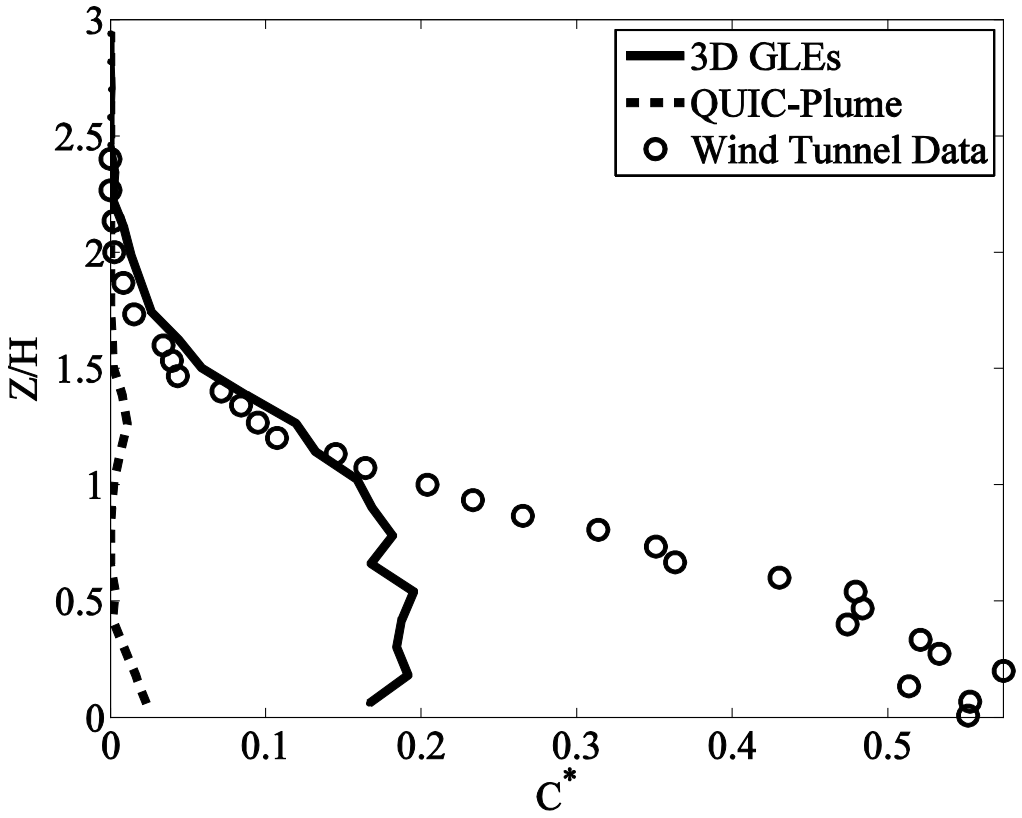


Figure 4.40: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the test data at *x/H*=3.5 and *y/H*=1 (center of the street channel).

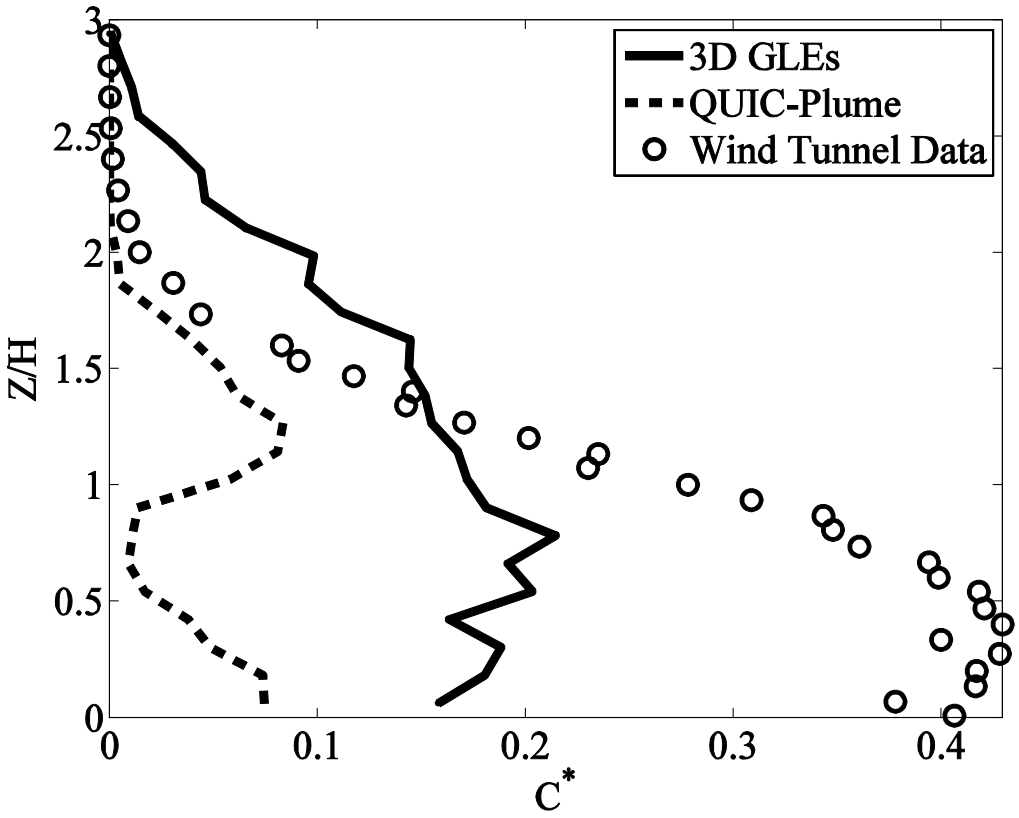


Figure 4.41: Vertical normalized concentration profile comparison between the 3D GLEs model, QUIC Plume model and the test data at *x/H*=5.5 and *y/H*=1 (center of the street channel).

5. CONCLUSIONS

The buildings and other structures present in an urban area alters the flow field significantly by causing updrafts and downdrafts, channeling between buildings, areas of calm winds adjacent to strong winds, and horizontally and vertically rotating-eddies between buildings, at street corners, and other places within the urban canopy (see review by Hosker (1984)). This makes it very difficult to devise fast response urban dispersion models that will work at the street canyon to neighborhood scales.

The methodology developed by Röckle (1990) to quickly compute a 3D wind field around buildings using an empirical-diagnostic approach is unique and a powerful tool. Röckle-type models do not solve transport equations for momentum or energy; rather, they rely heavily on empirical parameterizations and mass conservation. The MR (the so-called "modified" Röckle) model attempts to build on the strengths of the SR (standard Röckle) model and introduces additional physically-based but simple parameterizations that significantly improve the results in most regions of the flow in the 7 x 11 array and wide street canyon. The MR model produces vortices within street canyons that have velocities that compare much more favorably to the experimental results with the vortices shifted inward away from the edges of the street canyon. This is largely accomplished by modeling the effect of advection and momentum diffusion from outside the street canyon into the street canyon on the sides and from aloft.

We expect that these improvements in the wind field will result in improved dispersion calculations in built environments. We stress the importance of testing multibuilding parameterizations under a wide range of nonidealized conditions. Since it is quite rare that buildings in real cities take on the form from which the original parameterizations were developed, the model may not yield physically reasonable results when generalized. Hence, it is imperative to rigorously evaluate the model for a wide range of scenarios. This is one of the greatest challenges in utilizing Röckle type wind models.

The fast response urban wind model enables the fast response atmospheric dispersion modeling as the mean wind field is readily available for the dispersion estimates. These dispersion models are valuable tools that can aid first responders in making decisions regarding accidental or deliberate releases of chemical or biological agents in complex urban environments. Chapter 3 demonstrates that dispersion models implemented on the GPU (Graphics Processing Unit) enables real-time performance of these models and also makes them suitable for applications involving the display of dispersion phenomena in urban virtual environments. To our knowledge, this is the first attempt to integrate a fast response dispersion model into an urban virtual environment running in real-time. The results indicate that by using the GPU, a substantial performance benefit can be obtained in the advection process of the dispersion model. The performance of GPU Plume is two orders of magnitude faster than its CPU implementation, while preserving computational accuracy. The concentration profiles obtained from GPU Plume are in good agreement with the CPU implementation with small variations attributed to the differences in the GPU and CPU implementations. The

single building algorithm does show a small performance penalty over the uniform flow case and is associated with the added memory textures and reflection algorithm.

The immediate benefit of real-time visualization, obtained by rendering the dispersion data on the screen, is a novel approach for probing the evolving dispersion field, which further enables one to qualitatively understand the dispersion phenomena better. The visualization aspect of GPU Plume is a powerful tool that also enables the user to view the turbulence contours and the evolving concentration field in real-time. An additional exploitation of the benefits of real-time visualization of GPU Plume is modification of simulation parameters and models during the simulation. This will allow users to receive immediate feedback regarding changes in material properties, building positions and physical submodels.

Programming the GPU is currently a nontrivial task, but does provide increased performance over CPU implementations. Programming GPUs is likely to become easier as higher level languages are developed to access the graphics hardware. Graphics card manufacturers, such as NVIDIA, are developing additional tools to help program these cards using C APIs (Application Programming Interfaces) to access the hardware. As part of our continued work, we are now investigating how NVIDIA's CUDA (Compute Unified Device Architecture) framework might compare in performance and functionality to our current implementation. For our future work, we plan to optimize our system for increased functionality, including support for generalized building structures, greater than 10 million particles, and multi-GPU configurations. Multi-GPU arrangements may help with speeding up the advection of an increased numbers of particles since the work could be spread across the set of GPUs.

The GPU Plume model utilizes the simplified Langevin equations (SLEs) for transport and dispersion of a release due to the presence of a large number of terms in the full or generalized form of Langevin equations (GLEs). The presence of unstable modes due to the stiffness of the GLEs has also been a problem for the direct implementation of the GLEs into a Lagrangian dispersion model. Although unstable modes also exist in the SLEs, the SLEs are considered slightly stable due to the drastic reduction in the number of terms in the SLEs. The GLEs can be implemented using a fractional step method (FSM) which partially solves the stiffness problem exhibited by the GLEs (Yee and Wilson 2007). The FSM decomposes the GLEs into three steps and the first two steps are solved analytically, circumventing the unstable modes. The third step is performed using the Euler forward method, which does not guarantee the absence of the unstable modes as no analytical mapping is available for this third step.

The analytical mapping functions used in the first two steps of the FSM helps in revealing the existence of potential unstable modes in the particle trajectory before the advection process even begins in the dispersion model. For example, in the first step of the FSM, the eigenvalues of the $\mathbf{A}_1$ matrix (see Chapter 4) are computed. The presence of positive eigenvalues indicates the existence of the unstable modes in the particle trajectory. In the dispersion model, the eigenvalues of the $\mathbf{A}_1$ matrix are computed before the particle advection process starts. Therefore, if the eigenvalues are positive, the dispersion modeler can further investigate the potential weaknesses in the turbulence model and the background wind field. On the other hand, in the Euler forward method (unlike analytical integration functions), no such information is available to the dispersion modeler beforehand to identify the potential unstable modes. Therefore, it is

recommended to use the analytical mapping functions for solving the SLEs also rather than explicitly using the Euler forward method to integrate the SLEs (in conjunction with coordinate rotation in the mean wind direction for accommodating horizontally homogenous assumptions).

The Gaussian and non-Gaussian test case results demonstrated that the 3D GLEs implementation with the FSM works well for these simple test cases. For the Gaussian test case, the second and third steps of the FSM yield zero due to the absence of the velocity as well as turbulence stress gradients. Also, the first step of the FSM yielded a symmetric $\mathbf{A_1}$ matrix due to the absence of the drift terms (as no velocity and turbulence stress gradients exist). Therefore, there was no possibility of the existence of the unstable modes in the Gaussian test case. This information is available to the dispersion modeler before the actual advection of the particle starts in the dispersion model. In the non-Gaussian test case, the second and third step did not yield zero values due to the presence of the velocity and turbulence stress gradients. The $\mathbf{A_1}$ matrix for the non-Gaussian test case was symmetric, as the second term of the $\mathbf{A_1}$ matrix was zero (see Eqs. 4.15) for this case. The second and third steps did not yield unstable modes for this test case possibly due to the simplicity of this case.

Another simple test case of a spatially varying shear layer was used to investigate possible unstable modes associated with the positive eigenvalues of the $\mathbf{A_1}$ matrix. The turbulence model used for this test case yielded favorable results when compared with the experimental data. This test case demonstrated that whenever the model computed product of C_0 and dissipation (ε) is less than the gradient of the turbulence stress, the unstable modes may exist during the advection process. These unstable modes may yield

rogue trajectories where a particle may travel a significant distance within a small time step. To prevent these rogue trajectories, the dispersion modelers normally limit the size of the fluctuating component of the velocity by some multiple of the standard deviation of the wind in the respective directions (Yee and Wilson 2007). The QUIC Plume model uses 4.0 as a limiting factor and the 3D GLEs model uses 2.5 as the limiting factor to limit the unphysical rogue trajectories. The last test case of a more realistic urban setup (7 x 11 cubical array of buildings) yielded positive eigenvalues for the $\mathbf{A_1}$ matrix at some regions close to the cubical buildings. The values of the positive eigenvalues were however close to zero for the most part. Part of the reason for the existence of the positive eigenvalues may be the background wind field and a simple turbulence model. Personal communication with Dr. Yee confirmed the existence of positive eigenvalues even for the wind fields obtained from CFD (Computational Fluid Dynamics) models, which are considered more accurate than the diagnostic wind models like QUIC-URB.

All the test cases considered for validation of the 3D GLEs model yielded good results in comparison with the test data as well as the SLEs-based model QUIC Plume. The improved wind model used to drive the 7 x 11 test case in both the 3D GLEs and QUIC Plume model was in good agreement with the test data (see Chapter 2). The turbulence model used to drive the 3D GLEs was a local mixing length model with an addition of a constant 0.3 to enhance the turbulence within the street canyons. The constant 0.3 was added to match the turbulence level within the street canyons with the test data. Although this addition raised the turbulence level at the edges of the street canyons, the turbulence levels within the street canyons matched the test data well. The vertical turbulence profiles also showed a good match with the test data. The QUIC

Plume model uses its nonlocal mixing length model (Williams et al. 2004) for driving the 7 x 11 test case. The concentration contours showed that the 3D GLEs model exhibits more lateral and vertical dispersion as compared with the QUIC Plume model. The plume in the street channels (adjoining centerline buildings) was well-mixed in the case of the 3D GLEs model, indicating enhanced lateral dispersion exhibited by the 3D GLEs model. The centerline concentration profiles showed that the 3D GLEs model performs comparable or better than the QUIC Plume model for the most part. The concentration profiles in the street channel indicate that the test data exhibit more lateral dispersion close to the ground as compared with both the 3D GLEs model and the QUIC Plume model. The paired scatter plot also indicates that the 3D GLEs model was in good agreement with the test data.

The future work will include more validation studies of the 3D GLEs model and implementation of the 3D GLEs model on a GPU-based architecture to realize real-time run times. As the FSM is capable of providing insights into the unstable modes before the actual advection of the particles, these insights may be utilized to further improve the diagnostic wind model and turbulence models to better suit the needs of a Lagrangian dispersion model based on Langevin equations.

5.1 References

Hosker, R. P., 1984: Flow and diffusion near obstacles. In Atmospheric Science and Power Production (edited by D. Randerson), 241-326. Publication DOE/TIC-27601, Technical Information Centre, U. S. Department of Energy, Washington, D.C.

Röckle, R., 1990: Bestimmung der Stromungsverhaltnisse im Bereich komplexer Bebauungsstrukturen . Ph.D. dissertation, Vom Fachbereich Mechanik, der Technischen Hochschule Darmstadt, Germany.

Williams, M. D., M. J. Brown, D. Boswell, B. Singh, and E. M. Pardyjak, 2004: Testing of the QUIC-plume model with wind-tunnel measurements for a high-rise building. *5th AMS Urban Env. Conf.*, Vancouver, BC.

Yee, E., and J. Wilson, 2007: Instability in Lagrangian stochastic trajectory models, and a method for its cure. *Boundary-Layer Meteorology*, **122,** 243-261.