5-11-2002

Simulation of Pollutant Transport in an Urban Area

Luxin Wang

Simulation of Pollutant Transport in an Urban Area

By

Luxin Wang

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computational Engineering
in the College of Engineering

Mississippi State, Mississippi

May 2002

Simulation of Pollutant Transport in an Urban Area

By

Luxin Wang

Approved:

Pasquale Cinnella
Associate Professor of
Aerospace Engineering
(Major Professor)

Bharat Soni
Professor of Aerospace
Engineering
(Committee Member)

Edward Luke
Assistant Professor of
Computer Science
(Committee Member)

Lafayette Taylor
Associate Research Professor of
Computational Engineering
(Committee Member)

Boyd Gatlin
Associate Professor of Aerospace
Engineering andGraduate Coordinator

A. Wayne Bennett
Dean of the College of Engineering

Name: Luxin Wang

Date of Degree: May 11, 2002

Institution: Mississippi State University

Major Field: Computational Engineering

Major Professor: Dr.Pasquale Cinnella

Title of Study: Simulation of Pollutant Transport in an Urban Area

Pages in Study: 70

Candidate for Degree of Master of Science

The present study is concerned with the feasibility of simulating the release and transport of a “generic” pollutant within an urban area. The motivation behind this effort lies in the need for accurate and timely predictions of the effects of a pollutant release in a densely populated region, in order to coordinate relief and/or evacuation efforts or design rapid-response scenarios in the event of a possible accident.

In order to achieve the above goal, following geometrical, physical, and algorithmic steps have to be taken, generation of a three-dimensional grid, numerical solution of the governing fluid dynamics equations, and numerically solving the transport (reaction/advection/diffusion) equations for a generic pollutant.

Some preliminary results will be presented at this time. A sample of representative results will be shown for the generation and transport of a generic pollutant “plume”, using different starting wind conditions and release points.

Future efforts in this endeavor will focus on obtaining reasonably accurate simulations of pollutant transport at a reasonable computational price. Specifically, “reduced” models for the geometry and/or the governing equations will be investigated.

ACKNOWLEDGMENTS

Working with Dr. Pasquale Cinnella, my major professor, is an unforgettable pleasant experience. Dr. Cinnella supported my study selfishlessly and continuously in the past 20 months. I would like to express appreciation to Dr. Soni, Dr. Taylor, and Dr. Luke for being my committee members. I also want to say thanks to other members in Center for Computational System. I would like to thank Junxiao Wu and Lin Tang for their support and help in finishing my research work. There is one person who is supporting my study and will do so all through my life, my wife Min. I would like to thank her gratefully for her understanding and patience for our 'separation' during the past 8 months.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

NOMENCLATURE

A_j	Area of the jth face, m^2
C_i	The ith cell number
$c_{v\,s}$	Species Specific Heat at Constant Volume, J/kg $\cdot$ K
D_s	Diffusion Coefficient for species s, m^2/s
e_0	Gas total energy, J/kg
$h_{f\,s}$	Species Heat of Formation, J/kg
h_s	Enthalpy of species s, J/kg
I	Identity Matrix
$\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$	Unit vectors for a Cartesian frame of reference (x,y,z)
$K_{e,r}$	Reaction Equilibrium Constant
$k_{f,r}$	Forward Reaction Rate
$k_{b,r}$	Backward Reaction Rate
$\mathbf{n}$	Unit vector normal to a control surface
NR	Number of chemical reactions
NS	Number of species
p	Pressure, Pa
p_{ref}	Reference Pressure, Pa
$\mathbf{Q}$	Conservative variables
$\mathbf{q}$	Primitive variables

R_i	Residual for cell i
R_s	Gas Constant for species s, J/kg $\cdot$ K
S	Boundary surface for a Control Volume
s_{ref}	Entropy measured at Reference conditions, J/kg $\cdot$ K
$\mathbf{S}$	Inviscid flux vector in Navier Stokes equations
$\mathbf{S}_v$	Viscous flux vector in Navier Stokes equations
T	Temperature, K
T_{ref}	Reference Temperature, K
t	Time, s
$\mathbf{u}$	gas velocity vector, m/s
u, v, w	gas velocity in $\mathbf{i}, \mathbf{j}, \mathbf{k}$ directions, respectively, m/s
V	Volume, m^3
V_s	species diffusion velocity, m/s
$\mathbf{W}$	Source term vector in Navier Stokes equations
$\dot{w}_s$	Species Production Rate
Y_s	Species Mass Fraction (ρ_s/ρ)
x, y, z	Cartesian body axes, m
$\mathcal{M}_l$	Species Atomic Mass, Kmole
δ_{ij}	Kronecker tensor
$\delta\mathbf{r}$	Displacement vector
Θ	Energy flux term in the energy transport equation, J/m^2 $\cdot$ s
θ_v	Vibrational Temperature
λ	Conduction coefficient, J/s $\cdot$ m $\cdot$ K
μ	Viscosity coefficient, kg/s $\cdot$ m
ρ	Fluid Density, kg/m^3

ρ_s	Species Density, kg/m^3
τ	Stress tensor, kg/s$^2 \cdot$ m

CHAPTER I

INTRODUCTION

In recent years, the study of pollutant transport has become a very important topic in many scientific fields, such as environmental engineering, pollution control, wind engineering, and risk management. In our highly industrialized and populated society it is impossible to avoid pollution completely. Therefore, it is important to fully understand the complex process of transport and dispersion of pollutants.

Typically, the study of pollutant transport falls in three different categories: urban-local scale, regional scale, and mesoscale. Each of them presents different challenges and difficulties in reaching a satisfactory level of understanding. The present study is focused on the urban-local scale. The transport of pollutants over a portion of a city such as New Orleans will be studied. Modeling of urban area pollutant transport can provide very useful information for the following scenarios:

1. Industrial accidents and the prediction of the transport of polluting substances;

2. Intentional release of toxic substances in populated areas;

3. The establishment of policies/rules for responding to a pollutant release.

The simulation results can be used by many different interest groups, including rescue personnel such as the fire and police departments and insurance companies. In general, when an accidental release of harmful pollutants into the atmosphere happens, it is necessary to urgently estimate release conditions, e.g., emission point, period, and amount (hereinafter source parameters), and to predict the environmental impacts as quickly as possible. Thus quick response, i.e. the realization of real-time predictions, is demanded of the simulation code. The estimation procedure typically consists of an atmospheric dispersion simulation for the possible source parameters and a statistical analysis to find the release condition providing the best fitted prediction. Since the atmospheric simulation part follows a large number of trial and error calculations, it also needs much computational resources, which are not available at a single site. Therefore, it is very significant to develop a communication-intensive system using supercomputer resources all over the world if necessary, which are connected through high bandwidth networks. This is the trend of future research in pollutant transport simulation and prediction.

The urban-local scale can include geometry extending several miles, and large clusters of buildings. The flow and dispersion around buildings has been investigated for many years, especially in the fields of wind and environmental engineering. The detailed descriptions of the flow behavior around bluff bodies were done during the Eighties. The effort done in those years was based mostly on experimental results. In many cases the description was qualitative and/or led to empirical relations between physical variables of the problem(such as average wind inflow, density, temperature, etc.) and characteristic measurements of the object. A notable experimental study has been that of Robins and Castro[1][2], in which the authors have simulated an atmospheric boundary layer flow over a model building within a wind-tunnel. The arrival of supercomputers brought the possibility to attack the problem from a completely different perspective, one that can complement the experimental procedures. Various authors have conducted numerical studies of flows around either buildings or surface mounted cubes (see for example works by Cheatham, Cybyk and Boris[3], Camelli and Lohner [4], Dawson, Stock and Lamb[5], Murakami and Mochida[6], Paterson and Apelt[7] and Scanlon[8]).

From this work, it is noted that air flow in the vicinity of large buildings is extremely complicated in nature, containing strong pressure gradients, acute streamline curvature, separation, reattachment, high turbulence intensities and recirculation zones. Such flows thus represent a significant challenge to any numerical model. The experiment of Robins and Castro[2] also considered the release of neutrally buoyant pollutant from the center of the roof of the model cube: pressure coefficients, velocities, turbulence parameters, and species concentrations were obtained. This work has provided the basis for much numerical comparison work in recent years.

Delaunay[9] discussed experimental simulations of gas transport around a rectangular-shaped building. Numerical simulations of the experiments was also conducted by solving the Reynold-Averaged Navier-Stokes(RANS) equations, and the simulation result was compared with the available wind tunnel measurements. In this study, the flow around the building was computed using the *K-epsilon* turbulence model. An eddy viscosity based on the *K-epsilon* turbulence model was used for calculating pollutant transport.

More studies have been conducted on pollutant transport around buildings. Selvam[10] simulated the pollutant transport using the finite element method. Again, the *K-epsilon* turbulence model was used for the flow simulation. Pollutant concentrations around the building were computed using both upwind and Streamline Upwind Petro-Galerkine(SUPG) finite elements. The SUPG scheme produces concentration trends much closer to the wind tunnel measurements than the upwind procedure.

Cheatham, Cybyk and Boris [3], investigated flow over a surface-mounted cube, using the computational model FAST3D. Computational results are compared with wind tunnel and water tank data of a number of authors (see also Castro and Robins[1] [2]). The computed flow around a surface-mounted cube is sensitive to both the form of the upstream velocity profile and the conditions at the baseplate surface, and time-dependent effects are crucial to the accumulation of accurate flow field statistics and thus a correct understanding of flow and dispersion patterns.

Camelli and Lohner[4] performed a series of simulations in order to test the accuracy of FEFLO, a general-purpose flow solver. One of these cases was extracted from their evaluation of modeling uncertainty and further discussed [4]. In their work, they focused on the urban-local scale. The geometry is a L shape building in a plane terrain. The problem was solved with two approaches: first, all viscous effects were ignored, i.e. the Euler equations were solved. Secondly, the Navier-Stokes equations (RANS) with a Smagorinsky turbulence model were solved. The results obtained from both approaches were in very close agreement with the published wind tunnel data for the case selected.

Scanlon[8] attempts to model the characteristics of an atmospheric boundary layer type flow over a building which has a pollutant source at its roof center. Two different numerical schemes for the dispersion of convection are employed: the second order accurate QUICK scheme and the first order HYBRID-UPWIND scheme. For the velocity field the results appear to be in quite close agreement with experimental data.

The above publications are just a small sample of such studies in related topics. Very important and useful data from both experimental and/or numerical research on such flows has been accumulated. However, most of these studies only focus on a cube or a single building. No consideration has been given to the study of pollutant transport and dispersion around clusters of buildings. And there is no effort in considering even basic chemical reactions involving pollutants present in these flows.

Besides the above publications from academic institutions, some commercial software has been developed by software companies focusing on environmental engineering. This software is now widely used in Defense, Chemical industry, Petroleum industry, and Environmental Impact Assessment. Fluidyn-Panache[11] is a computer code for numerical simulation of pollutant dispersion into the atmosphere in short or medium range. It uses Computational Fluid Dynamics tools in a finite-volume-based approach to solve the differential equations governing mass, momentum, and energy transfer. Another successful software is SLAB View[12], which is an integrated modeling environment for toxic gas release simulation. SLAB View comes with an extensive database of toxic materials. Both of them provide full-featured processing: intuitive data input, model run, and postprocessing with automatic gridding and contour plotting of results. However, they are not suitable for either large geometry or problems with chemical reactions.

Recently, the wide availability of high performance computing resources at relatively low cost has allowed scientist to perform increasingly more complex simulations, which in turn push the development of advanced simulation software. In his dissertation, Dr. Luke proposed the Loci platform, a rule based specification system for Computational Fluid Dynamics(CFD), to eliminate the "accidental" complexity encountered in numerical simulation software development[13]. In this study, a series of simulations for different scenarios was performed by using CHEM, a finite-rate chemical reactive flow solver based on Loci. This is also the first application of Loci to large scale problems. Two different geometries were employed in this study. One is an artificial city model, a model city block, the other is an actual district of the city of New Orleans.

In this thesis, we focused our study on the accidental chemical pollutant release in densely populated urban areas. A model city block includes one block of buildings, and was developed as the initial geometry to test the validity of the approach. Later, a grid for the city of New Orleans was developed, and a part of the this grid was modified as the geometry for the simulations in this study. This grid contains several clusters of buildings with complex shapes. Such a topology, even with the coarsest grid description, is large enough to challenge the accuracy and efficiency of any CFD software. Although the speed of CPU doubles every 18 months and the cost of a supercomputer class workstation cluster has been dropping radically, no one has the access to unlimited computing resources. Therefore, we had to make some compromises between timely predictions and the accuracy of the results.

In the following, the governing equations will be introduced in Chapter II, and we will present the numerical formulation and the development of the numerical algorithm in Chapter III. At last some results will be presented in Chapter IV, with detailed analysis. In the last chapter we will show some preliminary conclusions, and talk about the limitation of the current research, and outline future work.

CHAPTER II

GOVERNING EQUATIONS

The governing equations for chemically reactive flows represent the conservation of physical properties such as species mass, momentum and energy. When they are expressed for a control volume, an integrodifferential form of the equations can be established. On the other hand, writing these equations for a point in the flow field will result in a nonlinear differential system. In this chapter, the governing equations in integral form will be introduced, which are suitable for discretization using finite volumes. The numerical formulation will be discussed in chapter III.

The governing equations are expressed in integral form for flow in an arbitrary volume V, closed by a boundary S, as follows:

$$\frac{\partial}{\partial t}\int_V \mathbf{Q}dV + \oint_S (\mathbf{S} - \mathbf{S}_v) \cdot \mathbf{n}dS = \int_V \mathbf{W}dV, \tag{2.1}$$

where $\mathbf{Q}$ is the vector of conserved variables, $\mathbf{W}$ is the vector of source terms, and $\mathbf{S}$ and $\mathbf{S}_v$ are the inviscid and viscous flux vectors, respectively. The unit vector $\mathbf{n}$ is normal to the infinitesimal area dS and points outwards.

The vectors $\mathbf{Q}$ and $\mathbf{W}$ in the previous equation can be expressed as:

$$\mathbf{Q} = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_s \\ \vdots \\ \rho_{NS} \\ \rho u \\ \rho v \\ \rho w \\ \rho e_0 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \dot{w}_1 \\ \vdots \\ \dot{w}_s \\ \vdots \\ \dot{w}_{NS} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{2.2}$$

Where ρ, NS, u, v, w and all other symbols can be found in the nomenclature. Utilizing a Cartesian frame of reference (x, y, z) whose unit vectors are $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$, respectively, the flux vectors $\mathbf{S}$ and $\mathbf{S}_v$ can be written as:

$$\mathbf{S} = \begin{bmatrix} \rho_1 \mathbf{u} \\ \vdots \\ \rho_s \mathbf{u} \\ \vdots \\ \rho_{NS} \mathbf{u} \\ \rho u \mathbf{u} + p\mathbf{i} \\ \rho v \mathbf{u} + p\mathbf{j} \\ \rho w \mathbf{u} + p\mathbf{k} \\ \rho e_0 \mathbf{u} + p\mathbf{u} \end{bmatrix}, \quad \mathbf{S}_v = \begin{bmatrix} -\rho_1 \mathbf{V}_1 \\ \vdots \\ -\rho_s \mathbf{V}_s \\ \vdots \\ -\rho_{NS} \mathbf{V}_{NS} \\ \tau_{xx}\mathbf{i} + \tau_{xy}\mathbf{j} + \tau_{xz}\mathbf{k} \\ \tau_{yx}\mathbf{i} + \tau_{yy}\mathbf{j} + \tau_{yz}\mathbf{k} \\ \tau_{zx}\mathbf{i} + \tau_{zy}\mathbf{j} + \tau_{zz}\mathbf{k} \\ \Theta \end{bmatrix}, \tag{2.3}$$

where

$$\begin{aligned}\Theta &= (\tau_{xx}u + \tau_{yx}v + \tau_{zx}w)\mathbf{i} + (\tau_{xy}u + \tau_{yy}v + \tau_{zy}w)\mathbf{j} \\ &+(\tau_{xz}u + \tau_{yz}v + \tau_{zz}w)\mathbf{k} - \mathbf{q} - \sum_{s=1}^{N} \rho_s h_s \mathbf{V}_s.\end{aligned} \tag{2.4}$$

In the above formulas, the mass-averaged velocity $\mathbf{u}$ for the mixture has been utilized, whose Cartesian components are (u, v, w) respectively, all other symbols can be found in the normenclature.

The first NS equations are species continuity equations, which relate the time rate of change in species density to convective and diffusive transport and to the creation or destruction of the species due to chemical reactions. The next three equations are momentum equations, and the last one is engery equation.

In this study, pressure is determined by Dalton's law, which states that the pressure of a gas mixture is the sum of partial pressures of each individual species, and each species behaves as a thermally perfect gas. Thus the mixture pressure can be modeled as:

$$p = \sum_{s=1}^{NS} \rho_s R_s T, \tag{2.5}$$

where R_s is the species gas constant and T is the mixture temperature. The latter is obtained through the caloric equation of state, with the vibrational energy contribution modeled by using a simple harmonic oscillator formula [14]:

$$e_s = n_s R_s T + \sum_{v=1}^{NVT_s} \left(\frac{R_s \theta_{v,s}}{e^{\theta_{v,s}/T} - 1} \right) + h_{f\,s}, \tag{2.6}$$

The translational and rotational contributions to the internal energy are included by using appropriate values for the constant n_s. The internal energy of a mixture is computed as the sum of species internal energy,e_s:

$$e_{internal} = \sum_{s=1}^{NS} Y_s e_s. \tag{2.7}$$

where Y_s is the mass fraction of species s. Then the caloric equation of state can be written as follows:

$$e_0 = e_{internal} + \frac{(u^2 + v^2 + w^2)}{2}. \tag{2.8}$$

When we study chemically reactive flows, a classification of the relevant physical phenomena is based on the time available for chemical reactions. If such times are very large compared to the time for fluid transport, then reactions have almost no time to occur, and a frozen flow can be assumed. On the other hand, when the available time for chemical reactions is very small compared to the fluid transport time, then the reactions have enough time to reach their local chemical equilibrium. However, typically, at least for a portion of the flow field, the reaction time is comparable to the fluid transport time. This is the case of finite-rate chemistry, when it becomes necessary to simulate the actual kinetic behavior of the chemical system.

For a system of NS species and NR chemical reactions among these species, each reaction can be represented as:

$$\begin{aligned} \nu'_{1,r} X_1 + \cdots + \nu'_{s,r} X_s + \cdots + \nu_{NS,r} X_{NS} &\leftrightarrows \\ \nu''_{1,r} X_1 + \cdots + \nu''_{s,r} X_s + \cdots + \nu''_{NS,r} X_{NS}&, \\ r = 1, \cdots, NR& \end{aligned} \tag{2.9}$$

where $\nu'_{s,r}$ and $\nu''_{s,r}$ are the stoichiometric coefficients of the reactants and products of species s in the rth reaction, and X_s represents the species s. The production rate for species s, $\dot{w}_s$, can be represented as:

$$\begin{aligned}\dot{w}_s = \left(\frac{d\rho_s}{dt}\right) = \mathcal{M}_s \sum_{r=1}^{\mathbf{NR}} (\nu_{s,r}^{''} - \nu_{s,r}^{'}) \times \\ \left[k_{f,r} \prod_{l=1}^{\mathbf{NS}} \left(\frac{\rho_l}{\mathcal{M}_l}\right)^{\nu_{l,r}^{'}} - k_{b,r} \prod_{l=1}^{\mathbf{NS}} \left(\frac{\rho_l}{\mathcal{M}_l}\right)^{\nu_{l,r}^{''}} \right],\end{aligned} \tag{2.10}$$

where $\mathcal{M}_l$ is the atomic mass of species l. In the above, $k_{f,r}$, $k_{b,r}$ represents forward and backward reaction rates, respectively. The forward reaction rates are evaluated by Arrhenius curve fits:

$$k_{f,r}(T) = CT^{\eta} e^{-\theta/T}, \tag{2.11}$$

and the backward reaction rates are obtained by using the following relation:

$$k_{b,r} = \frac{k_{f,r}}{K_{e,r}}, \tag{2.12}$$

where $K_{e,r}$ is the equilibrium constant, which is determined from thermodynamics [13].

Generally only Newtonian fluids are considered, where there is a linear relationship between stress and rate of deformation. Moreover, bulk viscosity effects are neglected. Under these assumptions, the stress-tensor components are expressed in Cartesian form as:

$$\tau_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) - \frac{2}{3}\mu \nabla \cdot \mathbf{u} \delta_{ij}, \quad i = 1, 2, 3, \quad j = 1, 2, 3 \tag{2.13}$$

where μ is the dynamic viscosity coefficients and δ_{ij} is the Kronecker delta. The heat flux vector $\mathbf{q}$ is modeled using Fourier's law:

$$\mathbf{q} = -\lambda \nabla T, \tag{2.14}$$

where λ is the conduction coefficient. The viscosity coefficient μ and thermal conductivity λ are usually evaluated in two steps: first, one determines the transport properties for each species; then a mixing rule is invoked in order to obtain mixture values.

Two models are applied to compute species transport properties. At temperature lower than 1000K, Sutherland's law is used:

$$t_i = T^{3/2}\frac{F_{t,i}}{T + G_{t,i}}, \tag{2.15}$$

where t_i stands for either μ_i or λ_i, and $F_{t,i}$, $G_{t,i}$ are constants determined empirically. At temperature higher than 1000K, a more accurate model based on curve fit tabulation proposed by Gupta [15] can be utilized.

Once the transport properties for individual species are obtained, Wilke's mixture rule is applied to determine transport values for the mixture, as follows:

$$t = \sum_{i=1}^{NS} W_i t_i, \tag{2.16}$$

where t denotes transport properties for the mixture (either μ or λ) and the weighting function W_i is given by

$$W_i = \frac{X_i}{\sum_{j=1}^{NS} X_j \eta_{i,j}}, \tag{2.17}$$

where $X_i = \rho_i/\mathcal{M}_i$ is the species concentration and the coefficient $\eta_{i,j}$ is given by

$$\eta_{i,j} = \frac{1}{\sqrt{8}}\Big(1 + \frac{\mathcal{M}_i}{\mathcal{M}_j}\Big)^{-1/2}\Big[1 + \sqrt{\frac{\mu_i}{\mu_j}}\Big(\frac{\mathcal{M}_i}{\mathcal{M}_j}\Big)^{1/4}\Big]^2. \tag{2.18}$$

Fike's law of diffusion is employed to model the species diffusion velocity $\mathbf{V}_s$:

$$\rho_s \mathbf{V}_s = -\rho D_s \nabla(\frac{\rho_s}{\rho}), \qquad s = 1, \cdots, NS, \tag{2.19}$$

where D_s is the diffusion coefficient for species s. The simplest way of modeling the diffusion coefficients is obtained by assuming constant Lewis number, L_e, and Prandtl number, P_r:

$$D_s = D = \frac{L_e}{\rho} \frac{\mu}{P_r}, \qquad s = 1, \cdots, NS, \tag{2.20}$$

where only a sigle global diffusion coefficient is used. Multi-component effects can be partially taken into account by more complex choices of diffusion coefficients(see also [16]). In this current study, only the sigle global diffusion coefficient has been used.

CHAPTER III

NUMERICAL FORMULATION

In order to perform a computer simulation, the governing equations in the previous chapter have to be discretized, that is reduce to a set of algebraic equations that can be solved by a computer. There are several discretization techniques, such as finite-volume, finite-difference, and spectral techniques that have been widely used in the scientific literature. By taking advantage of the integral form of the governing equations, we discretized the governing equations using the finite-volume technique, solving for the unknown volume averages of conserved variables in some small, but finite, control volume. The current formulation is designed to be applied on hybrid grids(see also[17]). The cell averaged flow variables are calculated as an integral average:

$$Q(C_i) = \frac{1}{V_i} \int_{V_i} Q(x, y, z) dV, \tag{3.1}$$

where C_i is the cell number and V_i is the cell volume:

$$V_i = \int_{V_i} dV. \tag{3.2}$$

The governing Equation 2.1 for a cell with volume V_i can be written as:

$$\frac{\partial}{\partial t} \int_{V_i} Q_i dV + \oint_S \mathbf{F} \cdot \mathbf{n} dS = \int_{V_i} \mathbf{W} dV. \tag{3.3}$$

where $\mathbf{F}$ represents $\mathbf{S} - \mathbf{S}_v$ for simplicity. The numerical integration of the surface integral in the governing equations is computed by summing the contributions of each of the k_i faces of cell C_i, assuming that the flux can be approximated by a function, $\hat{F}(Q_l, Q_r)$, of conservative values to the left and right of the face. Then, the numerical integration of $\mathbf{F}$ is given by:

$$\oint_S \mathbf{F} \cdot \mathbf{n} dS = \sum_{j=1}^{k_i} \mathbf{F}_{ij} \cdot \mathbf{n}_j A_j \approx \sum_{j=1}^{k_i} \hat{\mathbf{F}}(Q_l, Q_r) \cdot \mathbf{n}_j A_j. \tag{3.4}$$

where the area of the face, A_j, is defined as

$$A_j = \oint_S dS. \tag{3.5}$$

Thus a spatially discretized form of Equation 3.3 is:

$$\frac{\partial}{\partial t} \int_{V_i} Q_i dV + \sum_{j=1}^{k_i} \mathbf{F}_{ij} \cdot \mathbf{n}_j A_j = \int_{V_i} \mathbf{W} dV, \tag{3.6}$$

where the indices i and j denote the cell and face number, respectively.

Using volume averages (see Equation 3.1 and Equation 3.2) Equation 3.6 can be numerically approximated by the following equation:

$$\frac{d}{dt} [V_i Q_i] + \sum_{j=1}^{k_i} \mathbf{F}_{ij} \cdot \mathbf{n}_j A_j = V_i W_i. \tag{3.7}$$

3.1 Space discretization

A second-order scheme is used in the current study. The primitive flow variables $\mathbf{q} = [\rho_1, \ldots, \rho_{NS}, u, v, w, p]^T$ are assumed to be distributed linearly within each cell, and the linear distribution is reconstructed from the cell-averaged values. During the solution process, the flow variables at each cell face are extrapolated from the cell-averaged values using a linear reconstruction procedure [18].

The piecewise linear reconstruction of the primitive variables is done by Taylor's series expansion:

$$\mathbf{q}(x,y,z) = \mathbf{q}(x_i,y_i,z_i) + \nabla\mathbf{q}(x_i,y_i,z_i)\cdot\delta\mathbf{r} + O((\delta\mathbf{r})^2), \tag{3.8}$$

In the above equation, $\nabla\mathbf{q}$ is the gradient of the primitive variables $\mathbf{q}$, and $\delta\mathbf{r}$ is the vector from the center of the cell (x_i, y_i, z_i) to the desired point (x, y, z). The gradient of $\mathbf{q}$ at the cell center is estimated using a least-squares technique that has been described elsewhere [19]. To remove non-physical values near regions of high solution gradients and preserve monotonicity, a limiter function has to be applied to $\nabla\mathbf{q}$. The current study employs Venkatakrishna's limiter [20], which is expressed by:

$$\phi_{ij} = \frac{1}{\Delta_-}\left[\frac{(\Delta_+^2+\epsilon^2)\Delta_- + 2\Delta_-^2\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_+\Delta_- + \epsilon^2}\right], \tag{3.9}$$

and

$$\phi_i = min(\phi_{ij}), \tag{3.10}$$

where ϵ^2 is taken as $(K\Delta x)^3$, Δx represents the average edge size of the cells, and K is typically set in the range form 0.1 to 0.3. The other variables appearing in Equation 3.9 are defined as follows:

$$\Delta_- = q_{ij} - q_i,$$

$$\Delta_+ = \begin{cases} q_i^{max} - q_i & : \quad q_{ij} > q_i \\ q_i^{min} - q_i & : \quad q_{ij} < q_i \end{cases}, \tag{3.11}$$

where q_i^{min} and q_i^{max} are the minimum and maximum cell values respectively, among the cells adjacent to the ith cell, including the ith cell itself. The face value of primitive variables can be then written as:

$$\mathbf{q}(x, y, z) = \mathbf{q}(x_i, y_i, z_i) + \phi_i \nabla \mathbf{q}(x_i, y_i, z_i) \cdot \delta \mathbf{r}. \tag{3.12}$$

Right and left values of conserved variables (Q_l, Q_r) can then be evaluated from the appropriate values of $\mathbf{q}$.

3.2 Roe flux

The discretization of the inviscid fluxes can be accomplished by means of flux-split techniques. One of the most popular and accurate flux-split schemes is the flux-difference splitting technique developed by Roe [21]. The Roe flux calculation is based on the exact solution of an approximate Riemann problem, where an arbitrary discontinuity is supposed to exist at the cell surface between the left and the right state, and an approximate solution is expressed in terms of waves propagations(upstream and downstream). This technique differs from the popular flux-vector splitting schemes in that the former reconstructs the flux vector from upstream and downstream contributions, instead of splitting the flux vector. The form suitable for chemical reactive flow problems has been developed by Cinnella[22]. The related literature and formulations can be found in [23].

3.3 Time integration

If the control volumes are fixed and non-deforming , the cell governing equation can be expressed as:

$$\frac{d}{dt} Q_i = R_i, \tag{3.13}$$

where the steady-state residual R_i is given by the expression:

$$R_i = \frac{1}{V_i}\left[V_i W_i - \sum_{j=1}^{k_i} \mathbf{F}_{ij} \cdot \mathbf{n}_j A_j\right]. \tag{3.14}$$

There are two categories of schemes that are commonly used in numerical time integration: explicit methods and implicit schemes. The latter utilize linearized estimations of fluxes and source terms at new time steps when advancing solution in time, whereas explicit methods evaluates fluxes and source term at the previous time step prior to moving to the next step.

3.4 Explicit Method

An m-stage explicit Runge-Kutta time integration method has been used for explicit time integrations. It is second-order accurate in time. The major advantage of this scheme is its computational inexpensiveness. However, all explicit schemes have a major drawback: they become extremely inefficient for stiff equations, which is typically the case when chemical reactions are present. In these circumstances, the time step necessary for stability can become orders of magnitude smaller than the value which would be necessary for an efficient resolution of the overall gas-dynamic transient.

The m-stage explicit Runge-Kutta method advances the solution in time as follow:

$$\begin{aligned} Q^0 &= Q^n, \\ Q^1 &= Q^0 + \alpha_1 \Delta t R(Q^0), \\ &\vdots \\ Q^m &= Q^0 + \alpha_m \Delta t R(Q^{m-1}), \\ Q^{n+1} &= Q^m, \end{aligned} \tag{3.15}$$

where the weighting coefficients α_i guarantees second-order time-accuracy when

$$\alpha_i = \frac{1}{m - i + 1}. \tag{3.16}$$

In the above, the superscript n, denotes the conditions at time level n, and Δt is the time step.

3.5 Implicit method

Implicit schemes are very popular, especially for steady state problems. Several methods can be used in the numerical integration of Equation 3.13. In particular, a family of two-parameter schemes can be used for the implicit time integration, given by the time-discretized equation:

$$\begin{gathered}(1+\psi)\Delta Q_i^n - \psi \Delta Q_i^{n-1} = \\ \Delta t\{(1-\theta)R_i^n(Q_i^n) + \theta R_i^{n+1}(Q_i^{n+1})\},\end{gathered} \tag{3.17}$$

In the above, R_i stands for the residual term for the ith cell, including all source terms and fluxes, n is the current time step, and $\Delta Q_i^n = Q_i^{n+1} - Q_i^n$. In this scheme, θ and ψ form a two parameter family of algorithms. For example, the combination of $\theta = 1$ and $\psi = 0$ gives the implicit backward Euler method for steady state simulations, while setting $\theta = 1$ and $\psi = \frac{1}{2}$ create a second-order three-point backward method for time-accurate simulations.

The rearranged Equation 3.17 can be written as a non-linear system of equations for Q^{n+1}, as follows:

$$\begin{gathered}Q_i^{n+1} - Q_i^n - \\ \frac{\Delta t}{1+\psi}\left[(1-\theta)R_i^n(Q_i^n) + \theta R_i^{n+1}(Q_i^{n+1})\right] - \\ \frac{\psi}{1+\psi}(Q_i^n - Q_i^{n-1}) = \mathcal{L}(Q_i^{n+1}) = 0.\end{gathered} \tag{3.18}$$

The above nonlinear system of equations can be solved by the Newton iterative method as:

$$\mathcal{L}'(Q_i^{n+1,p})(Q_i^{n+1,p+1} - Q_i^{n+1,p}) = -\mathcal{L}(Q_i^{n+1,p}), \tag{3.19}$$

where $p \geq 0$, and the Newton iteration is initialized using the previous time step value ($Q_i^{n+1,p=0} = Q_i^n$). The Jacobian $\mathcal{L}'(Q_i^{n+1,p})$ is expressed as:

$$\mathcal{L}'(Q_i^{n+1,p}) = I - \frac{\theta \Delta t}{1+\psi}\left[\frac{\partial}{\partial Q_i} R_i^{n+1,p}(Q_i)\right]. \tag{3.20}$$

After linearizing the non-linear system of equations using Newton method, the symmetric Gauss-Seidel approach is applied to solve the resulting (linear) system.

3.6 Preconditioning

The flow in this study is a low-speed flow. When flow solvers designed for high-speed flow are used on such a low-speed flow scenario, the performance of the solver will degrade due to the stiffness, which can be measured by the ratio between the maximum and the minimum eigenvalue of the flux Jacobians. To overcome the difficulty of solving nearly incompressible problems by means of compressible flow solvers, a preconditioning technique is applied to the finite rate chemistry equations. This preconditioning method re-scales the eigenvalues of the problem and renders their ratio nonsingular as Mach number goes to zero[24][25].

The difficulty in solving the compressible equations for low Mach flows is attributed to the large disparity between the acoustic wave speed and the fluid speed. The application of preconditioning changes the eigenvalues of the system of compressible flow equations and reduces this disparity. For instance, the time derivatives are multiplied by a matrix that slows the speed of acoustic waves relative to the fluid speed. For implicit method, the Jacobian equation 3.20 will be modified by applying the preconditioning matrix. When explicit method is used, the preconditioning matrix is multiplied before the residual term in the m-stage explicit Runge-Kutta method expression. A detailed description for the modified Roe flux formulation can also be found in [24].

3.7 Solving the Linear System

The linear system that results from solving the non-linear system using Newton method can be written as:

$$Ax = b, \tag{3.21}$$

where

$$\begin{aligned} A &= \mathcal{L}'(Q_i^{n+1,p}), \\ x &= (Q_i^{n+1,p+1} - Q_i^{n+1,p}), \\ b &= -\mathcal{L}(Q_i^{n+1,p}). \end{aligned} \tag{3.22}$$

Matrix A is given by Equation 3.20, and is a typical matrix with sparse structure that is composed of dense sub-blocks. The matrix A is factored into lower, upper and diagonal blocks as:

$$A = L + D + U. \tag{3.23}$$

The symmetric Gauss-Seidel iterative solver works by using a two-pass approach. These two passes, a forward pass and a backward pass, are a consequence of the solution of following equations:

$$\begin{aligned} (L + D)x^{*i+1} + Ux^i &= b, \\ Lx^{*i+1} + (D + U)x^{i+1} &= b. \end{aligned} \tag{3.24}$$

where x^{*i+1} is the result of the forward pass of the symmetric Gauss-Seidel iteration. The iteration is initialized using the first pass of a block Jacobi iterative method:

$$x^0 = D^{-1}b. \tag{3.25}$$

The superscript i, represents a nested subiteration for the Newton iteration. When converged, the subiteration in i will yield the exact solution of the linearized Equation 3.21 (or 3.19) at Newton iteration level p.

3.8 Geometry and grid

In the beginning of this study, an artificial city(a model city block) grid was constructed, including a small cluster of buildings. Most of the buildings in this grid are regular-shaped and arranged in the center of the ground. The dimension of this grid is 2909m by 2658m by 270m(L,W,H). The grid has 59200 triangles and 484917 tetrahedra. The purpose of this grid was to test the validity of the approach selected for this study. Therefore, the size of the grid is not very large, and the geometry is not complex relative to the later grid for the city of New Orleans.

The other geometry studied in this simulation represents a small portion of the downtown area of the city of New Orleans. A three-dimensional grid was first generated for the city of New Orleans, and one of its districts was extracted from the original grid and used for our simulations. The grid was enclosed by ground, upper and surrounding(in/out) surfaces(boundaries). The ground surface lies in the same elevation(in the original grid for the whole city, there is elevation changes among different districts of the city). In the middle of the ground area, there is a cluster of 20 buildings of both regular and irregular shapes. Most of the grid curves were established according to the real dimensions of these buildings(some of these buildings have complex profiles, such as L-shaped building, U-shaped buildings and T-shaped buildings). After establishing the grid lines, engineers constructed the unstructured grid using SolidMesh[26], an unstructured grid generation tool developed at the ERC. The most obvious feature of this grid that distinguishes it from those used in previous studies is the complexity of the shape and the number of the buildings. Due to its nature, even the most coarse description of this geometry will result in a very large grid: this grid has 678893 tetrahedra. Its size itself is very challenging for an efficient simulation with acceptable accuracy. In this larger grid the upper boundary is located at two times the height of the highest building (approximately 200 meters). The length of the grid is about five times the length of the building-block, while grid is six times as wide as the building-block. The dimension of the grid is 3500m by 3000m by 400m(L,W,H). All the surfaces in both grids are defined into four different groups(the surfaces,the wall, the ground and the ceiling), according to their different boundary conditions.

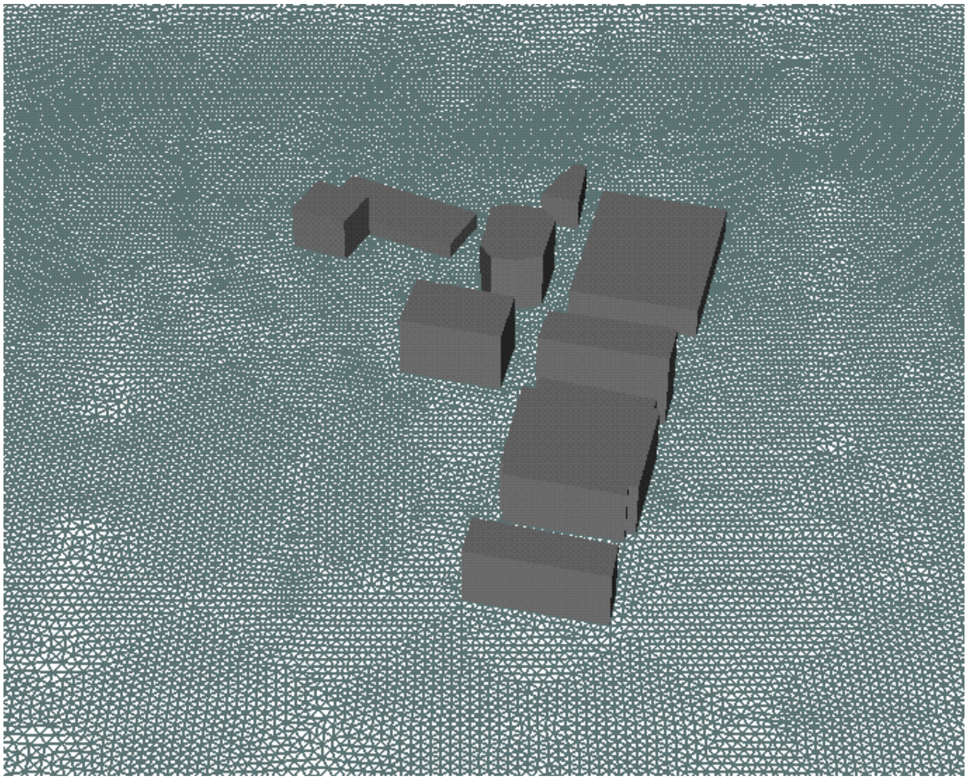

Figure 3.1: A model city block Geometry

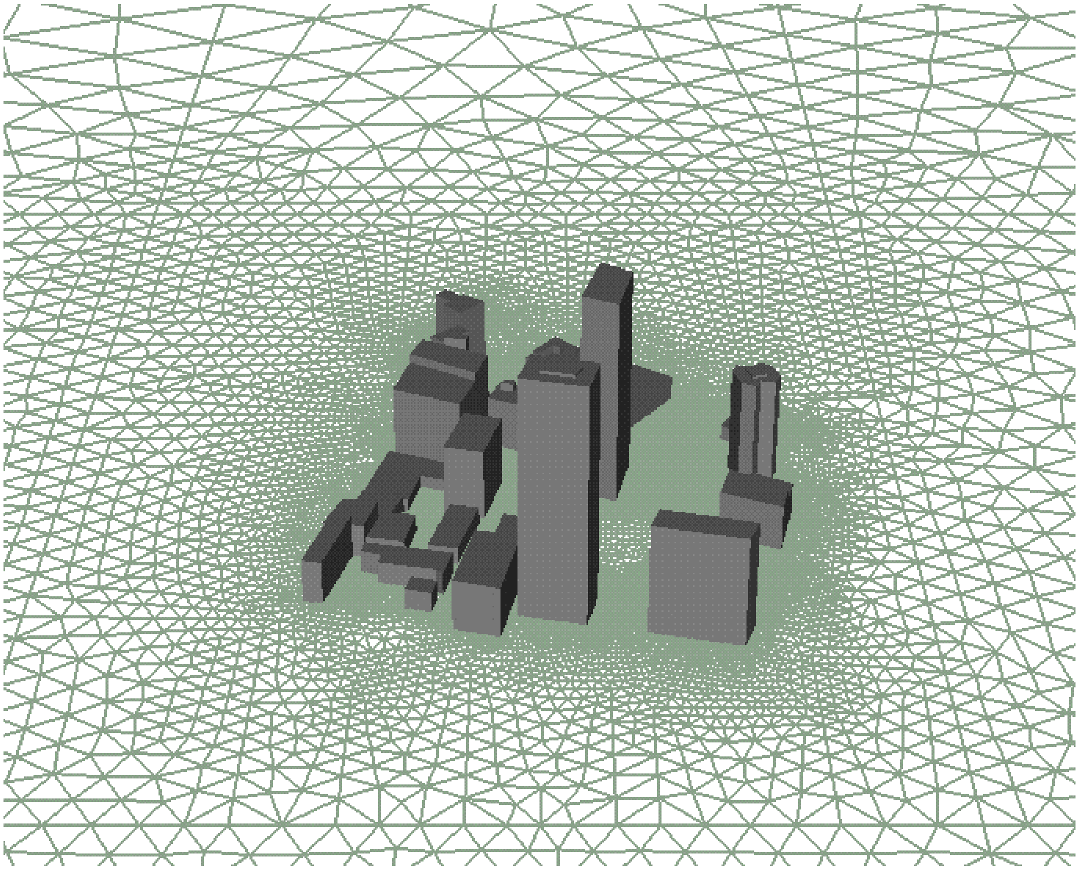

Figure 3.2: New Orleans Geometry

3.9 Loci and CHEM

The code CHEM has been developed to implement the above numerical schemes, and has been recently validated. Before introducing CHEM, it is appropriate to briefly review Loci, a Deductive Framework for Graph-Based applications, which is the framework that incorporates CHEM [13].

Modern CFD software is more and more complex. Typical CFD simulations often involve complex geometries, flexible boundary conditions, and multiple numerical models (such as heat conduction and gas dynamics). This complexity gives us a challenge on how to guarantee the correct coordination of complex component interactions. The Loci system addresses this issue by introducing a deductive framework for the coordination of numerical value classes constructed in C++.

The Loci system is an application framework that seeks to reduce the complexity of assembling large-scale finite-difference or finite-element applications, although it could be applied to many algorithms that are described with respect to a connectivity network or graph. The fundamental element of Loci system is the *rule*, which is used to calculate specific variables. Each rule can be viewed as a subroutine which transforms a set of data to another. For example, the calculation of cell volumes can be interpreted as a transformation from node positions into volumes. An approximation of the inviscid flux using a Riemann solver can be viewed as a transformation from left and right values into an inviscid flux. A simulation is conducted by repeatedly calling these rules.

The Loci system works by providing a fact database where the grid information, grid topology and problem description are stored. Loci also has a rule database, whose various rules are used to calculate useful variables. Now with user input(i.e. the goal of the simulation) the Loci system can automatically generate a scheduling file. This process includes three basic steps: the first step involves creating a dependency graph, connecting the variables stored in the fact database to the goal using the rules in the rule database. Once this graph is produced, it is pruned to only those rules that generate the requested goal, sorted into iteration hierarchies, and reduced to a directed acyclic graph (DAG) by clumping recursive dependency loops. The next step is an existential deduction phase, which determines what attributes can be assigned to which entities. A final optimization pass is then conducted by pruning this schedule. Having successfully generated the schedule file, the solver starts to execute the schedule and finally obtains the solution requested.

The Loci system was used to develop a finite-rate chemical reactive flow solver-CHEM. CHEM is the first application of Loci, and it can solve both structured and unstructured chemical reactive flow problems. Currently there are two versions of CHEM, the shared memory and the distributed memory models. The shared memory CHEM was implemented on a SGI machine,with eight R10000 processors, and 4 gigabytes of memory. The recently developed distributed memory version is implemented on a sixteen nodes cluster, with four Ultrasparc processors and 2 gigabytes of memory on each node. The present calculations are among the very first ones testing and validating the distributed memory version.

CHAPTER IV

RESULTS

In this study, simulations of pollutant release and transport have been implemented on two grids, a model city block and a district of New Orleans. Here we will consider different permutations of initial and boundary conditions: only wind flow(perfect gas); chlorine injected into the air(or nitrogen); ammonia injected into the air(or nitrogen); injection from the ground at different locations; multiple injections. All the cases will be discussed in detail in this chapter. The explicit method introduced in the previous chapter was employed for time integrations in all the cases. All plots were done with FAST[27].

In all the following pictures, the x axis is pointing(from west) to east, z axis is pointing(from south) to north. Both x and z axes are in a plane parallel to the ground boundary. y axis is perpendicular to the ground and pointing to the ceiling. W/E denotes the flow direction from west to east. N/S denotes the flow direction from north to south.

4.1 Wind flow

In the simulation of perfect gas flow on both geometries, the following cases are considered: 1)wind flow from west side to east side, 2)wind flow from north to south. There are four groups of surfaces in each grid: the ceiling, the ground, the wall(outer boundary), the building surface. A non-reflecting(inflow/outflow) boundary condition is applied to the ceiling and the outer boundary, while an impermeable boundary condition is prescribed at the ground and the surfaces(see also [28]). The inlet conditions are given as: p=1atm, T=300K, M=0.01, mixture=[N_2=1.0]. The initial conditions are: p=1atm, T=300K, M=0. A Runge-Kutta method was employed in time integration.

Fig. 4.1 to fig. 4.11 are velocity contour plots for these two geometries. As shown in the New Orleans plots, the velocity u is ranging from -3.9 m/s to 5.9 m/s for cases with wind from West to East, and velocity w ranges from -3.6 m/s to 5.35 m/s for cases with wind from North to South. In the model city block cases, the velocity range is only slightly smaller. Fig. 4.3 shows the contour lines of velocity u in a vertical plane that is parallel to the direction of the main wind. One can clearly see the separations due to the bluff corner of the buildings, especially the one on the highest building. The detail patterns of separation and reattachment are discussed in [29]. In fig. 4.1, due to the sharp changes of the cross section in the area between two buildings(see also fig. 4.2 and 4.9), the flow speed jumps when wind first comes to the block of buildings and passes through it. Unlike the results for the flow around a single obstacle, these contour plots do not show any symmetry. In the model city block geometry, there is one L-shaped building which is similar to the one in [4]. However, because the wind flow direction is not perpendicular to the surface of this building, the resulting contour plots show larger recirculation zones behind this building(see fig. 4.8 and 4.9). For the New Orleans case, there is a high-speed region in a relatively open area inside the building block. This phenomenon is due to the fact that the area behind the two towers allows enough space for reattachment of the flow. The difference of the contour line plots with both winds(N/S or W/E) between the two geometries can be attributed to their geometrical properties. In the model city block, there is only one block and almost all the buildings are located in a line that parallel to the North-South direction. Buildings in New Orleans grid are located unevenly. Most of the buildings are clustered in the center region, with some other buildings scattered far away in the west. These differences account for the different velocity contour plots.

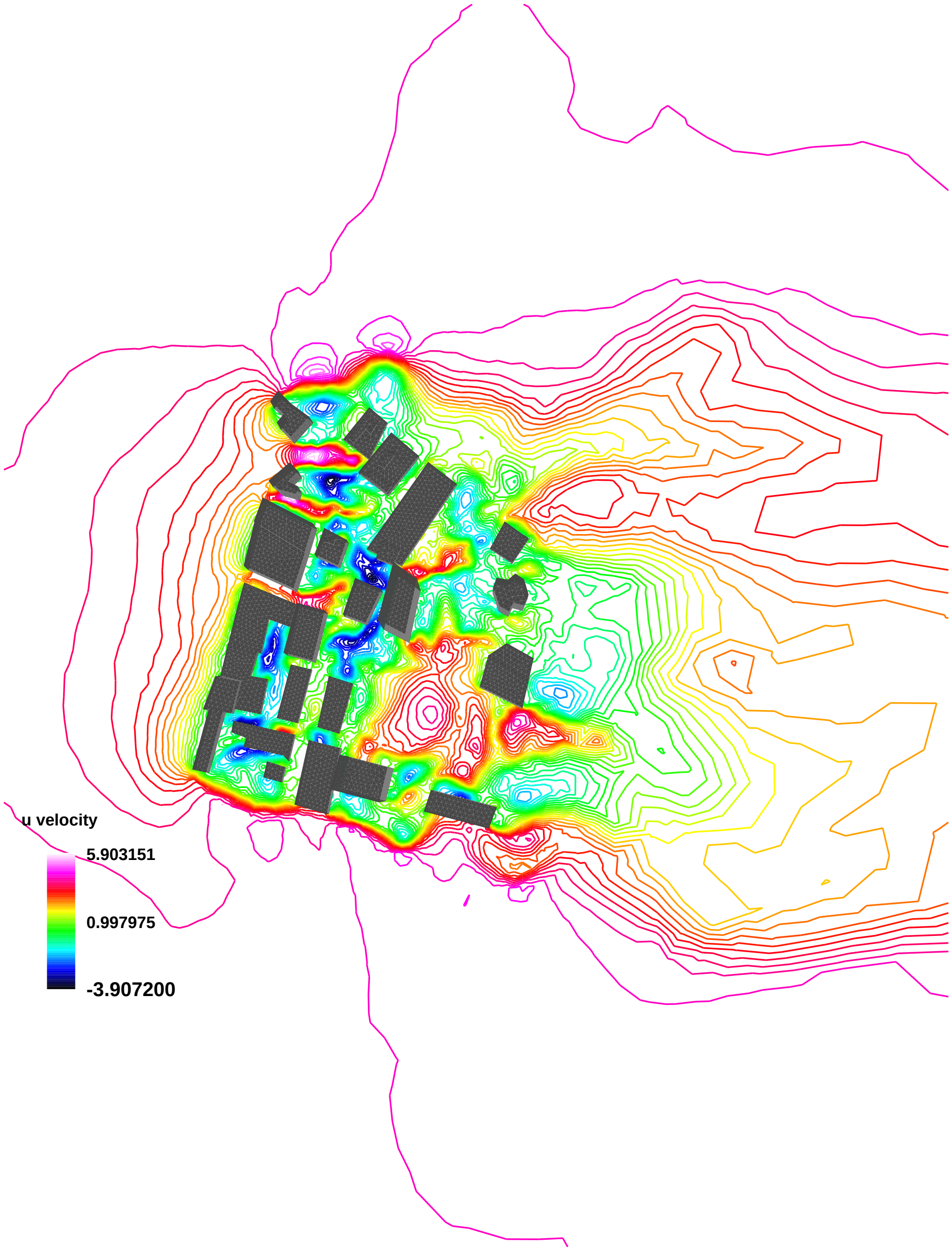

Figure 4.1: Velocity contours for W/E perfect gas flow in New Orleans(top view)

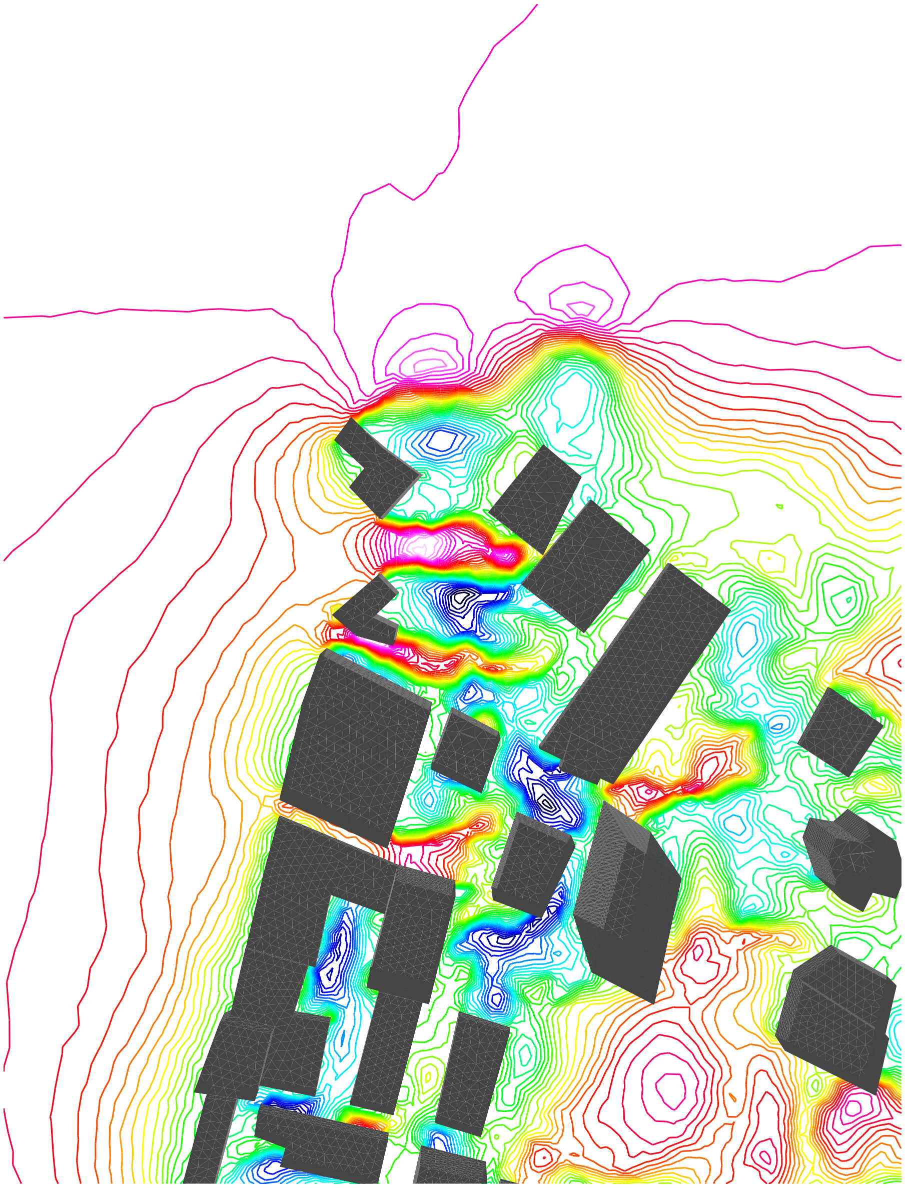

Figure 4.2: Velocity contours for W/E perfect gas flow in New Orleans(enlarged top view). (for legend refer to fig. 4.1)

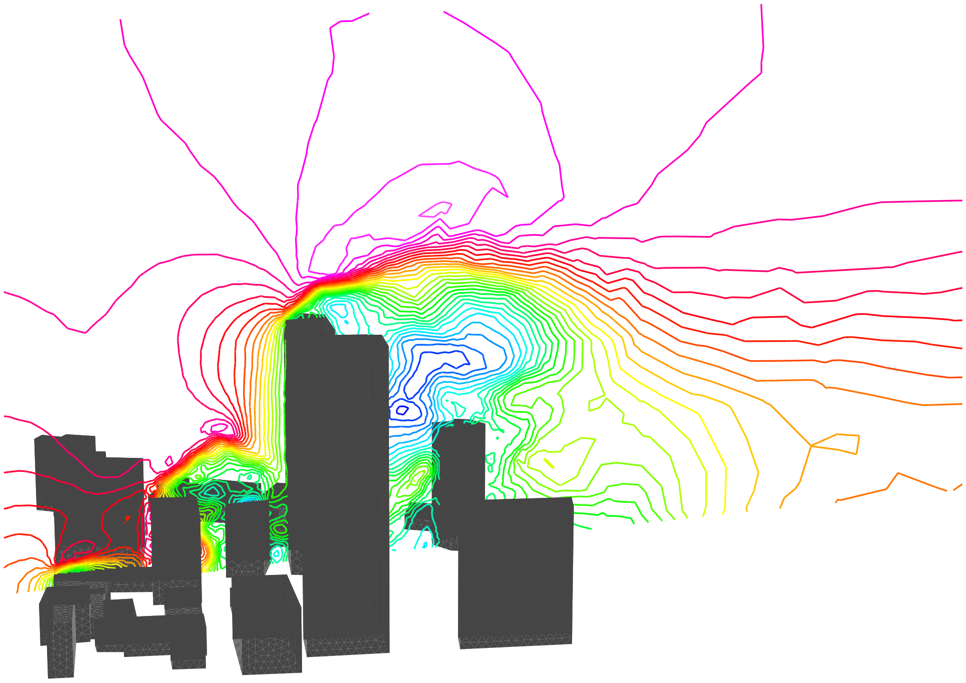

Figure 4.3: Velocity contours in a vertical plane for W/E perfect gas flow in New Orleans(top view). (for legend refer to fig. 4.1)

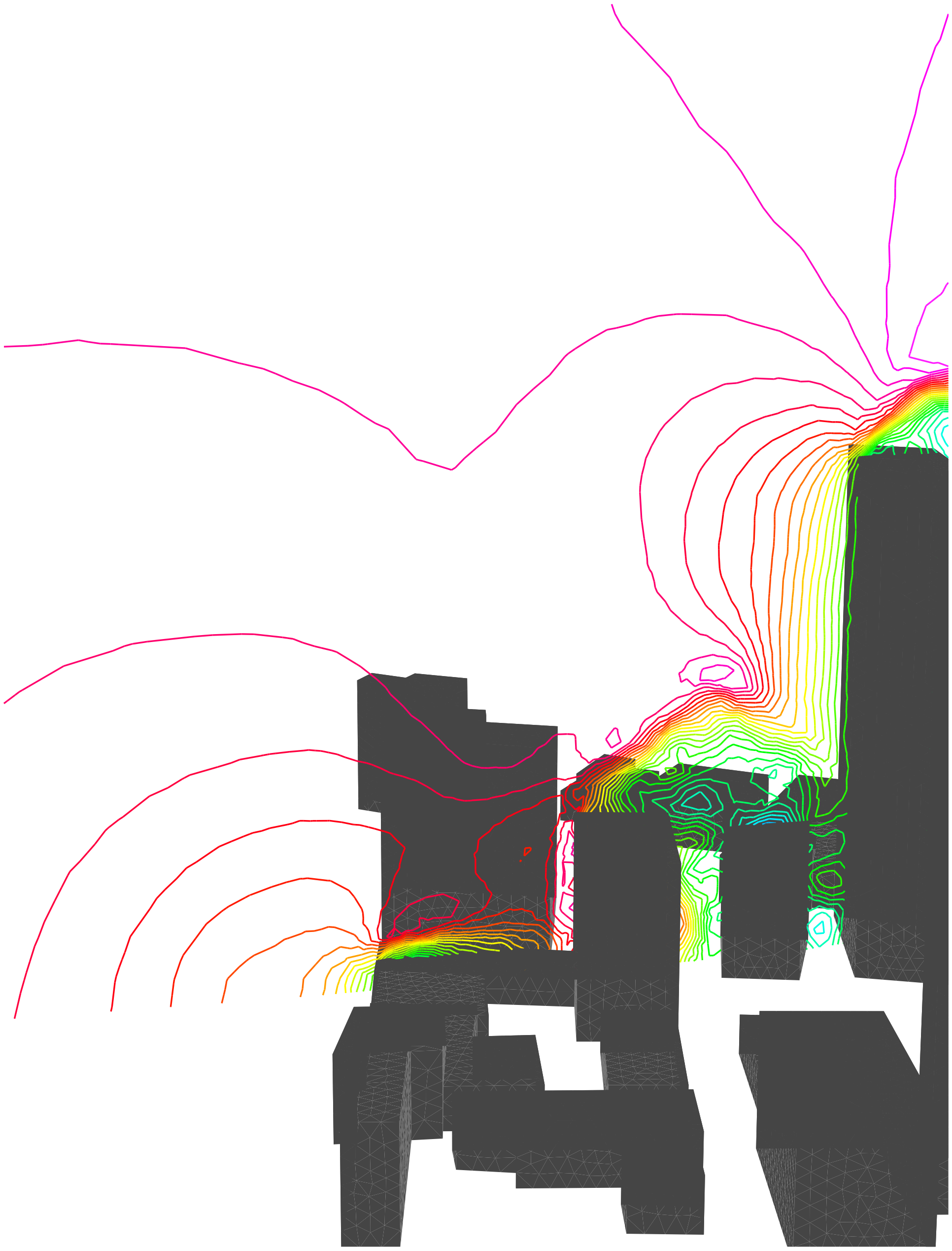

Figure 4.4: Velocity contours in a vertical plane for W/E perfect gas flow in New Orleans(close view). (for legend refer to fig. 4.1)

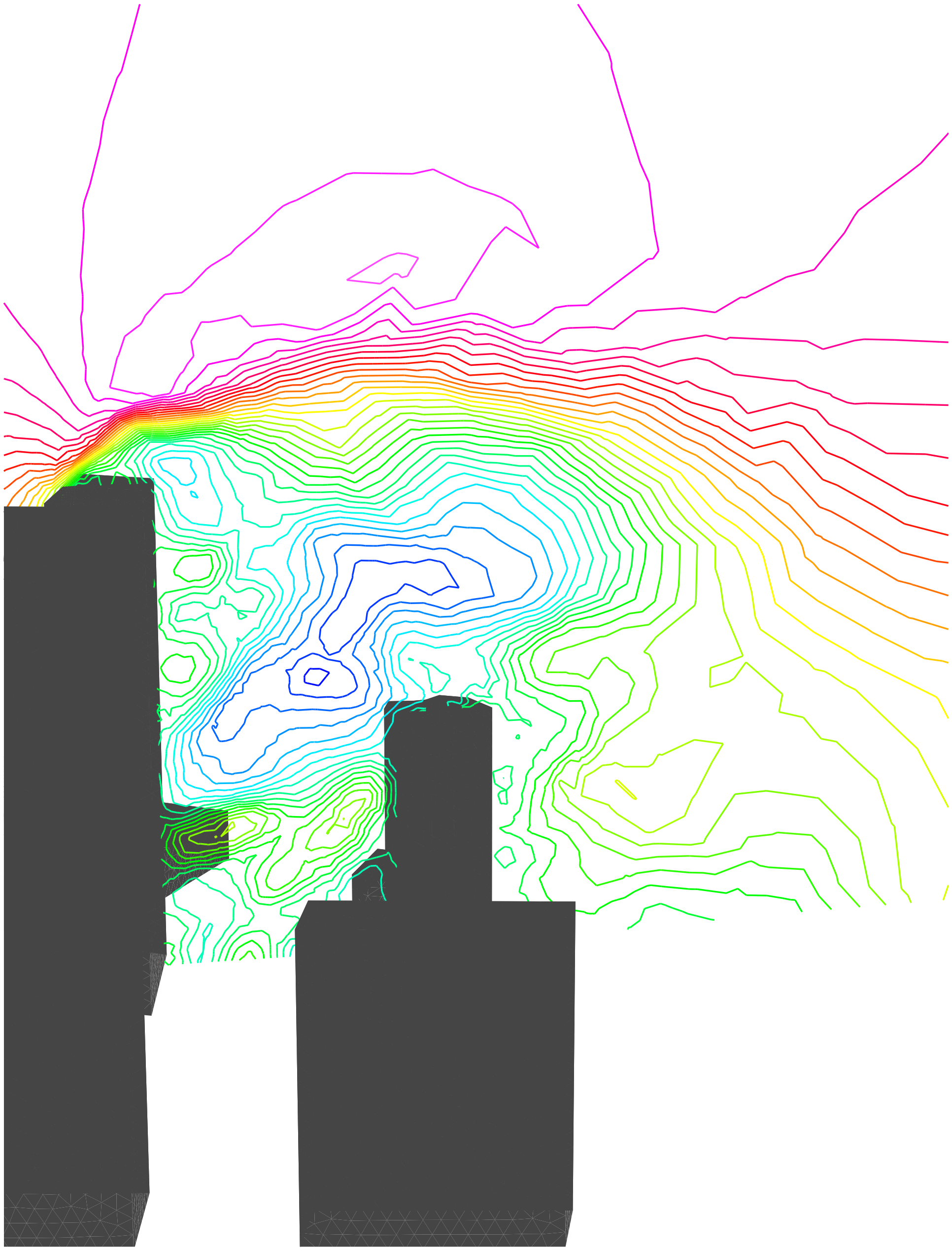

Figure 4.5: Velocity contours in a vertical plane for W/E perfect gas flow in New Orleans(close view). (for legend refer to fig. 4.1)

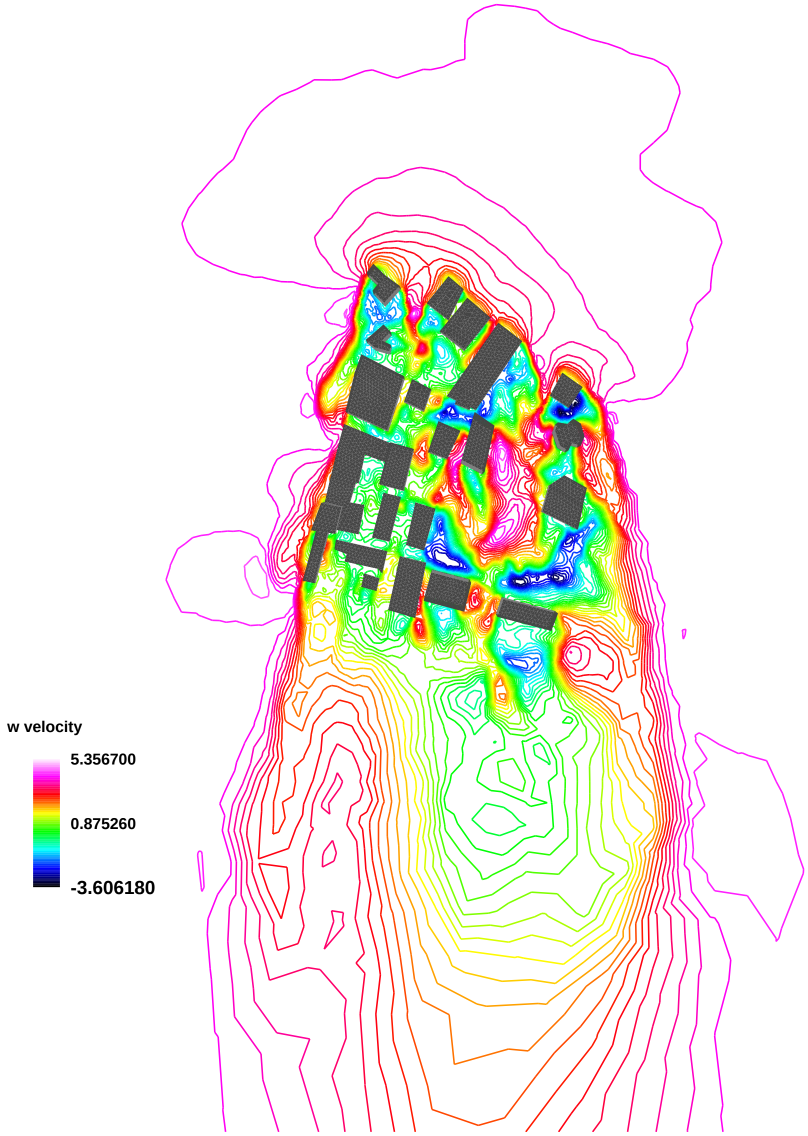

Figure 4.6: Velocity contours for N/S perfect gas flow in New Orleans(top view)

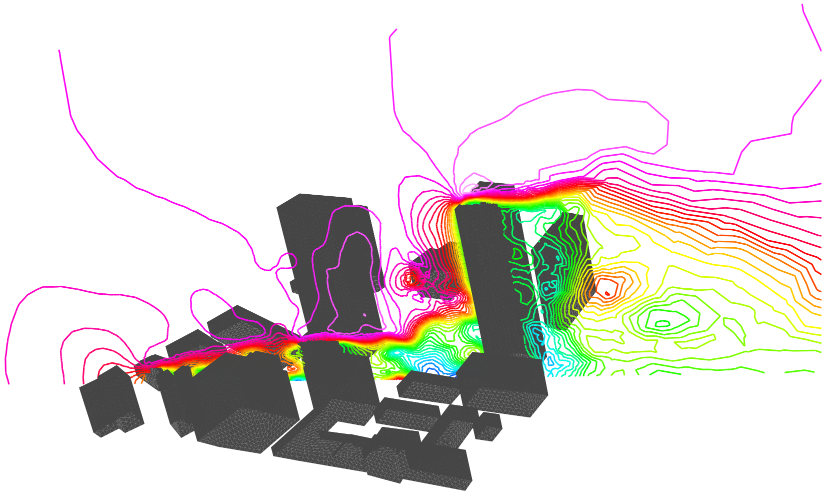

Figure 4.7: Velocity contours in a vertical plane for N/S perfect gas flow in New Orleans. (for legend refer to fig. 4.6)

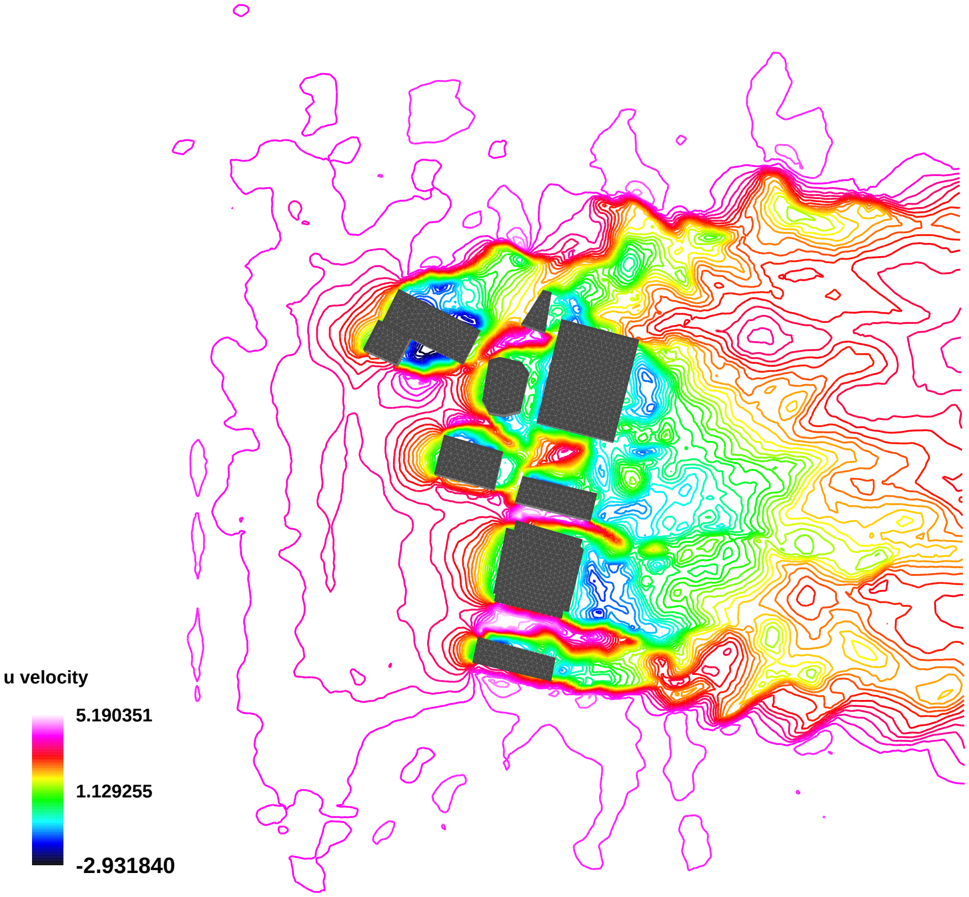

Figure 4.8: Velocity contours for W/E perfect gas flow in a model city block(top view)

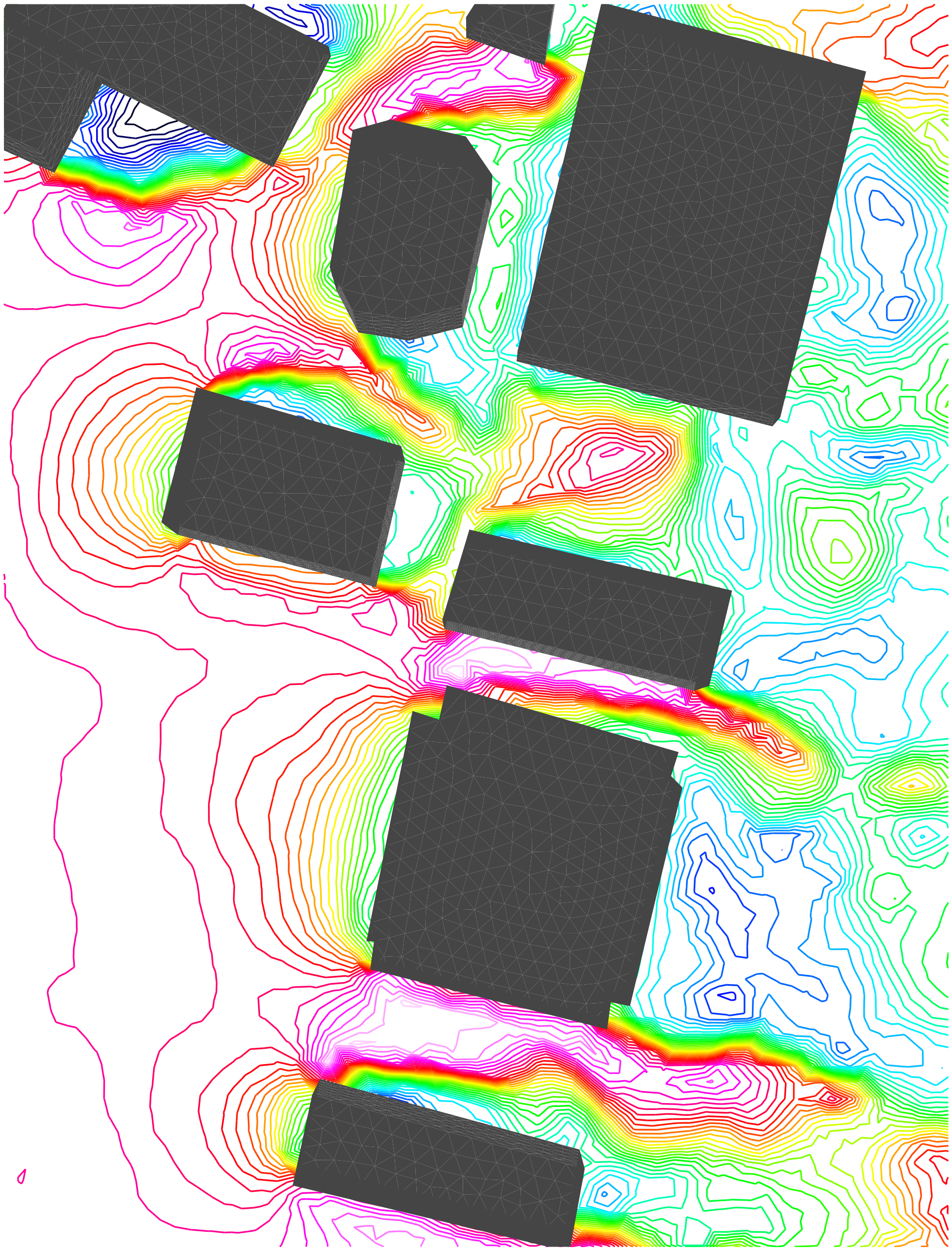

Figure 4.9: Velocity contours for W/E perfect gas flow in a model city block(close view). (for legend refer to fig. 4.8)

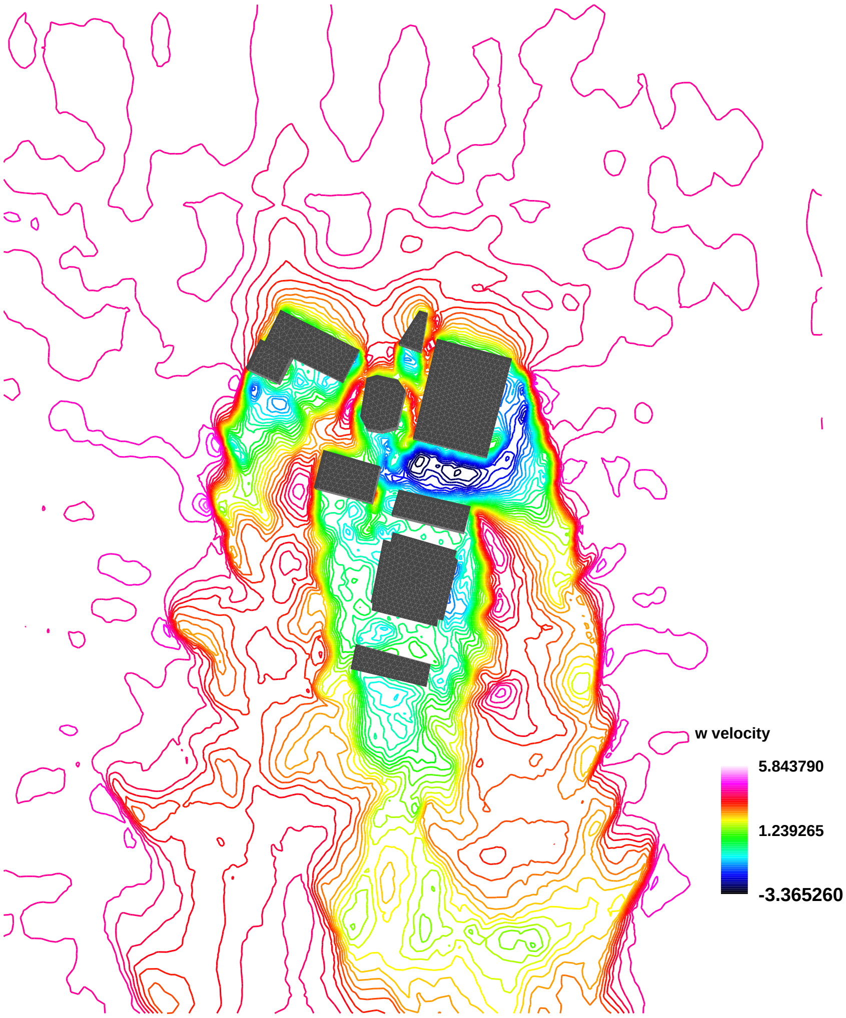

Figure 4.10: Velocity contours for N/S perfect gas flow in a model city block(top view)

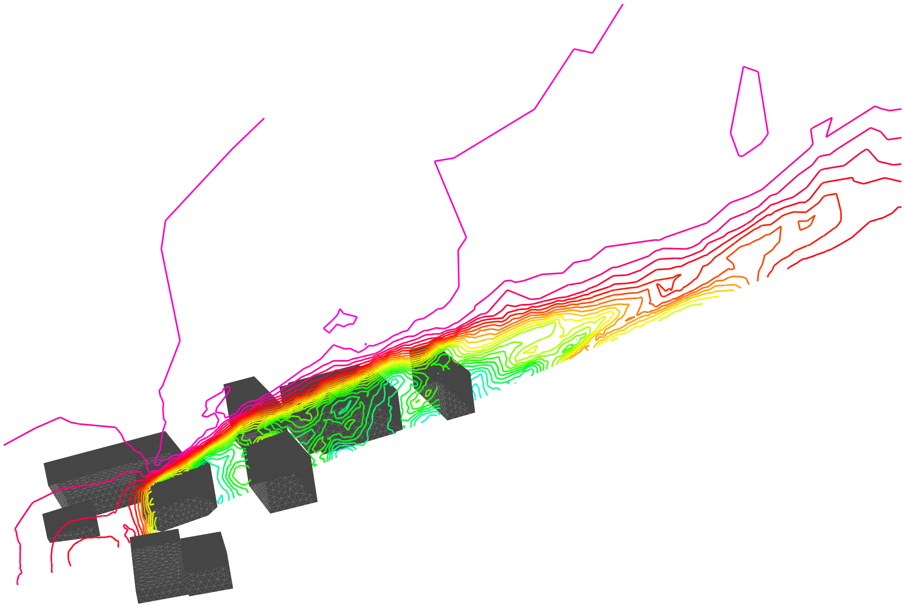

Figure 4.11: Velocity contours in a vertical plane for N/S perfect gas flow in a model city block. (for legend refer to fig. 4.10)

4.2 Chemical injection

The simulations of wind flow are in line with expectations(no previous studies of these geometries exist). In the following, several chemical release simulations have been done for both grids, and the contour plots obtained are shown and discussed below.

This study mainly considers the injection(or release) of two kinds of chemicals, Chlorine and Ammonia. The patterns of injection include, 1)injection from ground to the air(or Nitrogen), 2)injection from multiple locations in an urban area and 3)injections near a building. All the injection directions are vertical and pointing to the ceiling. The inflow/outflow boundary condition is still the same as the wind flow cases (M=0.01, T=300K, p=1atm). The injection mixture is made up of 0.1 percent of chemical pollutant and 99.9 percent of Nitrogen(or air). The injection velocity is the same as the wind speed. In order to accommodate injection, a change had to be made to the boundary conditions. The injection surface is treated as a fifth kind of boundary surface, with an inflow boundary condition applied to it.

Fig. 4.12 to fig. 4.15 show contour plots for chlorine releases in the model city block, with W/S wind. Fig. 4.16 through fig. 4.19 are results for chlorine(or ammonia) release under N/S wind condition in the model city block. The same configurations are also considered for New Orleans cases(fig. 4.20 to fig. 4.27). Comparing chlorine injections with ammonia injections, one can find that, under the same conditions, chlorine's maximum species density are always equal to or higher than ammonia's. From results of these injections, it is clear that chlorine transport is slower than ammonia. This phenomenon can be partly explained by the fact that chlorine has a much higher molecular mass number than does ammonia. When looking at chemical(chlorine or ammonia) releases in both geometries, it is clear that the region that has pollutant concentration is smaller and narrower in the model city block than that in New Orleans cases. This effect is due to the difference between the two geometries: New Orleans grid has more obstacles than the model city block has, and the average distance between two buildings is larger in a model city block than

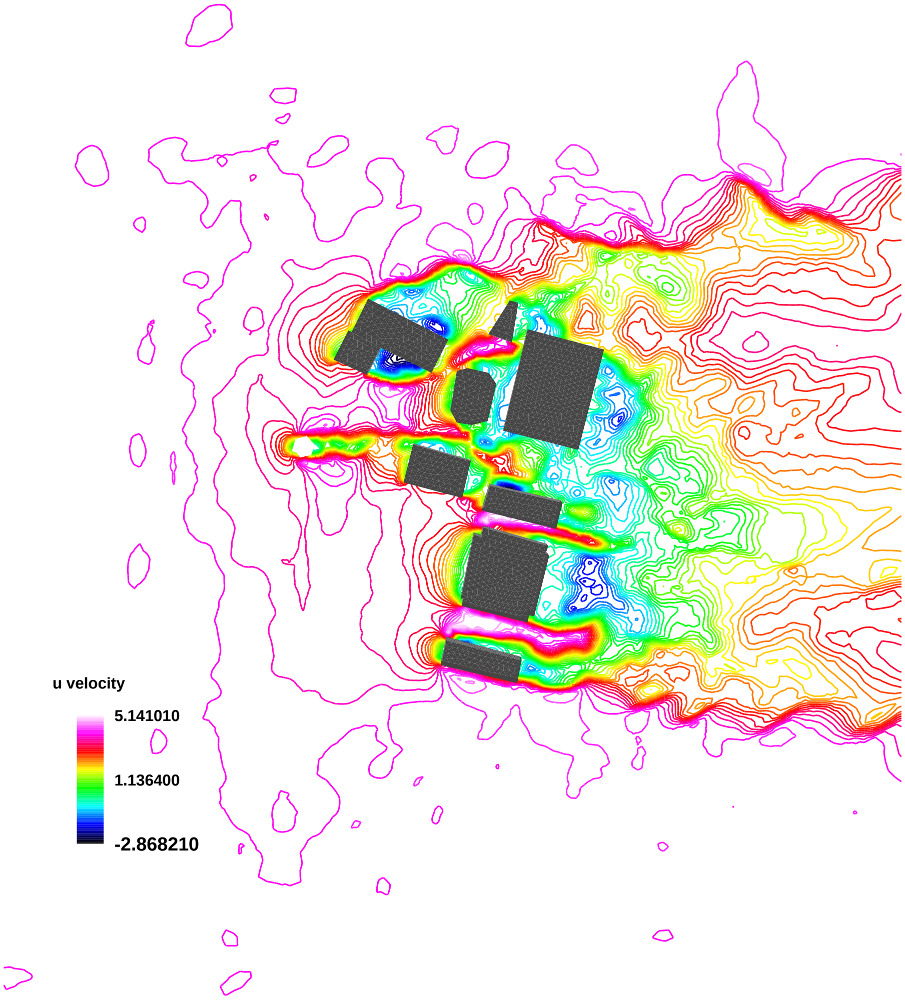

Figure 4.12: Chlorine release in W/E wind in a model city block: u velocity contours(top view)

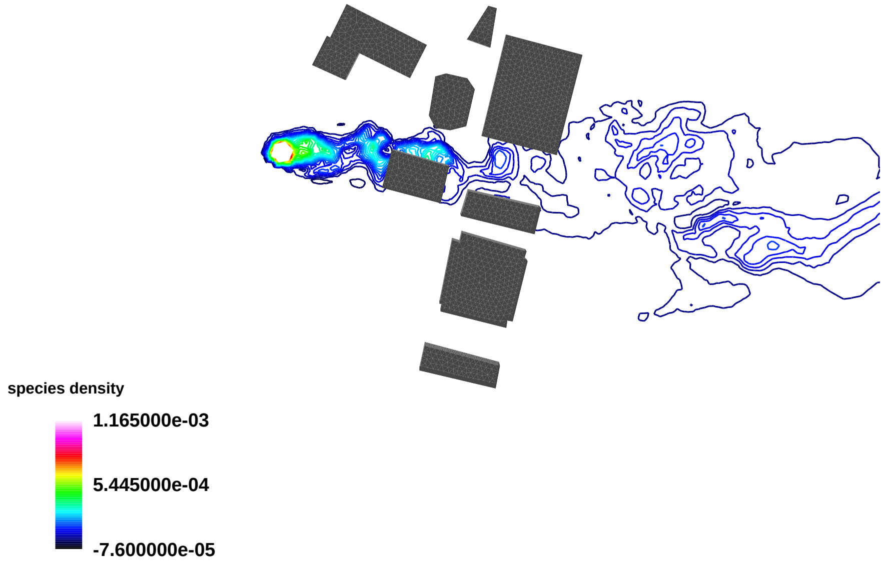

Figure 4.13: Chlorine release in W/E wind in a model city block: species density contours(top view)

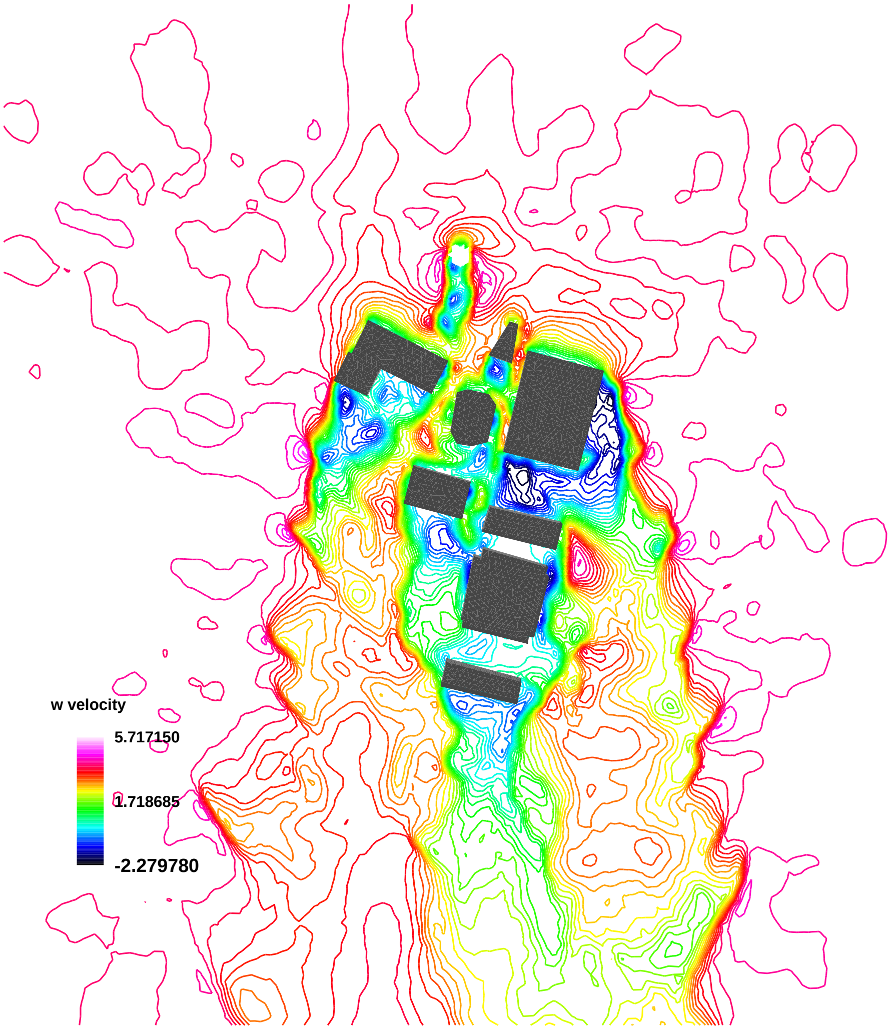

Figure 4.14: Chlorine release in N/S wind in a model city block: w velocity contours(top view)

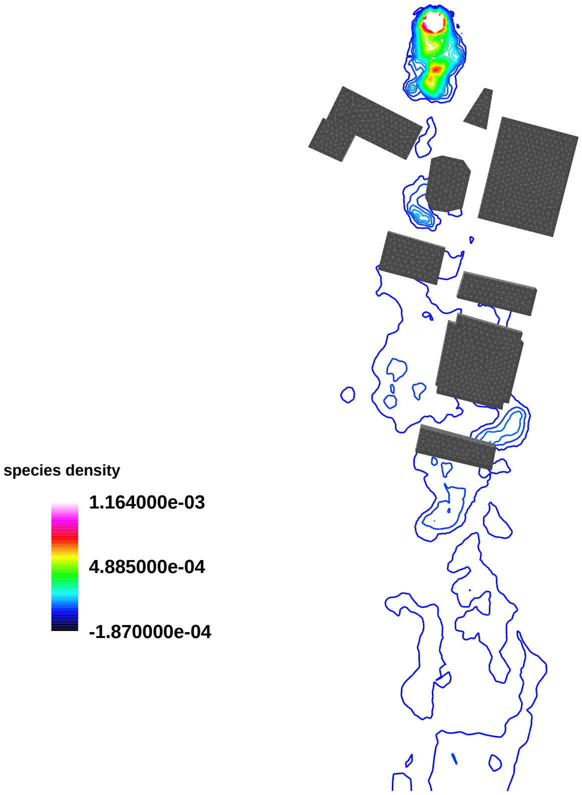

Figure 4.15: Chlorine release in N/S wind in a model city block: species density contours(top view)

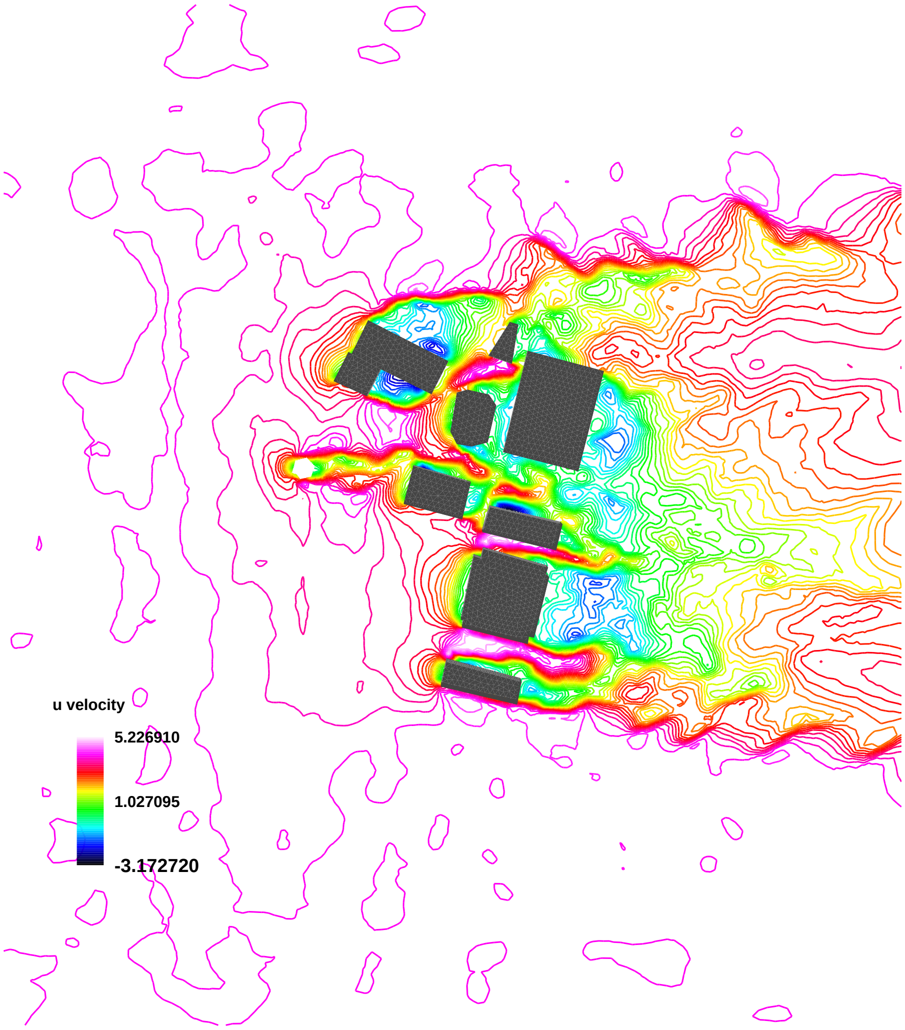

Figure 4.16: Ammonia release in W/E wind in a model city block: u velocity contours(top view)

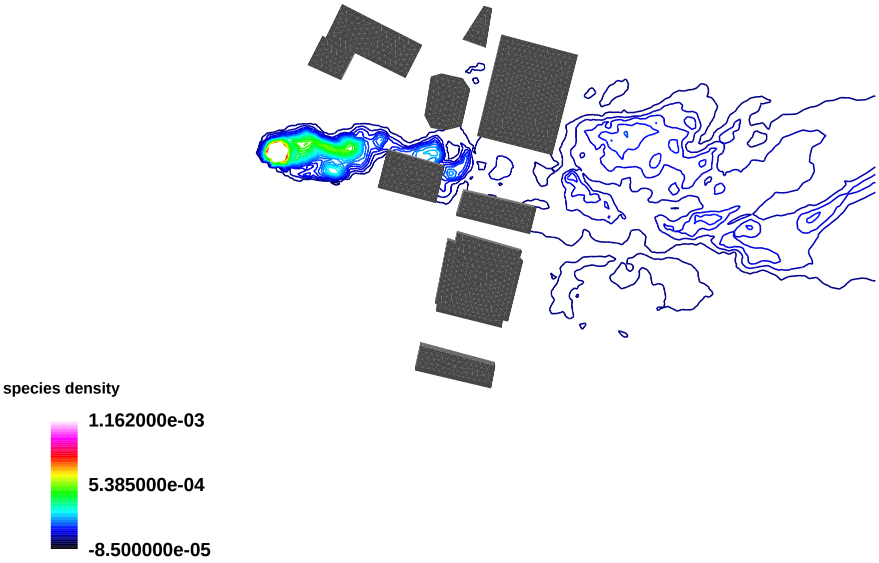

Figure 4.17: Ammonia release in W/E wind in a model city block: species density contours(top view)

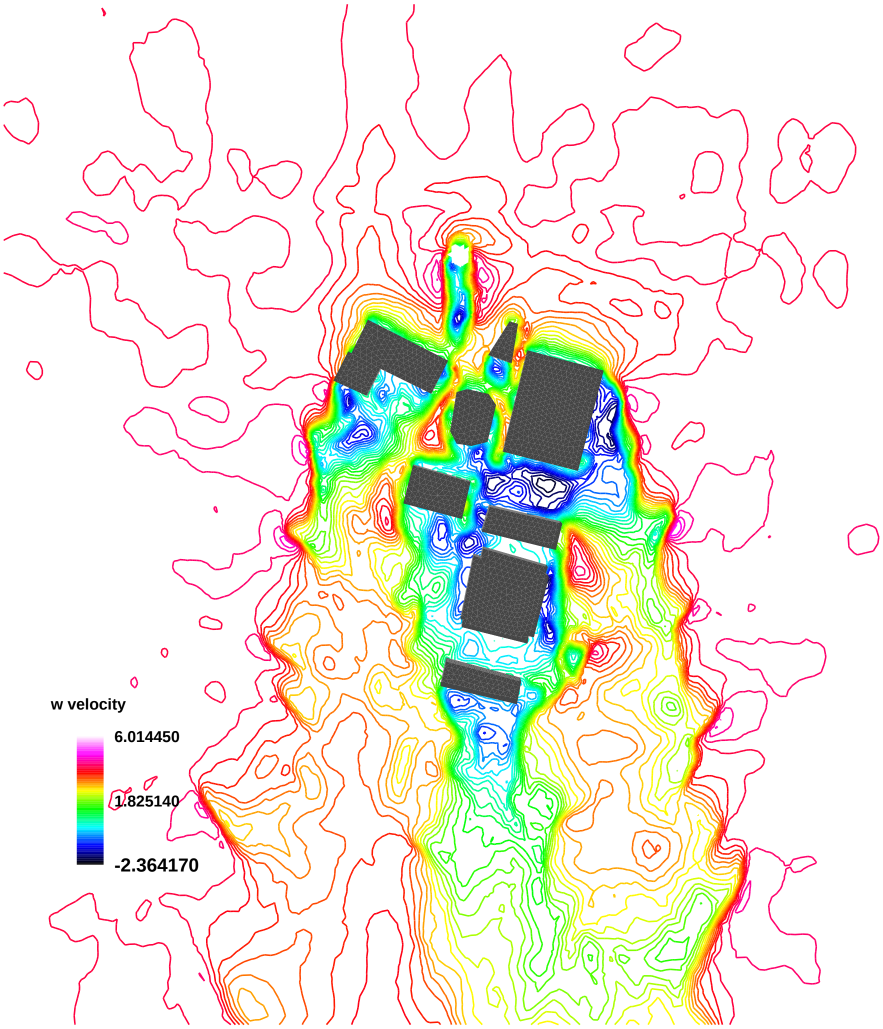

Figure 4.18: Ammonia release in N/S wind in a model city block: w velocity contours(top view)

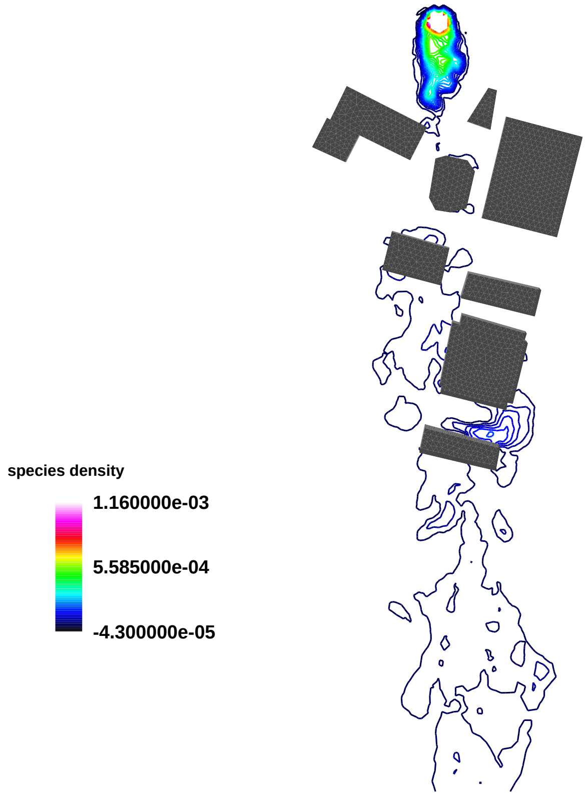

Figure 4.19: Ammonia release in N/S wind in a model city block: species density contours(top view)

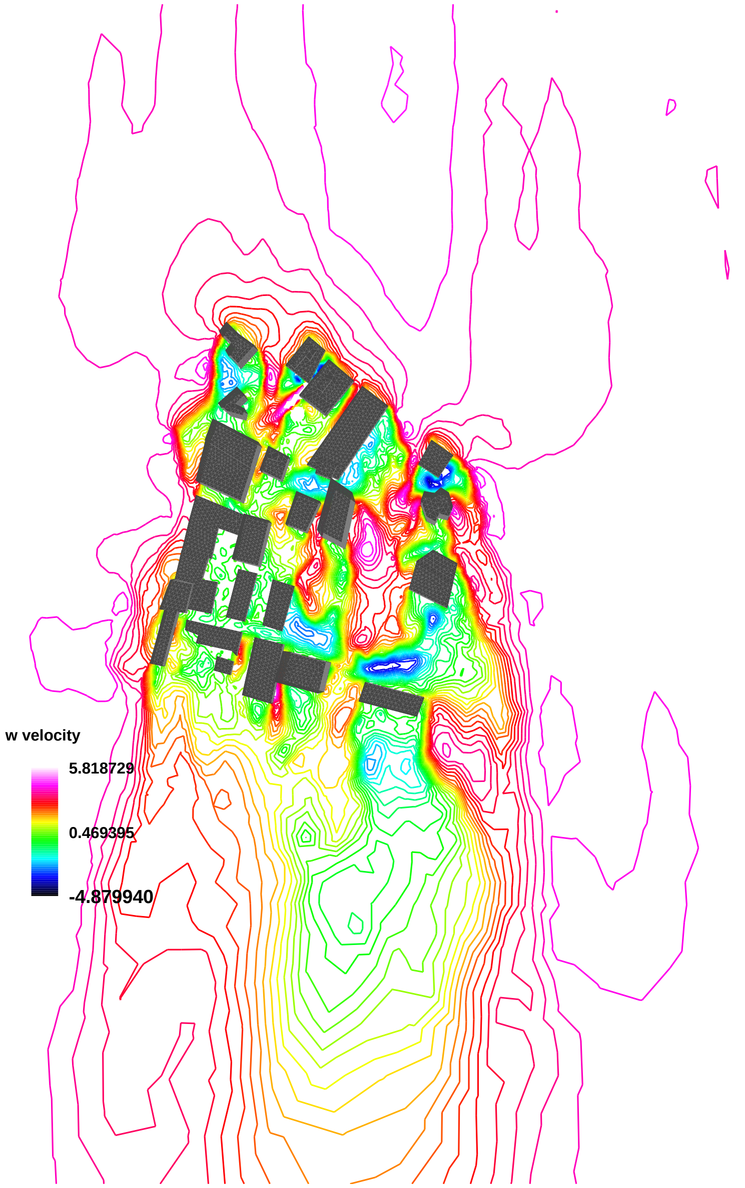

Figure 4.20: Chlorine release in N/S wind in New Orleans: w velocity contours(top view)

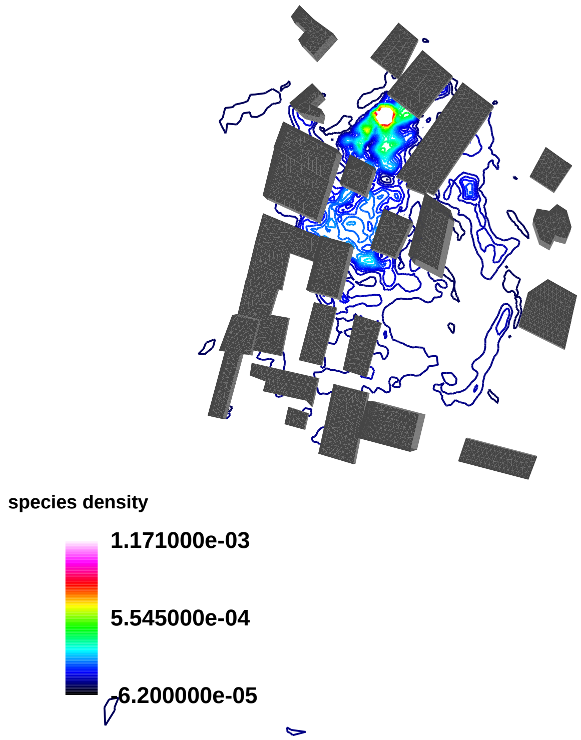

Figure 4.21: Chlorine release in N/S wind in New Orleans: species density contours(top view)

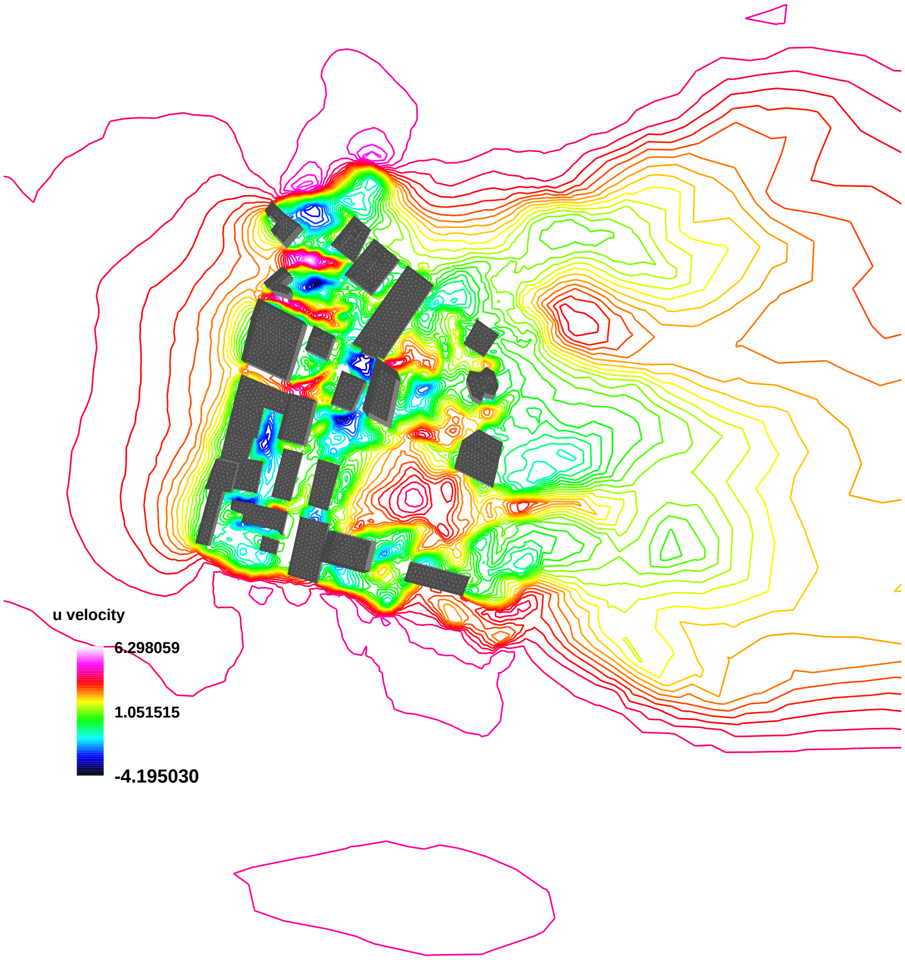

Figure 4.22: Chlorine release in W/E wind in New Orleans: u velocity contours(top view)

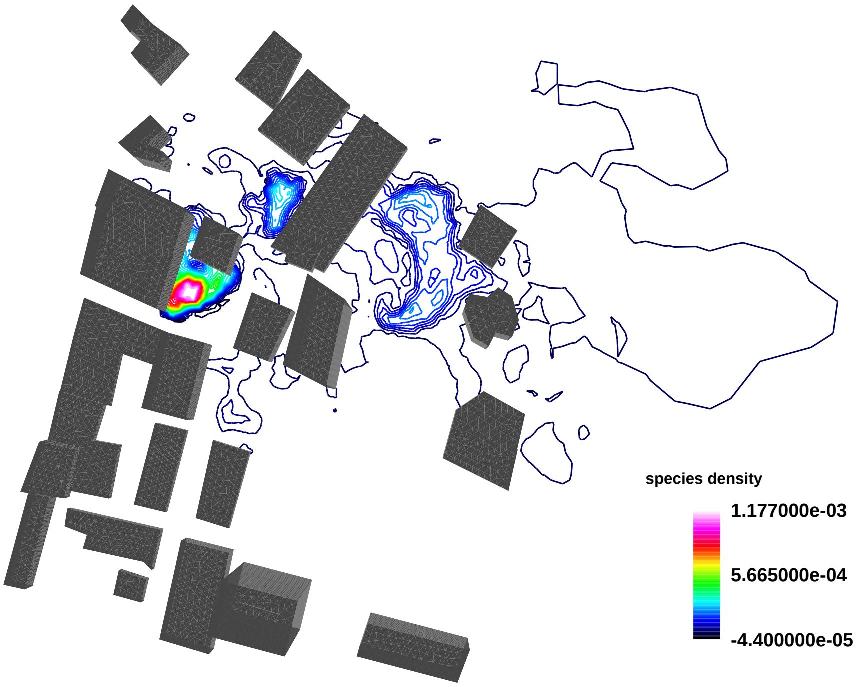

Figure 4.23: Chlorine release in W/E wind in New Orleans: species density contours(top view)

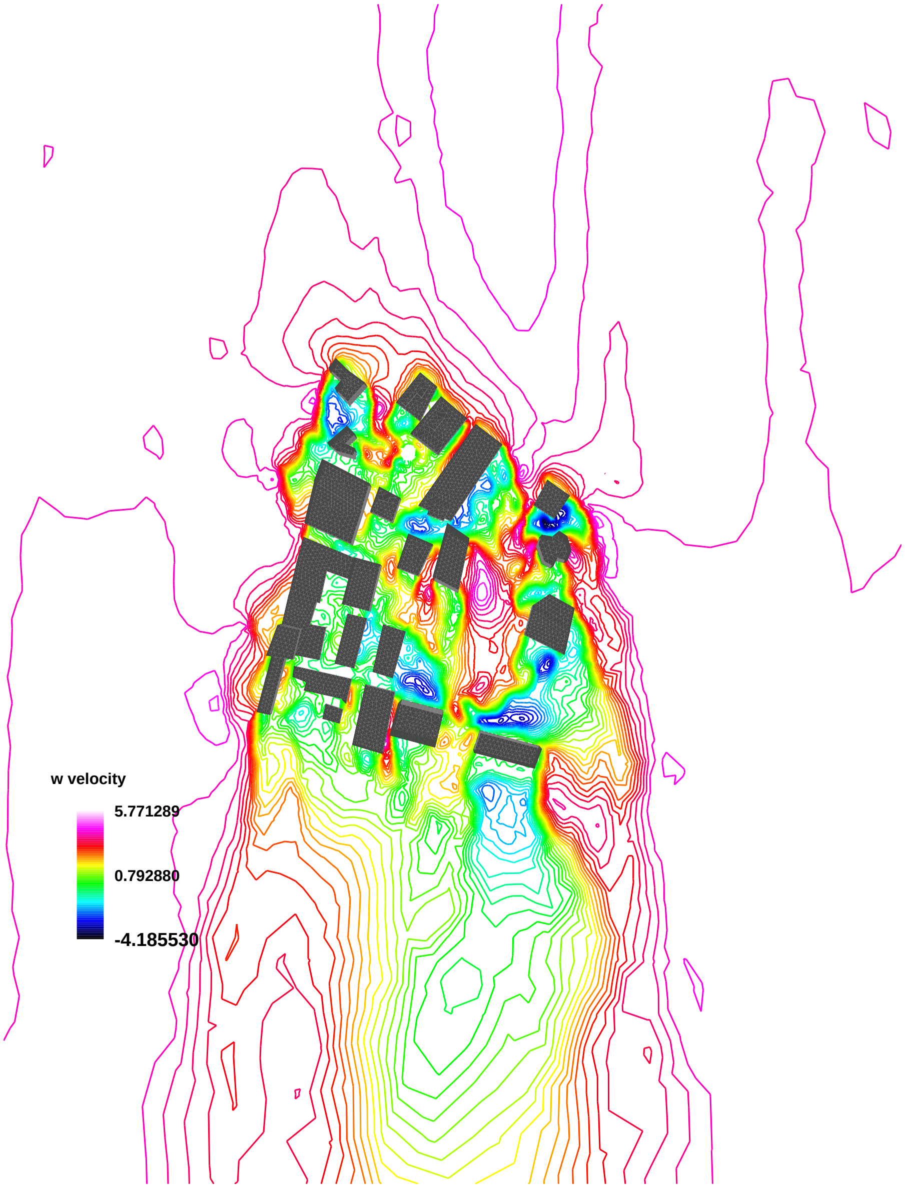

Figure 4.24: Ammonia release in N/S wind in New Orleans: w velocity contours(top view)

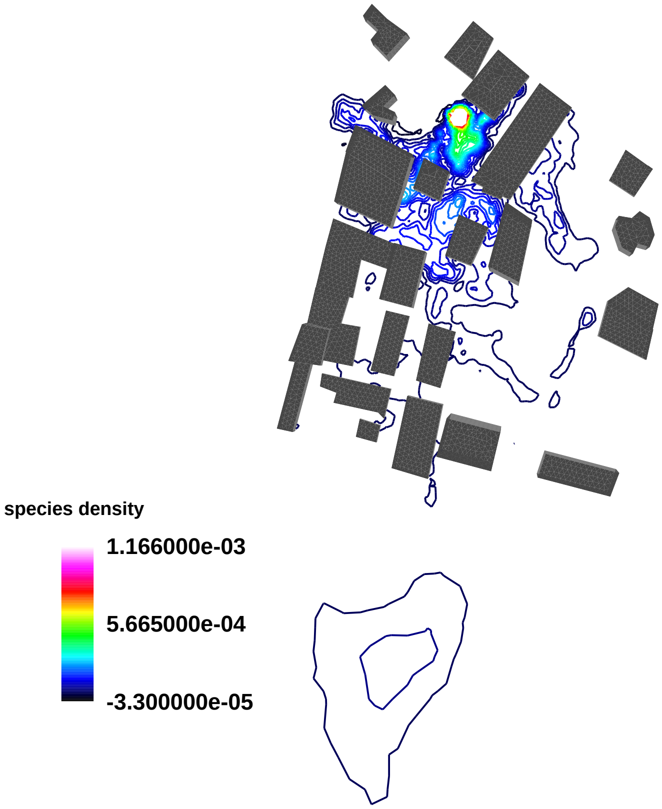

Figure 4.25: Ammonia release in N/S wind in New Orleans: species density contours(top view)

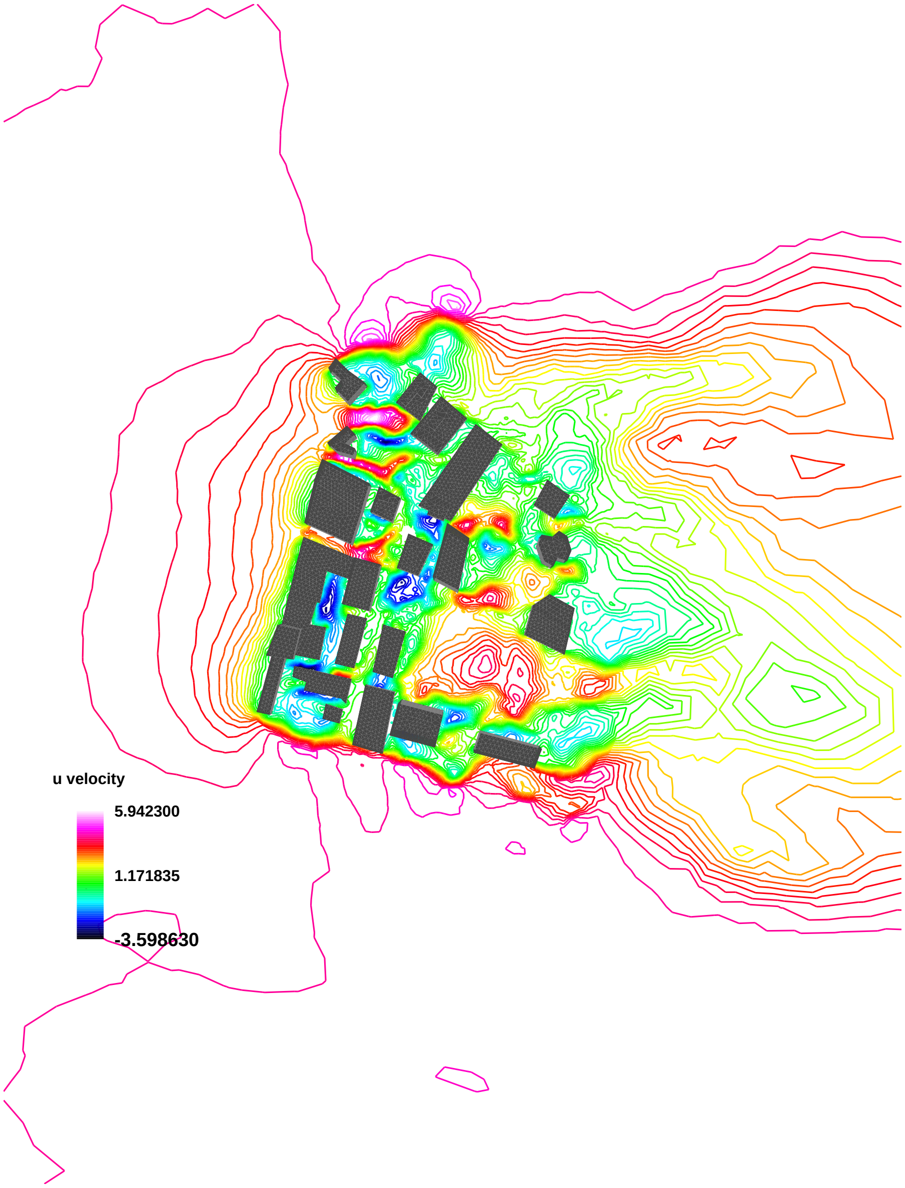

Figure 4.26: Ammonia release in W/E wind in New Orleans: u velocity contours(top view)

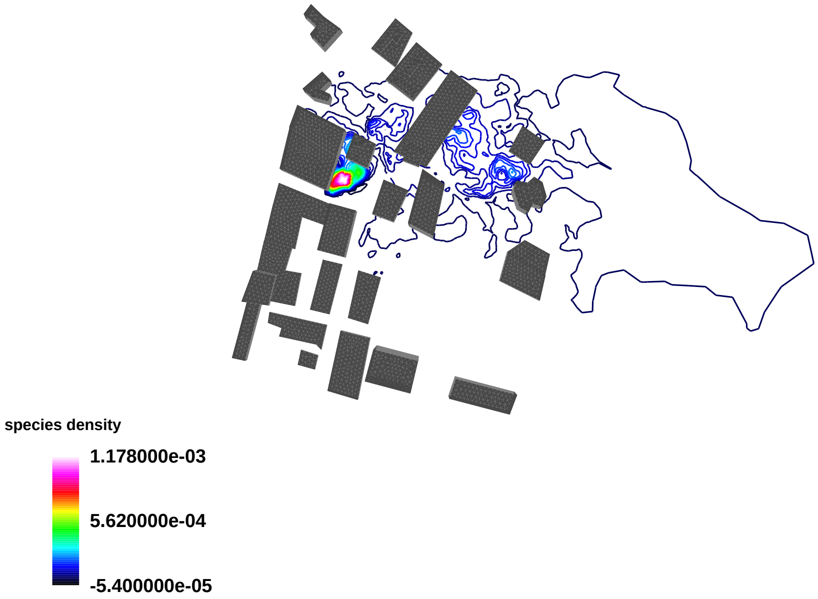

Figure 4.27: Ammonia release in W/E wind in New Orleans: species density contours(top view)

that in New Orleans. At the same time the injection location also contributes to the effect. The release sites are all outside the building cluster in the model city block, while those injection points in New Orleans are surrounded by buildings. Analysis of different chemical release cases will provide helpful information for industry applications. These results can help fire fighters and other emergency rescue crews to predict the pollution region and to shorten response time. This information can be used by government to prevent terrorism biological/chemical attack. For instance, anti-terrorism specialist can predict the possible biological/chemical attack locations by analyzing the simulation results.

Fig. 4.28, 4.29, 4.30 and 4.31 show contours of a vertical plane in the direction of wind flow. These pictures clearly tell how the fluid moves in vertical direction.

4.3 Multiple injection

In reality, pollutant release(or terrorism biological/chemical attack) usually happens in several locations. This circumstance extremely increases the difficulty in accurate prediction of the pollution effect. It can be seen in the above pictures, that for the New Orleans topology pollutant only passes several buildings in the case with only one injection site under W/E wind condition. Therefore, a multi-injection case is proposed and discussed here. The only change that has to be made was cutting out a new injection boundary from the previous grid, with a inflow boundary condition prescribed to it. Both the inlet description and the initial conditions are the same as in the previous cases.

Comparing fig. 4.33 and 4.27 one can see the different patterns of the pollutant transport. In fig. 4.27 there is a relative small recirculation zone, with lower pollutant density. However, in the two-injection case(fig. 4.33 and 4.34), one can observe a larger recirculation zone coupled with higher pollutant density. This fact can be attributed to the contribution of the additional

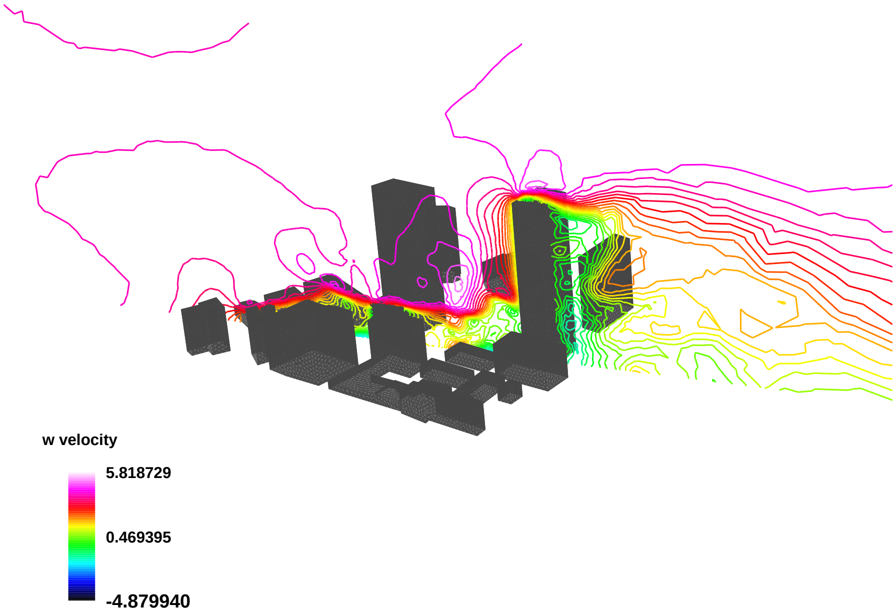

Figure 4.28: Chlorine release in N/S wind in New Orleans: velocity contours in a vertical plane

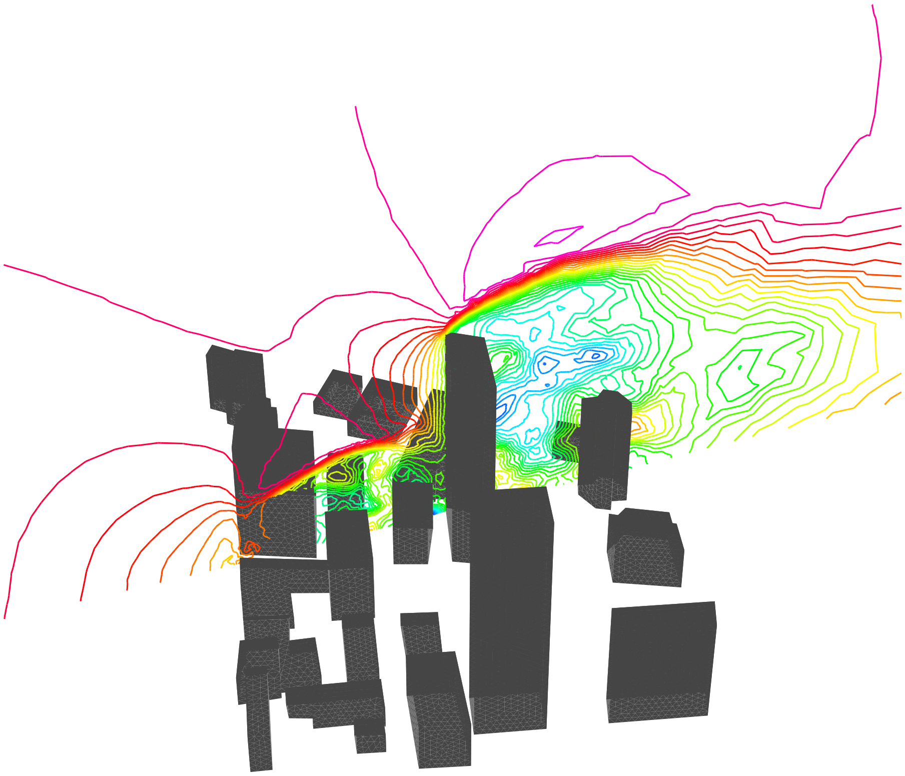

Figure 4.29: Chlorine release in W/E wind in New Orleans: velocity contours in a vertical plane. (for legend refer to fig. 4.22)

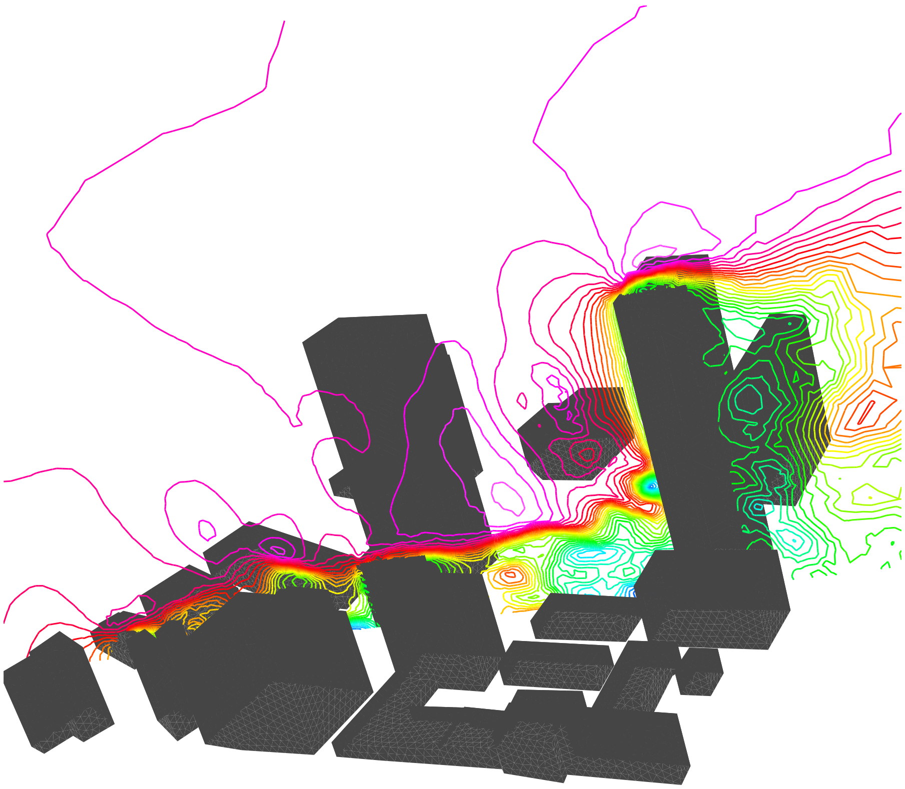

Figure 4.30: Ammonia release in N/S wind in New Orleans: velocity contours in a vertical plane. (for legend refer to fig. 4.24)

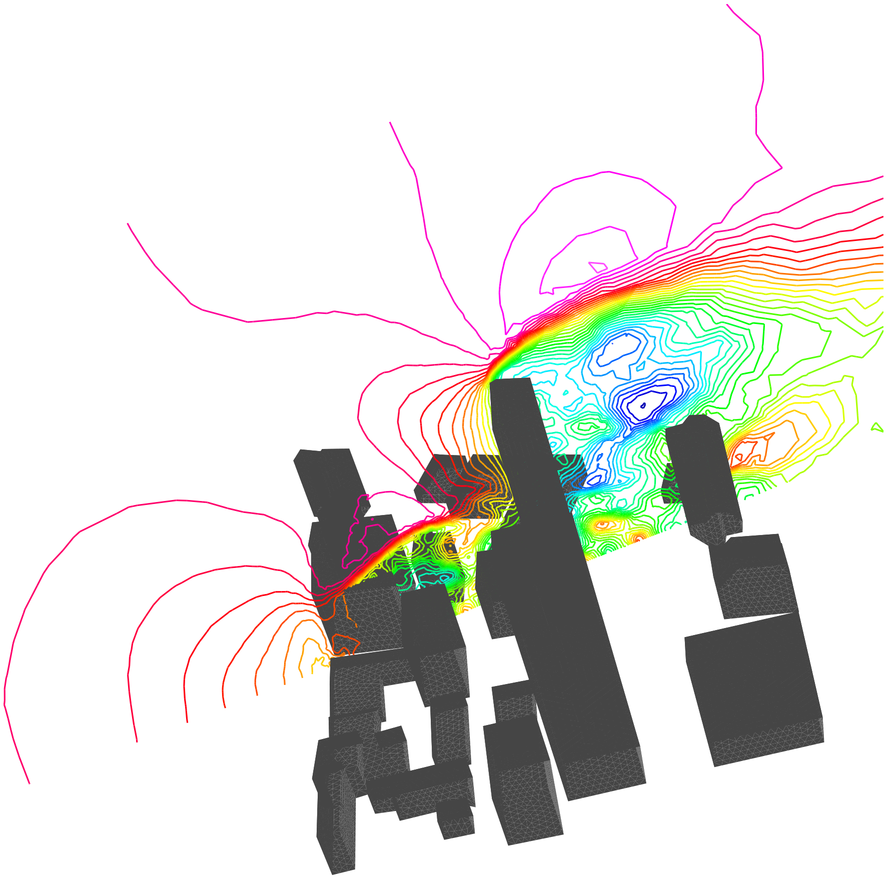

Figure 4.31: Ammonia release in W/E wind in New Orleans: velocity contours in a vertical plane. (for legend refers to fig. 4.26)

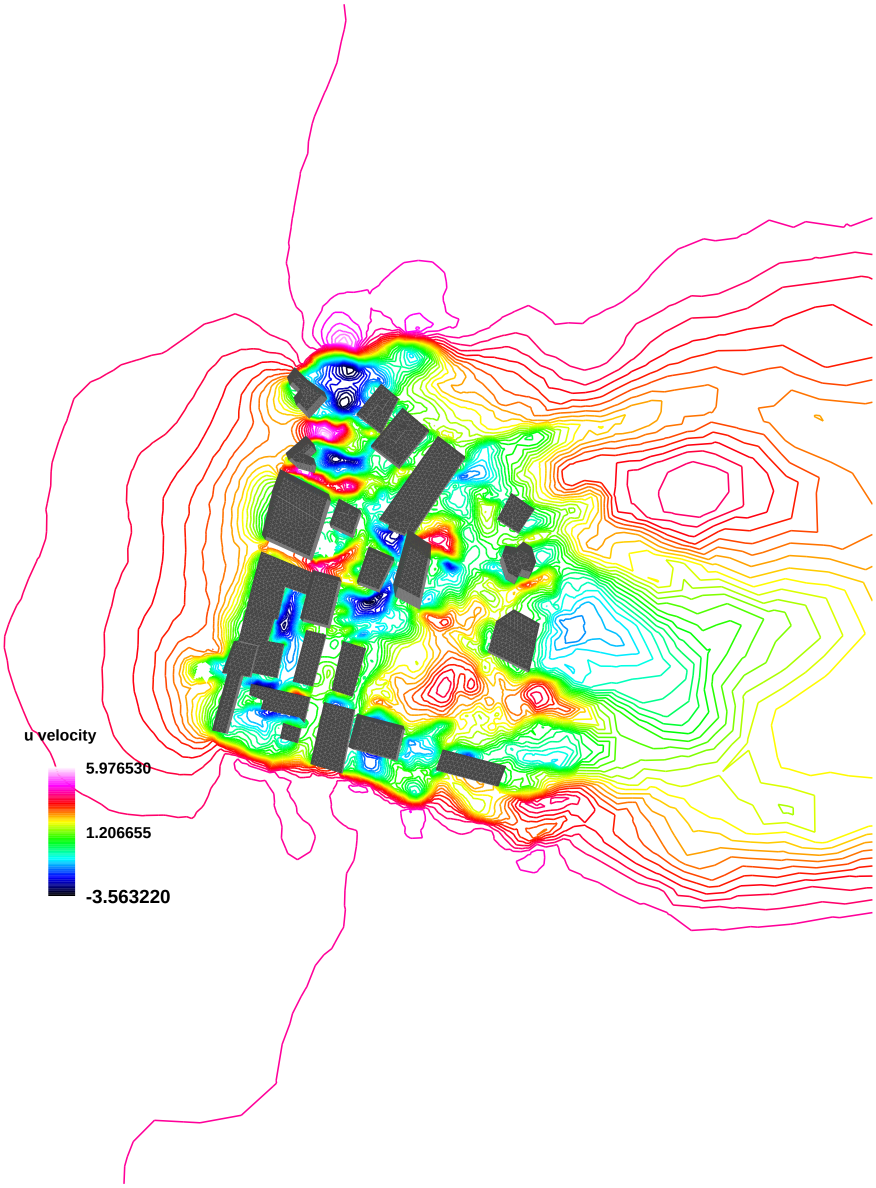

Figure 4.32: Two injections in W/E wind in New Orleans: u velocity contours(top view)

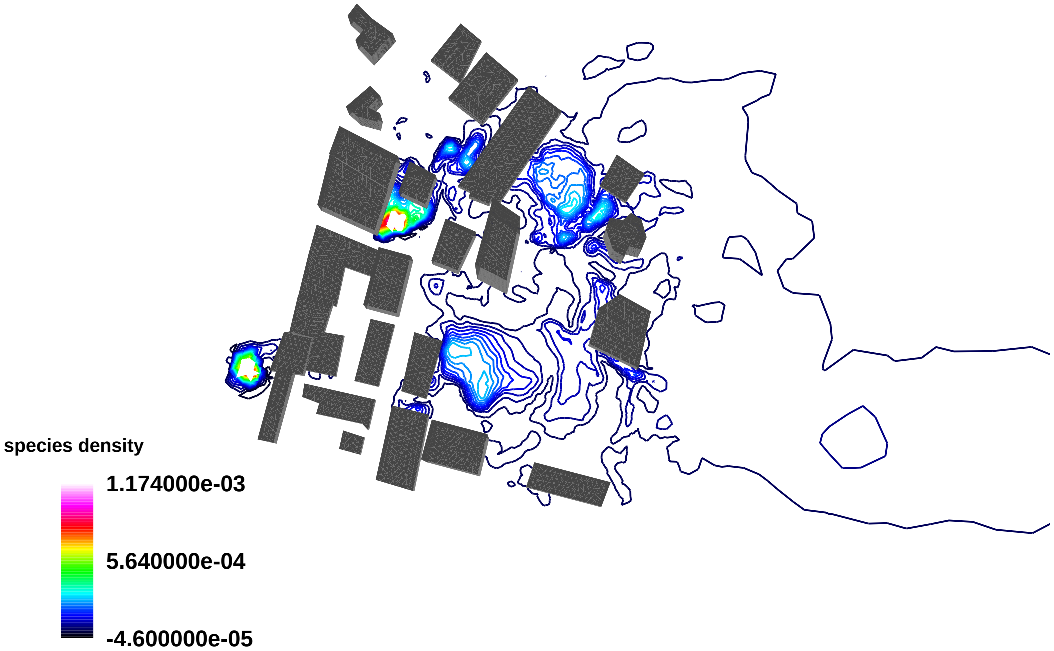

Figure 4.33: Two injections in W/E wind in New Orleans: species density contours(top view)

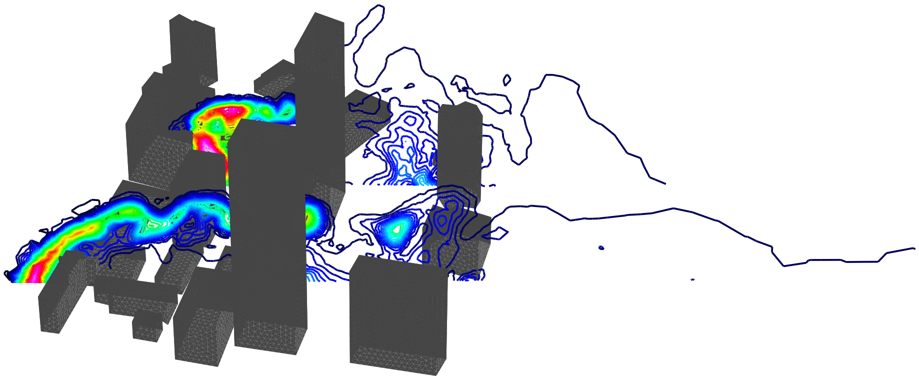

Figure 4.34: Two injections in W/E wind in New Orleans: species density contours in a vertical plane. (for legend refer to fig. 4.33)

pollutant release. The location of reattachment is not changed even in the two injection case. This location is closely related with the geographical property and wind conditions.

4.4 Performance of CHEM

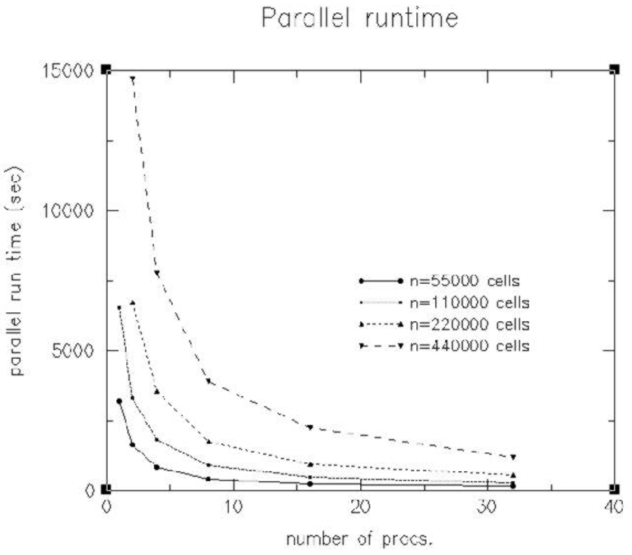

Figure 4.35: Parallel runtime for 3-D jet impingement

To measure the performance of distributed memory version of CHEM, a simple three-dimension case, the jet impingement on inclined plate, was used. The nozzle geometry used is a 15 degree half angle conical-type nozzle. The distance between nozzle exit and the center of the plate is $3.0D_N$, where D_N is the the nozzle exit diameter. The angle that the plate makes with the nozzle are 30^o. The ratio between nozzle exit pressure and ambient pressure are chose as 2. All the performance data was after 100 time-step simulation. Fig. 4.35 shows the parallel run time for problems with different size. For a fix-sized problem, the parallel runtime is decreasing when the number of processors is increased. For example, for the case that has 440,000 cells, the single-processor runtime is 1480 seconds and the runtime on two processors is 760 seconds. In fig. 4.36, one can see, for small number of processors (such as 2, 4,8 and 16) speedup is close to the

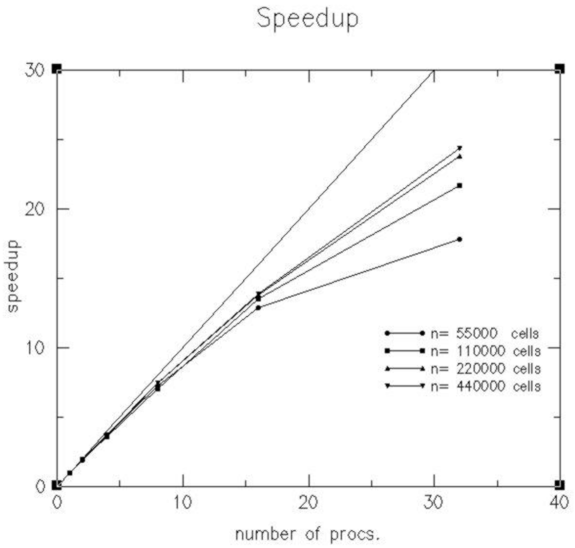

Figure 4.36: Speed up for 3-D jet impingement

processor number. However for larger number of processors, the speedup increases slower. The performance of distributed memory version of CHEM are mainly in line with our expectation. However the optimal performance of distributed memory version can be obtained by ensuring load balancing and minimizing the communication overhead.

CHAPTER V

SUMMARY AND CONCLUSIONS

Several computationally intensive simulations have been attempted and performed in this study. Different combinations of wind conditions and chemical releases on two geometries have been tried. These cases include: pure wind flow, injection from the north side of the building cluster, injection from the west side of the building cluster, injections inside the building clusters with both winds, and a two-injection case with west prevalent wind on the New Orleans grid. These injection locations were chosen because pollutant release at these sites may cause the largest effect to the surrounding environment. Some preliminary results (contour plots for species density and velocity in the wind direction) of these scenarios have been presented in this study. Results with acceptable accuracy were reached with a reasonable computational cost, which is measured by the cpu time and number of processors. Although the current simulation does not include a turbulence model, the contour plots shows a similar flow pattern when compared to previous works which included turbulence models. Moreover, it is more interesting to know where the pollutant will accumulate than the flow patterns close to any single building. Comparison of numerical simulations with experiments has not been possible at the present stage, because no experimental data for these geometries exist. However, the perfect gas results are in line with expectations. One goal of the current study was to test the distributed memory version of CHEM on large problems. This version of the code provides researchers with the option of faster and cheaper computing resources. Simulation on a workstation cluster becomes possible and affordable. For example, by running the code on more processors, one can use more memory, which means larger problems can be simulated. A series of large scale problems were successfully solved by means of the distributed memory version of CHEM. Geometries in previous studies were either a surface mounted cube or a regular-shaped single building: this study carries out pollutant transport simulations on more complex geometries. The two geometries in this study are composed of clusters of irregular-shaped buildings. This study considers more realistic circumstances in pollutant transport, such as different chemical pollutant, different injection location, and different inflow conditions. The simulation results for these realistic circumstances provides very useful information. For instance, by carefully analyzing the contour plots similar to those presented in the previous chapter, rescue personnel could establish an efficient plan for emergency response.

Though it seems that the results from this study are quite promising, there is more work left to do. While the current study shows reasonable computational time, a reduced model is needed for fast response. Therefore, simulation for even larger geometries could be performed in shorter time. Due to the coarse nature of the current grids, the simulations in this study do not consider viscous effects. Further development of viscous grids is highly expected. A turbulence model suitable for simulation on complex city geometry is necessary for performing accurate simulation. With better understanding of more detailed mechanisms of chemical reactions among pollutants, reactive flow can be simulated.

REFERENCES

[1] A.G.Robins and I.P.Castro, “A wind tunnel investigation of plume dispersion in the vicinity of a surface mounted cube -i. the flow field,” *Journal of Atmospheric Environment*, vol. 11, pp. 291–297, 1977.

[2] A.G.Robins and I.P.Castro, “A wind tunnel investigation of plume dispersion in the vicinity of a surface mounted cube -ii. the concentration field,” *Journal of Atmospheric Environment*, vol. 11, pp. 299–311, 1977.

[3] S. A. Cheatham, B. Z. Cybyk, and J. P.Boris, “Simulation of flow and dispersion around a surface-mounted cube,” *Journal of the Atmospheric Sciences*, 2000.

[4] F. E.Camelli and R. Lohner, “Flow and dispersion around buildings: An application with feflo,” *European Congress on Computational Methods in Applied Sciences and Engineering*, 2000.

[5] P.Dawson, D.E.Stock, and B.Lamb, “The numerical simulation of airflow and dispersion in three-dimensional atmospheric recirculation zones,” *Journal of Applied Meterology*, vol. 30, pp. 1005–1024, 1991.

[6] S.Murakami and A.Mochida, “3-d numerical simulation of airflow around a cubic model by means of the k-ϵ model,” *Journal of Wind Engineering and Industrial Aerodynamics*, 1988.

[7] D.A.Paterson and C.J.Apelt, “Simulation of wind flow around three-dimensional buildings,” *Building and Environment*, vol. 24, pp. 39–50, 1989.

[8] T.J.Scanlon, “A numerical analysis of flow and dispersion around a cube,” 1996.

[9] D.Delaunay, D. Lakehal, C.Barre, and C.Sacre, “Numerical and wind tunnel simulation of gas dispersion around a rectangular building,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 67,68, pp. 721–732, 1997.

[10] R.P.Sevlam, “Numerical simulation of pollutant dispersion around a building using fem,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 67,68, pp. 805–814, 1997.

[11] Transoft International, *Fluidyn: www.fluidyn.com*, 2000.

[12] Scientific Software Group, *SLAB: www.scisoftware.com*, 2000.

[13] E.A.Luke, *A Rule-based Specification System for Computational Fluid Dynamics*. PhD thesis, 1999.

[14] T. Hill, *An Introduction to Statistical Thermodynamics*. Addison-Wesley Publishing Company Inc., 1960.

[15] R. N. Gupta, J. M. Yos, R. A. Thompson, and K. P. Lee, "A review of reaction rates and thermodynamic and transport properties for an 11-species air model for chemical and thermal nonequilibrium calculation to 30,000k," tech. rep., 1990.

[16] J.D.Jr.Anderson, *Hypersonic and High Temperature Gas Dynamics.* McGraw-Hill, 1989.

[17] J.A.Shaw, "Hybrid grids," *Hand Book of Grid Generation*, vol. 23, 1999.

[18] T. J. Barth and D. C. Jespersen, "The design and application of upwind schemes on unstructured meshes," tech. rep., AIAA, 1989. AIAA 89-0366.

[19] W.Strang, R.F.Tormaro, and M.J.Grismer, "the defining methods of cobalt60: A parallel, implicit, unstructured euler/navier-stokes flow solver," *AIAA paper*, vol. Jan, 1999.

[20] V. Venkatakrishnan, "On the accuracy of limiters and convergence to steady state solutions," *AIAA Paper*, vol. Jan, pp. 93–0880, 1993.

[21] P.L.Roe, "Approximate riemann solvers, parameter vector, and difference schemes," *Journal of Computational Physics*, vol. 43, pp. 357–372, 1989.

[22] P. Cinnella, "Flux-split algorithms for flows with non-equilibrium chemistry and thermodynamics," *Ph.D Dissertation in Aerospace Engineering, Virginia Tech*, 1989.

[23] C. Hirsch, *Numerical Computation of Internal and External Flows: Volume 1 Fundamentals of Numerical Discretization.* John Wiley & Sons, 1988.

[24] J. Wu, "A chorin turkel preconditioning method applied to finite rate chemistry equations," *Engineering Research Center, Mississippi State*, 2001.

[25] M. R. Zaccanti, *Analysis and Design of Preconditioning Methods for the Euler Equations.* PhD thesis, 1999.

[26] D.L.Marcum, *SolidMesh: User's Manual.* Mississippi State Univ. ERC.

[27] P. P. Walatka, J.Clucas, R.K.McCabe, and R.Potter, *FAST: User's Manual.* NASA Ames Research Center.

[28] Ed.A.Luke, X.-L. Tong, J. Wu, and P. Cinnella, *Chem: a finite rate chemistry solver - the use guide.* Mississippi State Univ. ERC, 2000.

[29] R.P.Hosker, "Flow and diffusion new obstacles," *Atmosphere Science and Power Production*, pp. DOE/CR–2521, 1980.