Physically-based Modeling and Simulation

Radim Dvořák*
Department of Intelligent Systems
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66 Brno, Czech Republic
idvorak@fit.vutbr.cz

Abstract

The thesis deals with the modeling of air pollution transportation and dispersion processes in the atmosphere, more precisely with the numerical approaches to solve such models. The modeling of air pollution has a great importance for prediction of the contaminations and it helps with understanding of the process and with elimination of its consequences.

The models which are described by partial differential equations, namely advection-diffusion equations, and thus they can be solved by numerous analytical/numerical methods are in the scope of the thesis. In particular, well known method of lines (MoL) and several models based on it together with the possibility to accelerate the computation are studied in the first half of the work. It is shown that MoL approach is still suitable for many concrete models and it has a great potential for parallelization on graphics cards.

Quite young ELLAM method and its application to solve atmospheric advection-diffusion equations is the second objective. A concrete form of ELLAM method and its proposed adaptation approaches are evaluated and it is shown that it overcomes the current state of the art methods in many cases.

Categories and Subject Descriptors

G.1.0 [**Numerical Analysis**]: General—*Parallel algorithms*; G.1.8 [**Numerical Analysis**]: Partial Differential Equations—*Method of lines, Finite difference methods, Finite element methods, Parabolic equations*

Keywords

Physically-based modeling, modeling of air pollution, partial differential equations, advection-diffusion equation, numerical methods, ELLAM

1. Introduction

The thesis deals with the modeling of air pollution dispersion in the atmosphere, more precisely with the numerical approaches to solve such models. The modeling of air pollution has a great importance for prediction of the contaminations and it helps with understanding of the process and with elimination of its consequences. The very important in the latter case is to know what space and what concentrations the species could reach and thus to be able to better remove or prevent damages.

The history of these kinds of models is dated to the 19^{th} century when Reynolds formulated a criterion for the change of laminar to turbulent flow [5]. Since then the diffusion phenomena has started to be studied and on the edge of the 19^{th} and 20^{th} centuries the first aerosol deposition models were described. Later on, the models with chemical reactions appeared because chemical reactions influence the amount of pollutant concentration.

There are two approaches for the solution of the atmospheric equations being developed. The first one is the analytical solution that is applied to the simpler cases where the certain parameters or attributes can be omitted. These techniques can serve as a validation for the numerical methods dealing with the more complex problems. Many numerical approaches do exist and they have been developing extensively, particularly in last decades. However, there is still much to enhance. The more precise and the more faster the calculation is the more complex problems can be solved in more details. Therefore, the theme of this thesis is the accurate and fast solution of the models of atmospheric pollutant dispersion.

The paper is organized as follows. The basic terms and the modeling technique used in the thesis is described in the second section. The basic properties of the model are stated in the third section. The concrete methods used in this paper together with the current state of the art method are described in the fourth section. The performed experiments done with the methods are described in detail in the fifth section and the final section contains the conclusions.

2. Pollutant Dispersion Modeling

If one thinks about pollution it is important to answer the question what amount of matter in the air is just impurity and what amount shall be considered as pollutant.

*Recommended by thesis supervisor: Assoc. Prof František V. Zbořil. Defended at Faculty of Information Technology, Brno University of Technology on 19. November 2014.

Dvořák, R.: Physically-based Modeling and Simulation. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 7, No. 1 (2015) 28-46

The answer on the above stated question is not straightforward but it can be shown that the awareness of several aspects can reveal it quite well. First, the context of pollutant is a very important factor. When pollutant reaches the receptor one shall ask what is physical, chemical and biological nature of the receiver (e.g. person, species or the entire population), what is the health condition of the receiver, what is the composition of the pollutant etc. The answers to such questions have to be known in order to state the degree of harm.

The further viewpoint is related to response of the receiver to the pollutant. In case of air pollution it holds in most cases that the more poison to which one is exposed the greater the harm. The above statements can lead us to the following definition [28] - *The presence of contaminants or pollutant substances in the air that interfere with human health or welfare, or produce other harmful environmental effects.*

2.1 Scale Factor

Air pollution exists at all scales, from extremely local to very global ones. The scales can be categorized into several areas: local, urban, regional, continental and global. The range of influence of the pollution can be from molecular level (e.g. nanoparticles) to entire planetary (e.g. greenhouse gases diffusion in troposphere). The local scale is up to about 5 km of the earth's surface. The urban scale extends to the order of 50 km. The regional scale is from 50 to 500 km. Continental scales are from 500 to several 1000 km. The global scale extends worldwide.

The scope of this thesis deals with scale factor of local/partially urban categories. These air pollution problems are usually characterized by one or several large emitters or a large number of relatively small emitters. The lower the release height of a source, the larger the potential impact for a given release might be.

2.2 Basic Terms and Concepts

There are several conceptual terms needed to be considered before the details of model are described. These are sources/sinks of pollution, receptors, transport and dispersion.

2.2.1 Sources and Sinks

The places pollutants are emitted from are called *sources*. The sources can be of artificial or natural manner. The artificial ones include gas pollutions from industry, vehicles and other facilities built by human. The natural sources can be the respirations from plants, animals and fallout of what was once living matter. Other natural sources include volcanoes and naturally caused forest fires. The pollutants disappear in places called *sinks*. These are soil, vegetation and water areas such as oceans.

2.2.2 Receptors

A *receptor* can be the plant or animal that is affected by a pollutant. The interface between a pollutant and a receptor can be its surface (e.g. skin) or its part, lungs when the pollutant is inspired by animal breathing or when its eyes are irritated. Also a material can be the receptor - paper, leather, clothes, etc. Some artificial receptors are made to measure the concentration of the pollution in specific places. These can be used either for pollution statistical measurement or for further processing such as future prediction.

2.2.3 Transport and Dispersion

A *transport* is the process that moves the pollution from the source to the receptor. The simplest examples of the source-receptor system is the point source and single receptor tuple such as a chimney and a building which is 5 km far away. The pollutant flows directly to the receptor when wind blows from source to receptor along the line connecting the two points and when its direction is from the source to the receptor. The receptor is affected by a pollutant, however, the matter does not form the same circular shape all the way it passes. On the contrary, the plume particles move from the edges to the surrounding air and the particles from surrounding air are moving inside plume due to turbulent eddies. Next, if wind speed is smaller/larger than emitting speed plume slows/accelerates and is deformed until it reaches the wind speed.

The two processes, mixing with surroundings and plume deforming (stretch-out), tend to alter the concentration of the pollutant less at the receptor than at the source. The sum of these processes is called *diffusion*. However, the term diffusion has a substantially different meaning in chemistry. Substances diffuse according to Fick's law of diffusion [28], wherein the concentration diminishes with distance from the source. This is known as a concentration gradient. Therefore, *dispersion* is the preferred term.

2.2.4 Transformation

Next to the transportation one should consider also the process of *transformation* which refers to those processes that change a substance of interest into other substance. The two primary modes of transformation are physical (transformations caused by physical laws, such as radioactive decay) and chemical (transformations caused by chemical or biological reactions, such as dissolution and respiration) [26].

2.3 Gradient Transport Models

There exist many techniques and corresponding models describing air pollution phenomena. These could be *Gaussian plume model*, *Narrow plume hypothesis*, *Trajectory models* and *Gradient transport model*. The latter is in main focus of the thesis and it is described below in detail.

There are defined certain physical variables in fluid mechanics which describe fluid behaviour in time and space. Fluid behaviour can be expressed as the change of its concentration in space and time. Concentration C (kg m^{-3}) is defined as $C = M/V$, where M (kg) is matter and V (m^{-3}) is volume. The concentration change of fluid in atmosphere is characterized by two main processes - diffusion and advection.

Diffusion is the fluid property to randomly spread from places with higher concentrations to places with lower concentrations. The process is caused by random movement of molecules, so called Brownian motion. Because it is case of random process, the average case of the motion can be statistically described in one-dimensional case as follows [26]

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2}. \qquad (1)$$

In case of advection, matter moves in one-dimensional space by certain speed a_x (m s^{-1}). The advection equation expressed using concentrations has the following form

$$\frac{\partial C}{\partial t} = -\frac{\partial a_x C}{\partial x}. \qquad (2)$$

When both processes are taken into account they lead to basic one-dimensional advection-diffusion equation in the form

$$\frac{\partial C}{\partial t} + \frac{\partial a_x C}{\partial x} = D_x \frac{\partial^2 C}{\partial x^2}. \qquad (3)$$

Extending equation (3) into three dimensions leads to the basic advection-diffusion equation in three-dimensional space with x and y and z axes in the form

$$\frac{\partial C}{\partial t} + \nabla \vec{a} C = \vec{D} \nabla^2 C, \qquad (4)$$

where $\nabla = \vec{x}\frac{\partial}{\partial x} + \vec{y}\frac{\partial}{\partial y} + \vec{z}\frac{\partial}{\partial z}$ is nabla operator with unit vectors $\vec{x}$, $\vec{y}$, $\vec{z}$ along particular axes, $\vec{a}$ is vector (a_x, a_y, a_z) and $\vec{D}$ is vector (D_x, D_y, D_z).

Finally, when diffusion coefficients are not constant, the general form of advection-diffusion equation used in gradient transport modeling will have the form

$$\frac{\partial C}{\partial t} + \nabla \vec{a} C = \nabla \left(\vec{D} \nabla C \right). \qquad (5)$$

3. Properties of Advection-Diffusion Equation

In this section the attention is given to basic terms and properties of ADE equations which are special case of general PDE.

3.1 Monotonicity

Advection-diffusion equation underlies the laws of mass conservation, therefore if $C(x,t)$ is interpreted as a concentration of some species then the integral $M(t)$ defined by equation (6) represents the mass on interval $[a,b]$. It can be shown that $\frac{d}{dt}M(t) = 0$ and thus mass balance is preserved [15].

$$M(t) = \int_a^b C(x,t)dx. \qquad (6)$$

3.1.1 Positivity

Because of the physical nature of ADE equations whose solutions are concentrations of chemical species, it is natural that the following holds

$$\begin{aligned} C(x,0) \geq 0 \quad \text{for all } x \quad \Rightarrow \\ C(x,t) \geq 0 \quad \text{for all } x \text{ and } t > 0. \end{aligned} \qquad (7)$$

In general, there is no guarantee that the spatial discretization scheme, whose computed values are denoted below with w, maintains the above non-negative property. Lets consider the system of ordinary differential equations in $\mathbb{R}^m$ for $t > 0$

$$w'(t) = F(t, w(t)). \qquad (8)$$

This system will be called *positive* if

$$w(0) \geq 0 \quad \Rightarrow \quad w(t) \geq 0 \quad \text{for all } t > 0. \qquad (9)$$

3.1.2 Maximum principle

Another very important property of ADE equations is called the *maximum principle* which can be expressed, assuming N is number of grid points, as

$$\begin{gathered} \min_j w_j(0) \leq w_i(t) \leq \max_j w_j(0) \\ \forall t \geq 0, \forall i,j \in \{1,2,...,N\}. \end{gathered} \qquad (10)$$

The maximum principle says that the concentration anywhere in space $[a,b]$ cannot be lower/larger than minimum/maximum concentration value of the initial concentration profile $C(x,0)$. The maximum principle property means that there are no *global overshoots* or *undershoots* in the system. There is in general no guarantee that this principle is always satisfied.

3.2 Flux and its limiting

Some of the numerical schemes dealing mainly with advection equation part can cause oscillations around the exact solution which can lead also to global overshoots or undershoots. The problem is seen mostly in higher-order schemes like third-order upwind biased scheme [15]. The reason for such behaviour is inaccurate calculation of local fluxes between grid points or cells. Therefore, the description of the flux form of the ADE equation and its limitation is further discussed.

3.2.1 Flux form of equation

To find out the cause of such behavior lets consider the discretization of pure advection equation (2). The space is discretized into the uniform vertex centered grid Ω_h. Lets further consider the auxiliary grid points $x_{j\pm1/2} = \frac{1}{2}(x_{j\pm1} + x_j)$ lying in the middle of grid points. The auxiliary grid points $x_{j\pm1/2}$ now delimit the grid cell $\Omega_j = \left[x_{j-1/2}, x_{j+1/2}\right]$. Further, the cell averages are defined as

$$\overline{C}(x_j,t) = \frac{1}{h}\int_{\Omega_j} C(x,t)dx = C(x_j,t) + \mathcal{O}(h^2).$$

The moving mass into/out from/to cell is called *inflow / outflow flux.* It is physically correct to assume that the cell averages can change only by moving the concentrations out or into the cell. At each cell interface $x_{j\pm1/2}$, fluxes $f_{j\pm1/2}$ can be formulated and advection equation can be written as

$$\begin{aligned} h\frac{d}{dt}\overline{C}(x_j,t) =& f_{j-\frac{1}{2}}\left(t, C_{j-\frac{1}{2}}(t)\right) - \\ & f_{j+\frac{1}{2}}\left(t, C_{j+\frac{1}{2}}(t)\right), \end{aligned} \tag{11}$$

where $f_{j\pm1/2}(t, C_{j\pm1/2}(t)) = a(x_{j\pm1/2})C(x_{j\pm1/2}, t)$ and a is advection coefficient. The equation (11) is the *flux form* of advection equation (2).

3.2.2 Flux limiting

Higher-order schemes can be viewed as first-order schemes, characterized by low accuracy, with some correction function. The corrections can be often simply too large and can result in oscillations around the exact solution. The equation of higher-order scheme in flux form can be generalized into [15]

$$f_{j+\frac{1}{2}}(t, w) = a(w_j + \psi(\theta_j)(w_{j+1} - w_j)), \quad a > 0, \tag{12}$$

with ψ entitled as *limiter function.* The limiter function is chosen such as it has better accuracy than the first-order upwind scheme but still preserves the positivity property. It can be deduced [15] that to maintain positivity it is sufficient for limiter function ψ presented in equation (12) that

$$0 \leq \psi(0) \leq 1, \quad 0 \leq \frac{1}{\theta}\psi(\theta) \leq \mu \quad \text{for all } \theta \in \mathcal{R}, \tag{13}$$

where μ is any positive real number, however, the evaluations revealed that 1 is a reasonable value. An example of piece-wise linear limiter function is the one introduced in [16]

$$\psi(\theta) = \max\left(0, \min\left(1, \frac{1}{3} + \frac{1}{6}\theta, \theta\right)\right). \tag{14}$$

4. Solution of Advection-diffusion Equation

In the past century, many numerical methods have been developed for solving ADE equations. The main and most frequently used methods are *finite difference method* (FDM), *method of lines* (MoL) and *finite element method* (FEM). Since the thesis is focused on atmospheric ADE, the state of the art Walcek's method, which is used for comparison with the methods proposed in this work, is presented in the following section.

4.1 Walcek method

Many schemes dedicated to atmospheric advection have been proposed. The well known and widely used is the scheme proposed by Bott [3] on whose basis quite new and very accurate scheme was designed by Walcek et al. [29].

The method uses rather the *mixing ratios* instead of concentrations in order to allow model the scenarios with variable density of the fluid. The mixing ratio R (kg kg^{-1}) of a tracer is defined as ratio between mass or concentration C (kg m^{-3}) of a tracer and density q (kg m^{-3}) of the fluid

$$R \equiv \frac{C}{q}.$$

To describe the algorithm, lets further assume one-dimensional advection equation and uniform grid. The initial guess of the mixing ratio R_i^{guess} in grid cell i at time $t+\Delta t$ is obtained as

$$R_i^{guess} = \left(R_i^t D_{d-1} - \frac{F_{i+1/2}}{\Delta x_i} + \frac{F_{i-1/2}}{\Delta x_i}\right)/D_d, \tag{15}$$

where $F_{i\pm1/2}$ (kg m^{-2}) are fluxes of tracer across the cell boundaries $i \pm 1/2$ and D_d (kg m^{-3}) are dimensional dependent fluid densities. For one-dimensional calculations in incompressible fluids, $D_{d-1} = D_d = 1$. For multi-dimensional calculations in incompressible fluids, $D_{d-1} = 1$ and $D_d = 1 - \Delta t \Delta a_i / \Delta x_i$, where Δa_i (m s^{-1}) is relative wind speed in cell i.

Fluxes and velocities are defined at the edges (faces) of the grid cells where the mixing ratios are defined. Fluxes at cell faces are defined by introducing an *outflowing* mixing ratio R_f as

$$F_{i+1/2} = (q_0 a)_{i+1/2} \Delta t R_f, \tag{16}$$

where $(q_0 a)_{i+1/2}$ is mass flux across the cell boundary $i + 1/2$, defined using the initial fluid density q_0, Δt is integration time step and a is advection variable. The initial density q_0 can be defined using upwinding as one of the cell boundary densities or as average of edge densities in the cell.

R_f is the average mixing ratio in the fluid that is advected into the neighbouring grid cell. The definition of R_f allows to limit the fluxes in reasonable physical values. The initial guess R_f^* of R_f can be algebraically derived as a Courant number-dependent linear combination of the mixing ratios in the three cells closest to the cell face where fluxes are calculated [29]. The linear combination includes the *sharpening factor* α which can be adjusted to actual conditions.

The initial guess R_f^* can produce unrealistic estimates of mixing ratios at cell boundaries (local overshoot or undershoot). It is therefore reasonable to define the physical boundaries of the fluxes and thus to limit the fluxes as

$$\begin{aligned} R_{min} = \min(R_i, R_{i+1}) \leq R_f \leq \\ \max(R_i, R_{i+1}) = R_{max}. \end{aligned} \tag{17}$$

The method introduces also mixing ratio limiting at the end of step calculation. The idea is the following. As long as not all the fluid in one grid is replaced in one time step (Courant number < 1), at time $t+\Delta t$, it is physically impossible for mixing ratios to be greater than the highest mixing ratio $R_{max}^{t+\Delta t}$ or lower than the lowest mixing ratio $R_{min}^{t+\Delta t}$ of the upwind cell or the mixing ratio of the cell initially. Such mixing ratio limitations can be expressed as

$$R_i^{t+\Delta t} = \max[\min(R_{max}^{t+\Delta t}, R_i^{guess}), R_{min}^{t+\Delta t}], \tag{18}$$

where R_i^{guess} is equal to R_f^* limited by equation (17). When mixing ratios are limited the appropriate fluxes at upwind cell boundaries need to be recalculated for the next step.

4.2 Method of lines utilization

There exist huge number of air pollution models that were designed and evaluated in past decades. The general form of the equation describing atmospheric dispersion, which extends ADE equation (5), can be expressed as follows [5]

$$\begin{aligned}\frac{\partial C}{\partial t} + \nabla C\vec{a} =& \nabla\left(\vec{D}\nabla C\right) + \\ & \textit{chemistry} + \textit{emissions} + \\ & \textit{dry\_deposition} + \textit{wet\_deposition},\end{aligned} \tag{19}$$

where C (kg m^{-3}) is pollution concentration, $\vec{a}$ (m s^{-1}) is wind velocity field and $\vec{D}$ (m^2 s^{-1}) is diffusion vector. The chemistry term presents atmospheric chemistry term that is used for the determination of a chemical substance influence to the atmosphere and to the dispersion process itself. The emissions term expresses the rate of the emissions in the atmosphere and its relation to the atmospheric dispersion of the specific pollutant. The last two terms, dry and wet depositions, are the major sink terms in the model and besides they determine the pollutant behaviour above the terrain surface.

If the chemistry, emissions and wet deposition terms are neglected in equation (19) then the following ADE with deposition term is formed

$$\frac{\partial C}{\partial t} + \nabla C\vec{a} = \nabla\left(\vec{D}\nabla C\right) + W\frac{\partial C}{\partial z}, \tag{20}$$

where W (m s^{-1}) is pollutant gravitational settling velocity.

Equation (20) can be furthermore simplified considering the following assumptions. When the wind speed value is sufficiently large, a diffusive transport is negligible in wind direction with respect to advection [10]. Moreover, the coefficients D_y and D_z depend on the downwind distance x only and they are therefore independent on the crosswind distance y and height distance z. From these facts, the diffusive terms can be simplified - the brackets are not needed any more and the second derivatives appear. Last assumption is the presence of stationary source with constant strength during time. Therefore, the result of the simplification is the steady state form

$$a_x\frac{\partial C}{\partial x} = D_y(x)\frac{\partial^2 C}{\partial y^2} + D_z(x)\frac{\partial^2 C}{\partial z^2} + W\frac{\partial C}{\partial z}, \tag{21}$$

where a_x is wind speed along x axis. The complete description of the problem needs boundary conditions to be specified. The first one follows from an assumption of continuous point source with constant strength located in $(0, 0, h_s)$ coordinates

$$C(0, y, z) = \frac{Q}{a_x}\delta(y)\delta(z - h_s), \tag{22}$$

where Q (kg s^{-1}) is source strength and δ is Dirac function. The ground boundary condition comes from fact that pollutant deposition onto the ground occurs at a rate proportional to local air concentration [10] (the flat ground is taken into account here for simplicity)

$$\left[D_z(\infty)\frac{\partial C}{\partial z} + WC\right]_{z=0} = [vC]_{z=0}\,. \tag{23}$$

Here, deposition velocity v (m s^{-1}) depends on many factors such as type and size of pollutant particles, the terrain roughness and the meteorological conditions. The last three boundary conditions follow from natural assumption that pollutant concentration approaches zero far from the source in lateral y directions and high above the ground

$$C(x, +\infty, z) = 0, \tag{24}$$
$$C(x, -\infty, z) = 0, \tag{25}$$
$$C(x, y, +\infty) = 0. \tag{26}$$

4.2.1 Time-dependent variant

The simple steady state equation (21) can be easily extended to its time-dependent variant by adding time derivative $\frac{\partial C}{\partial t}$ to the equation. The still simple variant, used further in the work for its parallel version evaluation, has the form

$$\frac{\partial C}{\partial t} = -a_x\frac{\partial C}{\partial x} + D_y(x)\frac{\partial^2 C}{\partial y^2} + D_z(x)\frac{\partial^2 C}{\partial z^2} + W\frac{\partial C}{\partial z}. \tag{27}$$

The boundary condition representing the source is expressed as

$$C(t, 0, y, z) = \frac{Q}{a_x}\delta(y)\delta(z - H). \tag{28}$$

The ground boundary condition, for $z = 0$ m, has the same form as before (23). Finally, the lateral and longitudinal boundary conditions are

$$C(t, x, +\infty, z) = 0, \tag{29}$$
$$C(t, x, -\infty, z) = 0, \tag{30}$$
$$C(t, x, y, +\infty) = 0. \tag{31}$$

Lets further discretized equation (27) using MoL framework. By approximating spatial derivatives by central differences and letting time derivative continuous, the following semi-discretized model is obtained

$$
\begin{aligned}
&\frac{\partial C(t,i,j,k)}{\partial t} = \\
&-\frac{a_x}{2\Delta x}\left[C(t,i+1,j,k) - C(t,i-1,j,k)\right] + \\
&\frac{D_y(x)}{\Delta y^2}\left[C(t,i,j+1,k) - 2C(t,i,j,k) + C(t,i,j-1,k)\right] + \\
&\frac{D_z(x)}{\Delta z^2}\left[C(t,i,j,k+1) - 2C(t,i,j,k) + C(t,i,j,k-1)\right] + \\
&\frac{W}{2\Delta z}\left[C(t,i,j,k+1) - C(t,i,j,k-1)\right].
\end{aligned} \tag{32}
$$

The appropriate boundary conditions in terms of the MoL discretization are

$$
\begin{aligned}
C(t,0,0,h_s) &= \frac{Q}{a_x \Delta y \Delta z} \\
C(t,0,j,k) &= 0 \quad \text{otherwise},
\end{aligned} \tag{33}
$$

$$C(t,i,+N_j,k) = 0, \tag{34}$$

$$C(t,i,-N_j,k) = 0, \tag{35}$$

$$C(t,i,j,+N_k) = 0, \tag{36}$$

$$C(t,i,j,0) = \frac{D_z(\infty)}{2\Delta z(v-W) + D_z(\infty)} C(t,i,j,2). \tag{37}$$

The model described by equation (32) and corresponding boundary conditions is then solved using the common methods to solve systems of ordinary differential equations. The methods could be Euler, Runge-Kutta and others in the explicit or implicit form [24].

4.2.2 Wortmann's advection-diffusion model

The second model used for MoL evaluation is the model presented by Wortmann et al. [30]. It is the relatively simple steady-state model depicted by equation (38). More specific, it is the PDE in two dimensions

$$a(z)\frac{\partial C}{\partial x} = D_z \frac{\partial^2 C}{\partial z^2} + \left(\frac{\partial D_z}{\partial z}\right)\frac{\partial C}{\partial z}, \tag{38}$$

where C (kg m^{-3}) is concentration, $a(z)$ (m s^{-1}) is wind velocity, D_z (m^2 s^{-1}) is vertical diffusion coefficient.

The needed boundary conditions are defined as follows

$$D_z \frac{\partial C}{\partial z} = 0 \quad \text{for } z \in \{0, H\}, \tag{39}$$

$$a(z)C(0,z) = Q\delta(z - h_s) \quad \text{for } x = 0, \tag{40}$$

where H (m) is domain height, Q (kg s^{-1}) is point source emission rate and h_s (m) is point source height.

The numerical solution obtained by the method of lines has very simple form. Letting x variable continuous the equation after transformation is as follows

$$
\begin{aligned}
&\frac{\partial C}{\partial x} = \\
&\frac{D_z}{a(z)} \cdot \frac{C(x,z+1) - 2C(x,z) + C(x,z-1)}{\Delta z^2} + \\
&\frac{1}{a(z)} \cdot \frac{\partial D_z}{\Delta z} \cdot \frac{C(x,z+1) - C(x,z-1)}{2\Delta z}.
\end{aligned} \tag{41}
$$

Here, Δz is step size in z direction. The appropriate boundary conditions using central differences result in

$$C(x,0) = C(x,2), \tag{42}$$

$$C(x,h) = C(x,h-2), \tag{43}$$

$$C(0,h_s) = \frac{Q}{a(z)\Delta z}. \tag{44}$$

4.2.3 Parallel design

During the past few years, various papers dealing with the exploitation of GPUs for general-purpose computing tasks have emerged. For example, in a paper of Brandvik et al. [4], the authors have carried out numerous experiments with Euler solver implementation with BrookGPU and CUDA platforms. They reached the 29× speed-up in 2D with BrookGPU and 19× speed-up factor in 3D with CUDA.

Micikevicius [20] studied the ways, how to implement the finite difference approaches on single and multiple GPUs. He was able to reach an order of speed-up against high end CPU and linear scaling communication overhead when using multiple GPUs. In the work by Datta et al. [8], an optimal stencil computation kernel is implemented under CUDA. The performance evaluation is given through the number of points calculated per second. Several architectures were compared and the GPU platform was found to be the most efficient one.

Almost the same problem of air pollution and the possible way of computational acceleration is studied in work from Molnar et al. [21]. The model in their work was based on stochastic model rather than ADE approach used in this thesis. Nevertheless, they were able to reach 80-120 times faster computational time on single GPU than on CPU.

The works presented here were a big motivation to implement and test the concrete parallel version of the numerical model based on MoL framework described in this work.

CUDA/OpenCL architectures. The GPUs are especially well suited to address the problems that exhibit a data-parallel nature with high arithmetic intensity. Until recently, the communication link between a program running on the CPU and the graphics hardware had to be established by means of using a graphics API such as OpenGL. To perform computations on graphics hardware, the programmer was obliged to use graphic primitives and store the data in texture maps. Luckily, the advent of transparent and flexible programming frameworks, like CUDA [7] in 2007 or OpenCL [11] at the very end of 2008, which provide an interface between GPU and CPU host machine, enabled the majority of the aforementioned drawbacks to be mitigated. Moreover, the introduction of

OpenCL framework has enabled the possibility to write a single piece of code that can be further launched on various compatible architectures.

CUDA nad OpenCL frameworks have to be seen as a fusion of SW and HW parts. Lets first consider and example of GeForce GTX2xx series cards which were further used in our tests. It offers 240 stream processing elements organized into a collection of 30 identical multiprocessors. Each multiprocessor has its own shared memory, which is common to all the 8 processors inside. It also has a set of 32-bit registers, texture, and constant memory caches. At any given cycle, each element in the multiprocessor executes the same instruction on different data, which makes each a SIMD processor. Communication between multiprocessors is carried out through the device memory, which is available to all the processors of the multiprocessors.

From the SW point of view, a target application for CUDA / OpenCL is based on a collection of threads running in parallel. The computation is distributed in a grid of thread blocks (work-groups in case of OpenCL). All blocks contain the same number of threads that execute a program, known as kernel, on a device. Kernels are called work-items in case of OpenCL terminology. It is possible to use a 1D, 2D, or 3D index space to invoke and keep a hand on the kernel.

Several different types of memory are available in the frameworks. Global memory can be accessed by every work-item on the compute device, which mostly offers the slowest access speed and largest size. The purpose of global memory/constant memory cache is to improve the necessary latency associated with data transfers. It is readable only for most of the time. Local memory, which is available to all work-items in the same work-group, offers much faster speed than global memories. However, its size is very limited. Private memory of a work-item, which is only accessible by itself, has the lowest latency but most limited storage space.

The efficiency of a kernel can be significantly improved by taking an advantage of parallel access to shared memory and by avoiding bank conflicts. The performance of iterative or multi-phase algorithms can be improved if all the computations can be performed in the GPU, so that step 3 bellow can be run several times without the need to exchange the data between device and host. A typical algorithm execution flow for the frameworks consists of the following stages:

1. Allocate data on the device.
2. Transfer data from the host to the device.
3. Proceed with the execution of kernel(s). The result is stored in device (local) memory.
4. Retrieve data from device and transfer them to the host environment.

Outline of the solution. The selected numerical model of our solution divides the space into a finite set of discrete points where the concentration level is calculated for each of them. In case of time-dependent variant of PDE (32) the number of equations is obtained as $(N_i) \times (N_j) \times (N_k)$. Thus the amount of memory used for calculation on graphics hardware is clearly a multiple of this number. Of course, some auxiliary variables have also to be taken into account.

The gist of numerical solution includes three main parts. In every integration step, the derivative (32) is calculated. The derivative value has to be determined four times during the integration phase due to the requirement imposed by the principle of the fourth-order Runge-Kutta method. The last stage of the algorithm is responsible for a correct handling of boundary conditions. First two stages are done in a sequential manner with one thread assigned for each equation inside the specified discrete point space. Subsequently, the boundary values are calculated for boundary points only (see equations (33) - (37)).

Both arrangements of work-groups within the index space and configuration of kernels inside each work-group have an impact on the resulting performance. The usage of one-dimensional indexing of kernel has an advantage of reducing floating point operation to obtain the kernel location. Thus, it is worth to use it instead of 2D or 3D indexing (the performed evaluation also confirmed that). On the other hand, one-dimensional index reduces the number of addressable kernels. The dimensions of the index space and work-groups, respectively, are $[65535, 65535, 1]$ and $[512, 512, 64]$. The highest number of threads in each work-group is therefore $2^{32} \times 2^9 = 2^{41}$ (maximum block size is 512). In case of one-dimensional index, the maximum number of threads is reduced to $2^{16} \times 2^9 = 2^{25}$, which has to be taken into account during implementation.

4.3 Form of ELLAM

The formulation of ELLAM framework was originated around 1990 by the authors Herrera and Ewing in a paper that appeared in Advances in Water Resources [6] where its superior performance was shown in one-dimensional ADE case with constant coefficients. Since that time the method was applied to many other more complex problems in 2 or 3 dimensions (see for instance [2], [31], [32], [19], [18]). In all of these cases, ELLAM performs well because it combines a Lagrangian approach for the advective terms with appropriate approximations, consistent with the Lagrangian framework, for other terms in the equations.

In this section, the description of concrete form which is used in the paper is described in more detail. Moreover, the techniques to avoid oscillations and to improve the method accuracy are described in appropriate subsection.

4.3.1 Basic concepts

The form of ELLAM framework used in this thesis is formulated for two-dimensional space domain (Ω). Leading by the presented contaminant models, it is designed for the advection-diffusion equation and for the simplicity it uses a rectangular grid. The concrete implementation is inspired by the work of Liu [17] where the space discretization is based on the finite element method. The governing equation in this case is defined as

$$\frac{\partial C}{\partial t} - \nabla \left(\vec{a} C - \vec{D} \nabla C \right) = S\left(\vec{x}, t\right), \quad \vec{x} \in \Re^2, t > 0, \quad (45)$$

where S (kg m^{-3} s^{-1}) is function of the source of the pollution, C (kg m^{-3}) is concentration, $\vec{a}$ (m s^{-1}) is velocity field, $\vec{D}$ (m^2 s^{-1}) is diffusion vector and t (s) is time. In the following equations, time t at step n will be denoted as t_n and previous time as $t_{n-1} = t_n - \Delta t$, where Δt is step size. The resulting weak formulation for the specified time t_n after multiplication by test function $g(\vec{x}, t)$ and applying of Green's formula is [17]

$$\begin{aligned} &\int_\Omega (Cg)(\vec{x}, t_n)\, d\vec{x} + \\ &\int_{J_n}\int_\Omega \left(\vec{D}\nabla C\right) \nabla g(\vec{x}, t)\, d\vec{x}dt + \\ &\int_{\Gamma_n} \left(\vec{a}C - \vec{D}\nabla C\right) \cdot \vec{n} g(\vec{y}, t) d\vec{y}dt = \\ &\int_\Omega (Cg)(\vec{x}, t_{n-1})\, d\vec{x} + \int_{\Sigma_n} S(\vec{x}, t)\, g(\vec{x}, t)\, d\vec{x}dt, \end{aligned} \tag{46}$$

where $\vec{n}$ is normal outward unit vector from the element $dydt$, J_n is time domain, $\Sigma_n = \Omega \times J_n$ is space-time domain, $\Gamma_n = \partial\Omega \times J_n$ is boundary domain, $d\vec{y}dt \in \Gamma_n$ and $g(\vec{x}, t_{n-1}) = \lim_{t \to t_{n-1}} g(\vec{x}, t)$. The second integral on the left hand side of the equation (46) is a diffusion term, the third integral is a boundary term and the second integral on the right hand side is a source term.

To evaluate the equation (46), the following procedure is done. The test function g was chosen as piece-wise linear with the following form

$$g^n_{i,j} = \begin{cases} \left(\frac{x - x_{i-1}}{\Delta x} + a_x \frac{t_n - t}{\Delta x}\right)\left(\frac{y - y_{j-1}}{\Delta y} + a_y \frac{t_n - t}{\Delta y}\right) \\ x^*_{i-1} \le x \le x^*_i, \\ y^*_{j-1} \le y \le y^*_j, t_{n-1} < t < t_n \\ \left(\frac{x_{i+1} - x}{\Delta x} + a_x \frac{t_n - t}{\Delta x}\right)\left(\frac{y_{i+1} - y}{\Delta y} + a_y \frac{t_n - t}{\Delta y}\right) \\ x^*_i \le x \le x^*_{i+1}, \\ y^*_j \le y \le y^*_{j+1}, t_{n-1} < t < t_n \\ 0 \\ \text{for other } x, t. \end{cases} \tag{47}$$

where (x^*_k, y^*_l) $(k \in \{i-1, i, i+1\}, l \in \{j-1, j, j+1\})$ are points at time t_{n-1} corresponding to points (x_k, y_l) at time t_n along the characteristic curve.

The terms with time integration are discretized using backward Euler method. The source term of the equation is approximated as

$$\begin{aligned} &\int_{\Sigma_n} S(\vec{x}, t)\, g(\vec{x}, t)\, d\vec{x}dt = \\ &\int_\Omega \Delta t^I(\vec{x}, t_n) S(\vec{x}, t_n) g(\vec{x}, t_n) d\vec{x} + E(S, g), \end{aligned} \tag{48}$$

where $\Delta t^I(\vec{x}, t_n) = t_n - t_{n-1}$ and $E(S, g)$ is error of the approximation. Similarly, the diffusion term can be approximated as follows

$$\begin{aligned} &\int_{J_n}\int_\Omega \left(\vec{D}\nabla C\right) \nabla g(\vec{x}, t) d\vec{x}dt = \\ &\int_\Omega \Delta t^I(\vec{x}, t_n) \left(\vec{D}\nabla C\right) \nabla g(\vec{x}, t_n) d\vec{x} + E(\vec{D}, C, g), \end{aligned} \tag{49}$$

where $E(\vec{D}, C, g)$ is approximation error.

Assuming no boundary terms in the implementation, the corresponding integrals and error terms can be dropped and therefore final equation for one element will have the form

$$\begin{aligned} &\int_\Omega (Cg)(\vec{x}, t)\, d\vec{x} + \\ &\int_\Omega \Delta t^I(\vec{x}, t_n) \left(\vec{D}\nabla C\right) \nabla g(\vec{x}, t_n) d\vec{x} = \\ &\int_\Omega (Cg)(\vec{x}, t_{n-1})\, d\vec{x} + \\ &\int_\Omega \Delta t^I(\vec{x}, t_n) g(\vec{x}, t_n) g(\vec{x}, t_n) d\vec{x}dt. \end{aligned} \tag{50}$$

The integrals in equation (50) are evaluated by numerical integration using Gaussian quadrature with appropriate integration points. It remains to evaluate the equation $g(\vec{x}, t_{n-1}) = \lim_{t \to t_{n-1}} g(\vec{x}, t)$. This problem leads to the solution of the ordinary differential equation back in time. The common integration methods such as Euler method can be used. The 4^{th}-order Runge-Kutta method is used in the current implementation. It is a trade-off between speed and accuracy and it behaved very well in the cases of the performed experiments.

The last thing to explain is the space discretization. The rectangular grid of points and the standard FEM process is used. The equation (50) has to be solved on the whole domain, therefore the elements, on which the approximation of the unknown function C is defined, have to be assembled together. This leads to the system of algebraic equations that has to be solved at each time step.

As a summary, the procedure of the ELLAM method can be expressed as:

- Initialization. Assembly of elements into global matrix.
- Loop. In each iteration step, till $t < t_{end}$ holds, do:
    1. assembly source term to right-hand side of set of equations.
    2. assembly the global right-hand side for the old mass (characteristic tracking to time t_{n-1}).
    3. optionally, modify global system with boundary condition term.
    4. solve global system of equations for $t = t_n$.

The evaluation of the described form of ELLAM is presented in section 5.3.

4.3.2 Oscillation reduction

ELLAM often suffers from excessive numerical oscillations around the exact solution. The standard approach for this problem is to use *mass lumping* which fixes the oscillations but increases numerical diffusion [2]. In work of Russel and Binning [25] the *selective mass lumping* procedure is designed for 1D case which significantly reduces the arisen numerical diffusion.

Next to these, there actually exist two works that particularly deal with the problem. In the first one [22], the authors use lumping technique and a post-processing procedure, a change of the matrix representing the final system of equations in a way to preserve monotonicity properties of the solution. They showed the very good results of global overshoot reduction on problems in two dimensions on structured and unstructered grids.

The second work which deals with the problem of excessive numerical diffusion added when using mass lumping techniques is the one by Younes at al. [33]. Here, the authors use mass lumping and keep the same characteristics during the entire simulation. At the end of each time step the diffusion part is added using the interpolation technique (only the diffusion part is interpolated). The excellent properties of the technique were demonstrated in one-dimensional test cases.

By inspiration of Walcek method and based on experience with the ELLAM simple experiments few techniques to reduce oscillations and/or improve ELLAM method accuracy were designed in this work. They are described further.

Simple flux limiting. The idea of the first method is to use the simple flux limiting approach similar to one in Walcek scheme where the simple rules presented in section 3.2 are applied. If Courant number is less than 1 the amount of transferred mass to receiving cell cannot be bigger or less than the maximum or minimum concentrations in both donor and receiver cells. If the violation of these restrictions is detected the amount of concentration changed accordingly.

With such procedure, there is no guarantee that total amount of mass will be preserved, a certain amount of concentration will be dropped off. The question is what to do with this residual. During simple experiments of advection equation in one dimension using sine- and box-shaped concentration profiles, it was observed that an approach of homogeneous distribution of residual gives the good results. Actually, the distribution is done selectively to the cell with the reasonable amount of concentration (at least 0.01% of peak concentration) or a big change of concentration with respect to previous time step.

The results for one-dimensional experiments of the described limiting method are shown in figure 1. It could be seen there that the excessive numerical oscillations are avoided with the minimum additional diffusion.

Using artificial diffusion. In general, all flux limiters add kind of artificial diffusion to the method in order to avoid oscillations. Therefore, why not to add selectively the diffusion to the currently running simulation when needed. Here is important to define proper criteria and proper amount of artificial diffusion in order to not add too much or not too less. The big amount of diffusion leads to inaccurate shape preserving in case of pure advection problems. On the other hand, too less diffusion has a consequence of still existing oscillations and thus inaccuracy or even instability. The similar approach was used in [25] to use mass lumping selectively to avoid large numerical diffusion.

The criteria of the amount of diffusion added used here is based on a size of oscillations detected in current time step during a simulation. The question is how to determine the oscillation size. Because it is physically impossible to reach negative concentrations, these are very good indications of oscillations. Secondly, when model describes pure advection or advection-diffusion process without source or sink terms the criteria of global maximum can be used. In this case it is physically impossible to get global concentration maximum higher than at each step of simulation. The criteria can be in general stated in the following form

$$oscillations \begin{cases} \text{if } C^i_{t_n} < -C^{max}_{t_n}\mu \\ \text{if } C^i_{t_n} > C^{max}_{t_n}(1+\mu), \end{cases} \tag{51}$$

where i is the cell index, t_n denotes time step n and $\mu \geq 0$ is the defined oscillation detection ratio. The size of the oscillations is important to final decision if the artificial diffusion has to be increased or not. In order to obtain a relative oscillation size (a ratio) o_r, a simple equation can be used

$$o_r = \max_i(o_i/c^{max}_{t_n}), \tag{52}$$

where i is the cell index. The actual algorithm to adjust the diffusion in current time step t_n is then done through the iteration process where the current diffusion coefficient is adjusted using bisection algorithm to find the optimal value.

The results of the presented technique are shown in figure 1 in the middle. It can be seen there that the case of sharp gradients indicates that big diffusion had to be used in order to avoid oscillations. The question here is whether the amount of diffusion was not set too high. In more advanced experiments it was shown that it could be better in many cases to keep smaller oscillations to preserve proper concentration shape. The actual results are shown in section 5.3.

Time-step adaptation. During implementation and testing of the form of ELLAM used in this thesis it was observed that bigger steps lead to significantly smaller oscillations and thus better accuracy. This is with accordance of the results and derivations in [25], [22] and [33]. Therefore, the idea to gain better accuracy and less oscillations is to use as big step as possible during simulation - dynamic step sizing.

The criteria when to use bigger time steps is very similar to the case of oscillation avoidance using artificial diffusion. When oscillations are detected using equation (51)

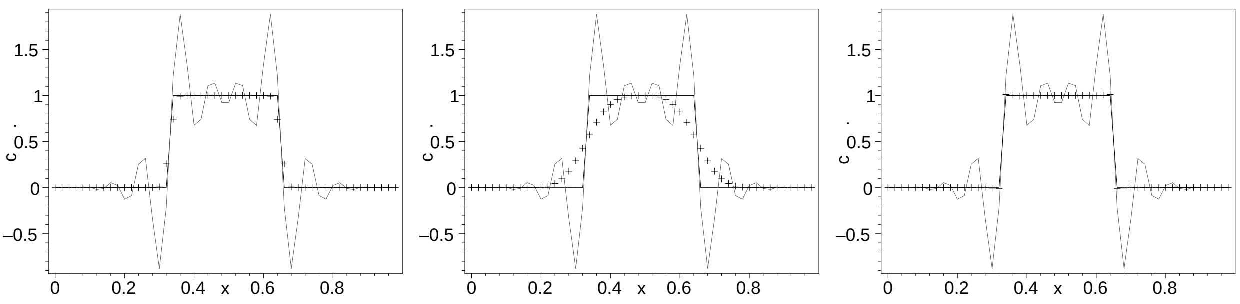

Figure 1: Test result of ELLAM scheme for box-shape initial profile with simple limiter (left), artificial diffusion (middle) and time-adaptation limiting techniques. Black solid line is the exact solution, grey solid line is the original scheme and cross are points of the scheme with the limiter.

and the oscillation ratio calculated by (52) is lower than in previous time step the step size is doubled. The procedure continuous till the condition holds. When the increase of step size leads to bigger oscillations than before the previous time step size is used.

It is very important to keep in mind that characteristic tracking has to remain accurate when step size is modified. Thus when it is increased the number of micro time steps of method for characteristic tracking has to be accordingly adjusted. One can use the technique similar to adaptive time-stepping as in case of MoL experiments with real coefficient models (see section 5.2.2). In current solution, the size of micro steps is deduced from the factor of time step increase. The actual micro-step size dt_n was empirically determined as

$$dt_n = \begin{cases} 1/(N_p * 1.35), & dt_n > dt_p \\ 1/(N_p/1.35), & \text{otherwise}, \end{cases} \tag{53}$$

where dt_p is micro step size from previous iteration and $N_p = 1/dt_p$ is number of previous micro steps.

5. Experiments

The designed methods described in sections 4.2 and 4.3 were put under extensive testing in various scenarios. The goal was first to verify the models against known analytical solution and secondly to use the real wind/diffusion models in order to prove whether the models are suitable for practical calculations.

The parallel versions of models whose solution is based on method of lines framework were tested in in three-dimensional space using simple artificial models for CUDA and OpenCL platforms. The models were also tested using real wind and dispersion models in single core CPU version.

Due to the complexity of ELLAM framework based models only two dimensional variants were implemented and tested in this work. First part of tests is dedicated to pure advection cases with rotating and divergent winds where the results of the method were compared with state of the art Walcek method described in section 4.1. Next set of experiments was done also with real advection-diffusion cases where the real models were used.

5.1 Error measurement and evaluation

The experiments serving to evaluate the models or the schemes, which are actually used to calculate their solution, differ a lot. Depending on the way what data does exist and what is the form of exact solution the approximation is compared to, the different error measures are defined.

The first one relates to law of mass conservation. As stated in equation (6), the final mass amount in the experiment should be the same as at the beginning. Taking the discretized domain into account, the mass conservation law for one-dimensional case and interval $[a, b]$ can be written in the form

$$M(t) = \sum_{j=a}^{b} w(x, t_0) = \sum_{j=a}^{b} w(x, t), \quad \text{for } t > t_0, \tag{54}$$

where $w(x, t)$ are values at points (x, t) calculated by the given scheme. The actual error of mass preservation (MassE) can be expressed as a ratio between mass at $t = t_0$ and mass at any $t > t_0$.

5.1.1 L_n-norm errors

Mathematically, a norm is a total size or length of all vectors in a vector space or matrices [23], [15]. Let $\vec{x} = (x_1, x_2, ..., x_M)$ is a vector in M-dimensional real space Ω ($x_j \in \mathcal{R}$). Then l_n norm of $\vec{x}$ is defined as

$$\|\vec{x}\| = \sqrt[n]{\sum_{j=1}^{M} |x_j|^n}. \tag{55}$$

Several concrete norms are actually used when evaluating the results (vector of values $\vec{r}$) of experiments when exact solution (vector of values $\vec{e}$) is known ($|\vec{r}| = |\vec{e}| = M$). The first one is l_1-norm, often called *Manhattan*, when the difference of two vectors is analysed

$$\|\vec{r} - \vec{e}\|_1 = \sum_{j=1}^{M} |r_j - e_j|. \tag{56}$$

It is better to normalize error norms to unit vector to overcome the problems of different vector lengths. Then

l_1-error, called *mean absolute error* (MAE), will have the form

$$MAE(\vec{r},\vec{e}) = \frac{1}{M}\|\vec{r}-\vec{e}\|_1 = \frac{1}{M}\sum_{j=1}^{M}|r_j - e_j|. \qquad (57)$$

Similarly to l_1-error, l_2-error, called *mean squared error* (MSE), can be defined as

$$MSE(\vec{r},\vec{e}) = \frac{1}{M}\|\vec{r}-\vec{e}\|_2^2 = \frac{1}{M}\sum_{j=1}^{M}(r_j - e_j)^2. \qquad (58)$$

The last l_n-error measure used in the thesis is l_∞-error. It is defined simply as the size of highest component in vector $\vec{r}-\vec{e}$, consequently

$$\|\vec{r}-\vec{e}\|_\infty = \max_{1\le j\le M}|r_j - e_j|. \qquad (59)$$

5.1.2 Concentration profile preserving

Lets consider the pure advection equation (2). Since there is no other part that can influence the mass movement throughout the volume than advection the initial profile shape has to be preserved for all $t > 0$ (assuming constant uniform velocity field). Therefore the solution of pure advection equation can be viewed as shifted initial profile of concentration in space. From this point of view, the following different error measures are often taken into account.

The first measure from this group is the *relative root mean squared error* (RMSE). The differences between the exact and calculated solution are normalized by the difference between peak and minimum concentration levels. The result is the number bigger or equal to zero where one means 100 percent error with respect to concentration interval among initial peak and initial zero levels. Let $\vec{r}$ be the vector of calculated values and $\vec{e}$ be the vector of exact values then RMSE for one-dimensional equation is calculated as

$$RMSE = \frac{\sqrt{\sum_{j=1}^{M}(r_j - e_j)^2/M}}{Peak_0 - Min_0}, \qquad (60)$$

where M is number of points in the domain, $Peak_0$ and Min_0 is peak and minimum concentration in the initial time.

The next error measure is the *peak error* (PeakE) represented by equation of the form

$$PeakE = 1 - \frac{Peak_c - Min_c}{Peak_0 - Min_0}, \qquad (61)$$

where $Peak_c$ and $Peak_0$ is the calculated and initial peak of the concentration, Min_c and Min_0 is the calculated and initial minimum of the concentration level.

The *mass distribution ratio* (DistrE) represents the shape preservation without reference to the advected shape. The algorithm can, e.g., nicely preserve shapes but it shifts the position of the shape to a wrong place. Thus its RMSE error would be relatively high. On the other hand the distribution error would be much smaller. The distribution error for one-dimensional case is defined as

$$DistrE = 1 - \frac{\sum_{j\in\Omega_j} r_j}{\sum_{j\in\Omega_j} e_j}, \qquad (62)$$

where Ω_j refers to domain where r_j and e_j differs from Min_0.

5.1.3 Error measures for real experiments

The experiments with the real data consists of only few places where the final concentrations were measured. Therefore, the special measures are taken into account when calculated solution is evaluated using the experimental data [14].

First one is the *normalized mean square error* (NMSE) which represents the quadratic error of the predicted quantities in relation to the observed ones. Best result is indicated by values nearest to 0. NMSE has the following form

$$NMSE = \frac{\overline{(C_o - C_p)^2}}{\overline{C_o}\cdot\overline{C_p}}, \qquad (63)$$

where C (kg m^{-3}) is concentration with subscripts with the meaning of: o - observed, p - predicted. The overbar determines the average of a quantity. The used symbols in equation (63) are also used in the following equations.

The second statistical index is represented by the *correlation coefficient* (COR). The correlation is maximal when it reaches 1 and is defined as

$$COR = \frac{\overline{(C_o - \overline{C_o})(C_p - \overline{C_p})}}{\overline{\sigma_o\sigma_p}}, \qquad (64)$$

where σ_o (kg m^{-3}) and σ_p (kg m^{-3}) are standard deviations of observed and predicted quantities.

The *fractional of data* (FA2) states the amount of samples that are within the defined space

$$FA2 \approx 0.5 \le \frac{C_p}{C_o} \le 2. \qquad (65)$$

The *fractional bias* (FB) denotes whether the predicted quantities underestimate or overestimate the observed ones (the closer to 0 the better)

$$FB = \frac{\overline{C_o} - \overline{C_p}}{0.5(\overline{C_o} + \overline{C_p})}. \qquad (66)$$

The last one is the *fractional standard deviation* (FS) that indicates the statistical precision as a fractional result and again if it is closer to 0 the results are more precise

$$FS = \frac{\sigma_o - \sigma_p}{0.5(\sigma_o + \sigma_p)}. \tag{67}$$

5.2 Method of lines

This section describes the experiments done with the designed numerical models that are defined in section 4.2.

5.2.1 Parallel versions - CUDA/OpenCL

The evaluation of model parallelization followed. The motivation is to improve the model computational speed in order to use more equations, and thus better accuracy within the similar or even smaller computational time.

The experiments were done on a bit older CPU and GPU platforms, however, the results still show great potential to use GPU for general purpose computation. It has to be noted that all computations were done using 32 bit floating point arithmetic [1] due to the lack of support of bigger precision on tested GPUs.

CUDA experiments. The parallel version of the model described by equation (32) was evaluated by CUDA framework using three different memory access scenarios. All tests were performed on a CPU and GPU separately. The experimental setup consisted of CPU Intel Core 2 Duo at 2.66 GHz. The selected graphic devices were GeForce 9600M GT as a representative of mobile devices, GeForce 8800 Ultra and new GeForce GTX280.

The model 9600M GT has 32 cores (laid out as 4 multiprocessors with 8 units) at a clock rate of 1.25 GHz. There are 8192 registers available per block. Model 8800 Ultra has 128 cores (16 multiprocessors with 8 processors) with clock rate of 1.5 GHz. Again, there are 8192 registers to be used for each block. GTX280, the last model used for evaluation, has 240 cores (30 multiprocessors with 8 processors). This time, there are 16384 registers available for each thread block.

Three different approaches how to use the graphics memory were implemented. The first method (A) used only global memory of graphics card which is directly accessible by CPU. The second method (B) used shared memory of GPU to store the attributes of ODE system to have faster access from kernels running on GPU. The last method (C) uses texture memory with cached access for the equation values from a previous step (it is accessed 4 times during integration phase in each thread). Moreover, shared memory contained a number of auxiliary variables of 4^{th}-order Runge-Kutta method. The final results of the fastest method (C) are shown in table 1.

CUDA/OpenCL comparison experiments. The second set of experiments was done on the same version of equation. The testing application was written entirely in C++ language where the individual versions have been prepared for OpenCL and CUDA frameworks alike. As in the previous case, all experiments were performed on a CPU and GPU separately. The reference performance indicators are specified in case of single-thread application running on CPU Core 2 Duo at 2.267 GHz. The entire set of the following measurement is compared against these initial values.

All data processed by the computation kernel were read from a global memory on GPU card for each step of computation. However, efficient usage of local and private memories (as referred to in OpenCL specification) during computation process, together with overlay of asynchronous data transfers, helps to mitigate inherent latency.

Here, the experimental setup consisted of two CPUs: Intel Core 2 Duo at 2.267 GHz with 3 MB of L2 cache and Intel Core 2 Quad at 2.66 GHz with 6 MB of L2 cache. In addition, the following GPUs were used during experiments: GeForce GT9600M as a representative of mobile GPU, GeForce GTX285 as the high-end platform from nVidia and finally ATI HD5870.

The model GTX285 has 240 cores (30 multiprocessors with 8 processors) at a clock rate of approximately 1.5 GHz. This time, 16384 registers are available for each thread block. Last example of GPU was ATI HD5870 which has 1600 cores organized into 20 so called SIMD engines, where each of them works at a clock rate of 850 MHz.

The final experiment results are summarized in table 2 where the relative speed-ups against single threaded version running on Intel Core 2 Duo at 2.267 GHz are presented. There, an interesting fact could be noted with Dual Core CPU. When multiple threads were used in case of OpenCL the performance was worse than in single thread version. On the other hand, processing time on the quad core CPU is significantly lower. Furthermore, GPU platforms performed as expected. The column in table 2, which contains the values of relative speed-up in case of 512 threads per block, contains x mark for GeForce GT9600M. The reason is that application fails to be launched due to critical lack of resources.

5.2.2 Real models

In the next set of experiments, the real models of wind and dispersion were intended to use. The models were evaluated in correspondence with the Copenhagen experiments [12], [13] where sulphurhexafluoride substance was used. The tracer was released without buoyancy from a tower at a height of 115 meters and then collected 2-3 meters above ground-level at positions in up to three crosswind arcs of tracer sampling units, positioned 2-6 km from the point of release. Three consecutive 20 min averaged tracer concentrations were measured, allowing for a total sampling time of 1 hour. The site was mainly residential having a roughness length of 0.6 m.

There were 9 experiments performed in Copenhagen, in which all of the required parameters were measured. The all parameters of the experiments that were used for calculations are shown in table 3.

The experiments were done using the discretization scheme described by equation (41) with the appropriate boundary conditions (44). The spatial axis z was discretized using the same step of size 8 m from ground to height of planetary boundary layer H (m) defined individually for each experiment. The ODE was solved using the fourth-

Platform	Block size								
	1	2	4	8	16	32	64	128	256
9600M GT	0.44	1.01	2.29	4.42	8.34	11.97	14.01	14.29	12.22
8800 Ultra	2.30	5.24	11.59	21.64	39.92	56.49	68.86	65.01	57.11
GTX280	4.67	10.59	22.61	40.57	74.08	105.67	115.58	115.87	115.61

Table 1: Time-dependent model - the comparison of computational speed-up expressed as ratios of GPU and CPU calculation times.

Platform	Block size				
	32	64	128	256	512
OpenCL CPU (P8400, 2.26 GHz)	0.91	0.89	0.90	0.87	0.45
OpenCL CPU (Q9400, 2.66 GHz)	2.09	2.09	2.06	2.12	0.54
CUDA GPU (GT9600M)	14.14	15.25	14.93	14.88	13.98
OpenCL GPU (GT9600M)	11.05	13.77	12.55	12.05	x
CUDA GPU (GTX285)	127.15	144.85	146.27	144.54	140.68
OpenCL GPU (GTX285)	122.87	136.67	136.93	130.97	126.31
OpenCL GPU (HD5870)	68.01	92.41	97.57	96.80	160.52

Table 2: The comparison of computational speed-up expressed as ratios of GPU and CPU calculation times. 262144 equations were calculated simultaneously in each step.

Exp. num.	h_s (m)	H (m)	L (m)	$a_\star$ (m s)$^{-1}$	$w_\star$ (m s)$^{-1}$	K	z_0 (m)
1	115	1980	-46	0.37	1.70	0.4	0.6
2	115	1920	-384	0.74	1.80	0.4	0.6
3	115	1120	-108	0.39	1.10	0.4	0.6
4	115	390	-173	0.39	0.74	0.4	0.6
5	115	820	-577	0.46	2.50	0.4	0.6
6	115	1300	-569	1.07	2.00	0.4	0.6
7	115	1850	-136	0.65	2.10	0.4	0.6
8	115	810	-72	0.70	2.10	0.4	0.6
9	115	2090	-382	0.77	2.00	0.4	0.6

Table 3: The parameters of the performed experiments in Copenhagen [13].

order Runge-Kutta method with varying step size which were adopted in each step to meet the criteria of local error less than $1e^{-4}$. The concentration amount was collected from 2-3 meters during the real experiments and because the closest grid points in the experiments were in 0 m and 8 m heights the concentration values in 2 m were interpolated.

It should be noted that MoL scheme represented by equation (41) contains the derivative of the diffusion function according to z. It can be obtained either in exact form or, if impossible, as numerical approximation. The treatment in concrete model cases will be shown in the following subsection.

Wind and Turbulent Parametrization. The wind speed in both profiles used in the experiments is dependent on height z variable and other meteorological parameters measured at site. The two concrete profiles in case of the height of the unstable boundary layer $H = 1980$ m (Copenhagen experiment number 9) are shown in figure 2.

The turbulent parametrization was done using three models. First two models where the diffusion is dependent on height z [30], [27] are shown in the top graph in figure 3.

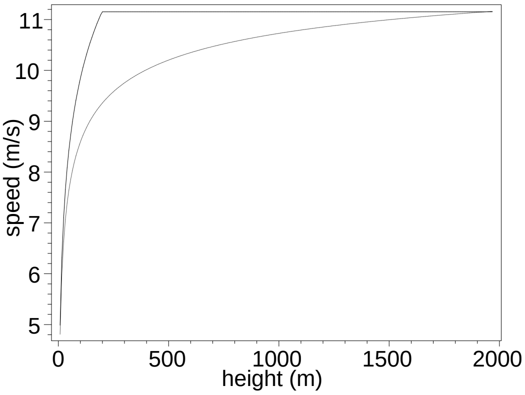

Figure 2: The real wind profiles used in the experiments - the model from [30] (black) and the model from [27] (grey).

The third model [9] dependent on height z and downwind distance from the source x is shown in its bottom graph.

As it was outlined above, the derivatives of diffusion equations have to be known in order to solve the numerical schemes. The algebraic form of the derivative was used in case of [30], [27] models, the third one [9] was approximated by the central difference.

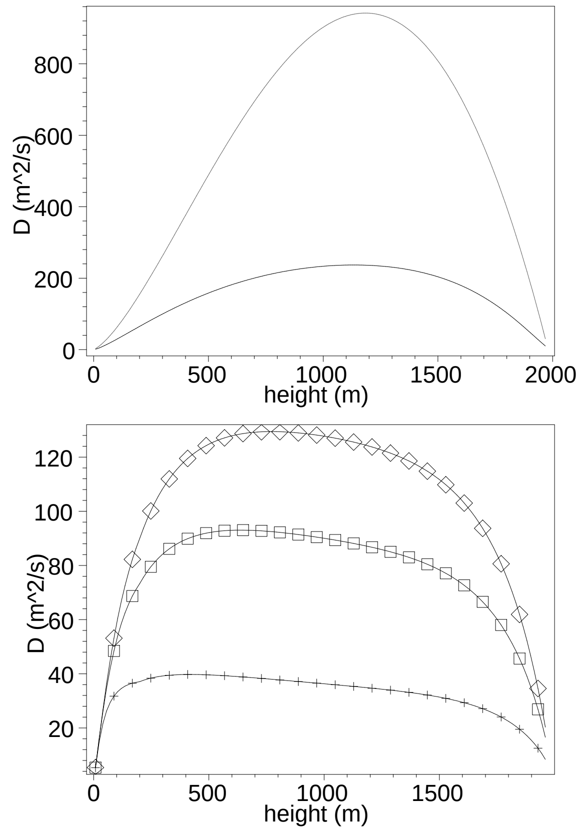

Figure 3: The real turbulent profiles used in the experiments. The turbulent profiles by [30] and [27] are shown in the top graph (black and grey). The turbulent profile by [9] for the downwind distances of 1000 m (cross), 3000 m (box) and 5000 m (diamond) are shown in the bottom graph.

Scheme	MAE	MSE	L_∞-error
MoL	$0.840 \cdot 10^{-1}$	$0.136 \cdot 10^{-1}$	0.274

Table 4: L_n-norm error measures of performed experiments with Wortmann turbulent parametrization where analytical solution is known.

Results. The first experiment was done using the the wind and turbulent parametrization by Wortmann et al. [30] where the analytical solution is known. The analytical method which basic description is in section 4.2.2 was implemented. The results from numerical model were evaluated and the corresponding L_n-norm error measures are shown in table 4. There, the very good correspondence between the two solutions could be seen.

The rest of experiments were done only using numerical solution of the defined models. Figure 4 shows the comparison between predicted and measured crosswind-integrated concentrations C (kg m^{-2}) in all experiment cases. For clarity, the concentrations are normalized according to source term and divided by 10^4: $C_{o|p} = 10^{-4}C/Q$, with units of (kg m^{-2})/(kg s^{-1}) = s m^{-2}. The ideal state would be if the points lie on the middle line. The other two lines border the space of factors 0.5 and 2 (see equation (65)) and it is seen that all the predicted values lie in this range.

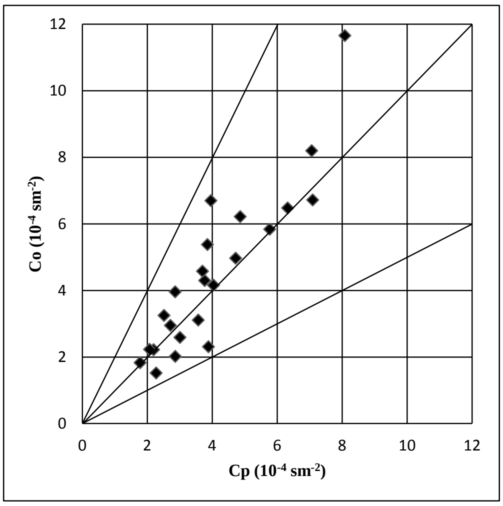

Figure 4: The comparison of the measured concentrations (C_o) and the predicted concentrations (C_p) using the dispersion parametrization by Wortmann [30].

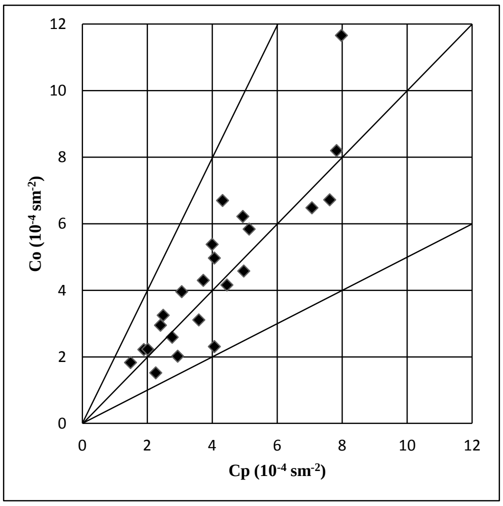

Figure 5: The comparison of the measured concentrations (C_o) and the predicted concentrations (C_p) using the dispersion parametrization by Degrazia [9].

For the second set of experiments the turbulent parametrization dependent on height and downwind distance was chosen. This overall model is more accurate than the one presented before. The results are shown in figure 5.

The measured statistical indices are shown in table 5. In the first two rows the statistics for the analytical and numerical solutions of Wortmann's model [30] are stated showing almost the same accuracy. The best results, shown in the last row, were achieved by using turbulent parametrization by Degrazia [9]. As a conclusion, it can be noted that the model results correspond to the observed concentration levels very well.

Model	NMSE	COR	FA2	FB	FS
Analytical (W)	0.08	0.90	1.00	0.11	0.32
MoL (W)	0.08	0.90	1.00	0.10	0.33
MoL (D)	0.08	0.88	1.00	0.08	0.23

Table 5: The statistical indexes values of all performed experiments with real wind/turbulent parametrization.

5.3 ELLAM framework

This section describes the experiments done with the original ELLAM method inspired by [17] and its modified version using adaptation techniques presented in section 4.3.2. The first set of experiments was done for pure advection cases using artificial rotating wind model. Here the exact solution is known and the computed values are directly evaluated. The second set of experiments was done for artificial divergent wind model where the global mass conservation was studied. The last set of experiments is dedicated to advection-diffusion phenomena with source term where the real models of advection and diffusion terms are used in order to show the suitability of the framework in this application.

5.3.1 Rotation wind

Rotation wind tests are quite common techniques to show the performance of the numerical schemes solving pure advection equations. Here, the domain is in two dimensions and has square shape. The wind rotates at a constant rate during the whole simulation. Under these conditions, it is obvious that concentration profile of any initial shape has to remain the same at the end of each simulation if the concentration is zero at the domain boundaries and thus it is not out-flowed away.

Configuration. As it is stated further, different initial concentration shapes cause smaller or bigger problems to the tested schemes which were used. The relatively easy shape is of non-steep cone. The moderate difficulties are caused by the cylindrical shape where vertical gradients are presented but horizontally the shape is smooth. The hardest shape also used in these tests is a slotted cylinder, i.e. cylinder from which the box shape is subtracted. Thus the shape has both vertical and horizontal discontinuities.

The experiment settings were as follows. All tests were done in a squared space which was divided into 100 × 100 points. The diameters of the initial shapes were set to 30 points for all tests. The time steps were set to 360 per one rotation, i.e., the Courant number was less than one in case of Walcek algorithm. On the other hand the time step was set to 8^{th} and 24^{th} multiple of Walcek setting in case of original ELLAM. The step size of ELLAM algorithm was chosen in this way to reach the approximately same calculation time as in case of Walcek scheme. Obviously, the adapt version of ELLAM sets its step size to necessary values to reach the (sub-)optimal time stepping to gain significantly smaller oscillations and as small numerical diffusion as possible.

All the results in rotating experiments were evaluated using the error measurements defined in section 5.1 in its *Concentration profile preserving* subsection.

Cone profile. The tests with a cone shape profile were done for short-, mid- and long-term simulations represented by 1, 6 and 60 rotations. The results at the end of the appropriate simulations are shown in table 6. Obviously, all schemes preserve mass very well. Since original ELLAM performs with cone profile very well and adapt version of ELLAM adds some diffusivity the peak error is smaller in case of original ELLAM scheme. RMSE and DistrE are smaller in both ELLAM schemes.

Cylinder-based profiles. The experiments with the cylinder-based initial profiles were done under the same conditions as in the previous case. The number of rotations was set to 1, 6 and 60. The results for cylinder and slotted cylinder are shown also in table 6. The results show again the similar very good mass preserving of all numerical schemes, Walcek has the smallest peak error and ELLAM and Adapt. ELLAM have significantly smaller RMSE and DistrE. Moreover, Adapt. ELLAM has significantly lower PeakE than original ELLAM scheme.

5.3.2 Divergent wind

The last set of artificial experiments was done with divergent wind model that was presented also in [29]. The wind speeds along x and y axes are computed using the following equations

$$a_x = \sin\left(\frac{\pi i}{25}\right)\sin\left(\frac{\pi j}{25}\right), \tag{68}$$

$$a_y = \cos\left(\frac{\pi(i+d_i)}{25}\right)\cos\left(\frac{\pi(j+d_j)}{25}\right), \tag{69}$$

where i and j are cell indexes along x and y axes and d_i and d_j lying in $[0, 1]$ interval are the displacements of the wind in y direction. The part of the velocity field is shown in figure 6 where also the base of the used initial profiles is displayed. It is presented there that the wind blows in circles within squares of size 25 × 25 cells. The maximum wind speed was set to 10 and it is reached at edges of the squares. The zero velocity of the wind is presented in the middle of the squares.

At long times after initialization, the concentration distribution becomes sheared within the swirls into infinitesimal "curtains" or sheets which wrap around one another while becoming thinner, and therefore are not resolved by a numerical grid mesh of $(100\Delta x)^2$ [29]. Therefore, if the exact solution is averaged over $(\Delta x)^2$ areas at long times after initialization, the tracer should become uniformly mixed along streamlines of the swirls, and have a ring-like structure within each vortex that contained any of the tracer initially.

The first set of experiments were done according to one performed in [29] with cone initial profile which center was placed at $[50, 50]$, i.e. in the middle of the domain 100 × 100 as it is shown in figure 6. The element size was set to 40000×40000 m^2 and the time step was set to 2637.6 s in case of Walcek method. Two cases of ELLAM were tested as before - original ELLAM version according to [17] with 24 time bigger time step than Walcek and Adapt. ELLAM which used adaptive time steps in combination with adaptive diffusion.

Scheme	MassE	RMSE	PeakE	DistrE
Walcek	0.118×10^{-10}	0.120×10^{-1}	0.414×10^{-1}	0.311×10^{-1}
ELLAM	0.905×10^{-4}	0.221×10^{-2}	0.607×10^{-1}	0.313×10^{-2}
Adapt. ELLAM	0.142×10^{-4}	0.321×10^{-2}	0.755×10^{-1}	0.206×10^{-3}
Walcek	0.151×10^{-9}	0.712×10^{-1}	0.0	0.162
ELLAM	0.511×10^{-4}	0.483×10^{-1}	0.889×10^{-1}	0.523×10^{-1}
Adapt. ELLAM	0.507×10^{-4}	0.537×10^{-1}	0.386×10^{-1}	0.117
Walcek	0.971×10^{-11}	0.822×10^{-1}	0.0	0.232
ELLAM	0.503×10^{-4}	0.591×10^{-1}	0.153	0.998×10^{-1}
Adapt. ELLAM	0.327×10^{-3}	0.497×10^{-1}	0.454×10^{-1}	0.991×10^{-1}

Table 6: The mean errors of all performed experiments with cone (first three rows), cylinder (middle three rows) and slotted cylinder (last three rows) initial profiles.

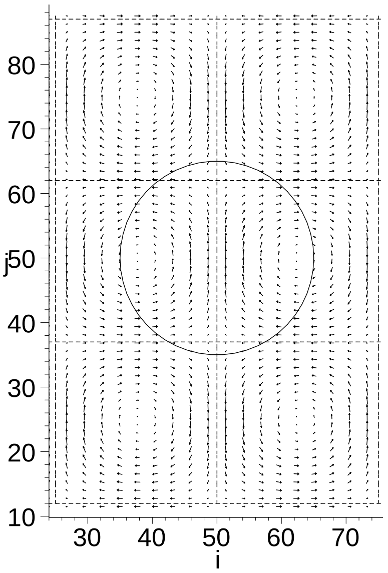

Figure 6: The velocity field of divergent wind used in experiments displayed with the base of the used concentration profile shapes.

Model	Init. profile	Scheme	MassE
O. wind	Cone	Walcek	0.339×10^{-3}
O. wind	Cone	ELLAM	0.132
O. wind	Cone	A. ELLAM	0.384×10^{-1}
O. wind	Cylinder	Walcek	0.641×10^{-2}
O. wind	Cylinder	ELLAM	0.157
O. wind	Cylinder	A. ELLAM	0.118
O. wind	Sl. cyl.	Walcek	0.658×10^{-2}
O. wind	Sl. cyl.	ELLAM	0.173
O. wind	Sl. cyl.	A. ELLAM	0.123
Sh. wind	Cone	Walcek	0.386
Sh. wind	Cone	ELLAM	0.237
Sh. wind	Cone	A. ELLAM	0.122
Sh. wind	Cylinder	Walcek	0.381
Sh. wind	Cylinder	ELLAM	0.256
Sh. wind	Cylinder	A. ELLAM	0.209
Sh. wind	Sl. cyl.	Walcek	0.394
Sh. wind	Sl. cyl.	ELLAM	0.277
Sh. wind	Sl. cyl.	A. ELLAM	0.215

Table 7: MassE measures of all performed experiments with cone, cylinder and slotted cylinder initial profiles. First 9 rows show the results for original wind model, the last 9 rows show the results where shifted wind model was used.

Summary. As a summary of this section, the results of relative MassE measured in each experiment are presented. The same experiments using divergent wind were done also with cylinder and slotted cylinder initial profiles with the same base diameter. In these cases the two divergent winds were also used - the original one and the shifted one.

The final results in the form of relative MassE are shown in table 7. It is clearly visible that Walcek has the best results in case of original divergent wind. On the other hand, it shows its flaws when shifted wind profile is used where it has bigger relative MassE. It is also evident that Adapt. ELLAM has smaller error than original ELLAM in all performed experiments.

5.3.3 Real Advection-Diffusion Models

The last set of experiments of ELLAM framework was done for real advection-diffusion models that were also used in the implementation of MoL method. As the input, the data from nine experiments performed in Copenhagen was used (see table 3). The wind parametrization from [30] (see figure 2 (black)) was used and the dispersion parametrization was used by [30] and [27] (see the top graph in figure 3 (black and grey)).

The testing models were similar as the ones used in section 5.2.2. The models used here were MoL, ELLAM and Adap. ELLAM. The form of the MoL designed in section 4.2.2 with up-winding was used during testing. ELLAM and Adapt. ELLAM models had the same form as the ones with the same name used previously in this section extended with point source term. All of the models were time-dependent with steady-state point source placed at the coordinates as it was in case of performed Copenhagen experiments. Therefore, the condition to stop the simulation was defined in the following way. The simulation stops when steady concentration level is reached in all places inside the domain.

The domain was discretized to 10 m × 10 m squares in vertical and horizontal directions. It means that for instance the space of Copenhagen experiment No. 1 with collecting distance of 1900 m from the source and mixing atmospheric height of 1980 m was discretized into 199

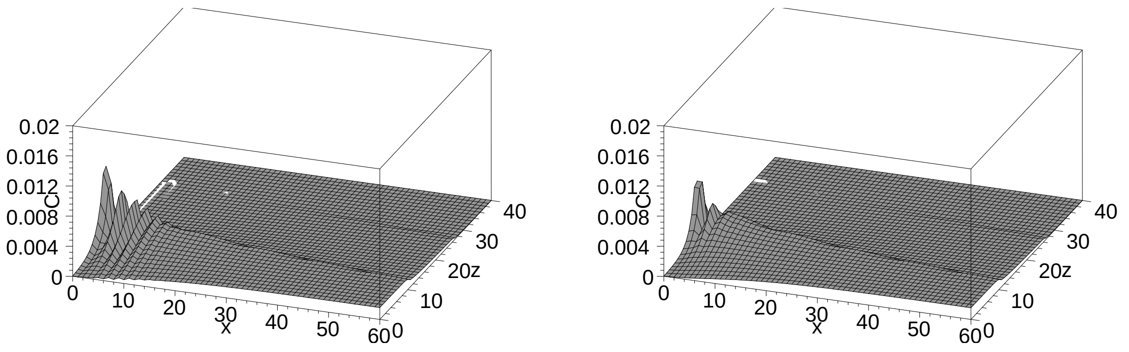

Figure 7: The results of simulation of Copenhagen experiment no. 9. The details of concentration near the source are shown for ELLAM (left) and Adap. ELLAM (right) methods. The units are 10^1 m in case of x and z axes, and 10^{-4} sm^{-2} in case of C axis.

Model	NMSE	COR	FA2	FB	FS
MoL (W)	0.08	0.90	1.00	0.11	0.32
MoL (U)	0.21	0.91	1.00	0.33	0.40
ELLAM (W)	0.14	0.83	0.96	0.18	0.32
ELLAM (U)	0.30	0.82	0.87	0.34	0.48
A. ELLAM (W)	0.15	0.82	0.96	0.19	0.33
A. ELLAM (U)	0.29	0.82	0.83	0.37	0.48

Table 8: The statistical indexes values of all performed experiments with real wind/turbulent parametrization.

points in case of MoL or 198 cells in case of ELLAM methods. The time stepping was set to 4 s at the beginning which remains the same only in case of ELLAM method, the other two methods used adaptive time stepping as described before.

The important fact is that the accurate point source could be only used in MoL approach since in ELLAM case the average concentrations in the cells are used and thus the source was actually area-based with 10 m^2 and thus this condition differs in the models. This influenced the results between MoL and ELLAM. The results are also influenced by the fact that there were no boundary condition terms (Neumann near the ground) implemented in case of ELLAM methods.

The results of performed experiments are shown in table 8. The results confirmed the assumptions that the MoL and ELLAM computed concentrations differ. However, the two ELLAM methods agreed with each other very well.

There was also an interesting observation in terms of the calculation speed. The overall time to calculate the final results of all experiments differed in all methods. The fastest was ELLAM (3.46 hours), Adapt. ELLAM was the second fastest (3.89 hours) and the slowest was MoL (5.33 hours). Adapt. ELLAM was slightly slower because of the extra calculations needed for the time-stepping and diffusion adaptations. MoL used adaptive time-stepping in order to maintain accuracy and finally leaded to the big calculation time.

From the results, it can be deduced that there is no advantage of Adapt. version of ELLAM. The fact that samples of concentrations were collected from far distances from the source leads to the very similar results of the two ELLAM methods. However, the difference of the methods is seen near the source of the pollution where due to relative big time-steps the oscillation of ELLAM appears. An example is shown in figure 7 on left where it is clearly visible that original ELLAM could suffer from oscillations and that better results are provided by Adapt. ELLAM although it does not compensate oscillations completely. The problem here is that not all the types of oscillations are detected due to presence of the point source and thus the concentration nearby is the highest.

6. Conclusions

This extended abstract of my dissertation thesis deals with numerical solutions of advection-diffusion equation describing the pollutant dispersion in planetary boundary layer. The work contains studies of the two concrete methods, method of lines and ELLAM. It was shown in artificial and real models that both methods are suitable for the models of local/urban scales. The experiments of method of lines and its parallel versions show great potential to be used on modern graphic cards.

Quite young ELLAM method was the second topic to study in this thesis. The certain form of ELLAM was implemented for structured grid which was directly compared in various experiments of pure advection problems with Walcek's method. The result showed that ELLAM performed very well, especially its adaptive version which prevents the oscillations the original form suffered from.

Acknowledgements. This work was partially supported by the BUT FIT grants Information Technology in Biomedical Engineering, GA102 / 09 / H083 and the EU / Czech IT4 Innovations Centre of Excellence project CZ.1.05 / 1.1.00 / 02.0070, and Advanced secured, reliable and adaptive IT, FIT-S-11-1.

References

[1] IEEE standard for floating-point arithmetic. *IEEE Std 754-2008*, pages 1–70, Aug 2008.

[2] P. Binning and M. A. Celia. A forward particle tracking Eulerian-Lagrangian Localized Adjoint Method for solution of the contaminant transport equation in three dimensions. *Advances in Water Resources*, 25(2):147 – 157, 2002.

[3] A. Bott. Monotone flux limitation in the area-preserving flux-form advection algorithm. *Monthly Weather Review*, 120(1):2592 – 2602, 1992.

[4] T. Brandvik and G. Pullan. Acceleration of a 3D Euler solver using commodity graphics hardware. In 46^{th} *AIAA aerospace sciences meeting and exhibit*, page 607. Curran Assoc., Inc., 2008.

[5] P. J. H. Builtjes. Major twentieth century milestones in air pollution modelling and its application. In S.-E. Gryning and F. Schiermeier, editors, *Air Pollution Modeling and Its Application XIV*, pages 3–16. Springer US, 2001.

[6] M. A. Celia, T. F. Russell, I. Herrera, and E. R. E. An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation. *Advanced Water Resources*, 13:187–206, 1990.

[7] N. Corporation. *NVIDIA CUDA C Programming Guide*, February 2014.

[8] K. Datta, M. Murphy, V. Volkov, S. Williams, J. Carter, L. Oliker, D. Patterson, J. Shalf, and K. Yelick. Stencil computation optimization and auto-tuning on state-of-the-art multicore architectures. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, page 4. IEEE Press, 2008.

[9] G. A. Degrazia, D. M. Moreira, C. R. J. Campos, J. C. Carvalho, and M. T. Vilhena. Comparison between an integral and algebraic formulation for the eddy diffusivity using the Copenhagen experimental dataset. *Il Nuovo Cimento*, 25:207–218, 2002.

[10] D. L. Ermak. An analytical model for air pollutant transport and deposition from a point source. *Atmospheric Environment (1967)*, 11:231–237, 1977.

[11] K. Group. *The OpenCL Specification*, March 2014.

[12] S. E. Gryning and E. Lyck. Atmospheric dispersion from elevated sources in an urban area: Comparison between tracer experiments and model calculations. *Journal of Climate and Applied Meteorology*, 23:651–660, 1984.

[13] S. E. Gryning and E. Lyck. The Copenhagen tracer experiments: Reporting of measurements. Technical report, 1998.

[14] S. R. Hanna. Confidence limits for air quality model evaluations, as estimated by bootstrap and jackknife resampling methods. *Atmospheric Environment*, 23:1385–1398, 1989.

[15] W. Hundsdorfer and J. G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag, 2007.

[16] B. Koren. A robust upwind discretization for advection, diffusion and source terms. In C. Vreugdenhil and B. Koren, editors, *Numerical Methods for Advection - Diffusion Problems*, pages 117–138. Friedrich Vieweg & Sohn Verlagsgesellschaft mbH, 1993.

[17] J. Liu. The white paper on ELLAM implementation in C++. Technical report, 2009.

[18] J. Liu, S. Tavener, and H. Chen. ELLAM for resolving the kinematics of two-dimensional resistive magnetohydrodynamic flows. *Journal of Computational Physics*, 227:1372–1386, Dec. 2007.

[19] W. F. Matthew, E. K. Christopher, F. R. Thomas, and T. M. Cass. An ELLAM approximation for advective-dispersive transport with nonlinear sorption. *Advances in Water Resources*, 29(5):657 – 675, 2006.

[20] P. Micikevicius. 3D finite difference computation on GPUs using CUDA. In *Proceedings of* 2^{nd} *Workshop on General Purpose Processing on Graphics Processing Units*, pages 79–84. ACM, 2009.

[21] F. Molnar Jr, T. Szakaly, R. Meszaros, and I. Lagzi. Air pollution modelling using a graphics processing unit with CUDA. *Computer Physics Communications*, 181(1):105–112, 2010.

[22] T. Neubauer and P. Bastian. On a monotonicity preserving Eulerian-Lagrangian localized adjoint method for advection-diffusion equations. *Advances in Water Resources*, 28(12):1292 – 1309, 2005.

[23] K. Rektorys. *Přehled užité matematiky: I [Survey of applicable mathematics: I]*, volume 7. Prometheus, 2000.

[24] K. Rektorys. *Přehled užité matematiky: II [Survey of applicable mathematics: II]*, volume 7. Prometheus, 2000.

[25] T. F. Russell and P. Binning. Oh no, not the wiggles again! A revisit of an old problem and a new approach. In W. G. G. Cass T. Miller, Matthew W. Farthing and G. F. Pinder, editors, *Computational Methods in Water Resources: Volume 1*, volume 55, Part 1 of *Developments in Water Science*, pages 483 – 494. Elsevier, 2004.

[26] S. A. Socolofsky and G. H. Jirka. *Environmental Fluid Mechanics 1: Mixing and Transport Processes in the Environment*. Texas A&M University, 5^{th} edition, 2005.

[27] A. G. Ulke. New turbulent parameterization for a dispersion model in the atmospheric boundary layer. *Atmospheric Environment*, 34(7):1029–1042, 2000.

[28] D. A. Vallero. *Fundamentals of Air Pollution (Fourth Edition)*. Elsevier, 2008.

[29] C. J. Walcek and N. M. Aleksic. A simple but accurate mass conservative, peak-preserving, mixing ratio bounded advection algorithm with FORTRAN code. *Atmospheric Environment*, 32(22):3863 – 3880, 1998.

[30] S. Wortmann, M. T. Vilhena, D. M. Moreira, and D. Buske. A new analytical approach to simulate the pollutant dispersion in the PBL. *Atmospheric Environment*, 39(12):2171–2178, 2005.

[31] A. Younes and P. Ackerer. Solving the advection-diffusion equation with the Eulerian-Lagrangian localized adjoint method on unstructured meshes and non uniform time stepping. *Journal of Computational Physics*, 208:384–402, 2005.

[32] A. Younes, P. Ackerer, and F. Lehmann. A new efficient Eulerian-Lagrangian localized adjoint method for solving the advection-dispersion equation on unstructured meshes. *Advances in Water Resources*, 29:1056–1074, 2006.

[33] A. Younes, M. Fahs, and P. Ackerer. A new approach to avoid excessive numerical diffusion in Eulerian-Lagrangian methods. *Communications in Numerical Methods in Engineering*, 24(11):897–910, 2008.

Selected Papers by the Author

R. Dvorak, F. Zboril. On the Usage of ELLAM to Solve Advection-diffusion Equation Describing the Pollutant Transport in Planetary Boundary Layer. In *Acta Electrotechnica et Informatica*, volume 12, pages 44–48. 2012.

R. Dvorak, F. Zboril. On the Usage of ELLAM to Solve Advection-diffusion Equation Describing the Pollutant Transport in Planetary Boundary Layer. In *Proceedings of the Eleventh International Conference on Informatics*, pages 227–230, Kosice, Slovakia, 2011.

R. Dvorak, F. Zboril. Preliminary Study of Using Ellam Framework for Solution of Atmospheric Advection Diffusion Reaction Equation. In *Proceedings of the* 7^{th} *EUROSIM Congress on Modelling and Simulation*, pages 6, Prague, Czech Republic, 2010.

V. Simek, R. Dvorak, F. Zboril, V. Drabek. Performance Evaluation of OpenCL Framework for Numerical Solver of Advection Diffusion Equation. In *Proceedings of CSE 2010 International Scientific Conference on Computer Science and Engineering*, pages 279–286, Kosice, Slovakia, 2010.

R. Dvorak, J. Krajicek, J. Zendulka, F. Zboril. The Design of Self-Developmental Model of Atmospheric Pollutant Dispersion. In *Proceedings of the* 7^{th} *EUROSIM Congress on Modelling and Simulation*, pages 6, Prague, Czech Republic, 2010.

V. Simek, R. Dvorak, F. Zboril, J. Kunovsky. Towards accelerated computation of atmospheric equations using CUDA. In *Proceedings of Eleventh International Conference on Computer Modelling and Simulation*, pages 449–454, Cambridge, United Kingdom, 2009. CS IEEE Press.

V. Simek, R. Dvorak, F. Zboril, V. Drabek. GPU accelerated solver of time-dependent air pollutant transport equations. In *Proceedings of* 12^{th} *EUROMICRO Conference on Digital System Design*, pages 707–713, Patras, Greece, 2009. CS IEEE Press.

R. Dvorak, F. Zboril, V. Simek. A Numerical Solution of the Dispersion Modeling in the Planetary Boundary Layer. In *Proceedings of 2009 International Conference on Computational Intelligence, Modelling and Simulation*, pages 6–10, Brno, Czech Republic, 2009. CS IEEE Press.

R. Dvorak, F. Zboril. Simulation of Atmospheric Pollution Dispersion. In *Proceedings MATHMOD 09 Vienna - Full Papers CD Volume*, pages 2671–2674, Vienna, Austria, 2009.

R. Dvorak, F. Zboril, P. Srnec. Adding a Rough Terrain to Atmospheric Pollutant Dispersion Model. In *Proceedings of the Tenth International Conference of Informatics*, pages 221–226, Kosice, Slovakia, 2009.

R. Dvorak, F. Zboril. Modeling of Atmospheric Dispersion from Point Source. In *Proceedings of the* 2^{nd} *UKSim European Symposium on Computer Modelling and Simulation*, pages 46–51, Cambridge, United Kingdom, 2008. CS IEEE Press.

R. Dvorak, F. Zboril. Solving of Advection-diffusion Equation Using Method of Lines. In *Proceedings* 8^{th} *International Scientific Conference on Computers Science and Engineering*, pages 305–311, Kosice, Slovakia, 2008.

M. Kapoun, R. Dvorak, F. Zboril, I. Masek. Modeling of Dispersion of Windborne Material in Atmosphere. In *Chemicke listy*, volume 102, pages 386–391, Brno, Czech Republic, 2008.