Application of local defect correction to a passive tracer in a turbulent channel flow

Application of Local Defect Correction to a passive tracer in a turbulent channel flow

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op donderdag 15 maart 2007 om 16.00 uur

door

Joost de Hoogh

geboren te Breda

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. J.J.H. Brouwers
en
prof.dr. R.M.M. Mattheij

Copromotor:
dr. J.G.M. Kuerten

Cover design by Bregje Schoffelen, Oranje Vormgevers

Printed by the Eindhoven University Press.

This project was funded by the Netherlands Organization for Scientific Research, under Computational Science research grant 635.000.002

A catalogue record is available from the Library Eindhoven University of Technology

ISBN: 978-90-386-0875-4

Contents

Chapter 1

A statistical approach to turbulence

1.1 Introduction

When people ask me what my research is about, I tell them that my research is about turbulence. The reaction that I normally get is that I must be working on airplanes, because that is the general thought that people have about turbulence. Turbulence is more than the shaking of an airplane and to be exact, the shaking of the plane is an effect of actual turbulence on the plane and not turbulence itself. I explain that turbulence is a flow phenomenon and is present almost everywhere. The lift force on an aircraft wing can be determined by the integral of the forces (pressure and viscous) over the circumference of the airfoil. When the influence of viscosity is ignored, the pressure at the airfoil can be computed with the aid of Bernoulli's law. Bernoulli states that $1/2\rho u^2 + p$ is constant along a streamline [40], where u, ρ and p are the velocity, density and pressure, respectively. In steady flow, the velocity around the airfoil is different for the top and bottom half and this results in a pressure difference. If the pressure is integrated along the airfoil, a force is found that can be decomposed into a lift and a drag force. Under the influence of turbulence, the velocity and pressure fluctuate. These fluctuations influence the lift and drag force and make the aircraft shake.

Turbulence can be found in many other areas. Fish have a very rough skin that reduces drag resistance and helps them swim more efficiently, meaning either faster or using less energy. A golf ball has dimples for a similar reason: to make it fly further. In stirring cream into a cup of coffee, a highly turbulent mixing process is in progress. In the beginning, turbulence is visible through the difference in color between the cream and the coffee. Without the presence of turbulence, the coffee would be cold by the time it was mixed. In chemical reactions between different components, turbulence plays an important role as well as the mixing process is accelerated by turbulence. When turbulent mixing and turbulent flow are combined, combustion can be better understood. Combustion is a complex combination of the mixing of fuel and air,

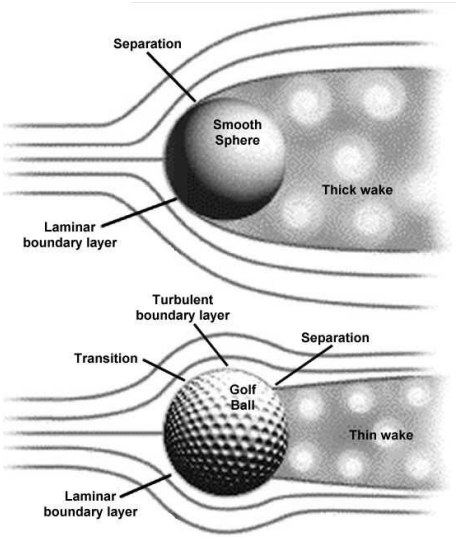


Figure 1.1. A (golf) ball that is released in a flow. The properties of the ball (with or without dimples) cause the flow to become turbulent or remain laminar. In a turbulent flow, the drag force on the ball is reduced and therefore it can fly further [53].

chemistry and a turbulent flow. Due to large density and temperature differences, velocity fluctuations are generated near the flame front. From these examples it is possible to conclude that turbulence is a flow phenomenon and is present in a broad range of fluids. Exceptions are fluids where the viscosity is very large, such as syrup.

A proper definition of turbulence is difficult to give. When the flow is smooth, no turbulence is present and the flow is called laminar. When the flow is not laminar it is either turbulent or is in the process of becoming turbulent, which is called a transitional flow. The transition from laminar to turbulent flow is a gradual process and occurs with increasing velocity. At some point, small perturbations on the flow field are not damped by viscosity and are allowed to grow. At this stage the velocity field is starting to become turbulent. This transition can be observed very well in smoke that is slowly rising from a cigarette, where temperature and density differences cause the flow to become turbulent and this is visualized by the smoke. At first the flow is laminar and the smoke is traveling upwards smoothly. After some time, the smoothness is gone and at that point the flow has become turbulent. Going back one step to the golf ball, the drag force on the ball is depending on the state of the flow around the ball. If the flow is turbulent, the drag force is less than when the flow is laminar. This drag reduction is due to the fact that a turbulent boundary layer around the ball is thinner and has a separation point that is further downstream [53], [40]. This is visualized in figure 1.1, where a ball with dimples is put into a flow and the flow around the ball is schematically visualized. For comparison, the laminar flow around a ball without any dimples is shown as well.

1.2 History of turbulence

People have always been fascinated by turbulence. The first person known to have made some scientific remarks about turbulence was Leonardo da Vinci [17],[60]. He examined and wondered what was happening with the water running out of a pipe falling into a pond. Although he could not understand what was really happening, he made a series of drawings about the ongoing process, see figure 1.2, with the following comment: *"... The small eddies are almost numberless, and large things are rotated only by large eddies and not by small ones, and small things are turned by both small eddies and large..."*.

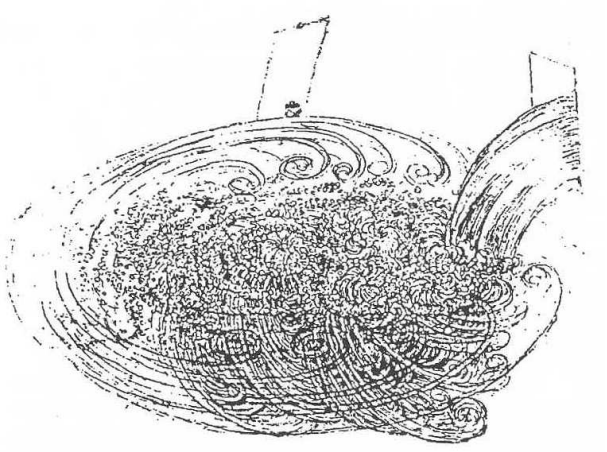


Figure 1.2. Early drawings of turbulence by Da Vinci, showing a pair of counter rotating vortices in the middle of a random wake [17].

Navier-Stokes

Long after the observation of Da Vinci, Newton proposed the equation of motion for solid bodies (1687), which was later extended by Euler and Bernoulli towards frictionless fluids. It took until the 19th century before a generally applicable set of equations was derived that describe the complete motion of a fluid, including friction. This set of equations was derived independently by Charles Navier (1827) and George Stokes (1845), in the same time period, and is called the Navier-Stokes equation. The total set of equations is given by,

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{u} + \vec{f}, \tag{1.1}$$

where $\vec{u}$ is the velocity vector, p the pressure and $\vec{f}$ an external force vector. One can think of gravity ($\vec{f} = (0, 0, g)$), with g the gravitational acceleration, or other body

forces that act on the flow. Furthermore, the equation contains properties of the involved medium, such as the density ρ and the kinematic viscosity ν. Together with the Navier-Stokes equation, the continuity equation is used that describes the balance of mass in a system. For an incompressible flow, the continuity equation becomes,

$$\nabla \cdot \vec{u} = 0. \tag{1.2}$$

The continuity equation and the Navier-Stokes equation together form a set of non-linear partial differential equations that describes the velocity and the pressure of a fluid as a function of space and time. This set of equations can be solved when proper initial and boundary conditions are provided. However, the solution is extremely sensitive towards small fluctuations in the initial state and these lead towards completely different solutions over time. For laminar flow exact solutions to the Navier-Stokes equation, such as a Poiseuille flow, are available. For a turbulent flow, an analytical solution of the complete set of Navier-Stokes equation is not known. The only way to solve the complete set of equations is to simplify them by a dimensional analysis or solve them numerically. The first option allows the influence of the different components to be evaluated with respect to the other terms. When the influence of a certain term is small compared to other terms, it can safely be neglected.

1.3 Dimensional analysis

The first step towards a dimensional analysis is the introduction of a proper scaling for the different quantities of the Navier-Stokes equation. Each of the quantities is divided by a quantity with the same dimension and typical scale. This leads to the following set of dimensionless quantities,

$$\begin{aligned} \vec{u} &= \vec{u'} u^* \\ \vec{x} &= \vec{x'} L \\ t &= t' \frac{L}{u^*} \\ \vec{f} &= \vec{f'} \frac{u^{*2}}{L} \\ p &= p' \rho u^{*2}, \end{aligned} \tag{1.3}$$

where u^* is a reference velocity and L a typical length scale that can be the length or height of a channel or the diameter of a pipe. Inserting these dimensionless quantities into equation (1.1) and dropping the $'$, results in,

$$\frac{u^{*2}}{L}\frac{\partial \vec{u}}{\partial t} + \frac{u^{*2}}{L}\vec{u} \cdot \nabla \vec{u} = -\frac{u^{*2}}{L}\nabla p + \frac{\nu u^*}{L^2}\nabla^2 \vec{u} + \frac{u^{*2}}{L}\vec{f}. \tag{1.4}$$

After some rewriting the dimensionless form of the Navier-Stokes equation becomes,

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla p + \frac{1}{Re}\nabla^2 \vec{u} + \vec{f}, \tag{1.5}$$

with Re the Reynolds number that represents the ratio between the nonlinear advection term and the viscous forces and is defined as,

$$Re = \frac{u^* L}{\nu}. \tag{1.6}$$

The Reynolds number determines the character of the flow. Two limiting cases can be identified: the case where the Reynolds number $Re \gg 1$ and where $Re \ll 1$. When the Reynolds number is much smaller than unity, the viscous forces are dominant and the flow is reduced to a so called Stokes flow, as can for example be found in the motion of micro-organisms.

In normal fluid flow, the Reynolds number is usually larger than unity and the nonlinear advection term dominates the viscous dissipation term. With increasing Reynolds number, the flow changes from laminar to turbulent. After the flow has become turbulent, the Reynolds number is a measure for the turbulence intensity. This property tells something about the amount and size of the different structures that are present in the flow. When the turbulence intensity is large, many small-scaled structures are present.

1.4 Different scales

In images of turbulent flows, such as Da Vinci's early drawings, a large range of length scales can be distinguished. It seems as if turbulence is a chaotic and random process, but still a mean flow component is clearly present. This chaotic or random nature of turbulence allows it to be described as a combination of a mean flow to which fluctuations of different magnitude are added. This can be represented by the following expression for the velocity and pressure that is known as the Reynolds decomposition:

$$\begin{aligned} \vec{u} &= \bar{\vec{u}} + \vec{u}' \\ p &= \bar{p} + p' \end{aligned} \tag{1.7}$$

where $\bar{\vec{u}}$ and $\bar{p}$ are the mean velocity and mean pressure and u' and p' the fluctuating components. These fluctuations have the structure of a set of vortices of different size and shape and are also known as eddies. Large eddies can have the size of the geometry and determine the transport of fluid, particles and tracers. For these large structures in a turbulent flow, the Reynolds-similarity holds [60]. The similarity states that the large scales are independent of the medium in which they occur. When $Re \to \infty$, viscosity is not important and the turbulence is dominated by the large scale structures. However, the large structures are also unstable and continuously break up into smaller and smaller structures. That is the reason why a large range of length scales can be distinguished in snapshots of turbulence. The small scales in a turbulent flow govern the mixing properties of turbulence. The mixing properties of turbulence can be seen as the result of some kind of diffusion process and can be made clearer when going back to the stirring into the coffee. Both molecular diffusion and turbulence ensure that in the end a well mixed solution is present. An important difference between molecular and turbulent diffusion is that the latter is orders of magnitude faster.

Energy cascade

In 1922, Richardson proposed a theory that the turbulent eddies continue to break up until the point where viscosity becomes dominant. At this point, energy is dissipated. The transport of energy from the large scales towards the small scales is called the energy cascade. Without a continuous input of energy, a turbulent flow will dissipate its own energy and be extinguished. A mean pressure difference over a channel is an example of a driving force that provides the large scales with sufficient energy. The total amount of energy that is present within the flow is determined by the kinetic energy (k) of the fluid. When the turbulent structures are scaled using a specific velocity U, the kinetic energy of a turbulent flow scales as, $k \propto U^2$. This energy is transported by the cascade process and the energy flux or dissipation rate is given by the time derivative of the kinetic energy,

$$\frac{\partial k}{\partial t} = -\epsilon, \tag{1.8}$$

where ϵ is the energy dissipation rate. The specific time in which a turbulent eddy looses its energy, scales as L/U. The energy dissipation rate can therefore be quantified by substituting both approximations for the kinetic energy and time scale into equation (1.8), which leads to,

$$\epsilon \propto \frac{U^3}{L}. \tag{1.9}$$

This kind of scaling of the energy flux through the cascade process was first derived by Kolmogorov [31], [32]. During the energy cascade, the properties of the turbulent flow change from convection towards diffusion dominated. In 1941 Kolmogorov made the assumption that, when the whirls become small enough, anisotropy is lost and the turbulence locally becomes homogeneous and isotropic. This means that the turbulence has no preferential direction in a statistical sense. The size of the smallest length scales that are present within the turbulent flow field can be described using the kinematic viscosity (ν) and the average energy dissipation rate ($\langle\epsilon\rangle$). Using both parameters, the smallest length scale that is present within a turbulent flow, is defined as

$$\eta = \left(\frac{\nu^3}{\langle\epsilon\rangle}\right)^{\frac{1}{4}}. \tag{1.10}$$

This length scale is called the Kolmogorov length scale. The description of the small scales that follow from homogenous and isotropic turbulence is not of interest when practical problems are considered. Practical problems are mainly dominated by the large structures because they form the main contribution to transport of heat and mass. Almost all research into the subject of turbulence is focused on finding better models to describe these large scales.

1.5 Current turbulence investigation

1.5.1 Particle trajectories

To improve the models that are used in CFD packages and to improve our global understanding of turbulence, much research has to be done. The parameters of the different (statistical) models can only be computed by investigating large quantities of trajectories of particles that are released in a turbulent flow. The ensemble average of these particle trajectories is used to compute the probability density functions and allows the different scaling parameters to be calculated. However, it is rather time-consuming to obtain the individual particle trajectories. There are two possible approaches to obtain particle trajectories in a turbulent flow. The first approach is physical experiment and the second numerical simulation.

1.5.2 Physical experiments

In the past, many techniques have been developed to measure (turbulent) velocity fields. One can think of Laser Doppler Anemometry (LDA), Particle Image Velocimetry (PIV), Particle Tracking Velocimetry (PTV) and Hot Wire Anemometry (HWA). Both LDA and HWA are limited to one or a few points in space and are not suitable for measuring particle trajectories in an unsteady flow. PIV uses a different approach with a laser sheet positioned in the flow to which tracer particles are added. The reflected light is captured by a camera and a series of snapshots is made. These snapshots are used to determine the two-dimensional flow field. The disadvantage of PIV is that it uses correlation functions to determine the flow field from the information of clouds of particles and does not give any information about individual particle tracks. PTV is able to determine individual particle trajectories but is limited to a small number of particles and, like PIV, in only two dimensions. The latter is disadvantageous because turbulence is a three-dimensional phenomenon. Suzuki [71], and later Walpot [80] have shown that it is possible to obtain accurate Lagrangian measurement data in turbulent pipe flow when using a three-dimensional PTV method to track fluid particles. Using recorded images from three cameras the position of a particle can be determined as a function of time. The velocity of each individual particle can be obtained by taking the time derivative of that trajectory.

1.5.3 Numerical simulations

Apart from physical experiments, the development in computer technology allows for a second option to obtain individual particle trajectories. Numerical simulations have the advantage that the whole velocity field is known, in discrete intervals, with as much detail as the computational grid allows. To be able to perform this kind of detailed simulations, a technique called Direct Numerical Simulation (DNS) is used. This approach does not use any models or assumptions, but solves the complete Navier-Stokes equations. With proper initial and boundary conditions and a computational grid that is fine enough, the development of a turbulent flow can be accurately computed. Hence, Direct Numerical Simulation seems an ideal solution

strategy to perform research on turbulence. However, in each grid point, the complete set of equations needs to be solved. For DNS to be accurate, the grid size needs to be smaller than the smallest length scale that is present within the flow field. This requirement causes DNS to be a computationally very expensive method.

Required grid points

It is possible to give an estimate of the number of grid points that are actually needed to compute all details in a turbulent flow. This estimate is based on the Kolmogorov length scale for turbulence. By combining equation (1.9) and (1.10), the Kolmogorov length scale can be expressed as a function of the Reynolds number, which results in,

$$\frac{\eta}{L} \sim Re^{-\frac{3}{4}}, \tag{1.11}$$

with the Reynolds number, defined as $Re = UL/\nu$. From the requirement that the grid size must be smaller or equal to the smallest length scale that is present in the flow field, the total number of grid points becomes proportional to,

$$N \sim \left(\frac{L}{\eta}\right)^3 \sim Re^{\frac{9}{4}}. \tag{1.12}$$

With the currently available computational resources it is possible to push the Reynolds number towards the range of 1×10^4. For real applications this is still too low as Reynolds number can go up to 1×10^8 and higher [53]. In these situations, turbulence models, based on empirical information, are used that estimate the effect of the smallest scales on the large scales. These turbulence models give the opportunity for more complex geometries or higher Reynolds numbers to be treated numerically. Numerical techniques that are used are called Large Eddy Simulations (LES) or Reynolds Averaged Navier-Stokes (RANS) [60]. In LES, the original Navier-Stokes equations are filtered in space and the effect of all scales that are smaller than the filter width on the larger scales is modeled. The influence of the filtered small scales on the larger scales is represented by an additional stress term, called the sub-grid stress. This sub-grid stress needs to be computed from the large-scale quantities and a well-known model is the Smagorinsky model. RANS goes even further and takes the ensemble average of the Navier-Stokes equation and this removes most of the small details. Due to the modeling of the effect of the small scales, approximations to the actual solution are made when LES or RANS is applied. Therefore, if one is interested in all the details of the solution, the only possible approach is to preform DNS.

1.6 Diffusion theory

1.6.1 Statistical analysis

During his investigations into diffusion, dispersion and turbulence, Taylor developed the idea of isotropic and homogeneous turbulence [72]. This mathematical idealization of turbulence can be used to derive a statistical description of the position of a marked

fluid particle. More information about statistical fluid mechanics is found in books by Hinze [22], Monin and Yaglom [55] or Tennekes [73]. Kolmogorov later used the idea of Taylor to develop the scaling laws for turbulent structures. Taylor found that, within the theoretical description of turbulence, the time correlation of a fluctuating part of the velocity of a marked fluid particle is related to diffusion of matter.

Under the assumption that the correlation time of the velocity of a marked fluid particle is small, its motion can be described using a continuous Markov process [64]. A Markov process has the property that future actions only depend on the present situation and that they do not depend on how this situation was reached. The stochastic model that Taylor developed for the position of a marked fluid particle ($\vec{x}$) is given by,

$$\dot{x}_\nu = A_{\nu\mu} x_\mu + B_{\nu\mu} w_\mu(t), \tag{1.13}$$

where $A_{\nu\mu}$ is called the drift tensor, $B_{\nu\mu}$ the diffusion tensor and $w_\mu(t)$ is a Gaussian white noise process. Because of the earlier usage of the Markov model, this type of equation is also called a Langevin model. In 1908, Langevin used a Markov process to describe the Brownian motion of molecules. The Langevin model can be written in terms of a Fokker-Planck equation that describes the probability function of a marked fluid particle, released from a reference position, see for example an article written by Pope [64]. Van Kampen [28] derived the Fokker-Planck equation in a different way, using an admixture term $\Psi(\vec{x}(t))$. The ensemble average of the admixture term is equivalent to the probability function when the Markov assumption is applied to the displacement of the fluid particle, $p(\vec{x}, t) = \langle \Psi(\vec{x}(t)) \rangle$. The coefficients $A_{\nu\mu}$ and $B_{\nu\mu}$ in this Langevin model need to be obtained from accurate data (either DNS or measurements) and depend on the flow.

It is not an easy task to develop a stochastic model for turbulence and the question is raised why spend so much effort to obtain these Langevin models when the accurate numerical and measurement data is already available? The advantage of using Langevin models is that statistical properties of individual particle tracks can be obtained without computing the flow field. This means that particle tracks can be obtained much faster than with DNS or measurements. This is useful when the same type of flow problem needs to be repeated, but with different boundary or initial conditions. In that case, the Langevin model can be used directly, whereas the DNS and/or measurements need to be repeated to account for the new situation, saving much time. Moreover, it is generally accepted that the coefficients of the Langevin equation do not depend on the Reynolds number for large Reynolds numbers.

Validity of the diffusion theory

In the past, many people have asked under which conditions the Markov approximation is valid. Instead of assuming the equivalence between the Fokker-Planck equation and the stochastic description, Brouwers [10] derived the Fokker-Planck equation using asymptotic analysis in order to check the validity of the diffusion theory. During the derivation of the diffusion theory, the assumption is made that the dimensionless correlation time ($\tilde{\tau_c}$) is much smaller than unity. When this condition ($\tilde{\tau_c} \ll 1$) is met, higher-order terms can be neglected during the derivation of the probability

equation. Brouwers [10] has shown that for all turbulent flows occurring in practice, the dimensionless correlation time is of the order one, $\tilde{\tau_c} = O(1)$ and does not tend to zero if the Reynolds number goes to infinity. Since $\tilde{\tau_c} = O(1)$ and not much larger, the scalar gradient hypothesis, used in many turbulence models in computational fluid dynamics, is an estimate that gives the correct order of magnitude.

Considering the validity of the diffusion theory, the models derived from it are useful in practical, engineering applications [79], where one is mainly interested in the large scale behavior. A proper order of magnitude estimate of the solution to a specific problem is, in these situations, much more valuable than an unknown or very costly scientific solution. Therefore, the order of magnitude estimates, that the current CFD packages can provide, are very useful.

1.6.2 A consistent stochastic model

A consistent stochastic model for turbulence still needs to be found. A stochastic model based on the velocity of a marked fluid particle gives much better results than one based on the displacement. The correlation time of the acceleration of a marked fluid particle is lower than the correlation time for the velocity and tends to zero if $Re \rightarrow \infty$. The consistent stochastic description is based on the following relation between the position and velocity of a marked fluid particle,

$$\begin{cases} \dot{x}_\nu = v_\nu^0 + v_\nu^1 \\ \dot{v}_\nu^1 = A_{\nu\mu} v_\mu^1 + \sqrt{C_0 \epsilon} w_\nu(t), \end{cases}$$

with C_0 the Kolmogorov constant. The corresponding Fokker-Planck equation for the probability density function $p(\vec{x}, \vec{v}, t)$ is given by,

$$\frac{\partial p}{\partial t} = -\frac{\partial p}{\partial x_\nu}(v_\nu p) - \frac{\partial}{\partial v_\nu}(A_{\nu\mu} v_\mu p) + \frac{1}{2} C_0 \epsilon \frac{\partial^2 p}{\partial v_\nu^2}. \tag{1.14}$$

More information on this subject is found for example in the work done by Sawford [66], [67] and Pope [64].

For inhomogeneous turbulence, the damping coefficients $A_{\nu\mu}$ cannot be obtained from Eulerian information. Veenman [77] showed, using numerical simulations, that they can be found from Lagrangian velocity correlation functions, for example a DNS of a turbulent pipe flow. More recently, Walpot [80]-[81] has performed experiments on the same type of flow using 3D-PTV in order to obtain particle tracks to validate the results by Veenman.

An alternative approach to determine the damping coefficients $A_{\nu\mu}$ would be to use the Fokker-Planck equation (1.14). This could be done by determining the probability density function p from results of a DNS of turbulent flow coupled to a convection-diffusion equation for a passive scalar. However, in order to make the connection to passive particles, the effects of molecular diffusion in this convection-diffusion equation should be kept as small as possible [65]. This means that the Schmidt number is high, and this leads to structures in the concentration field that are much smaller than the smallest structures in the velocity field. In order to resolve the smallest length scales, a very fine computational grid is needed to compute the concentration. This topic is discussed in more detail in chapter 3.

1.6.3 Goal of this thesis

As a first step towards the long-term goal of finding the damping coefficients $A_{\nu\mu}$ from the Fokker-Planck equation, in this thesis a numerical method will be developed for the simulation of a passive scalar that is coupled to a turbulent flow at high Schmidt numbers. Apart from that goal, the study of the behavior of a passive scalar in a turbulent flow at high Schmidt numbers is an interesting research topic in its own right, since most practical examples of dispersion in turbulent flow in nature are characterized by high Schmidt numbers. However, for high Schmidt numbers the required number of points is too large to use a uniform grid on the total computational domain. Therefore, a local grid refinement technique has to be applied. In this thesis it will be investigated whether a technique called Local Defect Correction (LDC) can be used to this purpose. With LDC it is possible to focus the available computational power to (small) areas where the admixture is present and be able to efficiently obtain an accurate solution. In the past, LDC has mainly been applied to two-dimensional steady problems by Hackbush [20] and Anthonissen [1]-[3]. Kramer [33]-[34] and Minero [48]-[51] recently applied LDC to a two-dimensional time-dependent problem, but it remains to be seen if LDC is suitable to be applied in a complex three-dimensional and time-dependent problem.

1.7 Outline of this thesis

In order to compute the motion of individual particles or a passive tracer in a turbulent flow, the channel flow itself must be computed. The numerical methods that are used to obtain a fully developed turbulent velocity field are explained in chapter 2. The governing equations are discretized by a pseudo-spectral spatial integration method. In the stream- and spanwise direction, which are assumed periodic, a Fourier-Galerkin spectral method is applied, whereas in wall-normal direction a Chebyshev expansion method is used. Time integration is performed using a combination of an implicit Crank-Nicolson method for the viscous and pressure terms and the Adams-Bashforth method for the remaining terms and follows an approach that was introduced by Kleiser and Schumann [30]. In the last section of chapter 2, the results of the velocity solver are validated with results obtained from literature.

If the turbulent flow field is known, it is possible to compute the development of the passive tracer and of individual particles, which is discussed in chapter 3. First, the numerical approach that is used to track individual particles is explained. Tracking individual particles is not very efficient if the number of particles is large. Together with the fact that an admixture term can be used to compute the coefficients in the Langevin equation, the convection-diffusion equation for the concentration of a passive scalar is introduced. When this equation is made dimensionless, the Schmidt number appears. If the Schmidt number is larger than one, structures appear in the concentration field, which are smaller than those present within the velocity field. To resolve all details that can be found within the concentration field, a very fine computational grid is needed. The convection-diffusion equation is discretized using a finite volume method and the resulting flux terms can be approximated by several

methods. The finite volume method is applied instead of a spectral approach, because a Chebyshev distribution of grid points is not suited to compute the development of the tracer and, in view of later application of LDC, the method has to be capable of treating non-periodic boundary conditions. Time integration of the convection-diffusion equation is performed using the Euler forward method. Due to the usage of different numerical methods, the coordinates of concentration and velocity grid differ and the velocity needs to be interpolated in space and time to obtain the velocity components at the correct locations. Numerical simulations, using different values for the Schmidt number are used to compare the numerical method and to choose the method to be used in the implementation of Local Defect Correction. The results reveal that, without the use of a grid refinement method, it is not possible to increase the Schmidt number to values that are much larger than unity.

One of the most important aspects of a grid refinement technique, when applied to a time dependent problem, is the continuous movement of the refinement area and this is one of the main subjects of chapter 4. The movement consists of three steps, of which the first is the detection of the high activity region, which is based on the work of Bennet and Smooke [7] and later Anthonissen [3]. The other two steps are the actual movement of the refinement area and the interpolation of new fine grid concentration, required when the refinement area is moved into an area where no fine grid information is available. The last part of chapter 4 will discuss the advantages of using multiple nested levels of refinement.

In chapter 5, the implementation of Local Defect Correction to a three-dimensional time-dependent problem is discussed. LDC was first developed by Hackbush [20] and later adapted for a finite volume method by Anthonissen [3]. Anthonissen applied LDC to a two-dimensional, steady problem and recently Minero [51] and Kramer [34] extended this finite volume LDC-approach for time-dependent problems. In this chapter, the step from two towards three dimensions is made. The major difficulty when LDC is applied to a three-dimensional time-dependent problem is the large increase in number of grid points. With a limited possible number of grid points, the requirement that the coarse grid solution must be smooth cannot always be satisfied. A solution to this problem was found that prevents the coarse grid solution from becoming unstable. Simulations are performed on the transport problem, but now with LDC. However, the simulations from chapter 3 could not be repeated due the fact that the required refinement area was too large. In the last section of chapter 5, LDC is compared to Local Uniform Grid Refinement, another grid refinement method that is capable of updating the coarse grid solution using fine grid information.

In chapter 6, the application and behavior of LDC is tested for more general convection-diffusion equations. A different convection velocity, a continuously increasing concentration gradient or refinement area and a positive source term are implemented. Simplified physical problems such as sedimentation, population growth and a flamefront are used to give a physical background to the governing equations. In the last chapter of this thesis, chapter 7, a short overview of the most important conclusions of the individual chapters is given, including several recommendations for the future development of grid refinement methods, applied to a three-dimensional time-dependent problem.

Chapter 2

Flow solver

2.1 Introduction

Before any numerical research on the behavior of passive scalars in a turbulent flow can be performed, the turbulent flow must be computed. This chapter will describe the numerical methods that are needed to compute a fully developed turbulent channel flow. In section 2.2, the Navier-Stokes equation is written in Cartesian coordinates that are most suitable for a channel flow. The set of equations is solved using a spectral method in span- and streamwise direction and a Chebyshev expansion in wall-normal direction. This approach is discussed in section 2.3. To march the solution in time, a second order accurate time integration method is implemented. This method is a combination of an implicit Crank-Nicolson method for the viscous and pressure terms and the Adams-Bashforth method for the remaining terms. This method is based on work done by Kleiser and Schumann [30] and has a pressure correction step that ensures that the velocity field is divergence free, see section 2.4. In the last section (2.5) a comparison is made with results found in literature.

2.2 Cartesian coordinates

In this thesis, a channel flow is used to investigate turbulence. The channel flow uses a Cartesian coordinate system where x, y and z denote the wall-normal, span- and streamwise directions, respectively. To be able to create an infinitely long and wide channel, both span- and streamwise directions are periodic. The dimensions of the computational domain are finite and the length is equal to $L = 4\pi H$ and the width equals $W = 2\pi H$. The height of the channel is equal to $2H$. The walls are located at $x = -H$ and $x = H$. The length scale L used as typical scale is half the channel height H. Figure 2.1 shows a schematic overview of the channel geometry. For convenience, the original Navier-Stokes equation is written in a more convenient form that makes use of the vorticity of the flow. The vorticity is defined as, $\vec{\omega} = \nabla \times \vec{u}$ and combining this definition for the vorticity with the dimensionless Navier-Stokes equation (1.4),

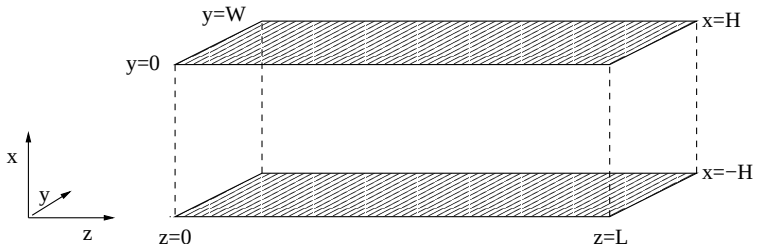


Figure 2.1. Schematic overview of the geometry of the channel.

results in

$$\begin{cases} \frac{\partial u_x}{\partial t} + \omega_y u_z - \omega_z u_y = -\frac{\partial P}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2}\right) \\ \frac{\partial u_y}{\partial t} + \omega_z u_x - \omega_x u_z = -\frac{\partial P}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2}\right) \\ \frac{\partial u_z}{\partial t} + \omega_x u_y - \omega_y u_x = -\frac{\partial P}{\partial z} + \frac{1}{Re}\left(\frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2}\right) + f_z \end{cases} \tag{2.1}$$

In equation (2.1) P denotes the total pressure $P = p + \frac{1}{2}\vec{u}^2$, with p the static pressure. The wall-normal, span- and streamwise components of the velocity vector $\vec{u}$ are denoted with u_x, u_y and u_z, respectively. This notation is used throughout this thesis and the same notation is also used for other vectors or coefficients.

External forcing

Because viscous dissipation constantly removes energy from the system, energy has to be added through the external forcing vector to prevent the flow to come to a halt. Two options are available to apply this external forcing term: the first is to keep a constant pressure gradient and the second is to keep a constant volumetric flow. In this work the second option, a constant volumetric flow, is implemented and this improves convergence to a fully developed turbulent flow [77]. The forcing term can be determined by taking the volume integral of the streamwise component of the Navier-Stokes equation. The requirement of constant volumetric flow is written as

$$\int_V \frac{\partial u_z}{\partial t} dV = 0, \tag{2.2}$$

which gives the following relation for the forcing term,

$$f_z = -\frac{A_z}{VRe}\left(\left.\frac{\partial \bar{u}_z}{\partial x}\right|_{x=1} - \left.\frac{\partial \bar{u}_z}{\partial x}\right|_{x=-1}\right) \tag{2.3}$$

where $\bar{u}_z$ is the streamwise velocity averaged over the stream- and spanwise direction, $A_z = 2WH$ and V denotes the volume of the flow domain.

2.3 Spatial discretization

The above set of differential equations needs to be discretized in space and time. Because of the periodicity in both the stream- and spanwise directions, the most obvious choice for spatial discretization is a spectral method. In the wall-normal direction, no periodic boundaries are present and an expansion based on Chebyshev polynomials is used. Combining both techniques creates a method that computes the velocity in stream- and spanwise direction using a Fourier-Galerkin method and a Chebyshev collocation method in wall-normal direction. This approach has been discussed earlier by Shan *et al.* [70] and was used by Kuerten [38] and Veenman [77] for a pipe flow. In this thesis, the code developed by Veenman is adapted to a channel geometry, since this geometry is better suited to study Local Defect Correction. A further adaptation concerns a more accurate implementation of the continuity equation based on Canuto *et al.* [12].

Spectral method

When a spectral method is used, both the velocity and the pressure are written as a Fourier expansion. Due to the Chebyshev expansion, the amplitude of each Fourier mode depends on the wall-normal coordinate. Each of the velocity components is defined as,

$$u(x,y,z,t) = \sum_{k_y=-M_y/2+1}^{M_y/2-1} \sum_{k_z=-M_z/2+1}^{M_z/2-1} \tilde{u}_{k_y,k_z}(x,t) \cdot e^{\left(\frac{2\pi i k_y y}{W} + \frac{2\pi i k_z z}{L}\right)}, \tag{2.4}$$

where k_y and k_z are the wave numbers in span- and streamwise direction, M_y and M_z are the number of Fourier modes in these directions and $\tilde{u}_{k_y,k_z}(x,t)$ is the amplitude of the Fourier mode at wall-normal position x and time t. The pressure is expressed using a similar representation.

A spectral method has several advantages over other methods. First, the spatial derivatives of the flow quantities are easy to compute and second, if an equal number of points is used, the spectral approach is more accurate than non-spectral methods, provided that the number of Fourier modes is sufficiently large. However, many problems are not suitable for a spectral approach using a Fourier-Galerkin method, because of the impossibility to apply periodic boundary conditions.

Chebyshev expansion

For the wall-normal direction a Chebyshev collocation method is used, see Canuto *et al.* [12] and Fox and Parker [16] for more details. Similar to the Fourier-Galerkin spectral method, derivatives in wall-normal direction can be computed by a multiplication of the amplitudes of the Fourier modes with a derivative matrix. Properties of this derivative matrix, which originates from differentiating Lagrange polynomials [12]. For both the first and second order derivatives, derivative matrices are constructed. The coefficients of these derivative matrices do not change over time and are computed once and stored in memory. In the Chebyshev collocation approach the

collocation points are distributed in a specific way, corresponding to the projection of equally spaced points on a unit circle. The points are within the interval [-1,1] and are given by $x_j = \cos\left(\frac{j\pi}{N}\right)$, where $N+1$ is the total number of points.

The Chebyshev collocation point distribution has the property that, for a given number of points, the approximation error of an arbitrary function is minimal [45]. Moreover, this point distribution ensures that near both walls more points are located than in the center of the channel. The turbulent boundary layer contains small structures and requires a higher resolution than is needed near the center of the channel.

2.4 Temporal integration

Time integration of the Navier-Stokes equations is performed using a second order accurate scheme that is a combination of two methods. The viscous and pressure terms are computed using a second order accurate implicit Crank-Nicolson scheme and the remaining terms are computed using a second order accurate Adams-Bashforth method. The complete equation for the velocity on the next time level reads,

$$\Delta\vec{u}^{n+1} - \frac{2Re}{\Delta t}\vec{u}^{n+1} - Re\nabla P^{n+1} = -\frac{2Re}{\Delta t}\vec{u}^{n} + 3\vec{G}^{n} - \vec{G}^{n-1} + Re\nabla P^{n} - \Delta\vec{u}^{n}, \quad (2.5)$$

with $\vec{G} = Re\left(\vec{\omega}\times\vec{u} - \vec{f}\right)$ the nonlinear term of the Navier-Stokes equation and n is the index for the discrete time interval. To compute the velocity on the next time level, several steps have to be taken and the approach follows the method described by Kleiser and Schumann [30], but than in the collocation formulation instead of the Chebyshev tau-formulation. This method first computes an intermediate velocity field from known terms of equation (2.5). The nonlinear term $(\vec{\omega}^n \times \vec{u}^n)$ is determined in physical rather than in spectral space by transforming back and forth the required components. With this intermediate field, the pressure can be estimated. The continuity equation is satisfied in each collocation point by the application of a pressure-correction step using Green's functions. With this corrected pressure, the divergence-free velocity field at the next time level is obtained.

Efficiency

The correction step does require additional computational effort, but a large part of the parameters are independent of the velocity or the pressure and can be computed during the initialization of the program and stored in memory. Another advantage of spectral methods is that the above equations are decoupled for each Fourier mode. For each mode and velocity component a one-dimensional equation remains that can be solved efficiently, using standard techniques. The decoupling of Fourier modes makes the method very suitable for parallel computing. Each mode can be computed on a different processor, allowing for a large gain in efficiency. If enough memory is available, the efficiency of the solver can be increased further because the LU-decomposition of the matrices, used in Green's functions, can be stored in memory after the initialization phase.

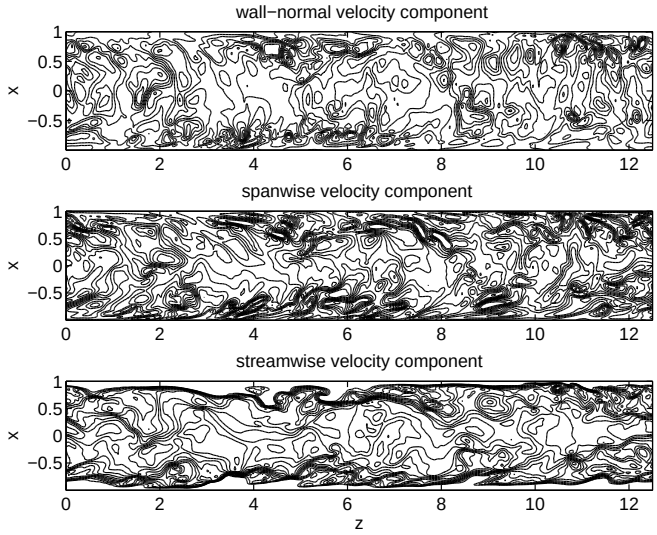


Figure 2.2. Three contour plots in the Z-X-plane of the individual velocity components. The magnitude of the wall-normal and spanwise components is approximately 10% of the streamwise component. The turbulence intensity is approximately 7%.

2.4.1 Initial Condition

The method described above can be used to calculate the time evolution of the turbulent flow field, if an initial velocity field is specified. The creation of a fully developed turbulent flow field is time consuming and is done only once. After the velocity field is fully developed, it is stored and this flow field is used as an initial condition for different realizations of the computation of the particle tracks and the concentration equation. An example of the fully developed turbulent velocity field is given in figure 2.2, where contour plots show a snapshot of each velocity component in the Z-X-plane. Details on how this flow field was obtained can be found in the next section. The magnitudes of the wall-normal and spanwise velocity component are almost equal and approximately 10% of the streamwise component. This leads to a turbulence intensity, which is defined as the kinetic energy of the velocity fluctuations divided by the kinetic energy of the mean flow, of approximately 7%.

2.5 Numerical results

In this thesis, the previously described numerical method serves as a tool to perform research on the development of an admixture in a turbulent flow. Since the admixture is assumed to be passive, the velocity field is not influenced by the admixture term. To demonstrate that the velocity field is correct, results are presented and compared

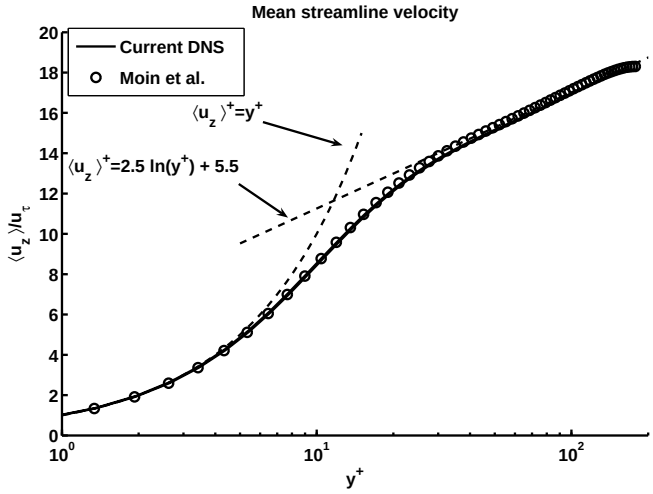


Figure 2.3. The mean streamwise velocity component as a function of y^+. The solid lines are the DNS, the ∘ the measurement data from Moin *et al.* [54].

with data found in literature. The Reynolds number based on the bulk velocity (U_b) is equal to $Re = U_b H/\nu = 2800$. The computational grid contains 129 collocation points in wall normal direction and 128 Fourier modes in both span- and streamwise direction. The time step of the simulation is set to $\Delta t = 2 \times 10^{-5} H/u_\tau$,where u_τ the friction velocity defined as,

$$u_\tau = \sqrt{\nu \left(\left.\frac{du_z}{dx}\right|_{x=-1} - \left.\frac{du_z}{dx}\right|_{x=1} \right)}. \tag{2.6}$$

The Reynolds number based on this friction velocity is equal to $Re_\tau = 180$. At this Reynolds number results can be compared with DNS results by Moin *et al.* [54]. Figure 2.3 shows the mean velocity normalized with the wall shear stress. In total 39 statistical independent velocity fields are used to compute the averaged quantities. The distance to the wall is defined as $y^+ = u_\tau(H - |x|)/\nu$. Good agreement is found between the current results and the data obtained by Moin *et al.* [54]. In figure 2.3 two theoretical lines are drawn for comparison. The first corresponds with the viscous sub-layer, $\langle u_z \rangle^+ = y^+$, and the second is the inertial sub-layer that is given by $\langle u_z \rangle^+ = 2.5 \ln y^+ + 5.5$. The current DNS results are in good agreement with both theoretical laws in the region where they are valid. A second comparison is the velocity fluctuations in figure 2.4, where both simulations also agree well. When other quantities, such as the terms in the balance equation for the turbulent kinetic energy, are compared the same conclusion can be drawn. This means that the velocity field is computed correctly and can be used to compute the development of passive tracer in a turbulent flow.

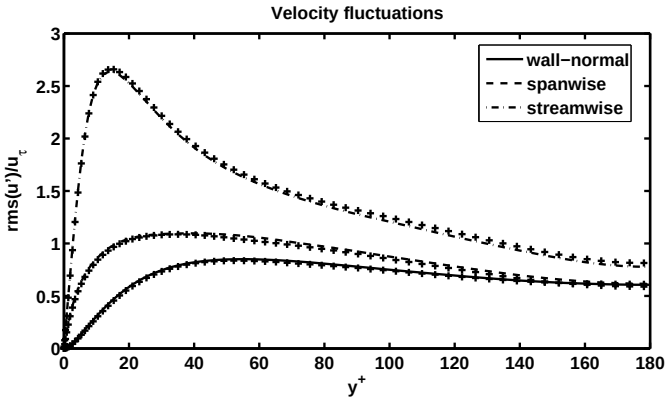


Figure 2.4. Velocity fluctuations as functions of the wall-normal coordinate wall units. Lines: present results, symbols: Moin *et al.* [54]

2.6 Conclusions

With the numerical methods that are described in this chapter, it is possible to compute a fully developed turbulent flow field of a channel. A quick comparison results from similar simulations, that are found in literature, has demonstrated that the flow solver producer physically correct results and can be used to compute the development of a passive tracer in a turbulent channel flow. To this purpose a fully developed flow field is stored and used as an initial condition for each simulation of the passive tracer.

Chapter 3

Passive scalar in a turbulent channel flow

3.1 Introduction

In chapter 2 the method for the computation of a turbulent velocity field was explained. The DNS of the velocity field is used to compute different statistical properties of turbulence. Common methods to study turbulence use statistical quantities such as correlation, structure and probability-density functions. The required data is based on tracks of individual marked fluid particles that can be obtained through numerical simulations, measurements or a combination of both. In chapter 1 the relation between the tracks of individual marked fluid particles and a concentration for an passive scalar was explained. This chapter will focus on how to compute the concentration of a passive scalar in a turbulent flow. Section 3.2 demonstrates the numerical methods that are needed to obtain Lagrangian particle tracks from DNS of turbulent channel flow. In section 3.3 the convection-diffusion equation for a passive tracer is introduced and made dimensionless, which reveals a new dimensionless number: the Schmidt number. The influence of the Schmidt number on the behavior of the concentration in a turbulent flow is discussed in section 3.3.2. The convection-diffusion equation is discretized in section 3.4, using the finite volume method. The flux terms can be obtained using various numerical methods, which are compared later on. Because the fluid velocity and the concentration equation are solved separately, interpolations between the different (fluid and concentration) computational grids are needed in space and time. Time integration of the convection-diffusion equation is performed using the first-order accurate Euler forward method and is considered in section 3.5. In section 3.6, using a Total Variation analysis the, for stability reasons, minimal required grid size is determined. Section 3.7 will present the results of numerical simulations that are performed to compare the different numerical schemes. Both a visual and quantitative analysis is done to choose a method that is best suited to implement Local Defect Correction. In the final section of this chapter, the positions of marked fluid particles are compared to a snapshot of the concentration field.

3.2 Particle trajectories

The importance of particle tracking and a summary of the current investigations was given in chapter 1. Since a turbulent flow solver is available, it is possible to compute an Eulerian velocity field. Using particles that are assumed to be passive tracers, Lagrangian particle tracks can be obtained from the Eulerian flow field. In an Eulerian representation, the fluid properties, such as the velocity and pressure are observed at fixed points in space. The Lagrangian approach is followed when one moves along with a fluid particle and this approach is required to determine of the coefficients in the Langevin model for turbulent dispersion. From the Eulerian velocity field, the Lagrangian particles trajectories can be derived using,

$$\frac{\partial \vec{x}}{\partial t} = \vec{u}(\vec{x}(t), t), \tag{3.1}$$

where $\vec{x}(t)$ is the position of the particle and $\vec{u}(\vec{x}, t)$ the velocity field of the fluid as a function of space and time. To accurately predict the time-derivative of the position of each particle, the fluid velocity has to be known with high accuracy. Because the velocity field is only known at discrete points, the chance is negligible that a particle position exactly matches the location of a grid point. Therefore, the velocity field has to be interpolated in space with high accuracy, to obtain the velocity of the particle.

Different interpolation methods can be used to obtain the required information. Choi *et al.* [13] have shown that higher-order interpolation methods are needed to obtain a result that is accurate enough to be used to compute high-order statistics. Lower-order interpolation methods are only allowed if one is interested in mean particle properties. The method chosen here interpolates the velocity in physical space, is fourth-order accurate and consists of a cubic Hermite interpolation in the wall-normal direction. In the two periodic directions a cubic Lagrangian interpolation is applied. A different approach, see Balachandar *et al.* [4] and Yeung *et al.* [82], is to obtain the particle velocity from the known Fourier coefficients, using direct summation. The advantage of such a method is that no interpolation errors are introduced, but the computational effort is much larger and a direct summation method is only practical for a very limited amount of particles. To give an indication of the required computational time, the interpolation of only 1000 random particles with direct summation takes approximately 35 seconds, whereas the fourth-order interpolation method requires less than one second.

Time integration of the particle position

To predict the position of each particle at the next time level, equation (3.1) has to be integrated in time. A standard Euler forward method is used, that is first-order accurate in time. This means that,

$$\Delta \vec{x}_p = \vec{u}_p \Delta t, \tag{3.2}$$

where Δt is the time step and $\Delta \vec{x}_p$ and $\vec{u}_p$ are the displacement and velocity of that particle, respectively. Due to the small time step of the DNS, the errors that are made

when using an Euler forward method are small. Veenman [77], p29-30 has shown that a higher-order two-stage Runge-Kutta method offers no real improvement over the less complicated Euler forward method.

3.3 Convection-diffusion equation

From the computed particle trajectories, Lagragian statistics can be computed by taking the ensemble average over all particle tracks that are released from the same reference wall-normal position. A restriction to this approach is that the individual tracks have to be statistically independent, meaning that the distance between different particles has to be relatively large. This sincerely limits the total number of particles that can be inserted into the flow. A more efficient method is to use a convection-diffusion equation that describes the particle concentration. In chapter 1 it was already mentioned that the ensemble averaged concentration field is identical to a probability density function of the position of a marked fluid particle. Note that this only holds when the Schmidt number goes to infinity, as the particles are not affected by diffusion. The definition of this dimensionless number is discussed further in section 3.3.2. To be able to use the convection-diffusion equation to describe the particle concentration, the properties of the particles need to be addressed. The particles that are used in the physical experiments and numerical simulations are assumed to be passive tracers and this means that they are only transported by the fluid, but do not influence the flow field. The governing equation for a convection-diffusion equation for the concentration of a passive scalar is given by,

$$\frac{\partial c}{\partial t} + \vec{u} \cdot \nabla c = \mathcal{D}\Delta c + S \tag{3.3}$$

where c is the concentration, S a source/sink term, $\mathcal{D}$ the diffusivity and $\vec{u}$ the fluid velocity. When looking at pure transport of a passive tracer, the source term is equal to zero as no mass is added to the system. Because the tracer is passive, it is only transported by the fluid. From a numerical point of view this has the advantage that the velocity and concentration field can be solved independently, using different numerical methods.

3.3.1 Fields of interest

The convection-diffusion equation can be used to study the statistical properties of a passive scalar. This is not the only reason why a passive tracer is added to a turbulent channel flow. Other fields of interest are the turbulent mixing of different species, turbulent heat transfer and combustion. Brethouwer [9] numerically investigated the influence of turbulence on the reaction rate of two different reacting scalars. Equation (3.3) needs to be solved for each species that participates in the turbulent mixing process. The chemical reaction rate of the process is accounted for through the source term S, that adds or removes the reactants. Better predictions of the time and space needed to mix two or more reactants are very valuable for the optimization of industrial processes. Other simulations performed by Brethouwer consider a point

source of concentration that generates a plume of concentration. These kinds of simulations are useful when predictions of the position of hazardous materials over time are needed, for example, in the case of a plant failure.

When the concentration is replaced with temperature, heat transfer problems can be investigated. Note that buoyancy effects cannot be neglected when large temperature differences are present and the flow cannot be assumed to be incompressible. However, the flow solver discussed in chapter 2 is not designed to handle compressible flows. Good examples are forced cooling of micro-electronic equipment such as specific chip-sets (cpu/gpu) or the evolution of heat-plumes originating from a heated cylinder. Combining both turbulent mixing and heat transfer, combustion processes can be described. With respect to combustion problems, special interest is given to the work done by Anthonissen [1]-[3], who modeled the heat transfer in a steady two-dimensional Bunsen flame, using Local Defect Correction. The properties of LDC are considered in chapter 5.

3.3.2 Dimensionless quantities

To be able to estimate the magnitude of the convective and diffusive terms that are present in the convection-diffusion equation, a dimensional analysis is performed. Following the same approach as in section 1.3, two new dimensionless parameters are introduced,

$$\begin{aligned} c &= c' c^* \\ S &= S' \frac{u^* c^*}{L}, \end{aligned} \tag{3.4}$$

where c^* is a reference concentration. Inserting these dimensionless quantities together with those of (1.3) into equation (3.3) and dropping the $'$ results after some rewriting in,

$$\frac{\partial c}{\partial t} + \vec{u} \cdot \nabla c = \frac{1}{ReSc} \Delta c + S, \tag{3.5}$$

where Re is the Reynolds number and Sc the Schmidt number. The Schmidt number is the ratio between the kinematic viscosity and the molecular diffusivity,

$$Sc = \frac{\nu}{\mathcal{D}}. \tag{3.6}$$

In literature, both dimensionless numbers are combined into the Peclet number, defined as $Pe = ReSc$. If $Pe \gg 1$, the convective term is dominant over the diffusive term. In a turbulent flow, the Reynolds number determines the length scales of the velocity field. Because the tracer acts as a passive scalar, all length scales that are present within the velocity field are directly transferred to the scalar field. The shortest length scales of the scalar field are determined by the Schmidt number. An expression for the smallest length scales that are present in the velocity field was determined by Kolmogorov, see equation (1.10). According to Batchelor [5], the size of the smallest scales that can be found within the concentration field is proportional to,

$$\eta_B = \frac{\eta}{\sqrt{Sc}} \tag{3.7}$$

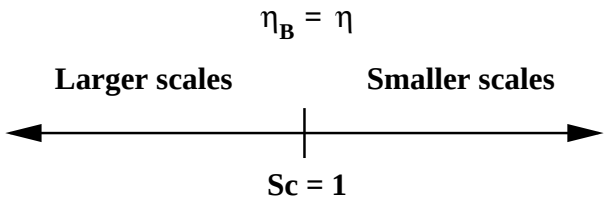


Figure 3.1. Schematic representation of the size of the length scales that are present within the concentration field, with respect to the size of the velocity length scales. At a Schmidt number of one the scales of the velocity field are equal to the scales in the concentration field.

and is referred to as the Batchelor length scale. Figure 3.1 shows a schematic representation of the size of the length scales as a function of the Schmidt number, with respect to the size of the velocity scales For a scalar in a liquid the Schmidt number is usually much larger than unity, whereas for gases $Sc \leq 1$, see Brethouwer [9]. With a Schmidt number that is much larger than one, structures that are present within the scalar field are much smaller than that of the velocity field. If the Schmidt number is smaller than one, diffusion ensures that the structures are larger.

3.3.3 Numerical approach

From a numerical point of view, if all length scales have to be resolved, Schmidt numbers that are larger than unity cause the computational effort to increase rapidly. With the appearance of smaller structures, a finer computational grid is needed to capture every detail of the concentration. Real applications, where $Sc > 1$ and $Re \gg 1$ are not within reach of the available computer power. In these cases, the minimum possible grid size is larger than the smallest length scales and not every detail is captured. Using the scalar gradient hypothesis [28], it is possible to model the influence of the smallest scales on the large scales, similar to what is done with LES or RANS for fluids, see for example an article from Warhaft [79]. However, the scalar gradient hypothesis is based on the diffusion theory for turbulence and it was shown by Brouwers [10] that it is not asymptotically correct when $Re \rightarrow \infty$, see chapter 1. If with a high Schmidt number all small-scale structures have to be resolved, a different, more efficient approach needs to be found.

If one thinks at the initial shape of a blob of dye in a turbulent flow, the dye is only present in a very small part of the (computational) domain. With a local grid refinement technique, it is possible to focus the computational power to the small regions where a high resolution is actually needed: in the region where the concentration is unequal to zero or where large concentration gradients occur. This approach opens a possibility to compute the evolution of passive tracer in a turbulent flow at higher Schmidt numbers. Before these grid refinement techniques can be implemented, one first needs to be able to compute the concentration field. The next section will present details about the numerical methods that are applied to solve the convection-diffusion equation.

3.4 Discretization method

The goal of this thesis is to apply a grid refinement method called Local Defect Correction to a concentration equation for a passive tracer. This tracer is only present in the center of the channel and remains near the center during the computations. The spectral approach, used to compute the velocity field, uses a high resolution near the wall, but not near the center of the channel. The Chebyshev distribution is therefore not suited to be used to compute the development of the passive scalar. Furthermore, with the future application of LDC in mind, periodic boundary conditions on the refinement area are not possible and a spectral method cannot be applied. Instead of a spectral method, a finite volume method is used to compute the concentration field. This method has the advantage, over finite difference and spectral methods, that the integral concentration over the whole computational domain is automatically conserved. A possible exception to this can occur near the edge of the computational domain where the boundary conditions cannot always satisfy the requirement of conservation. Another exception, but of physical origin, is when the source term is unequal to zero.

When the finite volume approach is applied to the convection-diffusion equation this results in,

$$\frac{\partial c}{\partial t}\int_V dV + \int_V (\vec{u}\cdot\nabla c)dV = \int_V \frac{1}{ReSc}\Delta c dV + \int_V S dV, \tag{3.8}$$

with V the volume of a control volume. This equation is very difficult to solve in an exact way because of the volume integrals over the convective and diffusive terms.

The integrals over an arbitrary volume V of a quantity κ can be expressed by an integral over its surface area A, by applying Gauss's theorem. The only restriction to this is that the theorem is only valid when the quantity is differentiable. Gauss's theorem applied to an arbitrary quantity κ, is expressed as,

$$\int_V \mathrm{div}\vec{\kappa} dV = \int_A \vec{\kappa}\cdot\vec{n} dA, \tag{3.9}$$

where $\vec{n}$ is the outward normal on the surface A. The fluxes through the surface are only working in the direction normal to that surface. On a rectangular grid, the flux terms can be computed individually for each surface, splitting the problem into multiple sub-problems that can be treated much more easily.

Applying Gauss's theorem to equation (3.8) leads to,

$$\frac{\partial c}{\partial t}\int_V dV + \int_A \vec{u}c\cdot\vec{n} dA = \frac{1}{ScRe}\int_A \frac{\partial c}{\partial n} dA + \int_V S dV. \tag{3.10}$$

Equation (3.10) states that the concentration in a specific control volume can only change due to a flux (either caused by convection or diffusion) that is entering or leaving through a surface or due to a non-zero source term. Before equation (3.10) can be used to compute the solution, it is necessary to look closer into the computation of the different flux terms.

3.4.1 Computation of the flux terms

In a finite volume approach, the center of the control volume is used to store the value of the different quantities such as the velocity and concentration. However, the different fluxes need to be evaluated at the surfaces of the control volume and not in the center point. In the next section, the position of a quantity on one of the surfaces of a control volume is denoted with the index $\frac{1}{2}$. This notation will be used throughout the rest of this thesis. Before some of the available methods to obtain the flux of concentration are reviewed, equation (3.10) needs to be written in discrete form, suitable for numerical computations. The time derivative will be treated in section 3.5. The concentration in the center of the control volume is denote by $c_{j,k,l}$ and is an abbreviation of $c(x_j, y_k, z_l)$. Furthermore, because of the finite volume method, $c(x_j, y_k, z_l)$ is the volume averaged concentration and is exact, whereas $\int_V S dV = S_{j,k,l}V$ is a numerical approximation. Using this information and the knowledge that the computational grid is rectangular, the surface area of each direction is a constant and the concentration equation becomes,

$$\begin{aligned} V\frac{dc_{j,k,l}}{dt} &= S_{j,k,l}V - A_j\big(u_x c|_{j+\frac{1}{2},k,l} - u_x c|_{j-\frac{1}{2},k,l} \\ &- A_k\big(u_y c|_{j,k+\frac{1}{2},l} - u_y c|_{j,k-\frac{1}{2},l}\big) \\ &- A_l\big(u_z c|_{j,k,l+\frac{1}{2}} - u_z c|_{j,k,l-\frac{1}{2}}\big) \\ &+ \frac{A_j}{ReSc}\Big(\frac{\partial c}{\partial x}\Big|_{j+\frac{1}{2},k,l} - \frac{\partial c}{\partial x}\Big|_{j-\frac{1}{2},k,l}\Big) \\ &+ \frac{A_k}{ReSc}\Big(\frac{\partial c}{\partial y}\Big|_{j,k+\frac{1}{2},l} - \frac{\partial c}{\partial y}\Big|_{j,k-\frac{1}{2},l}\Big) \\ &+ \frac{A_l}{ReSc}\Big(\frac{\partial c}{\partial z}\Big|_{j,k,l+\frac{1}{2}} - \frac{\partial c}{\partial z}\Big|_{j,k,l-\frac{1}{2}}\Big), \end{aligned} \tag{3.11}$$

where,

$$u_x c|_{j+\frac{1}{2},k,l} = (u_{x,j+\frac{1}{2},k,l} c_{j+\frac{1}{2},k,l}). \tag{3.12}$$

The gradient terms from equation (3.11) are the gradients of the concentration evaluated at the corresponding surface. When all terms containing the concentration gradient are grouped into a diffusive flux term and the remainder into a convective flux term, a shorter notation for equation (3.11) is given by,

$$V\frac{dc_{j,k,l}}{dt} + \sum_{i=1}^{3} Q_{c,i} = \sum_{i=1}^{3} Q_{d,i} + S_{j,k,l}V \tag{3.13}$$

where Q_c is the convective flux term and Q_d the diffusive part, which both need to be summed over the three directions to obtain the total flux that is acting on a specific control volume. Treating all three directions individually would be a repetition of identical steps and therefore only one direction (wall-normal) is treated as an example. For readability the indices denoting the span- and streamwise direction are omitted.

The terms for the convective and diffusive fluxes become,

$$Q_{c,j} = A_j\Big(u_x c|_{j+\frac{1}{2}} - u_x c|_{j-\frac{1}{2}}\Big) \tag{3.14}$$

$$Q_{d,j} = \frac{A_j}{ReSc}\Big(\frac{\partial c}{\partial x}\Big|_{j+\frac{1}{2}} - \frac{\partial c}{\partial x}\Big|_{j-\frac{1}{2}}\Big). \tag{3.15}$$

The diffusive term only contains the gradient of the concentration and that can be evaluated using the second-order accurate central difference method.

3.4.2 Generalized flux method

For the convective part of the flux, several numerical methods are available. Following the approach found in work done by Kuerten [35], a standard equation for the flux is introduced. This method allows for a good comparison and a fast adaptation between the different numerical schemes. To this purpose the convective term from equation (3.14) is split further into,

$$Q_{c,j} = A_j\big(Q1|_{j+\frac{1}{2}} - Q1|_{j-\frac{1}{2}} + Q2|_{j-\frac{1}{2}} - Q2|_{j+\frac{1}{2}}\big), \tag{3.16}$$

where $Q1$ and $Q2$ are flux terms depending on the implemented numerical method.

Central difference method

The most straightforward method to obtain the flux of concentration is using a central differencing method. The terms for the general approach become,

$$\begin{aligned} Q1|_{j+\frac{1}{2}} &= \frac{1}{2}(u_{x,j+1}c_{j+1} + u_{x,j}c_j) \\ Q2|_{j+\frac{1}{2}} &= 0. \end{aligned} \tag{3.17}$$

The central difference method contains no numerical diffusion and numerical instabilities are likely to be occur. However, when stable the solution is assumed to give a good representation of the physics.

First-order upwind method

One way to prevent these oscillations is by using an upwind method that adds numerical diffusion. The coefficients [35] for the first-order upwind method are given by,

$$\begin{aligned} Q1|_{j+\frac{1}{2}} &= \frac{1}{2}\big(u_{x,j+1}c_{j+1} + u_{x,j}c_j\big) \\ Q2|_{j+\frac{1}{2}} &= \frac{1}{4}\big(|u_{x,j+1} + u_{x,j}|(c_j - c_{j+1})\big). \end{aligned} \tag{3.18}$$

From literature it is known that the first-order upwind method adds a significant amount of numerical diffusion to the solution.

Third-order accurate upwind method

To obtain a higher accuracy than with the first-order upwind method, but keeping a small amount of numerical diffusion to damp oscillations, a second upwind scheme is implemented, that is third-order accurate in space. The coefficients for this method are given by,

$$\begin{aligned} Q1|_{j+\frac{1}{2}} &= \frac{1}{12}\left(-u_{x,j+2}c_{j+2} + 7u_{x,j+1}c_{j+1} + 7u_{x,j}c_j - u_{x,j-1}c_{j-1}\right) \\ Q2|_{j+\frac{1}{2}} &= \frac{1}{12}\left(-|u_{x,j+2}|c_{j+2} + 3|u_{x,j+1}|c_{j+1} - 3|u_{x,j}|c_j + |u_{x,j-1}|c_{j-1}\right) \end{aligned} \quad (3.19)$$

The disadvantage of this method is that it requires a larger stencil and is computationally more expensive.

Other methods

A commonly used technique is the application of limiter functions [9], [24], which limit the maximum flux of concentration. An example of such a method is the MUSCL method, developed by Van Leer [41]. The advantage of flux limiting methods is that they prevent oscillations, but they have the disadvantage that they are computationally expensive. Many other schemes are available, but the goal of this research is to implement a grid refinement technique and not a comparison of the numerous available numerical methods. With the future application of LDC, it is most practical to use a fairly simple scheme that combines stability, accuracy and efficiency. Higher-order schemes, with or without flux limiters, are more accurate but also computationally expensive. Which of the above methods has the best properties to compute the evolution of a passive tracer in a turbulent flow, in combination with LDC, cannot be predicted on forehand. A comparison between the three methods is found in section 3.7.2.

Boundary conditions

For every method used, boundary conditions have to be implemented to allow a well determined solution. This means that in stream- and spanwise direction, periodic boundary conditions are applied. Near the wall of the channel the concentration gradient is set to zero, because the concentration cannot penetrate the wall. Since the initial concentration field is usually chosen to be in the center of the channel, the boundary values of the concentration are almost equal to zero throughout the computations.

3.4.3 Spatial velocity interpolation

The computational grid on which the velocity field is computed differs from the concentration grid. Therefore the coordinates of the concentration field do not match those of the velocity field and the velocity field needs to be interpolated in space. Interpolation can be performed either by linear interpolation or by the hybrid fourth-order method, see section 3.2. The higher-order spatial velocity interpolation gives a

much smoother velocity on the concentration grid, but is more expensive. Together the fact that all details of the concentration field have to be resolved, the hybrid fourth-order interpolation method is preferred over linear interpolation. Another option would be to obtain the velocity directly from the known Fourier coefficients, but this is computationally very expensive, see section 3.2, and therefore not considered.

3.5 Time integration

3.5.1 Euler forward

The final step, before some numerical simulations can be performed, is the treatment of the time integration. Using a standard Euler forward scheme, the time derivative, in discrete form, is written as,

$$\frac{\partial c}{\partial t} = \frac{c^{t+\Delta t} - c^t}{\Delta t}. \tag{3.20}$$

First-order time integration is allowed because the time step used to compute the velocity field is very small. Since the grid size of the concentration is generally smaller than the grid size of the velocity, the time step for the concentration needs to decrease accordingly. This can easily be demonstrated if one takes into account the convective flux of the concentration. Note that $Pe \gg 1$ and the influence of the diffusive flux is very small. The flux that enters a control volume with a certain velocity should not be able to travel further than the total length of that control volume. This means that the time step needs to be smaller than the time needed for a marked fluid particle to travel a distance equal to the grid size. To give an estimate of the minimum required time step, the velocity in streamwise direction is used. The DNS has shown that the velocity components in both spanwise and wall-normal direction are, at most, on the order of 10% of the streamwise velocity. As the maximum streamwise velocity equals one, the time step must satisfy

$$\Delta t < \Delta z, \tag{3.21}$$

to maintain a stable solution.

3.5.2 Velocity interpolation

After the velocity field is interpolated in space, it is only known on the discrete time levels used in the DNS of the channel flow. To be able to estimate the fluxes that enter or leave the control volumes at the correct time level, the velocity field is interpolated a second time, but now in time.

The time step of the velocity field is small enough to resolve all fluctuations that are present within the velocity field. This also means that the velocity difference between two subsequent time levels is very small. Unlike the spatial interpolation, the temporal interpolation has to be performed at every intermediate time step of the concentration computations. Using large Schmidt numbers, more small steps are needed and this would mean that a significant amount of computational time is spent to interpolate the velocity field in time. Veenman [77] has mentioned that, when

computing particle statistics, the improved accuracy of the 2^{nd} order Runge-Kutta scheme offered no improved on the results. With this remark and the above reasons taken into account, linear interpolation is the most suitable method to interpolate the velocity field in time.

3.6 Total Variation Diminishing property

To check if the grid size is small enough and numerical oscillations that occur are damped instead of amplified, one can look at the total variation that is present in the solution. The total variation in a one-dimensional system is defined as

$$TV(c) = \sum_{j=1}^{n_x} |c_j - c_{j-1}|, \tag{3.22}$$

where n_x is the number of grid points in wall-normal direction. The total variation tells something about the amount of oscillations that are present in the solution. Because the absolute value of the difference between two neighboring grid points is used, any oscillations that appear increase the total variation of the solution. A numerical scheme that has the property that the total variation does not increase over time is called Total Variation Diminishing (TVD) and this can be expressed by the following,

$$TV(c^{n+1}) \leq TV(c^n). \tag{3.23}$$

When the TVD condition is met, smoothness and stability of the solution are assured.

Desired grid size

To check the stability of the current problem, the following part will derive sufficient conditions for the current method to be TVD. For this purpose the central difference method is used, but a similar derivation can be done for other numerical schemes. First, an expression for the concentration on the next time level is given by combining equation (3.13),(3.15) and (3.17) to,

$$c_j^{n+1} = c_j^n - \frac{u_{x,j}^n \Delta t}{2\Delta x}(c_{j+1}^n - c_{j-1}^n) + \frac{\Delta t}{Pe\Delta x^2}(c_{j+1}^n - 2c_j^n + c_{j-1}^n). \tag{3.24}$$

Inserting this into equation (3.22) one obtains a relation between the TV-values at two subsequent time levels,

$$\begin{aligned} \sum_{j=1}^{n_x} \left|c_{j+1}^{n+1} - c_j^{n+1}\right| &= \sum_{j=1}^{n_x} \Big| \big(1 - \frac{2\Delta t}{Pe\Delta x^2}\big)(c_{j+1}^n - c_j^n) \\ &- \big(\frac{u_{x,j}^n \Delta t}{2\Delta x} - \frac{\Delta t}{Pe\Delta x^2}\big)(c_{j+2}^n - c_{j+1}^n) \\ &+ \big(\frac{u_{x,j}^n \Delta t}{2\Delta x} + \frac{\Delta t}{Pe\Delta x^2}\big)(c_j^n - c_{j-1}^n) \Big| \end{aligned} \tag{3.25}$$

It follows that,

$$\begin{aligned}\sum_{j=1}^{n_x}\left|c_{j+1}^{n+1}-c_j^{n+1}\right| &\leq \left|1-\frac{2\Delta t}{Pe\Delta x^2}\right|\sum_{j=1}^{n_x}\left|c_{j+1}^{n}-c_j^{n}\right| \\ &+ \sum_{j=1}^{n_x}\left|\frac{u_{x,j}^n\Delta t}{2\Delta x}-\frac{\Delta t}{Pe\Delta x^2}\right|\left|c_{j+2}^{n}-c_{j+1}^{n}\right| \\ &+ \sum_{j=1}^{n_x}\left|\frac{\Delta t}{Pe\Delta x^2}+\frac{u_{x,j}^n\Delta t}{2\Delta x}\right|\left|c_{j}^{n}-c_{j-1}^{n}\right| \end{aligned} \tag{3.26}$$

where the following rule has been applied,

$$|a+b|\leq|a|+|b|. \tag{3.27}$$

Inequality (3.26) is an expression for the total variation on the next time level, written in quantities that are known on the current time level. If this equation is inserted into the left hand side of equation (3.23), an expression is obtained for the TVD-requirement at the current time level. After some rewriting and using the relation that,

$$\sum_{j=1}^{n_x}\left|c_j^n-c_{j-1}^n\right|=\sum_{j=1}^{n_x}\left|c_{j+2}^n-c_{j+1}^n\right|=\sum_{j=1}^{n_x}\left|c_{j+1}^n-c_j^n\right|, \tag{3.28}$$

the following condition is found,

$$\frac{\Delta t}{Pe\Delta x^2}>\frac{|u_{x,j}^n|\Delta t}{2\Delta x}. \tag{3.29}$$

When this condition is met the numerical scheme is TVD for convection-diffusion equation. With some additional rewriting,

$$\Delta x<\frac{2}{|u_{x,j}^n|Pe}, \tag{3.30}$$

a requirement for the maximum grid size is found, that is a function of the Peclet number. This derivation can be repeated for the first-order upwind method but will not be repeated here. For this method it can be shown that it is stable if $|u_x|\Delta t<\Delta x$, even without molecular diffusion. For the third-order upwind method, this kind of derivation is not possible. However, upwinding introduces numerical diffusion and this stabilizes the solution and a larger grid size can be used than what would be required for the central difference method.

Required grid size

If the central difference method is again taken as an example, an indication of the large number of required grid points can be obtained. In the simulations that are performed, the maximum streamwise velocity is approximately equal to one. The Reynolds number is equal to $Re=2800$, the Schmidt number is set to $Sc=1$ and the

length of the channel is $L = 4\pi$. According to criterion (3.30) this leads to more than 15000 grid points in the streamwise direction. The use of such a large number of grid points in one direction is impossible in a three-dimensional simulation. In practice, a compromise between accuracy and stability has to be found. This can be either a higher order upwind method or the central difference method with slightly relaxing condition (3.30). Numerical simulations that are performed in the next section have to provide an answer which option yields the best results.

3.7 Numerical simulations

3.7.1 Initial conditions

To investigate the influence of the Schmidt number and the three mentioned numerical methods on the evolution of the passive tracer, several numerical simulations are performed. In all computations, the Reynolds number is kept constant ($Re = 2800$) to maintain an identical velocity field. The influence of the different numerical methods can be investigated by keeping a constant Schmidt number and they other way around. A series of experiments has been performed containing three different Schmidt numbers and three different numerical schemes, bringing the total amount of possible combinations to nine. The initial condition for the concentration field is set to be a Gaussian shaped blob of concentration, that is given by,

$$c^0_{j,k,l} = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x-x_0)^2}{\sigma_x^2} - \frac{(y-y_0)^2}{\sigma_y^2} - \frac{(z-z_0)^2}{\sigma_z^2}} \tag{3.31}$$

where σ_x, σ_y and σ_z are the standard deviations of the concentration in wall-normal, span- and streamwise direction respectively. The parameters x_0, y_0 and z_0 are the offset of the blob and are chosen in such a way that the concentration at the boundaries of the computational domain is equal to zero. The values of the standard deviations are set to $\sigma_x = 5 \times 10^{-2}$ and $\sigma_y = \sigma_z = 5 \times 10^{-1}$. The computational grid contains $128 \times 256 \times 1024$ points in wall-normal, span- and streamwise direction and the grid size becomes: $\Delta x = \frac{1}{64}$, $\Delta y = \frac{\pi}{128}$ and $\Delta z = \frac{\pi}{256}$.

Such a large initial condition was required for stability reasons. A smaller initial condition contains larger gradients and needs a higher resolution, which was not possible, due to the higher computational time involved. In view of the future application of LDC, this large initial condition is problematic as well. To be applicable, LDC requires a small refinement area, compared to the computational domain, which is with these initial conditions not satisfied. Furthermore, decreasing the grid size for LDC on the finest level, to match the above values, leads to a computational grid on the global grid that is too coarse to maintain a stable solution. More details about the implementation and the encountered difficulties are discussed in chapter 5. This remark about LDC is made here to anticipate that it will not be possible to compare LDC with the results that are computed in this chapter. The results shown here solely serve the purpose of selecting a numerical scheme that can be used to provide an accurate and stable solution to investigate the applicability of LDC.

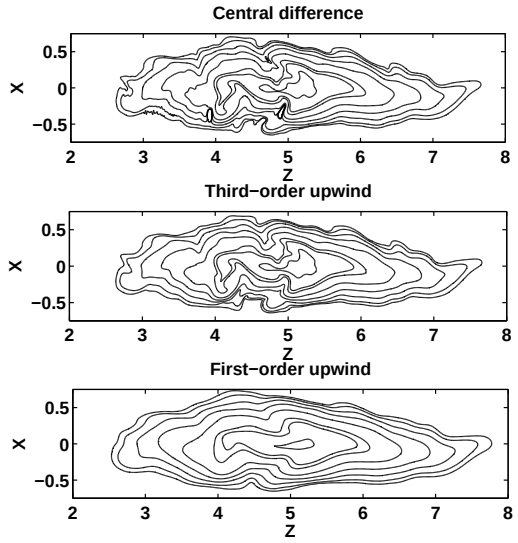


Figure 3.2. Three different contour plots of concentration through the center of the blob with $Sc = 1.0$ at $t = 0.14H/u_\tau$. Top: central differencing; Middle: third-order upwind; Bottom: first-order upwind method.

3.7.2 Results

In figure 3.2 contour lines of the concentration in a cross-section in the Z-X plane are shown for the different numerical schemes. From this figure it can be concluded that the central differencing method shows more details, but also some oscillations, especially upstream of the center of the concentration blob (see figure 3.2, bottom-left corner). Adding numerical diffusion will damp these oscillations, but also removes some of the details. The results of the first-order upwind method clearly reveal that too much numerical diffusion is present, resulting in the removal of all details in the solution. The third-order upwind method provides the best of both worlds: the majority of the details can still be seen, whereas oscillations are absent. To better visualize the difference between the three different numerical methods, figure 3.3 shows the concentration profile in streamwise direction, zoomed in at the center of the blob of concentration.

If the Schmidt is lower than unity, $Sc = 0.1$, the difference between the numerical methods is not as clear anymore. Simulations showed that the results of the three numerical methods are almost identical. This behavior is explained by the fact that physical diffusion is dominant over the numerical diffusion in the two upwind methods.

When the simulation is repeated for a third time, using $Sc = 10$, different results are obtained, see figure 3.4. The solution becomes unstable when the central difference method is used, caused by the lack of numerical diffusion that stabilizes the solution. A solution to prevent the instabilities is to decrease the grid size, but this is in this

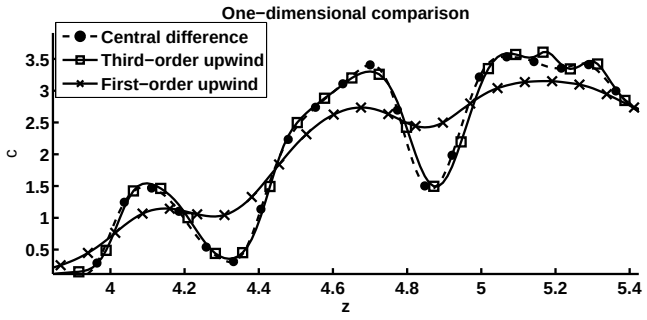


Figure 3.3. The concentration as a function of the streamwise coordinate in the center of the blob for all three methods. All fields are computed for $Sc = 1.0$ at a snapshot is made at $t = 0.14H/u_\tau$.

situation not an option. It should be noted that the oscillations are only present in the left part of the computational domain and the solution near the center of the blob can be used for a visual comparison with both upwind methods. Both upwind methods maintain a stable solution, but the solution provided by the first-order upwind method does not contain any detail and is almost the same as for $Sc = 1$. The third-order upwind method is the only method that provides both a stable and meaningful solution.

3.7.3 Surface area

To be able to quantify the influence of turbulence on the concentration field, the total surface area of a concentration isosurface is calculated. The surface area is computed using a method developed by Geurts [18]. Using the evolution of the surface area, for specific values of the concentration, different properties can be distinguished. Due to numerical or natural diffusion, the surface area increases. If the both numerical and natural diffusion are not dominant, small-scaled structures that appear in the concentration field increase the size of the surface area. Furthermore, the appearance of numerical instabilities causes a very rapid increase in the total surface area. With another approach such as the mean radius of the concentration, these oscillations will be averaged out. Two remarks are that the solution must be stable and, if the contour-value is chosen too large, the surface area will first increase and later decrease towards zero, because the maximum concentration drops below the chosen value. Figure 3.5 shows the evolution of the total surface area in time for three methods at two different values of the concentration and all three Schmidt numbers. Without any surprise, the central difference method has the largest increase in surface area over time. Due to the lack of numerical diffusion, small scaled structures (and oscillations) are not removed and the surface area grows most rapidly. The third-order upwind method shows almost identical behavior, but the influence of the numerical diffusion is clearly present. When comparing the first-order upwind method with the central difference method, it seems that the evolution of the surface area of the upwind

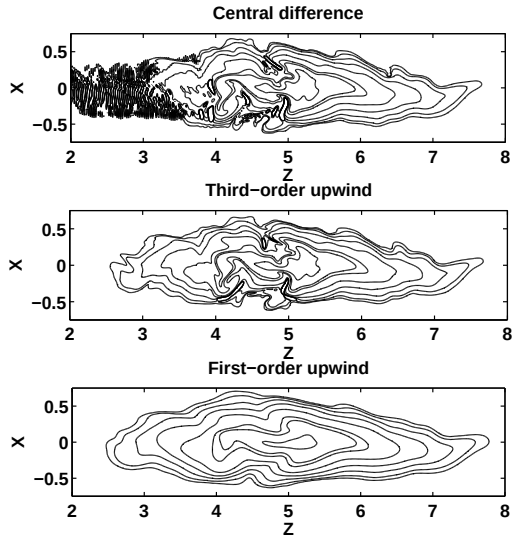


Figure 3.4. Three different contour plots of concentration through the center of the blob with $Sc = 10$ at $t = 0.14H/u_\tau$. Top: central differencing; Middle: third-order upwind; Bottom: the first-order upwind method.

method at a Schmidt number of one is equal to that of the central difference method at $Sc = 0.1$. This results in a statement that increasing the Schmidt number and using a very dissipative numerical method is not very meaningful. When the central difference method is used as reference, numerical diffusion can be seen as equivalent to a decrease of the Schmidt number. For each numerical method, this decrease in Schmidt number is different.

3.7.4 L_2-norm

In order to assign a single value to a concentration field, the L_2-norm of the concentration can be used, which is defined as,

$$\|c\|_2 = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(c_i)^2}, \tag{3.32}$$

with N the total number of grid points. This norm is most useful to quantify the difference between two results of numerical simulations or the difference between a numerical and an exact solution of the same problem, as this gives an indication about the quality of the solution. However, in the simulations that are performed in this thesis, the exact solution is not known and an approximation to it is required. Such an approximation is usually found by increasing the accuracy through a significant

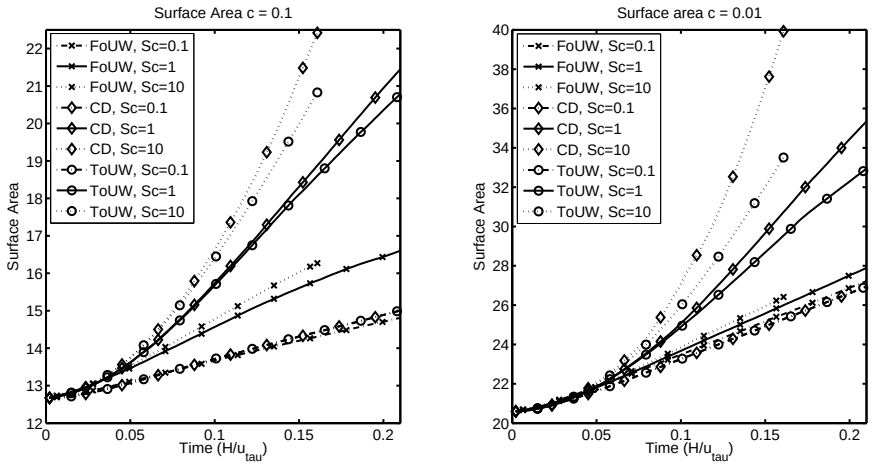


Figure 3.5. The evolution of the surface area at three different Schmidt numbers, using the three computational methods discussed. In the left figure, the surface area is computed at a concentration, $c = 0.1$ and in the right figure $c = 0.01$. In the legend the abbreviations stand for: CD = Central difference, FoUW = first-order upwind, ToUW = third-order upwind.

decrease in grid size, but this decrease is not affordable due to the required computational time. However, in a large part of the computational domain, the concentration is equal to zero. If only half the length of the channel is used to compute the approximate exact solution, the grid size can be decreased by a factor of two, whilst the computational costs do not increase.

The choice for a decrease in grid size in streamwise direction was made because it is assumed that numerical errors are largest by the grid size in this direction, since it has the largest velocity component. The oscillations in the solution at $Sc = 10$ already indicated that this assumption is valid. To be able to follow the blob of concentration, the grid is adjusted at specific moments in time by moving points from the downstream towards the upstream side. At the new streamwise boundaries, the concentration gradient is set to zero. With this simplified grid refinement approach an approximation of the exact solution (c_e) is computed using the central difference method and a Schmidt number of one.

For two different moments in time and all three Schmidt numbers, equation (3.32) is used to compute $\|c - c_e\|_2$ that is normalized with $\|c_e\|_2$. The results, shown in table 3.1, corroborate the results obtained from a visual comparison and the surface area. Note that the relative error for the central difference method at a Schmidt number of one is not small, which implies that the grid size is not fine enough.

	central difference	3^{rd} order upwind	1^{st} order upwind
$t = 250, Sc = 0.1$	5.0×10^{-3}	5.1×10^{-3}	3.8×10^{-2}
$t = 250, Sc = 1$	1.3×10^{-2}	1.4×10^{-2}	6.7×10^{-2}
$t = 250, Sc = 10$	1.6×10^{-2}	2.0×10^{-2}	7.5×10^{-2}
$t = 500\ Sc = 1$	2.6×10^{-2}	3.6×10^{-2}	1.5×10^{-1}

Table 3.1. The normalized L_2-norm ($\|c - c_e\|_2 / \|c_e\|_2$) of the different numerical methods determined at two different moments in time.

3.8 Conclusions on the numerical methods

From the previous section four conclusions about the implemented numerical methods can be drawn:

- The first-order upwind method adds too much numerical diffusion and causes some of the important details to be removed. This method is not suitable for computation with Schmidt numbers that are larger than unity.
- The central differencing and the third-order upwind method both give good results, but a small difference, caused by the additional numerical diffusion is noticeable.
- Due to the absence of numerical diffusion, the central difference method is assumed to provide results that provides the best representation of the physics, but at a Schmidt number of ten, this method becomes unstable.
- With a Schmidt number of $Sc = 0.1$, too much diffusion is present and most details are lost. A visual comparison does not reveal a large difference between the concentration fields that are computed with the different numerical methods.

The goal of this work is to apply LDC to a three-dimensional time-dependent problem at an increased Schmidt number. To be able to investigate the implementation of LDC, a stable and accurate solution is required and therefore the third-order upwind method is preferred over the central difference method and the first-order upwind method. If the implementation of LDC is successful, the less dissipative central difference method can always be re-applied.

3.9 Comparison with particle data

The previous simulations all focused on the concentration of the passive scalar and individual particle tracks are not used. To check if the concentration field matches the position of marked fluid particles, 25000 particles are released into the flow, with the same initial Gaussian distribution. The simulation of the concentration uses the central difference method in combination with a Schmidt number of one. Note that the particle Schmidt number is infinite, because they are not influenced by diffusion.

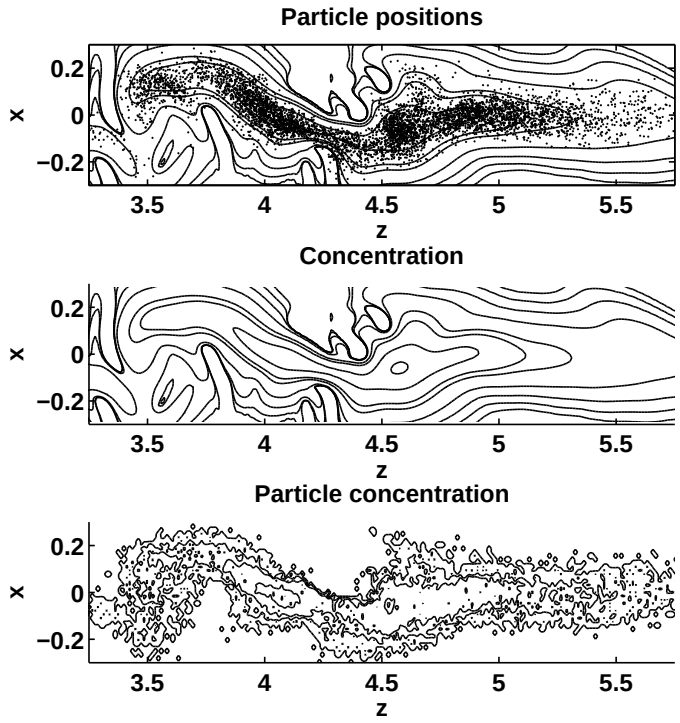


Figure 3.6. Comparison of the concentration field computed from 25000 particles that are projected onto the Z-X plane. The snapshots are made after 500 time steps. Top: the position of individual particles with the concentration field. Middle: the concentration field. Bottom: the particle concentration field.

From the particle data, a concentration field can be reconstructed using,

$$C_{p,j,k,l} = \frac{1}{N_p} N_{c,j,k,l} \tag{3.33}$$

where N_c the represents the number of particles that is present in a control volume, N_p the total number of particles that is released into the flow and C_p is the particle concentration. In figure 3.6, the result of the comparison of the concentration field with the position of the particles is shown at one specific moment in time. The top figure compares the positions of individual particles with the concentration field. The middle figure shows the concentration field and in the bottom figure, the particle concentration C_p is shown. From figure 3.6 can be concluded that the concentration field and the positions of the marked fluid particles are, at this moment in time, in good agreement.

3.10 Conclusion

During the computation of a passive scalar that is released into a turbulent flow, several problems were encountered that needed to be addressed. The main difficulty is found when the Schmidt number is increased to a value larger than one. With an increase in Schmidt number, smaller structures appear within the concentration field of the passive tracer. The finite volume method, in combination with the spectral solver for the velocity, can be used to compute the evolution of a passive tracer in a three-dimensional turbulent channel flow. However, due to the currently available computational resources, the number of grid points cannot be large enough to compute a solution at Schmidt numbers that are larger than unity, without using a dissipate numerical scheme. Even with the usage of such a scheme, the increase in Schmidt number is limited.

However, in many situations, the tracer is only present in a small area of the total domain. This opens the possibility to treat these situations using grid refinement techniques, where the available computational power is focused on those parts of the domain where high activity is present. The idea behind a grid refinement technique is that a small refinement area is created around the high-activity region to accurately compute the local solution. Afterwards, the fine grid solution is combined with a coarser global solution and in this way the accuracy is improved without a large increase in computational costs. From the numerical schemes that were discussed in this chapter, the third-order upwind method provides the best combination of stability, accuracy and efficiency and is used to implement a grid refinement technique.

Before it is possible to treat specific details about grid refinement techniques such as Local Defect Correction or Local Uniform Grid Refinement, some general aspects of a grid refinement method, applied to a time dependent problem, need to be addressed. The next chapter will concentrate on the positioning and movement strategy for the refinement area.

Chapter 4

Refinement area

4.1 Introduction

In a time-dependent problem grid refinement techniques require some additional work compared to the conventional approach. In this chapter the basic requirements for the implementation of a grid refinement method are introduced without specifying any particular strategy. In section 4.2, the coordinate system of the refinement area is introduced. When the initial position of the refinement is defined, the numerical methods from chapter 3 can be used to compute the concentration on the next time level for both the coarse and the fine grid. The only information that is still required are the boundary conditions for the refinement area. This boundary is not located at the boundary of the global computational domain and the concentration values have to be interpolated from the coarse grid, see section 4.3.

One of the difficulties with respect to a grid refinement technique applied to a time-dependent problem is the continuous movement of the refinement area. In section 4.4, following a method that was developed by Bennet and Smooke [8], the detection of the high-activity region is discussed. A small adaptation with respect to time-dependent situations is made to ensure a smooth detection. The actual movement of the refinement area can be different from the new position that was detected earlier and some remarks on this subject are made in section 4.5. When the refinement area is moved into areas where no fine grid information is available, the concentration in the new grid points has to be interpolated from the known coarse grid information. In section 4.6 the method that is used to ensure a smooth and positive concentration field will be explained. In the last section (4.7) of this chapter, the advantages of using multiple, nested levels of refinement are discussed together on how to determine an optimal refinement topology.

4.2 Coordinates of the refinement area

The concentration is computed on a rectangular grid, using Cartesian coordinates, where the refinement area has an independent grid that consists of $n_x \times n_y \times n_z$ grid

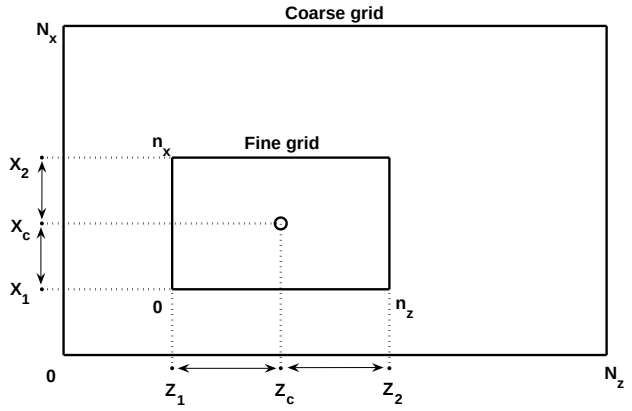


Figure 4.1. Schematic two-dimensional overview of the fine grid coordinates and their position relative to the corresponding coarse grid. The third dimension has similar coordinates.

points in wall-normal, span- and streamwise direction. For each refinement area these numbers are different. A two-dimensional example of the coordinate system is shown in figure 4.1 and table 4.1 compares the coordinates of the fine and coarse grid. As multiple refinement areas in a nested form are allowed, it is convenient to position each refinement area with respect to the origin of its corresponding coarse grid. In that way, multiple systems containing a coarse and fine grid are stacked on top of each other. Depending on the level of refinement, the grid acts either as a fine or as a coarse grid, with the exception of the global coarse grid and the finest level of refinement.

	Coarse grid	Fine grid
Wall-normal grid points	$0 < J < N_x$	$0 < j < n_x$
Spanwise grid points	$0 < K < N_y$	$0 < k < n_y$
Streamwise grid points	$0 < L < N_z$	$0 < l < n_z$
Wall-normal coordinate	$0 < X < X_{N_x}$	$0 < x < x_{n_x}$
Spanwise coordinate	$0 < Y < Y_{N_y}$	$0 < y < y_{n_y}$
Streamwise coordinate	$0 < Z < Z_{N_z}$	$0 < z < z_{n_z}$
Coarse grid origin	$(0, 0, 0)$	-
Center coordinate	(X_c, Y_c, Z_c)	(x_c, y_c, z_c)
Fine grid origin	$(X1, Y1, Z1)$	$(0, 0, 0)$
Fine grid upper corner	$(X2, Y2, Z2)$	(n_x, n_y, n_z)
Grid size	$\Delta X, \Delta Y, \Delta Z$	$\delta x, \delta y, \delta z$

Table 4.1. Overview of the coarse and corresponding fine grid coordinates.

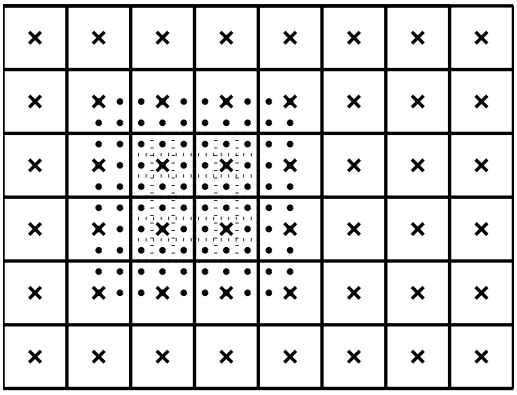


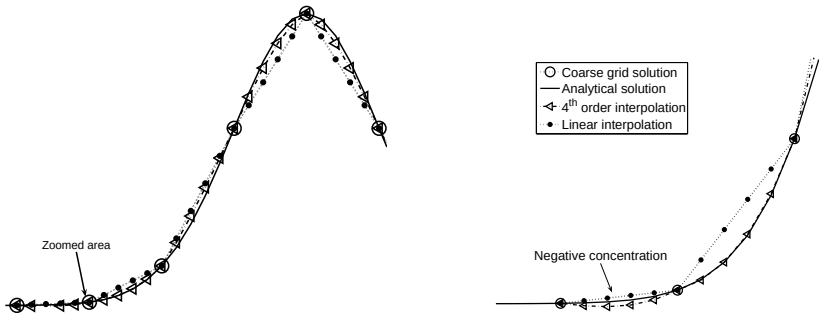
Figure 4.2. Two-dimensional overview of the coarse and fine grid points that are used inside the refinement area. The coarse grid points are marked with a (×) and the fine grid points with a (•). The refinement factor in this example is equal to three.

Refinement factor

One important parameter in grid refinement is the refinement factor σ_r which is defined as the ratio between the coarse and fine grid size. In the wall-normal direction, the refinement factor is defined as:

$$\sigma_{r,x} = \frac{\Delta X}{\delta x} \tag{4.1}$$

where ΔX and δx are the grid size on the coarse and fine grid respectively. The refinement factor for the finite volume adapted LDC method needs to be an odd integer that is equal to or larger than three. With an odd integer in either direction, the coarse and fine grids always have a mutual point. Note that the refinement factor can be different in each direction. It might be useful to be able to refine the grid in only one or two directions, but this feature is not implemented in the current code. Figure 4.2 shows a simplified example of a refinement area that is positioned in the bottom left corner of a coarser grid. The control volumes of the refinement area are indicated with a dashed line and cover four coarse control volumes. Additional grid points are added around the refinement area and act as a buffer region between the coarse and fine grid. The main purpose of the buffer region is to reduce the dimension of the required interpolation by one (from three to two dimensions). A second advantage is that the interpolated boundaries do not directly influence the fine grid solution.



(a) Interpolation overview (b) Zoomed area

Figure 4.3. Example of a fourth-order interpolation. The left figure shows the overview, with the concentration interpolated correctly in the middle. As the curve is symmetric, only the left half is shown. Near the edge, the interpolation fails and linear interpolation is used. The right figure is close-up from the bottom-left part of figure (a).

4.3 Boundary interpolation methods

When initial conditions are supplied to both the coarse and fine grid, the concentration on the next time level can be computed on all points of the refinement area, except for the boundaries. To be able to solve the concentration on the fine grid, boundary conditions have to be specified by interpolation from the known coarse grid concentration. The simplest method to obtain fine grid information from the coarse grid is through linear interpolation. Linear interpolation is as good as the coarse grid representation and usually the shape of the fine grid solution is poorly represented. To improve the accuracy and predicted shape of the interpolated boundary conditions, a higher-order interpolation method is implemented. The chosen method is based on the cubic Lagrangian interpolation method that was earlier used to determine the particle velocity. Despite the higher order of the interpolation, this method does not guarantee a positive and smooth solution. This is demonstrated by a one-dimensional example of a Gaussian distribution that is shown in figure 4.3a. In this figure, the coarse concentration is interpolated using the fourth-order interpolation method. In the center of the distribution, the fine grid solution is interpolated correctly. Near the edge of the refinement area, the interpolation 'fails' as is shown in figure 4.3b. Due to the large second derivative of the concentration, the interpolated concentration becomes non-monotonic. To prevent this, linear interpolation is used, if

$$\big(c_{j+1} - c_j\big)\big(c_j - c_{j-1}\big) < 0, \tag{4.2}$$

Condition (4.2) is also satisfied near an extremum of the coarse grid concentration, where fourth-order interpolation is very well possible. Therefore, fourth-order inter-

	Fourth-order	Linear	exponential fit
base - Δ	7.2×10^{-1}	6.5×10^{-1}	1.1×10^{-1}
base - 2Δ	1.3×10^{-1}	3.0×10^{-1}	2.0×10^{-2}
top - Δ	3.2×10^{-2}	1.0×10^{-1}	-
top - 2Δ	3.5×10^{-3}	2.9×10^{-2}	-

Table 4.2. Comparison of the different methods that are used to interpolate the boundary conditions. In this table, the maximum difference between the exact and interpolated values of the concentration is shown for two values of the grid spacing Δ. For comparison, the values are divided by the average coarse grid concentration in between which the interpolated value was obtained.

polation is always used if,

$$\big(C_{J+1} - C_J\big)\big(C_J - C_{J-1}\big) < 0, \tag{4.3}$$

Using this approach, accurate boundary conditions are obtained and the concentration can be computed on the next time level.

Quantitive comparison

To give a more quantitive comparison between the interpolation methods, the previous example of the Gaussian curve, is used to test the interpolation methods. The methods are judged by the maximum of the absolute difference between the interpolated and the exact solution, divided by the average coarse concentration in between which the interpolation is performed. In this comparison, the method that has the smallest difference is assumed to have the best fit. The results of this comparison can be found in table 4.2. Note that an exponential fit value is given as well, this method is introduced in section 4.6.2 and the results from this exponential fit can be disregarded for the time being. The results of the small comparison are that with a decreasing grid size, the fitted values improve and that fourth-order interpolation is a general improvement over linear interpolation. Except for the base of the exponential, where negative concentrations can occur. In the case that the fourth-order method 'fails' near the base of the exponential, linear interpolation is justified.

4.4 Locating the high-activity region

Before the refinement area can be adjusted, the position of the high-activity region must be determined by evaluating the concentration field. The high-activity region consists of these parts of the computational domain, where large concentrations and/or large concentration gradients are present. When high activity is detected, the corresponding control volume is marked for refinement. In the past, research was performed on steady problems by Bennet and Smooke [7]-[8], Anthonissen [3] and recently on time-dependent problems by Minero [51] and Kramer [34]. The detection

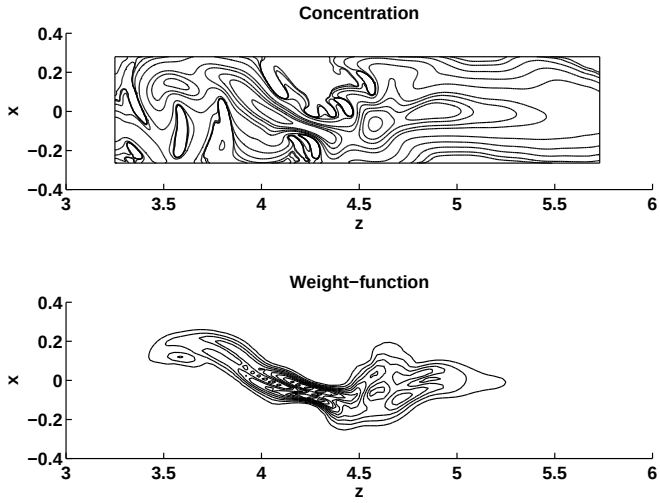


Figure 4.4. Contour plot through the Z-X-plane of the concentration field (top), together with the rectangular bounding box. In the bottom figure a contour plot of the computed weight function is shown at the same moment in time. Computations are performed at a Schmidt number of five and the snapshot is made at after 500 time steps.

method itself is independent of time, because it only analyzes the current concentration field. However, some adaptations are needed to create a detection method that is suitable for a moving high-activity region in a turbulent flow.

Following the method that was introduced by Bennett and Smooke [8], a positive weight function is introduced. The value of the weight function is based on the change in the gradient of the solution and is, after normalization, used to mark the individual control volumes. This normalization is required to make the method suitable for all kinds of different situations with respect to the magnitude of the concentration gradient and grid size.

When the value of the weight function exceeds a certain user defined threshold, $W_{j,k,l} > \eta_{upper}$, the control volume is marked for refinement. Around the marked control volumes a rectangular box is drawn that resembles the refinement area. Figure 4.4 shows a contour plot of both concentration and corresponding weight function. A drawback of this rectangular approach is that, as can already be noted from the figure, the shape of the blob of concentration, affected by a turbulent flow, does not have a rectangular shape. Future adaptations that aim to improve the performance of grid refinement methods should consider implementing a more arbitrary shaped refinement area.

Oscillations in the detection

In the case of a steady solution, the refinement area can be positioned during the initialization of the simulation and does not need much adjustment during the computations. In a time-dependent problem, the solution is continuously evolving and the position of the high-activity region needs to be monitored to keep it inside the refinement area. If the value of the weight function is near the threshold η_u, oscillations in the position or size of the refinement area can occur. To prevent these fluctuations, a second (lower) threshold η_l is introduced that is used to unmark the control volumes. An already marked control volume can only be reset when the normalized weight function drops below this second threshold. In practice, the lower threshold is set to a fixed value, ϵ_l, of the upper threshold,

$$\eta_l = \epsilon_l \eta_u \qquad \text{with} \qquad 0.5 < \epsilon_l < 1. \tag{4.4}$$

During the simulations the value of ϵ_l was set to 0.75. Using both thresholds, the detection method is able to predict a steady rectangular refinement area that captures the high-activity region that is transported along with the fluid flow.

4.5 Movement of the refinement area

Now the desired size of the refinement area is known, it is possible to determine the actual movement of the grid. This is done by comparing the desired new position, detected as described in the previous section, with the current dimensions of the fine grid. Each adjustment can introduce errors and is computationally expensive. To keep the number of adjustments as low as possible, several conditions are imposed that help to determine the minimum and maximum allowed movement.

The minimum movement is limited to one the dimension of the coarse control volume, but maximum allowed movement depends on whether the grid size increases or decreases. When points are removed from the refinement area there are no limitations, because the lower threshold prevents the premature removal of marked control volumes. When the size of the fine grid increases, the concentration in new grid points needs to be estimated and to be successful, the increase in grid size is restricted by the following set of rules:

- Interpolation is most accurate on points that are closest to the old refinement area and the maximum allowed adjustment is restricted to one coarse control volume.
- Priority is set to the finest level of refinement. All levels of refinement are checked each time step. When more than one area needs to be adjusted, the finest grid has priority over the coarser grids as it contains the most accurate information.
- The grid on a certain level of refinement is not allowed to move at two subsequent time levels. This serves two purposes. First, after the interpolation, the concentration field is given some time to 'adjust' to the altered situation.

Second, when multiple adjustments are needed in more than one direction and on multiple refinement levels, this rule prevents the continuous adjustment of one refinement area.

- An adjustment of the refinement area is not allowed to move the refinement area near or outside its parent coarse grid.

Due to the above set of rules, an adjustment that is predicted during the detection method can be overruled. When this happens, the adjustment is postponed to a later time. Note that a very small grid size, in combination with the first and third rule, can create a situation where the refinement area cannot keep up with the high-activity region. In that special situation, the first rule is relaxed.

4.6 Interpolation of new fine grid points

4.6.1 Requirements

The actual movement of the refinement area can be performed without much effort, because the concentration on the old refinement area is copied towards the coordinates of the adjusted grid. If the size of the refinement area increases the new, fine grid, information has to be obtained by interpolation of the known fine and coarse grid concentration. Proper interpolation is the key to a successful application of a grid refinement method in a time-dependent problem. To ensure that the refinement area can be adjusted correctly, the interpolation method needs to meet several conditions:

- The transition from the known fine grid concentration towards the interpolated concentration needs to be smooth. Jumps in concentration or a large difference in the concentration gradient can cause the numerical integration scheme to become unstable. Small differences are allowed because of the presence of numerical and/or molecular diffusion.

- The interpolated concentration needs to be positive as the concentration can physically never drop below zero.

- The interpolated values should not add mass to the system or remove it.

In a three-dimensional problem, meeting all three requirements at the same time is very difficult. Therefore, they are relaxed by allowing some difference in mass. The two reasons behind this choice are that the other two requirements are more important for the stability and at the edge of the refinement area the concentration is small compared to the concentration near the center. The total amount of mass that is added by interpolation, is orders of magnitude smaller then the total amount of mass. Interpolation methods that are found in Salama [69] and Lyman [43] that are mass conserving, are not accurate enough to account for these small differences.

Conservative interpolation method

Minero [51] introduced a method that is able to conserve the total mass of the concentration by imposing the following requirement,

$$C_{j,k,l}V = \sum_{i=1}^{k} c_i v, \tag{4.5}$$

where k denotes all fine grid points that are inside a coarse control volume. After interpolation, the difference between the left end right-hand-side of equation (4.5) is computed. This difference is multiplied by a scaling factor based on the height of the concentration and added to the fine grid concentration. The scaling factor is determined by dividing the concentration on each fine grid point by the sum of the concentration at all fine grid points that lie inside one coarse control volume. Due to turbulence, the magnitude of the difference that needs to be corrected is not predictable. When steep gradients are present, a relative large difference in concentration needs to be redistributed and a smooth and positive solution is not guaranteed.

4.6.2 Exponential fit

When the boundary interpolation methods from section 4.3 are applied, it is found that both linear and fourth-order interpolation do not satisfy the requirements for both a positive and a smooth solution. When the solution of the convection-diffusion equation is examined in more detail, it shows that it is usually exponentially shaped near the boundary of a refinement area. This property can be used to fit an exponential function through the known concentration values. Fitting an exponential function in three dimensions is computationally expensive and difficult. Instead the assumption is made that the exponential can be fitted in the direction in which the refinement area is adjusted. This reduces the three-dimensional interpolation towards a one-dimensional fit, which is much cheaper.

To demonstrate this approach, the grid size is increased in the streamwise direction as an example. The points on the fine grid have a refinement factor equal to three and need to be interpolated from $0 < j < n_x$, $0 < k < n_y$ and $n_z - 2 < l < n_z - 1$. The boundary of the new refinement area is located at z_{n_z}. If the refinement area is moved, the last known fine grid concentration, from before the adjustment, is located at z_{n_z-3}. The exponential fit is based on the following function,

$$c_{f,j,k,l} = \alpha e^{(\beta z_l - \lambda)} + \gamma, \tag{4.6}$$

where c_f is the fitted concentration, α, β and γ are the fit parameters and λ is a coordinate offset that allows the fit to be used independent of the position. This function is always positive when α and γ are larger than zero. The parameters are obtained by performing a nonlinear least-square fit through the known concentration values [21]. Note that the boundary points of the refinement area can be interpolated using the methods that was described in section 4.3. To ensure a smooth transition between the coarse and fine grid, the new fine concentration is scaled to match both sides. This is done in two steps. First the difference between the fitted concentration

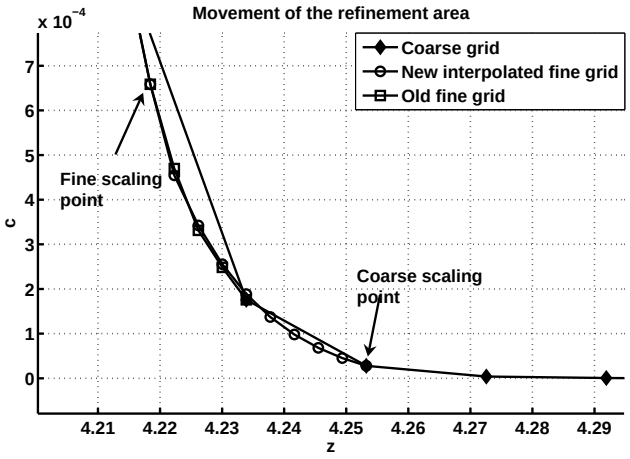


Figure 4.5. Exponential fit of the new concentration, after the refinement area has moved. The figure shows the coarse grid as well as the fine grid concentration before and after the interpolation. The points used for scaling the exponential fit are denoted by arrows.

and the boundary concentration is determined and is added to the fitted exponential to match the coarse grid. Second, the exponent is scaled towards a known fine grid point, using a linear scaling function. In figure 4.5 a result of the above exponential fitting procedure is shown. It can easily be seen that this interpolation returns positive values of the concentration, matches both the coarse and the fine grid and furthermore preserves the shape of the concentration. In some cases, e.g. where the coarse concentration is too small or the coarse grid solution is not monotonic, an exponential fit does not work and linear interpolation is applied. Apart from figure 4.5, the method was tested on a one-dimensional Gaussian curve identical to one that was used in section 4.3. Results of this comparison are found in table 4.2 and demonstrates that this approach gives the best results when compared to the fourth-order and linear interpolation methods.

4.7 Multiple levels of refinement

4.7.1 Advantages

In the previous sections, only one level of refinement was considered. However, as was already mentioned, multiple nested levels of refinement can be used to focus the available computational power to an even finer area. Each level of refinement can act as a coarse or a fine grid and the detection, movement and interpolation strategy was designed from this point of view.

The use of multiple refinement levels has several advantages over one single level of refinement. Multiple levels of refinement keep the refinement factor low, which

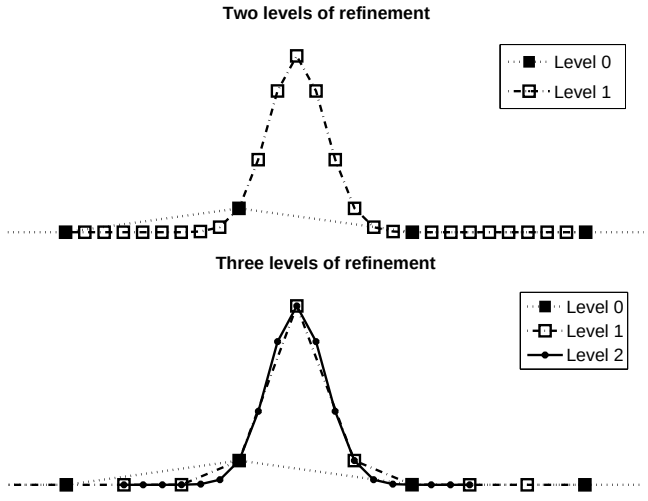


Figure 4.6. A one-dimensional example of the difference in the number of refinement levels and their corresponding refinement factor. In both cases the grid size on the finest levels is identical. The top figure has a refinement factor of nine, whilst the lower figure has two times a refinement factor of three.

is beneficial for the various interpolations. When the difference in grid size is large, the shape of the fine grid solution is not clearly present in the coarse grid solution and the interpolation method is not able to accurately determine the unknown fine grid values. A second advantage is that the positioning of the high-activity region is more precise. Each level of refinement can be adjusted individually with respect to its corresponding coarse grid. With better positioning, the total number of grid points that is required per level of refinement decreases and this is beneficial for the efficiency. An example of the difference between the application of two or three levels of refinement is shown in figure 4.6, using a one-dimensional Gaussian curve. In both situations, the grid size on the finest level is equal, but in the top figure one refinement is used with a factor of nine, whereas in the lower figure two levels with each a factor of three are applied. The difference in grid size and positioning can clearly be noticed.

Instead of nesting levels of refinement, it is possible to use refinements in parallel. One can think of a two-dimensional dipole collision where the two vortices each move to one side [34],[51]. With multiple, parallel refinement areas, both vortices can be tracked individually. Another example is when multiple high-activity regions are present during the initialization. However, parallel refinement is not used in any of the numerical simulations that are presented in this thesis and is mentioned as a recommendation for future development on grid refinement methods.

4.7.2 The optimum number of refinement areas

While having more levels of refinement has many advantages, the computational time that is needed to perform one complete time step significantly increases with each additional level. Because the Schmidt number and initial conditions are known, the finest grid size can be computed. From the finest grid, the coarser grids can be constructed until the whole domain is covered and a trade-off has to be made between the number of levels and the refinement factor on each level.

Two different, opposite situations can be distinguished during the initialization of the problem. The first situation occurs when the high-activity region is large and is relatively smooth. In this case it is beneficial to have a larger refinement factor in combination with a low number of refinement levels. Because the minimal refinement factor is equal to three, a large refinement area, in combination with more than two levels of refinement, will lead to too much grid points on the finer levels. The exact opposite situation is found when the high-activity region is small, but contains large gradients. Because of the limited size of the finest level, more refinement areas with a smaller refinement factor (three or five) can be used. Furthermore, a small high-activity region requires a more demanding positioning and with multiple levels of refinement this is much easier.

The trade-off between the number of refinement levels, the refinement factor and the necessary computational time depends on many different factors and can, at this point, only be set during the initialization phase. The initial shape of the blob of concentration changes rapidly under the influence of turbulence, but it is not possible to change the refinement topology during the simulation. If LDC is further developed, the refinement topology should be allowed to be adjusted during the computations in order to obtain the optimal strategy for every situation.

4.8 Conclusion

With the detection method that is described in this chapter it is possible to create a moving refinement area that travels along with the flow. A small buffer region is positioned around this refinement area that helps to reduce the required interpolation with one dimension. The boundary conditions are computed with a fourth order accurate Lagrangian interpolation method. Using a slightly adapted detection method from Bennet and Smooke [8], the time-dependent movement of the refinement area can be predicted accurately. When the refinement area is moved into areas where no fine grid information is available, a one-dimensional exponential function is used to predict the unknown fine grid concentration. This approach ensures a smooth and positive transition between the coarse and the fine grid values, but the total amount of concentration is not conserved. The detection, movement and interpolation strategy is developed with more than one level of refinement in mind and additional levels can be added without much effort. However, the optimal refinement topology depends on the problem and needs to be set manually for every situation.

Chapter 5

Local Defect Correction

5.1 Introduction

In chapter 3 it was shown that without a grid refinement technique, it is very difficult to compute the evolution a passive tracer if the Schmidt number is larger than unity. In chapter 4 the methods that are needed to position and move a refinement area were introduced. The present chapter will focus on the implementation and application of a specific local grid refinement method, called Local Defect Correction [20], to a time-dependent three-dimensional problem: the concentration equation for a passive scalar in a turbulent channel flow. The fist section of this chapter (5.2) explains the general idea behind LDC, together with some of the work that has already been done on this subject. After the general introduction, an explanation of the numerical implementation of LDC, following a method introduced by Minero [51], is given in section 5.3. In this section all individual steps that are needed to successfully implement LDC to a convection-diffusion equation, are discussed together with the additions that are required to apply more than one nested level of refinement. Section 5.4 explains the numerical methods that are needed to solve the system of equations using an implicit solver. During the implementation and testing of LDC it became clear that the limited possible number of grid points was one of the major difficulties for a three-dimensional problem. Due to the maximum allowed number of grid points, the constraint for a smooth coarse grid [3] cannot always be satisfied. In section 5.5, a solution to this problem is shown. When this improvement is applied, the required number of LDC-iterations is determined in section 5.6 and in section 5.7 the results of several numerical simulations are shown. Another grid refinement method that is available, next to LDC, is Local Uniform Grid Refinement. The difference between the two is explained in section 5.8. The final section (5.9) will discuss the applicability and efficiency of LDC.

5.2 Local Defect Correction

5.2.1 The basic principle

For increasing Schmidt number the grid size needs to be decreased in order to capture the smaller structures that appear in the concentration. In a three-dimensional problem, the total number of required grid points increases rapidly, beyond the point that is affordable with the current computational power. A promising approach to this problem is to apply a grid refinement technique that concentrates the available computational resources in those places in the domain where high activity occurs. Grid refinement methods are based on the fact that a solution computed on a fine grid is more accurate than the solution of the same problem computed on a coarser grid. Around the high-activity region, a fine grid is located that is able to accurately compute the detailed solution to the problem. This refinement area is much smaller than the global domain, but covers the high-activity region. The global grid is fine enough to provide a reasonably smooth solution, but too coarse to capture all the details. Grid refinement methods combine both the fine and the coarse grid solution to obtain an optimal solution to the problem. The simplest form of grid refinement contains just one single refinement area on top of the global grid. The previous chapter already mentioned that the different refinement grids used in this thesis must be uniform and rectangular. This is not required for grid refinement methods to function, but it does simplify the movement of the refinement area and the interpolation strategy.

The difference between Local Defect Correction and other grid refinement methods becomes clear when both the coarse and fine grid solution are computed. LDC uses the fine grid information to improve the whole global solution. Other methods, such as LUGR [75], do not have this feature. The actual correction of LDC is preformed via a defect term that depends on the difference between the coarse and fine grid approximations. The defect is fed back, via a source term, into the original set of equations for the coarse grid, but only exists inside the refinement area. In LDC an implicit solver must be used to compute the (corrected) coarse grid solution. This is required if the local defect term is to influence the global solution. With the updated/corrected coarse solution, a, presumed better, estimate of the boundary conditions on the refinement area is obtained and the fine grid solution is re-computed using these better boundary conditions. The corrected fine and coarse grid solution can be used as approximations for a second LDC iteration to further improve the solution. This process can be repeated until convergence. The efficiency and usability of the LDC method are directly coupled to the number of iterations that are required to reach convergence. The convergence properties of LDC applied to an elliptic problem were studied by Anthonissen [3] and Minero [47], [51]. Minero found that LDC applied to a time-dependent problem is unconditionally convergent and in practice one iteration is sufficient.

5.2.2 Recent applications of LDC

Local Defect Correction was first developed by Hackbush [20], who used it as a discretization method for elliptic boundary value problems on a composite grid. In 1995 Ferket [14]-[15] used LDC in combination with a finite difference methods on several combustion problems. Later on, Anthonissen [1]-[3] and Van 't Hof [23] adapted the LDC method in order to use it in combination with a finite volume discretization method. The adaptation of the original LDC approach was needed to maintain a conservative solution over the boundaries of the refinement area. This adapted LDC method was later used to study the behavior of flow and heat transfer in a glass tank by Nefedov [58]. In the above examples, LDC is applied in combination with various discretization methods. This is possible because LDC is not a numerical method, but a strategy to solve problems numerically. An exception to the available discretization techniques are spectral methods, that need periodic boundaries. Even when the global solution satisfies the requirement for periodic boundaries, the refinement area generally does not. Furthermore, the solution of a spectral method is a summation over the all the Fourier modes and it is not possible to locally apply a defect. This is one of the reasons to change from a spectral flow solver to a finite volume method for the convection-diffusion equation, see chapter 3.

Other authors have discussed the usage of composite grid methods. McCormick [46] proposed the Fast Adaptive Composite (FAC) grid algorithm that uses a Finite Volume Element (FVE) method. The method is almost identical to LDC, with the exception that LDC is able to combine standard discretization methods on uniform grids, whereas the FAC method requires the composite grid to be known beforehand. Another method is the Multi-Level Adaptive Technique (MLAT) that was studied by Brandt [6]. The difference between LDC and MLAT is in the details of the treatment of the solution inside the refinement area. In LDC, this solution is computed exactly while for the MLAT algorithm only an approximation to the fine grid solution is computed. Another difference is that the refinement factor, the ratio between the coarse and fine grid size, is typically a factor of two for MLAT, whereas in LDC this factor can be larger.

5.2.3 LDC applied to a time-dependent problem

Two-dimensional problems

In the previously described work, LDC was mainly applied to efficiently solve steady, two-dimensional problems. Both properties are beneficial for grid refinement methods and especially LDC. In two dimensions, the number of grid points that are available for each direction is large and a smooth coarse grid solution can be assured. In three dimensions this property cannot be taken for granted.

Steady problems are ideal because the position and size of the refinement area can, beforehand, be determined with high probability. Some adaptations to the size and position of the refinement area are sometimes needed, but these are not as common and frequent as in a time-dependent problem. The combination of LDC and time-dependent problems gives rise to an additional problem: the treatment of the defect term. The defect is the difference between the coarse and fine grid solution and needs

to be determined and applied at the correct moment in time. Recently, Minero [48]-[51] extended the finite volume adapted LDC method to time-dependent problems. The main reason to apply LDC to a time-dependent problem is to allow transport problems to be solved with very high efficiency and accuracy, without sacrificing computational costs. As a test-case, Minero investigated the transport of a passive tracer in a two-dimensional turbulent flow: a dipole collision with a wall. This flow is highly turbulent and small-scaled structures appear within the tracer field, when it is affected by the passing vortex. The results of these simulations were compared with results from Kramer [33]-[34], who investigated the same problem using a spectral solver. The result of this comparison is that both strategies produce similar results and this demonstrates that LDC is capable of handling two-dimensional time-dependent problems. When the computational effort needed for LDC is compared with the spectral method, both methods perform almost identically. However, recommendations are made to further improve the efficiency of the LDC-method. Without any gain in efficiency, it is not meaningful to apply LDC instead of the spectral method.

Three-dimensional problems

To check the application, efficiency and usability of Local Defect Correction to an even more complex problem, LDC will be applied to compute the evolution of a passive tracer in a three-dimensional turbulent channel flow. Several differences can be pointed out between the work by De Hoogh [26]-[27] and the work by Minero [48]-[51] and Kramer [33]-[34]. The major complication when moving from two to three dimensions is the large increase in grid points. For a large number of grid points the limit of current computational resources is rapidly reached. Furthermore, LDC requires that the global coarse grid solution is reasonably smooth [3]. The smoothness of a solution is directly coupled to the grid size and for a limited number of available grid points, the required smoothness on the coarse grid is difficult to achieve. Even if the smoothness can be reached on the coarse grid, it remains to be seen whether sufficient computational resources are available to be used on the refinement area.

Consider the shape of a blob of concentration that is inserted in a turbulent flow. Under the influence of the turbulent flow field, the initially small blob will be stretched or even split into multiple smaller blobs. In order to have a single refinement area cover the whole high-activity region, many more grid points are needed than during the initial phase. In view of the requirement that the refinement area is rectangular, even more points are necessary as the stretched blob of concentration will probably not have a perfectly rectangular shape.

Quantitative comparison of LDC with other methods often poses a problem. Chapter 3 already demonstrated that, without a grid refinement method, it is not really possible to accurately compute the evolution of a small blob of concentration in a turbulent flow. On the other hand, it is very difficult to reproduce the simulations that were performed in chapter 3 with LDC. This is not because of the fact that the number of grid points becomes too large, but if the size of the finest grid has to match the grid size of the conventional simulations, the coarser grids become too coarse to maintain stability. Another problem is that the coordinates of the moving refinement area do not match the coordinates of the conventional approach. This

requires interpolation of one of the two data fields, which can introduce additional errors.

Despite these drawbacks, in this chapter the application of LDC to a time-dependent three-dimensional problem is being investigated. The next section will discuss the details concerning the implementation of LDC.

5.3 Implementation of LDC

5.3.1 LDC: Step by Step

This section will demonstrate the steps that are needed to apply LDC to a time-dependent problem. From this point onwards, all upper-case symbols denote quantities on the coarse grid. Lower-case symbols refer to the same quantities on the fine grid. In this explanation, the example refinement area that was shown in figure 4.2 is used. The figures that are needed to clarify the flux terms that are involved in each step show a close-up of this figure around the refinement area. The refinement factor, see equation (4.1) is in each direction set to three and is the ratio between the coarse and fine grid size.

Following the approach that was developed by Minero [51], the application of LDC is explained step by step. The steps that have to be taken are treated individually using the convection-diffusion equation for a passive tracer. The simplified, one-dimensional notation, that was introduced in section 3.4, is continued here. To successfully apply LDC to a time-dependent problem, the following steps have to be taken:

1. **Compute the coarse grid approximation**

   The coarse grid is computed using a second order accurate Crank-Nicolson scheme,

   $$\frac{C_{a,j}^n - C_j^{n-1}}{\Delta T} + \frac{Q_j^n}{2V_j} + \frac{Q_j^{n-1}}{2V_j} = 0, \tag{5.1}$$

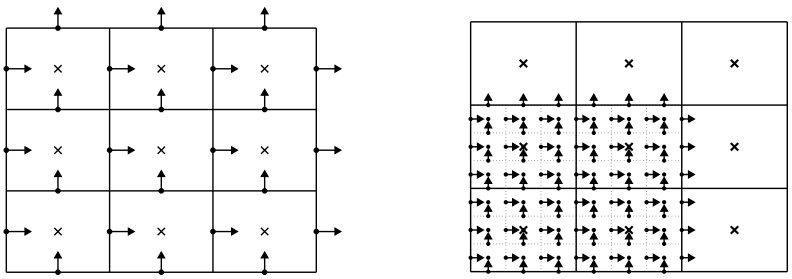
   where C_a is the approximate concentration and Q a flux term that contains both the convective and the diffusive flux terms. Figure 5.1a shows the coarse grid points with their corresponding flux terms. Note that all terms on time level t^{n-1} are known and the goal is to find the approximate solution at time t^n.

2. **Compute the fine grid approximation**

   On the fine grid, no corrections need to be applied and it is possible to apply the first order accurate Euler forward method. The equation to determine the fine grid solution becomes,

   $$\frac{c_{a,j}^n - c_j^{n-1}}{\Delta t} + \frac{q_j^{n-1}}{v_j} = 0. \tag{5.2}$$

   The advantage of treating the fine grid with an explicit method is that it is efficient, especially in a three-dimensional problem. The grid points, together



(a) coarse (b) fine

Figure 5.1. Schematic overview of grid points and flux terms needed to compute the coarse (a) and fine (b) grid approximations. Note that this figure is a close up of the refinement area that was shown in the previous chapter (figure 4.2)

with their corresponding flux terms are shown in figure 5.1b. A detail that needs mentioning is the difference in time step on the different levels of refinement. The time step is generally smaller on the refinement area and multiple steps have to be taken to reach the correct moment in time. In each step, the boundary conditions and the velocity field are computed using linear interpolation in time. For the boundary conditions the interpolation is performed between the old coarse grid solution and the solution computed in step 1.

3. **Assemble the best possible coarse grid composite solution**

   The composite solution, C, is the best possible combination of the, at this point, known solutions and is constructed as,

$$C_j^n \begin{cases} C_{a,j}^n & \text{outside the refinement area} \\ c_{a,k}^n & \text{inside the refinement area} \end{cases} .$$

   with k the points where the coarse and fine grid coincide. Figure 5.2a shows the grid points that are involved in the assembly of the composite grid solution. Other grid refinement methods stop at this point and use the composite solution as the updated coarse grid solution. This means that a lot of information from the fine grid (those values of concentration that are not in the coinciding grid points) is not used to improve the coarse grid solution. In LDC this information is used to determine the best possible coarse grid solution, created from the fine grid information.

4. **Create a best possible solution on the coarse grid**

I Best possible concentration: To be able to compute the defect term with the highest accuracy, the best possible fine grid solution has to be determined. Three possible solutions can be found of which two are trivial as they do not use the fine grid information and are the coarse approximation from step one and the composite solution from step three. The third option is to take the average of the fine grid concentration. Because of the finite volume method, the average concentration value, determined from the fine grid points, is found by,

$$C^n_{b,j} = \frac{1}{N}\sum_{k=1}^{N} c^n_{a,k}, \tag{5.3}$$

where k denotes all the fine grid points that are inside the coarse control volume and C_b is the best possible concentration. This summation of fine grid values is only possible inside the refinement area. Outside the refinement area the composite solution is used, which results in the following scheme for the best possible coarse grid solution:

$$C^n_{b,j} \begin{cases} C^n_{a,j} & \text{outside the refinement area} \\ \frac{1}{N}\sum\limits_{k=1}^{N} c^n_{a,k} & \text{inside the refinement area} \end{cases}. \tag{5.4}$$

II Best possible flux: An identical approach needs to be applied to compute the best possible approximation for the flux of the concentration. As the flux of concentration is computed from the concentration itself, it is possible to distinguish four different flux approximations. The first two are again trivial, being either the fluxes computed using the coarse grid approximation from step one or the composite grid solution from step three. The third and fourth approaches are based on the summation of the fine grid flux terms, computed with the fine grid approximation from step 2, and are denoted with q_a. The expression for the flux summation becomes,

$$Q^n_{b,j} = \sum_{k=1}^{N} q^n_{a,k}, \tag{5.5}$$

with Q_b the best possible flux. Note that the time level at which the flux terms is evaluated is not yet specified. The flux is a measure for the concentration that is moving from one control volume towards the other, during a specific period of time. Because of the difference in time steps on the coarse and the fine grid, two methods are available. The third approach to compute the best approximation for the flux uses the fine flux at the next coarse time level, multiplied with the coarse times step,

$$Q^n_{b,j}\Delta T = \sum_{k=1}^{N} q^n_{a,k}\Delta T. \tag{5.6}$$

The fourth method also uses the higher accuracy of the small time step and results in a second summation of the flux at every time level on the fine grid. When applied to equation (5.5), the expression for the fine grid flux approximation becomes,

$$Q^n_{b,j}\Delta T = \sum_{i=1}^{\tau}\sum_{k=1}^{N} q^{n-1+i/\tau}_{a,k}\delta t. \tag{5.7}$$

where τ is the fraction between the coarse and the fine grid time step,

$$\tau = \frac{\Delta T}{\delta t}. \tag{5.8}$$

The best possible flux approximation on the whole coarse grid is defined as,

$$Q^n_{b,j}\Delta T = \begin{cases} Q^n_{a,j}\Delta T & \text{outside the refinement area} \\ \sum\limits_{i=1}^{\tau}\sum\limits_{k=1}^{N} q^{n-1+i/\tau}_{a,k}\delta t & \text{inside the refinement area} \end{cases}, \tag{5.9}$$

5. **Determine the defect term**

   Take the difference between the composite solution and the best possible solution for both the concentration,

$$D^n_{c,j} = (C^n_j - C^n_{b,j}), \tag{5.10}$$

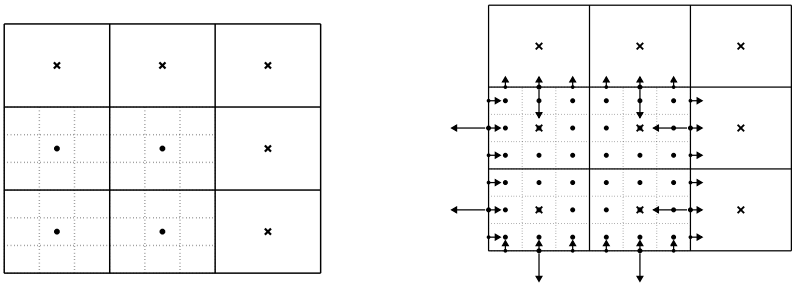
$$D^n_{q,j} = (Q^n_j - Q^n_{b,j}), \tag{5.11}$$

   Note that the defect term is equal to zero outside the refinement area because the best approximations for both the flux and the concentration are equal to the composite solution. An exception to this rule is at those points where the fluxes through the boundary of the refinement area are balanced. On the edge of the refinement area, special care has to be taken to maintain the balance between the coarse and the fine grid fluxes. After the application of the defect, the balance must still be present and is ensured by replacing the coarse grid flux by the sum of the fine grid flux at the edge of the refinement area. This is identical to how the defect is computed in the inner region of the refinement area, with the exception that this only covers the flux through the shared interfaces. Figure 5.2b shows the flux terms that are needed to compute the defect term on the edge of the refinement area. If the flux terms are not balanced, a discontinuous solution can appear. For some practical considerations for the implementation of the defect term the author would like to refer to the work by Minero, [51], p73-76.

6. **Correct the coarse grid solution**

   The coarse grid solution can be corrected by applying the defect terms to each term of equation (5.1),

$$\frac{(C^n_{c,j} - C^{n-1}_j) - (D^n_{c,j} - D^{n-1}_{c,j})}{\Delta T} + \frac{(Q^n_j - D^n_{q,j})}{2V_j} + \frac{(Q^{n-1}_j - D^{n-1}_{q,j})}{2V_j} = 0 \tag{5.12}$$



(a) Composite grid (b) Boundary defect

Figure 5.2. Schematic overview of composite grid (a) and the fluxes that are needed to determine the defect term on the edge of the refinement area (b). The fluxes are in opposite direction because the sum of the fine flux replaces the coarse flux on each surface of a coarse control volume.

and solve the updated system of equations to obtain the corrected solution $C^n_{c,j}$.

7. **Update the boundary conditions of the refinement area**

   After the correction step, the coarse grid solution is altered. With an updated coarse grid solution, the boundary conditions of the refinement area can be recomputed and are assumed to be improved with respect to the coarse grid solution from step 1. With the updated boundary conditions, the fine grid solution needs to be recomputed. As the only feedback from the coarse to the fine grid is through the boundary conditions, this step is important.

8. **Compute the updated composite solution**

   From the updated local and corrected coarse solution a new composite solution can be constructed that contains the corrected solution on both the fine and coarse grid.

9. **Iterate from point 3 until convergence is reached**

   Convergence is reached when,

$$\|C^n_{c,w} - C^n_{c,w-1}\|_2 < \text{tolerance}, \tag{5.13}$$

   where the index w stands for the iteration number. Note that Minero [47] has shown that for LDC applied to a time-dependent problem, convergence is fast and in practice only one LDC-iteration is required.

Using the above approach, it is possible to apply Local Defect Correction to a time-dependent problem. To be successful, the different steps need to be performed at the

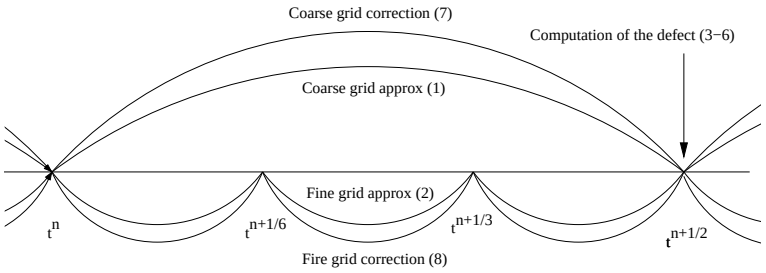


Figure 5.3. A schematic overview of the time levels involved in one LDC-iteration step, with $\tau = 3$. The numbers correspond to the numbers used in the previous section. Note that the time step on the coarse grid is set to half the time step of fluid velocity.

correct moment in time. Figure 5.3 shows the different steps at their correct moment in time that are needed to perform one complete LDC iteration, where $\tau = 3$. This figure serves furthermore as a schematic summary of the steps that are required to perform LDC in a time-dependent problem.

5.3.2 Restriction on the numerical methods

Since LDC is a computational strategy, it is independent of the numerical method that is used to determine the flux terms. However, the method needs to be identical on the coarse and the fine grid. In theory, this requirement is not necessary for LDC to function correctly. In principle only the higher accuracy of the solution inside the refinement area is used and it should not matter how that solution is obtained. In practice, however, a difference in numerical methods causes discontinuities at the edge of the refinement area, where the flux and the defect have to be balanced.

Consider the example that the solution on the coarse grid is computed using an upwind method, whereas the solution inside the refinement area is computed using the central difference method. On the coarse grid, numerical instabilities are damped due to the added numerical diffusion with the disadvantage that the solution spreads faster. On the fine grid, no additional numerical diffusion is present. When the boundary conditions are computed or when the refinement area is moved, the coarse grid solution is interpolated to give an estimate of the fine grid solution. The interpolated values are affected by the additional numerical diffusion and in a channel flow this effect is very much noticeable in streamwise direction. The interpolation of new values of concentration is always at the downstream side of the refinement area. When the new fine grid solution is interpolated on the coarse grid, the fine grid solution is affected by the additional numerical diffusion. Because of the continuous movement in one direction, the influence of the added numerical diffusion is, during each downstream movement, pushed further into the refinement area. Despite the usage of the central difference method, the fine grid solution is, albeit indirectly, affected by numerical diffusion.

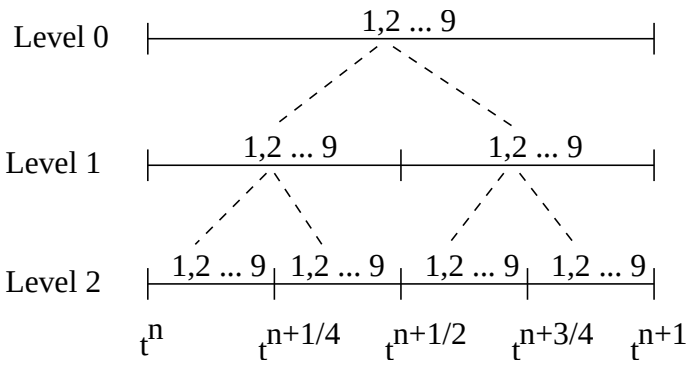


Figure 5.4. A schematic overview of the numbered steps that are needed to apply LDC on three levels of refinement. The numbers of the steps correspond to the LDC-procedure that was explained in section 5.3. In this figure level 0 corresponds to the global grid and, with an increasing number, the grid becomes finer.

5.3.3 Multiple levels of refinement

The computation and application of the defect terms is independent of how the fine grid solution is obtained. This means that it is possible to nest several layers of defect correction, see section 4.7. Consider two nested levels of refinement. The solution on the middle grid is corrected using the information from the finest level. After the correction is applied to the middle grid, the global grid is corrected using already improved information from the middle level of refinement. For multiple nested levels of refinement the LDC-approach is almost identical to the previously described procedure. Each combination of subsequent levels can be treated as an individual set of a coarse grid containing a refinement area. During the computation, only the direct neighbors of a specific refinement area are affected either through the application of the defect (coarser grid) or the boundary interpolation (finer grid). When one LDC time step is completed, the subsequent application of defect terms allows the finest grid to influence the global solution. The only adaptation that needs to be made when more than two levels of refinement are present is the way in which the fine grid solution is obtained. On the finest level, the explicit Euler forward method is used to compute the solution. On every other level, a defect correction needs to be applied and the computation of the fine grid solution (step 2) is replaced by a complete LDC-process. A schematic drawing is given in figure 5.4, where the numbers correspond to the individual steps that were discussed in section 5.3. Because the time step is reduced by a factor of two on each nested level, more steps are needed to reach the same moment in time on the coarse and fine grid. More levels of refinement can be added in a similar way, but the total number of correction steps will increase rapidly.

5.4 Implicit solver

When the implementation of LDC to a time-dependent problem was explained in section 5.3, the details on how to actually compute the (corrected) coarse grid solution, were not revealed. It is not difficult to assemble a system of equations from the discretized version of the convection-diffusion equation. This system can be solved by either sweeping the matrix or using an iterative method. The system of equations that is obtained needs to be solved at least twice, for each level of refinement, depending on the number of required LDC-iterations: the first time to compute the coarse grid approximation and, at least, a second time to apply the defect correction. The defect is added to the right-hand side of the system of equations as a source term.

For three-dimensional problems, the size of the matrix that holds the system of equations becomes problematic. Because LDC is an iterative solution strategy and each level of refinement has its own system of equations, the same type of system must be solved several times per time step and this requires a solver that is as efficient as possible. Consider a computational grid that contains 50x50x100 grid points. The size of the matrix that holds the whole system of equations becomes $(2.5 \times 10^5)^2$. The amount of memory that is needed to store the complete matrix would be enormous, not mentioning the total number of operations that are needed to solve that system. If the matrix, obtained from the system of equations, is examined in more detail, it appears that a very large part is equal to zero and all non-zero elements are located on a small band. For a central difference method the total number of non-zero elements is roughly equal to 2×10^6. Using the third-order upwind method, the total number of non-zero matrix elements is increased by a factor of two, but is still small. A matrix with this property is called sparse and specially designed solvers can be used to compute the solution efficiently. The main advantage of a sparse solver is the fact that only the non-zero elements are used to compute the solution. This drastically reduces the number of operations. A second advantage is that only non-zero elements, with their corresponding location, need to be stored into memory.

All computations are performed using a software package called SPARSKIT, that is developed by Saad [68]. This package provides an iterative sparse solver, that can use different strategies, CG, BCG, GMRES, BCGSTAB, etc. to solve a large, sparse system of equations. Each method has different properties, mostly related to the efficiency of the algorithm and the convergence rate. There is hardly any difference in the solution obtained by the different algorithm as there is in computational time required. BCGSTAB is the fastest method (by 2%) and is therefore used in the simulations throughout this thesis. For more details about numerical algorithms, the author refers to Heath [21] and Vuik [78].

5.5 Difficulties with respect to LDC

5.5.1 Local Schmidt number

One of the major difficulties when LDC is applied to a time-dependent three-dimensional problem is the limited number of grid points that can be used. This makes it difficult

to satisfy the condition for a relatively smooth coarse grid solution [3]. The ideal solution would be to apply different numerical methods on each level of refinement, but this is not possible for the reasons mentioned in section 5.3.2. As the goal of LDC is to provide accurate results, the usage of a very dissipative method such as the first-order upwind method, is not wanted. This means that a solution for the stability problem on the coarser levels of refinement needs to be found.

In chapter 3 it was demonstrated that a decrease in Schmidt number has the same effect as an increase in the numerical diffusion. Therefore, if an oscillation is detected on the coarser grid, the Schmidt number is locally decreased to have the artificial diffusion smooth the solution. This decreased local Schmidt number is applied until the oscillations have disappeared. The actual value of the local Schmidt number can be determined by the stability analysis that was performed in section 3.6. Using the requirement for a TVD scheme, it is possible to give an estimate of the Peclet number that guarantees a stable solution for a central difference method. It is assumed that for the third-order upwind method equation (3.30) overestimates the amount of diffusion that is required. Because the Schmidt number is only temporarily decreased, this is not considered a large drawback. In the following derivation only the streamwise direction is used to demonstrate how the value of the Schmidt number can be determined. Using the fact that $Pe = ReSc$, equation (3.30) is rewritten into,

$$Sc_l = \frac{2}{|u^n_{z,j,k,l}|\Delta Z Re_l}, \tag{5.14}$$

where Sc_l and Re_l are the local Schmidt and local Reynolds number. The Reynolds number is a property of the flow field and it is not adjusted and the local Reynolds number equals $Re_l = Re = Pe/Sc$. The local fluid velocity can be replaced by the maximum value of the streamwise velocity component, to match the worst-case scenario. Using that simplification, the local Schmidt number is determined as,

$$Sc_l = \frac{2Sc}{\Delta Z Pe}. \tag{5.15}$$

The locally decreased Schmidt number ensures that numerical diffusion is only added where oscillations are detected. If the solution is smooth again, the Schmidt number is increased to the global value. Furthermore, the approach is independent of the applied numerical method and does not break the conservation of concentration that is accounted for by the finite volume method. This is possible because the Schmidt number acts on the diffusive flux term between two control volumes. Only on the edge of the refinement area, special care has to be taken that the defect term is computed with the correct value of the Schmidt number. Note that the Schmidt number is never decreased on the finest level of refinement as this would remove the details inside the high-activity region.

Results

To give an impression of the improvement of this approach, two nearly identical simulations are performed, where the only difference is the application of the locally decreased Schmidt number. The simulation uses three levels of refinement at a Schmidt

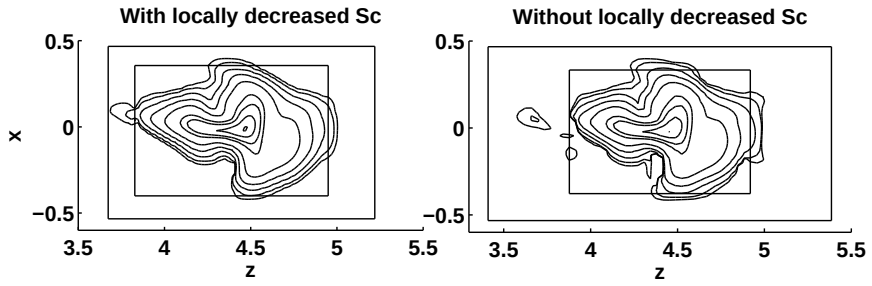


Figure 5.5. A snapshot of a contour of concentration that is obtained with and without the usage of the locally decreased Schmidt number. The figure on the left shows the solution with the local decrease in the Schmidt number and right figure without this. On all but the finest level, the Schmidt number is locally decreased.

number of one. The initial condition for the concentration is a Gaussian shaped blob of concentration with $\sigma_x = \sigma_y = \sigma_z = 1 \times 10^{-1}$ with, on the finest refinement level, a grid size of 6.5×10^{-3} in each direction and a time step of $5 \times 10^{-3}/8$. The results of these simulations are shown in figure 5.5.

With the local Schmidt number, the solution outside the refinement area is much smoother and this automatically affects the fine grid through the interpolation of the boundary conditions. The addition of the locally added diffusion has an additional advantageous effect that the size of the middle level of refinement is decreased. The solution, influenced by the locally added numerical diffusion is much smoother and the detection method uses this smoothness to decrease the size of the refinement areas.

5.5.2 Safety region

A property that improves the overall convergence rate of the LDC-method comes with the introduction of the safety region, see Minero [51]. The principle behind the safety region is that the total refinement area is larger than the area that is used for defect correction. Errors that are introduced by the interpolation of the boundary conditions or the movement of the refinement area do not immediately influence the correction. The only interaction that the fine grid has with its corresponding coarse grid is through the defect correction and the boundary conditions. If an interpolation error occurs, this error is, without a safety region, almost directly fed back into the defect term that is applied to 'improve' the coarse grid solution. With a safety region of at least one coarse grid control volume, the direct feedback between the interpolated boundary condition and the defect term is removed. To demonstrate the difference between the solution with and without a safety region, a one-dimensional cross-section of the concentration is shown if figure 5.6. For this example it can be concluded that a

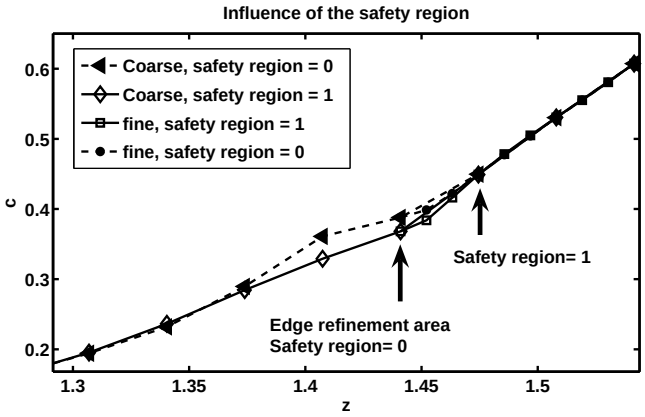


Figure 5.6. A one-dimensional example of the influence of the safety region on the quality of the solution.

less smooth solution is obtained if no safety region is applied. Without safety region, an increase in concentration is observed, caused by the direct interaction between the defect and the boundary interpolation.

Minero [51] concluded that the safety region is a tool that helps to speed up the convergence. In a three-dimensional problem, it is absolutely valuable for the stability and accuracy of the solution.

5.6 The required number of LDC-iterations

In order to make it a useful method, Local Defect Correction needs to be competitive with other solution methods such as a conventional or a spectral approach. The amount of time that is needed to perform one time step depends to a large extent on the number of LDC-iterations that are required. Due to the fact that, in a time-dependent problem, the solution at the previous time step is a good initial guess for the iteration process, the total number of iteration steps that are needed to reach convergence is small. Numerical simulations by Minero [47] have shown that one LDC-iteration is sufficient to reach convergence. If many iteration steps would be needed, this would severely reduce the overall efficiency of LDC.

To test this on a three-dimensional problem, simulations are performed with one and two LDC iterations. Figure 5.7 shows a comparison of the area of two isosurfaces of concentration as a function of time for both simulations. From this figure it can be concluded that the behavior of the surface area is almost identical for both situations. The L_2-norm of the error, $\|\mathrm{LDC}_2 - \mathrm{LDC}_1\|_2 / \|\mathrm{LDC}_1\|_2$, demonstrates that the difference is indeed small. Here, $\mathrm{LDC}_{1,2}$ denotes the solution computed with one or two LDC-iterations. Table 5.1 shows the results of this relative error for the different levels of refinement at three moments in time. Note that after 500 time steps the difference between the two simulations is still small and the continuous increase is

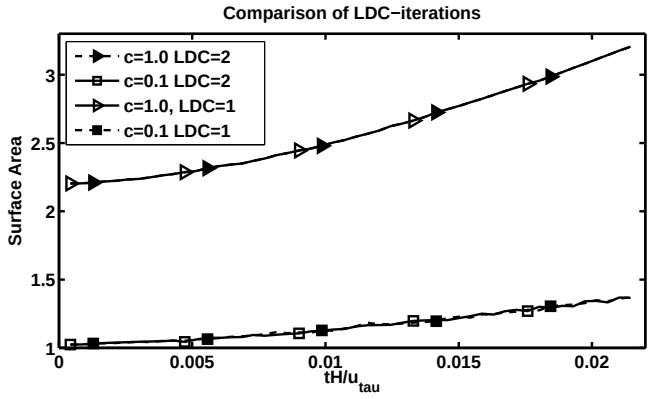


Figure 5.7. Comparison of the area of two isosurfaces of concentration on the middle refinement level between one and two LDC iterations. The solid markers are for one iteration, the open markers represent the results for two iterations.

	$t = 100$	$t = 250$	$t = 500$
level 0	2.6×10^{-12}	1.8×10^{-7}	1.4×10^{-5}
level 1	3.1×10^{-11}	8.9×10^{-7}	2.1×10^{-5}
level 2	1.7×10^{-11}	2.2×10^{-10}	1.2×10^{-5}

Table 5.1. The L_2-norm of the difference between solutions, $\|\mathrm{LDC}_2 - \mathrm{LDC}_1\|_2 / \|\mathrm{LDC}_1\|_2$, with one or two LDC iterations. The values are shown on each level of refinement for different moments in time (denoted in time steps).

due to turbulence and not caused by the LDC iterations. After one time step the difference between the solution with one and two LDC iterations is negligible.

Additional LDC iterations are computationally expensive and this result is very beneficial for the performance of LDC. For example, a specific problem takes takes approximately 50 seconds per time step, when one LDC iteration is applied. Increasing the number of iteration on all levels of refinement to two, 325 seconds are needed. This nonlinear increase in computational time has to do with the fact that the refinement levels are nested.

5.7 Numerical results

5.7.1 Increasing the Schmidt number

When all improvements discussed in the previous section are applied and the number of iterations per LDC time step is set to one, it is possible to perform simulations at an increased Schmidt number. To this purpose, a Gaussian shaped blob of concentration is released, that has a standard deviation $\sigma_x = \sigma_y = 5 \times 10^{-2}$ and $\sigma_z = 5 \times 10^{-1}$. The

grid size is set to approximately 5.5×10^{-3} in wall-normal and spanwise direction, whilst the grid size in streamwise direction is two times as large: 1.1×10^{-2}. Note that these settings are different from section 5.2.3, where this issue was already mentioned. The total number of refinement levels is set to three, with a safety region of one. The Schmidt numbers studied were $Sc = 1, 2, 5$ and 10. For increasing Schmidt numbers the number of grid points should be increased as well, but here it is taken constant on all levels of refinement to enable a better comparison. In figure 5.8 a contour plot of the solution is shown for the four different Schmidt numbers. What can clearly be seen is the decrease in scales: the larger the Schmidt number, the smaller the scales. In the one-dimensional cross-section in streamwise direction, shown in figure 5.9, more details of the concentration become visible. It can also be noted that at Schmidt number larger than or equal to five the solution contains regions of negative concentration.

In figure 5.10 the area of a specific concentration isosurface is plotted for two different values of the concentration as a function of time. Only small differences between the Schmidt numbers are noticeable. In the left figure, the area of a lower concentration isosurface is shown. Here, the surface area increases more rapidly for decreasing Schmidt number. This is due to the influence of molecular and numerical diffusion as the surface area is computed from the middle refinement area, where the local decrease in Schmidt number is applied. In the right figure, the area at a larger value of the concentration is shown that is present in the finest level of refinement. Here, the surface area increases more rapidly for increasing Schmidt number. This can be explained by the appearance of smaller scaled structures. The coarseness of the surface-area profile can be explained by the continuous movement of the refinement area. If the isosurface touches the edge of the refinement area, rapid changes in the total surface area are noticeable when the refinement area moves.

5.7.2 Increasing the number of grid points

For a Schmidt number larger than or equal to five, the previous results demonstrated that regions of negative concentration appear. A negative concentration is not physical and is the result of an insufficient resolution. In the following simulation, the Schmidt number is set to five and the grid size on the finest level is set to approximately 5.5×10^{-3} in each direction. This represents a decrease by a factor of two in streamwise direction with respect to the previous simulation. It is assumed that the resolution in this direction causes the non-physical behavior, since it has the largest velocity component. The decrease in grid spacing is obtained by a change in the refinement topology. Note that multiple options are available to change the refinement topology and reach the desired grid size.

In figure 5.11 contour plots of concentration are shown and in figure 5.12 the concentration as a function of the streamwise coordinate for both resolutions. From both figures it can be concluded that indeed the resolution in streamwise direction is the cause for the appearance of negative concentration. However, the increase in resolution goes at the expense of an increased computational time by a factor of 1.6.

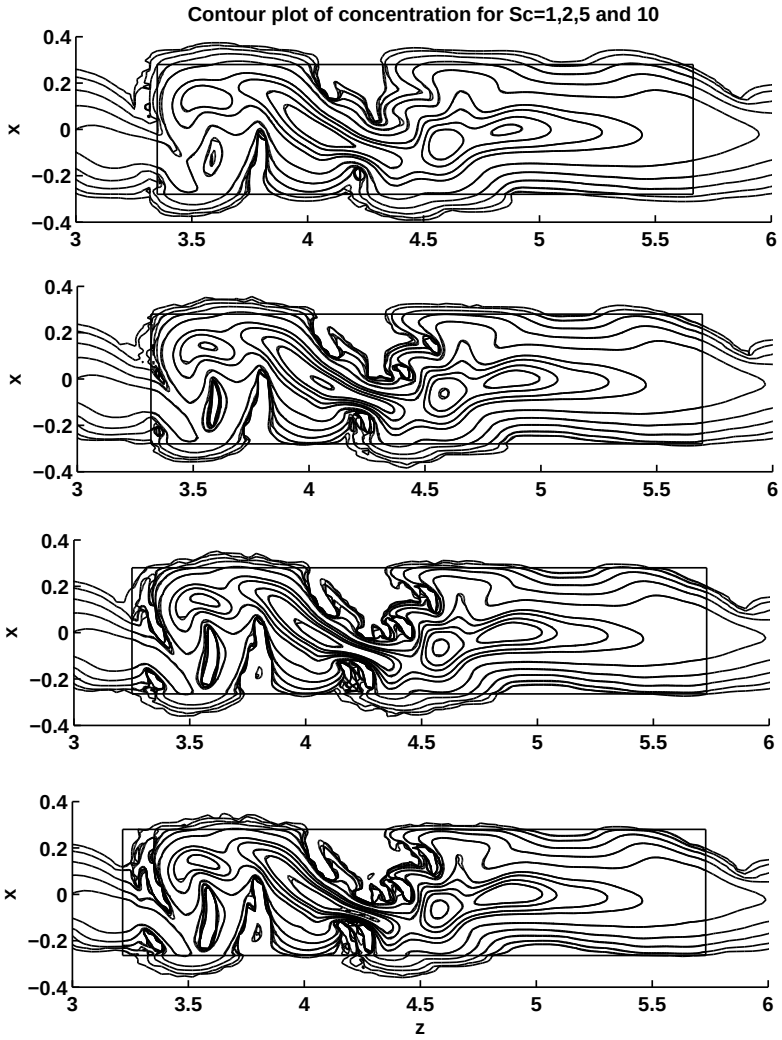


Figure 5.8. Comparison of a contour plot of concentration through the Z-X-plane at the same position in time and space. From top to bottom, the Schmidt number equals $Sc = 1, 2, 5$ and $Sc = 10$. For reference, the position of the finest level is denoted with a rectangle.

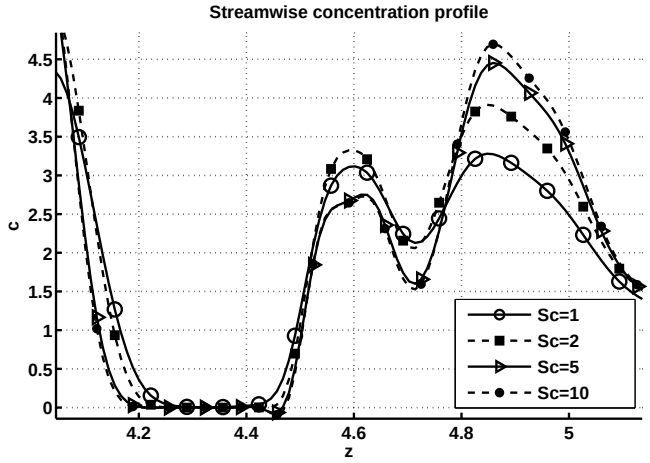
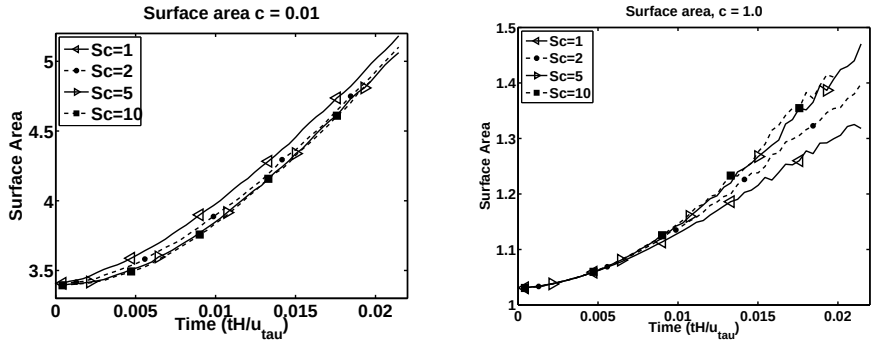


Figure 5.9. A comparison of the concentration as a function of the streamwise coordinate for Schmidt numbers equal to $Sc = 1, 2, 5$ and $Sc = 10$. Note that for clarity not all fine grid points are marked



(a) Surface area at $c = 1 \times 10^{-2}$

(b) Surface area at $c = 1$

Figure 5.10. A comparison of the total surface area for two different values of concentration. On the left the concentration is equal to 1×10^{-2} and in the right figure the concentration equals one. The markers each represent a different Schmidt number, ranging from $Sc = 1, 2, 5$ and $Sc = 10$.

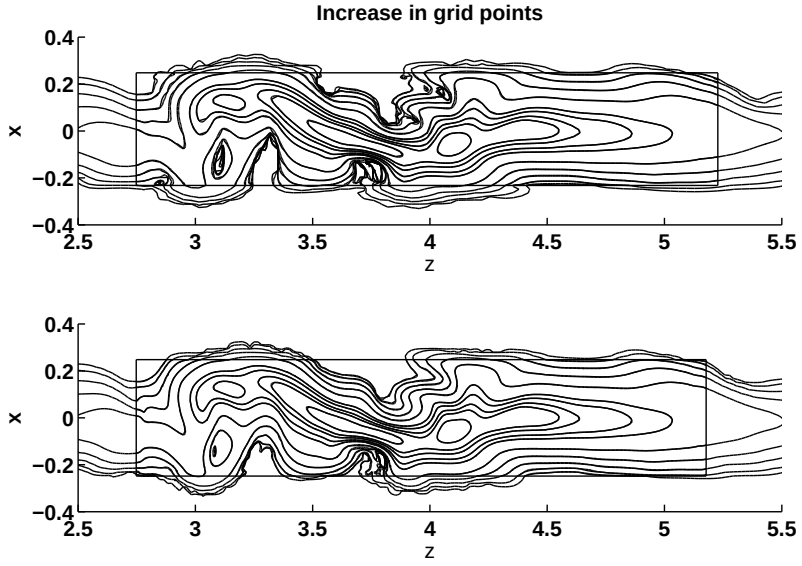
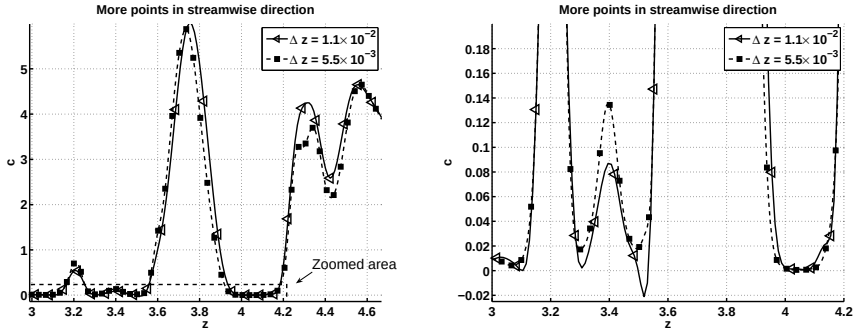


Figure 5.11. A comparison of a contour plot of concentration through the Z-X-plane at the same position in time and space. In the bottom figure more points are used in the streamwise direction. The Schmidt number is in both simulations equal to five.



(a) Overview (b) Zoomed area

Figure 5.12. A comparison of concentration as a function of the streamwise coordinate. The right figure is close-up from the left figure. The Schmidt number is in both simulations equal to five. Note that for clarity not all fine grid points are marked

5.8 Local Uniform Grid Refinement

5.8.1 Explicit correction step

Apart from Local Defect Correction, Local Uniform Grid Refinement (LUGR) is also a method that focuses computational power on areas of high activity, see for example the work done by Trompert [74]-[75]. Both strategies use the higher accuracy of the fine grid solution to improve the coarse global grid solution. The difference between LDC and LUGR is that the former applies an implicit defect correction, whereas LUGR uses an explicit approach. Implicit defect correction improves the global coarse grid solution by the information obtained from the refinement area. With an explicit method, the defect only influences the coarse grid points at the same location of the refinement area. This means that the coarse grid points, outside of the refinement area that are needed to interpolate the boundary conditions, are not updated by LUGR. The advantage of an explicit defect correction is that it is less expensive, especially in a three-dimensional problem. Due to the fact that the time step is limited by accuracy requirements, the time step used in the implicit method cannot be increased.

In a time-dependent problem, with a small time step and a moving refinement area, the implicit treatment of the defect term is both wanted and unwanted. As the position of the refinement area is continuously adjusted, the interpolation of new fine grid concentration requires a solution that is as accurate as possible in order to prevent interpolation errors. Using implicit defect correction, the coarse grid points that are needed for interpolation in future positions of the refinement area are included in the correction. On the other hand, the solution inside the refinement area can hardly have any significant effect on the coarse grid solution near the edges of the coarse domain. In the following the solutions calculated with LDC will be compared with results of LUGR.

5.8.2 Numerical comparison

In the test case designed to compare LUGR with LDC, the initial condition for both simulations is a Gaussian shaped blob of concentration with a standard deviation of $\sigma_x = \sigma_y = 5 \times 10^{-2}$ and $\sigma_z = 2.5 \times 10^{-1}$. Three levels of refinement are applied and the refinement topology is constructed in such a way that the finest grid size is equal to 5×10^{-3} in all three directions. The time step on the finest grid equals $5 \times 10^{-3}/8$.

A global comparison in the form of a contour plot in the Z-X plane through the center of the blob after 500 time steps is given in figure 5.13. Small differences can be detected in the size of the refinement area and in the concentration near edges of the refinement area, whereas the solutions in the center of the finest level are indistinguishable. An indication of the difference between both solutions is provided by the area of concentration isosurfaces as a function of time, plotted in figure 5.14. The differences in surface area between LUGR and LDC are again small and are a combination of difference in the shape and size of the refinement area and the actual difference between LUGR and LDC. The latter is found in the concentration profiles at a lower value of the concentration, determined form the middle level of refinement. Here, the difference in the application of the defect term leads to a small difference in

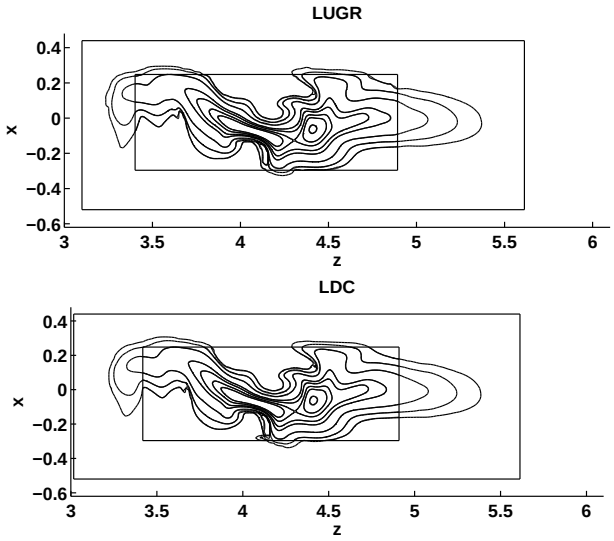


Figure 5.13. Contour plots of concentration after 500 time steps, made at the same location. The upper figure shows the solution for LUGR and the lower figure for LDC.

the obtained surface profile. In both contour plots and surface area, no significant differences can be found. In figure 5.15 a one-dimensional cross-section is shown for both LDC and LUGR that is meant to visualize the existing small differences. Figure 5.15 shows the cross-section through the center of the Z-X plane, where a small difference between LDC and LUGR is found near the edge of the refinement area. Near the edge of the refinement area, see figure 5.16, the difference between LUGR and LDC becomes more clear. In this figure, the additional global correction of LDC ensures that the solution remains smooth and positive, which is not the case for LUGR.

5.8.3 Accuracy analysis

To give an estimate of the accuracy of LUGR and LDC a more accurate approximation of the solution is required, as the exact solution is not known, see section 3.7.4. Several simulations are performed with a decreased grid size in stream wise direction. All other parameters are identical to the ones in section 5.8.2. The choice to decrease the grid size in the stream wise direction is based on the fact that this velocity component is largest and numerical instabilities are most likely to occur in that direction. The grid size in stream wise direction was decreased by approximately a factor of two and three by changing the refinement topology. The time step was decreased accordingly and a total of six simulations are performed.

In order compute the L_2-norm of the error, the coordinates of each computational grid must be identical. Because the decrease in grid size is obtained by altering

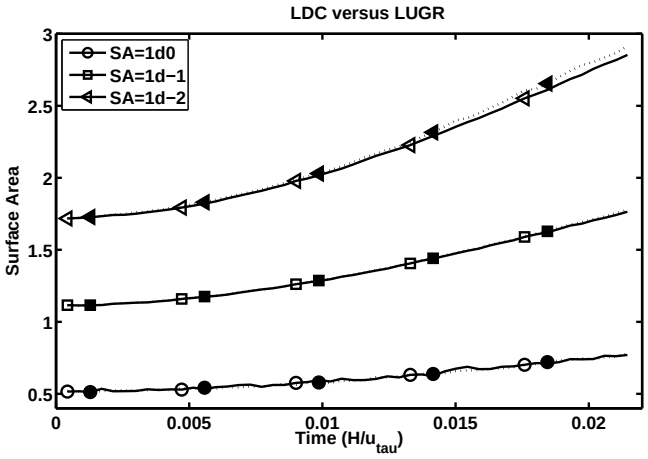


Figure 5.14. Comparison of the area of concentration isosurfaces between LUGR and LDC. The dashed lines are for LDC, whereas the solid lines correspond to LUGR. Different markers are used to identify different values of the concentration at which the surface is computed.

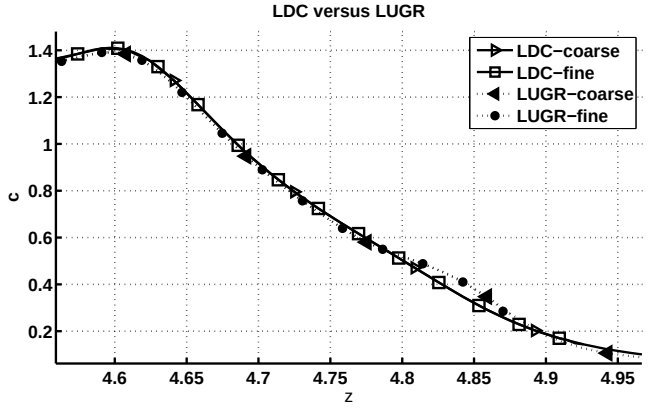


Figure 5.15. A one-dimensional cross-section of the concentration field that shows the difference between the solutions obtained from LUGR and LDC through the center of the blob of concentration. Note that for clarity not all fine grid points are marked.

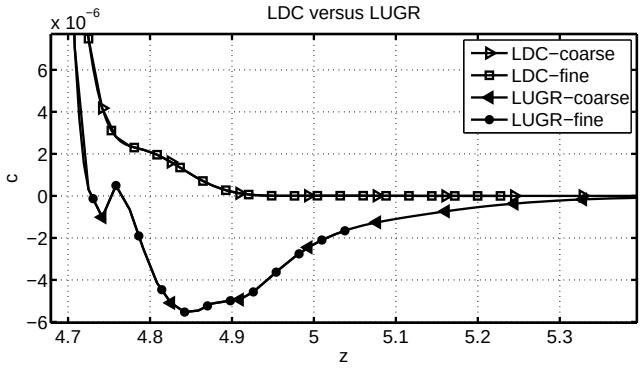


Figure 5.16. A one-dimensional cross-section of the concentration field that shows the difference between the solutions obtained from LUGR and LDC near the edge of the refinement area. Note that for clarity not all fine grid points are marked.

	L_2-norm	$t = 200$	$t = 300$	$t = 400$	$t = 500$
LDC	$(\Delta_z - \frac{1}{3}\Delta_z)$	3.7×10^{-3}	6.8×10^{-3}	1.2×10^{-2}	1.9×10^{-2}
LDC	$(\frac{1}{2}\Delta_z - \frac{1}{3}\Delta_z)$	2.0×10^{-3}	2.2×10^{-3}	2.6×10^{-3}	3.1×10^{-3}
LUGR	$(\Delta_z - \frac{1}{3}\Delta_z)$	6.4×10^{-3}	8.7×10^{-3}	1.3×10^{-2}	2.0×10^{-2}
LUGR	$(\frac{1}{2}\Delta_z - \frac{1}{3}\Delta_z)$	5.8×10^{-3}	6.2×10^{-3}	6.5×10^{-3}	7.1×10^{-3}
Δ_z	(LUGR-LDC)	4.6×10^{-3}	6.4×10^{-3}	9.1×10^{-3}	1.0×10^{-2}
$\frac{1}{2}\Delta_z$	(LUGR-LDC)	3.8×10^{-3}	5.1×10^{-3}	7.2×10^{-3}	7.6×10^{-3}
$\frac{1}{3}\Delta_z$	(LUGR-LDC)	4.8×10^{-3}	5.9×10^{-3}	8.5×10^{-3}	9.7×10^{-3}

Table 5.2. A comparison of the relative errors for LDC, LUGR and the comparison of LUGR with LDC at different moments in time (in time steps) for different values of the streamwise grid size. In this table, $a - b$ is an abbreviation for $\|a - b\|_2/\|b\|_2$.

the refinement topology, the grid points and position of the refinement area of each simulation are different. To be able to compare the concentration fields, they are interpolated in space using linear interpolation. Each concentration field is interpolated to the grid points of the finest grid available. Note that the interpolation introduces errors, but these are assumed to be smaller than the differences between two solutions. In table 5.2 the relative errors are shown and it can be concluded that with a decrease in grid size the difference between the solutions decreases, both for LDC and LUGR. However, LDC converges faster when the grid size is reduced than LUGR. The differences between LDC and LUGR are small too and this corroborates the previous results. Note that the results shown in table 5.2 are only an indication of the accuracy of LDC and LUGR, as the exact solution is not known.

5.9 Applicability of LDC

5.9.1 Requirements for a successful application

Not only accuracy, but also efficiency is an important issue for numerical methods. If a numerical method is not efficient, the required computational times rises and if the gain in accuracy does not outweigh the increase in computational effort, the method is not applied. Therefore, it is important for Local Defect Correction to be at least nearly as efficient as other solution strategies. Before a comparison is possible, it is useful to treat several points that have a large influence on the efficiency of LDC. Normally, more grid points and a more complex, higher order, method require more computational effort. For LDC, several other aspects influence required computational time as well. A list of important factors is given below:

- The size of the high-activity region.
- The required grid size.
- The total number of refinement levels.
- The number of LDC-iterations to reach convergence.
- The size of the time step.
- The order and complexity of the applied numerical method.

The first five criteria are all directly related and can be summarized into the following statement: *If the size of the high activity is and remains small compared to the total computational domain, the application of Local Defect Correction to a three-dimensional time-dependent problem can be successful.* A more quantitative statement cannot be given, since it depends on the properties of the computer, such as available memory and optimization strategy.

5.9.2 Efficiency with respect to other strategies

LDC versus the conventional approach

In chapter 3, the evolution of the passive scalar was computed without any form of local grid refinement. Compared to the conventional approach, the computational time of LDC is decreased by a factor of eight. In this comparison, the grid size and time step of the finest level of refinement are equal to those used in the conventional approach. Since LDC becomes unstable on the global grid, only a few time steps that give an indication of the required computational time have been performed.

LDC versus LUGR

The differences in accuracy (and mathematical approach) between LUGR and LDC are small. The computational time that is needed to compute one time step for the concentration is compared in table 5.3. In this table the computational time in the

	3-levels		4-levels	
	LDC	LUGR	LDC	LUGR
t_{begin}	40s	22s	315s	205s
t_{end}	95s	58s	-	-

Table 5.3. Comparison of the computational times that are needed to perform one time step for LUGR and LDC. The computational time is measured during the first few time steps and last few time steps of the simulation. The computational time at t_{end} is not shown for the 4-level example as that moment in time was not reached.

initial stages and at the end of the simulation is shown. As a test case, an additional level of refinement was added, but the computational time was increasing rapidly and the simulation was stopped before the end was reached. It can be concluded that the difference in required computational time is not small and much in favor of LUGR. With an increasing number of grid points or refinement levels, the additional costs for the implicit solver cause an increasing difference in the required computational time between LUGR and LDC. Note that in LUGR, the last re-computation (step 9) of the fine grid solution is also performed, despite the fact that the coarse grid solution is not updated. If this re-computation step is removed, the computational time for LUGR decreases approximately by a factor of two. In section 5.8 it was shown that LDC produces a more accurate solution than LUGR, but if this is worth the additional computational effort, depends on the required accuracy. It is questionable whether LDC remains the more accurate method if both methods use the same amount of computational time per time step.

5.10 Conclusion

In this chapter, Local Defect Correction was introduced and applied to a three-dimensional time-dependent problem. Due to the large increase in number of grid points when moving from two towards three dimensions, a locally decreased Schmidt number and a safety region are required to maintain a stable solution. With these additions, only one LDC-iteration per time step is required. Additional iteration steps hardly improve the accuracy, but require significantly more computational time. This can be explained by a combination of the fact that the initial guess at the beginning of each time step is already very accurate and that the time step is small.

Simulations showed that LDC is capable of computing the evolution of a passive tracer in a three-dimensional turbulent channel flow, with a Schmidt number larger than one. Without the availability of an exact solution to the problem, the accuracy of the LDC result cannot be assessed quantitatively. However, comparison with an approximate exact solution, obtained by decreasing the grid size in streamwise direction, showed that the relative error is small. In a direct comparison of LDC with LUGR, it was shown that LDC results in a more accurate solution, but requires more computational effort. The choice between one of these methods depends on the desired accuracy and available computational resources.

Chapter 6

Modified convection problems

6.1 Introduction

In the previous chapters, methods, advantages, disadvantages and requirements that allow a successful application of Local Defect Correction to a convection-diffusion equation for the concentration of a passive tracer that is released in turbulent channel flow, were discussed. The purpose of this chapter is to investigate different situations, which have a large similarity with the previous problem, but have a small addition to the original set of equations. With this addition, the properties of the solution change and it is investigated if Local Defect Correction is capable of handling these properties. To make the chapter more interesting from a physical point of view, the governing equations are introduced through simplified physical problems. In section 6.2 sedimentation is used to investigate the behavior of LDC when the velocity component is slightly altered, creating a sinking blob of concentration. If the sedimentation speed depends on the value of the concentration, the behavior of LDC for increasing gradients can be investigated. Population growth, see section 6.3, allows the investigation of the behavior of an increasing amount of concentration. To balance the source term, predators are released into the flow that feed on the plankton. In the last section (6.4) of this chapter a combustion problem with a traveling flamefront will be discussed. This problem can only be refined in streamwise direction and has a continuously increasing grid size in combination with increasing gradients. Note that the goal of this chapter in not to describe detailed physics, but to check the behavior of LDC under certain conditions.

6.2 Sedimentation

Particles that are released into a turbulent flow are assumed to be passive tracers that are without volume and mass. This is an idealized situation and can almost never be observed in real life. Particles with a finite volume and mass are affected by gravity and, depending on the difference in density, sink or rise. This phenomenon is called sedimentation and the speed at which the particles sink can be determined by

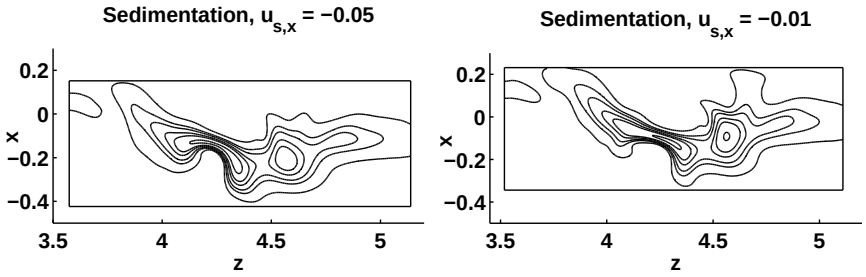


Figure 6.1. Two contour plots of concentration at the same moment in time, after 500 time steps, using different values of the wall-normal sedimentation speed. The left figure shows the solution using a sedimentation speed of $u_{s,x} = -0.05$ and the right figure $u_{s,x} = -0.01$.

analyzing a single particle. Following the approach as described by Maxey and Riley [44] and Necker [59], the particles that are inserted into the flow are assumed to be small, spherical and their motion is dominated by convective rather than volumetric forces. Consider the fact that the density of a particle is much larger than that of the fluid, $\rho_p \gg \rho_f$. If the inertia of the particle is negligible, the particle velocity equals the sum of the local fluid velocity and the sedimentation velocity,

$$\vec{u}_p = \vec{u}_f + \vec{u}_s, \tag{6.1}$$

where $\vec{u}_s$ denotes the sedimentation velocity, which follows from the balance of gravity, buoyancy and drag. If all particles are assumed to be equal in shape and size and no clustering of particles occurs, the sedimentation velocity is a constant. Combining this property with equation (3.3), the influence of gravity on the concentration field of a passive scalar can be modeled by the following expression,

$$\frac{\partial c}{\partial t} + (\vec{u}_f + \vec{u}_s) \cdot \nabla c = \mathcal{D}\Delta c + S. \tag{6.2}$$

6.2.1 Numerical simulation

Simulations are performed using a Gaussian shaped blob of concentration as an initial condition and two different situations are investigated. The standard deviation of the blob of concentration is set to $\sigma_x = \sigma_y = 5\times10^{-2}$ and $\sigma_z = 2.5\times10^{-1}$, with a Schmidt number of five. On the finest level, the grid size is in all direction equal to 5.5×10^{-3} and the time step is set to $5 \times 10^{-3}/8$. In the first situation gravity acts in wall-normal direction, $\vec{u}_s = (-0.01, 0, 0)$ and $\vec{u}_s = (-0.05, 0, 0)$, and additional movement of the refinement area is needed to follow the high activity region. Figure 6.1 shows a contour plot of concentration in the Z-X-plane of the two different wall-normal sedimentation velocities. The influence of the additional gravitational force on the

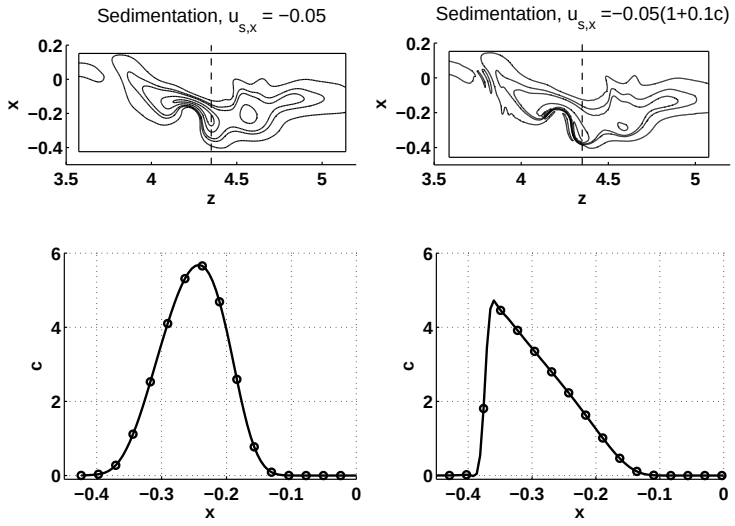


Figure 6.2. Comparison of sedimentation with and without concentration-dependent sedimentation velocity. On the left, solutions are plotted for constant sedimentation velocity and on the right for concentration-dependent sedimentation velocity. The top figures show a contour plot of concentration through the Z-X plane after 500 time steps on the finest grid. The bottom figures show the concentration as a function of the wall-normal coordinate. The dashed vertical line marks the position at which the cross-section is made.

concentration field can clearly be noted. In the second situation, gravity works in the direction opposite to the mean flow and the sedimentation speed is set to be equal to the mean velocity component near the center of the channel, $\vec{u}_s = (0, 0, -0.75)$. In this situation the blob of concentration stays at its initial position and less movement of the refinement area is required. Small perturbations in the shape of the concentration are noticeable, but the position is almost constant.

From the results of both simulations can be concluded that the sinking blob of concentration can be followed without much additional effort in combination with LDC. The only situation that can become problematic is when the concentration reaches the wall. At the wall, the concentration gradient in wall-normal direction is set to zero, but simulations are stopped before this situation is reached. For more information about the collision of a dipole with a wall in combination with tracer transport, the author refer to work done by Kramer [34].

6.2.2 Particle clustering

In the previous section, the concentration field was influenced by a constant sedimentation speed. Consider the situation that with increasing concentration the sedimentation velocity increases accordingly. Physically this is analogous to drag reduction by

particle clustering at large concentration. To model this behavior, the sedimentation velocity becomes a function of the concentration and is given by,

$$\vec{u}_s = \vec{u}_s^0\big(1 + \alpha_s c\big), \tag{6.3}$$

with $\vec{u}_s^0$ the sedimentation velocity for infinitely small concentration and α_s is a tuning parameter. A numerical simulation, identical to the one without clustering, is performed using $\vec{u}_s^0 = (-0.05, 0, 0)$ and $\alpha_s = 0.1$.

In figure 6.2 the solution is shown for a simulation with and without concentration-dependent sedimentation velocity. If $\alpha_s > 0$, the higher concentration region around the center of the concentration blob sinks faster than the region near the edges of the refinement area. In the direction of gravity, the concentration gradients increase, as can be seen from the lower plots in figure 6.2 and this causes the instabilities that can be observed in the contour plot. The situation where the gradient of the solution increases, is problematic for LDC, because a continuously decreasing grid size is needed to prevent the solution from becoming unstable. A solution would be to add an additional level of refinement, so that the steeper gradient can be better captured. A test-simulation showed that this only shifts the problem to later times and increases the required computational time, per time step, significantly.

6.3 Population growth

The prediction of the growth of a population of a certain species can be used in many different areas. These range from calculating necessary food supplies to incubation times for bacterial colonies. In the 18^{th} century, the first models for population growth were derived for bacteria and were based on exponential growth [57]. Later, in 1838, the model was extended by Verhulst [57], who included a mortality rate next to the growth rate of a population. This extension removed the undesirable effect of unlimited growth and this model is known as the logistic growth or Malthus-Verhulst equation and is given by,

$$\frac{dp}{dt} = \alpha_g p - \beta_g p^2, \tag{6.4}$$

where p is the population density, and α_g and β_g are the growth rate and mortality of a population, respectively. The logistic equation can be used in combination with turbulent flow simulation in the modeling of dispersion of a population, for instance plankton, in the ocean. Plankton is the name for a large diversity of micro-organisms that are so small that they are transported along with the (turbulent) oceanic current without having any from of significant self-induced motion. This allows plankton to be modeled as a passive tracer. Because the plankton rich areas are small compared to the size of the ocean, LDC can be used to efficiently compute the evolution of the population. To model logistic growth, a source term (S_g) is added to equation (3.3) and becomes,

$$S_g = \alpha_g c(1 - \frac{\beta_g}{\alpha_g} c). \tag{6.5}$$

If the dependent variable c is replaced by $c\gamma_g$, where $\gamma_g = \frac{\beta_g}{\alpha_g}$, the ratio between the growth and mortality rate can be removed from equation (6.5). After this scaling

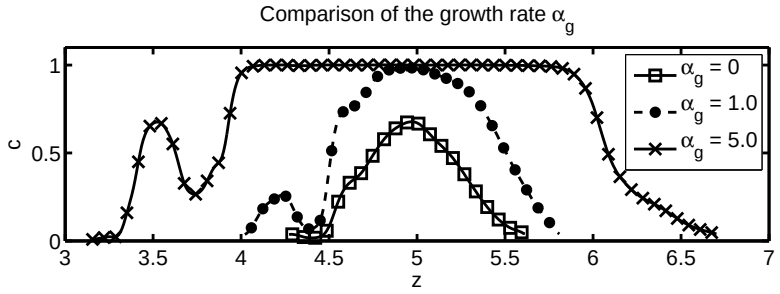


Figure 6.3. Concentration as a function of streamwise coordinate for three different values of the growth rate after 500 time steps. The Schmidt number is equal to one. Note that for clarity not all fine grid points are marked.

$c = 1$ is a stable solution of the logistic equation. The rate at which the concentration grows or decreases towards this stable solution can be influenced by the growth rate. The simulations are performed with the same grid size and time step that was for the sedimentation simulation, but the Gaussian initial concentration field has a maximum equal to two. In figure 6.3 the solution at a later time is compared for three different values of the growth rate as a function of the streamwise coordinate. The result is not very surprising as the solution with the largest growth rate reaches its stable concentration faster and this concentration is hence spread out over a much larger area. However, for the resulting larger refinement area, the efficiency of LDC decreases rapidly. It can be concluded that LDC can be applied to convection-diffusion equations with a source term, but in order to keep the area of the refinement region limited, the magnitude of the source term should be small compared to the magnitude of the convective term. This specific problem could be computed more efficiently if the refinement area is allowed to have an arbitrary shape, so that a circular refinement region is possible.

6.3.1 Predators

As was already shown, the plankton population is allowed to cover larger and larger regions over time. Although this sometimes happens in reality, usually this is prevented by the presence of another species that feeds on the plankton. More information about the interaction between predator and prey in such situations can be found in the work by Osborn [63] and Boffetta [11], who investigated detailed predator-prey models, but that is beyond the scope of this thesis. Here the predators are modeled as discrete particle which are tracked in a Lagrangian way as described in chapter 3. The resulting predator particle-concentration, which is calculated as in equation (3.33), enters the convection-diffusion equation for the plankton in the form of a neg-

ative source term. To be able to scale the removal of plankton by the predators, the source term becomes,

$$S_{pr} = \vartheta_p c_p \tag{6.6}$$

where c_p is the predator concentration, ϑ_p is the feeding parameter, set to $\vartheta_p = -0.01$. Note that the value of ϑ_p is important to balance the removal of plankton. If set too high, al plankton is instantly removed and when set too low, the influence of the particles cannot be noticed. If this simplified model is to be successful, the predators cannot eat more plankton than is locally available and the total number of predators is fixed.

If the predators are very small, they are assumed to be passive tracers, just like the plankton. This means that the feeding only depends on the diffusion of plankton towards the location of the predator, which is a very slow process. In the test-simulations preformed, the local food supply was depleted rapidly and after a while there was not much to eat anymore in the direct area around the particle.

Larger predator species cannot be regarded as passive tracers, because they have the capability of self-induced motion that allows them to actively hunt for food. Although there is no physical basis for this, the predators are modeled to have a feeding velocity, with a magnitude dependent on the local plankton concentration and directed towards larger plankton concentrations. Moreover, a random velocity component is added in order to prevent large clusters of predators. In the results shown thousand predators are released, using the same distribution as the concentration field. In figure 6.4 several snapshots of the concentration field and predator locations are shown. It can be seen that initially the predators move fast towards the region of largest plankton concentration. Later in time this region is depleted of plankton and the predators spread out to the edges of the initial region of high plankton concentration in search of food. During this stage the concentration in the center has time to recover. In order to do so, the predators are allowed to have a velocity that is somewhat larger than the maximum fluid velocity in each direction.

The inserted particles acts as a sink term that helps to controls the size of the refinement area. The non-physical simulation with the active predators is very difficult to perform without any form of grid refinement as a very rough concentration profile is created when the concentration is locally depleted.

6.4 Flamefront

Combustion is a subject that was earlier used to demonstrate that LDC is a valuable grid refinement method. Anthonissen [1]-[3], for example, studied a Bunsen flame, which is a laminar and steady burning problem that can be solved in two dimensions because of symmetry. Near the flamefront large gradients are present and these can be captured very well with LDC. To see if a more complex combustion problem can be treated, a simplified model of a flamefront is implemented that travels in opposite direction of the mean flow. During the initialization, the flamefront is located in a very thin region and a very fine, local grid is required to be able to capture all details.

The combustion model assumes that temperature differences do not cause large differences in density and the flow remains incompressible. Furthermore, all chemical

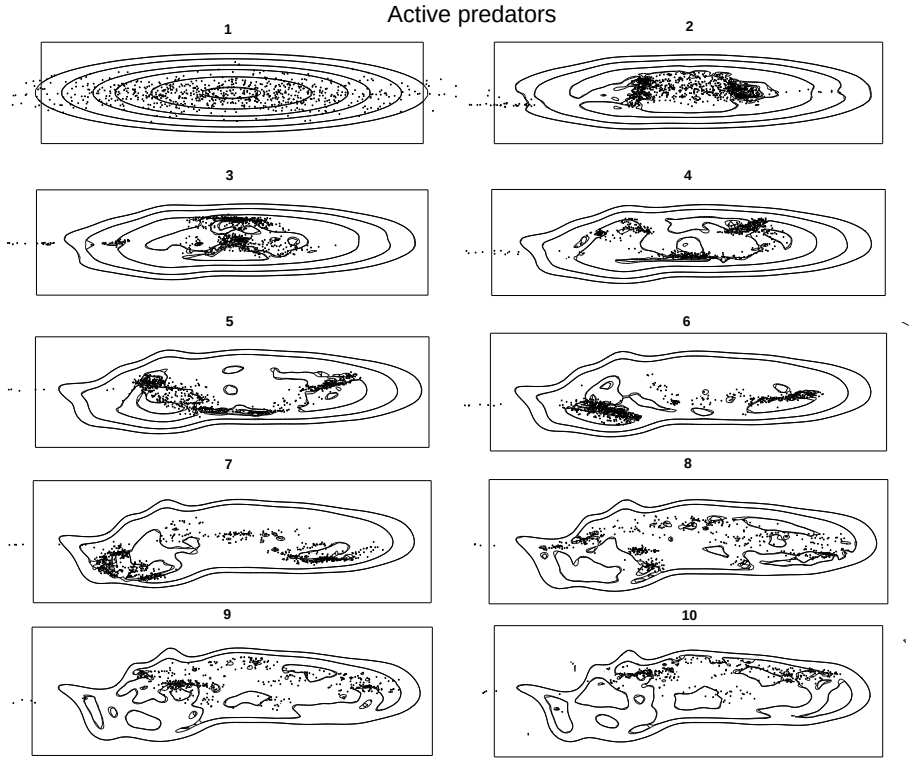


Figure 6.4. Contour plots of concentration in the Z-X plane, with individual predators indicated by solid markers. The snapshots are in sequence of time with time interval equal to 50 time steps.

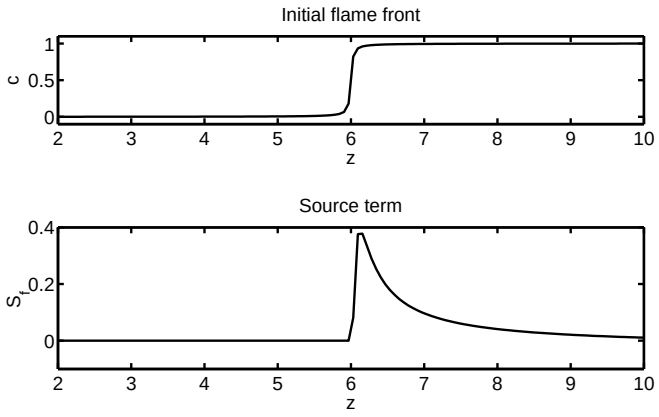


Figure 6.5. A one-dimensional representation of the initial flamefront and the corresponding source term.

reactions are considered as infinitely fast and only one progress variable is modeled that can be described by the convection-diffusion equation. More information about the modeling of combustion problems can be found in De Goey and Ten Thije Boonkkamp [19] and Van Oijen [61]-[62]. The source term that accounts for combustion is modeled by,

$$S_f = A_f\Big((1-c)e^{\frac{-\beta_f(1-c)}{1-\alpha_f(1-c)}}\Big), \tag{6.7}$$

where α_f and β_f are parameters that control the reaction rate and c is the progress variable. Both parameters represent the rate at which heat is released and are different for each type of fuel. For a stoichiometric methane flame $\alpha_f = 0.85$ and $\beta_f = 18$. A_f is a scaling parameter that can be used to influence the speed of the flamefront and is set to twenty. The initial condition of the flame depends on the streamwise coordinate only and is set equal to:

$$c^0(x,y,z) = \frac{(\arctan 50(z-z_f)+1) - (\arctan 50(-z_f)+1)}{(\arctan 50(L-z_f)+1) - (\arctan 50(-z_f)+1)} \tag{6.8}$$

where z_f is the offset where the initial flamefront is positioned. Since the initial condition depends only on one coordinate, the grid will only be refined in that direction. Figure 6.5 shows the one-dimensional representation of the initial flamefront, together with the corresponding source term.

6.4.1 Numerical simulation

Simulations are performed with three levels of refinement, using the same settings as in the two previous sections. The source term of the progress variable leads to a continuous increase of the gradients in the progress variable. From section 6.2.2 it

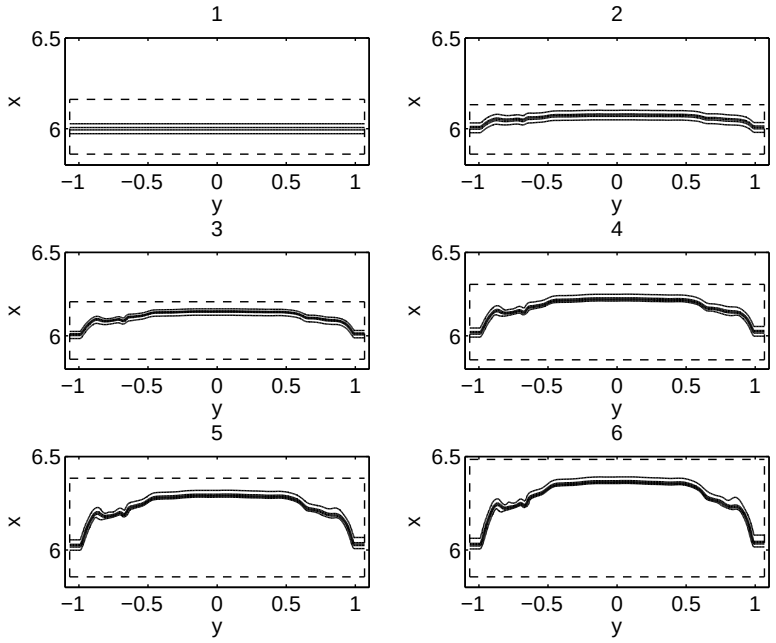


Figure 6.6. A top view of the flamefront, using contour lines of the progress variable at time steps, t = 0, 20, 40, 60, 80 and 100. The dashes lines represent the bounding box of the finest level of refinement. The axis and contour lines are kept the same in all contour plots.

is already known that LDC has difficulties handling such problems. The simulation results show that oscillations start to occur at the lower part of the flamefront and subsequently, the progress variable becomes negative. Increasing the resolution of the refinement area or adding an additional level of refinement only shifts the problem to later times, but does not solve the problem.

In figure 6.6, using contour lines of concentration, a top view of the flamefront is shown at subsequent moments in time. Apart from the flamefront being stretched, a more important conclusion is that the size of the finest level of refinement is continuously increasing. This is not beneficial for LDC, as was concluded in section 6.3 for a different problem. Here, together with the fact that there is no reduction of grid points in wall-normal and spanwise direction, this is problematic as the computational time per time step increases very rapidly. If the propagation speed of the flamefront is set to match the mean streamwise velocity component, the problem is shifted from the center, towards the near-wall regions.

6.5 Conclusion

From this chapter it can be concluded that LDC is very sensitive to small changes in the properties of the passive tracer in a turbulent flow. If only the velocity component is altered, no real problems are encountered. This is expected as this only has impact on the movement of the refinement area. Interpolation of new fine grid concentration occurs more often, but when done properly, this does not cause much additional difficulties. If through adaptations in the velocity vector, the refinement area hovers around its initial position, the total number of grid adjustments reduces and this is beneficial as each interpolation can introduce inaccuracies and takes additional computational effort.

When the grid size of the refinement area is continuously increasing, the efficiency gain of LDC over a conventional approach is threatened when the same number of grid points is applied. LDC requires more effort per time step and if it is to be successful, the refinement area must remain small. There are various reason why the refinement area is continuously growing in size, but turbulent stretching and a source term that is rapidly increasing the total mass of concentration can be become problematic over time. A hard requirement is difficult to formulate, but if the volume of the refinement area doubles in only a fraction of the total required time steps, the application of LDC is not very useful.

Another difficulty is encountered when the gradients that are present within the solution are continuously increasing as was found in the flamefront and the clustering of particles. The steepening gradients require a continuous decrease in grid size. Whilst LDC can locally provide a very high resolution, it only takes time before the resolution needs to be increased again to prevent the solution from becoming unstable. In these situations, LDC can be used to postpone the occurrence of instabilities, but not remove the problem.

One last property that is not beneficial for the application of LDC, is when the grid cannot be refinement in one or more of the available dimensions. Grid refinement methods fully rely on a large reduction of grid point in each available dimension. In combination with one of the above disadvantageous situations the application of LDC is very unsuccessful. One should keep in mind that all investigations of this chapter are not only applicable to LDC, but to other grid refinement methods as well.

Chapter 7

Conclusions and recommendations

7.1 Conclusions

In the previous chapters the numerical methods were discussed that allow a grid refinement method called Local Defect Correction to be applied to a three-dimensional time-dependent problem: the evolution of a passive tracer in turbulent channel flow. One of the reasons for this application is to determine the coefficients in a Langevin equation that is used to model turbulence in a consistent stochastic way [10]. However, to minimize the influence of natural diffusion, a high Schmidt number is required [65]. This causes small-scale structures to appear, which require a very fine computational grid. The currently available computational resources do not allow for such a fine grid. In this thesis an attempt is made to circumvent that problem by applying a method called Local Defect Correction.

Flow solver

Before the passive tracer can be added to a turbulent channel flow, this flow must be computed and this is done by means of a Direct Numerical Simulation. The solver is based on earlier work done by Kuerten [38] and Veenman [77], but was adapted towards a channel flow. The solver is a combination of a Fourier-Galerkin spectral method in span- and streamwise direction and a Chebyshev expansion in wall-normal direction. Time integration is performed using a second-order accurate method, based on the work by Kleiser and Schumann [30]. By a comparison with results from literature [54], it was shown that the results that are produced by the solver are correct and can be used to compute the development of the passive tracer.

Convection-diffusion equation

The Chebyshev expansion used in the flow solver, ensures that many grid points are present in the near wall region. The passive tracer, however, is released as a small

Gaussian shaped blob of concentration near the center of the channel. The Chebyshev point distribution is not suitable to be used to compute the solution to the convection-diffusion equation. Because the tracer is fully passive, it is possible to apply different numerical methods for the tracer- and velocity field and a finite volume method is implemented to compute the concentration of a passive tracer. If the Schmidt number is increased, small-scaled structures appear within the concentration field, which require a very fine grid. An efficient approach is to use a grid refinement method that concentrates the available computational power to those locations where it is really required.

When a different computational grid is used to solve the convection-diffusion equation, the velocity field needs to be interpolated in space. Interpolation is performed with a fourth-order accurate interpolation method, [4], [82]. Direct summation from spectral space is possible, but too expensive in practice. Time integration is performed using an Euler forward method. Simulations have shown that, with the future application of LDC, a third order upwind method gives an optimal combination of efficiency, accuracy and stability. Without any form of local grid refinement the computation of a passive tracer in a turbulent flow is limited to low Schmidt numbers.

Detection and movement of the refinement area

One of the largest problems of a grid refinement method in a time-dependent problem is the continuous movement of the high-activity region. Small adaptations to an existing approach from Bennet and Smooke [7]-[8] were needed to create a detection method that is able to accurately follow the location of the refinement area. When the refinement area is adjusted into areas where no fine grid solution is available, an interpolation method based on a one-dimensional exponential fit was developed to ensure a smooth and positive solution.

Local Defect Correction

One specific grid refinement method is called Local Defect Correction. This method was first developed by Hackbush [20] and later adapted to be used for a finite volume approach by Anthonissen [1]-[3], but was only applied to two-dimensional steady problems. Only recently, Minero [48]-[51] adapted the finite volume LDC-method to time-dependent problems. In this thesis the step from two towards three dimensions was made. In a three-dimensional problem, the large required number of grid points becomes problematic. In a turbulent flow, the initial small blob of concentration is stretched and folded by turbulence and the size of the required refinement area increases rapidly. One of the requirements for LDC is that the coarse grid solution must be reasonably smooth. This requirement is not always possible to fulfill and instabilities are likely to occur. A solution to this problem is found by a temporary local decrease of the Schmidt number. Numerical simulations show that it is possible to compute the evolution of a passive tracer in a turbulent flow at an increased Schmidt number, using LDC. However, a comparison with a conventional approach has not been possible due to the too large computational requirements. For a two-dimensional problem such a comparison has been made by Kramer [34] and Minero [51]. Without

the availability of an exact solution to the problem, the accuracy of the LDC result cannot be assessed quantitatively. However, comparison with an approximate exact solution, obtained by decreasing the grid size in streamwise direction, showed that the relative error is small.

Several restrictions apply to the applicability of LDC to a convection-diffusion equation. These restrictions can be summarized by stating that the size of the refinement area must remain small with respect to the total computational domain. Local Uniform Grid Refinement is an alternative to LDC that is almost similar, with the difference that an explicit instead of an implicit correction can be used. Simulations have shown that differences in solution between LUGR and LDC are small. LDC is more accurate than LUGR, but the latter requires much less computational effort.

Applicability of LDC

If the convection-diffusion equation is slightly changed, using a different source term or a different convection velocity, the properties of the problem change. This change in properties is used to investigate the behavior of LDC in these different situations. It can be concluded that a rapidly increasing grid size, due to a positive source term or irregular shape of the solution is problematic. If the size of the refinement area is controlled by, for example a negative source term, the application of LDC is much more successful. Furthermore, if only one direction can be refined due to the physical properties of the problem, the application of LDC is unsuccessful.

7.2 Recommendations

For future investigations into the application of grid refinement methods, such as LDC or LUGR, to a three-dimensional time-dependent problem, several recommendations are given below. Most recommendations are not directly related to LDC, but towards minimizing the number of grid points by improving the shape of the refinement area and refinement topology.

Arbitrary refinement area

One of the major bottlenecks of grid refinement in time-dependent problems is the irregular shape of the solution. The currently implemented detection method only allows for a rectangular shaped area of refinement and this is, in almost every situation, not the most optimal form. If the refinement area is allowed to have a more arbitrary shape, the efficiency of the grid refinement method increases significantly. A good example is a rotated, rectangular high activity region. If a rectangular refinement area is mapped around this irregular shape, much more grid points are needed than when the refinement only contains those control volumes that are actually marked for refinement.

An arbitrary shape in combination with parallel levels of refinement would allow for an even better efficiency or allow a larger variety of problems to be investigated. Consider the situation where two blobs of concentration are initially present, or when

the large blob of concentration is split into multiple smaller blobs, as is found in the collision of a dipole with a wall.

Refinement topology

Another recommendation is the use of a more flexible refinement topology that allows a refinement factor of one. In a channel flow, the streamwise dimension is larger than the spanwise and wall-normal dimensions. Whereas the streamwise direction could benefit from an additional level of refinement, the other two directions do not require an additional level and this could be solved by refining the global grid in streamwise direction only. Using a large refinement factor is already possible, but this can have disadvantages for the positioning and interpolation.

After some time, the initial shape of the solution has changed completely. The initial refinement topology might not be optimal for this new situation and a second recommendation on the refinement topology is to allow it to be adjusted during the computation to match the actual situation. A good example is a small blob of concentration in a turbulent flow. Under the influence of diffusion, the gradients are smoothened, but the total area the refinement area covers has increased. A change in the number of refinement levels or refinement factor can provide a more optimal approach than to keep the one that was applied during the initialization of the simulation.

LDC in combination with LUGR

A recommendation on LDC is that it should be investigated whether the global correction of the coarsest levels of refinement is required for accuracy reasons. A combination of LDC and LUGR on different levels of refinement should be able to improve the efficiency without loosing much accuracy.

Bibliography

[1] Anthonissen, M.J.H., Hof van 't, B., Reuskens, A.A., A finite volume scheme for solving elliptic boundary value problems on composite grids, *Computing*, **61**, pp. 285-305, 1998.

[2] Anthonissen, M.J.H., Hof van 't, B., Reuskens, A.A., An iterative finite volume discretization method for solving elliptic boundary value problems on locally refined grids, *Nieuw Archief voor Wiskunde*, **17**, pp. 111-123, 1999.

[3] Anthonissen, M.J.H., *Local defect correction techniques: analysis and application to combustion*, Ph.D. Thesis, Eindhoven University of Technology, 2001.

[4] Balachandar, S., Maxey, M.R., Methods for Evaluating Fluid Velocities in Spectral Simulation of Turbulence, *J. of Comp. Phys.*, **83**, pp. 96-125, 1989.

[5] Batchelor, G.K., *The Theory of Homogeneous Turbulence*, Cambridge University Press, Cambridge, 1956.

[6] Brandt, A., Multi-level adaptive solutions to boundary value problems. *Mathematics of Computation*, **31**, pp. 333-390, 1977.

[7] Bennet, B.A.V., and Smooke, M.D., Local rectangular refinement with application to axisymmetric laminar flames, *Combustion Theory Modeling*, **2**, pp. 221-258, 1998.

[8] Bennet, B.A.V., and Smooke, M.D., Local rectangular refinement with application to non reacting and reacting fluid problems, *Journal of Computational Physics*, **151**, pp. 684-727, 1999.

[9] Brethouwer, G., *Mixing of passive and reactive scalars in turbulent flows: A numerical study.*, Ph.D. Thesis, Delft University of Technology, 2000.

[10] Brouwers J.J.H., On diffusion theory in inhomogeneous turbulence, *J. Eng. Math.*, **44**, pp. 277-295, 2002.

[11] Boffetta, G., Pesceli, H.L., Trulsen, J., Numerical studies of turbulent particle fluxes to perfectly absorbing spherical surfaces, *Journal of Turbulence*, **22**, 2006.

[12] Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A., *Spectral methods in fluid dynamics*, Springer-Verlag, Berlin, 1988.

[13] Choi, J.I., Kyongmin, Y., Changhoon, L., Lagrangian Statistics in Turbulent Channel Flow, *Phys. Fluids*, **16** (3), pp. 779-793, 2004.

[14] Ferket, P.J.J., *Coupling of a global coarse discretization and local fine discretizations*, In Numerical Treatment of Coupled Systems, Notes on Numerical Fluid Mechanics; Editors: W. Hackbush, G. Wittum, Braunschweig, Vieweg, **51**, pp. 47 - 58 , 1995.

[15] Ferket, P.J.J., *Solving boundary value problems on composite grids with an application to combustion*, Ph.D. Thesis, Eindhoven University of Technology, 1996.

[16] Fox, D.G., Parker, I.B., *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, London, 1968.

[17] Gad-el-Hak, M., *Flow Control: Passive, Active, and Reactive Flow Management*, Cambridge University Press, 2000.

[18] Geurts, B.J., Mixing efficiency in turbulent shear layers, *Journal of Turbulence*, **2**, pp. 1-23, 2001.

[19] Goey de, L.H.P., ten Thije Boonkkamp, J.H.M., *A flamelet description of premixed laminar flames and the relation with flame stretch*, Technical Report, Eindhoven University of Eindhoven, 1998

[20] Hackbush, W., *Local defect correction and domain decomposition techniques* In Defect Correction Methods. Theory and Applications, Computing; Editors: K. Bohmer and H.J. Stetter, Suppl. **5**, 83-113, Wien, New York, 1984. Springer.

[21] Heath, M.T., *Scientific Computing: an Introductory Survey*, McGraw-Hill Book Company, New York, 1997.

[22] Hinze, J.O., *Turbulence*, McGraw-Hill, New York, 1987

[23] Hof van 't, B. *Numerical aspects of laminar flame simulation*, Ph.D. Thesis, Eindhoven University of Technology, 1999.

[24] Hoogh de, J., *Modeling of a Passive Scalar in a Turbulent Pipe Flow, Using a Direct Numerical Simulation*, M.Sc. Thesis, Technische Universiteit Eindhoven, Department of Process Technology, 2003.

[25] Hoogh de, J., Kuerten, J.G.M., Brouwers, J.J.H., *Modeling of a passive scalar in a turbulent pipe flow using DNS*, in Direct and Large-Eddy Simulation V; Editors: R. Friedrich, B.J. Geurts and O. Metais, Munich, Germany, pp. 315-322, 2004.

[26] Hoogh de, J., Kuerten, J.G.M., *Application of local defect correction to a time dependent problem*, in Conference on Modeling of Fluid Flow, ISBN 9630603829; Editors: T. Janos, J. Vad, Budapest, Hungary, pp. 870-877, 2006.

[27] Hoogh de, J., Kuerten, J.G.M., *Application of local defect correction in a 3D turbulent channel flow*, in European Conference on Computational Fluid Dynamics, ECCOMAS 2006; Editors: P. Wesseling, E. Onate and J. Periaux, Egmond aan Zee, Netherlands, **207**, pp. 1-14, 2006.

[28] Kampen van, N.G., *Stochastic processes in Physics and Chemistry, revised and enlarged edition.* Elsevier, Amsterdam, 1992.

[29] Karniadakis, G.E., Israeli, M., Orszag, S.A., High-order splitting methods for the incompressible Navier-Stokes equations, *J. Comp. Phys.*, **97**, pp. 414-443, 1991.

[30] Kleiser, L. and Schumann, U., *Treatment of incompressibility andboundary conditions in 3-D numerical spectral simulations of plane channel flows*, in Proceeding of the Third GAMM-Conference on Numerical Methods in Fluid Mechanics; Editors: E.H. Hirschel, Vieweg-Verlag, Braunschweig, pp. 165-173, 1980.

[31] Kolmogorov, A.N., The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers, *Proc. R. Soc. Lond. A*, **434**, pp. 9-13, 1991*.

[32] Kolmogorov, A.N., Energy dissipation in locally isotropic turbulence. *Doklady AN SSSR*, **32**, pp. 1921, 1941

[33] Kramer, W., Minero, R., Clercx, H.J.H., Mattheij, R.M.M., *A finite volume local defect correction method for solving the transport equation*, submitted to Elsevier Science, 2006

[34] Kramer, W., *Dispersion of passive tracers in two-dimensional bounded flows*, Ph.D. Thesis, Eindhoven University of Technology, 2006.

[35] Kuerten J.G.M., *Computational Fluid Dynamics III*, Lecture notes, University of Twente, 1999.

[36] Kuerten J.G.M., Veenman, M.P.B., Brouwers, J.J.H., *Simulation of the Motion of Particles in Turbulent Flow*, in Direct and Large-Eddy Simulation IV; Editors: B.J. Geurts, R. Friedrich. O. Métais, Enschede, pp. 11-20, 2001.

[37] Kuerten, J.G.M., *Modeling of Physical Phenomena*, College paper Tue, 2001.

[38] Kuerten, J.G.M., Veenman, M.P.B. and Brouwers, J.J.H., *DNS and LES of flows with particles*, in WCCM V, Fifth World Congress on Computational Mechanics; Editors: H.A. Mang, F.G. Rummerstorfer and J. Eberhardsteiner, Vienna, 2002.

[39] Kuerten, J.G.M., *Large-eddy simulation and turbophoresis*, in Advances in Turbulence X; Editors: H.I. Andersson and P.-A Krogstad, Trondheim, Norway, pp. 225-228, 2004.

[40] Kundu, P.K., Cohen, I.M., *Fluid Mechanics*, Academic Press, San Diego, 2000.

[41] Van Leer, B., Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme, *J. Comput. Phys.*, **14**, pp. 361-370, 1974.

*English translation of the original Russian 1941-article in Dokl. Akad. Nauk SSSR

[42] LeVeque R.J., *Numerical Methods for Conservation Laws*, Birkäuser Verlag, 1992.

[43] Lyman, G.J., Methods for interpolation of 2-D histogram data: application to mineral libration data, *Powder Technology*, **83**, pp. 133-138, 1994.

[44] Maxey, M.R., Riley, J.J., Equations of motion for a small rigid sphere in a nonuniform flow, *Physics of fluids*, **26**, pp. 883-889, 1983.

[45] Mathews, J.H., and Fink, K.D., *Numerical methods using MATLAB*, Prentice Hall, 1999.

[46] McCormick, S., *Fast adaptive composite grid (FAC) methods: Theory for the variational case*, in Defect Correction Methods. Theory and Applications, Computing, Suppl.5; Editors: B ohmer,K. and Stetter, H.J. editors, pp. 115-121, Wien, Springer, New York, 1984.

[47] Minero, R., ter Morsche, H.G., Anthonissen, M.J.H., Convergence properties of the local defect correction method for parabolic problems, Technical Report CASA 05-40, Eindhoven University of Technology, Eindhoven, 2005.

[48] Minero, R., Anthonissen, M.J.H., Mattheij, R.M.M., Local defect correction for time-dependent partial differential equations. In *Proceedings of the 16th International Conference on Domain Decomposition Methods*, 2005. Available online at http://cims.nyu.edu /dd16/proceedings.html.

[49] Minero, R., Anthonissen, M.J.H., Mattheij, R.M.M., A local defect correction technique for time-dependent problems, *Numerical Methods for Partial Differential Equations*, **22**, pp. 128-144, 2006.

[50] Minero, R., Anthonissen, M.J.H., Mattheij, R.M.M., Solving parabolic problems using local defect correction in combination with the finite volume method, *Numerical Methods for Partial Differential Equations*, 2006.

[51] Minero, R., *Local Defect Correction for Time-Dependent problems*, Ph.D. Thesis, Eindhoven University of Technology, 2006.

[52] Moin, P., Kim, J., Numerical investigation of turbulent channel flow, *J. Fluid Mech.*, **118**, pp 341-377, 1982.

[53] Moin, P., Kim, J., Tackling Turbulence with Supercomputers, *Scientific American*, **97**, pp. 46-52, 1997.

[54] Moin P., Mahesh K., Direct Numerical Simulation: A Tool in Turbulence Research, *Annu. Rev. Fluid Mech.*, **30**, pp. 539-578, 1998.

[55] Monin, A.S., Yaglom, A.M., Statistical Fluid Mechanics, M.I.T. Press, London, 1971.

[56] Moser, R.D., Kim, J., and Mansour, N.N., Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$, *Phys. Fluids*, **11**, pp. 943-945, 1999.

[57] Murray, J.D., *Mathematical Biology*, Springer-Verlag, Berlin, 1990.

[58] Nefedov, S., *Simulation of glass flow in an oven, in: Progress in Industrial Mathematics at ECMI98*; Editors: L. Arketyd, J. Bergh, P. Brenner, R. Pettersson, pp. 106-113, Leipzig. B. G. Teubner, Stuttgart, 1999.

[59] Necker, F., Hartel, C., Kleiser, L., Meiburg, E., High-resolution simulations of particle-driven gravity currents, *International Journal of Multiphase Flow*, **28**, pp. 279-300, 2002.

[60] Nieuwstadt, F.T.M., *Turbulentie, inleiding in de theorie en toepassingen van turbulente stromingen*, Elinkwijk b.v., Utrecht, 1998.

[61] Oijen van, J.A., *Flamelet-Generated Manifolds: Development and Application to Premixed Laminar Flames*, Ph.D. Thesis, Eindhoven University of Technology, 2002.

[62] Oijen van, J.A., Goey de, L.P.H., A numerical study of confined triple flames using a flamelet-generated manifold, *Combustion theory and modeling*, pp. 144-163, 2004.

[63] Osborn, T., The role of turbulent diffusion for copepods with feeding currents, *Journal of Plankton research*, **18**, pp. 185-195, 1996.

[64] Pope, S.B., Lagrangian pdf methods for turbulent flows, *Annu. Rev. Fluid Mechanics*, **26**, pp. 23-63, 1994.

[65] Pope, S.B., The vanishing effect of molecular diffusivity on turbulent dispersion: implications for turbulent mixing and scalar flux, *J. Fluid Mech.*, **359**, pp. 299-312 , 1998.

[66] Sawford, B.L., Reynolds Number Effect in Lagrangian Stochastic Models of Turbulent dispersion, *Physics of Fluids A*, **3**, pp. 157-1588, 1991.

[67] Sawford, B.L., Yeung, P.K., Lagrangian Statistics in Uniform Shear Flow: Direct Numerical Simulation And Lagrangian Stochastic Models, *Physics of Fluids*, **13**, pp. 2627-2634, 2001.

[68] Saad, Y., SPARSKIT package, *http://www-users.cs.umn.edu/saad/*, 2006

[69] Salama, A.I.A., A technique for interpolation and smoothing of mass-size-density data, *International Journal of Mineral Processing*, **70**, pp. 123-146 , 2003.

[70] Shan, H., Ma, B., Zhang, Z., Nieuwstadt, F.T.M., Direct Numerical Simulation of a Puff and a Slug in Transitional Cylindrical Pipe Flow, *J. Fluid Mech.*, **387**, pp. 39-60, 1999.

[71] Suzuki, Y., Kasagi,Y., Turbulent air-flow measurement with the aid of 3-D particle tracking velocimetry in a curved square bend, *Flow, turbulence and combustion*, **63**, pp. 415-442, 2000.

[72] Taylor G.I., Statistical theory of turbulence, I-III, *Proc. Roy. Soc.*, **151**, pp. 421-464, 1935.

[73] Tennekes, H. and Lumley, J.L.,*A First Course in Turbulence.*, M.I.T. Press, 1994.

[74] Trompert, R.A., Local uniform grid refinement and systems of coupled partial differential equations. *Applied Numerical Mathematics*, **13**, pp. 251-270, 1993.

[75] Trompert, R.A., *Local uniform grid refinement for Time-Dependent Partial Differential Equations*, Ph.D Thesis, University of Amsterdam, Amsterdam 1994.

[76] Veenman, M.P.B., Kuerten, J.G.M. and Brouwers, J.J.H., *Particle dispersion in inhomogeneous turbulent flow*, in: Liu C., Sakell L., Beutner T. (eds.), "DNS/LES progress and Challenges", Arlington, Texas, 549-556, 2001.

[77] Veenman, M.P.B., *Statistical Analysis of turbulent pipe flow: a numerical approach*, Ph.D Thesis, Eindhoven University of Technology, 2004.

[78] Vuik, C., *Iterative solution methods*, Lecture notes, Delft University of Technology, 2004.

[79] Warhaft, Z., Passive scalars in turbulent flows, *Annual Revie of Fluid Mechanics*, **32**, pp. 203-240, 2000.

[80] Walpot, R.J.E., van der Geld, C.W.M., Kuerten, J.G.M., 3D particle tracking velocimetry in turbulent pipe flow: Eulerian and Lagrangian statistics at moderate Reynolds numbers, *Phys. Fluids*, **1(1)**, pp. 1-20, submitted, 2006.

[81] Walpot, R.J.E., Rosielle, P.C.J.N., van der Geld, C.W.M., Design of a set-up for high-accuracy 3D PTV measurements in turbulent pipe flow, *Meas. Sci.Techn.*, **17**, pp. 3015-3026, 2006.

[82] Yeung, P.K., Pope, S.B., An Algorithm for Tracking Fluid Particles in Numerical Simulation of Homogeneous Turbulence, *J. of Comp. Phys.*, **79**, pp. 373-416, 1988.

Summary

Application of Local Defect Correction to a passive tracer in a turbulent channel flow.

Turbulence is a flow phenomenon that contains a broad spectrum of structures. In the viscous subrange, turbulence becomes isotropic and this allows for stochastic modeling of turbulence. These models are often used in computational fluid dynamics and more information on turbulent flows is needed to improve them. Individual tracks of marked fluid particles, released from a reference position, can be extracted from measurements or simulations. A different approach is to compute a probability density function of these marked particles, which is equivalent to the concentration of a passive tracer. This thesis will discuss the implementation of a grid refinement method called Local Defect Correction to a three-dimensional time-dependent problem: the development of a passive tracer in a turbulent channel flow. The grid refinement technique is aimed at resolving the small concentration scales occurring at high Schmidt numbers.

The turbulent flow field is computed using Direct Numerical Simulation of a channel flow that is periodic in stream and spanwise direction. Because of the periodicity, a Fourier-Galerkin spectral approach is applied in those two directions. In wall normal direction, a Chebyshev expansion is used. Time integration is performed using a second order accurate method that is a combination of an implicit Crank-Nicolson method for the viscous and pressure terms and the Adams-Bashforth method for the nonlinear terms. In this approach the pressure is corrected in such a way that the velocity field is divergence free.

When the convection-diffusion equation for the passive tracer is made dimensionless, the Schmidt number appears, which is the ratio between kinematic viscosity and diffusivity. If the Schmidt number increases, the structures that appear within the tracer field become smaller. For a high Schmidt number, these structures are smaller than those present within the velocity field. In order to resolve all details of the tracer field, a very fine computational grid is needed and that is very difficult with the current computational resources. The convection-diffusion equation is discretized using a finite volume method and different numerical schemes are compared. The result of this comparison is that, with an increased Schmidt number, a third order upwind method provides the best combination of accuracy, stability and efficiency. Furthermore, the results reveal that, for an increased Schmidt number, the required number of grid points becomes prohibitively large for a conventional technique to be applied. A very promising and efficient strategy is called local grid refinement.

The idea behind this approach is to focus the available computational power on areas where the concentration is present. Grid refinement methods contain a local fine grid (fine enough to capture the smallest scales) and a global coarse grid and after each time step, both solutions are combined.

One of the main issues of a grid refinement method in a time-dependent problem is the positioning and movement of the refinement area. Following the work done by Bennet and Smooke and Anthonissen, adaptations are made that allow the method to be used in a time-dependent problem. If the refinement area is moved into an area where no fine grid information is available, this information has to be obtained via interpolation of the coarse grid concentration. A new interpolation strategy was developed that ensures smooth and positive concentration.

One specific grid refinement method is called Local Defect Correction. Following the work done by Anthonissen and Minero, LDC is applied to the convection-diffusion equation. LDC has the advantage over other solution strategies, that the fine grid information is used to correct the global coarse grid solution, using a defect term. This defect is the difference between the coarse and the fine grid solution. Using an implicit method, the global coarse grid estimate is corrected by applying the defect to the original equation for the coarse grid. Where previous work focused on two-dimensional problems, in this thesis the step from two towards three dimensions is made. When moving towards three dimensions, the large increase in grid points becomes problematic and several adaptations to the original strategy are needed to ensure a proper functioning of LDC. The most important adaptation is the introduction of the local Schmidt number that ensures a smooth solution on the coarser grids, where a limited number of grid points can cause stability problems. With the improvements applied, simulations are performed and from the results it can be concluded that LDC is capable of computing an accurate solution to a three-dimensional time-dependent problem. However, the problem has to satisfy a constraint that can be summarized by the following: *if the high activity region is and remains small compared to the global computational domain, the application of LDC to a three-dimensional time-dependent problem can be successful*. Another grid refinement method is called Local Uniform Grid Refinement that uses an explicit instead of an implicit defect correction. When the performance of LDC and LUGR is compared, it can be shown that LDC is more accurate, but requires more than twice the computational time.

With a slightly altered convection-diffusion equation, different types of problems can be simulated. The altered properties of the concentration field can be examined with respect to the applicability of LDC and are introduced using very simplified physical models of sedimentation, population growth and a turbulent flamefront. Using these physical problems as a background, altered velocity components, increasing gradients, a positive source term and a rapidly increasing refinement area can be created. Simulations reveal the behavior of LDC with respect to the above problems. Especially a rapidly increasing refinement area or increasing gradients are problematic in combination with LDC. The positive source term does cause a growth in the refinement area, but the growth speed can be influenced by altering the magnitude of the source term and does allow LDC to be applied. The sedimentation problem with the altered velocity components does not cause any additional difficulties.

Samenvatting

Het toepassen van Lokale Defect Correctie op een passieve grootheid in een turbulente kanaalstroming.

Turbulentie is een fenomeen met een zeer breed spectrum van schalen. In de viskeuze sublaag mag turbulentie als isotroop worden verondersteld. Deze eigenschap kan worden gebruikt om turbulentie te modelleren met behulp van zogenaamde stochastische rekenmodellen. Meer informatie over turbulente stromingen is noodzakelijk om deze modellen te verbeteren. Deze informatie kan worden verkregen met behulp van fysieke experimenten of numerieke simulaties. Het doel van beiden is het vergaren van banen van gemarkeerde vloeistofdeeltjes, die worden losgelaten van een vaste positie. Een alternatief voor deze deeltjesbanen is het uitrekenen van de concentratie van een passieve grootheid. In dit proefschrift wordt de implementatie van een roosterverfijningsmethode genaamd Lokale Defect Correctie besproken, welke wordt toegepast op een driedimensionaal, tijdsafhankelijk probleem: het verloop van een passieve grootheid in een turbulente kanaalstroming. De roosterverfijningsmethode is bedoeld om de kleine schalen van de concentratie uit te kunnen rekenen welke ontstaan bij hoge Schmidt getallen.

Het benodigde turbulente stromingsveld wordt berekend door middel van een Directe Numerieke Simulatie die, in zowel de stroom als de dwarsrichting, periodieke grenzen heeft. In de richting loodrecht op de wand wordt een Chebyshev expansie toegepast. Tijdsintegratie wordt gedaan met behulp van een tweede orde nauwkeurige Crank-Nicolson methode voor de viskeuze en de drukterm. De niet-lineaire term wordt uitgerekend met behulp van een Adams-Bashforth methode. Door een drukcorrectie methode wordt een divergentie vrij stromingsveld gegenereerd.

In de dimensieloze convectie-diffusie vergelijking komt het Schmidt getal voor, dat de verhouding tussen viscositeit en diffusie weergeeft. Als het Schmidt getal toeneemt, ontstaan er structuren in het concentratieveld die kleiner zijn dan de lengteschalen in het stromingsveld, waarvoor een erg fijn rekenrooster benodigd is. De convectie-diffusie vergelijking wordt gediscretizeerd door middel van een eindige volume methode waarbij verschillende numerieke schema's met elkaar worden vergeleken met als resultaat dat een derde orde nauwkeurige upwind methode het meest geschikt is. Deze methode heeft de beste verhouding tussen nauwkeurigheid, stabiliteit en efficiëntie. Een tweede resultaat is dat, bij een verhoogd Schmidt getal, het aantal roosterpunten zo hoog wordt dat een conventionele aanpak niet meer mogelijk is. Een veelbelovende oplossing voor dit probleem is het rekenrooster lokaal te verfijnen. Het idee achter

roosterverfijning is dat het merendeel van de beschikbare rekenkracht gebruikt wordt in die gebieden waar daadwerkelijk concentratie aanwezig is. Hiertoe wordt gebruik gemaakt van een lokaal fijn rooster (om alle details uit te rekenen) en een globaal grof rooster. Na iedere tijdstap worden de oplossingen van beide roosters gecombineerd.

Een van de moeilijkheden bij het toepassen van een verfijningsmethode op een tijdsafhankelijk probleem is het bepalen van de positie van het fijne rooster. Voortbordurend op het werk van Bennet and Smooke en Anthonissen worden een aantal aanpassingen gemaakt om de detectiemethode geschikt te maken voor een tijdsafhankelijk probleem. Als het fijne rooster verschoven wordt naar een gebied waar alleen grove informatie beschikbaar is, is het alleen via interpolatie van het grove rooster mogelijk om de juiste waardes op het fijne rooster te achterhalen. Een nieuwe interpolatie strategie is ontwikkeld, die een glad en positief verloop van de oplossing garandeert.

Een specifieke verfijningsmethode heet Lokale Defect Correctie. Gebruik makend van het werk van Anthonissen en Minero is LDC toegepast op een driedimensionaal tijdsafhankelijk probleem. LDC heeft als voordeel dat het in staat is om het globale rooster te corrigeren met behulp van een defectterm. Dit defect wordt bepaald uit het verschil tussen de oplossing op het fijne en het grove rooster en zal, met behulp van een impliciete rekenmethode, gebruikt worden om het grove rooster te verbeteren. Eerder werk met betrekking tot LDC richtte zich vooral op tweedimensionale problemen en in dit proefschrift wordt de stap van twee naar drie dimensies gemaakt. Het grootste probleem dat zich voordoet bij het implementeren van LDC in drie dimensies, is de grote hoeveelheid roosterpunten. Om LDC correct te laten werken zijn een aantal aanpassingen nodig, waaronder het toepassen van een lokaal Schmidt getal, welke een gladde oplossing garandeerd, desondanks dat er op de grovere roosters maar een beperkt aantal punten beschikbaar is. Uit de resultaten van verscheidene simulaties blijkt dat het goed mogelijk is om Lokale Defect Correctie toe te passen op een driedimensionaal tijdsafhankelijk probleem. Het probleem moet echter wel aan een aantal voorwaarden voldoen, welke samengevat neerkomen op: *als het verfijningsgebied klein is en klein blijft ten opzicht van het globale rekendomein, kan LDC een geschikte oplosmethode zijn voor driedimensionale tijdsafhankelijke problemen.* Een andere methode is Lokale Uniforme Rooster Verfijning welke gebruikt maakt van een expliciete in plaats van een impliciete correctie. Het blijkt dat, als LDC met LURV wordt vergeleken, LDC nauwkeuriger is, maar ook twee maal zoveel rekentijd gebruikt.

Met een aangepaste convectie-diffusie vergelijking is the mogelijk om problemen te simuleren met verschillende eigenschappen, waardoor de toepasbaarheid van LDC onder veranderende condities onderzocht kan worden. De verschillende eigenschappen van de convectie-diffusie vergelijking worden geïntroduceerd aan de hand van versimpelde modellen van sedimentatie, populatiegroei en een turbulent vlamfront. Hiermee kunnen een veranderende snelheidscomponent, toenemende gradiënten, een positieve bronterm en een snel, in volume, toenemend verfijningsgebied worden gerealiseerd. Met name een snel, in volume, toenemend verfijningsgebied of continue toenemende gradiënten zijn moeilijk op te lossen met LDC. De bronterm zorgt eveneens voor een toenemend verfijningsgebied, maar de grootte van deze term kan zo worden gekozen dat LDC toch toegepast kan worden. Een probleem waarin de snelheidsvector wordt aangepast, zorgt voor geen problemen in combinatie met LDC.

Dankwoord

Dit proefschrift zou niet compleet zijn zonder iedereen te bedanken die de afgelopen vier jaar bij mij en mijn onderzoek betrokken zijn geweest.

Allereerst gaat mijn dank uit naar de NWO (Nederlandse Organisatie voor Wetenschappelijk Onderzoek) die dit onderzoek gefinancierd heeft. Daarnaast maakte dit onderzoek deel uit van het interdisciplinaire SMARTER-project, waaraan vier mensen werkten: Ion Barosan (Wiskunde & Informatica), Werner Kramer (Natuurkunde), Remo Minero (Wiskunde & Informatica) en ikzelf (Werktuigbouwkunde) en hun respectievelijke begeleiders, Jack van Wijk, Herman Clercx, Bob Mattheij en Hans Kuerten. Gedurende de afgelopen vier jaar heb ik met deze personen prettig kunnen samenwerken en discussiëren over het implementeren van Lokale Defect Correctie. Via het SMARTER-project heb ik de mogelijkheid gekregen om gebruik te maken van de supercomputer-faciliteiten van de NCF (Stichting Nationale Computer Faciliteiten). Mijn dank hiervoor.

Een promotie onderzoek doe je niet alleen. Ik wil graag een aantal mensen bedanken die me geholpen hebben tijdens het gehele traject. Bert Brouwers onder wiens vlag ik veel heb geleerd over turbulentie en stochastiek, Cees van der Geld voor zijn altijd praktische insteek op Lokale Defect Correctie (LDC) en verder: Herman Clercx, Erik Dick, Bernard Geurts en Bob Mattheij voor doorlezen van het conceptproefschrift en de daaruit voortvloeiende opbouwende kritiek.

Één persoon wil ik in het bijzonder bedanken die mij de afgelopen jaren door middel van woord en daad heeft voorzien van kennis, advies, humor en tevens een hoop werk: Hans Kuerten. Hans, je deur stond altijd open en je adviezen hebben me er altijd toe gezet door te gaan met het uitpluizen van allerlei nieuwe invalshoeken om LDC toe te kunnen passen op een driedimensionaal probleem. Ook je geduld en hulp bij het tot stand komen van dit proefschrift is iets waar ik je zeer dankbaar voor ben. Niet alleen heb je me geholpen op wetenschappelijk gebied, maar ook onze gesprekken over sport, klassieke muziek, reizen en de alledaagse zaken heb ik altijd erg gewaardeerd. Het feit dat je naar Athene bent gekomen om naar mijn Paralympische finale te komen kijken zegt in mijn ogen meer dan genoeg over je betrokkenheid bij mijn promotie.

Niet alleen Hans, maar ook mijn twee kamergenoten Maurice en David hebben mij altijd van veel plezier en hun (on)gevraagde professionele mening voorzien over de meest onmogelijke zinsconstructies, de beste film-quotes, onze tennis- en zwemavonturen en natuurlijk over eersteklas software van eigen makelij. Deze software werkte

in de praktijk, helaas meer dan eens, minder goed dan op papier. Van Maurice heb ik de eerste twee jaar erg veel geleerd en ik heb dan ook dankbaar gebruik gemaakt van de door hem ontwikkelde DNS-code en de lay-out voor dit proefschrift. Al dit bovenstaande geldt ook voor Natascha die vaak, standaard om drie uur, een kop thee mee kwam drinken. Deze traditie kwam voort uit de wekelijkse 'turbo-meetings', waarin er veelvuldig werd nagedacht over de praktische problemen met betrekking tot turbulentie, meetopstellingen en programmatuur.

Een ander belangrijk punt van promoveren is het relativeren van de wetenschap. De dagelijkse lunch met de andere promovendi: Arend, Carlo, David, Guy, Nicole, Ralph, Raymond was hiervoor altijd een zeer geschikt moment. Ook de 'studiereis' naar Zwitserland is iets waar ik, ondanks de overstroming, met veel plezier op terug kijk. In de zomer met de vakgroep een week naar Zwitserland is niet iets wat je overal ziet, Bert, Cees, Erik, Frank, Hans en natuurlijk de bovenstaande aio's hebben dat tot een zeer geslaagde en leerzame week weten te maken. Ook het secretariaat was, mede door de aanwezigheid van Linda en Mechline, een plek waar altijd wel een praatje te maken viel. Daarnaast zorgden zij altijd keurig voor het afhandelen van de papierwinkel, welke zonder de wetenschap niet zou kunnen bestaan.

In mijn dagelijks leven buiten de wetenschap heb ik altijd graag mijn baantjes getrokken in het zwembad de Tongelreep en ben ik vanwege het zwemmen de wereld over geweest. Iedereen van PSV-zwemmen en de Nationale ploeg van de Nebas heeft ervoor gezorgd dat ik momenten heb mogen meemaken die ik niet snel zal vergeten. Onder begeleiding van Erik of Erwin waren de dagelijkse trainingen in Eindhoven een welkome afwisseling op het programmeren. Dit geldt zelfs voor de ochtendtrainingen om zes uur, waar ik, met een gezonde dosis frisse tegenzin, naar toe ging. Het nationale zwemteam stond de laatste jaren onder leiding van Chris en Hans en ik heb genoten van iedere reis die ik met hun, en mijn vaste kamergenoot Kasper, heb gemaakt. Het deelnemen en behalen van medailles op de Paralympische Spelen en de Wereldkampioenschappen is een magische belevenis! Hierbij wil ik ook de TUE en mijn collega's bedanken voor het verlenen van hun volledige medewerking in het bedrijven van topsport.

Als je meer dan twintig uur in de week aan sport besteed en daarnaast bezig bent met de implementatie van LDC op een driedimensionaal en tijdsafhankelijk probleem, is er weinig tijd en energie over voor andere dingen. Er is echter altijd tijd geweest voor etentjes, films, spelletjes en sport met Bart, Carlie, Joost, Ronald, Roos en vele anderen. Hierdoor kwam ik, met veel plezier, ook nog eens op andere plaatsen dan de TUE en het zwembad.

Als ik dan in het weekend, meer dan eens, uitgeput in Ulvenhout bij mijn ouders op de bank lag, heb ik daar altijd de steun en liefde gekregen waardoor ik dit allemaal heb kunnen bereiken. Niet alleen mijn ouders, maar ook mijn beide oma's, Floor en Marnix en mijn vriendin Ingrid hebben me in alles door dik en dun gesteund en ik ben ze daarvoor heel erg dankbaar.

Curriculum Vitae

Joost de Hoogh was born in Breda on September 30, 1979. From 1991 untill 1997 he attended VWO (Voorbereidend Wetenschappenlijk Onderwijs) at the Mencia de Mendoza lyceum in Breda. In 1997 he started to study Mechanical Engineering at the Technische Universiteit Eindhoven. In his second and third year a side track within the Mechanical Engineering department was taken that gave attention to several aspects of the medical side of engineering. In the two last years of his study, he did an internship to develop a numerical model that could predict the effectiveness of cooling the human head, wearing a coldcap. A second internship was done in collaboration with ECN (Energie Centrum Nederland) and focused on performing measurements together with an efficiency analysis on an experimental solar panel. His master thesis was about the modeling of a passive tracer in a turbulent pipe flow, using a Direct Numerical Simulation and was performed in the Process Technology Group of prof.dr.ir. J.J.H. Brouwers.

Directly after his graduation in 2003, Joost continued his graduation work by starting a Ph.D research in the same group under the supervision of dr. J.G.M. Kuerten. In four years of research, he investigated the possibility to apply a technique called Local Defect Correction could to a three-dimensional time-dependent problem: the concentration equation for a passive tracer in a turbulent channel flow. His daily work consisted of finding practical solutions to various numerical problems that arose during the implementation and testing of LDC. In the end it became clear that LDC is applicable, but restricted to very specific situations. From his daily work he gained experience in programming, developing and realizing new ideas and writing. Aside from his daily work he guided two students in their bachelor projects, assisted with classical exercise and tutored a group of students with their OGO-projects.

Next to his daily occupation as a student, Joost spent much time in the swimming pool, where he trained in preparation of (inter)national competitions. He was a participant at many Dutch, European and World Championships and three Paralympic Games in Atlanta (1996), Sydney (2000) and Athens (2004). Highlights of his career are: a bronze medal on the 400 meter free-style during the Paralympic Games in Sydney, a world record on the 400 meter individual medley, a gold medal on the European Championship in Braunschweig (1999) and several silver and bronze medals in European and World Championships.