Infotaxis in a turbulent 3D channel flow

Eggels, A.W.

Award date:
2015

TU Eindhoven

Master thesis

Infotaxis in a turbulent 3D channel flow

Author:
Anne Eggels

Supervisors:
dr.ir. R.P.J. Kunnen
prof.dr.ir. B. Koren

April 20, 2015

Abstract

In this report, the infotaxis-based search algorithm is tested in several turbulent (channel) flows. A kinematic simulation is used to test this search algorithm by detection of particles both in two and three dimensions. Also, the algorithm is adapted to detect plumes of high concentration instead of independent particles. Direct numerical simulation is used to test this adapted search algorithm by detection of high concentration levels in turbulent channel flows with Schmidt number Sc = 1.00 and Reynolds numbers of 5600 and 28000.

For the kinematic simulations, a positive relation was found between the starting distances to the source and the running time. For the direct numerical simulation with the adapted algorithm, this relation holds for Re = 5600 but is not observed at Re = 28000. This is caused by the low Schmidt number and the high velocity, which leads to the source very fast after the first detection of a high concentration level.

The search algorithm is also tested in reverse to detect whether a fluid is well-mixed. The time required for a detection and the number of detections are used as a measure for success. By applying the algorithm to some prescribed concentration distributions in two dimensions, it is found that the algorithm is very sensitive to the threshold values for the mixing indicators.

Contents

Chapter 1

Introduction

Have you ever wondered why insects in a garden follow a zigzag pattern most of the time? By mixing, regions of high concentrations of odour disconnect into random patches of odour, which makes gradient searching an ineffective strategy to find the source, i.e. the flowers. Hence, another strategy is needed, which should incorporate the availability of only sparse and partial information. One strategy available is called *infotaxis*. This strategy maximizes the expected gain in information locally. This gain can be measured by means of an entropy function. Using information which the searcher receives during its search, a probability distribution can be computed which is based on the information available.

The infotaxis strategy search algorithm has been proposed by Vergassola et al. [1] and it was tested by experimental data on mixing flows. Moraud & Martinez [2] assessed the performance of infotaxis by combining robotic experiments and simulations. They conclude that the biomimetic characteristic of infotaxis (i.e., the zigzag pattern) is conserved when a robot is searching in a real environment. Barbieri et al. [3] presented a continuous-space version in both two and three dimensions which is analysed both analytically and numerically. Karpas & Schneidman [4] extended infotaxis to group behaviour, where each individual tries to maximize its own information, but is also able to use information from others. Recently, Hesselberth [5] tested the infotaxis algorithm in a two-dimensional kinematic simulation.

The infotaxis algorithm can be applied more broadly, for example to find oil in water when an oil pipeline on the bottom of the ocean is leaking, or when air around an industrial zone is polluted with some pollutant. While these situations vary in dimensions, substances involved and other conditions, the general idea of searching the source by using only the information of small patches of some material (measurable in concentration) stays the same.

First, in Chapter 2, the infotaxis algorithm will be explained. In Chapters 3 and 4, this algorithm will be applied to a channel flow, in which the source emits particles which can be detected. The channel flow is set up by a kinematic simulation in two or three dimensions to simulate a turbulent flow and the movement of the particles is calculated with a Runge-Kutta method. The searcher is limited to move in certain directions.
Afterwards, in Chapters 5, 6 and 7, the channel flow is governed by the Navier-Stokes equations and solved by a direct numerical simulation. This is done for two turbulent flows with Reynolds numbers 5600 and 28000. This flow is periodic in the streamwise and spanwise direction, and bounded by plates in the normal direction. Here, the source is continuous and the flow of the emitted concentration is calculated by an upwind discretisation. Also in these situations, the searcher is bounded to move on a grid.
In Chapter 8, two applications of the infotaxis-based search algorithm are set up with various time and length scales.
Finally, in Chapter 9, the infotaxis algorithm is used in the reverse way: to find a measure to check whether a fluid is well-mixed. Too high or too low concentrations might indicate a source or sink, which is absent in a well-mixed fluid.

Chapter 2

Infotaxis

Infotaxis is a search algorithm to find a source of some detectable stuff, which can vary from odour traces to oil and from poisonous gases to pollen which cause hay fever. All these examples have one thing in common: they diffuse in the carrier fluid and their traces are sparse and whimsical. Conventional search algorithms often use gradients: the source will often be found if the searcher moves towards the direction with highest concentration of detectable stuff. Here, this will not work because the traces are sparse. This is where infotaxis comes into play.

The infotaxis algorithm is not based on gradients, but on the expected gain of information. This information consists of a calculated probability density of the presence of a source at certain positions in the search area. By making use of properties of the carrier fluid and the detectable stuff, one can estimate the probability that the stuff is detected at a certain position in space. By waiting some time at one position, the searcher tries to detect the stuff. These detections are called *hits*. Both when stuff is and is not registered, this gives new information for the probability density. From this density, it is calculated which of the following two options is executed: the searcher moves to a neighbouring position or stays at the same position, according to which of the two is expected to give more information. This is repeated until the source is found.

The properties of the carrier fluid and the detectable stuff are represented in the expected rate function $R(\mathbf{r}|\mathbf{r}_0)$. This function calculates the expected number of hits at position $\mathbf{r}$ given the source position $\mathbf{r}_0$. When it is assumed that the stuff satisfies the advection-diffusion equation with a source term, as given in the following equation

$$0 = V\nabla_y C(\mathbf{r}|\mathbf{r}_0) + D\Delta C(\mathbf{r}|\mathbf{r}_0) - \frac{1}{\tau}C(\mathbf{r}|\mathbf{r}_0) + S\delta(\mathbf{r}-\mathbf{r}_0), \tag{2.1}$$

in which V is the mean velocity in the y-direction, D the isotropic diffusion coefficient, τ the lifetime of the stuff once emitted, S the emission rate of the source and δ the Kronecker delta. In two dimensions, the solution of (2.1) is given by [6] to be

$$C(\mathbf{r}|\mathbf{r}_0) = \frac{S}{2\pi D}\exp\left(\frac{-(y_0-y)V}{2D}\right)K_0\left(\frac{|\mathbf{r}-\mathbf{r}_0|}{\lambda}\right), \quad \lambda = \sqrt{\frac{D\tau}{1+\frac{V^2\tau}{4D}}}, \tag{2.2}$$

in which K_0 is the modified Bessel function of the second kind of order 0. Using the Smoluchowski expression for the rate of encounters between a spherical particle of radius a and molecules diffusing with effective diffusivity D, as given in [7], the expected rate function becomes

$$R(\mathbf{r}|\mathbf{r}_0) = \frac{S}{\log\left(\frac{\lambda}{a}\right)}\exp\left(\frac{-(y_0-y)V}{2D}\right)K_0\left(\frac{|\mathbf{r}-\mathbf{r}_0|}{\lambda}\right), \quad \lambda = \sqrt{\frac{D\tau}{1+\frac{V^2\tau}{4D}}}, \tag{2.3}$$

in which a correction has been made regarding the finite lifetime of the particles. In this function, a represents the radius of the searcher, $\mathbf{r} = (x, y, z)$ the current position and $\mathbf{r}_0 = (x_0, y_0, z_0)$ the position of the source. This holds when the mean velocity V is in the positive y-direction. It is visible that the logarithm induces some problems: λ should be larger than a, which gives a maximum value for V, otherwise the rate of encounters becomes negative. This is due to the fact that the model is derived for 3D, and mathematically, there is also a 2D solution, as given in (2.2), which does not represent the

physics for some aspects. In 3D, the solution of equation (2.1) is in [6] given by

$$C(\mathbf{r}|\mathbf{r}_0) = \frac{S}{4\pi D\,|\mathbf{r}-\mathbf{r}_0|}\exp\left(\frac{-(y_0-y)V}{2D}\right)\exp\left(-\frac{|\mathbf{r}-\mathbf{r}_0|}{\lambda}\right), \tag{2.4}$$

with λ the same as before and the expected rate of encounters is now given by

$$R(\mathbf{r}|\mathbf{r}_0) = \frac{aS}{|\mathbf{r}-\mathbf{r}_0|}\exp\left(\frac{-(y_0-y)V}{2D}\right)\exp\left(-\frac{|\mathbf{r}-\mathbf{r}_0|}{\lambda}\right). \tag{2.5}$$

To make use of this expected rate of encounters, a probability distribution $P(t|\mathbf{r}_0)$ is used, which gives the probability at time t that the source can be found at place $\mathbf{r}_0$. When starting the algorithm, the probability is equal everywhere in the search space. It is assumed that the detections of individual hits are independent, and hence the detections follow a Poisson process. In this way, the probability at each position might be updated after a time δt using the rate of encounters $R(\mathbf{r}|\mathbf{r}_0)$ and the number of hits η detected by

$$P(t+\delta t|\mathbf{r}_0) = P(t|\mathbf{r}_0)\,(R(\mathbf{r}|\mathbf{r}_0))^{\eta}\,e^{\delta t\cdot R(\mathbf{r}|\mathbf{r}_0)}/Z_t, \tag{2.6}$$

in which Z_t is a normalizing constant. Also, when the searcher is at a location without the source, then the location's probability is set to 0, because the source does not move. Furthermore, this position is excluded from the entropy calculations.
From this probability function only, it is not clear at first sight in which direction the searcher should move, and the goal is to maximize the expected gain of information. For this, Shannon's entropy is used:

$$E_S = -\int P(\mathbf{x})\log(P(\mathbf{x}))\,\mathrm{d}\mathbf{x}, \tag{2.7}$$

in which the integral encloses the whole search area. This is an entropy which becomes 0 when the probability density becomes a delta function, i.e., when the source is found with certainty. Also, the entropy decreases when more information is available. This will be the main step in the search algorithm: for each neighbouring position of the searcher, the decrease in entropy is calculated by

$$\Delta E_S(\mathbf{r}\to\mathbf{r}_j) = P(t,\mathbf{r}_j)\cdot(-E_S) + (1-P(t,\mathbf{r}_j))\cdot\left(\sum_{k=1}^{\infty}\rho_k(\mathbf{r}_j)\Delta E_{S_k}\right). \tag{2.8}$$

The first term on the righthand side represents the probability that the source is found, which would decrease the entropy to 0. The second term represents the probability that the source is not found, but k hits are registered, which also decreases the entropy. ρ_k represents the probability that k hits are registered, which is calculated by the Poisson law:

$$\rho_k = h^k e^{-h}/k!, \quad h(\mathbf{r}_j) = \Delta t\int P(t,\mathbf{r_0})R(\mathbf{r}_j|\mathbf{r}_0)\,\mathrm{d}\mathbf{r}_0 \tag{2.9}$$

for independent detections. It has to be kept in mind that in this case, to calculate ρ_k, it is known that $P(t,\mathbf{r}_j) = 0$, because the source was not found.
The searcher then chooses the neighbouring position with the largest decrease in entropy, or stays at the same position if that is the best option.

Part I

Particle model

Chapter 3

Setup

We will first apply the infotaxis algorithm to point particles emitted by a source in a flow in which the turbulent flow is represented by a kinematically simulated flow. This to ease the simulation of the flow and to comply with the situation in [5].

3.1 Setup of the flow

The flow is represented by a kinematic simulation of a turbulent velocity field. The following explanation of the setup of a kinematic simulation follows the introduction of kinematic simulations by Elenbaas [8]. In a kinematic simulation, the velocity field is constructed by a sum of Fourier components:

$$\mathbf{u}(\mathbf{r},t) = \sum_{n=1}^{N_{max}} \left(\mathbf{v_n}\cos(\mathbf{k_n}\cdot\mathbf{r}+\omega_n t) + \mathbf{w_n}\sin(\mathbf{k_n}\cdot\mathbf{r}+\omega_n t)\right). \tag{3.1}$$

The directions of $\mathbf{v_n}$ and $\mathbf{w_n}$ are perpendicular to the wave vectors $\mathbf{k_n}$ to make the velocity field incompressible, but this still leaves multiple options. This option is chosen randomly. The frequency ω_n is added to make the flow unsteady, and this frequency is proportional to the eddy turnover time for $\mathbf{k_n}$, the wave vector of mode n, by

$$\omega_n = \gamma\sqrt{k_n^3 E(k_n)}, \tag{3.2}$$

in which γ is a weighing factor and $E(k)$ the Kolmogorov spectrum given by

$$E(k) = C\epsilon^{2/3}k^{-5/3}, \tag{3.3}$$

in which C is the Kolmogorov constant, ϵ the turbulent kinetic energy dissipation per mass unit and k the wave number.
The wave numbers k_n are geometrically distributed from k_c to k_η, in which $k_c = \frac{2\pi}{L}$ is the smallest wavenumber possible and $k_\eta = \frac{2\pi}{\eta}$ is the largest wavenumber possible with Kolmogorov length scale $\eta = \left(\frac{\nu^3}{\epsilon}\right)^{1/4}$, in which ν is the kinematic viscosity. For k-values outside this region, the energy is set to 0. The amplitudes of $\mathbf{v_n}$ and $\mathbf{w_n}$ are chosen as

$$|\mathbf{v_n}| = |\mathbf{w_n}| = \sqrt{2E(k_n)\Delta k_n}, \quad \Delta k_n = \frac{1}{2}\left(k_{n+1}-k_{n-1}\right), \quad \Delta k_1 = \frac{1}{2}(k_2-k_1), \quad \Delta k_{N_{max}} = \frac{1}{2}(k_{N_{max}}-k_{N_{max}-1}). \tag{3.4}$$

Furthermore, the time-integration of the velocity field is done by the fourth-order Runge-Kutta method. The time step is chosen such that the smallest time scales are properly represented. For this, $\Delta t < t_\eta$, where $t_\eta = \left(\frac{\nu}{\epsilon}\right)^{1/2}$ is the Kolmogorov time scale.

3.1.1 Parameters

The chosen physical parameters are:

- $\Delta x = \Delta y = \Delta z = 1$ m - distance between grid points,
- $N = 32$ (2D) resp. 16 (3D) - number of grid points for the width of the domain (x-axis),
- $M = 80$ (2D) resp. 32 (3D) - number of grid points for the length of the domain (y-axis),

- $Z = 1$ (2D) resp. 32 (3D) - number of grid points for the depth of the domain (z-axis),
- $\mathbf{r}_0 = (16, 12, 1)$ (2D) resp. $(8, 5, 8)$ (3D) - the position of the source,
- $D = 0.72$ m^2/s - isotropic effective diffusivity,
- $\tau = 1 \cdot 10^6$ s - lifetime of emitted particles,
- $a = 0.5$ m - radius of the searcher.

The values of D, τ and a are chosen to comply with the simulations in [5]. The kinematic parameters are:

- $V = 1.4$ m/s - mean velocity in y-direction,
- $N_{max} = 20$ - number of components in the kinematic simulation,
- $\nu = 1.568 \cdot 10^{-5}$ m^2/s - kinematic viscosity (of air at 300K),
- $L = M$ - characteristic macroscopic length,
- $\epsilon = 1$ m^2/s^3 - turbulent dissipation per unit mass,
- $C = 1.5$ - Kolmogorov constant,
- $\gamma = 500$, weighing factor in ω,

from which is calculated that $t_\eta = 0.0040$ s.
Numerical parameters:

- $\Delta t = 1 \cdot 10^{-3}$ s - the numerical time step,
- $\delta t = 1$ s - the algorithm time step, waiting time for searcher,
- $T_{max} = 300$ s (2D) resp. 500 s (3D) - the maximum simulation time.

Furthermore, S in equations (2.3) and (2.5) needs to be calculated for a specific choice of number of particles released per second. In two dimensions, this is one particle per second and this is four particles per second in three dimensions. This is necessary because the expected rate function works with concentrations instead of particles. To do so, we sum the values of $R(\mathbf{r}|\mathbf{r}_0)$ in all grid points using for $\mathbf{r}_0$ the real source position for $S = 1$ particle per second. In two dimensions, the calculation is as follows:
The number of particles in a simulation is comparable to $\frac{M-y_0}{V} = \frac{(80-12)}{1.4} = 49$, because each time step, one new particle is generated, and it takes about $\frac{M-y_0}{V} = 49$ steps of δt time to reach the border. Hence, if the values of $R(\mathbf{r}|\mathbf{r}_0)$ at all possible positions are summed, the sum should be equal to 49. This summation has been executed in MATLAB and the sum was equal to 301.9718, hence S can be computed to be $\frac{49}{301.9718} = 0.1623$ s^{-1}. This holds if each δt, one particle is released.
In three dimensions, the number of particles in a simulation is comparable to $4\frac{M-y_0}{V} = 4 \cdot \frac{(32-5)}{1.4} = 77$, because each time step, 4 new particles are generated. Hence, if the values of $R(\mathbf{r}|\mathbf{r}_0)$ at all possible positions are summed, their sum should equal 77. Calculating with $S = 1$ particle per second gives an expected number of particles 80.7140, hence $S = \frac{77}{80.7140} = 0.9540$ s^{-1} compares to 4 released particles in each δt time step.

3.2 Execution of the algorithm

At each algorithm time step δt, the trajectories of the particles are calculated by the RK4-method. The algorithm checks at each of the numerical time steps Δt whether a particle is present within its diameter (or search range) $2a$. When this is the case, a hit is registered and the particle is removed, to avoid duplication of this hit. After one algorithm time step, the probability distribution is updated with the number of hits. On the basis of this new probability distribution, the next position is calculated and the algorithm starts again.

Chapter 4

Results

In this Chapter, the results for the setup given in Chapter 3 will be given. These results consist of an indication of the diffusion of the particles across the domain, one of the travelled trajectories, a relation between starting distance from the searcher to the source and an averaged entropy plot.

4.1 Results in 2D

4.1.1 Plume

First, the distribution of particles over the grid is represented in Figure 4.1. In this Figure, red represents a high particle density and blue a low or zero particle density visualised on a logarithmic scale. It is visible that around the source, the density has its maximum, and the particles are taken away in the flow direction. This Figure is made by calculating $2 \cdot 10^4$ δt time units, with one particle released each time unit, and registrering all the positions each 10 time units. To see the distribution for a certain y-coordinate, Figure 4.2 is made. From this, it is visible that the distribution resembles a normal distribution for $y/\Delta y = 20$. However, the simulation should take much longer to give a clearer result, also for the other y-coordinates.

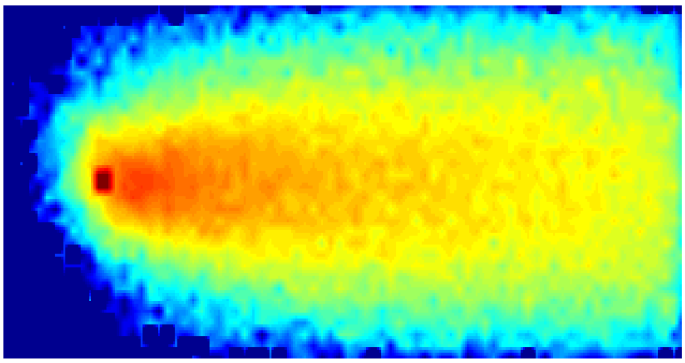


Figure 4.1: Plume - the source is located at the red square and the flow is directed from left to right. The x-axis is vertical and the y-axis horizontal, with the origin in the lower left corner.

4.1.2 Trajectory

In Sections 4.1.2 up until 4.1.4, the simulations were performed with a value of S which is 10% larger than calculated. This will cause the searcher to receive less hits than expected and thereby to have less information than in the ideal case. In Figure 4.3, one of the simulated search trajectories is visualised. The flow is directed from left to right and the colour indicates time which increases from blue to red. It is visible that first the searcher passes by the source, but then returns.

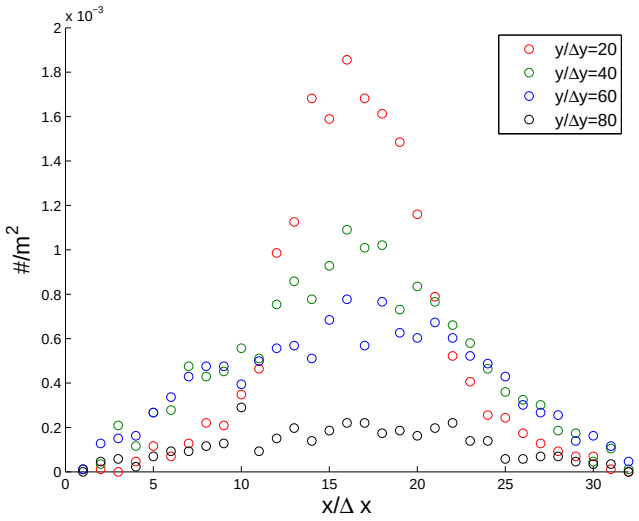


Figure 4.2: Density distribution for various $y/\Delta y$-values.

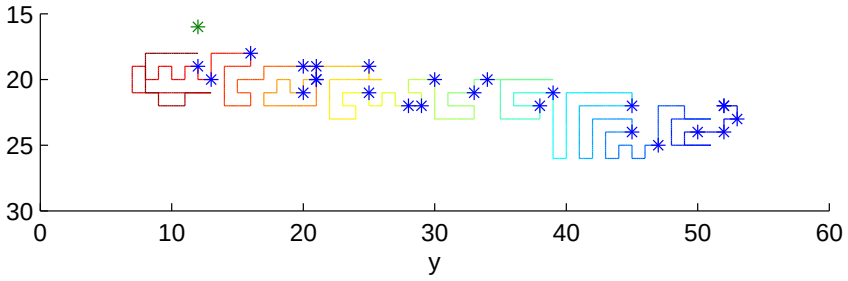


Figure 4.3: Search trajectory - the source is located at the green asterisk at (16,12) and the searcher starts at (22,52).

4.1.3 Relation between distance and search time

The algorithm has been executed 75 times with different starting positions, with 59 successes. Failures occurred because the simulation took too long or the searcher hit the edge. In the particle model simulations, this will not be allowed. The starting positions were uniformly distributed in $[15, 25] \times [30, 50]$. The results can be found in Figure 4.4. In this picture, only the successful runs are shown. Comparing the two fits, the residual sum of squares is a little smaller for the exponential fit, but this fit has more parameters which can cause this. However, it can not be expected that for a small starting distance, e.g. 5, the number of steps will be negative. Hence, the exponential fit might better resemble the results.

For the large starting distances, the needed number of steps reached the maximum of 300 steps. When this number is increased, more experiments will succeed. To check this, also 25 executions have been performed with the starting positions uniformly in $[15, 25] \times [50, 70]$ with $T_{max} = 500$, of which 21 executions succeeded. However, none of the extra experiments succeeded with more than 300 steps. This can be seen in Figure 4.5. The results of the fits can be found in Figure 4.6.

For this extended data set, again, the residual sum of squares is less for the exponential case. However, now the curvature has changed sign. The exponential fit is negative for small start distances and the linear fit is still positive for small start distances. Because of the added data points, these fits should be more reliable. When comparing the results to those in [5], which had a linear fit of $y = 38.0 + 1.8x$, the results are similar.

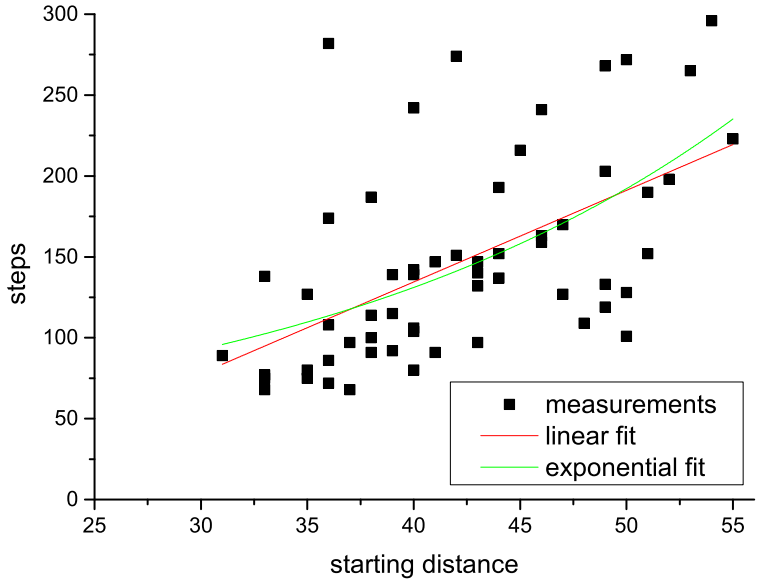


Figure 4.4: Relation between starting distance and search time - the linear fit is given by $y = a + bx$ with $a = -91.1$ and $b = 5.7$ and the exponential fit by $y = a + b\exp(cx)$ with $a = 28.1$, $b = 16.01$ and $c = 0.0466$.

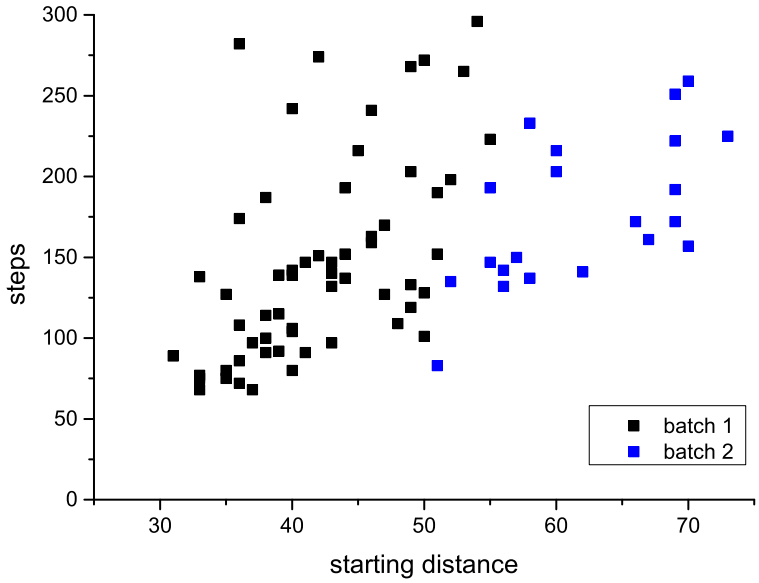


Figure 4.5: Relation between starting distance and search time - the extra simulations are given in blue.

4.1.4 Relation between search time and entropy

The average entropy and its standard deviation are given in Figure 4.7. This averaging includes also the runs which failed or ended. In these cases, the entropy is kept at the same value until $T = 300\delta t$. It is visible that over time, the entropy decreases in general. However, it does not decrease to 0, because still not all information is available at the end of a run. For individual runs, the entropy might increase at some point, for example, when at an unlikely position, a lot of hits are recorded.

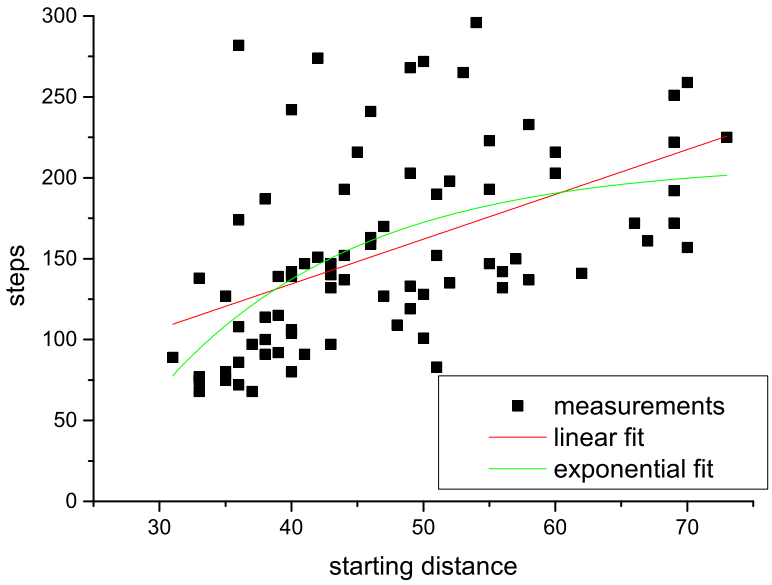


Figure 4.6: Relation between starting distance and search time - the linear fit is given by $y = a + bx$ with $a = 23.6$ and $b = 2.8$ and the exponential fit by $y = a + b\exp(cx)$ with $a = 210$, $b = -1.04 \cdot 10^3$ and $c = -0.0667$.

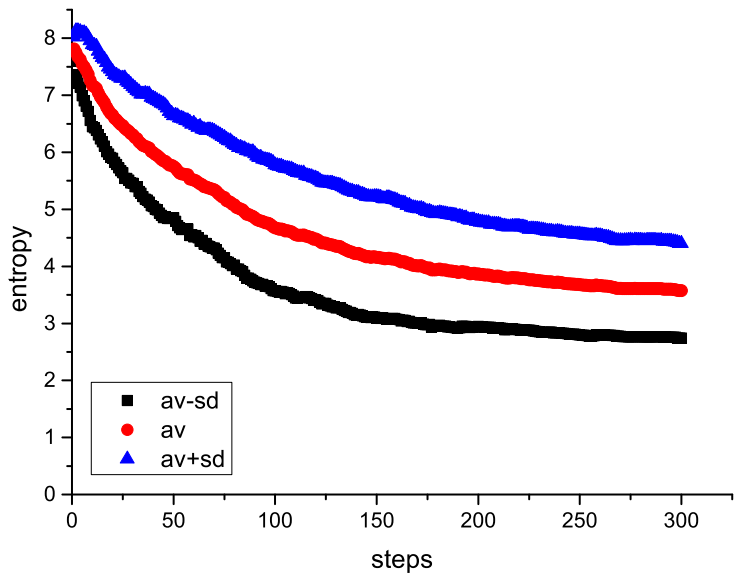


Figure 4.7: Relation between time and entropy - both the average and the average plus and minus one standard deviation are plotted against time, the calculations are based on 100 runs.

4.2 Results in 3D

The algorithm has been executed 50 times, with 42 successes. The 50 starting positions were uniformly distributed in the $[8, 12] \times [10, 20] \times [8, 12]$-grid. Also, 25 simulations were performed with $T_{max} = 750\delta t$ and starting positions uniformly distributed in the $[8, 12] \times [20, 30] \times [8, 12]$-grid, with 12 successes. This gave a success rate of 72%. Failures are due to running out of time or hitting the wall.

4.2.1 Trajectory

In Figure 4.8, one of the simulated trajectories is visualised. The color indicates time which increases from blue to red and the flow is directed in the positive y-direction. For these simulations, the hits were not saved, hence, they are not visualised in the Figure.

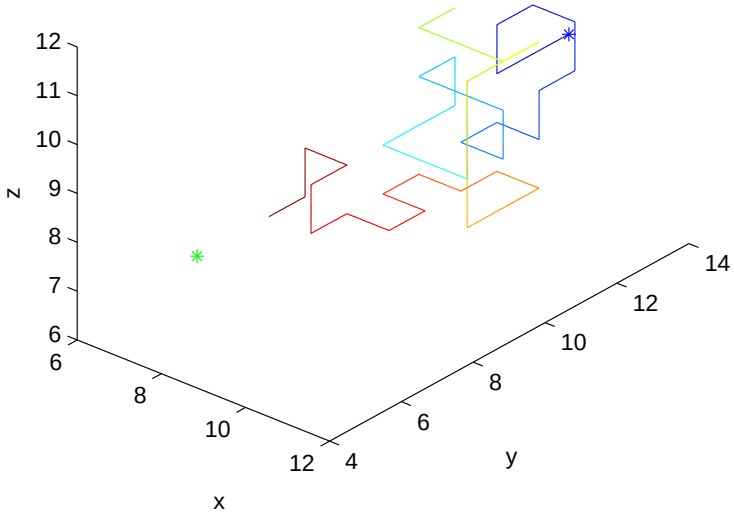


Figure 4.8: Search trajectory - the source is located at the green asterisk at (8,5,8) and the searcher starts at (10,13,10).

4.2.2 Relation between distance and search time

For the 2D case, the fits were given by $a = 23.6$ and $b = 2.8$ for the linear fit and by $a = 210$, $b = -1.04 \cdot 10^3$ and $c = -0.0667$ for the exponential fit. In Figure 4.9, the results and fits for the 3D case are depicted. Comparing the values for the linear fit reveals that the 3D case costs about 4 times more steps per increase of distance than the 2D case. The intercept values are comparable when keeping in mind the confidence intervals for the fits. For the exponential fits, the signs of the parameters agree, just as their orders of magnitude.

4.2.3 Relation between search time and entropy

The average entropy and its standard deviation are given in Figure 4.10. The averaging includes also the failed and ended runs, in which case the entropy is kept constant until $T = 500\delta t$. The profile is similar to the 2D case, but here the deviation is in general larger. This is partially caused by the lower number of simulations, but also by the increase in search area.

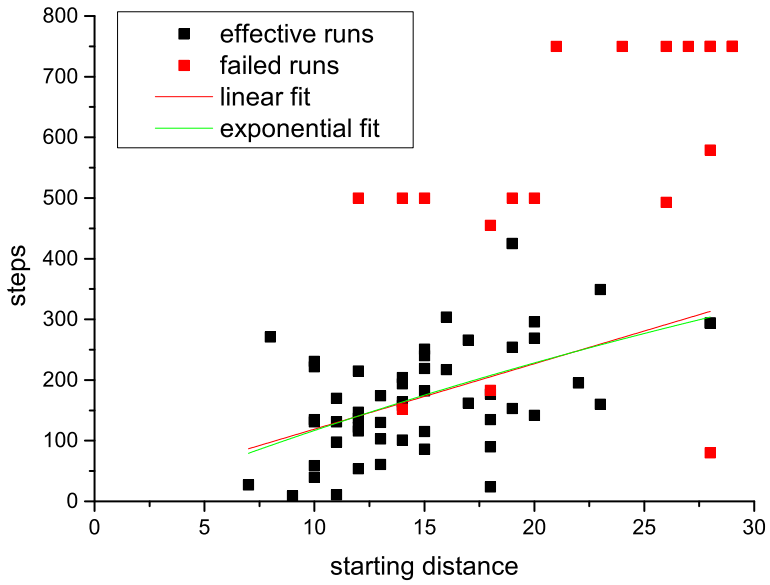


Figure 4.9: Relation between starting distance and search time, 3-dimensional - the linear fit is given by $y = a + bx$ with $a = 11.2$ and $b = 10.8$. The exponential fit is given by $y = a + b\exp(cx)$ with $a = 786$, $b = -802$ and $c = -0.0182$.

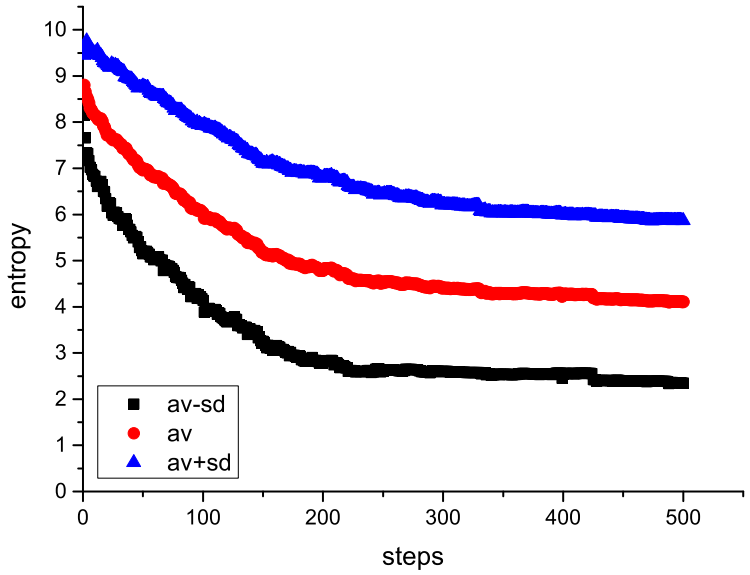


Figure 4.10: Relation between time and entropy - both the average and the average plus and minus one standard deviation are plotted against time. The calculations are based on 75 runs.

Part II

Continuum model

Chapter 5

Setup of the flow

We now want to apply the infotaxis algorithm to two flows given by the solution of the Navier-Stokes equations in a channel geometry, with a passive scalar released by a point source. This gives a situation in which no longer particles, but a high concentration will be detected.
Therefore, we will explain in this Chapter the setup and the discretisation of the two flows. In these cases, the equations, constants, parameters and variables are dimensionless.

5.1 Parameters

The algorithm will be executed for two setups, as given in the following text. The source will release its substance in a fully developed turbulent flow, with a strength $S_{max} = 1.0 \cdot 10^{-7}$. To do so, the velocity field will be simulated until the flow is fully developed at some time T_{dev}. This T_{dev} will be given in Sections 5.4.1 and 5.7.1. The dimensions of the channel are $X \times Y \times Z = 2\pi \times 1 \times \pi$ and the grid has size $nx \times ny \times nz = 64 \times 64 \times 32$ resp. $64 \times 128 \times 32$, in which x is the streamwise, y the normal and z the spanwise direction. The grid has a uniform spacing in x and z, while in the y-direction the grid points y_i, $i = 0, \ldots, ny/2$ are given by

$$y_j = \frac{\sinh(\gamma j/ny)}{2\sinh(\gamma/2)}, \quad \gamma = 6.5.$$

The pressure drop is in the x-direction with a given mass flow of ρYZU with a density ρ of 1. The spatial discretisation is fourth-order and the time step is $dt = 1.25 \cdot 10^{-3}$ resp. $5.00 \cdot 10^{-4}$.
In this way, the first situation is the same as in [9] and reference data for the characteristics of the flow are available.

5.2 Simulation of the flow

This section is based on an article written by Verstappen & Veldman [10]. The goal of their approach is to discretise the Navier-Stokes equations in such a way, that the difference operators have the same symmetry properties as their differential counterparts. This means that the convective operator should be skew-symmetric and the diffusive operator symmetric and positive-definite. In this way, the given discretisation is stable on any grid, and conserves the total mass, momentum and kinetic energy (when no physical dissipation occurs). Depending on the boundary conditions, physical dissipation might occur.
The dimensionless Navier-Stokes equations are given by

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} - \frac{1}{\mathrm{Re}} \nabla \cdot \nabla \mathbf{u} + \nabla p = \mathbf{0}, \quad \nabla \cdot \mathbf{u} = \mathbf{0}, \tag{5.1}$$

for an incompressible fluid, with $\mathbf{u}$ the velocity in three dimensions, Re the Reynolds number and p the pressure.

Furthermore, they compared their fourth-order method to physical experiments and previous numerical studies, to show that a $64 \times 64 \times 32$-grid on a $[0, 2\pi] \times [0, 1] \times [0, \pi]$-domain where the channel width is normalised, is sufficient to perform an accurate numerical simulation for $\mathrm{Re} = 5600$ based on the channel width and mean bulk velocity.

Their approach gives rise to the following algorithm for direct numerical simulation: at each time step, the momentum equations are integrated. After this, a correction is executed to satisfy the incompressibility constraint. Furthermore, $|\max \Delta v|$ is computed, just as the mass flow and the mean values. The velocities, pressure and kinetic energy are computed and written to file, after which a new time step starts.

5.2.1 Calculation of convective and diffusive fluxes

Notation

The velocities are defined on the grid lines and the pressure is defined in the centre of the cells. For an example, see Figure 5.1, as given in [10]. The k-component is directed out of the paper to the reader, with grid lines z_k.

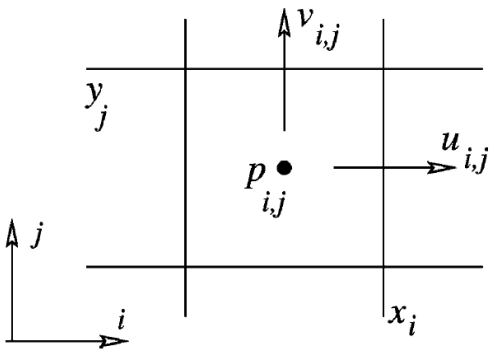


Figure 5.1: Definition of velocities and pressure related to the cells. (from [10])

Transport theorem

We will start with the transport theorem: for any function f of $\mathbf{x}$ and t, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega f \,\mathrm{d}V = \int_\Omega \frac{\partial f}{\partial t} \,\mathrm{d}V + \int_{\partial\Omega} f \mathbf{u} \cdot \mathbf{n} \,\mathrm{d}S, \tag{5.2}$$

where Ω is an arbitrary volume in the fluid at a specific time t. $\mathbf{u}$ denotes the velocity, which can vary over the boundary and $\mathbf{n}$ denotes the outward normal on the surface $\partial\Omega$ of Ω. When f is the mass density and the fluid is incompressible, then this theorem states that the net mass flux through the boundary of the control volume is zero. In terms of the velocities and the grid elements, this reduces for the control volume Ω_{ijk} to

$$\overline{u}_{ijk} + \overline{v}_{ijk} + \overline{w}_{ijk} - \overline{u}_{i-1jk} - \overline{v}_{ij-1k} - \overline{w}_{ijk-1} = 0 \tag{5.3}$$

where $\overline{u}_{ijk}$ is defined as

$$\overline{u}_{ijk} = \int_{y_{j-1}}^{y_j} \int_{z_{k-1}}^{z_k} u(x_i, y, z, t) \,\mathrm{d}y\mathrm{d}z, \tag{5.4}$$

and the other mass fluxes are defined analogously.

Convective discretisation

When f is one of the velocity components and the fluid is incompressible, then the transport theorem gives the conservation of momentum. Here, the calculation will be given for the u-component. To find $\frac{\mathrm{d}u_{ijk}}{\mathrm{d}t}$, the volume Ω is chosen differently, as depicted in Figure 5.2, also given in [10]. This because the velocities are defined at the cell faces, and now the velocity of concern is located in the centre of the

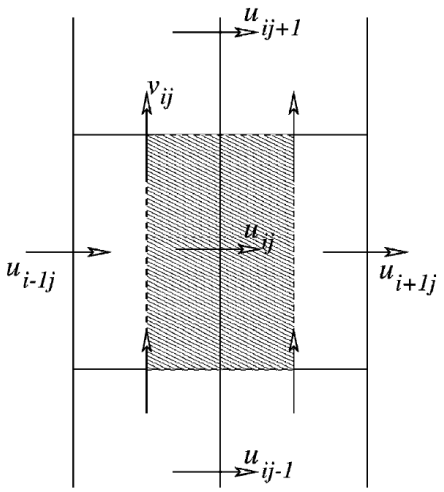


Figure 5.2: Definition of Ω for conservation of momentum in the x-direction. (from [10])

volume. Because mass and momentum are transported at the same velocity, the mass fluxes can be used to discretise the momentum fluxes. It is assumed that the velocities at the boundary are equal over the cell faces, and then the integration reduces to

$$\begin{aligned} |\Omega_{i+1/2jk}|\frac{\mathrm{d}u_{ijk}}{\mathrm{d}t} &= \overline{u}_{i+1/2jk}u_{i+1/2jk} + \overline{v}_{i+1/2jk}u_{ij+1/2k} + \overline{w}_{ijk+1/2}u_{ijk+1/2} \\ &\quad -\overline{u}_{i-1/2jk}u_{i-1/2jk} - \overline{v}_{i+1/2j-1k}u_{ij-1/2k} - \overline{w}_{i+1/2jk-1}u_{ijk-1/2}. \end{aligned} \tag{5.5}$$

The velocity at a cell face is approximated by the average of the velocity at both sides of the cell face:

$$u_{i+1/2jk} = \frac{u_{i+1jk} + u_{ijk}}{2}, \quad u_{ij+1/2k} = \frac{u_{ij+1k} + u_{ijk}}{2}, \quad u_{ijk+1/2} = \frac{u_{ijk+1} + u_{ijk}}{2} \tag{5.6}$$

When equations (5.3) and (5.6) hold, and the interpolation of mass fluxes to the cell faces is calculated in the following way:

$$\overline{u_{i+1/2jk}} = \frac{\overline{u_{i+1jk}} + \overline{u_{ijk}}}{2}, \quad \overline{v_{i+1/2jk}} = \frac{\overline{v_{i+1jk}} + \overline{v_{ijk}}}{2}, \quad \overline{w_{ijk+1/2}} = \frac{\overline{w_{ijk+1}} + \overline{w_{ijk}}}{2} \tag{5.7}$$

then the convective derivative is skew-symmetric. The only thing that remains, is the relation of the mass fluxes to the discrete velocities. This is done by means of the midpoint rule:

$$\overline{u_{ijk}} = (y_j - y_{j-1})(z_k - z_{k-1})u_{ijk}, \tag{5.8}$$

and analogously for the other mass fluxes.

Pressure gradient

By properties of the gradient operator ∇, any velocity field $\mathbf{u}$ and pressure p satisfy (using the $(\cdot,\cdot)$-notation for inner products)

$$(\nabla p, \mathbf{u}) = -\left(p, \nabla \cdot \mathbf{u}\right). \tag{5.9}$$

From this, the pressures are calculated.

Diffusion discretisation

The diffusive operator is viewed as the divergence of a gradient. The divergence is discretised and the discrete gradient is discretised as the transpose of the discretised divergence. This leads to a symmetric and positive-definite approximation of $-\nabla \cdot \nabla$. The diffusive flux through the cell faces of the control volume $\Omega_{i+1/2jk}$ of u_{ijk} reads

$$\overline{\phi_{i+1/2jk}} - \overline{\phi_{i-1/2jk}} + \overline{\psi_{ijk}} - \overline{\psi_{ij-1k}} + \overline{\chi_{ijk}} - \overline{\chi_{ijk-1}}, \tag{5.10}$$

with $\phi = \partial_x u/\mathrm{Re}$, $\psi = \partial_y u/\mathrm{Re}$ and $\chi = \partial_z u/\mathrm{Re}$. These surface integrals are then calculated by

$$\begin{aligned} \overline{\phi_{i+1/2jk}} &= (y_j - y_{j-1})(z_k - z_{k-1})\phi_{i+1/2jk}, \\ \overline{\psi_{ijk}} &= (x_{i+1/2} - x_{i-1/2})(z_k - z_{k-1})\psi_{ijk}, \\ \overline{\chi_{ijk}} &= (x_{i+1/2} - x_{i-1/2})(y_j - y_{j-1})\chi_{ijk}, \end{aligned} \tag{5.11}$$

and the others are calculated analogously. The only issue left is to calculate ϕ, ψ and χ at the cell faces. This gives

$$\phi_{i+1/2jk} = \frac{1}{\mathrm{Re}} \frac{u_{i+1jk} - u_{ijk}}{x_{i+1} - x_i}, \quad \psi_{ijk} = \frac{1}{\mathrm{Re}} \frac{u_{ij+1k} - u_{ijk}}{y_{j+1/2} - y_{j-1/2}}, \quad \chi_{ijk} = \frac{1}{\mathrm{Re}} \frac{u_{ij+1k} - u_{ijk}}{z_{k+1/2} - z_{k-1/2}}. \tag{5.12}$$

In the same fashion, the diffusive flux through the v_{ijk} and w_{ijk} volumes can be calculated.

5.2.2 Higher-order, symmetry-preserving approximation

To turn this second-order accurate approximation into a fourth-order approximation, the region $\Omega^{(3)}_{i+1/2jk} = [x_{i-3/2}, x_{i+3/2}] \times [y_{j-2}, y_{j+1}] \times [z_{k-2}, z_{k+1}]$ is introduced. On this volume, the fluxes and velocities are interpolated in the same way as before. To eliminate the leading term of the discretisation error in the continuity equation, the approximations are added with weights 3^5 resp. -1. These weights are also to be used on nonuniform grids.
For the momentum equations, the same approach is used. As a matter of notation, a double overline indicates a flux over the larger volume. To keep the skew-symmetry, it is necessary to interpolate the fluxes in the following way:

$$\overline{u_{i+1/2jk}} = \frac{9}{16}\left(\overline{u_{i+1jk}} + \overline{u_{ijk}}\right) - \frac{1}{16}\left(\overline{u_{i+2jk}} + \overline{u_{i-1jk}}\right) \tag{5.13}$$

and $\overline{v}_{i+1/2,j,k}$, $\overline{w}_{i+1/2,j,k}$, $\overline{\overline{u_{i+1/2jk}}}$, $\overline{\overline{v_{i+1/2jk}}}$, and $\overline{\overline{w_{i+1/2jk}}}$ analogously.
The same holds for the v_{ijk} and w_{ijk} volumes.
This approximation is fourth-order, symmetry-preserving, stable and conserves mass, momentum and energy.

5.2.3 Time integration

The time integration for the velocities is calculated by the one-leg scheme. This scheme is given by

$$(\beta + 1/2)\,\mathbf{u}^{n+1} - 2\beta\mathbf{u}^n + (\beta - 1/2)\,\mathbf{u}^{n-1} = \Delta t \cdot \mathbf{f}((1+\beta)\mathbf{u}^n - \beta\mathbf{u}^{n-1}) \tag{5.14}$$

for the problem

$$\partial_t \mathbf{u} = \mathbf{f}(\mathbf{u}). \tag{5.15}$$

The convective stability condition is given by $U\Delta t < \Delta x$, $V\Delta t < \Delta y$ and $W\Delta t < \Delta z$ with U, V and W the absolute maximum velocity in respectively the x-, y- and z-direction, which implies that $\Delta t < \max\left\{\frac{\Delta x}{U}, \frac{\Delta y}{V}, \frac{\Delta z}{W}\right\}$. The diffusive stability condition is given by $2\Delta t < \mathrm{Re}(\Delta x)^2$ etc., which implies $\Delta t < \max\left\{\frac{1}{2}\mathrm{Re}(\Delta x)^2, \frac{1}{2}\mathrm{Re}(\Delta y)^2, \frac{1}{2}\mathrm{Re}(\Delta z)^2\right\}$. It will turn out later on that the convective stability condition is much more restrictive than the diffusive stability condition, which will give eigenvalues of (5.15) as $\lambda = x_1 + ix_2$ with $x_2 >> x_1$. The stability domain for β will be pressed against the imaginary axis when β goes to zero. Hence, a small value of β will perform best, because then the eigenvalues of the PDE are covered in the stability domain. We will choose $\beta = 0.05$.

5.3 Modelling and simulation of the scalar

5.3.1 Modelling of the point source

The point source is modelled by several smaller point sources to improve the numerical stability. Suppose the total strength of the sources combined is given by S. Then the central point source is 4 times stronger than its direct neighbours, and these are twice as strong as the indirect neighbours. Considered as a cube of 27 blocks, the corners have no point source, the blocks in the middle of an edge have relative strength 1, the blocks in the middle of the outside planes have relative strength 2 and the block in the centre has relative strength 8. Adding these relative strengths adds up to 52, hence the absolute strengths can be calculated. The values are based on quadratic decay, and the source in the centre has no distance to the centre, hence its value is undetermined and can be chosen freely.

5.3.2 Numerical integration of the scalar equation

We use the non-dimensionalised advection-diffusion equation:

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = \frac{1}{\mathrm{Pe}} \nabla^2 C, \tag{5.16}$$

in which C is the concentration of the scalar and Pe the Péclet number given by $\mathrm{Pe} = \mathrm{Re} \cdot \mathrm{Sc}$ with Sc the Schmidt number which is the ratio of kinematic viscosity ν to mass diffusivity D.
At first, forward Euler is tried. This does not work out, because this restricts the space discretisation too much. Backward Euler should always give a stable result, but results in having to solve a very large system at each time step, which would take too much time. Modified Euler, also does not work. The answer is in the Péclet number. When $\mathrm{Pe} > 2$, then a central difference scheme causes oscillations, as described by Versteeg & Malalasekera [11]. This could be overcome by calculating the fluxes locally by an upwind discretisation scheme, for example as given by [12]. At each side of the integration volumes, it is calculated if the velocity is inward or outward, and the discretisation adapts to that. Here, the flux at the right boundary of a grid cell at position (i, j, k) with velocity (u, v, w) is calculated in the following way:

$$\text{if } u_{i,j,k} \geq 0 \text{ then } f = \left(\frac{1}{\mathrm{Pe}(x_{i+1/2} - x_{i-1/2})} \left(C_{i+1,j,k} - C_{i,j,k}\right) - u_{i,j,k} \cdot C_{i,j,k} \right) \cdot (y_j - y_{j-1}) \cdot (z_k - z_{k-1})$$
$$\text{else } f = \left(\frac{1}{\mathrm{Pe}(x_{i+1/2} - x_{i-1/2})} \left(C_{i+1,j,k} - C_{i,j,k}\right) - u_{i,j,k} \cdot C_{i+1,j,k} \right) \cdot (y_j - y_{j-1}) \cdot (z_k - z_{k-1}).$$

For the upper and top boundary, the calculation is analogous, for the left, lower and bottom boundary, the flux has a minus sign. Time integration is performed by calculating the fluxes F using the previous velocities ($F^{n+1} = f(u^n)$) and $C^{n+1}_{i,j,k} = C^n_{i,j,k} + dt \cdot F^{n+1}_{i,j,k}$.
Boundary conditions are taken as follows: at $y = y_{min} = 0$ and $y = y_{max} = 1$, $v_{i,0,k}$ and $v_{i,ny,k} = 0$, hence Neumann boundary conditions reduce to $C_{i,0,k} = C_{i,1,k}$ and $C_{i,ny+1,k} = C_{i,ny,k}$. At $z = z_{min} = 0$, the flux through the wall should be equal to zero, hence

$$\text{if } w_{i,j,0} \geq 0 \text{ then } C_{i,j,0} = C_{i,j,1} \frac{-1}{\mathrm{Pe}(z_{1/2} - z_{-1/2}) \left(\frac{-1}{\mathrm{Pe}(z_{1/2} - z_{-1/2})} - w_{i,j,0} \right)} = C_{i,j,1} \frac{1}{1 + \mathrm{Pe}(z_{1/2} - z_{-1/2}) w_{i,j,0}}$$
$$\text{else } C_{i,j,0} = (1 - \mathrm{Pe}(z_{1/2} - z_{-1/2}) w_{i,j,0}) C_{i,j,1}$$

Similar for $z = z_{max} = \pi$:

$$\text{if } w_{i,j,nz} \geq 0 \text{ then } C_{i,j,nz+1} = C_{i,j,nz} \left(1 + \mathrm{Pe}(z_{nz+1/2} - z_{nz-1/2}) w_{i,j,nz}\right)$$
$$\text{else } C_{i,j,nz+1} = C_{i,j,nz} \frac{1}{\mathrm{Pe}(z_{1/2} - z_{-1/2}) \left(\frac{1}{\mathrm{Pe}(z_{1/2} - z_{-1/2})} - w_{i,j,nz} \right)} = C_{i,j,nz} \frac{1}{1 - \mathrm{Pe}(z_{1/2} - z_{-1/2}) w_{i,j,nz}}$$

For $x = x_{min} = 0$, the boundary condition is similar to $z = z_{min} = 0$, but this will not be repeated. For $x = x_{max} = 2\pi$, $C_{nx+1,j,k} = 0$.

5.4 Mean values for Re = 5600

5.4.1 Flow

In Figure 5.3, the mean streamwise velocity and its fluctuations can be found on basis of which the averaging time of $T = 1500\ \delta t$ is chosen by [10]. The equations are made dimensionless by the channel width and the average streamwise velocity. Also, for the situation with $T_{max} = 1500\delta t$, another 1500 time units have been simulated and averaged, to get rid of starting effects. These averages have been compared to data publicly available from [9] in Figure 5.4. The velocities are scaled by the shear velocity which is given by $u_\tau = \sqrt{\frac{\tau}{\rho}}$ with τ the shear stress and calculated as $u_\tau = \sqrt{\frac{\mu \Delta V}{\rho \Delta y}}$ for a Δy such that the velocity increase is still linear, with μ the dynamic viscosity and ρ the density. This leads to the Re_τ number 180 based on the shear velocity and the channel half-width; calculated as $\mathrm{Re}_\tau = \frac{u_\tau \delta}{\nu}$. The agreement between the two results is good, except for the spanwise root mean square velocity fluctuations. This might come from inappropriately chosen initial conditions: after inspection, the initial conditions, chosen to increase the generation of turbulence, give a very small flux upward of 0.002 in the spanwise direction, compared to π in the streamwise direction. However, this could explain the difference in the velocity profile in Figure 5.4d.

5.4.2 Scalar

The scalar will not be released continuously, but periodically, to avoid one plume being detected (almost) everywhere. This will also decrease the number of hits, which will increase the independence of hits. The function for this is given by

$$S(t) = S_{max} \cdot \max\{0, \sin(\pi t)\} \tag{5.17}$$

5.4.3 Diffusion

In order to find the turbulent diffusion for the scalar, we use the gradient-diffusion hypothesis as decribed in [13] which says that $\Gamma_T(\mathbf{x}, t)$ is given by

$$\langle v'C' \rangle = -\Gamma_T \frac{\partial \langle C \rangle}{\partial y}, \tag{5.18}$$

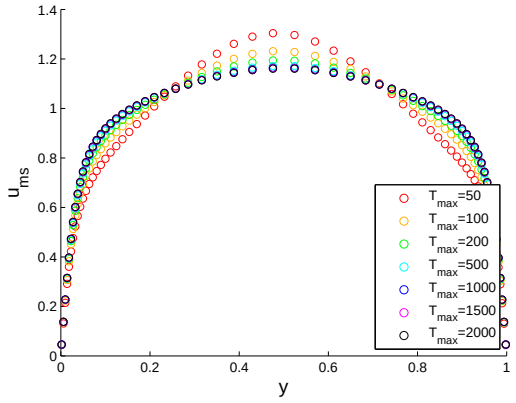
in the case of two-dimensional turbulent-boundary-layer equations. In this case, the mean velocity is predominantly in the x-direction, and variations in the mean quantities are predominantly in the y-direction. When Γ_T is positive, this description can be taken as the definition of Γ_T. In our case, the equations are three-dimensional, but it still holds that the mean velocity is predominantly in the x-direction and variations predominantly in the y-direction. For this, the averages are computed over x, z and t. Hence, we will find several values for Γ_T, which we will average again. To find $\langle v'C' \rangle$, the following is computed:

$$\langle v'C' \rangle = \langle (v-\bar{v})(C-\bar{C}) \rangle = \langle vC - v\bar{C} - \bar{v}C + \bar{v}\bar{C} \rangle = \langle vC \rangle - \langle v\bar{C} \rangle - \langle \bar{v}C \rangle + \langle \bar{v}\bar{C} \rangle = \langle vC \rangle - \bar{C}\langle v \rangle - \bar{v}\langle C \rangle + \bar{v}\bar{C} = \langle vC \rangle + \bar{v}\bar{C}$$

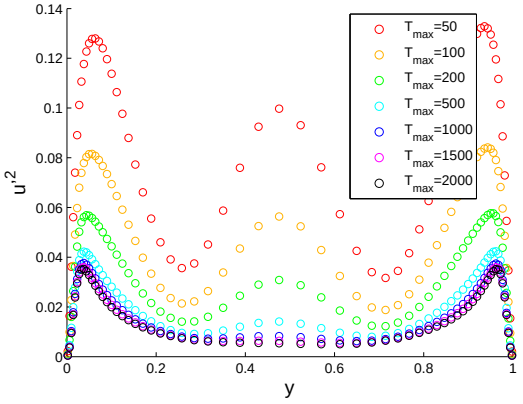
By computing $\langle vC \rangle$, $\bar{v}$ and $\bar{C}$, $\langle v'C' \rangle$ can be computed.
The calculated values for Γ_T are visible in Figure 5.5, from which it is visible that the turbulent diffusion coefficient is lower than expected. This is for a continuous point source of strength $S = 1.0 \cdot 10^{-7}$ at position $(5, 28, 12)$ with $\mathrm{Sc} = 0.57$. This continuous source is chosen to avoid the effects from the periodic source. The source position $psy = 28$ corresponds to $y = 0.3163$. However, this is not the position where the turbulent diffusion coefficient is at a maximum. This is because the turbulent diffusion coefficient increases when gradients in the concentration occur, and these are small near the y-coordinate of the source.

5.5 Implications

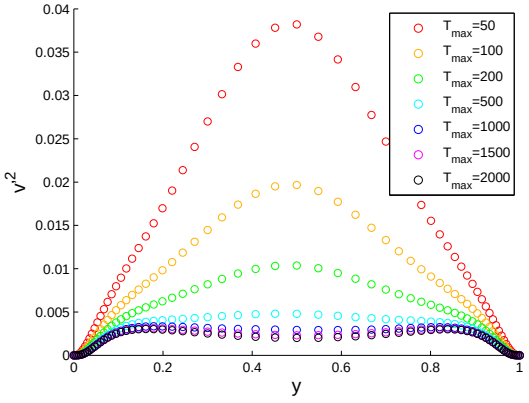
This result gives that, by the Schmidt number smaller than 1, almost no diffusion occurs. Because of the numerical method, it is not feasible to increase the Schmidt number to large values, e.g. 1000. To improve the diffusion a little bit, we change the Schmidt number to 1.0, which gives a Peclet number Pe of 5600. This value of the Schmidt number also implies that the hits will not be independent and thereby, one of the foundations of the algorithm is violated. Hence, a solution for this should be found.



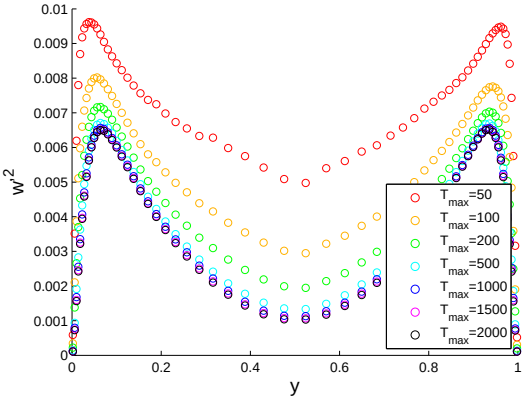
(a) Mean streamwise velocity



(b) Streamwise mean square velocity fluctuations

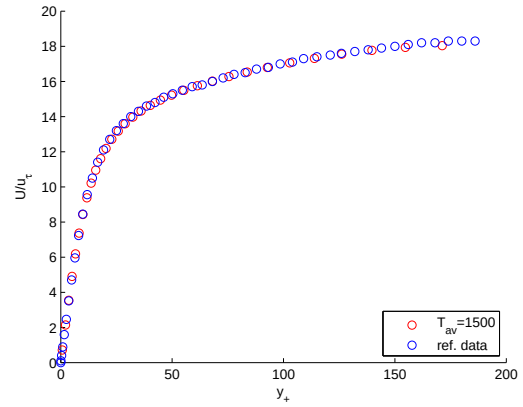


(c) Wallnormal mean square velocity fluctuations

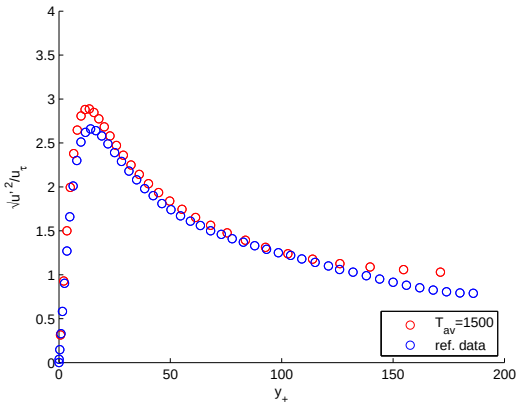


(d) Spanwise mean square velocity fluctuations

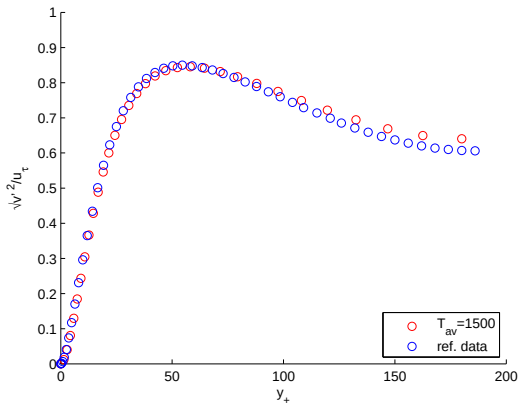
Figure 5.3: Characteristics of the flow for Re = 5600.



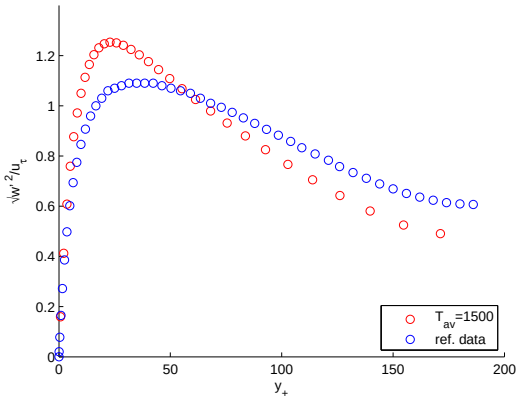
(a) Mean streamwise velocity



(b) Streamwise root mean square velocity fluctuations



(c) Wallnormal root mean square velocity fluctuations



(d) Spanwise root mean square velocity fluctuations

Figure 5.4: Comparison of the flow characteristics - they are compared with data from [9]. The flow characteristics are obtained by averaging 1500 time units after simulating another 1500 time units.

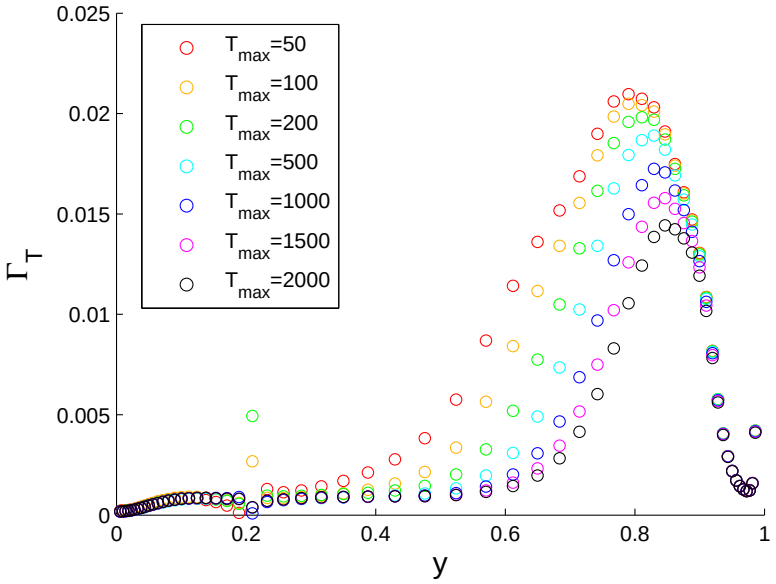


Figure 5.5: Estimation of the turbulent diffusion coefficient Γ_T - the values are only valid for positive values, hence the y-axis is adapted.

The solution which will be further deployed, is a change in the function which calculates the expected number of hits. For this, the Gaussian plume model will be used. Also, the registration of hits will be adapted.

5.6 Gaussian plume model

In the Gaussian plume model, the concentration profile is given by

$$C(\mathbf{r}|\mathbf{r}_0) = C(x, y, z, x_0, y_0, z_0) = \frac{S}{2\pi U \sigma_y \sigma_z} \exp\left(\frac{-(y-y_0)^2}{2\sigma_y^2} + \frac{-(z-z_0)^2}{2\sigma_z^2}\right), \tag{5.19}$$

with S the strength of the source and σ_y and σ_z to be determined. These two parameters also contain the x-dependence of the model. For this, we choose the following expression:

$$\sigma_y = \min\left\{\sqrt{\overline{v'^2}}\frac{(x-x_0)}{U}, \alpha \cdot D_y\right\}, \tag{5.20}$$

and analogous for σ_z, in which α is a shape parameter and D_y the diameter of the channel in the y-direction. Also, when $(x - x_0) < 0$, then $C = 0$.
To calculate $R(\mathbf{r}|\mathbf{r}_0)$, $C(\mathbf{r}|\mathbf{r}_0)$ needs to be multiplied by a^2, which is here the squared diameter of the searcher and by Q, which is a scaling parameter which includes also the strength of the source and the threshold value of the concentration, hence

$$R(\mathbf{r}|\mathbf{r}_0) = \begin{cases} \frac{a^2 Q}{2\pi U \sigma_y \sigma_z} \exp\left(\frac{-(y-y_0)^2}{2\sigma_y^2} + \frac{-(z-z_0)^2}{2\sigma_z^2}\right), & (x-x_0) > 0, \\ 0, & (x-x_0) \leq 0. \end{cases} \tag{5.21}$$

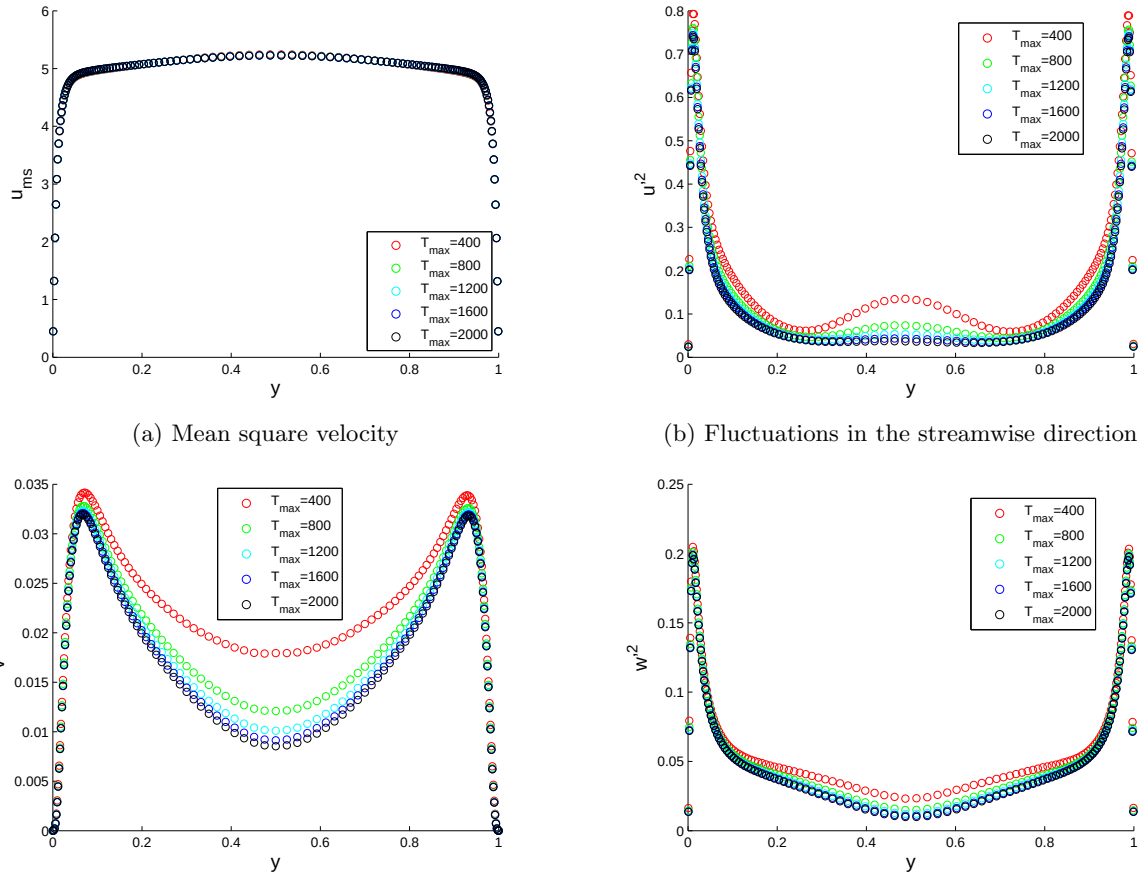
This gives the unfortunate result of $R(\mathbf{r}|\mathbf{r}_0) = 0$, but this is an undefined point.

5.7 Mean values for Re = 28000

When the Reynolds number is increased, the characteristics of the flow will change. Hence, the characteristics of the flow and the scalar will be shown in the following Sections.

5.7.1 Flow

In Figure 5.6, the mean streamwise velocity and its fluctuations can be seen. The average velocity has increased by a factor 5. This results in a flow with larger velocity gradients at the walls, which should increase the turbulence. For this Reynolds number and the corresponding Re_τ number of 630, calculated at the gridpoint at $y = 0.0131$, no reference data is available.



(a) Mean square velocity

(b) Fluctuations in the streamwise direction

(c) Fluctuations in the normal direction

(d) Fluctuations in the spanwise direction

Figure 5.6: Characteristics of the flow for Re = 28000.

5.7.2 Scalar

The scalar will not be released continuously, but periodically, to avoid one plume being detected (almost) everywhere. This will also decrease the number of hits, which will increase the independence of hits. The function for this is given by

$$S(t) = S_{max} \cdot \max\{0, \sin(5\pi t)\} \tag{5.22}$$

5.7.3 Diffusion

In the same way as before, Γ_T is calculated. The calculated values for Γ_T are visible in Figure 5.7, from which is visible that the turbulent diffusion coefficient is again very low. This is for the periodic point source given in the previous subsection of strength $S = 1.0 \cdot 10^{-7}$ at position $(5, 56, 14)$ with Sc = 1.00.

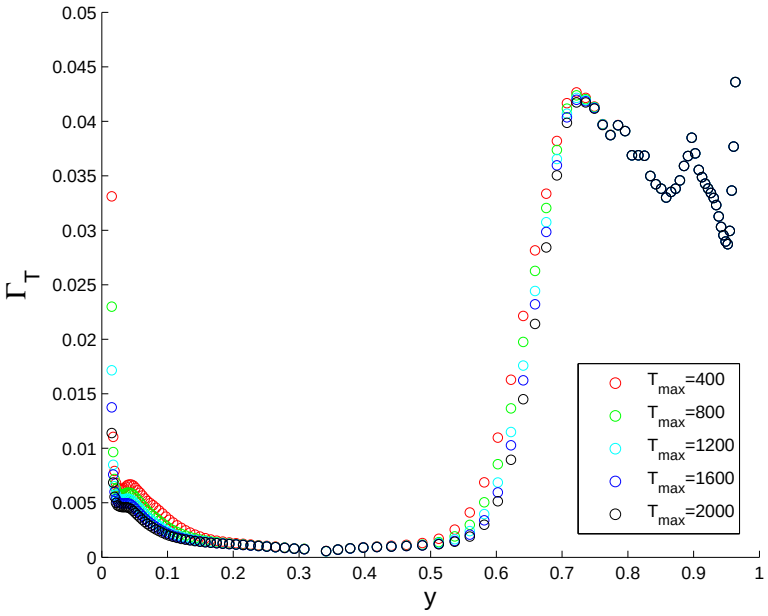


Figure 5.7: Estimation of the turbulent diffusion coefficient Γ_T - the values are only valid for positive values, hence the y-axis is adapted.

Chapter 6

Setup of the algorithm

6.1 Parameters

The Reynolds number is 5600 resp. 28000 and the Schmidt number is 1.00, which leads to a Péclet number Pe of 5600 resp. 28000. The source of scalar is set at $(psx, psy, psz) = (5, 28, 14)$ resp. $(5, 56, 14)$. The searcher always starts at a random position L in the grid bounded by $L \in \{[15, 35] \times [1, 9] \times [8, 24]\}$. The experiments are executed after each other, and the searcher starts for $\mathrm{Re} = 5600$ at $T_{dev} = 1500$ and for $\mathrm{Re} = 28000$ at $T_{dev} = 2000$ such that the flow is developed. The parameters for the search algorithm are as follows: the waiting time W is equal to 1.0 δt, hence to 800 resp. 2000 dt time steps.
The diameter of the searcher a is $2\pi/64$, such that it equals the grid size in the x- and z-direction. For the y-direction, the algorithm is adapted such that it also looks at neighbouring cells if necessary, which gives a different y-grid with $nyp = 10$. The mean velocity in the flow direction V equals 1.0 resp. 5.0. The threshold value of the searcher for the concentration is calculated to be $10^{-10.1} \approx 7.9 \cdot 10^{-11}$ resp. $10^{-10.75} \approx 1.7 \cdot 10^{-11}$ in Sections 6.2.2 and 6.3.2.

6.2 $\mathbf{Re} = 5600$

First, the parameters for the search algorithm are calculated, using the characteristics of the flow. Of course, it is better if these parameters are estimated during the execution of the algorithm, but to test whether the algorithm works properly, we estimate them beforehand.

6.2.1 Calculation of $\sqrt{\overline{v'^2}}$ and $\sqrt{\overline{w'^2}}$

First, the flow is simulated after $T = 1500$ with $\mathrm{Sc} = 1.00$ until $T = 1550$ and $T = 1600$. Now, $\sqrt{\overline{v'^2}}$ is integrated from $y = 0$ to $y = 1$ by means of a Riemann sum, which gives $\sqrt{\overline{v'^2}} = 0.0471$ resp. 0.0473 for $T = 1550$ and $T = 1600$ and similar for $\sqrt{\overline{w'^2}}$, which gives $\sqrt{\overline{w'^2}} = 0.0539$ for both $T = 1550$ and $T = 1600$. Hence, $\sqrt{\overline{v'^2}}$ is chosen to be 0.0472.

6.2.2 Estimation of α and hit registration level

For $T = 1600$, Figures 6.1, 6.2 and 6.3 are made. Because the search area is three-dimensional, these Figures display slices at a certain x- or z-coordinate in the grid, denoted by x_i with $i = 1, \ldots, 64$ or z_k with $k = 1, \ldots, 32$. From now on, green/yellow-colored Figures concern data, in which green indicates low and yellow indicates high concentration. Rainbow-colored Figures concern estimates, in which blue indicates low and red indicates high detection rates. In the algorithm, the scalar will be cut off at $10^{-10.1} \approx 7.9 \cdot 10^{-11}$, hence this is the special value for the $\log_{10}$-scale. In Figures 6.4 up to and including 6.6, the Figures containing the logarithms of the scalar are compared with the expected rates for $\alpha = 0.2$, 0.1 and 0.05 (also on a $\log_{10}$-scale) with scaling parameter $Q = 1$ as defined for (5.21). From the Figures, we choose α to be 0.2.
We can also give a prescription for hit registration based on these Figures: *A hit is registered, if in more than half of the staying time, a signal is registered.* This is, because the mean velocity is 1.0, and the width of the detectable blobs varies from 2 to 1 non-dimensionalised length units. Hence, a blob is registered almost always between $x = 4$ and $x = 5$ and gives even 2 hits for some cases when the searcher is in between $x = 1$ and $x = 3$.

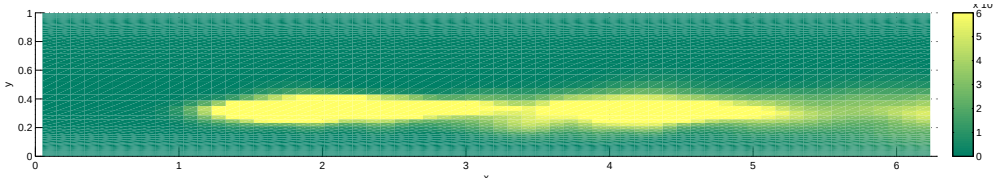


Figure 6.1: Scalar field at aspect ratio 1:1 for $T = 1600$ at z_{14}.

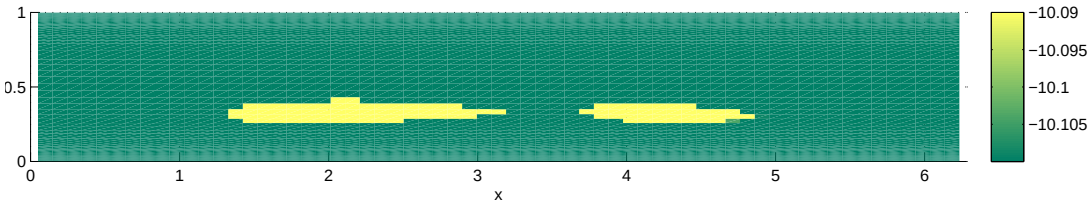
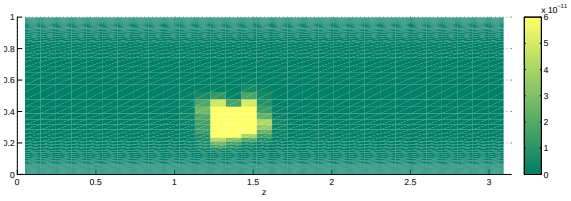
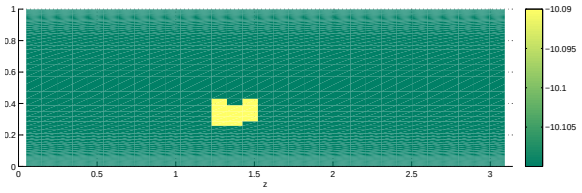


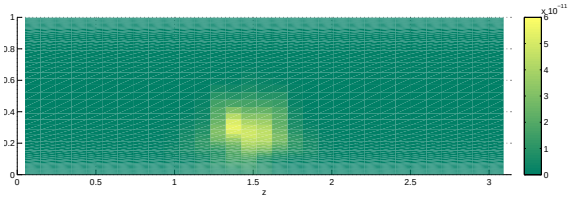
Figure 6.2: Scalar field, $\log_{10}$-scale, at aspect ratio 1:1 for $T = 1600$ at z_{14}.



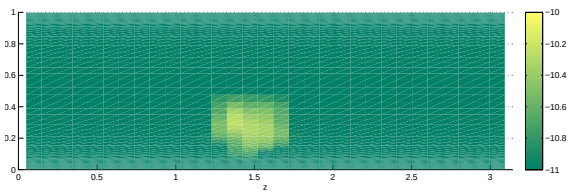
(a) Scalar field at x_{20}



(b) Scalar field, $\log_{10}$-scale at x_{20}

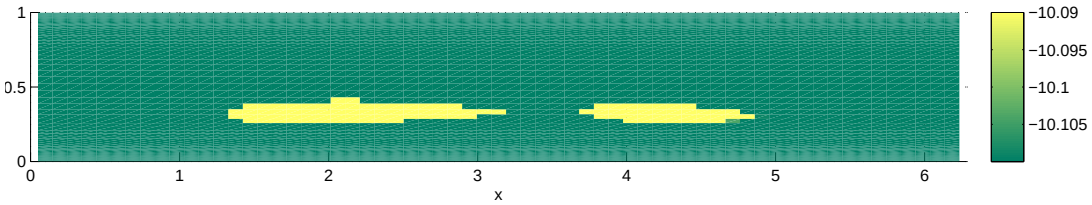


(c) Scalar field at x_{64}

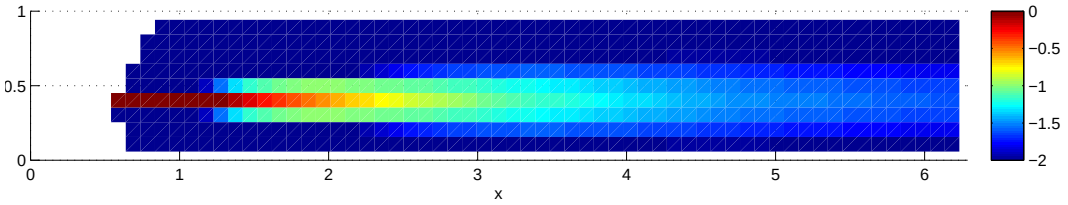


(d) Scalar field, $\log_{10}$-scale at x_{64}

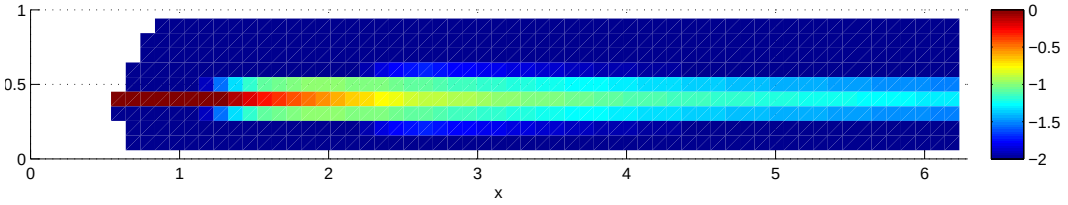
Figure 6.3: Plot of the characteristics of the scalar for $T = 1600$.



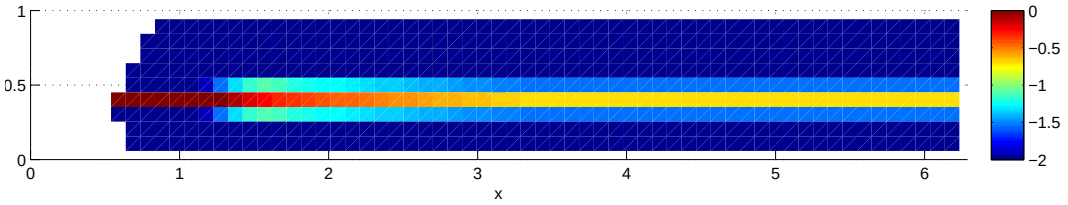
(a) Scalar field, $\log_{10}$-scale



(b) Expected rate of encounters for $\alpha = 0.2$

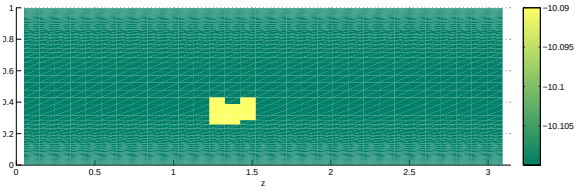


(c) Expected rate of encounters for $\alpha = 0.1$

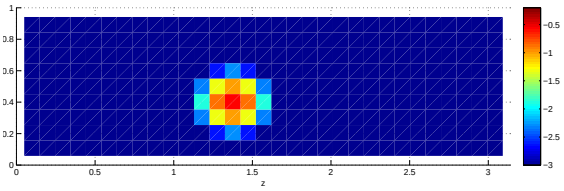


(d) Expected rate of encounters for $\alpha = 0.05$

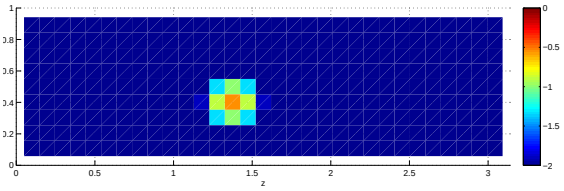
Figure 6.4: Comparison of measured data with expected rate for several values of α at z_{14}.



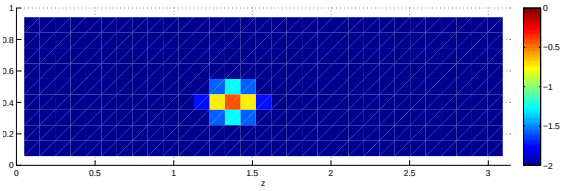
(a) Scalar field, $\log_{10}$-scale



(b) Expected rate of encounters for $\alpha = 0.2$

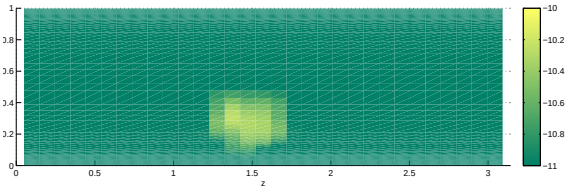


(c) Expected rate of encounters for $\alpha = 0.1$

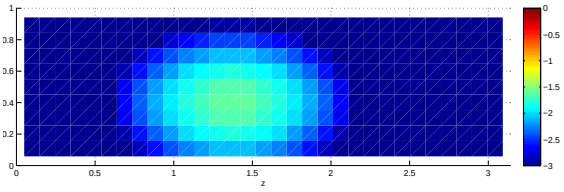


(d) Expected rate of encounters for $\alpha = 0.05$

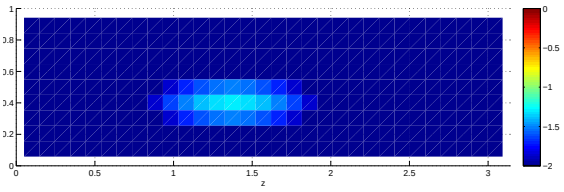
Figure 6.5: Comparison of measured data with expected rate for several values of α at x_{20}.



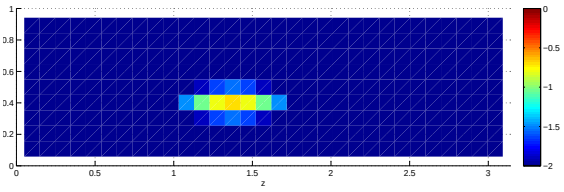
(a) Scalar field, $\log_{10}$-scale



(b) Expected rate of encounters for $\alpha = 0.2$



(c) Expected rate of encounters for $\alpha = 0.1$



(d) Expected rate of encounters for $\alpha = 0.05$

Figure 6.6: Comparison of measured data with expected rate for several values of α at x_{64}.

6.3 Re = 28000

6.3.1 Calculation of $\sqrt{\overline{v'^2}}$ and $\sqrt{\overline{w'^2}}$

From the simulations of the flow until $T_{dev} = 2000$ with Sc = 1.00, we can find the values of $\sqrt{\overline{v'^2}}$ and $\sqrt{\overline{w'^2}}$. For several values of T_{max}, we have calculated these and plotted in Figure 6.7.

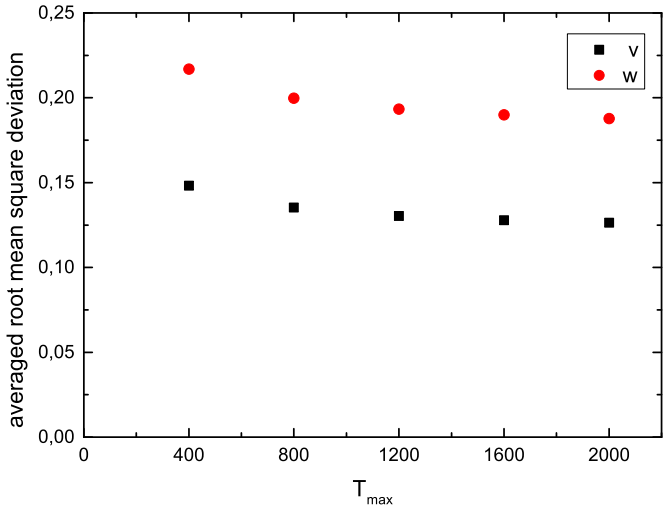


Figure 6.7: $\sqrt{\overline{v'^2}}$ and $\sqrt{\overline{w'^2}}$ calculated for different values of T_{max}.

From this, we see that the values are decreasing, hence we choose the lowest value for our algorithm. This gives $\sqrt{\overline{v'^2}} = 0.1264$ and $\sqrt{\overline{w'^2}} = 0.1878$. This is a factor 3 larger as for Re = 5600.

6.3.2 Estimation of α and hit registration level

For $T = 2000$, Figures 6.8, 6.9 and 6.10 are made. In the algorithm, the scalar will be cut off at $10^{-10.75} \approx 1.7 \cdot 10^{-11}$, hence this is the special value for the $\log_{10}$-scale. Now, the Figures containing the logarithms of the scalar are compared with the expected rates for $\alpha = 0.1$, 0.05 and 0.01 (also on a $\log_{10}$-scale), while we have again set $Q = 1.0$. From these Figures, α is chosen to be 0.01. We can now also give a new prescription for hit registration:
A hit is registered, if in more than 35% of the staying time, a signal is registered.
This is because the mean velocity is 5.0, and the width of the detectable blobs varies around 2 length units, hence, a blob passes the searcher in around 0.4 time units.

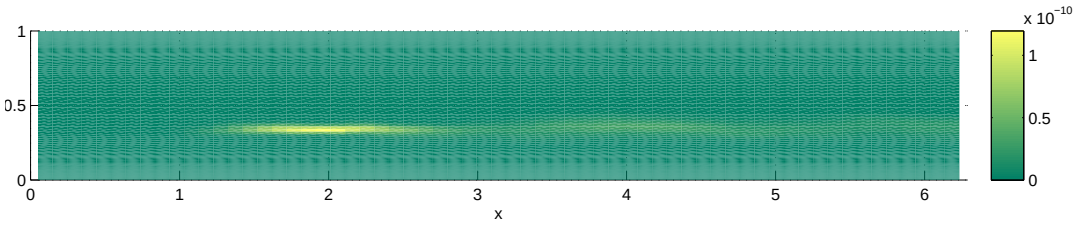


Figure 6.8: Scalar field at aspect ratio 1:1 for $T = 2000$ at z_{14}.

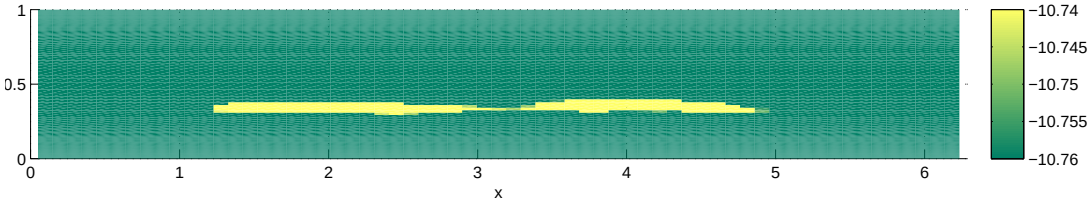
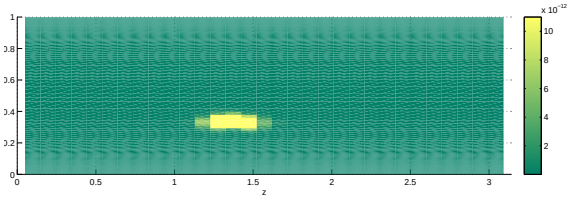
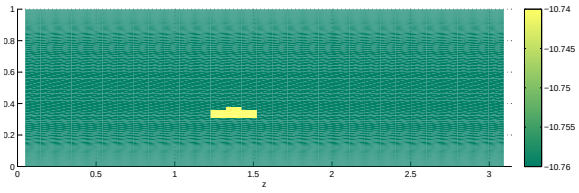


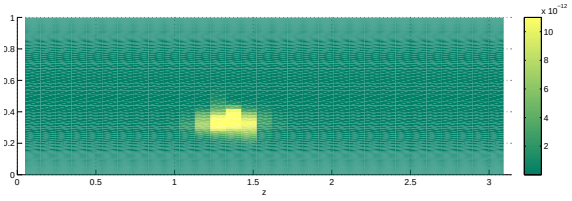
Figure 6.9: Scalar field, $\log_{10}$-scale, at aspect ratio 1:1 for $T = 2000$ at z_{14}.



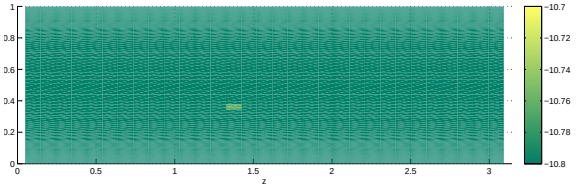
(a) Scalar field at x_{20}



(b) Scalar field, $\log_{10}$-scale at x_{20}

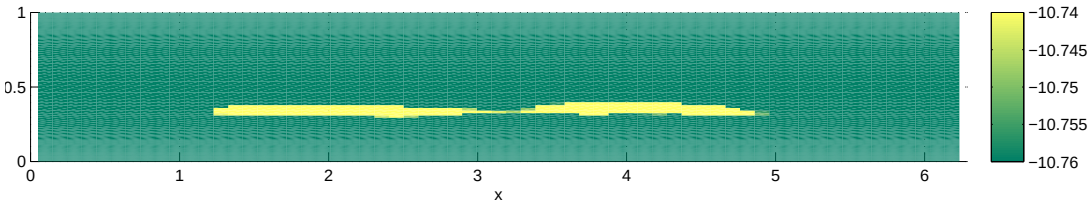


(c) Scalar field at x_{64}

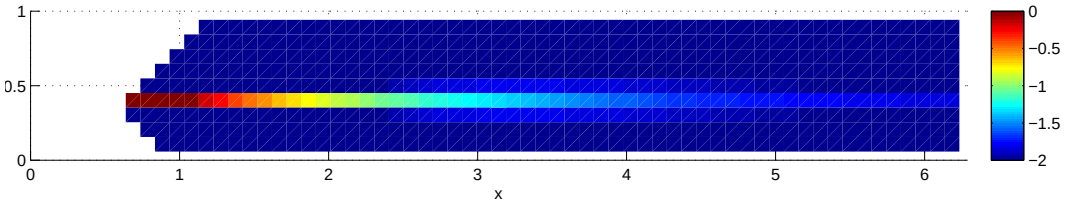


(d) Scalar field, $\log_{10}$-scale at x_{64}

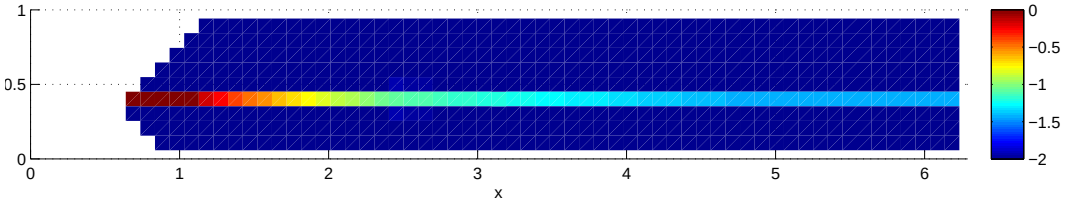
Figure 6.10: Plot of the characteristics of the scalar for $T = 2000$.



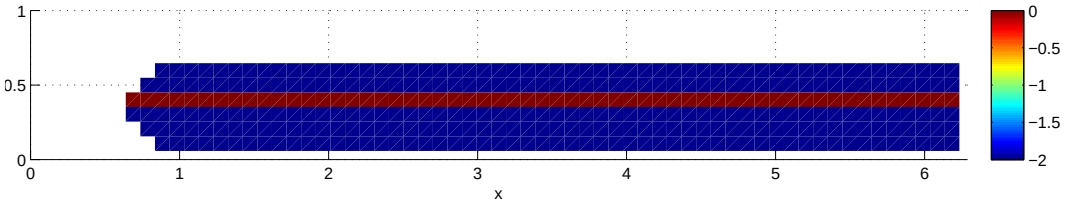
(a) Scalar field, $\log_{10}$-scale



(b) Expected rate of encounters for $\alpha = 0.1$

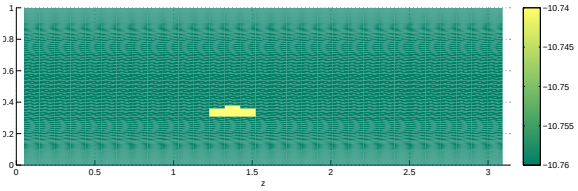


(c) Expected rate of encounters for $\alpha = 0.05$

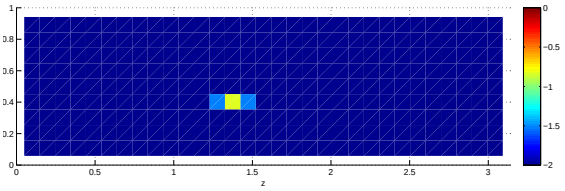


(d) Expected rate of encounters for $\alpha = 0.01$

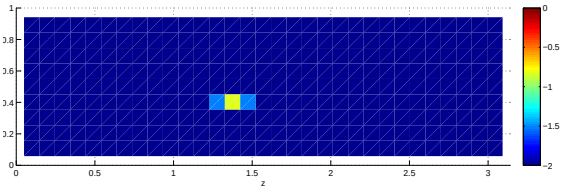
Figure 6.11: Comparison of measured data with expected rate for several values of α at z_{14}.



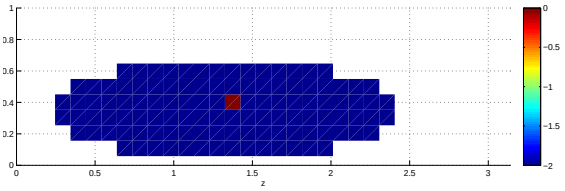
(a) Scalar field, $\log_{10}$-scale



(b) Expected rate of encounters for $\alpha = 0.1$

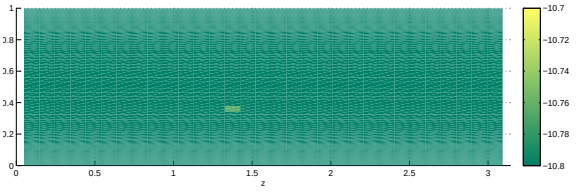


(c) Expected rate of encounters for $\alpha = 0.05$

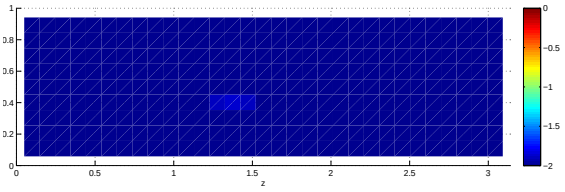


(d) Expected rate of encounters for $\alpha = 0.01$

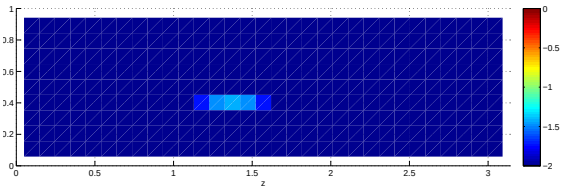
Figure 6.12: Comparison of measured data with expected rate for several values of α at x_{20}.



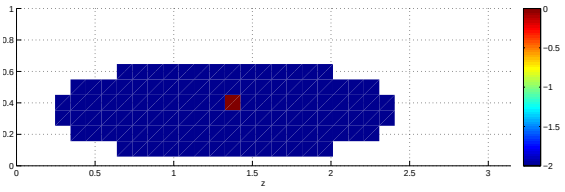
(a) Scalar field, $\log_{10}$-scale



(b) Expected rate of encounters for $\alpha = 0.1$



(c) Expected rate of encounters for $\alpha = 0.05$



(d) Expected rate of encounters for $\alpha = 0.01$

Figure 6.13: Comparison of measured data with expected rate for several values of α at x_{64}.

Chapter 7

Results of infotaxis in a 3D turbulent flow

In this Chapter, we will give the results of applying the infotaxis algorithm for Re = 5600 and Re = 28000. These results consist of the trajectory of one of the simulations, an estimated relation between distance and search time and an estimated relation between time and success.

7.1 Re = 5600

7.1.1 Trajectory

In Figure 7.1, one of the simulated search trajectories is depicted. The color indicates time, with dark blue at the beginning evolving to red at the end. It is visible that when finding the first two hits, the algorithm increases the z-position of the searcher which results in (almost) no hits, hence the searcher returns and continues its search at a lower z-coordinate (with more success).

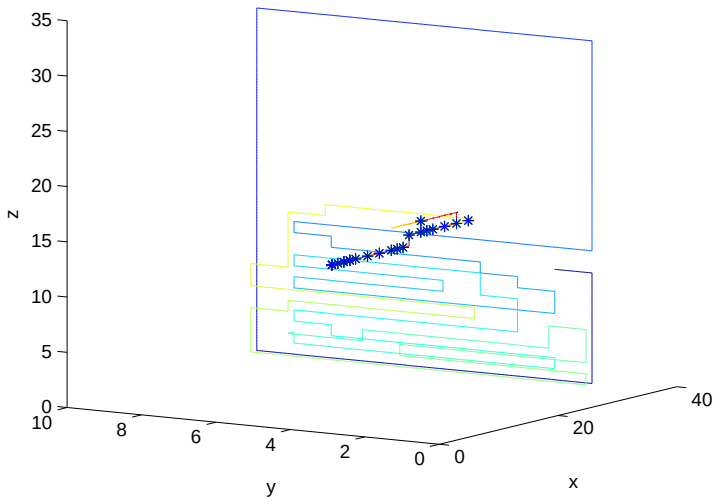


Figure 7.1: Search trajectory - the search starts at (32,3,11) and finishes at (6,4,14) after 362 time steps and 15 hits. The hits are indicated with asterisks.

7.1.2 Relation between distance and search time

The algorithm has been executed 50 times, with 46 successes for 50 starting positions in $[15, 35] \times [1, 9] \times [8, 24]$. Furthermore, 25 executions with starting positions uniformly distributed in $[30, 50] \times [1, 9] \times [8, 24]$

have been executed, with 17 successes. Failures occurred because simulations took too long or the entropy became almost zero before finding the source. Here, it was allowed to hit the wall and change direction instead of an immediate failure as in the particle case. These results have been combined in Figure 7.2.

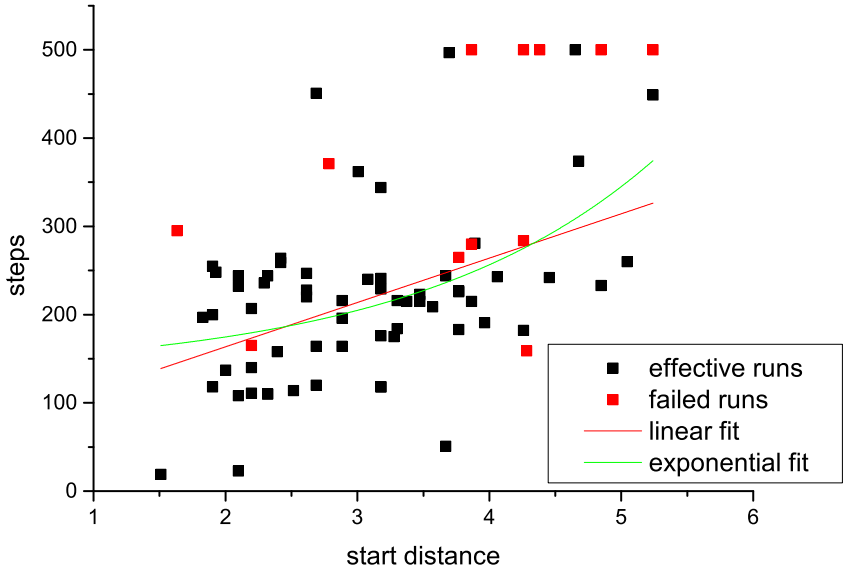


Figure 7.2: Relation between starting distance and search time - the linear fit is given by $y = a + bx$ with $a = 63.0$ and $b = 50.3$. The exponential fit is given by $y = a + b\exp(cx)$ with $a = 132.6$, $b = 14.3$ and $c = 0.54$.

The exponential fit has a lower residual sum of squares and seems to better incorporate the behaviour for large starting distances. This should be further explored in the case of a higher Reynolds number.

7.1.3 Relation between time and success

In contrast to the particle simulations, here the time already starts when no hit is detected. Also, the entropy decreases very fast once a hit is detected. This makes that there is a large spreading in the entropy, which makes it useless as an indicator when combining multiple runs.
Hence, to give an indication of the evolution of a run, we will show an empirical distribution which indicates the probability that a run has succeeded after t time steps ($P(t)$). This probability distribution is unknown, but we will estimate it together with a 95%-confidence interval.
We start with assuming that the simulations are identical and independent, which makes them trials of an experiment. Each trial has at each time step a probability of success $P(t)$, and can be indicated with success or failure. Hence, $P(t)$ is, for a certain t, given by a binomial distribution. The estimate of $P(t)$, $\hat{P}(t)$ is now for each t calculated by the number of successes divided by the total number of experiments. Combining all values of $\hat{P}(t)$ will give an empirical distribution which we will model by a gamma distribution. This distribution is often used in queueing theory to model the time required to perform some operation (in this case, i.e., to find the source), and its cumulative distribution function is given by

$$F(x, k, \theta) = \frac{1}{\Gamma(k)}\gamma(k, \frac{x}{\theta}) \tag{7.1}$$

in which k is the shape and θ the scale parameter, $\Gamma(k)$ the gamma function evaluated at k and $\gamma(k, \frac{x}{\theta})$ the incomplete gamma function. This cumulative distribution function can also be written as

$$F(x, k, \theta) = \frac{\int_0^{\frac{x}{t}} t^{k-1}e^{-t}\mathrm{d}t}{\int_0^{\infty} t^{k-1}e^{-t}\mathrm{d}t}. \tag{7.2}$$

For $k = 1$, the resulting distribution is an exponential distribution and for large k, the resulting distribution converges to a normal distribution with $\mu = k\theta$ and $\sigma = \theta\sqrt{k}$. The larger θ, the more the distribution

is spread out.
Now, we should validate that the simulations are identical and independent. Because we added simulations with a larger starting distance, the starting distances are not uniformly distributed in one box, hence we will take 48 simulations such that the distribution over the starting distances is uniform. However, this still does not result in identical simulations, because there is a relation between starting distance and running time. At the moment, we will neglect this.
The maximum likelihood estimates for the parameters k and θ are given by $k = 3.3126$ and $\theta = 62.2247$. This gives an estimated distribution $\tilde{P}(t)$. Using these values of $\tilde{P}(t)$, we calculate the 95%-confidence intervals from the binomial distribution. Combining these results gives Figure 7.3.

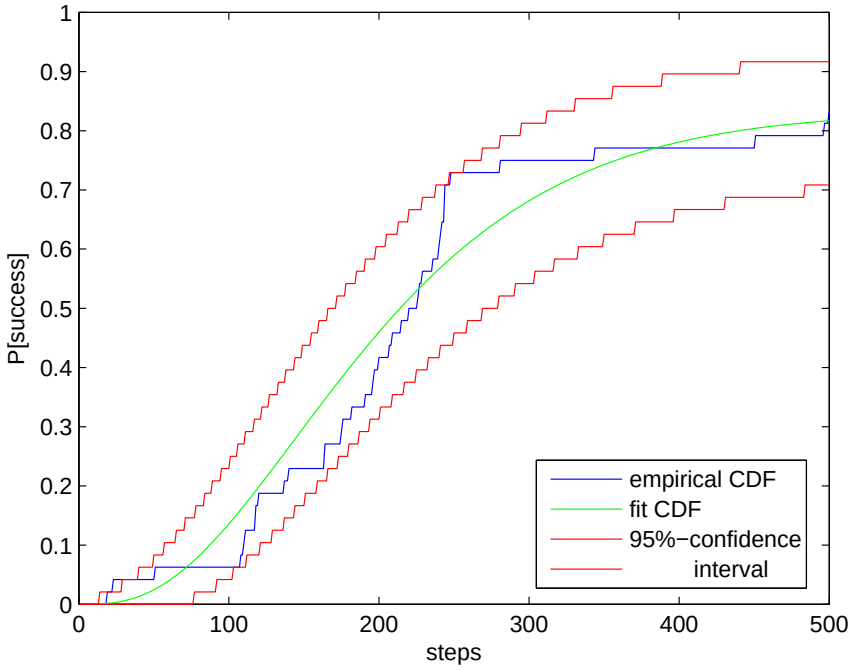
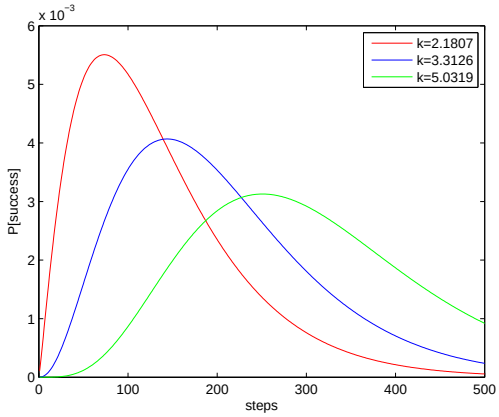
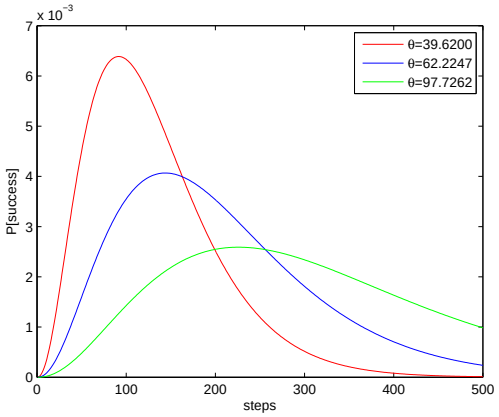


Figure 7.3: Relation between time and success - The gamma distribution has been fit using 40 succeeded simulations and rescaled with a factor 0.833 to include the failed runs. The fit parameters are $k = 3.3126$ and $\theta = 62.2247$. The 95%-confidence intervals are calculated using the binomial distribution with $\tilde{P}(t)$.

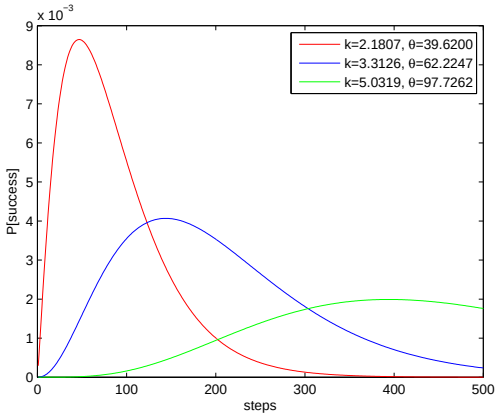
It is visible that sometimes the empirical cumulative distribution function coincides with the 95%-interval. This can be explained by the fact that the fit parameters also contain an error, which is neglected in the calculations. For these parameters, the 95%-confidence intervals are given by $[2.1807; 5.0319]$ for k and $[39.6200; 97.7262]$ for θ. In Figure 7.4, the effects of the parameters on the distribution are visualised. These Figures give the probability density functions for different combinations of k- and θ-values. It is visible that the 95%-confidence intervals vary a lot. Therefore, the cumulative distribution functions for the combined values of k and θ which amplify each other are depicted in Figure 7.5. Now, it is clear why the empirical CDF at some points coincides with the 95%-confidence intervals for the binomial function given the estimate $\tilde{P}(t)$.



(a) Spreading in k - The gamma density distribution is depicted with $\theta = 62.2247$, and for k both the estimated value and the values on the boundary of the confidence interval.



(b) Spreading in θ - The gamma distribution is depicted with $k = 3.3126$, and for θ both the estimated value and the values on the boundary of the confidence interval.



(c) Spreading in distribution - The gamma distribution is depicted with for both k and θ the estimated value and the values on the boundary of the confidence interval, in such a way that they amplify each other's effects.

Figure 7.4: Spreading in distribution - for both k and θ, the 95%-confidence intervals are visualised using the probability density functions. They are also combined, which gives even a larger spreading.

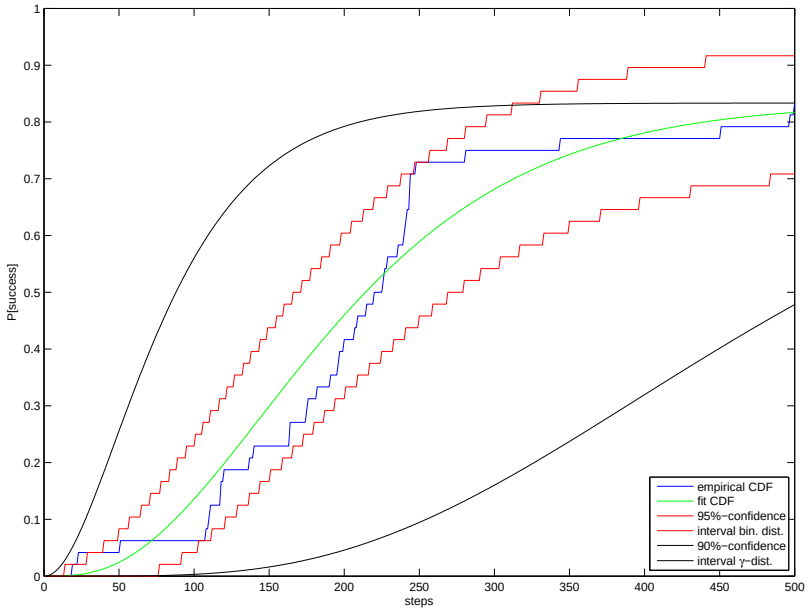
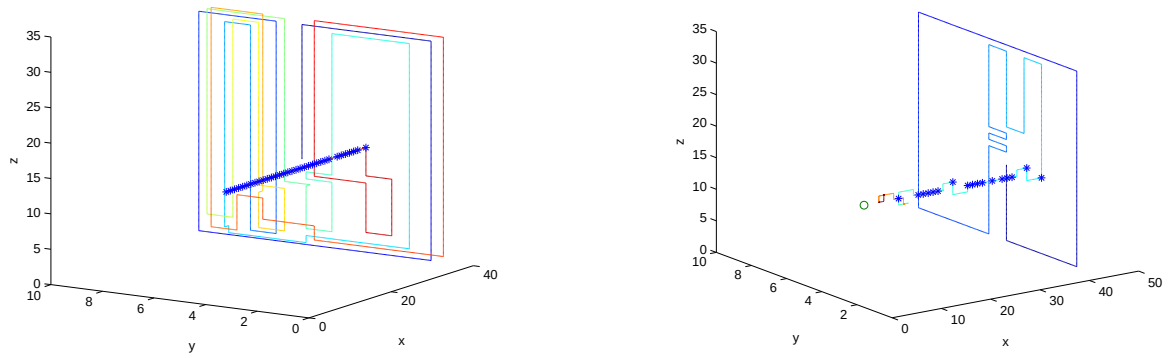


Figure 7.5: Relation between time and success - The gamma distribution has been fit using 40 succeeded simulations and rescaled with a factor 0.833 to include the failed runs. The fit parameters are $k = 3.3126$ and $\theta = 62.2247$. The 95%-confidence intervals are calculated for the binomial distributions based on $\tilde{P}(t)$ and the 90%-confidence interval is calculated for the rescaled gamma distribution.

7.2 Re = 28000

7.2.1 Trajectory

In Figure 7.6, two of the simulated search trajectories are depicted. The color again indicates time, with dark blue at the beginning evolving to red at the end. The trajectory depicted in Figure 7.6a first searches with constant x-coordinate until a hit is found and continues in a straight line to the source. The trajectory depicted in Figure 7.6b fails. The first hit is found quite fast, but the entropy decreases too fast, which results in very slow moving from (13,3,13) onwards and a failure because the entropy becomes less than the threshold value. In this case, the algorithm expects the source to be at (10,3,14).



(a) The search starts at (36,6,12) and finishes at (5,4,15) after 584 time steps and 33 hits.

(b) The search starts at (41,5,14) and finishes at (8,3,14) after 479 time steps and 17 hits.

Figure 7.6: Simulated search trajectories - the first search succeeds, while the second one fails based on entropy. The hits are indicated with asterisks.

7.2.2 Relation between distance and search time

The algorithm has been executed 50 times, with 47 successes for 50 starting positions in $[10, 50] \times [1, 9] \times [8, 24]$. Failures occurred because simulations took too long or the entropy became numerically zero before finding the source. The results are visualised in Figure 7.7.
The fits both relate to unexpected behaviour: for the linear fit, the search time would decrease for larger starting distances, however, the confidence interval for a does not exclude a positive coefficient. The exponential fit would lead to a very large negative search time for small distances, e.g., 0.5. Hence, we will reject the fits and assume there is in this case no direct relation between starting distances and search times.

7.2.3 Relation between time and success

We will try to give a relation between time and success based on the gamma distribution. The starting positions are uniformly distributed with no relation between starting distance and simulation time, hence we will assume they are identical and independent trials of an experiment, such that we are allowed to use the binomial distribution for the 95%-confidence intervals.
The maximum likelihood estimates for the parameters k and θ are given by $k = 4.6388$ and $\theta = 49.5727$. This gives an estimated distribution $\tilde{P}(t)$. Using these values of $\tilde{P}(t)$, we calculate the 95%-confidence intervals from the binomial distribution. Combining these results gives Figure 7.8.
Also in this case, the empirical cumulative distribution function sometimes coincides with the 95%-interval. The 95%-confidence intervals are now given by $[3.1387; 6.8558]$ for k and $[32.8143; 74.8898]$ for θ. In Figure 7.9, the effects of the parameters on the distribution are visualised. These Figures give the probability density functions for different combinations of k- and θ-values. It is visible that the 95%-confidence intervals again vary a lot. Therefore, the cumulative distribution functions for the combined values of k and θ which amplify each other are depicted in Figure 7.10.

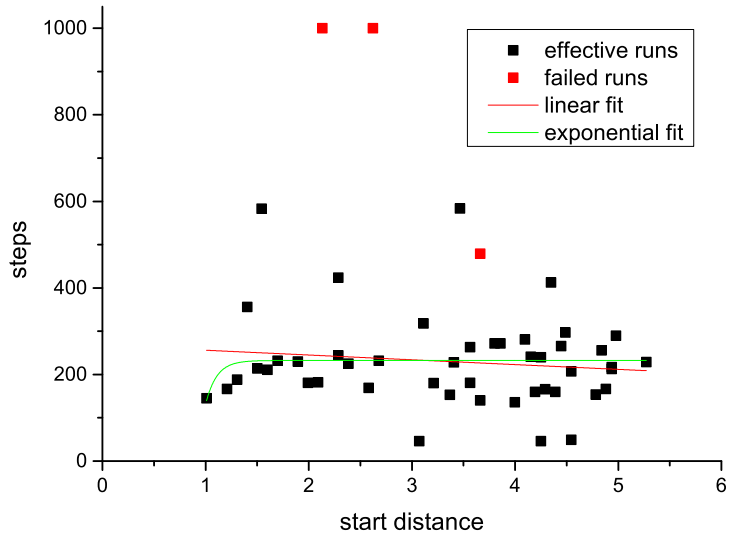


Figure 7.7: Relation between starting distance and search time - the linear fit is given by $y = a + bx$ with $a = 267.2$ and $b = -11.1$. The exponential fit is given by $y = a + b\exp(cx)$ with $a = 232$, $b = -1.35 \cdot 10^6$ and $c = -9.5$.

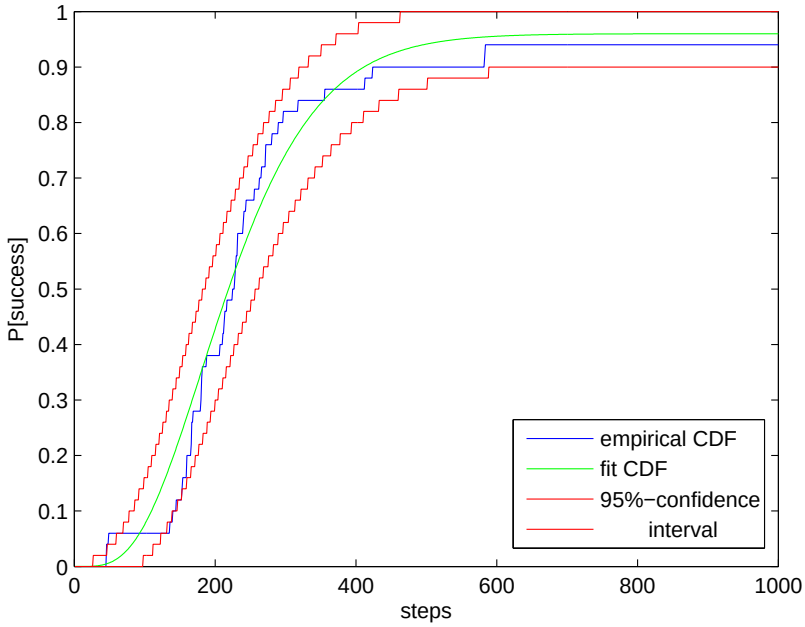
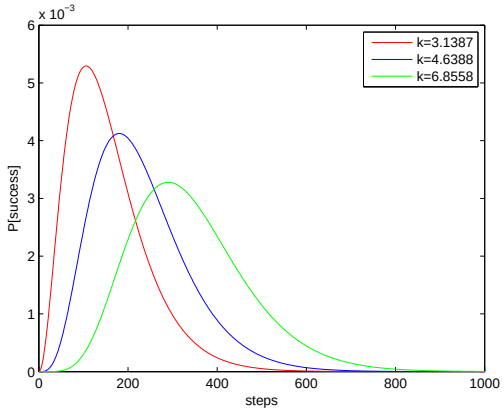
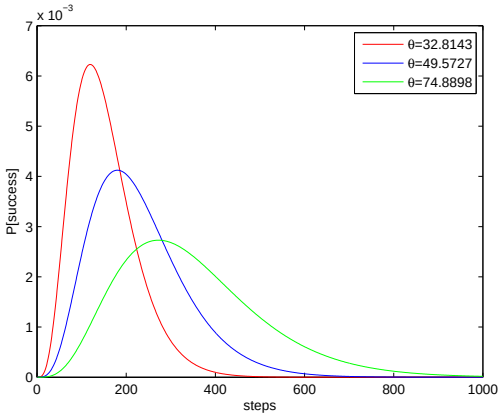


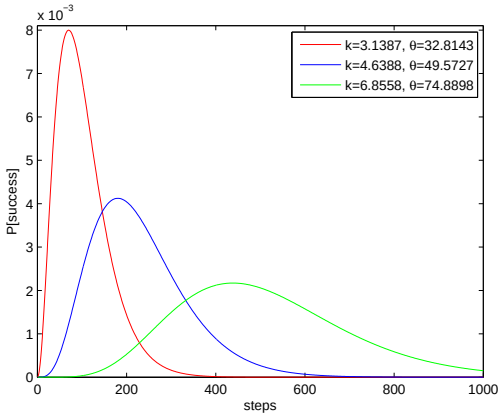
Figure 7.8: Relation between time and success - The gamma distribution has been fit using 47 succeeded simulations and rescaled with a factor 0.94 to include the failed runs. The fit parameters are $k = 4.6388$ and $\theta = 49.5727$. The 95%-confidence intervals are calculated using the binomial distribution with $\tilde{P}(t)$.



(a) Spreading in k - The gamma density distribution is depicted with $\theta = 49.5727$, and for k both the estimated value and the values on the boundary of the confidence interval.



(b) Spreading in θ - The gamma distribution is depicted with $k = 4.6388$, and for θ both the estimated value and the values on the boundary of the confidence interval.



(c) Spreading in distribution - The gamma distribution is depicted with for both k and θ the estimated value and the values on the boundary of the confidence interval, in such a way that they amplify each other's effects.

Figure 7.9: Spreading in distribution - for both k and θ, the 95%-confidence intervals are visualised using the probability density functions. They are also combined, which gives even a larger spreading.

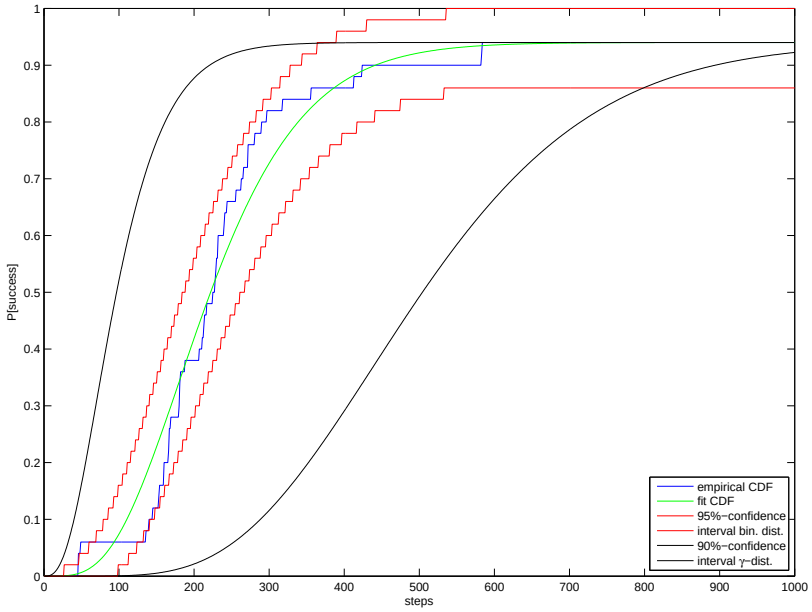


Figure 7.10: Relation between time and success - The gamma distribution has been fit using 47 succeeded simulations and rescaled with a factor 0.94 to include the failed runs. The fit parameters are $k = 4.6388$ and $\theta = 49.5727$. The 95%-confidence intervals are calculated for the binomial distributions based on $\tilde{P}(t)$ and the 90%-confidence interval is calculated for the rescaled gamma distribution.

Part III

Possible applications

Chapter 8

Practical cases

All of the previous results were for a general setup in an air flow with arbitrarily chosen parameters. However, to apply this model in real-world applications, the parameters need to be chosen differently. For two situations, this will be further explored. One situation is a garden, in which insects search for odour traces of flowers. This is a small-scale application, because the search area is several cubic metres and the search takes place at relatively small wind speeds. The other situation is a leak in a subsea oil pipeline, in which the oil spreads into the sea. This is a large-scale application, because the search area can be cubic kilometres large.

8.1 Insects and flowers

For this situation, the Schmidt number is in general of order 1, because air is the transport fluid. This means that the continuum model will suit best, because the odour will spread as a plume. To get the parameters for the simulation, the situation is made dimensionless. For this, the characteristic length scale will be 1 decimetre and the characteristic time scale 1 second. The calculated values are visible in Table 8.1.

Table 8.1: Parameters for the insects and flowers case.

Characteristic length scale Characteristic time scale	0.1 m 0.0625 s	dimensionless	current simulation
Grid size	$10 \times 10 \times 3$ m^3	$100 \times 100 \times 30$	$64 \times 128 \times 32$
T_{max}	15 min	14400	1000
Mean velocity	2 Bft (1.6 m/s)	1.0	5.0
Time step	1 s	16	1
Diameter searcher	1 cm (range 0.1 m)	0.1	0.09817
Kinematic viscosity (ν) (0°C)	$1.330{\cdot}10^{-5}$ m^2/s [14][15]	$8.313{\cdot}10^{-5}$	$1.785 \cdot 10^{-4}$
Source strength	1 ppm	$1.0 \cdot 10^{-6}$	$1.0{\cdot}10^{-7}$
Source position		(50,5,5)	(5,28,14)
Starting position		$(40 \pm 30, 40 \pm 30, 15 \pm 5)$	$(25 \pm 10, 5 \pm 4, 16 \pm 8)$

The grid size is based on the range of the searcher, because a grid to compute the fluid flow will be much finer.
The source strength can best be estimated in parts per million (ppm), but this will vary across different types of flowers. Hence, the 1 ppm is a rough estimate. In terms of concentration, the ppm is transformed into the ratio of odour particles and all particles.
From these values, it can be calculated that the Reynolds number will be $3.6 \cdot 10^5$. This implies that the simulation grid needs a lot of refinement, both in space and in time. Hence, this application will not be tested in this report.

8.2 Robots and oil

For this case, we will assume that the leak is very small, hence small droplets will escape from the pipeline into the salt water (30‰ salt). In this situation, probably the particle model works best, because oil does

not dissolve in the water. Also in this situation, the chosen parameters are made dimensionless and compared against the current simulation. These values are given in Table 8.2. Some values are not yet given in the table, but these will be given below.

Table 8.2: Parameters for the robots and oil case.

Characteristic length scale Characteristic time scale	1 km 1000 s	dimensionless	current simulation
Grid size	$50 \times 50 \times 1$ km^3	$500 \times 500 \times 10$	$16 \times 32 \times 16$
T_{max}	30 days	2592	300
Mean velocity y-direction	1.0 m/s	1.0	1.4
Time step	10 min	0.9	1
Diameter searcher	1 m (range 0.1 km)	0.1	1
Kinematic viscosity (ν) (20°C)	$9.86{\cdot}10^{-7}$ m^2/s [15]	$9.86{\cdot}10^{-10}$	$1.568{\cdot}10^{-5}$
Effective diffusion coefficient	0.0047 m^2/s**(1)**	$4.7{\cdot}10^{-6}$ **(1)**	0.72
Emission rate	3 min^{-1}	$1.3{\cdot}10^{4}$ **(2)**	0.4770
Source position		(250,100,2)	(8,5,8)
Starting position		$(250 \pm 100, 250 \pm 200, 8)$	$(10 \pm 2, 15 \pm 5, 10 \pm 2)$

The grid size is based on the range of the searcher, because a grid to compute the fluid flow will be much finer.
(1): For this, I will consider the shear velocity u_τ which was given by

$$u_\tau = \sqrt{\tfrac{\mu \Delta V}{\rho \Delta y}}. \tag{8.1}$$

The turbulent diffusion coefficient is proportional to

$$D_t \sim u_\tau d, \tag{8.2}$$

with for water a proportionality constant $C = 0.15$ for the transversal direction in a river, estimated by Fischer et al. [16]. Here D_t is not isotropic, but we assume so to have an isotropic diffusion coefficient. This gives

$$D_t = 0.15 d \sqrt{\frac{\mu V}{\rho d}} = 0.15 \sqrt{V \nu d} = 0.15 \sqrt{1.0 \cdot 9.86 \cdot 10^{-7} \cdot 1000} = 0.0047 \text{ m}^2/\text{s}, \tag{8.3}$$

or $4.7 \cdot 10^{-6}$ in the dimensionless case by using the characteristic length and time scale.
(2): Now the total of $R(\mathbf{r}|\mathbf{r}_0)$ over the grid is 328.4965 with $S = 50$ and there are approximately $50 \cdot 40/1.0 = 2.0 \cdot 10^3$ particles in the simulation, hence $S = 2.0 \cdot 10^3 \cdot 328.4695/50 = 1.3 \cdot 10^4$.
Furthermore, the number of grid points has increased by about a factor 300, which would dramatically increase the calculation time. Also, in this problem, it is not included that the oil moves up in the water because of the difference in density between them. This can be corrected with an anisotropic diffusion coefficient or by including buoyancy, but this will also increase the complexity of the problem. Next to that, the formation of oil droplets is not explored further here. Because of surface tension, there will be a preferred droplet size, and this influences the fluid. In the case of point particles, this effect is not visible. On average, this influence will probably not be very large.

8.3 Comparison

The problems are very different in nature, but it is expected that both problems can be solved using the infotaxis algorithm. However, both problems need substantially more calculation time than the situations in the previous Chapters, hence this will not be tested in this report. Also, not all parameters can be estimated properly, and experiments have to be conducted to find these.

Chapter 9

Mixing

The infotaxis algorithm is based on the fact that only partial information is available to find a source in a given fluid based on traces of evidence. These traces can be any chemical substance, as long as it diffuses in the base fluid. On the basis of this partial information, it is possible to find the source of the tracers. This "source" contains often a higher concentration of the searched chemical, which will increase the expected number of hits and thereby give more information.
In industry, when producing all kinds of fluids, polymers, etc., it is often needed that the fluid in a container or tank is well-mixed to get optimal production results. However, it is impossible to measure the concentration at all points in the tank. Hence, here also the problem of partial information occurs. The decision whether the fluid is well-mixed, depends on statistical indicators, for example the mean and variance of the concentration when measured at several points. But how to decide which points in the tank will be used? This also depends on the method of mixing, the characteristics of the chemical and the base fluid. To avoid all of this, the infotaxis algorithm might be an option.

In short, this will lead to the following statement: *When in such a tank, the infotaxis algorithm is applied to find a source, and it finds a source, then, the fluid is not well-mixed.* This can appear in several forms: there are patches with higher or lower concentrations, or there are significant concentration gradients in the fluid, or both. Hence, for properly quantified indicators, the infotaxis algorithm could give a decisive answer.

9.1 Algorithm

The success of this numerical experiment will depend on the chosen indicators and the setup of the algorithm. It will be assumed that the total amount of chemical and the total volume of the contents of the tank are known. Hence, also the average concentration of chemical is known. When there is a too large deviation from this concentration, this will be recorded as a hit, independent of whether the deviation is positive or negative. To end the algorithm, several options are possible. When it takes too long to find a hit, when too many hits occur, when it takes too long to find a source, or when a source is found, then the algorithm should end. A source is found will here be interpreted as (almost) zero entropy: the searcher knows almost for sure that a source is present at some place in the tank, or when the probability for a certain place exceeds a certain value. Quantitatively, this gives the following indicators:

- **Deviation indicator** α: when $S(x, y, z, t) \geq (1 \pm \alpha)S_{mean}$, a hit occurs.
- **First stopping time** β: when no hit is recorded at time β, the fluid is well-mixed.
- **First certainty indicator** γ: when the entropy drops below γ, the fluid is not well-mixed.
- **Second certainty indicator** ζ: when the probability for one position exceeds ζ, the fluid is not well-mixed.
- **Third certainty indicator** μ: when more than μ percent of the time, a hit is recorded, the fluid is not well-mixed. This indicator starts after time β.
- **Second stopping time** ξ: when at time ξ, none of the previous indicators gave a decisive answer, the fluid is assumed to be well-mixed.

The values of these indicators should be identified by the manager of the production line.

Also, there should be made an assumption on the behaviour of the fluid and the chemical to provide a function for the expected number of hits. This will be done for an on average non-moving fluid, in which diffusion occurs. This also includes a fluid which has been heavily moved such that large concentration gradients occur and the turbulent, or real diffusion is much larger than the molecular diffusion. For the tests, a two-dimensional tank will be used, hence, also a two-dimensional tank will be used to calculate the expected rate function.
The governing equation is the diffusion equation, given by

$$\begin{aligned} \frac{\partial C}{\partial t} &= D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}\right), \\ S(x,y,0) &= \delta(x_0,y_0), \end{aligned} \tag{9.1}$$

in which $\delta(x_0, y_0)$ is the Kronecker-delta function and D the diffusion coefficient. In this equation, the Péclet number does not occur because there is no flow involved. The solution of this equation is given by

$$S(x,y,t) = \frac{1}{\sqrt{4\pi Dt}}\exp\left(\frac{-(x-x_0)^2}{4Dt}\right)\frac{1}{\sqrt{4\pi Dt}}\exp\left(\frac{-(y-y_0)^2}{4Dt}\right) = \frac{1}{4\pi Dt}\exp\left(-\frac{(x-x_0)^2+(y-y_0)^2}{4Dt}\right) \tag{9.2}$$

To derive the expected rate function, the concentration function is multiplied by the cell size and divided by the total volume. In this way, the Riemann sum over the positions for a certain source position will be a little smaller than 1, which is not unusual, because in fact, no hits are expected for a well-mixed fluid. Also, looking at one position and summing over all possible source positions, will also give a value smaller than 1. This gives the following expected rate function:

$$R(\mathbf{r}|\mathbf{r_0},t) = R(x,y,x_0,y_0,t) = \frac{\Delta x\Delta y}{4\pi\Phi Dt}\exp\left(-\frac{(x-x_0)^2+(y-y_0)^2}{4Dt}\right), \tag{9.3}$$

in which Φ is the fluid volume, t the time, $\mathbf{r}_0 = (x_0, y_0)$ the source position and $\mathbf{r} = (x, y)$ the searcher position.

9.2 Testing

The tank will be represented by a box of size π times π, with 32×32 grid points, uniformly divided over the given area. The evolution of the concentration in time will be calculated through the RK4-method, while the discretisation of the space derivatives is central in space. The boundaries will be assumed periodic for numerical purposes. Furthermore, it is assumed that the fluid has been actively stirred, but has come to rest. For this situation, three cases will be investigated. In the first case, there is a higher concentration in the centre than at the edges, in patched form. In the second case, there are multiple deviations throughout space and in the third case, only at a few grid points, the concentration is out of bounds.

9.2.1 One patch

The initial concentration profile is chosen as follows:

$$S(i,j) = 0.60 + 0.1\pi^2 \sin\left(\frac{(i-\frac{1}{2})\pi}{32}\right)\cdot\sin\left(\frac{(j-\frac{1}{2})\pi}{32}\right), \quad i,j = 1,\ldots,32. \tag{9.4}$$

This gives an average concentration of 1.0003, and 932 of the 1024 positions will give a hit. In this case, the RK4-time step is 0.001 unit time, the algorithm checks the concentration each unit of time, the turbulent diffusion coefficient is set at 10^{-4}, and the indicators are $\alpha = 0.05$, $\beta = 60$, $\gamma = 10^{-4}$, $\zeta = 0.1$, $\mu = 20\%$ and $\xi = 60$. Executing the algorithm 100 times for starting positions uniform in $[6, 26] \times [6, 26]$ gives for 99 of the executions the result "possible source found" on the basis of the γ-indicator, after 1 time unit for 98 cases and after 2 time units for 1 case. In the other case, the result is "possible source found" on the basis of the ζ-indicator, after 4 time units. This indicates that finding a hit in this case (almost) automatically results in indicating an insufficiently mixed fluid.

9.2.2 Small deviations everywhere

The initial concentration profile is chosen as follows:

$$S(i,j) = 1.00 + 0.06 \sin\left(\frac{(i-\frac{1}{2})\pi}{32}\right) \cdot \sin\left(\frac{(j-\frac{1}{2})\pi}{32}\right), \quad i,j = 1,\ldots,32. \tag{9.5}$$

This gives an average concentration of 1.0000, and 128 of the 1024 positions will give a hit. In this case, the RK4-time step is 0.001 unit time, the algorithm checks the concentration each unit of time, the turbulent diffusion coefficient is set at 10^{-4}, and the indicators are $\alpha = 0.05$, $\beta = 60$, $\gamma = 10^{-4}$, $\zeta = 0.1$, $\mu = 20\%$ and $\xi = 60$. Executing the algorithm 100 times for starting positions uniform in $[6,26] \times [6,26]$ gives for 84 of the executions the result well-mixed on the basis of the β-indicator, for 13 of the executions the result not well-mixed on the basis of the γ-indicator and for 3 executions the result not well-mixed on the basis of the ζ-indicator. Hence, for this series of experiments, 84% of the experiments indicates well-mixed, while the fluid is not well-mixed. This could be caused by the fact that the deviations are not random enough, hence the searcher might search in straight lines without finding a hit.

9.2.3 Large deviation at some spots

The initial concentration profile is chosen as follows:

$$S(i,j) = \begin{cases} 1.06 & (i,j) = (6,6), (6,27), (27,6) \text{ or } (27,27), \\ 1.00 & \text{else.} \end{cases} \tag{9.6}$$

This gives an average concentration of 1.0002, and 4 of the 1024 positions will give a hit. In this case, the RK4-time step is 0.001 unit time, the algorithm checks the concentration each unit of time, the turbulent diffusion coefficient is set at 10^{-4}, and the indicators are $\alpha = 0.05$, $\beta = 60$, $\gamma = 10^{-4}$, $\zeta = 0.1$, $\mu = 20\%$ and $\xi = 60$. Executing the algorithm 100 times for starting positions uniform in $[6,26] \times [6,26]$ gives for 99 of the executions the result well-mixed on the basis of the β-indicator, while in 1 execution the result will be not well-mixed on the basis of the γ-indicator after 3 time units.

9.2.4 Results

For case 1, which is clearly not well-mixed, the algorithm works as it should. For case 2, the algorithm indicates sometimes a well-mixed and sometimes a non well-mixed fluid. In case 3, it is even harder to indicate the non-well-mixedness. Hence, also an indicator for the number of the experiments and the combination of the experiments should be given.

Chapter 10

Conclusion and recommendations

10.1 Conclusion

In this report, the infotaxis-based search algorithm has been tested in several turbulent (channel) flows. This algorithm is based on the trade-off between exploration and exploitation in case of partial information. This will often result in zigzag patterns (exploration) perpendicular to the flow direction followed by a trajectory in the flow direction to the source (exploitation).

A kinematic flow simulation is used to test the search algorithm by detection of particles both in two and three dimensions. This leads to a success rate of 80% in the two-dimensional case and a success rate of 72% in the three-dimensional case.

Next, the kinematic simulation is replaced by a direct numerical simulation of the Navier-Stokes equations. In this simulated flow, a substance is released from a periodic point source, which leads to a continuous distribution of the scalar instead of detectable particles. Detections occur when the concentration is above a certain threshold value. Because the Schmidt number in this simulation is given by $Sc = 1.00$, the detections are not independent anymore, and the algorithm is adapted to detect plumes of high concentration instead of independent detections. This includes the estimation of some extra parameters used in the search algorithm.
For a Reynolds number of 5600, the success rate is 84% and an exponential relation between starting distances and time steps is derived. Also, a relation including confidence intervals between time and success rate has been developed. For a Reynolds number of 28000, the success rate is 94% and there is no clear relation between starting distances and time steps. This is caused by the low Schmidt number and the high velocity, which leads to the source very fast after the first detection. Furthermore, in these simulations it is allowed to hit the wall in which case the searcher changes direction, in contrast to the kinematic simulations.

Because almost all of the parameters were arbitrarily chosen in these simulations, the parameters have also been estimated for two situations at different scales: insects searching for flowers in a garden and robots searching for oil in the sea. Both situations would need substantially more calculation time than the previous simulations, hence no simulations have been performed using these parameters.

Finally, the search algorithm is tested in reverse to detect whether a fluid is well-mixed. Detections in combination with time are then used as an indicator. By applying the algorithm to some prescribed concentration distributions in two dimensions, it is found that the algorithm is very sensitive to the parameter values.

New in this report compared to existing literature are the application of infotaxis on a 3D turbulent channel air flow, which includes the adaptation of the algorithm to detect clouds, the calculation of the gamma distributions as an estimate for the probability that a searcher has found the source after a certain time, and the suggestion to use infotaxis in reverse for mixing. All of these should be further investigated.

10.2 Recommendations

The first recommendation would be to increase the number of experiments. The results obtained in this report are based on a low number of experiments, increasing this number would lead to more reliable results, especially in the fits.
Other recommendations would relate to the setup of the flow and the algorithm.

- **Increase the Schmidt number.** When the Schmidt number is of order 1, then momentum and mass diffusion are convected at the same velocity. When the Schmidt number is large, e.g. of order 10^3, then mass diffuses much faster than momentum and small regions of high concentration appear, which will occur in e.g. water. This adaptation would make the detections of high concentration more independent and more realistic.
- **Develop the estimation of parameters.** In this report, properties of both the flow and the scalar are estimated beforehand. It would be better if the algorithm could estimate these during the run. These parameters would concern flow velocity and root mean square deviations of this velocity, but also the source strength, diffusion coefficient and Schmidt number.
- **Develop a better rate function.** When increasing the Schmidt number is not realistic, e.g. in air, where the Schmidt number is of order 1, then the rate function could be optimised in two ways. In this report, the rate function is estimated with the chosen parameter α to include boundary effects. For large simulation areas, these boundary effects might be neglected. Also, there could exist models which model the spreading of the scalar better than the Gaussian plume model for some cases.
- **Add randomness to the algorithm.** Sometimes, when the entropy is low, the searcher gets jammed and stays at the same position. When the algorithm would advise a random position if the searcher stays at the same position for some time, this situation might be broken through. It might also be interesting to add randomness in general, i.e., when determining the next position of the searcher, a random change of position is made with probability p.

Finally, a recommendation is made regarding the calculation of the relation between time and success. This should occur for several simulations with different starting positions, however, they should have the same starting distance for each calculation. In this way, the experiments can be considered identical and independent.

Bibliography

[1] M. Vergassola, E. Villermaux & B.I. Shraiman, 2007, 'Infotaxis' as a strategy for searching without gradients, *Nature* **445**, 406-409.

[2] E.M. Moraud & D. Martinez, 2010, Effectiveness and robustness of robot infotaxis for searching in dilute conditions, *Frontiers in neurorobotics* **4** (1), 1-8.

[3] C. Barbieri, S. Cocco, R. Monasson, 2011, On the trajectories and performance of Infotaxis, an information-based greedy search algorithm, *EPL (Europhysics Letters)* **94** (2), 20005.1-6

[4] E. Karpas, E. Schneidman, 2014, Collective Search by Group Infotaxis, *Collective Intelligence Conference*.

[5] M. Hesselberth, 2013, Infotaxis in practice, Technische Universiteit Eindhoven, *BSc Thesis*.

[6] M. Vergassola, E. Villermaux & B.I. Shraiman, 2007, Supplementary Materials for 'Infotaxis: searching without gradients', *Nature* **445**, 406-409.

[7] M.v. Smoluchowski, 1917, Versuch einer mathematischen Theorie des Koagulationskinetik kolloider Lösungen, *Zeitschrift für Physikalische Chemie* **92**, 129-168.

[8] T. Elenbaas, 2005, *Writing lines in turbulent air using Air Photolysis And Recombination Tracking*, Technische Universiteit Eindhoven, *PhD Thesis*.

[9] J. Kim, P. Moin & R. Moser, 1987, Turbulence statistics in fully developed channel flow at low Reynolds number, *Journal of Fluid Mechanics* **177**, 133-186.

[10] R.W.C.P. Verstappen, A.E.P. Veldman, 2003, Symmetry-preserving discretization of turbulent flow, *Journal of Computational Physics* **187**, 343-368.

[11] H.K. Versteeg & W. Malalasekera, 1995, *An introduction to computational fluid dynamics - The finite volume method*, Longman Scientific & Technical 105-114.

[12] H.K. Versteeg & W. Malalasekera, 1995, *An introduction to computational fluid dynamics - The finite volume method*, Longman Scientific & Technical 114-120.

[13] S.B. Pope, 2000, *Turbulent flows*, Cambridge University Press.

[14] I.S. Grigoriev & E.Z. Meilikhov, 1997, *Handbook of Physical Quantities*, CRC Press.

[15] G. Verkerk et al., 1998, *Binas informatieboek 4^e druk*, Wolters-Noordhoff Groningen.

[16] H.B. Fischer, E.J. List, R.C.Y. Koh, J. Imberger, N.A. Brooks, 1979, *Mixing in inland and coastal waters*, Academic Press Inc. New York.