Journal of Computational and Applied Mathematics 310 (2017) 44–58

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Infotaxis in a turbulent 3D channel flow

A.W. Eggels [a,*], R.P.J. Kunnen [b], B. Koren [a], A.S. Tijsseling [a]

[a] *Centre for Analysis, Scientific Computing and Applications, Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands*

[b] *Fluid Dynamics Laboratory, Department of Applied Physics, Eindhoven University of Technology, The Netherlands*

ARTICLE INFO

Article history:
Received 31 January 2016
Received in revised form 22 April 2016

Keywords:
Infotaxis
CFD
Mixing
Search algorithm

ABSTRACT

In this paper, the infotaxis-based search algorithm is tested in several simulated turbulent channel flows. 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 a Schmidt number Sc of 1.0 and Reynolds numbers Re of 5600 and 28000.

For the direct numerical simulation with the adapted algorithm, there is a positive relation between the initial distances to the source and the running time, which holds for Re $=$ 5600 but which is not observed at Re $=$ 28000. This is caused by the low Schmidt number and the high velocity, which leads the searcher 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 of a too high or low concentration and the number of detections are used as measures for success. By applying the algorithm to some prescribed concentration distributions in two dimensions, it is found that the method is very sensitive to the threshold values for the mixing indicators.

1. Introduction

There must be a reason why insects in a garden follow a zigzag pattern most of the time. By turbulent 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 locally maximises the expected gain in information. 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 and Martinez [2] assessed the performance of infotaxis by combining robotic experiments and simulations. They concluded 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 was analysed both analytically and numerically. Masson et al. [4] and later Karpas and Schneidman [5] extended infotaxis to group behaviour, where each individual tries to maximise its own information, but is also able to use information from others.

[*] Corresponding author.
E-mail addresses: a.w.eggels@tue.nl (A.W. Eggels), r.p.j.kunnen@tue.nl (R.P.J. Kunnen), b.koren@tue.nl (B. Koren), a.s.tijsseling@tue.nl (A.S. Tijsseling).

http://dx.doi.org/10.1016/j.cam.2016.06.004
0377-0427/© 2016 Elsevier B.V. All rights reserved.

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 chemical. 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 Section 2, the infotaxis algorithm will be explained. Afterwards, we explain the setup of the simulated flow in Section 3. We have a channel flow which 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 28 000. This flow is periodic in the streamwise and spanwise direction, and bounded by plates in the normal direction. The setup of the algorithm is given in Section 4 and the results are presented in Section 5. The source of substance is continuous and the flow of the emitted concentration is calculated by an upwind discretisation, while the searcher is bounded to move on a computational grid.

Furthermore, in Section 6, the infotaxis algorithm is used to find an inverse result: whether a fluid is well-mixed or not. Too high or too low concentrations might indicate a source or sink, which is absent in a well-mixed fluid.

The conclusion is in Section 7 and recommendations for further research are in Section 8.

2. Infotaxis

Infotaxis is a search algorithm to find a source of some detectable substance, 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 the highest concentration of detectable substance. This will not work here 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 substance, one can estimate the probability that the substance is detected at a certain position in space. By waiting some time at one position, the searcher tries to detect the substance. These detections are called *hits*. Both when substance is and is not registered, this gives new information for the probability density. From this density, it is decided 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 options is expected to give more information. This procedure is repeated until the source is found.

The properties of the carrier fluid and the detectable substance are represented in the expected rate function $R(\mathbf{r}|\mathbf{r}_0)$, which is the expected rate of hits per unit of time at position $\mathbf{r} = (x, y, z)$ given the (unknown) source position $\mathbf{r}_0 = (x_0, y_0, z_0)$. The choice of $R(\mathbf{r}|\mathbf{r}_0)$ is not crucial for the algorithm to succeed. Most often, $R(\mathbf{r}|\mathbf{r}_0)$ is derived from a simple analytical model [1–8]. In this case, it is assumed that the substance satisfies the advection–diffusion equation with a source term, as given in the following equation for unidirectional flow

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

in which V is the crossectional average velocity in the x-direction, C the concentration of the substance, D the isotropic diffusion coefficient, τ the lifetime of the substance once emitted, S the emission rate of the source and δ the Kronecker delta. This equation does not contain time, hence it can be seen as the time-dependent advection–diffusion equation which has converged to a steady state after some time, with the initial condition of zero concentration everywhere. The convection is only in the x-direction for reasons of convenience. In 3D, the solution of Eq. (1) is [6]:

$$C(\mathbf{r}|\mathbf{r}_0) = \frac{S}{4\pi D\,|\mathbf{r} - \mathbf{r}_0|}\exp\left(\frac{-(x_0 - x)V}{2D}\right)\exp\left(-\frac{|\mathbf{r} - \mathbf{r}_0|}{\lambda}\right), \quad \lambda = \sqrt{\frac{D\tau}{1 + \frac{V^2\tau}{4D}}}, \tag{2}$$

and the expected rate of encounters is given by

$$R(\mathbf{r}|\mathbf{r}_0) = \frac{aS}{|\mathbf{r} - \mathbf{r}_0|}\exp\left(\frac{-(x_0 - x)V}{2D}\right)\exp\left(-\frac{|\mathbf{r} - \mathbf{r}_0|}{\lambda}\right), \tag{3}$$

in which a represents the radius (and size) of the searcher. Eqs. (2) and (3) hold when the mean velocity V is in the positive x-direction. 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. This is not necessary though for the infotaxis algorithm, infotaxis also works for correlated detections. In this way, the probability at each position $\mathbf{r}_0$ might be updated after a time δt using the rate of encounters $R(\mathbf{r}|\mathbf{r}_0)$ given the position of the searcher $\mathbf{r}$ and the number of hits η detected by

$$P(t + \delta t, \mathbf{r}_0) = P(t, \mathbf{r}_0)\,(R(\mathbf{r}|\mathbf{r}_0))^{\eta}\,e^{\delta t R(\mathbf{r}|\mathbf{r}_0)}/Z_t, \tag{4}$$

in which Z_t is a normalising constant. Hence, $\mathbf{r}$ is not a direct, but an indirect argument of P. Also, when the searcher arrives at a location where no source is present, then the probability of that location is set to 0. Furthermore, this position is excluded from the entropy calculations which are explained next.

From this local probability function only, it is not clear at first sight in which direction the searcher should move. We want a function which maps the whole probability function to one scalar which is easier to interpret. The goal is to maximise the expected gain of information and therefore, Shannon's entropy is used:

$$E_S = -\int P(\mathbf{x}) \log(P(\mathbf{x}))\, \mathrm{d}\mathbf{x}, \tag{5}$$

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

$$\Delta E_S(\mathbf{r} \to \mathbf{r}_j) = P(t, \mathbf{r}_j)(-E_S) + (1 - P(t, \mathbf{r}_j)) \left(\sum_{k=1}^{\infty} \rho_k(\mathbf{r}_j) \Delta E_{S_k} \right). \tag{6}$$

The first term on the right-hand side represents the probability that the source is found, which would reduce 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. In practice, the infinite sum will be truncated at a certain number $k_{\max}$. In our case, we alter the definition of a hit (Section 4.1.2), which leads to only having zero hits or one hit. The function ρ_k represents the probability that k hits are registered, which is calculated by the Poisson law:

$$\rho_k = h^k e^{-h}/k!, \qquad h(\mathbf{r}_j) = \Delta t \int P(t, \mathbf{r}_0) R(\mathbf{r}_j | \mathbf{r}_0)\, \mathrm{d}\mathbf{r}_0 \tag{7}$$

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 this is the better option.

3. Setup of the flow

We now want to apply the infotaxis algorithm to two flows given by the numerical solution of the time-dependent Navier–Stokes equations in a channel geometry, with a passive scalar released by a point source. This gives a situation in which no particles, but a concentration above a certain threshold will be detected.

Therefore, we will explain in this section the setup and the discretisation of the two flows. In these cases, the equations, constants, parameters and variables are dimensionless.

3.1. Parameters

The algorithm will be executed for two setups which have different Reynolds numbers Re. Re = 5600 for the first setup and 28 000 for the second setup. The source will release its substance in a fully developed turbulent flow, with a strength $S = S_{\max} = 1.0 \cdot 10^{-7}$. 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 3.4.1 and 3.6.1. The dimensionless 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

$$y_j = \frac{\sinh(\gamma j/ny)}{2\sinh(\gamma/2)}, \quad \gamma = 6.5. \tag{8}$$

The linear pressure drop is in the x-direction with a given mass flow ρYZU with density ρ of 1 and average velocity in the x-direction $U = 1.0$ resp. 5.0. The spatial discretisation is fourth-order and the time step is $\Delta t = 1.25 \cdot 10^{-3}$ resp. $5.00 \cdot 10^{-4}$.

In this way, the first situation is the same as in [9] and these reference data for the characteristics of the flow are available.

3.2. Simulation of the flow

The flow is simulated with the method proposed by Verstappen and 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).

3.3. *Modelling and simulation of the scalar*

3.3.1. *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 an integer N. Then the central point source is taken 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 $N = 52$. The absolute strengths can then be calculated as the relative strength divided by N and multiplied by S. 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.

3.3.2. *Numerical integration of the scalar equation*

We use the non-dimensionalised advection–diffusion equation:

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = \frac{1}{\mathrm{Pe}} \nabla^2 C, \tag{9}$$

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.

When $\mathrm{Pe} > 2$, then a central difference scheme causes oscillations, as described by Versteeg and Malalasekera [11]. This can be overcome by calculating the fluxes locally by an upwind discretisation scheme, for example as given in [12]. 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:

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

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

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 ($F^n = f(u^n)$) and $C^{n+1}_{i,j,k} = C^n_{i,j,k} + \Delta t F^n_{i,j,k}$.

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

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

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

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

For $x = x_{\min} = 0$, the boundary condition is similar to $z = z_{\min} = 0$. For $x = x_{\max} = 2\pi$, $C_{nx+1,j,k} = 0$.

3.4. *Mean values for* $\mathrm{Re} = 5600$

3.4.1. *Flow*

The steadying time of $T_{dev} = 1500\delta t$ as proposed in [10] turned out to be a good choice. The equations are made dimensionless by the channel width and the average streamwise velocity. Also, for the situations with $T_{\max} = 1500\delta t$, another 1500 time units have been simulated and averaged, to get rid of transient effects. These averages have been compared to data publicly available from [9]. The agreement between the two results is good, except for the spanwise root mean square

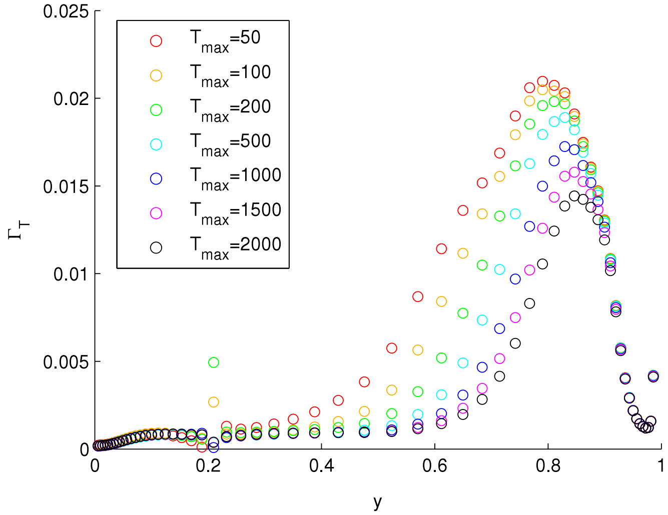


Fig. 1. Estimation of the turbulent diffusion coefficient Γ_T for Re $= 5600$—the values are only valid when positive; the y-axis has been adapted for this purpose.

velocity fluctuations. This is probably a remnant of the applied initial condition leading to a slight up/down asymmetry. The initial conditions, chosen to increase the generation of turbulence, appear to give a very small flux upward of 0.002 in the spanwise direction, compared to π in the streamwise direction.

3.4.2. Scalar

The scalar will not be released continuously, but periodically, to avoid one plume being detected (almost) everywhere. This is done to make a situation where chemotaxis would not work. Because of the low Schmidt number, the plumes will not break into complex structures. This periodic release of a scalar will also decrease the number of hits, which will make the search more difficult. The source function for the scalar is given by

$$S(t) = S_{\max} \max\{0, \sin(\pi t)\}. \tag{10}$$

3.4.3. Diffusion

In order to find the turbulent diffusion for the scalar, we use the gradient-diffusion hypothesis as described in [13], which states that the turbulent scalar diffusivity $\Gamma_T(\mathbf{x}, t)$ can be approximated by

$$\langle v'C'\rangle = -\Gamma_T \frac{\partial\langle C\rangle}{\partial y}, \tag{11}$$

for a mean flow which is only a function of the wall-normal coordinate y. It is not crucial to know the exact value of the turbulent diffusion for the infotaxis algorithm to work properly if the error is not too large, as is pointed out in [14]. On the other hand, a good estimate will not reduce the performance of infotaxis.

The calculated values for Γ_T are given in Fig. 1, 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 Sc $= 0.57$. This continuous source is chosen to avoid the effects from the periodic source. The source position y_{28} in the grid corresponds to $y = 0.3163$. This is not the position where the turbulent diffusion coefficient is at its maximum. The peak at around $y = 0.2$ is an effect of the initial condition which averages out in time.

3.5. Implications

This result gives that, with a Schmidt number smaller than 1, almost no turbulent diffusion occurs. In this case, it is possible that other search strategies are more efficient, but we are interested in the case of sparse traces. Because of the numerical method, it is not feasible to increase the Schmidt number to large values, such as, e.g., 1000. To enlarge the diffusion a little bit, we change the Schmidt number from 0.57 to 1.0, which gives a Péclet number Pe of 5600. A solution for this could be a change in the function which calculates the expected number of hits. For this, the Gaussian plume model will be used in Section 4. Also, the registration of hits will be adapted.

3.6. Mean values for Re $= 28\,000$

When the Reynolds number is increased, the characteristics of the flow will change. The corresponding characteristics of the flow and the scalar will be shown in this subsection.

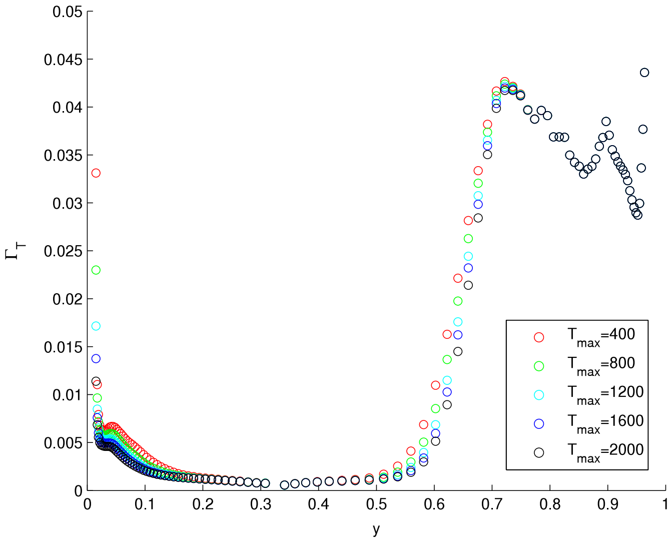


Fig. 2. Estimation of the turbulent diffusion coefficient Γ_T for Re = 28 000—again, the values are only valid when positive and the y-axis has been adapted accordingly.

3.6.1. Flow

The averaging time is now $T_{dev} = 2000\delta t$ and 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 intensity of turbulence. For this Reynolds number and the corresponding Reynolds number based on the wall shear velocity Re_τ of 630, calculated at the gridpoint at $y = 0.0131$, no reference data is available.

3.6.2. Scalar

The source function for the scalar is now given by

$$S(t) = S_{\max} \max\{0, \sin(5\pi t)\}. \qquad (12)$$

3.6.3. Diffusion

In the same way as before, Γ_T is calculated. The calculated values for Γ_T are given in Fig. 2, from which it 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 = 1.0.

4. Setup of the algorithm

The infotaxis algorithm, and in particular the expected rate function, was based on the fact that the flow followed the advection–diffusion equation with a source term. This has to be adapted to a rate function which incorporates the behaviour of the turbulent channel flow. As a basis for this, we choose the Gaussian plume model. In the Gaussian plume model, the concentration profile is given by

$$C(\mathbf{r}|\mathbf{r}_0) = C(x, y, z, x_0, y_0, z_0) = \frac{S}{2\pi U \sigma_y \sigma_z} \exp\left(\frac{-(y-y_0)^2}{2\sigma_y^2} + \frac{-(z-z_0)^2}{2\sigma_z^2}\right), \qquad (13)$$

with S the strength of the source, U the average velocity in the x-direction 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:

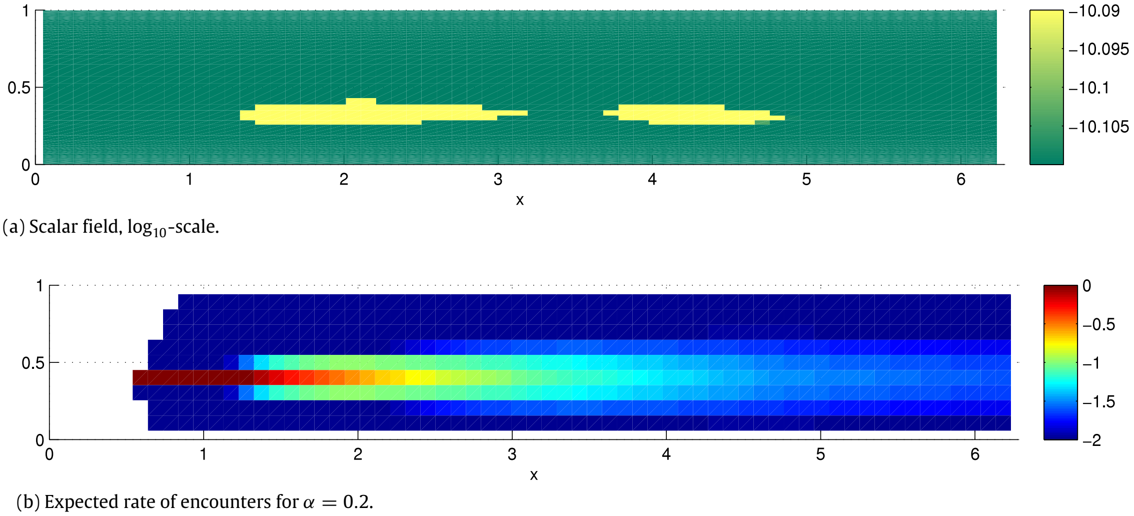
$$\sigma_y = \min\left\{\sqrt{\overline{v'^2}}\frac{(x-x_0)}{U}, \alpha D_y\right\}, \qquad (14)$$

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 and by Q, which is a scaling parameter which includes also the strength of the source and the threshold value of the concentration, hence

$$R(\mathbf{r}|\mathbf{r}_0) = \begin{cases} \dfrac{a^2 Q}{2\pi U \sigma_y \sigma_z} \exp\left(\dfrac{-(y-y_0)^2}{2\sigma_y^2} + \dfrac{-(z-z_0)^2}{2\sigma_z^2}\right), & (x - x_0) > 0, \\ 0, & (x - x_0) \le 0. \end{cases} \qquad (15)$$

This gives the unfortunate result of $R(\mathbf{r}_0|\mathbf{r}_0) = 0$, but fortunately this value will never be assigned in the simulation.



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

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

Fig. 3. Comparison of measured data with expected rate of encounters for several values of α at z_{14} for $T = 1600$ and Re $= 5600$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1. Parameters

The Reynolds number is 5600 resp. 28 000 and the Schmidt number is 1.0, which leads to Pe $= 5600$ and Pe $= 28\,000$, respectively. The source of the scalar is set in the grid points $(5, 28, 14)$ and $(5, 56, 14)$, respectively. 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 one after the other, and the searcher starts for Re $= 5600$ at $T_{dev} = 1500$ and for Re $= 28\,000$ at $T_{dev} = 2000$ in a flow that is developed. The parameters for the search algorithm are as follows: the waiting time W is equal to δt, i.e., to 800 and 2000 time steps Δt, respectively.

The diameter $2a$ of the searcher 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 10 grid points in the y-direction. Hence, there are two y-grids: one for the flow simulation and one for the algorithm. Q is determined to be 4. 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 $10^{-10.1} \approx 7.9 \cdot 10^{-11}$ and $10^{-10.75} \approx 1.7 \cdot 10^{-11}$, respectively, in Section 4.1.2.

We will calculate the parameters for the search algorithm beforehand, 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. Here, the same holds as for the estimation of the diffusion coefficient: it is not crucial for the infotaxis algorithm to work to have the correct value, as long as the error is not too large.

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

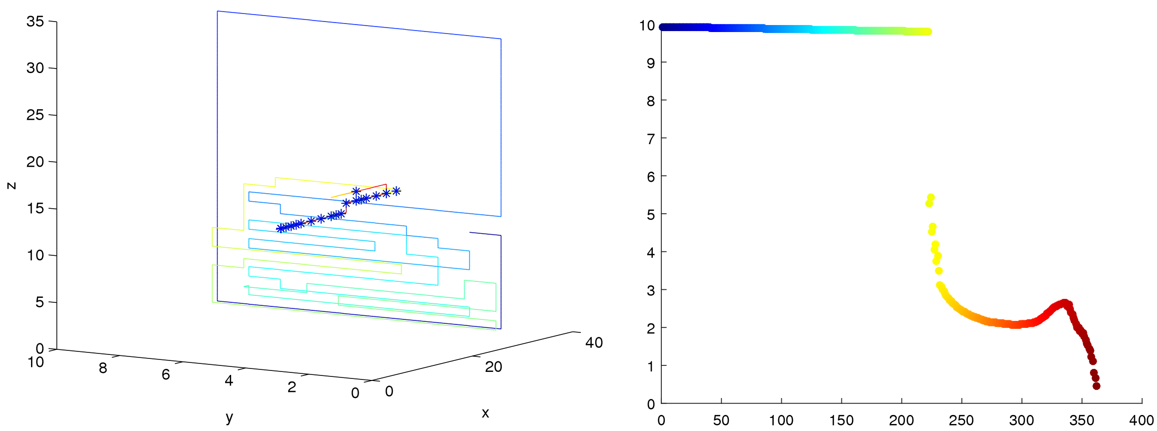
To calculate this, we simulate the flow a short time, starting at T_{dev}. Then, $\sqrt{\overline{v'^2}}$ is integrated from $y = 0$ to $y = 1$ by means of a Riemann sum and analogously for $\sqrt{\overline{w'^2}}$ from $z = 0$ to $z = \pi$. This gives for Re $= 5600$: $\sqrt{\overline{v'^2}} = 0.0472$ and $\sqrt{\overline{w'^2}} = 0.0539$, while for Re $= 28\,000$: $\sqrt{\overline{v'^2}} = 0.1264$ and $\sqrt{\overline{w'^2}} = 0.1878$.

4.1.2. Estimation of the shape parameter α and hit registration level

These are estimated by comparing figures of the simulation with figures with prescribed values for α. For clarity, we will show the figures for Re $= 5600$ and $\alpha = 0.2$ at z_{14}, which is the plane with grid coordinate $z = 14$, in Fig. 3. The other figures can be found in [15]. The green–yellow figure indicates simulated values, while the rainbow colours indicate expected values. Because of the very small values, we plotted the ^{10} log-values of the concentration and expected rate of encounters. For Re $= 5600$, we choose $\alpha = 0.2$ and register a hit if in more than half of the staying time a signal is registered. For Re $= 28\,000$, we choose $\alpha = 0.01$ and register a hit if in more than 35% of the staying time a signal is registered.

5. Results of infotaxis in a 3D turbulent flow

In this section, we will give the results of applying the infotaxis algorithm for Re $= 5600$ and Re $= 28\,000$. These results consist of the trajectory of one of the simulated searches, an estimated relation between distance and search time and an estimated relation between time and success.



(a) Search trajectory for Re = 5600—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.

(b) Corresponding entropy for the trajectory in (a).

Fig. 4. Search trajectory and corresponding entropy for Re = 5600. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5.1. Trajectories

In Figs. 4 and 5, some of the simulated search trajectories are depicted. The colour indicates time, with dark blue at the beginning evolving to red at the end. It is observed in Fig. 4(a) that when finding the first two hits, the algorithm increases the z-position of the searcher, which results in (almost) no further hits. Hence, the searcher returns and continues its search at a lower z-coordinate (with more success). The trajectory depicted in Fig. 5(a) first searches with constant x-coordinate until a hit occurs and continues in a straight line to the source. The trajectory depicted in Fig. 5(c) fails. The first hit occurs 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 and does not see the source as the position with the highest probability. In this case, the algorithm expects the source to be at (10, 3, 14). This is not equal to the place where the searcher is, and this is possible. In this case, the searcher did not register any hits for a long time, and the result basically says that the searcher had already passed the source (which is actually not the case).

5.2. Relation between distance and search time

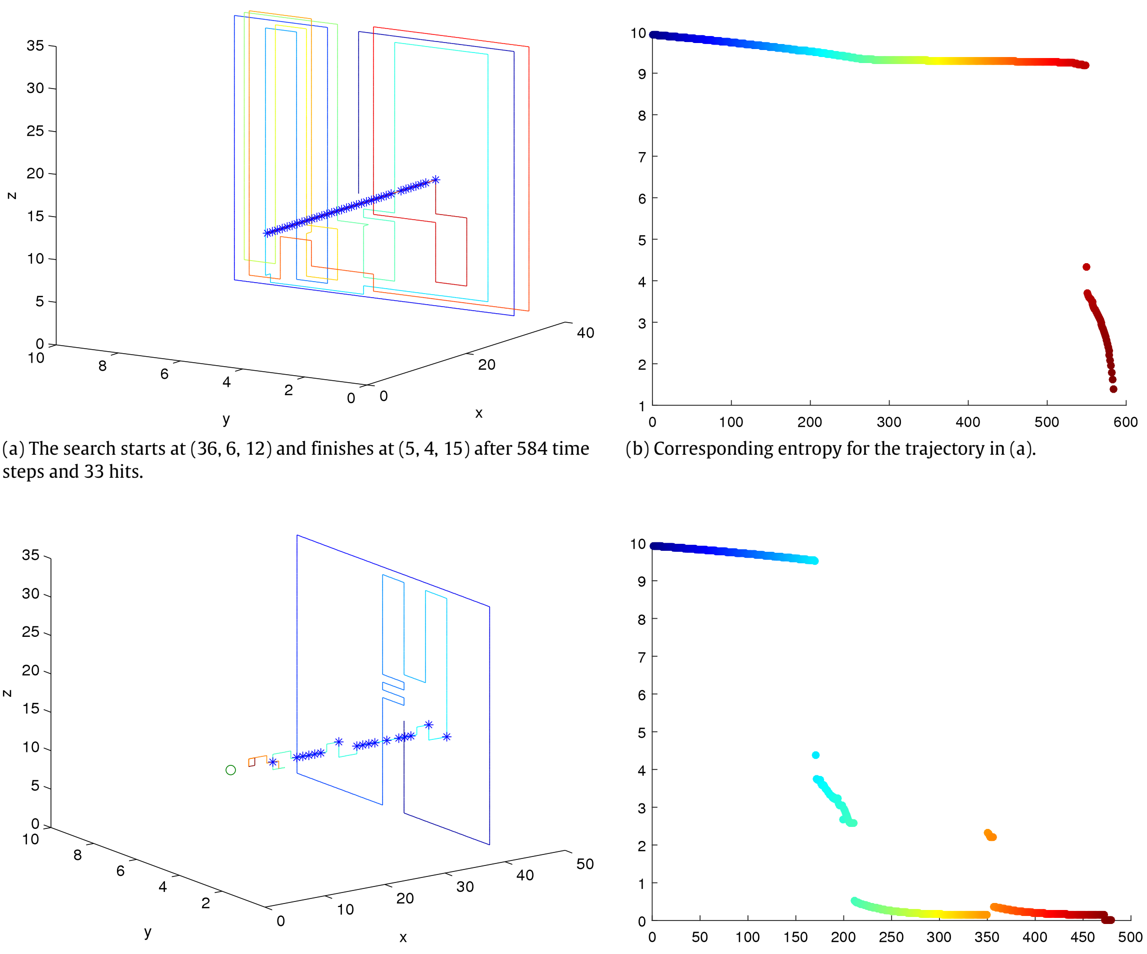
The algorithm has been executed 50 times for Re = 5600, with 46 successes for 50 arbitrary starting positions in the domain $[15, 35] \times [1, 9] \times [8, 24]$. Furthermore, 25 simulations with starting positions uniformly distributed in $[30, 50] \times [1, 9] \times [8, 24]$ have been performed, with 17 successes. Failures occurred because simulations took too long, because the entropy became almost zero before finding the source or because the position with highest probability was not located at the source. The results have been combined in Fig. 6. 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. We do not claim a linear or exponential relation, but we give the fits as an illustration. Moreover, in closed geometries such as a channel, the fluctuations of the search time (not its mean value) could dominate the search. This has been studied in [7].

For Re = 28 000 the algorithm has also been executed 50 times, with 47 successes for 50 arbitrary starting positions in the domain $[10, 50] \times [1, 9] \times [8, 24]$. Failures again occurred because simulations took too long or the entropy became numerically zero before finding the source. The results are visualised in Fig. 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 the regression parameter 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.

5.3. Relation between search time and success

The search time starts immediately and not when the first hit is detected. Also, the entropy decreases very fast once a hit is detected. Because the search time starts before registering the first hit, the entropy will only decrease slowly in time until the first hit is detected. Therefore, when we would look at a certain moment in time, some runs might have already



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

(b) Corresponding entropy for the trajectory in (a).

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

(d) Corresponding entropy for the trajectory in (c).

Fig. 5. Simulated search trajectories for Re = 28 000 in (a) and (c)—the first search succeeds, while the second one fails based on entropy. The hits are indicated with asterisks.

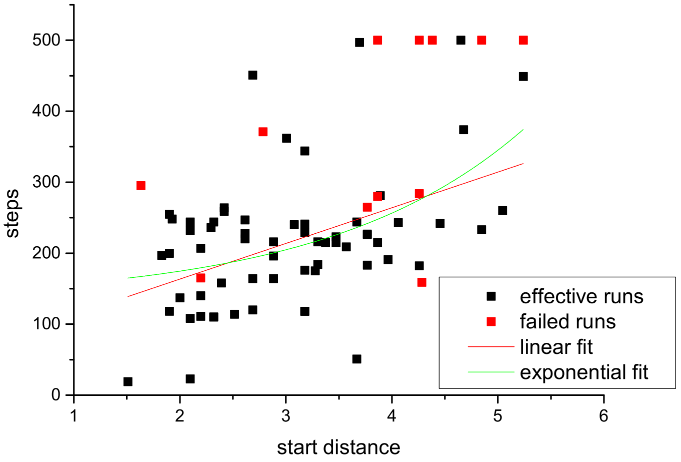


Fig. 6. Relation between starting distance and search time for Re = 5600—the linear fit is given by $y = a+bx$ with $a = 63.0$ and $b = 50.3$. The exponential fit is given by $y = a + b\exp(cx)$ with $a = 132.6$, $b = 14.3$ and $c = 0.54$.

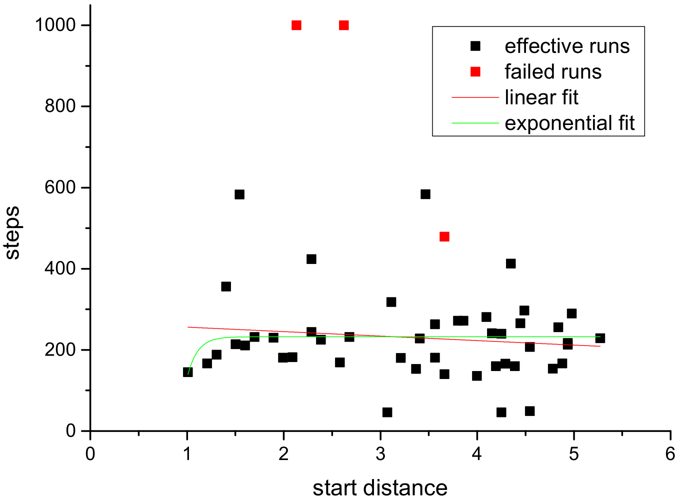


Fig. 7. Relation between starting distance and search time for Re = 28 000—the linear fit is given by $y = a + bx$ with $a = 267.2$ and $b = -11.1$. The exponential fit is given by $y = a + b\exp(cx)$ with $a = 232$, $b = -1.35 \cdot 10^6$ and $c = -9.5$.

succeeded, while other runs might still not have detected a single hit. Then, averaging the entropy over multiple runs does not distinguish between several runs with some hits, and a few which have not registered a hit yet.

Success is defined as the searcher being at a location from which it can reach the source within two steps. This can be interpreted as "being nearby enough to see the source". 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 with a 95%-confidence interval.

We start by 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, to find the source), and its cumulative distribution function is given by

$$F(x, k, \theta) = \frac{1}{\Gamma(k)}\gamma\left(k, \frac{x}{\theta}\right), \tag{16}$$

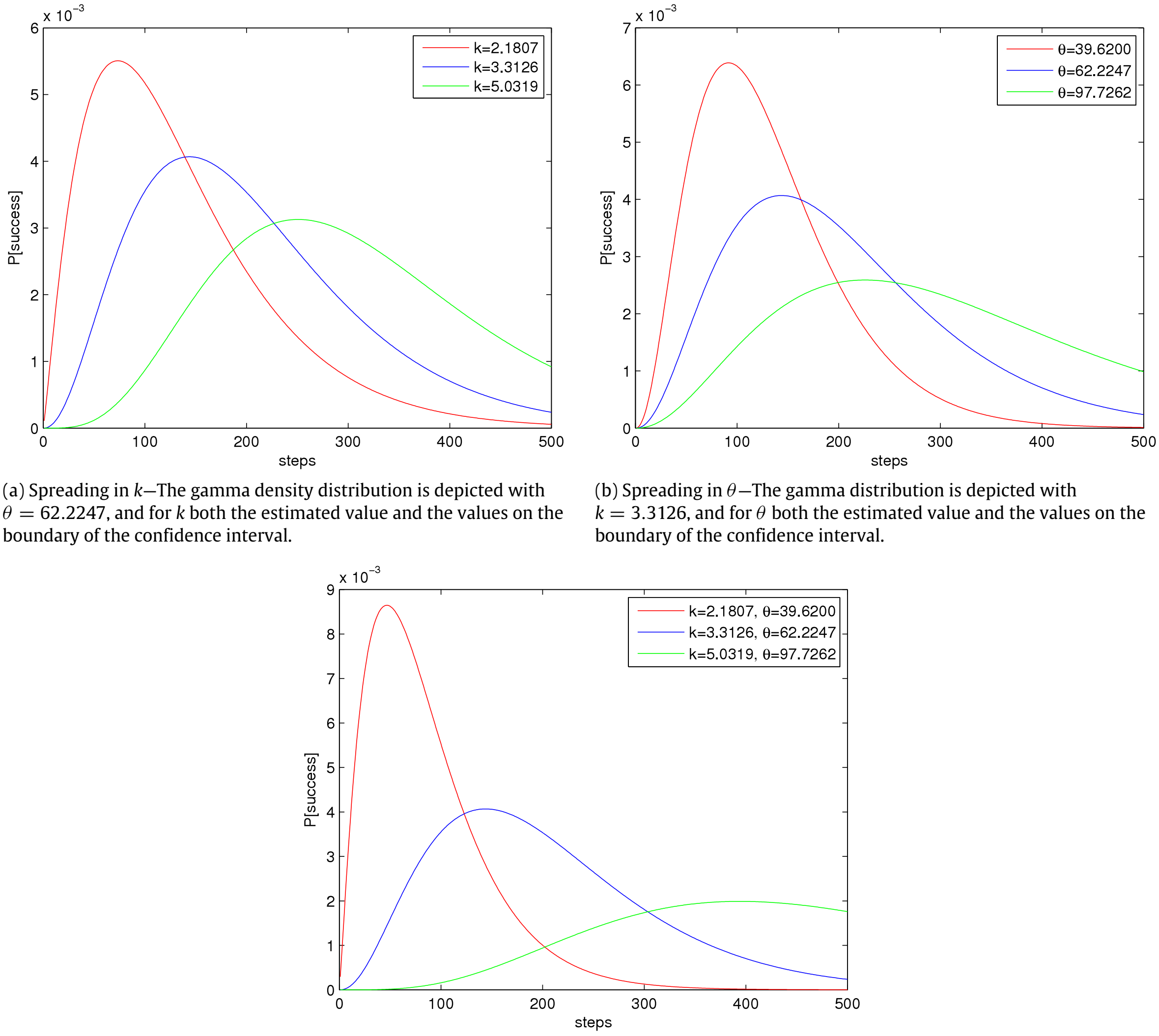
in which k is the shape and θ the scale parameter, $\Gamma(k)$ the gamma function evaluated at k and $\gamma(k, \frac{x}{\theta})$ the incomplete gamma function. This cumulative distribution function can also be written as

$$F(x, k, \theta) = \frac{\int_0^{\frac{x}{t}} t^{k-1}e^{-t}\mathrm{d}t}{\int_0^{\infty} t^{k-1}e^{-t}\mathrm{d}t}. \tag{17}$$

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 need to 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. We will take 48 simulations from Re = 5600 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 fact.

For Re = 5600, 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. It is observed that the empirical cumulative distribution function for some times coincides with the boundaries of the 95%-interval. This can be explained by the fact that the fit parameters also contain an error, which is ignored 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 Fig. 8, the effects of the parameters on the gamma distribution are visualised. These figures give the probability density functions for different combinations of k- and θ-values. It is seen 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 Fig. 9. Now, it is clear why the empirical cumulative distribution function at some points coincides with the boundaries of 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.

Fig. 8. Spreading in distribution for Re = 5600—for both k and θ, the 95%-confidence intervals are visualised using the probability density functions. They are also combined, which gives an even larger spreading.

For Re = 28 000, the starting positions are uniformly distributed with no relation between starting distance and simulation time, hence we will assume that they are identical and independent trials of one 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 $\theta = 49.5727$. Also in this case, the empirical cumulative distribution function sometimes coincides with the boundaries of the 95%-interval. The 95%-confidence intervals are now given by [3.1387; 6.8558] for k and [32.8143; 74.8898] for θ, and the parameters have a similar effect as before. Therefore, the cumulative distribution functions for the combined values of k and θ which amplify each other are depicted in Fig. 10.

6. 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. In this mixing case, the source is no real source, but should be interpreted as a region of higher concentration. This "source" will then increase the expected number of hits and thereby give more information.

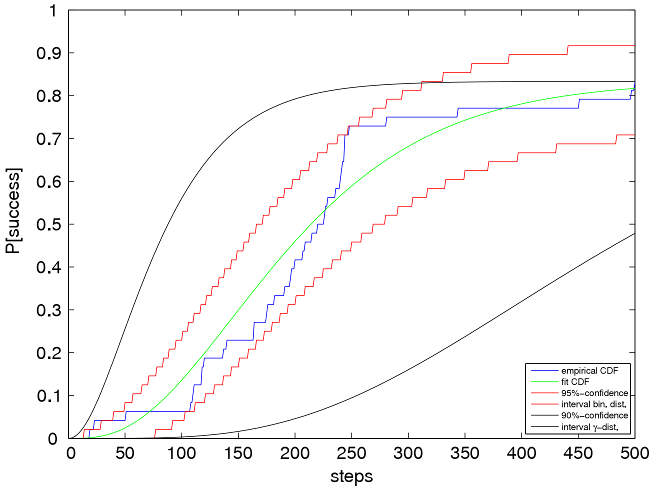


Fig. 9. Relation between time and success for Re $= 5600$—The gamma distribution has been fit using 40 successful simulations and rescaled with a factor 0.833 to include the failed runs. The fit parameters are $k = 3.3126$ and $\theta = 62.2247$. The 95%-confidence intervals are calculated for the binomial distributions based on $\tilde{P}(t)$ and the 90%-confidence interval is calculated for the rescaled gamma distribution.

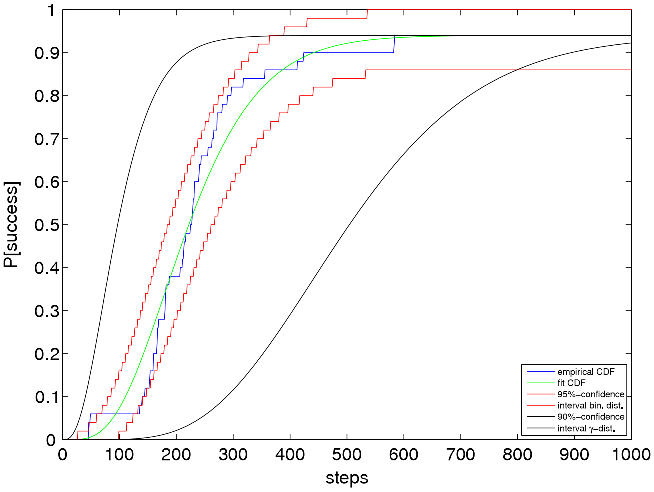


Fig. 10. Relation between time and success for Re $= 28\,000$—The gamma distribution has been fit using 47 successful simulations and rescaled with a factor 0.94 to include the failed runs. The fit parameters are $k = 4.6388$ and $\theta = 49.5727$. The 95%-confidence intervals are calculated for the binomial distributions based on $\tilde{P}(t)$ and the 90%-confidence interval is calculated for the rescaled gamma distribution.

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, also here 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.

6.1. 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 found source 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 $C(x, y, z, t) \geq (1 \pm \alpha)S_{mean}$, a hit occurs. This α differs from the shape parameter in the adapted expected rate function. This does not cause problems, because we will give a different expected rate function for the mixing case.
- **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 preferably 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. We will give a proof of principle with a simple problem statement. 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. We assume an ideal searcher, i.e. its presence does not induce additional stirring.

For the tests, a two-dimensional tank will be used. To calculate the expected rate function, a two-dimensional tank will be used as well.

The governing equation is the diffusion equation, given by

$$\begin{aligned} \frac{\partial C}{\partial t} &= D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}\right), \\ C(x, y, 0) &= \delta(x_0, y_0), \end{aligned} \tag{18}$$

with zero concentration at the boundaries of the infinite domain, 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

$$C(x, y, t) = \frac{1}{4\pi Dt} \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{4Dt}\right). \tag{19}$$

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 give a value smaller than 1 as well. This gives the following expected rate function:

$$R(\mathbf{r}|\mathbf{r}_0, t) = R(x, y, x_0, y_0, t) = \frac{\Delta x \Delta y}{4\pi \Phi Dt} \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{4Dt}\right), \tag{20}$$

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.

6.2. 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 with a time step of 0.001, while the discretisation of the space derivatives is central. The algorithm checks the concentration each time step and the turbulent diffusion coefficient is set at 10^{-4}. The indicators are set at $\alpha = 0.05$, $\beta = 60$, $\gamma = 10^{-4}$, $\zeta = 0.1$, $\mu = 20\%$ and $\xi = 60$. The algorithm will be applied 100 times for starting positions uniform in $[6, 26] \times [6, 26]$.

The boundaries will now 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, which differ in their initial concentration distribution. 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.

6.2.1. One patch

The initial concentration profile is chosen as follows:

$$C(i, j) = 0.6 + 0.1\pi^2 \sin\left(\frac{\left(i - \frac{1}{2}\right)\pi}{32}\right) \sin\left(\frac{\left(j - \frac{1}{2}\right)\pi}{32}\right), \quad i, j = 1, \ldots, 32. \tag{21}$$

This gives an average concentration of 1.0003, and 932 of the 1024 positions will give a hit. Out of 100 trials, 99 times the result is "not well-mixed" 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 "not well-mixed" 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.

6.2.2. *Small deviations everywhere*

The initial concentration profile is now as follows:

$$C(i,j) = 1.0 + 0.06 \sin\left(\frac{\left(i-\frac{1}{2}\right)\pi}{32}\right) \sin\left(\frac{\left(j-\frac{1}{2}\right)\pi}{32}\right), \quad i,j = 1,\ldots,32. \tag{22}$$

This gives an average concentration of 1.0, and 128 of the 1024 positions will give a hit. Out of 100 trials, 84 times the result is "well-mixed" on the basis of the β-indicator, 13 times the result is "not well-mixed" on the basis of the γ-indicator and 3 times the result is "not well-mixed" on the basis of the ζ-indicator. Hence, for this series of experiments, 84% of the experiments indicate 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 move in straight lines without experiencing a hit.

6.2.3. *Large deviation at some spots*

The initial concentration profile is given by:

$$C(i,j) = \begin{cases} 1.06 & (i,j) = (6,6), (6,27), (27,6) \text{ or } (27,27), \\ 1.0 & \text{else.} \end{cases} \tag{23}$$

This gives an average concentration of 1.0002, and 4 of the 1024 positions will give a hit. Now, 99 times the result is "well-mixed" on the basis of the β-indicator, while in 1 case the result is "not well-mixed" on the basis of the γ-indicator after 3 time units.

6.2.4. *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 not well-mixed fluid. In case 3, it is even harder to indicate the non-well-mixedness. An indicator for the number of experiments and the combination of the experiments is needed to make the method work.

7. Conclusion

In this paper, the infotaxis-based search algorithm has been tested in one turbulent channel flow at different Reynolds numbers. The algorithm is based on the trade-off between exploration and exploitation 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).

We used a direct numerical simulation of the Navier–Stokes equations. In this simulated flow, a substance is released from a periodic point source, leading 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 $Sc = 1.0$, the detections are not independent anymore. The algorithm has been 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 obtained success rate is 84% and an exponential relation between starting distances and time steps has been derived. Also, a relation including confidence intervals between time and success rate has been developed. For a Reynolds number of 28 000, 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 the searcher to the source very fast after the first detection.

Finally, the search algorithm has been 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 paper 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 estimating the level of mixing.

8. Recommendations

The first recommendation would be to increase the number of experiments. The results obtained in this paper are based on a limited 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 substance are convected at the same velocity. When the Schmidt number is large, e.g., of order 10^3, then substance 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 paper, 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 not only 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 paper, 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 may be better models for the spreading of the scalar than the adapted Gaussian plume model.
- **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.

Furthermore, several recommendations are possible in the case of using infotaxis to find whether a fluid is well-mixed.

- **Use real-life data to choose the parameters**. At the moment, the values of the indicators should be identified by an expert. The use of data can make the estimates for these values more reliable.
- **Investigate more testcases**. In this paper, only three testcases have been investigated. Increasing this number might give better insight into advantages and drawbacks of the method.
- **Extend to three dimensions and more complex flows**. This extension will show whether it is possible to use this method in real-life situations.

References

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

[2] E.M. Moraud, D. Martinez, Effectiveness and robustness of robot infotaxis for searching in dilute conditions, Front. Neurorobot. 4 (2010) 1–8.

[3] C. Barbieri, S. Cocco, R. Monasson, On the trajectories and performance of Infotaxis, an information-based greedy search algorithm, Europhys. Lett. 94 (2011) 1–6. 20005.

[4] J.-B. Masson, M. Bailly Bechet, M. Vergassola, Chasing information to search in random environments, J. Phys. A 42 (2009) 434009.

[5] E. Karpas, E. Schneidman, Collective search by group infotaxis, in: Collective Intelligence Conference, Massachusetts Institute of Technology, 2014.

[6] M. Vergassola, E. Villermaux, B.I. Shraiman, Supplementary materials for 'Infotaxis: searching without gradients', Nature 445 (2007) 406–409.

[7] J.D. Rodríguez, D. Gómez-Ullate, C. Mejía-Monasterio, Geometry-induced fluctuations of olfactory searches in bounded domains, Phys. Rev. E 89 (2014) 042145.

[8] J.D. Rodríguez, D. Gómez-Ullate, C. Mejía-Monasterio, Limits on the performance of Infotaxis under inaccurate modelling of the environment, 2014. arXiv:1408.1873.

[9] J. Kim, P. Moin, R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, J. Fluid Mech. 177 (1987) 133–186.

[10] R.W.C.P. Verstappen, A.E.P. Veldman, Symmetry-preserving discretization of turbulent flow, J. Comput. Phys. 187 (2003) 343–368.

[11] H.K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics—The Finite Volume Method, Longman Scientific & Technical, 1995, pp. 105–114.

[12] H.K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics—The Finite Volume Method, Longman Scientific & Technical, 1995, pp. 114–120.

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

[14] J.B. Masson, Olfactory searches with limited space perception, Proc. Natl. Acad. Sci. 110 (28) (2013) 11261–11266.

[15] A.W. Eggels, Infotaxis in a turbulent 3D channel flow (M.Sc. thesis), Technische Universiteit Eindhoven, 2015, http://alexandria.tue.nl/extra1/afstversl/wsk-i/Eggels_2015.pdf.