TW3750TU

Wednesday 9th February, 2022

Analysis of pollutant transport in a rectangular lake using a particle model

Simon Van Hulle, Elias Bögel

Delft University of Technology

TU Delft

Contents

1 Introduction

This work is a university assignment on Numerical Methods for Stochastic Differential Equations. The complete problem statement is given below.

Problem Statement

$$\text{Rectangular domain} : \begin{cases} -1 \leq x \leq 1 \\ -1 \leq y \leq 1 \end{cases} \tag{1.1}$$

$$H(x,y) = 15 + 5x \tag{1.2}$$

$$D_x(x,y) = 1 + \cos(\pi x) \tag{1.3}$$

$$D_y(x,y) = 1 + \cos(\pi y) \tag{1.4}$$

$$u(x,y) = -\frac{y\left(1-x^2\right)}{H(x,y)} \tag{1.5}$$

$$v(x,y) = \frac{x\left(1-y^2\right)}{H(x,y)} \tag{1.6}$$

(a) Derive a particle model for the simulation of the spreading of pollutants based on the 2D vertically averaged advection diffusion equation.
$\longrightarrow$ See Section 2.1.

(b) Explain that a particle can never cross a boundary of the lake.
$\longrightarrow$ See Section 2.2.

(c) What is the stationary particle density for $t \to \infty$ (use your physical intuition here).
$\longrightarrow$ See Section 2.3

(d) Implement the Euler and the Milstein scheme. Use these schemes for the simulation of the particles. Study by means of simulations the convergence behavior of the schemes for $\Delta t \to 0$ in the strong sense. Discuss the results.
$\longrightarrow$ See Section 3.1 for the schemes and Section 3.2.1 for the strong convergence.

(e) Verify experimentally the weak order of convergence. Discuss the results.
$\longrightarrow$ See Section 3.2.2.

(f) Because of numerical errors, particles can leave the domain of the problem. Study the performance of the various numerical schemes with respect to this numerical artefact.
$\longrightarrow$ See Section 3.3.

2

Theory

The first three parts of the assignment focus on several theoretical aspects of the problem. The particle model for the simulation of pollutant transport is derived in Section 2.1. After that, the reason that particles are not able to cross the domain boundaries is discussed in Section 2.2 and finally, the stationary particle density is predicted in Section 2.3.

2.1. Particle Model

(a) Derive a particle model for the simulation of the spreading of pollutants based on the 2D vertically averaged advection-diffusion equation.

The starting point for this derivation is the 2D advection-diffusion equation, as presented in the course reader (Heemink, 2021, p.28). This partial differential equation can be written as

$$\frac{\partial (HC)}{\partial t} = -\frac{\partial (uHC)}{\partial x} - \frac{\partial (vHC)}{\partial y} + \frac{\partial}{\partial x}\left(HD\frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(HD\frac{\partial C}{\partial y}\right) \qquad t \geq t_0. \tag{2.1}$$

Here $u(x, y, t)$ and $v(x, y, t)$ are the water velocity in x and y direction, respectively. Moreover, $C(x, y, t)$ is the concentration of particles, $H(x, y, t)$ is the water depth and $D(x, y) = D_x = D_y$ is the dispersion coefficient. However, for the specific case of this problem, the dispersion coefficient is different for the $x-$ and $y-$direction. The equation should therefore be adapted to

$$\frac{\partial (HC)}{\partial t} = -\frac{\partial (uHC)}{\partial x} - \frac{\partial (vHC)}{\partial y} + \frac{\partial}{\partial x}\left(HD_x\frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(HD_y\frac{\partial C}{\partial y}\right) \qquad t \geq t_0. \tag{2.2}$$

Now, one can also reformulate this in terms of a random walk model. Individual pollutant particles can then be simulated separately. When a large number of particles is used, this can accurately describe the pollutant spreading. The particle distribution is then simply a probability density function for the particles in the lake. In other words, the concentration is

the solution to a Fokker-Planck Equation. Recall that this equation, also called the Kolmogorov Forward Equation, is formulated as

$$\frac{\partial p}{\partial t} = -\sum_i \frac{\partial \left(f_i p\right)}{\partial x_i} + \frac{1}{2}\sum_i \sum_j \frac{\partial^2 \left(\left(gg^T\right)_{ij} p\right)}{\partial x_i \partial x_j}. \tag{2.3}$$

Thus, one can substitute $p = HC$ in Equation 2.2 and then rearrange the terms to get something of the form of Equation 2.3. Substituting yields

$$\frac{\partial \left(p\right)}{\partial t} = -\frac{\partial \left(up\right)}{\partial x} - \frac{\partial \left(vp\right)}{\partial y} + \frac{\partial}{\partial x}\left(HD_x \frac{\partial}{\partial x}\left(\frac{p}{H}\right)\right) + \frac{\partial}{\partial y}\left(HD_y \frac{\partial}{\partial y}\left(\frac{p}{H}\right)\right) \qquad t \geq t_0, \tag{2.4}$$

where the blue terms only contain $x-$derivatives and the gray terms only contain $y-$derivatives. Rearranging the terms is done exactly the same for the $x-$ and $y-$terms, so it is sufficient to work out one of them.

$$\begin{aligned}
&-\frac{\partial \left(up\right)}{\partial x} + \frac{\partial}{\partial x}\left[HD_x \frac{\partial}{\partial x}\left(\frac{p}{H}\right)\right] \\
&\quad = -\frac{\partial \left(up\right)}{\partial x} + \frac{\partial}{\partial x}\left(D_x \frac{\partial p}{\partial x} - \frac{D_x p}{H}\frac{\partial H}{\partial x}\right) \\
&\quad = -\frac{\partial \left(up\right)}{\partial x} + \frac{\partial}{\partial x}\left(D_x \frac{\partial p}{\partial x}\right) - \frac{\partial}{\partial x}\left(\frac{D_x p}{H}\frac{\partial H}{\partial x}\right) \\
&\quad = -\frac{\partial \left(up\right)}{\partial x} + \frac{\partial}{\partial x}\left(D_x \frac{\partial p}{\partial x} + p\frac{\partial D_x}{\partial x} - p\frac{\partial D_x}{\partial x}\right) - \frac{\partial}{\partial x}\left(\frac{D_x p}{H}\frac{\partial H}{\partial x}\right) \\
&\quad = -\frac{\partial \left(up\right)}{\partial x} + \frac{\partial}{\partial x}\left(D_x \frac{\partial p}{\partial x} + p\frac{\partial D_x}{\partial x}\right) - \frac{\partial}{\partial x}\left(\frac{D_x p}{H}\frac{\partial H}{\partial x} + p\frac{\partial D_x}{\partial x}\right) \\
&\quad = -\frac{\partial \left(up\right)}{\partial x} + \frac{\partial^2}{\partial x^2}\left(D_x p\right) - \frac{\partial}{\partial x}\left[\left(D_x \frac{\partial H}{\partial x} + H\frac{\partial D_x}{\partial x}\right)\frac{p}{H}\right] \\
&\quad = -\frac{\partial \left(up\right)}{\partial x} + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(2D_x p\right) - \frac{\partial}{\partial x}\left[\frac{\partial}{\partial x}\left(HD_x\right)\frac{p}{H}\right] \\
&\quad = +\frac{1}{2}\frac{\partial^2}{\partial x^2}\left(2D_x p\right) - \frac{\partial}{\partial x}\left[\left(u + \frac{1}{H}\frac{\partial \left(HD_x\right)}{\partial x}\right)p\right]
\end{aligned} \tag{2.5}$$

By analogy, the same rearrangement is done for the terms with $y-$derivatives. This results in

$$\begin{aligned}
&-\frac{\partial \left(vp\right)}{\partial y} + \frac{\partial}{\partial y}\left[HD_y \frac{\partial}{\partial y}\left(\frac{p}{H}\right)\right] \\
&\quad = \cdots \\
&\quad = +\frac{1}{2}\frac{\partial^2}{\partial y^2}\left(2D_y p\right) - \frac{\partial}{\partial y}\left[\left(v + \frac{1}{H}\frac{\partial \left(HD_y\right)}{\partial y}\right)p\right].
\end{aligned} \tag{2.6}$$

Substituting these rearrangements into Equation 2.4 yields a Fokker-Planck equation in the expected format:

$$\begin{aligned}\frac{\partial p}{\partial t} =& -\frac{\partial}{\partial x}\left[\left(u+\frac{1}{H}\frac{\partial\left(HD_x\right)}{\partial x}\right)p\right]-\frac{\partial}{\partial y}\left[\left(u+\frac{1}{H}\frac{\partial\left(HD_y\right)}{\partial y}\right)p\right]\\ &+\frac{1}{2}\left[\frac{\partial^2}{\partial x^2}\left(2D_x p\right)+\frac{\partial^2}{\partial y^2}\left(2D_y p\right)\right].\end{aligned} \tag{2.7}$$

Comparing this result to the definition in Equation 2.3 leads to identifying the components of the Îto SDE that governs the underlying particle model.

$$\begin{cases} f_x &= \left(u+\frac{1}{H}\frac{\partial\left(HD_x\right)}{\partial x}\right)\\ f_y &= \left(v+\frac{1}{H}\frac{\partial\left(HD_y\right)}{\partial y}\right)\\ g_x &= \sqrt{2D_x}\\ g_y &= \sqrt{2D_y}\end{cases} \tag{2.8}$$

$$\begin{aligned} dX_t &= \left(u+\frac{1}{H}\frac{\partial\left(HD_x\right)}{\partial x}\right)\,dt+\sqrt{2D_x}\,dW_t^x\\ dY_t &= \left(v+\frac{1}{H}\frac{\partial\left(HD_y\right)}{\partial y}\right)\,dt+\sqrt{2D_y}\,dW_t^y\end{aligned} \tag{2.9}$$

Equation 2.9 is thus the final particle model that can be used to model the evolution of pollutant concentrations in a lake.

2.2. Particles cannot cross Lake Boundary

(b) Explain that a particle can never cross a boundary of the lake.

While the intuition that, in reality, the water of a lake does not simply flow through the lake banks is obvious, it requires some more effort to verify and understand that this also holds for the mathematical model of the rectangular lake. This is also a property of our mathematical model that must be present in order to reflect the behaviour of the real system.

First it must be understood what the two modes of fluid and pollutant transport in the model of the lake are convection and dispersion:

1) Convection: Transport by convection occurs when the fluid itself is locally uniformly moving in one direction. In the real world this is a current in the lake. In which direction fluid or pollutant moves due to convection as part of the model is determined by the convection vector with components

$$\begin{aligned} u(x,y) &= -\frac{y(1-x^2)}{H(x,y)}\\ v(x,y) &= \frac{x(1-y^2)}{H(x,y)}\end{aligned} \tag{2.10}$$

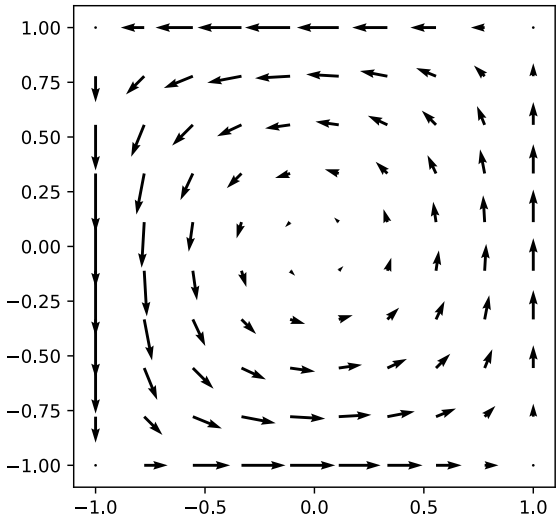

Figure 2.1: Convection velocity vector field on the domain

2) Dispersion: Transport by dispersion occurs through random motion of the fluid and pollutant in it on a small scale, inherent to fluids. In the mathematical model, this is modeled through the dispersion coefficient, which is again a vector with components

$$\begin{aligned} D_x(x,y) &= 1 + \cos(\pi x) \\ D_y(x,y) &= 1 + \cos(\pi y) \end{aligned} \tag{2.11}$$

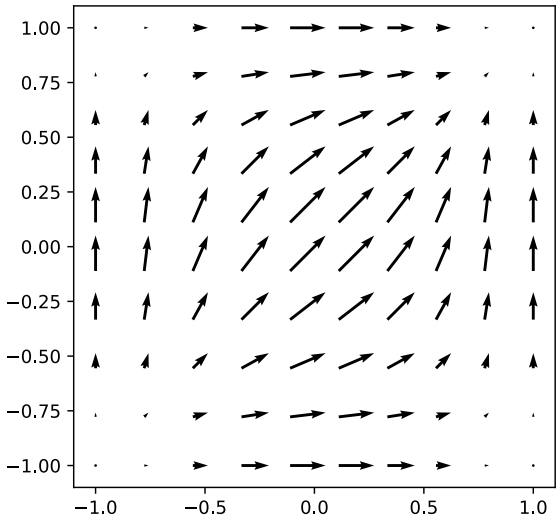

Figure 2.2: Dispersion coefficient vector field on the domain

The key to understanding why this model, in theory, conforms to the requirements that no transport over the domain boundaries occurs is to realize that both the convection and the

dispersion element of this transport model at the boundaries of the domain are always either zero or tangential to the boundary of the domain, as demonstrated in Figure 2.1 and Figure 2.2. This can be summarized in the conditions

$$\begin{cases} \vec{v} \cdot \vec{n} = 0 \\ \vec{d} \cdot \vec{n} = 0 \end{cases}, \tag{2.12}$$

where $\vec{v}$ is the convective flow vector, $\vec{d}$ is the vector consisting of the two dispersion coefficients and $\vec{n}$ is a vector normal to the boundary.

2.3. Stationary Density

(c) What is the stationary particle density for $t \to \infty$ (use your physical intuition here)?

In order to investigate the stationary particle density for this problem in a lake, we try to use our intuition of the real physical system that corresponds to the mathematical model. At this point, an analogy of a pool with depth varying like in the mathematical model and injected dye at the same position is useful: Intuition immediately shows that as $t \to \infty$, the dye will have spread to the entirety of the water body of the pool, even with varying initial pollutant release positions. Next, it is important to understand that convection in the mathematical model of the lake used is very weak and, for the intuitive thought experiment, can be ignored. However, the depth of the water at any point does have a large impact. Intuition predicts that the volumetric particle density (without vertical averaging) will be approximately constant if convection is neglected and only diffusive processes play a significant role. From this also follows that when then restricting the thought experiment to a perspective that looks at the pool from above and looking for a per-surface-area particle density, the conclusion is that regions with higher depth will contain proportionally more particles, and therefore their 2D particle density will also be higher. These considerations lead to an expectation for the steady-state particle density approximately directly proportional to the depth of the lake, irrespective of where the pollutant is inserted into the lake.

To confirm this perception, a numerical simulation of the system through the Euler or Milstein scheme introduced in Section 3.1 can be run for a long time until steady-state behaviour in the particle distribution is reached. Then, the particle probability density function may be recreated by placing a Gaussian kernel function on each particle and superimposing all kernel functions. The result of one such simulation and following kernel superposition can be found in Figure 2.3 and Figure 2.4.

While Figure 2.3 does not precisely match our expectation that the particle density varies almost purely with the height of the water column in the lake, these discrepancies found at the edges of the domain can be traced back to the way the kernel functions are superimposed. Naturally, close to the domain, a point will have less area for particles around it which contribute to the value of the sum of all kernel functions at this point. The most drastic examples are the points directly in one of the four corners of the domain, where there is only $1/4$ of the area where particles can be located compared to a point further in the center of the domain. The size and impact of this artefact are dependent on the size of the kernel functions used. The stationary particle distribution can similarly be analyzed using a histogram/binning approach to do away with the above mentioned. Such approach is shown in Figure 2.4. Both approaches also show some variation and noise across the domain, resulting from the inherently stochastic nature of data generation. A high number of particles was used in order to eliminate most noise. However, both ways of analyzing the particle distribution after a long time result in the same

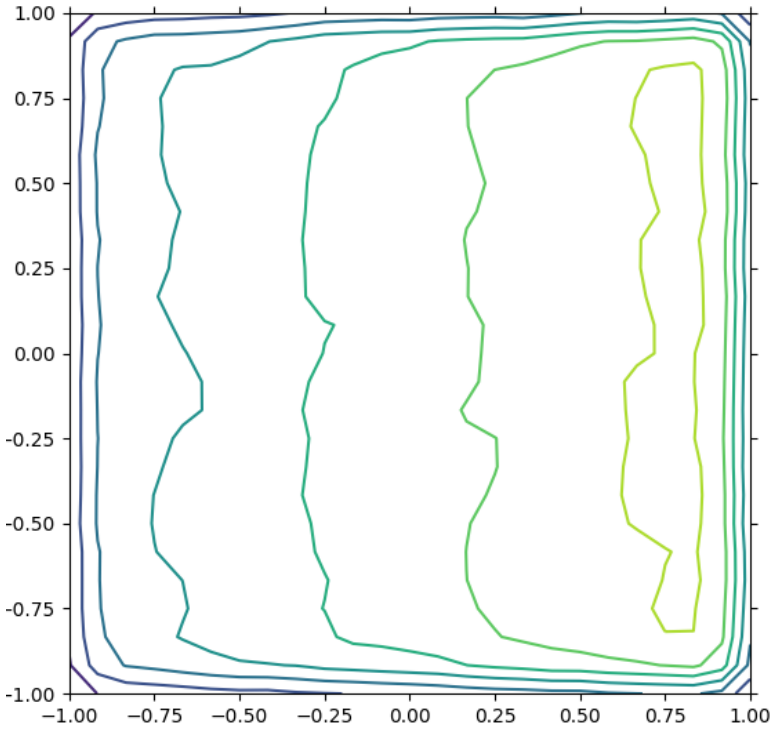

Figure 2.3: Contour plot showing the superposition of Gaussian kernels placed on each particle as reconstruction of the probability density function with $N = 2.5 \cdot 10^5$ particles and $\sigma = 0.1$.

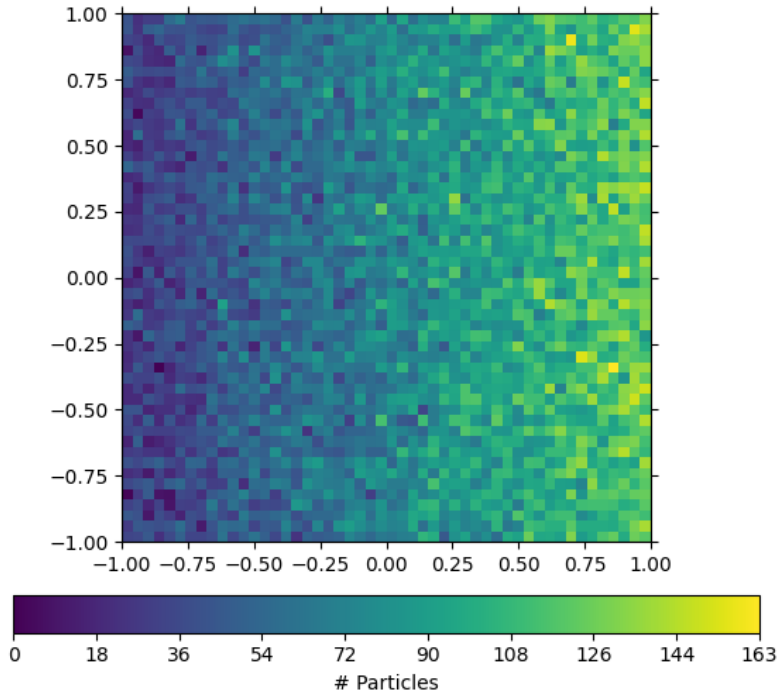

Figure 2.4: 2D histogram showing the stationary particle distribution with $N = 2.5 \cdot 10^5$ particles.

core understanding: the particle density distribution that follows the same distribution as the depth of the water column, therefore confirming the findings of the thought experiment.

3

Implementation and Analysis

3.1. Implementation of Numerical Schemes

(a) Implement the Euler and the Milstein scheme. Use these schemes for the simulation of the particles...

The particle model described in Equation 2.9 will now be solved by two different numerical methods. In any of the numerical schemes, the purpose is to progress in time and approximate the movement of the particles by integrating the particle model equations.

$$\begin{aligned} X_{t+\Delta t} &= X_t + \int_t^{t+\Delta t} \left(u + \frac{1}{H}\frac{\partial\left(HD_x\right)}{\partial x}\right)_s ds + \int_t^{t+\Delta t} \left(\sqrt{2D_x}\right)_s dW_s^x \\ Y_{t+\Delta t} &= Y_t + \int_t^{t+\Delta t} \left(v + \frac{1}{H}\frac{\partial\left(HD_y\right)}{\partial y}\right)_s ds + \int_t^{t+\Delta t} \left(\sqrt{2D_y}\right)_s dW_s^y \end{aligned} \tag{3.1}$$

These integrals are to be interpreted in Îto sense and both of the schemes aim to approximate them by several leading terms from the stochastic Taylor expansion.

3.1.1. Euler Scheme

The simplest approach for approximating the integrals is the Euler method, where the integrands are evaluated at the beginning of the interval and assumed constant for the entire time step. This is consistent with the Îto definition of the stochastic differential equation. The resulting steps in both dimensions x and y are given below.

$$\begin{aligned} X_{t+\Delta t} &= X_t + \left(u + \frac{1}{H}\frac{\partial\left(HD_x\right)}{\partial x}\right)_t \Delta t + \left(\sqrt{2D_x}\right)_t \Delta W_t^x \\ Y_{t+\Delta t} &= Y_t + \left(v + \frac{1}{H}\frac{\partial\left(HD_y\right)}{\partial y}\right)_t \Delta t + \left(\sqrt{2D_y}\right)_t \Delta W_t^y \end{aligned} \qquad (\Delta W_t = W_{t+\Delta t} - W_t) \tag{3.2}$$

This can also be interpreted as the leading terms of the stochastic Taylor approximation. This fact is elaborated on when the Milstein scheme is introduced in Section 3.1.2.

In Equation 3.2, u, v, D_x, D_y and H are simply functions of x and y. The relations were given by the problem statement of the assignment (see Equation 1.1 - 1.6). Furthermore, ΔW_t^x and ΔW_t^y are steps of independent Wiener processes, as detailed in Section 3.1.3. Finally, the derivatives in the scheme can easily be expanded as

$$\begin{aligned}\frac{\partial (HD_x)}{\partial x} &= \frac{\partial}{\partial x}\left[(15+5x)\cdot(1+\cos \pi x)\right] \\ &= 5\cdot(1+\cos(\pi x)) - (15+5x)\cdot\pi\cdot\sin(\pi x) \\ &= 5\cdot[1+\cos(\pi x) - \pi\cdot(3+x)\cdot\sin(\pi x)]\end{aligned} \tag{3.3}$$

$$\begin{aligned}\frac{\partial (HD_y)}{\partial y} &= \frac{\partial}{\partial x}\left[(15+5x)\cdot(1+\cos \pi y)\right] \\ &= -5\pi\cdot(3+x)\cdot\sin(\pi x)\end{aligned} \tag{3.4}$$

The performance of this scheme in terms of strong and weak convergence is discussed in Section 3.2.

3.1.2. Milstein Scheme

As already mentioned, the Euler scheme simply uses the leading terms of the stochastic Taylor approximation. It stands to reason that a better scheme could be constructed by specifying the most significant error terms. The term that dominates the strong order of convergence of the Euler scheme is

$$\int_{t_0}^{t}\int_{t_0}^{s} L^1 g \, dW_\tau dW_s. \tag{3.5}$$

The course notes describe in detail how this term can be broken down to reduce the error (Heemink, 2021, p.36). After applying Îto's differential rule on the integrand, one can substitute the result back in the Taylor expansion and move the terms around, in order to find an additional term to add to the Euler scheme:

$$\frac{1}{2} g(X_t, t)\frac{\partial g}{\partial x}\left(\Delta W_t^2 - \Delta t\right). \tag{3.6}$$

In the vector case that is relevant for the problem of pollutant transport, this term is of course different for both degrees of freedom. The $x-$component can be written as

$$\begin{aligned}\frac{1}{2} g_x \frac{\partial g}{\partial x}\left[(\Delta W_t^x)^2 - \Delta t\right] &= \frac{1}{2}\sqrt{2D_x}\frac{1}{\sqrt{2D_x}}\frac{\partial D_x}{\partial x}\left[(\Delta W_t^x)^2 - \Delta t\right] \\ &= \frac{1}{2}\frac{\partial D_x}{\partial x}\left[(\Delta W_t^x)^2 - \Delta t\right]\end{aligned} \tag{3.7}$$

By analogy, a similar term is found for the $y-$component, so that the Milstein scheme can be summarised as

$$\begin{aligned} X_{t+\Delta t} &= X_t + \left(u + \frac{1}{H}\frac{\partial\left(HD_x\right)}{\partial x}\right)\Delta t + \sqrt{2D_x}\,\Delta W_t^x + \frac{1}{2}\frac{\partial D_x}{\partial x}\left[\left(\Delta W_t^x\right)^2 - \Delta t\right] \\ Y_{t+\Delta t} &= Y_t + \left(v + \frac{1}{H}\frac{\partial\left(HD_y\right)}{\partial y}\right)\Delta t + \sqrt{2D_y}\,\Delta W_t^y + \frac{1}{2}\frac{\partial D_y}{\partial y}\left[\left(\Delta W_t^y\right)^2 - \Delta t\right] \end{aligned} \tag{3.8}$$

For completeness, the derivatives of the dispersion coefficients are worked out below. Recall that D_x and D_y are functions of x and y, as described in Equation 1.3 and 1.4.

$$\frac{\partial D_x}{\partial x} = \frac{\partial}{\partial x}\left(1 + \cos\left(\pi x\right)\right) = -\pi \sin\left(\pi x\right) \tag{3.9}$$

$$\frac{\partial D_y}{\partial y} = \frac{\partial}{\partial y}\left(1 + \cos\left(\pi y\right)\right) = -\pi \sin\left(\pi y\right) \tag{3.10}$$

The performance of this scheme in terms of strong and weak convergence is compared to that of the Euler scheme in Section 3.2.

3.1.3. Wiener Process

In both of the presented schemes, randomness is introduced into the simulation through Wiener processes. In particular, a realisation of the Wiener process is generated for both dimensions of each particle. By definition, a Wiener process W_t is a stochastic process with $W_0 = 0$ and increments ΔW_t that are Gaussian random variables with mean $E\left\{\Delta W_t\right\} = 0$ and variance $Var\left\{\Delta W_t\right\} = \Delta t$. An important additional property is that non-overlapping intervals are completely independent of each other. One example is visualised in Figure 3.1.

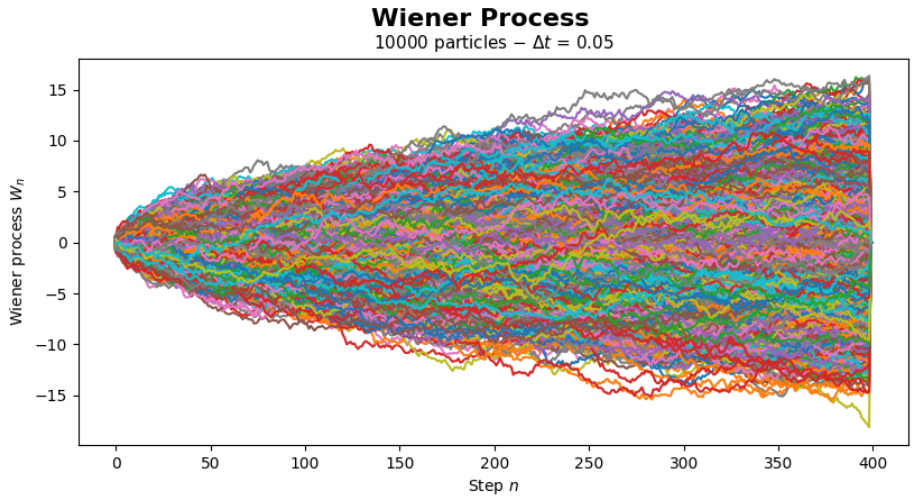

Figure 3.1: Scalar Wiener process with 10,000 particles, for 400 time steps of 0.05 seconds

3.2. Convergence

This section discusses the order of convergence of both the Euler and the Milstein scheme for the specific problem of pollutant transport. Strong and weak convergence are analysed separately.

3.2.1. Strong Convergence

(d) ... Study by means of simulations the convergence behavior of the schemes for $\Delta t \to 0$ in the strong sense. Discuss the results.

The strong order of convergence is defined by Heemink (2021) as

> *The strong order of convergence is j if there exists a positive constant K and a positive constant Δ such that for a fixed $T = N\Delta t$:*
>
> $$\begin{aligned} &\forall \Delta t, 0 < \Delta t < \Delta : \\ &\quad E\left\{|X_T - X_N|\right\} \leq K\left(\Delta t\right)^j . \end{aligned} \tag{3.11}$$
>
> *Here, X_T is the exact solution at time T and X_N is the numerical solution after N steps of ΔT.*

This means that the strong order of convergence relates to the absolute global error of all the random tracks, compared to their exact solution. The order of convergence indicates how fast this error reduces when the time step Δt of the simulation is made smaller.

One approach to finding the value of j is to generate a large number of different particle tracks and record their final position for various time steps. Then, a least-squares regression can be performed with Equation 3.11. However, several things must be noted before doing this type of analysis.

First of all, there is no way to know the actual exact solution to the particle tracks. If this were the case, there would be no use in employing numerical schemes. Therefore, the most accurate solution (smallest Δt value), will be regarded as exact. This introduces an inaccuracy into the analysis, since the absolute error of that solution is disregarded.

Furthermore, it is very important that the particle tracks with varying accuracy are compatible realisations of the Wiener process. In other words, whenever a Wiener process is evaluated at a given time t, its value must be independent of the time step Δt. At first, this requirement seems to violate the randomness and independence of the steps in the Wiener process. However, the problem can be easily solved by generating the Wiener process for the smallest possible Δt and then using integer multiples of that time step to sample the Wiener process. This idea is visualised in Figure 3.2, where one example realisation is generated for $\Delta t = 0.05$. The other tracks have different steps, but end up at the same values of W_t.

Now that a standard procedure for refinement of the time scale is established, the expectation of the absolute final error can be plotted against Δt to find the relation. When this is done for a sufficient number of Δt values, least-squares regression can be used to find the values of K and j. It is important to see that the constant K is dependent on the total simulated time. It is obvious that longer simulations will have larger global errors. However, the value of j will always be more or less the same, provided that $\Delta t < \Delta$ for all the data (this is the asymptotic region).

To collect many data points in a limited amount of processor time, it is good to use a very short simulation time, so that a small Δt can be reached for a relatively limited number of steps. Note that this does not ensure that the stationary density is reached, but that is not required to gain insight into the order of convergence. For this analysis, two seconds of the process are simulated with twenty different time steps Δt, ranging between 0.2 and 3.3 milliseconds (where

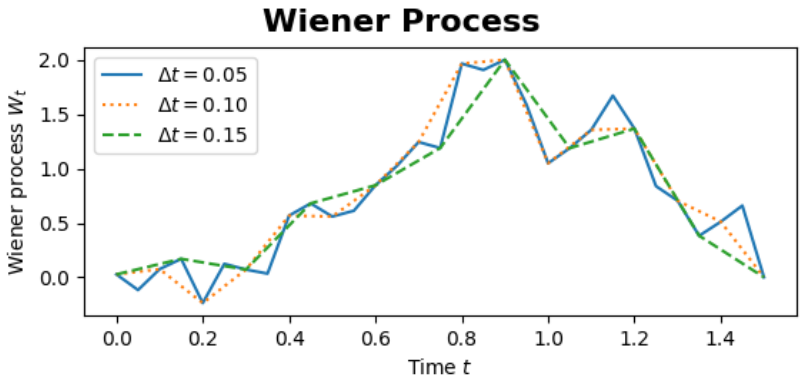

Figure 3.2: Refinement of the Wiener Process. The same process is sampled using different time steps Δt.

the Wiener processes are refined as described before). This is done using 10,000 particles and the same simulation is repeated twenty times to eliminate the influence of outliers.

The resulting data is presented with a linear scale in Figure 3.3. Note that the *Error* on the $y-$axis is defined as in Equation 3.11, where the norm is interpreted as the euclidean norm of the error vector:

$$\text{(strong) Error} = E\left\{\left\|\begin{pmatrix} X_T \\ Y_T \end{pmatrix} - \begin{pmatrix} X_N \\ Y_N \end{pmatrix}\right\|_2\right\} = \sqrt{(X_T - X_N)^2 + (Y_T - Y_N)^2}. \tag{3.12}$$

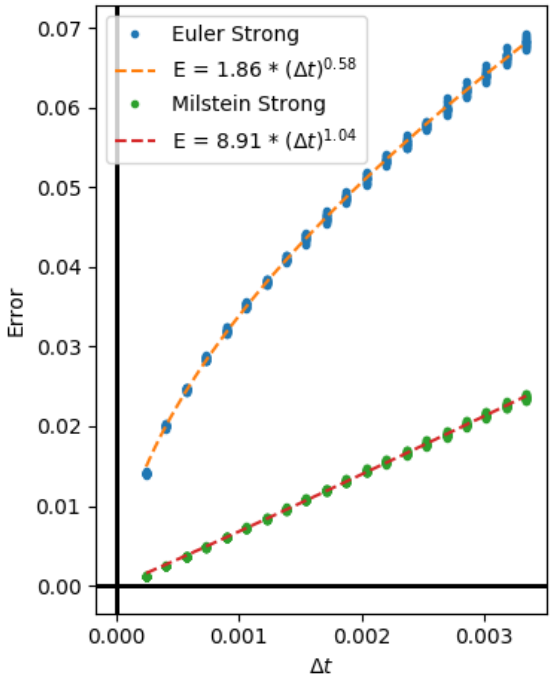

Figure 3.3: Absolute global error after 2 seconds of simulated time with varying time steps Δt. Scale is linear.

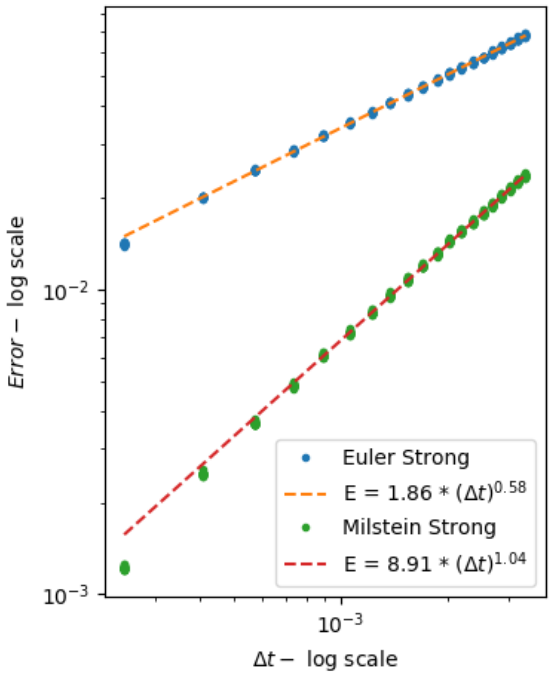

Figure 3.4: Exactly the same data as Figure 3.3, but both Euler and Milstein schemes are presented in log-log scale, to visualise the order of convergence (slope of the linear fit).

After the data was collected, least squares regression was performed to find the order of convergence j and the scaling factor K. The results are already plotted in the figure with

linear axes, however, to make this more visual and to check that all the Δt values were indeed in the asymptotic region, it is useful to also display everything on a log-log scale (as shown in Figure 3.4). After all, one can rewrite the definition for convergence as

$$\begin{aligned} E &= K \cdot (\Delta t)^j \\ \Rightarrow \log(E) &= \log(K) + j \cdot \log(\Delta t) \\ \Rightarrow \log(E) &= K' + j \cdot \log(\Delta t)\,. \end{aligned} \tag{3.13}$$

With this form of the definition, Figure 3.4 shows the order of convergence j as the slope of the curve fit and the offset $K' = \log(K)$ as the intersection with the $y-$axis. The parameters that result from the least-squares regression are shown $(\pm 2\sigma)$.

$$\text{Euler} \rightarrow \begin{cases} j & \approx \mathbf{0.580} \pm 0.003 \\ K & \approx 1.861 \pm 0.024 \end{cases} \tag{3.14}$$

$$\text{Milstein} \rightarrow \begin{cases} j & \approx \mathbf{1.039} \pm 0.004 \\ K & \approx 8.911 \pm 0.184 \end{cases} \tag{3.15}$$

It is important to note that the values of K are only shown to complete the regression results. However, they have no meaning at all with regards to the actual error. Recall that the simulation with the lowest Δt was assumed *exact* for this analysis. So, since this exact solution is itself dependent on the scheme, all errors are solely relevant within a data set where only Δt changes.

The most important information here is how fast the error decreases with decreasing time steps. It is found that the observed strong order of convergence is $\mathcal{O}\left(\Delta t^{0.58}\right)$ for the Euler scheme and $\mathcal{O}\left(\Delta t^{1.04}\right)$ for Milstein. Both of these are somewhat higher than expected from theory but definitely close enough to conclude that the implementation correctly uses these schemes. The fact that the observed order is too high, could be justified again by the error connected to the assumed *exact* solution.

For the Milstein scheme, the expected order of convergence is in most vector cases $\mathcal{O}\left(\Delta t^{0.5}\right)$, however, some very specific differential equations form an exception, where the order is actually $\mathcal{O}(\Delta t)$ (Heemink, 2021). The latter is very close to the observed $\mathcal{O}\left(\Delta t^{1.04}\right)$.

3.2.2. Weak Convergence

(e) Verify experimentally the weak order of convergence. Discuss the results.

Heemink (2021) defines the weak order of convergence as

> *The weak order of convergence is j if there exists a positive constant K and a positive constant Δ such that for a fixed $T = N\Delta t$:*
>
> $$\begin{aligned} &\forall \Delta t, 0 < \Delta t < \Delta : \\ &\quad |E\{h(X_t, T)\} - E\{h(X_t, T)\}| \leq K(\Delta t)^j \end{aligned} \tag{3.16}$$
>
> *for all functions h with polynomial growth.*

Similarly to what was done for the strong order of convergence, the norm in this definition is interpreted as the euclidean norm, so that it can be rewritten as

$$\text{(weak) Error} = \left\| E\left\{ h\begin{pmatrix} X_T \\ Y_T \end{pmatrix} \right\} - E\left\{ h\begin{pmatrix} X_N \\ Y_N \end{pmatrix} \right\} \right\|_2 . \tag{3.17}$$

Note that, with weak convergence, the norm is only taken after the expectation of all errors is computed. This intrinsically reduces the final error values and increases the order of convergence in most cases. For many Monte-Carlo simulations, this weaker convergence definition is more relevant, since one is usually not really interested in individual tracks, but rather in the properties that are observed on average.

The exact same simulation data is used as for the strong convergence, but post-processed using the weak error definition. For the Euler scheme, the results are shown in Figure 3.5. Evidently, the observation of Equation 3.13 is still valid, so a log-log plot is immediately used for more communicative output.

Notice that two different characteristics were used to measure the error, namely the position mean and the position variance. This is done by changing the function $h\left(X_t, t\right)$ in Equation 3.16 accordingly. For the mean, $h\left(X_T\right) = X_T$ and for the variance, $h\left(X_T\right) = \left(X_T\right)^2$ is used. The observed weak order of convergence is $\mathcal{O}\left(\Delta t^{1.10}\right)$ and $\mathcal{O}\left(\Delta t^{1.03}\right)$ for mean and variance, respectively. Both of these are acceptably close to the expected first order $\mathcal{O}\left(\Delta t\right)$ convergence.

Notice that the error values seem to be much more spread-out than before (see Figure 3.4). However, the lowest observed error is orders of magnitude smaller than for the strong case. This is because the norm is taken after averaging the data instead of the other way around.

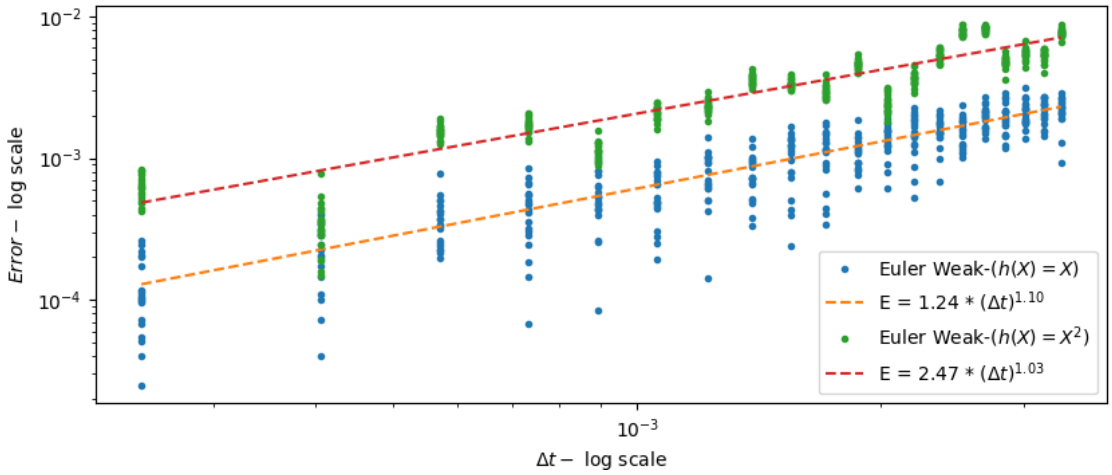

Figure 3.5: Euler Scheme - Weak error for the mean and variance of the particle position after 2 seconds of simulated time with varying time steps Δt. Scale is log-log.

Very similar results are obtained for the Milstein scheme (see Figure 3.6), where the observed weak order of convergence is $\mathcal{O}\left(\Delta t^{0.99}\right)$ for the mean $\left(h(X_T) = X_T\right)$ and $\mathcal{O}\left(\Delta t^{1.07}\right)$ for the variance $\left(h\left(X_T\right) = \left(X_T\right)^2\right)$. For Milstein, first-order accuracy was expected, so once more the implementation meets the theoretical predictions. Remarkably, the weak errors are much less spread out for Milstein than they were for the Euler scheme. This is intuitively explained by the fact that Milstein is already $\mathcal{O}\left(\Delta t\right)$ in the strong sense. As a result, the errors also deviate less from the mean.

Finally, the weak order of convergence results of the above observations can be summarised as (again with $\pm 2\sigma$):

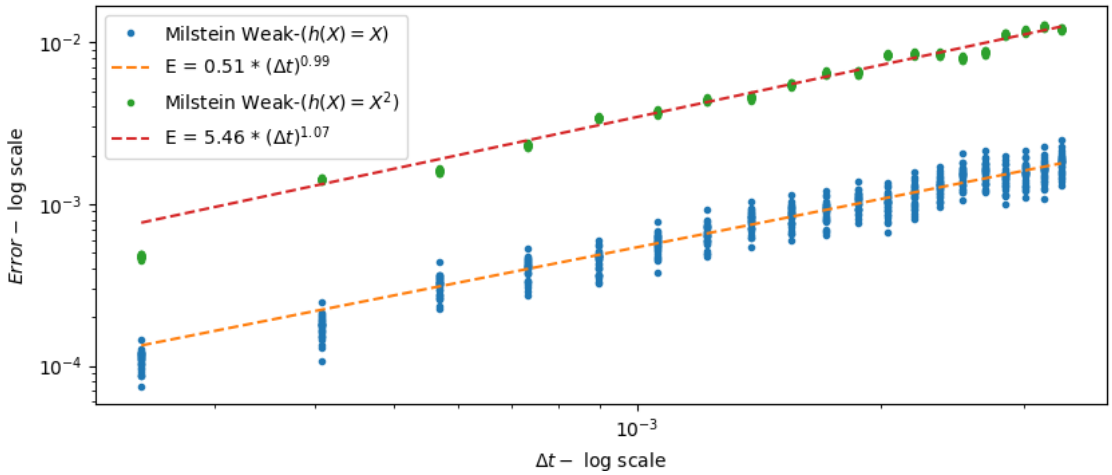

Figure 3.6: Milstein Scheme - Weak error for the mean and variance of the particle position after 2 seconds of simulated time with varying time steps Δt. Scale is log-log.

$$\text{Euler} \rightarrow \begin{cases} h(X_T) = X_T & : j \approx \mathbf{1.103} \pm 0.082 \\ H(X_T) = (X_T)^2 & : j \approx \mathbf{1.026} \pm 0.082 \end{cases} \tag{3.18}$$

$$\text{Milstein} \rightarrow \begin{cases} h(X_T) = X_T & : j \approx \mathbf{0.991} \pm 0.040 \\ H(X_T) = (X_T)^2 & : j \approx \mathbf{1.066} \pm 0.020 \end{cases} \tag{3.19}$$

3.3. Numerical Errors and Domain

(f) Because of numerical errors, particles can leave the domain of the problem. Study the performance of the various numerical schemes with respect to this numerical artefact.

In Section 2.2, it was discussed that particles have theoretically no way of leaving the domain. After all, both the velocity vector and the vector of dispersion coefficients are always perpendicular to the boundary. However, discretisation errors can occasionally cause particles to jump over the boundary anyway. After a particle has left the domain, the probability for the particle to re-enter the domain is very small and usually, these particles start drifting further away.

To understand how a particle can still leave the domain even with the given flow field and diffusion coefficients, it is important to understand how temporal discretisation influences the movement of particles. In discrete schemes like the Euler and Milstein methods presented earlier, particles are moved step-by-step and not in a truly continuous fashion. In the Euler and Milstein schemes specifically, a particle is moved based on only the information available at the particles' current position. In contrast to this, a scheme that uses information from the next, unknown, particle position can be devised, which would require solving an equation to determine the next position, analogous to a backward Euler method when solving ordinary differential equations. With the fields shown in Figure 2.1 and Figure 2.2, it can be observed that if this is the case, there will be some value for Δt for which any particle can step outside of the domain unless the particle is in a position where neither convection nor dispersion plays a role, i.e. precisely a corner of the domain. In the setting of this model of the lake and this particular initial pollution position, it is practically impossible for a particle to move into a position where it does not move at all. From observing Equation 3.2 and Equation 3.8, it can

also be seen that any and all movement terms are influenced in size by Δt: the smaller Δt, the smaller the movement per particle update. From this also follows that it is expected that the smaller Δt is, the lower the chance a particle crosses the domain boundary and diverges.

To analyse this problem numerically and to compare how much both schemes suffer from it, one approach can be to keep track of the number of particles that leave the domain. Recording this for a range of Δt values can provide insight into how often the issue presents itself. To this end, 10,000 particles were simulated using both schemes with a varying end-time T while the number of steps was kept constant at 2,000. This results in a changing Δt value while keeping equal opportunities for the particles to cross the domain. As before, every simulation was repeated 5 times to reduce the influence of outliers.

Figure 3.7 shows the percentage of particles that had left the domain by the end of the simulation. Only results where the simulation could be completed (no floating-point exceptions due to lost particles). For both schemes, there is a clear exponential trend, where the number of boundary crosses increases towards a critical Δt value, after which the simulation almost always diverges completely. Evidently, the Euler scheme is much more susceptible to this as the critical time step is less than half of what is observed for the Milstein scheme.

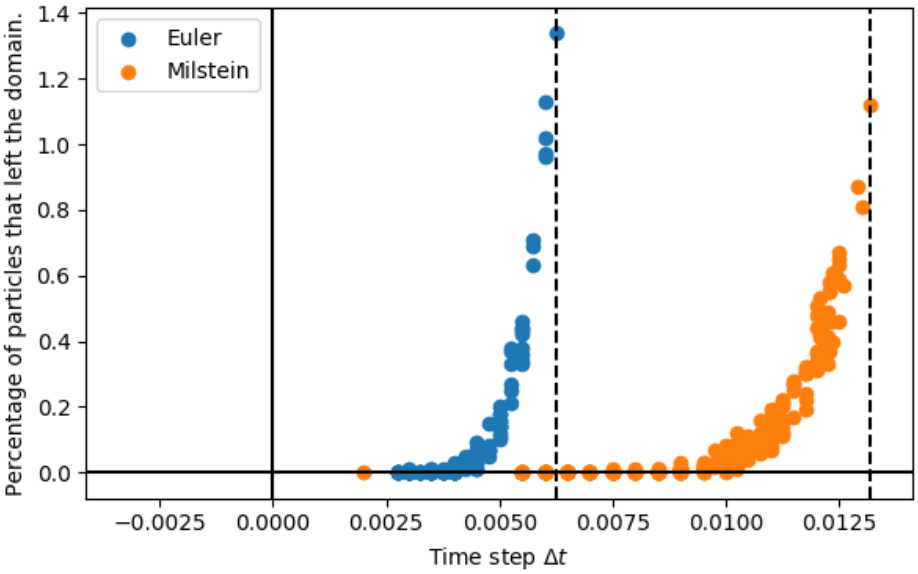

Figure 3.7: Particles that left the domain after 2,000 iterations with varying time step Δt. This is presented as a percentage of the 10,000 simulated particles.

Bibliography

Heemink, A. (2021). Numerical Methods for Stochastic Differential Equations - Course notes TW3750TU.