MATH-414: Stochastic Simulation
Pollutant Transport Rare Events

Giulia Mescolini

École Polytechnique Fédérale de Lausanne (EPFL)

3 Feb 2022

Prof. Fabio Nobile

Outline

The problem

The problem

Context: given an infinite 2D aquifer region, study the contamination of a drinking well by some particles of pollutant drifted by a current.

Model assumption: contamination happens if and only if a particle of pollutant hits the well in a given time horizon T.

Goal:

Estimate the probability that a particle of pollutant reaches the well before time T.

The problem - Notation

We will denote by τ the first passage time of the particle through the well B:

$$\tau = \inf\{t \geq 0 : (X(t), Y(t)) \in B\}$$

Given a particle starting at a given position (X_0, Y_0), the quantity we are interested in estimating is

$$\mu := \mathbb{P}(\tau \leq T | (X(0), Y(0)) = (X_0, Y_0))$$

Two strategies for the solution

1 - Stochastic Solution

The trajectories of the particles are described by the following system of stochastic differential equations:

$$\begin{cases} dX(t) = u_1(X(t), Y(t))dt + \sigma dW_1(t) & 0 \leq t \leq T \\ dY(t) = u_2(X(t), Y(t))dt + \sigma dW_2(t) & 0 \leq t \leq T \\ X(0) = X_0 \\ Y(0) = Y_0 \end{cases} \tag{1}$$

being $W_1(t), W_2(t)$ two independent standard Brownian motions and $\mathbf{u} = (u_1, u_2)$ the velocity of the current.

2 - Deterministic Solution

Another possibility is computing the solution of a backwards parabolic PDE:

$$\begin{cases} \phi_t + (\mathbf{u} \cdot \nabla)\phi + \frac{1}{2}(\sigma^2 \Delta \phi) = 0 & \text{in } D \times [0, T] \\ \phi = 1 & \text{on } \partial B \times [0, T] \\ \phi(\mathbf{x}, t) \to 0 & \text{as } |\mathbf{x}| \to \infty \\ \phi(\mathbf{x}, T) = 0 & \text{in } D \end{cases} \tag{2}$$

The probability of interest can then be obtained as:

$$\mu = \phi(\mathbf{X}(0), 0)$$

Monte Carlo Solution

First of all, we discretized the system of SDEs with the *Euler-Maruyama* scheme:

$$\begin{cases} X_{k+1} = X_k + u_1(X_k, Y_k)\Delta t + \sigma\sqrt{\Delta t} Z_k, & Z_k \sim \mathcal{N}(0,1) \\ Y_{k+1} = Y_k + u_2(X_k, Y_k)\Delta t + \sigma\sqrt{\Delta t} Z'_k, & Z'_k \sim \mathcal{N}(0,1) \\ X_0,\ Y_0 \text{ given, } Z_k \text{ and } Z'_k \text{ independent} \end{cases} \tag{3}$$

To get a sustainable computational effort, we chose $\Delta t = 10^{-2}$. In principle, this value should be suitable, as it respects the constraint $\Delta t < R^2$, but further analysis proved that it would be better to reduce it.

Introducing the random variable

$$\psi(\tau) = I_{\{\tau \leq T | (X_0, Y_0)\}} = \tilde{\psi}(Z_1, Z_1', ...)$$

The target is the estimation of

$$\mu = \mathbb{E}[\psi(\tau)] = \mathbb{P}(\tau \leq T | (X(0), Y(0)) = (X_0, Y_0))$$

The Monte Carlo setting:

- The random variable $\psi(\tau)$ has an unknown probability distribution
- We can generate i.i.d. replicas of $\psi(\tau)$

Two-stages Monte Carlo

Falling into the Monte Carlo typical setting, we implemented a *Monte Carlo estimator* $\hat{\mu}$ based on independent simulations of the particles' paths.

The number of replicas N was chosen by following the *two-stages Monte Carlo* algorithm with user-defined tolerance levels, set in such a way that variance of the first significant digit of the result is controlled.

Results

(X_0, Y_0)	*tol*	N	$\hat{\mu}_N$	$\hat{\sigma}_N^2$	α - Confidence Interval
$X_0 = 1.2, Y_0 = 1.1$	1×10^{-2}	38.005	0.4421	0.2467	$[0.4421 \pm 0.004993]$
$X_0 = 2.5, Y_0 = 2.5$	5×10^{-3}	31.430	0.0529	0.0501	$[0.0529 \pm 0.002476]$
$X_0 = 3.0, Y_0 = 4.0$	1×10^{-4}	50.000	0.0082	0.0081	$[0.0082 \pm 0.000789]$
$X_0 = 7.0, Y_0 = 7.0$	-	-	0.0000	0.0000	-

Table: Monte Carlo estimates of μ for $\Delta t = 10^{-2}$

Order of the discretization in time

Then, we analyzed the error as a function of Δt. We computed it using as reference exact solution the numerical one.

We ran simulations *on the same Brownian paths* starting from $\Delta t = 10^{-2}$ and performing 3 refinements, while keeping N fixed to 10.000.

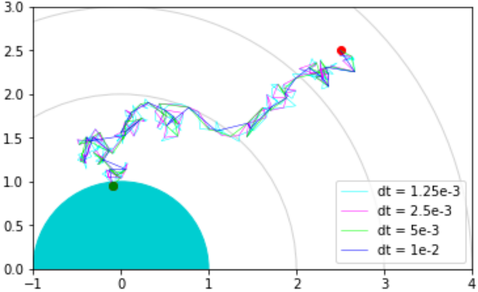

Figure: Trajectories with different resolution.

Error trends

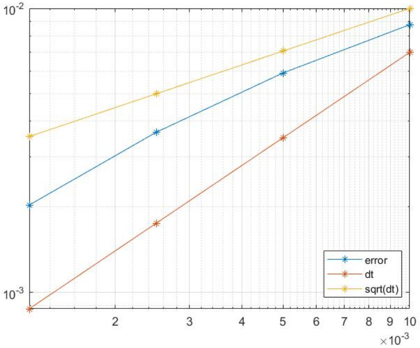

Figure: Plot of the error as a function of Δt, in logarithmic scale.

As $\Delta t \to 0$, we registered an empirical order of 1, which is coherent with the theory on the *Euler-Maruyama* method.

Comparison with numerical solution

We solved numerically the backwards PDE, exploiting the Python's library for finite elements `FeNiCS`.
We adopted the following strategy:

- *Implicit* scheme for the time discretization
- $\mathbb{P}_1^C$ finite elements

We can compare the results obtained with the finest time resolution ($\Delta t = 1.25 \times 10^{-3}$) with the numerical solution:

(X_0, Y_0)	μ_{num}	$\hat{\mu}_N$
$X_0 = 1.2, Y_0 = 1.1$	0.5063	0.4870
$X_0 = 2.5, Y_0 = 2.5$	0.0615	0.0612
$X_0 = 3.0, Y_0 = 4.0$	0.0095	0.0098
$X_0 = 7.0, Y_0 = 7.0$	4.5635×10^{-7}	0.0000

Variance Reduction

Antithetic Variables

We implemented the *Antithetic Variables* technique for variance reduction. Note that the underlying setting is suitable for the application of this algorithm since we fall into the hypothesis of the proposition below.

Proposition

- the random variable $\psi(\tau)$ is a function of the vector of Gaussian increments $\mathbf{Z} := \{(Z_k, Z'_k)\}_k$
- $\{(Z_k, Z'_k)\}_k$ are *independent* and have a *symmetric distribution* around their mean
- the function representing the relationship between ψ and $\mathbf{Z}$ is monotone *non-increasing* in $\mathbf{Z}$

Antithetic Paths

The antithetic paths generated are of the form:

$$\begin{cases} X_{k+1} = X_k + u_1(X_k, Y_k)\Delta t + \sigma\sqrt{\Delta t} Z_k \\ Y_{k+1} = Y_k + u_2(X_k, Y_k)\Delta t + \sigma\sqrt{\Delta t} Z'_k \end{cases}$$

$$\begin{cases} \tilde{X}_{k+1} = \tilde{X}_k + u_1(\tilde{X}_k, \tilde{Y}_k)\Delta t - \sigma\sqrt{\Delta t} Z_k \\ \tilde{Y}_{k+1} = \tilde{Y}_k + u_2(\tilde{X}_k, \tilde{Y}_k)\Delta t - \sigma\sqrt{\Delta t} Z'_k \end{cases}$$

with $Z_k \sim \mathcal{N}(0,1)$, $Z'_k \sim \mathcal{N}(0,1)$

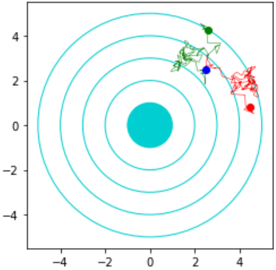

Figure: Antithetic Paths

Reduction of variance achieved

Method	$\hat{\mu}$	$\hat{\sigma}^2$	α - Confidence Interval
Crude Monte Carlo	0.0532	0.0503	$[0.0531 \pm 0.00248]$
Antithetic Variables	0.0514	0.0231	$[0.0514 \pm 0.00237]$

Table: Comparison between Crude Monte Carlo and Antithetic Variables in terms of output and variance.

Reduction of length of the Confidence Interval:

$$\frac{\ell_{MC}}{\ell_{AV}} = 1.04491 > 1$$

Rare Events Simulation

Problem

What happens if we consider a starting point which is very far from the well?

- As previously seen, traditional Monte-Carlo technique do not perform well and output always $\hat{\mu} = 0$.
- **Solution:** **splitting method** for rare events simulation.

Splitting Method

The method

The splitting method is based on the following principle:

Divide et impera

The probability of the rare event μ is viewed as the probability of an intersection of a nested sequence of events, and it is computed as a product of conditional probabilities which can be estimated more accurately.

The splitting algorithm

1. Partition the domain into a sequence of concentric circles $B = C_m \subset C_{m-1} \subset \cdots \subset C_1$.
2. Set s_i, the number of new paths to generate from each element of S_{i-1} (set of all valid candidate starting points in the previous level).
3. For each level C_i, for each valid starting point, generate s_i paths.
4. If the path hits the following level before time T, store the hitting coordinates in S_i; before passing to the next level, compute the fraction of successful hits c_i.
5. When all levels are over, estimate $\hat{\mu} = \prod_{i=1}^{m} \hat{c}_i$.

The splitting algorithm

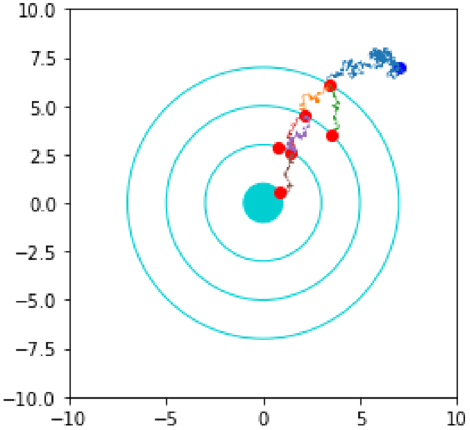

Figure: Splitting Method

Choice of the parameters

- **How many circular crowns?**
  $\Rightarrow$ using the optimal value for the case in which the success probabilities do not depend on the entrance state into that level, we got
  $$m^* = \frac{-log(\mu)}{2} \sim 7$$
  using as exact value for μ the numerical solution of the PDE.
- **How to choose s_i?**
  $\Rightarrow s_i := \lceil 1/p_i \rceil$, being $p_i = \mathbb{P}(\tau_i \leq T | \tau_{i-1} \leq T)$.
  This avoided an explosion of the computational effort, and the p_i were estimated with a pilot run.

Fixed Effort Method

Another way to control the computational effort is through the *Fixed Effort Variant* of the Splitting Method, consisting in creating at every stage a fixed total number of offspring.

Note that in this case s_i is automatically determined by the fixed effort.

Results

	Fixed Splitting	Fixed Effort	Numerical Solution
$\hat{\mu}$	5.1786×10^{-7}	4.9474×10^{-7}	4.5635×10^{-7}

Table: Estimated probabilities with the two variants of the Splitting Method and 7 circular crowns; $\Delta t = 10^{-3}$.

References

Garvels, MJJ. "The splitting method in rare event simulation". PhD thesis. Univerisry of Twente, 2000.
Kroese, DP., T. Taimre, and IB. Zdravko. *Handbook of Monte Carlo Methods*. Vol. 706. John Wiley and Sons, 2013.
Nobile, F. *Stochastic Simulation. Lecture Notes*. 2021.

Questions?

Back-up Slides

Back-up slides

- Link to source code:
  https://github.com/giuliamesc/pollutant_transport
- *Importance Sampling vs Splitting*: the similarity is based on the fact that *Splitting* involves a change of probability measure (switch to the conditional probability).
- Comparison stochastic vs numerical solution, with the same $\Delta t = 10^{-2}$:

(X_0, Y_0)	μ_{num}	$\hat{\mu}_N$
$X_0 = 1.2, Y_0 = 1.1$	0.5063	0.4421
$X_0 = 2.5, Y_0 = 2.5$	0.0615	0.0529
$X_0 = 3.0, Y_0 = 4.0$	0.0095	0.0082

Back-up slides

Algorithm 1 Two stages Monte Carlo

1: Fix the number of replicas for the pilot run, the desired tolerance and the level of confidence: $\bar{N} = 1000$, tol (depending on the starting position, see subsection 2.2), $\alpha = 0.05$.
2: Perform a pilot run with $\bar{N}$ replicas $(\psi^{(1)}, ..., \psi^{(\bar{N})})$ and compute:

$$\hat{\mu}_{\bar{N}} = \frac{1}{\bar{N}} \sum_{i=1}^{\bar{N}} \psi^{(i)} \quad \hat{\sigma}^2_{\bar{N}} = \frac{1}{\bar{N}-1} \sum_{i=1}^{\bar{N}} (\psi^{(i)} - \hat{\mu}_{\bar{N}})^2$$

3: Fix

$$N = \frac{c^2_{1-\alpha/2} \hat{\sigma}^2_{\bar{N}}}{tol^2}$$

being $c_{1-\alpha/2}$ the quantile of order $\alpha/2$ of a standard normal.
4: Generate a run with N replicas $(\psi^{(1)}, ..., \psi^{(N)})$ and output $\hat{\mu}_N, \hat{\sigma}^2_N$
5: **if** $\hat{\sigma}^2_N > \hat{\sigma}^2_{\bar{N}}$ **then**
6: Set $\bar{N} = N$ and go back to 2
7: **else**
8: Output $\hat{\mu}_N$ with its confidence interval[2] $\hat{I}_{\alpha,N} = \left[\hat{\mu}_N \pm c_{1-\alpha/2} \frac{\hat{\sigma}_N}{\sqrt{N}}\right]$
9: **end if**

Figure: Two stages MC Algorithm

Back-up slides

Algorithm 2 Splitting Algorithm

Choose in the domain D a sequence of concentric circles $B = C_m \subset C_{m-1} \subset \cdots \subset C_1$, where each C_i has radius R_i, with $R_i > R_{i+1}$.

For each level C_i, set s_i, the number of new paths to generate from each element of S_{i-1}, the set of all valid candidate starting points in the previous level (setting $S_0 = \{(X_0, X_0)\}$).

for all levels C_i, with $i = 1, \ldots, m-1$ **do**

  for all valid starting points in the previous level **do**

    Generate s_i paths.

    if a path hits C_{i+1} before the limit time T [4] **then**

      Increase a counter n_i and store the hitting coordinates in S_i.

    end if

  end for

  Being $\tau_i = \inf\{t \geq 0\ (X(t), Y(t)) \in C_i\}$, estimate the probability $\mathbb{P}(\tau_i \leq T | \tau_{i-1} \leq T)$ with

$$\hat{c}_i = \frac{n_i}{|S_{i-1}| s_i}$$

end for

Estimate μ with the unbiased estimator $\hat{\mu} = \prod_{i=1}^{m} \hat{c}_i$.

Figure: Splitting Algorithm