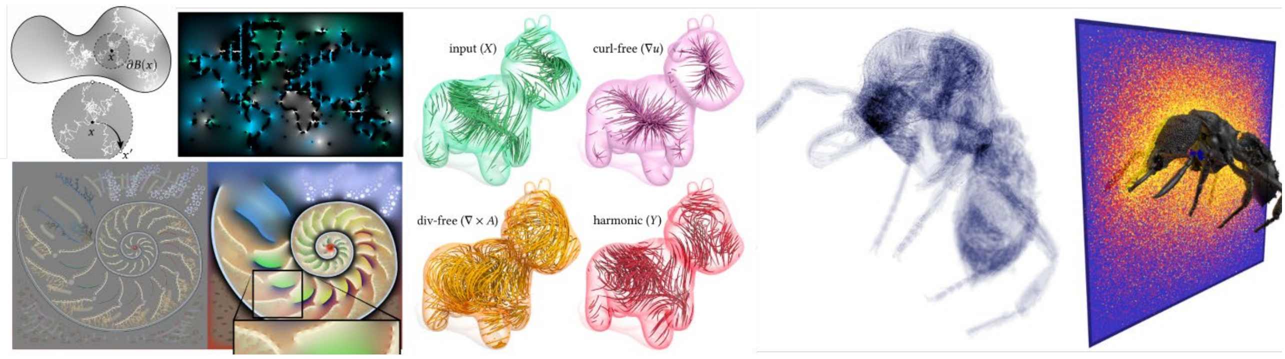
Monte Carlo Geometry Processing

Large-scale equation solves are always great enemies for efficiency and stability of graphics applications. Instead of tessellating a domain with tons of elements and solving associated discretized equations, stochastic approach formulates the solution to a PDE as the expectation of stochastic process. This surprising result does not only reveal a deep connection between PDEs and stochastic process in theory, but also offers us a Swiss knife for dealing with practical problems, which fits remarkably well into the modern parallel computing environment.



Objectives

In this project, you will implement the Walk-on-Sphere (WoS) algorithm, an efficient method to simulate random walks, for solving a variety of PDEs such as Laplace equation, Poisson equation and biharmonic equation etc [1]. Instead of constantly looking at boring numbers, you will see their enormous amazing applications in computer graphics, ranging from 2D image synthesis/editing to 3D shape/field modeling. You are highly encouraged to apply your WoS implementation to solve different tasks and push the performance of your code to limit as well, for instance by developing a GPU-based implementation for massive parallelization [2]. During the development, you will soon observe the error produced by taking only a limited number of random walks. To efficiently reduce notable artifacts while still keeping your program fast, you may need to analyze the source of error arising in stochastic simulations, and then explore some possible ways to effectively reduce errors yet without a sharp increase in computational cost.

Supervisor: Jiong Chen (jiong.chen@telecom-paris.fr), Office 5B20

References

[1] Sawhney, Rohan, and Keenan Crane. "Monte Carlo geometry processing: A grid-free approach to PDE-based methods on volumetric domains." ACM Transactions on Graphics 39.4 (2020).

[2] Mossberg, Linus. "GPU-Accelerated Monte Carlo Geometry Processing for Gradient-Domain Methods." (2021).