# COMP 477: Project Report

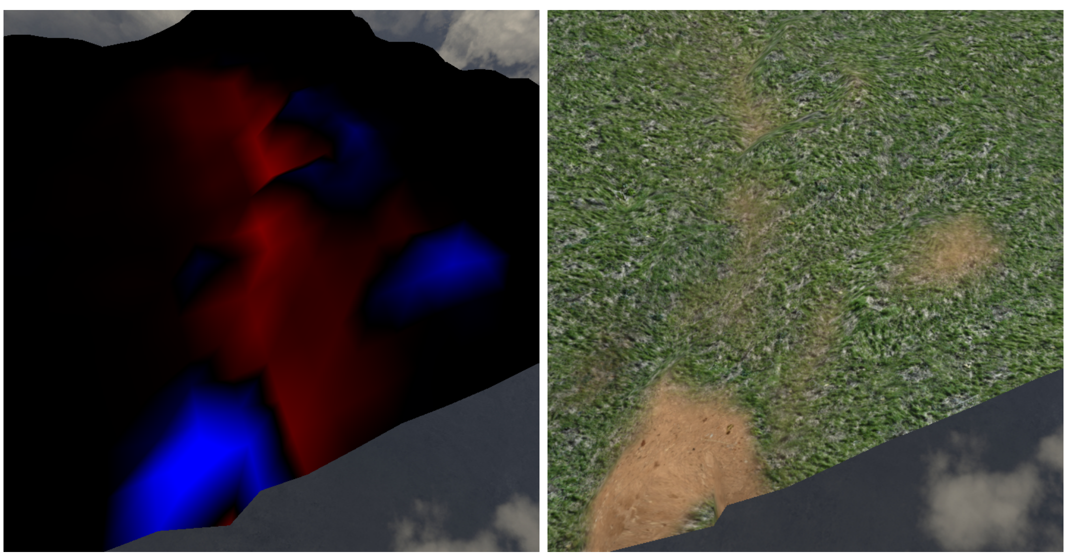
**Date:** December 14th, 2023
**Team:** Team 1 - Laurent Voisard (40176186), Étienne Racine (40167163)
**Project:** Hydraulic Simulation Model

## Project Description

This project was inspired by a research paper by Hydraulic Erosion Using Smoothed Particle Hydrodynamics which involves simulating hydraulic erosion using smoothed particle hydrodynamics (SPH) fluid simulation. We aimed to deliver an OpenGl application that could perform something similar.

### Idea

The erosive effects of fluid flow on terrain can be simulated by using SPH fluid dynamics and computing the sediment intake, diffusion and deposition for each particle. The bottom boundary (terrain) of the simulated area is dynamically adjusted with respect to this, in turn affecting how other fluid particles will behave.
In essence, we add imaginary sediment particles that are tied to the regular SPH particles as well as terrain particles, at each vertex position of the terrain. These sediment particles take sediment from the terrain particles, and diffuse it with its neighbors.



Eroded Sediment (Red), Deposited Sediment (Blue), Sediment Change (Dirt)

The sediment eroded from the terrain is a function of the particle's linear velocity, the distance between the SPHparticle and the boundary particles, and the terrain's critical shear value, in other words, how hard the material is. For the sake of simplicity, the terrain has the same hardness at any given place.

# Development

## Starting Point

Since Laurent had already completed a similar project in COMP 438 (geometric processing) which involved using a "piping model" to simulate water behavior, we decided to base our project on this existing codebase.

This was initially great, as it enabled us to directly focus on the core idea of the project, that is, SPH simulation and erosion, without needing to worry about the boilerplate code of setting up a new OpenGL application. However, this did lead to somewhat arbitrary restrictions down the road. The existing codebase made strict assumptions on how terrain had to be represented and accessed, making it difficult to adjust the density of boundary particles that composed it.

We decided to stick by these restrictions as removing them would have involved altering and re-writing significant parts of the existing codebase. We figured that the final results were still quite good and that our time and efforts were best spent elsewhere.

## Loss of Team Member

The largest hindrance we came across during this project revolved around our third team member, which had to be cut off due to lack of participation, followed by plagiarism. Considering the other group work of this course impacted by this loss (studio showcase, state-of-the-art analysis), this single handedly turned our project into something ambitious at 3 people to effectively impossible at 2. We estimate our time loss linked to this incident to be around 2 weeks. An extension of the same amount of time was given to us, however during the final exam period, our attention was divided between this project and exams so we couldn't recover all of the time lost.

## Erosion Model Simplification

The research paper we based ourselves on uses complex, physically accurate ways of determining the rate of sediment eroded from the terrain by particles moving over it and describes how particles exchange sediment between themselves. However, the paper was not particularly explicit in describing the sediment particles, especially when it came to the sediment pressure pressure parameter, resulting in a function giving undefined values. The sediment transfer from gravity could not be completed, but the sediment diffusion between particles still gave results that were believable. This is not as accurate as the original paper, but was relatively much simpler to implement.

# Results

For its rough development, we believe that this project yields good results. Although purposefully exaggerated for demonstration purposes in the showcased videos, the erosion resulting from the fluid's traversal looks believable.
The project stands as a nice proof of concept that could be expanded if desired.

## Accomplished

In this project, we implemented the following:

- **Smoothed Particle Hydrodynamic (SPH) fluid simulation**. Despite this not being the direct focus of our project, we think that the simulated fluid flow down slopes is quite nice.

- **Sediment erosion, diffusion and deposition logic** (albeit simplified).

## Dropped

We decided to drop the following objectives due to time constraints:

- **Fluid sources and sinks**. We initially wanted the system to be able to dynamically add and remove SPH particles to simulate constant fluid flow as opposed to single bursts. We are able to add particles dynamically, but the water source was not added in.

- **Real-time terrain editing**. We wanted the user to be able to add or remove terrain, enabling users to quickly create custom simulation conditions, rather than relying on externally-created height-maps. This was unrealistic because of the static nature of the implementation of the neighbour search of SPHparticles. The search grid bounded the terrain, so adding terrain and removing it would have the terrain exit this bound, thus our particles could not make contact with the boundary particles.

# References


Koschier, D., Bender, J., Solenthaler, B., & Teschner, M. (2020). Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. *arXiv preprint arXiv:2009.06944*.

Krištof, P., Beneš, B., Křivánek, J., & Šťava, O. (2009, April). Hydraulic erosion using smoothed particle hydrodynamics. In *Computer graphics forum* (Vol. 28, No. 2, pp. 219-228). Oxford, UK: Blackwell Publishing Ltd.

https://github.com/SebLague/Fluid-Sim