An Investigation into the Effects of Environmental Deformation Over Time Using Node-Based River and Sediment Simulation

## Abstract

I intend to create a C++ OpenGL program to visually represent river formation, sediment transfer, erosion, and rainfall’s effects on landscape over time to the user. It should allow specification of values for average temperature, probability of mountains, probability of natural water sources, and simulation time steps. I aim to use a node-based system and hydraulic erosion model(1) to create a simple yet very realistic representation of river formation, based on the key concepts of fluid dynamics. This will be a portfolio project, showing my ability to implement and iterate on complex algorithms to simulate a real-life scenario. I will also investigate the benefits of running such simulations on both the GPU and CPU of the computer, and the complexity and advantages of both.

## Introduction and Rationale

Simulation of rivers has always proven difficult when creating a virtual landscape. Several models exist to mimic the flow of water (The Lattice-Boltzman algorithm(2) is a commonly used example.) Many approaches to water simulation stray away from particle-or-node based systems, favouring vector-based approximations to calculate both path and flow(3).

Node-based flow simulations allow a virtual representation of deformation of landscape features, and the transfer of sediment(4). Such a system could, over time, allow for the creation of complex geographical features found in rivers (such as ox-bow lakes(5)) and allow for natural-looking landscapes to be created by running the simulation on any terrain.

In this portfolio project, I will create a tool that allows for the generation of a landscape, which is then deformed over time by the movement of water- simulating both natural springs and rainfall. This should provide an environment for the natural generation of real-world features that are sparsely seen in digital representations of rivers. Features like bank erosion and sediment transfer(6) are rarely present in these models due to their complexity(7) and computational intensity(8).

## Literature Review

Various studies into the potential paths of rivers exist for flooding avoidance(9). As flooding poses a significant risk in some areas, they are regularly monitored and simulated to prevent damage to infrastructure(10). While my aim is not to simulate existing rivers and instead focus on the landscape deformation over many years, pre-existing tools for flood avoidance can prove to be useful sources for my investigation, as they tackle similar concepts(11).

A key focus for my studies is the hydraulic erosion model(1). This will provide the bulk of algorithmic function to simulate water and sediment movement and functions as the core concept of my research. I will take the existing algorithm and iterate on it to support multiple types of sediment in a soil map.

A study into the erosion of landscape by Nicholas McDonald(4) tackles some similar concepts to my investigation. However, this study lacks a soil map to allow for different densities of soil and sediment for its nodes. I intend to use a soil map for increased real-world accuracy, tracking different forms of sediment and their transfer(12).

Studies into fluid dynamics can provide a digital representation of a range of features(13). Editors have been created to simulate physics in games through real-time editing(14), studies into terrain generation based on pre-existing river splines exist(15), and forms of fluid dynamics models have existed for centuries, since the time of Archimedes(16).

Studies also exist into the usage of real-time terrain creation and destruction, and the effect of it on water flow(17). However, I intend to use pre-generated landscapes as opposed to real-time generation, so this is not something to consider for this project.

## Aims

My aim is to create a tool that generates a landscape and manipulates it over a user-defined period of time to form rivers and lakes through simulation of fluid dynamics, including the pick-up and deposit of sediment.

The tool will allow the user to generate a random landscape with height differences, a simple representation of foliage, and soil maps. It will then step through various periods of time, simulating the effect that fluid would have on the landscape through means of erosion and sediment deposit. Fluids should be able to both flow and form pools, as they commonly do in traditional fluid dynamics(18), and transfer sediment from banks and cliffsides. This will be done through a node-based simulation, considering differences in height and density to create the effects of erosion over time. The precision of the model should be variable, allowing for both large and small-scale simulation, from a single riverbank to acres of land.

The simulation is not planned to account for the effect of animals on the landscape, the freezing and thawing of water, or the effects of altitude on the behaviour of fluids and gasses. The inherent complexity of these features would require a full study(19) and would prove difficult to simulate in the scope of my project.

## Objectives

My key objectives are:

* To manipulate the existing hydraulic erosion formula to allow specification of a soil map, with varying kinds of sediment.
* To create a tool that generates a random landscape with varying terrain height.
* To simulate the formation of rivers and flow of water over the landscape, including the effect of rain flowing and pooling without use of a pre-defined river spline(20).
* To simulate the erosion of terrain and deposit of sediment to manipulate the landscape over time.
* To evaluate the advantages and disadvantages of running simulation on GPU and CPU of the computer.
* To allow specification of properties, so the user can manipulate both terrain generation and water behaviour to affect the simulation.
* To compare the results of this updated model with real-life geographical data.

These should combine to allow creation of complex geographical features that are not usually seen in digital landscapes.

## Methodology

For this project, I intend to employ a node-based system of terrain generation and simulation. Nodes will contain information regarding the soil and rock densities at different depths, the depth of any water present at that node, and the current flow direction and speed of the water. Randomization of initial terrain generation will use Perlin Noise, manipulated based on user specification of average landscape height and mountain formation chance.

Algorithms for cell-based fluid dynamics will be used, such as the hydraulic erosion model(1). This method will need to be modified to provide varying sediment values and erosion rates due to the soil map providing varying densities of sediment. The hydraulic erosion model should provide sufficient basis for simple water movement and erosion, however.

The algorithm consists largely of five major steps, calculating the individual elements of fluid simulation. The algorithm is designed with GPU-based optimization in mind, but I will experiment with both CPU and GPU-based implementations of this algorithm in terms of complexity and runtime efficiency. The project will be a C++ OpenGL application in Visual Studio 2019.

The system will iterate through all nodes in the scene (stored in a data structure), performing various operations to simulate the movement of water through the landscape.

Firstly, rain simulation will be performed, adding a small amount of fluid to multiple nodes in the world to allow for additional deformation of landscapes that are not naturally near water, and allowing water movement in locations that might not see a regular flow. This could potentially lead to pools and new rivers forming. Rainfall averages will be specified by the user but will vary as they do naturally.

The system will individually check each simulated node and calculate any deformation, sediment pick-up or loss, and changes to flow, using existing fluid dynamics methodology for calculation of frictional force(21) and the hydraulic erosion model.

It will then edit the landscape data accordingly. This, in turn, will affect the visual representation of the landscape drawn using OpenGL, connecting nodes with simple quads (cells). Two render passes will be made- one for the landscape, and one for the water. Visual fidelity is not the focus of the project, so simple designs will be used in this regard.

When water and sediment simulation is complete, evaporation occurs, leaving sediment deposits in positions where water containing sediment no longer exists. This will affect landscape data and increase land height values in associated areas.

This cycle should allow sufficient simulation to create geographical features that are often missing from digital representations of rivers. For example, sediment pick up, deposit, and erosion should combine to allow the creation of ox-bow lakes.

## Project Plan

See the table below for my estimated completion date of key milestones (allowing for one week of flexible time in case of unforeseen circumstance.) The dissertation will include both the evaluation of CPU and GPU simulation and the comparison with geographical data and will be written at the same time that these are taking place.

|  |  |
| --- | --- |
| **Milestone** | **Estimated Completion Date** |
| Simple terrain creation and display | March 7th |
| Random terrain generation & soil map | March 14th |
| Implementation of hydraulic erosion model (CPU) | March 28th |
| Implementation of hydraulic erosion model (GPU) | April 11th |
| Implementation of user-defined generation and simulation properties | April 25th |
| Evaluation of CPU vs GPU simulation | May 2nd |
| Comparison with real-life geographical data | May 9th |

# References

1. Mei, X., Decaudin, P. and Hu, B.-G. (2007). Fast Hydraulic Erosion Simulation and Visualization on GPU. 15th Pacific Conference on Computer Graphics and Applications. Available at: <https://hal.inria.fr/inria-00402079/document>.
2. Schroeder, D. (2019). Fluid Dynamics Simulation. Available at: <https://physics.weber.edu/schroeder/fluids/>.
3. Peric, F., Ferziger, J.H. (2002). Computational Methods for Fluid Dynamics. Available at: <http://servidor.demec.ufpr.br/CFD/bibliografia/Ferziger%20Peric%20-%20Computational%20Methods%20for%20Fluid%20Dynamics,%203rd%20Ed%20-%202002.pdf>
4. McDonald, N. (2020). Procedural Hydrology: Dynamic Lake and River Simulation. Available at: [https://nickmcd.me/2020/04/15/procedural-hydrology/](https://nickmcd.me/2020/04/15/procedural-hydrology/%20)
5. National Geographic Society (2012). Oxbow lake. National Geographic Society. Available at: <https://www.nationalgeographic.org/encyclopedia/oxbow-lake/>
6. Defra, Environment Agency (2008). Flood and Coastal Erosion Risk Management. Available at: <https://www.therrc.co.uk/MOT/References/EA_DEFRA_Sediment_transport_and_alluvial_resistance_in_rivers.pdf>
7. Kipfer, P., Westermann, R. (2006). Realistic and Interactive Simulation of Rivers. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.449.5576&rep=rep1&type=pdf>
8. McLeod, D., Chen, H., Hu, N. (2020). Interactive Hydraulic Erosion Simulator. Available at: <https://huw-man.github.io/Interactive-Erosion-Simulator-on-GPU/>
9. Hadimlioglu, I.A., King, S.A. and Starek, M.J. (2020). FloodSim: Flood Simulation and Visualization Framework Using Position-Based Fluids. Available at: <https://www.mdpi.com/2220-9964/9/3/163/pdf>
10. Waterco. (2020). River Modelling. Available at: https://waterco.co.uk/river-modelling/
11. Cole, S., Robson, A. and Moore, B. (2010). The Grid-to-Grid Model for nationwide flood forecasting and its use of weather radar. Available at: <http://nora.nerc.ac.uk/id/eprint/10422/1/Cole_et_al_Bristol_Radar_Workshop_28_July_2010.pdf>
12. Geosciences LibreTexts. (2015). 3.1: Sources and Types of Marine Sediment, Oceanography. Available at: <https://geo.libretexts.org/@go/page/760>
13. Grau-Bové, J., Mazzei, L., Strlic, M. and Cassar, M. (2019). Fluid simulations in heritage science. Available at: <https://heritagesciencejournal.springeropen.com/articles/10.1186/s40494-019-0259-9>
14. Grenier, J-P., Tokarev, K. (2018). River Editor: Water Simulation in Real-Time. Available at: <https://80.lv/articles/river-editor-water-simulation-in-real-time/>
15. Génevaux, J.-D., Galin, É., Guérin, E., Peytavie, A. and Beneš, B. (2013). Terrain generation using procedural models based on hydrology. ACM Transactions on Graphics. Available at: <https://hal.archives-ouvertes.fr/hal-01339224/document>
16. Archimedes and Heath, (2009). ON FLOATING BODIES, BOOK I. Cambridge University Press. Available at: <https://www.cambridge.org/core/books/abs/works-of-archimedes/on-floating-bodies-book-i/256198AE9365D4B099BF23A7BBD50D41>
17. Yu, Q., Neyret, F., Bruneton, E. and Holzschuch, N. (2009). Scalable real-time animation of rivers. Available at: <http://maverick.inria.fr/Publications/2009/YNBH09/riversEG09.pdf>
18. Hutter, K., Wang, Y., Chubarenko, I. (2014). Physics of Lakes. Available at: <http://www.jlakes.org/config/hpkx/news_category/2015-06-01/PhysicsofLakesVolume3MethodsofUnderstandingLakesasComponents.pdf>
19. Shahid, F., Hussain, M., Baig, M.M. and Haq, I. ul (2017). Variation in aerodynamic coefficients with altitude. Results in Physics. Available at: <https://www.sciencedirect.com/science/article/pii/S2211379717302437>.
20. Unreal Engine. (2016). Landscape Splines. Available at: <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/Landscape/Editing/Splines/>
21. Columbia University. (n.d.). Fluid Dynamics. Available at: <https://www.ldeo.columbia.edu/~martins/hydro/lectures/fluid_dynamics.html>