university of groningen

faculty of science and engineering

Stochastic Simulation of Porous 3D Structures with Realistic Grain Shapes

Bachelor's Project Computing Science

July 2025

Author: Joël Kiers
Student Number: s4042204
First supervisor: Dr. Christian Kehl
Second supervisor: Dr. Johannes Miocic

Abstract

This thesis presents the development of a stochastic 3D simulator for modeling porous geological structures using realistically shaped grains. Unlike many existing approaches that assume spherical or ellipsoidal particles, this simulator combines the cooperative rearrangement packing algorithm with grain geometries derived from real 2D contours, represented using implicit functions. The software allows control through parameters such as grain size, grain distribution, and target porosity. The spherical grain model is validated by comparing its porosity results with those from another depositional model. While the realistic grain model is limited by computational constraints that hinder large-scale simulation, it produces complex shapes whose geometries are analyzed in detail.

Acknowledgements

I would like to express my sincere gratitude to Dr. Christian Kehl for helping me start up a Bachelor's project after a time of academic inactivity as well as his guidance, constructive feedback, and continuous support throughout the project. I would also like to thank my friends and family for their support during challenging times throughout the duration of this project. Their encouragement made it possible for me to keep moving forward.

Contents

List of Figures

List of Tables

1 Introduction

Geological modelling is of paramount importance for the comprehension of the subsurface. It is a key process for locating and understanding reservoirs of oil, gas and minerals. The development of modern energy systems, water management systems, waste disposal and even carbon capture and storage is facilitated by geological modelling as well. A variety of techniques are employed in the creation of geological models.

One such methodology is geometric grain-based modelling. In this process, individual particles are generated and simulated to form a larger structure made up of solid 'grains' and voids called 'pores'. The resulting geometry can be rendered into images or analyzed digitally.

Figure 1 shows a sample of a thin-section image of real sedimentary rock. It reveals that the rock is comprised of solid grains and pores. Note that the structure consists of various grain shapes. This thesis outlines the development of a stochastic simulator based on the geometric, grain-based modelling approach. The emphasis is on the combination of a packing algorithm known as the cooperative rearrangement algorithm with complex, plausibly-realistic grain shapes to generate images of 3D structures and investigate the effect of grain shape on porosity.

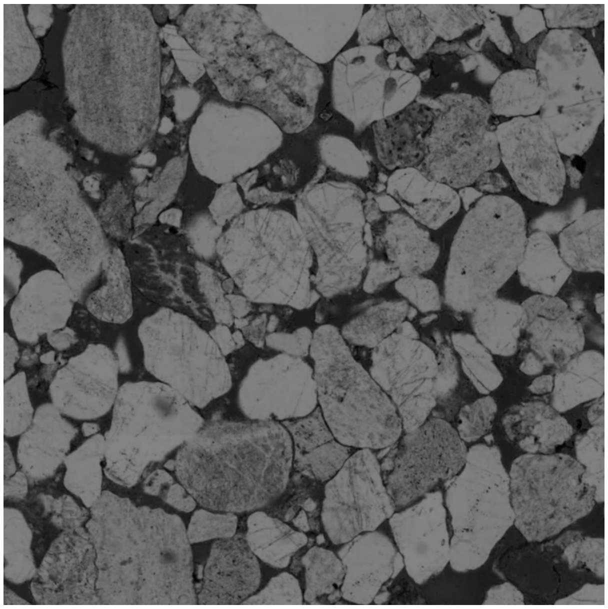

Figure 1: Sample of a rock thin-section image

1.1 Motivation

Many existing geometric models strictly use spheres or ellipsoids to represent grains [5, 8, 13]. However, a paper by Torskaya et al. [12] demonstrates that grain shape does have a significant impact on fluid transport properties such as permeability. Therefore, models utilizing realistic grain shapes have an advantage when it comes to predicting transport properties over models utilizing strictly spherical or ellipsoidal shapes.

Furthermore, the lack of available data for 3D porous structures, particularly those with plausibly realistic grain shapes, poses a challenge for data-driven approaches in geoscience. Neural networks used for grain segmentation and reservoir modeling require substantial amounts of high-quality training data, which are scarce. Deriving 3D structural data from seismic observations is uncommon and depends heavily on prior interpretation efforts [9]. In addition, acquiring such data through direct measurements is often constrained by industry confidentiality and policy restrictions. Fast, stochastic geometric models can help close this data gap by enabling the rapid generation of diverse 3D structures for training and analysis.

Lastly, Fragiacomo [4] has shown that interactive graphical models can greatly assist in geology education on stochastic geo-modelling. While not the main purpose of this project, the techniques developed here could benefit further attempts at creating interactive graphical models for education purposes.

1.2 Research Questions

The objective of this project is to provide answers to two fundamental questions. The primary research question to be addressed is the following:

How can we stochastically generate, distribute and position plausibly-realistic grains in a fast manner?

We attempt to answer this question by creating a geometric simulator of 3-dimensional porous structures: a program that is designed to stochastically generate, distribute and position plausibly-realistic grains in a fast manner. In order for a simulator to achieve this, we break down the research question into multiple components.

First, the subsurface is inherently uncertain due to a lack of data, and the complexity of accurately modelling the natural processes of rock formation require a stochastic approach [10]. In order for the simulation to be stochastic, we introduce randomness, or uncertainty, in the way grains are generated, distributed and positioned.

Stochastic simulation is a computational technique used to model systems with inherent randomness or uncertainty. It involves running simulations with different random inputs to understand the potential range of outcomes and their probabilities. In this case, grain generation could be handled by a probabilistic approach such as the use of a uniform distribution to model grains within a bounding volume.

Secondly, we need to choose a packing algorithm to handle the initial distribution and further potential repositioning of the grains. The goal of the packing algorithm is to create a dense packing, such that porosity is minimized. Ideally, the packing algorithm is computationally efficient and is compatible with the representation chosen for the complex grain shapes. Two approaches to grain packing algorithms that can achieve stochastic generation and distribution of grains are considered and outlined in this work.

Finally, the plausibly-realistic grains themselves need to be represented and generated. For this project, implicit functions are utilized to represent realistic grain geometry. An alternative approach would be to utilize mesh-based grains.

Following that, there is a secondary research question that this thesis attempts to answer:

How can we modify and control generated porous structures from parameters such as target porosity or lithological properties (grain shape, size, distribution, etc.)?

It is evident that a simulator which can be controlled through parameters is more useful than a static one without parameters. The process of grain generation can be directly influenced by lithological parameters. However, the same cannot be said for a target porosity parameter. Porosity is a measure of the empty space of which its final value is only known at the end of grain generation. Therefore it is required to tune and adjust the simulator in such a way that a target porosity parameter matches up closely to the actual final porosity of any given simulated structure.

2 Related Work

2.1 Background

Rocks are broadly classified into three categories based on their formation processes: igneous, sedimentary and metamorphic. Geologists further divide sedimentary rock into three major types: clastic, organic and chemical sedimentary rock. The type we are interested in simulating is clastic sedimentary rock. This rock consists of clasts, fragments, pieces and particles broken off of pre-existing rocks. For the purposes of this project, these fragments will be referred to as 'grains'. The formation of clastic sedimentary rock is a process that occurs through six distinct stages: weathering, erosion, transportation, deposition, compaction and cementation. [6]

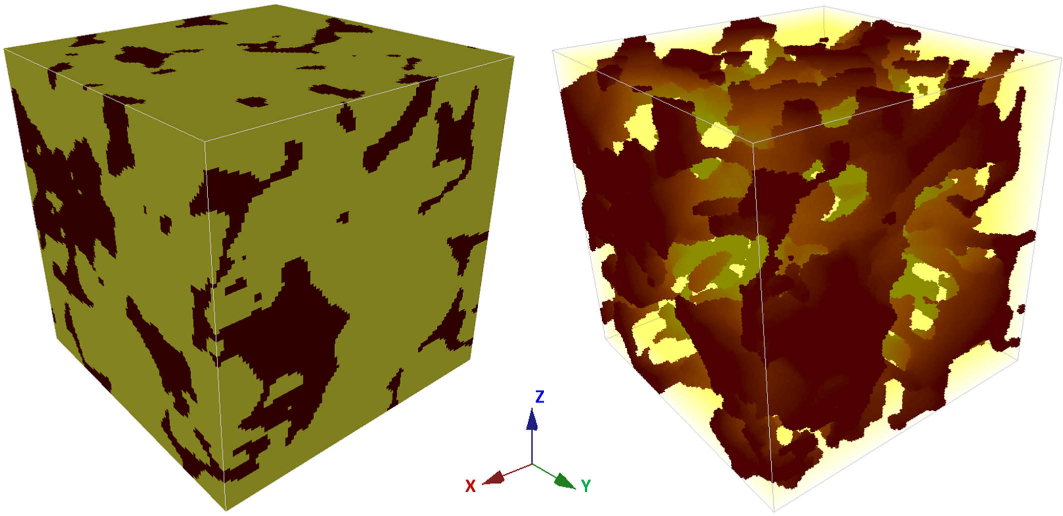

Figure 2: Voxel representation of a 3D porous structure from Chen et al. [1]

The result of these processes is a sponge-like structure with solid grains and inter-grain cavities, known as pore space. An example of these sponge-like structures can be seen in Figure 2. The ratio between the volume of pore space and the volume of grain space is known as porosity [10]. The ability of fluids to flow through the pore space is known as permeability. Porosity and permeability are properties of sedimentary rock that describe its ability to hold onto and move fluids and gasses [6].

Geological reservoirs are bodies of rock that retain useful resources such as oil, natural gasses and water. Sedimentary rocks, particularly clastic types, form the bulk of many reservoirs. Their pore structure governs key reservoir properties such as porosity, permeability, and capillary pressure. By studying and modelling sedimentary rock properties, we can gain a better understanding of geological reservoirs [9]. Understanding and predicting these properties at various scales is vital for efficient resource extraction, carbon storage, and groundwater management. Geological modelling specifically, is utilized as a tool to provide elements for reservoir modelling.

2.2 State-of-the-Art

In their paper, Torskaya et al. [12] determine how closely digital-rock models must represent real grain geometry to accurately predict flow and electrical behavior in sandstones. They extract individual grains from micro-CT scans of sandstone samples and build four grain models such that grain shape is the only controlled variable. The models work by generating shapes using fitted ellipsoids, equal-volume spheres, surface-area-matched spheres or real angular grains approximated with 62 surface points (as seen in Figure 3) and simulate sedimentation by dropping and rearranging them. A permeability analysis is conducted which finds that, at the same porosity, using spheres as grain shapes overestimates the porosity compared to the actual measured transport properties of the sandstone. They also find that even a little cementation can reduce permeability significantly. This paper serves as an example for modelling realistic grain shapes using surface points fitted on images. The method of dropping and rearranging grains to generate a dense packing is similar to a generate-settle algorithm. This approach to packing grains is one of two candidates considered for our packing algorithm.

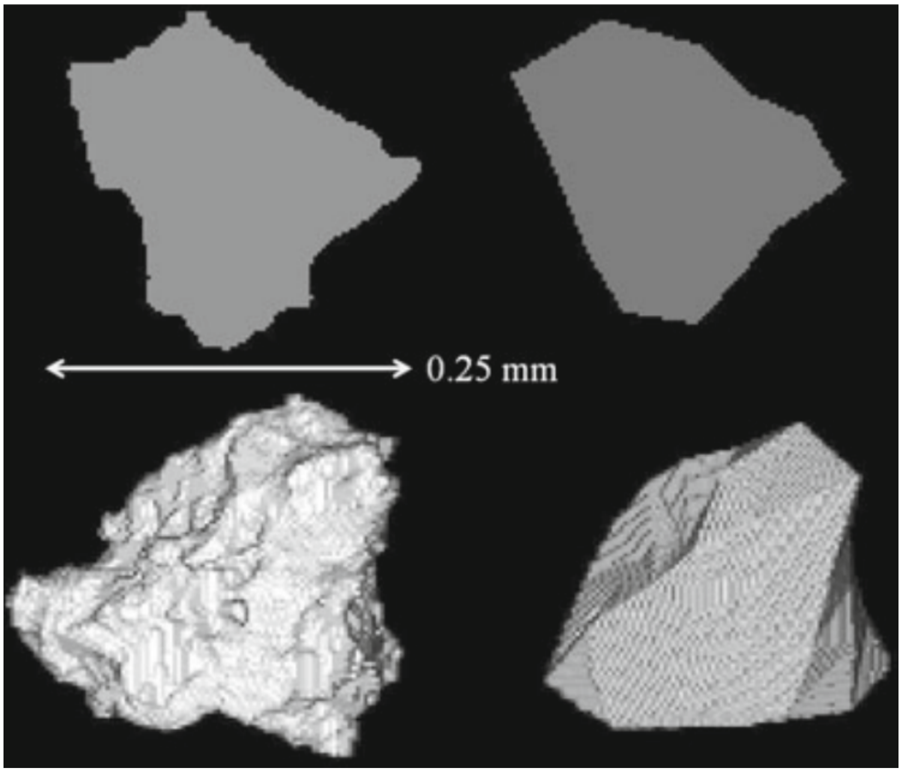

Figure 3: Scanned grains (left) and their discrete representations (right) in 2D and 3D from Torskaya et al. [12]

In another paper, Jin et al. [5] develop a depositional model that reconstructs the geometric, transport and mechanical properties with a focus on unconsolidated sands and sandstone. Their main contribution is through the introduction of a dynamic, process-based workflow to simulate grain sedimentation, compaction and diagenetic cementation with the Distinct-

Element Method (DEM). The sedimentation part of this approach is similar to the generate-settle algorithm, with a major difference being that this model actually employs physics in the form of gravity, contact forces and friction to move and slide the grains past each other. After the initial packing through sedimentation, the grains are compressed by moving the boundaries closer together simulating compaction. Finally, cement is added to the pore space around grains according to grain-size-dependent rules to simulate cementation. The model is evaluated by comparing the porosity trends, porosity as measured at each stage of the simulation, with porosity trends of rocks going through these stages in a laboratory. This paper lends credibility to the advantages of simulating natural processes of deposition, compaction and cementation, but crucially, only uses solid spherical grains, resulting in angular edges and real grain elasticity being under-represented.

In yet another paper, Mousavi and Bryant [8] present a geometric model which addresses the importance of deformation and elasticity of grains for porosity reduction. The model utilizes an augmented version of the cooperative rearrangement algorithm on ductile spheres which posses a solid core and deformable shell allowing for grain overlap. This approach approximates a tight packing without relying on heavy physics and is the second candidate for our packing algorithm. After simulating deposition and compaction using this approach, a layer of cement in the form of quartz growth is added according to grain size. The fraction of ductile grains and the core-to-shell ratio, the relative size of the solid core to the deformable shell, can be adjusted to match compaction experiments and predict transport properties. They also find a tipping point where a high enough fraction of ductile grains, in combination with small cores, causes the porosity to effectively reduce to zero. The limitations of this model are the assumptions of spherical grains, uniform cementation and the omission of deposition/compaction kinetics. The authors write that these could be improved by coupling with DEM, as used by Jin et al. [5], and including lifelike grain geometries.

2.3 Research Gap

These papers show that various assumptions made when designing geometric models of porous structures affect the predicting power of the model. Specifically, the assumption of spherical grains is prevalent when simulating or modelling sedimentary rock.

While these papers also show that emulating real physics in the form of compaction and cementation stages in their models improves their predictive power, for this project we choose to ignore the processes of compaction and cementation. Combining these physics-based stages with realistic grains is definitely something to be explored but out of scope for our purposes.

Additionally, current research models are often inaccessible to educators and researchers . There is a gap in lightweight simulation tools that can both contribute to research and serve as interactive models for teaching sedimentary processes and pore-scale phenomena.

This thesis addresses these gaps by introducing a stochastic simulator that enables the generation of realistically shaped grains using implicit functions, and allows the user to control the simulator through lithological parameters and target porosity. The simulator aims to be both computationally efficient and suitable for porosity and grain geometry analysis, thereby contributing to both research and educational contexts.

3 Methodology

In order to address the research gap, we want to develop a geometric model that generates realistically shaped grains in a dense packing. The development process is complex due to the many influencing factors, which requires prior definitions, decisions and trade-offs. First of all, we need to choose a suitable method for grain representation. Then, we need to select a packing algorithm that works with our choice of grain representation. Last, we determine which parameters the simulation utilizes and how they affect convergence to a dense packing.

3.1 Grain Representation

3.1.1 Image Representation

An image-based representation models grain geometry using a discrete 3D grid of values, known as voxels, or volumetric pixels that define the presence or absence of material at each point in space. In this format, each voxel corresponds to a small cube within the simulation volume and is typically assigned a binary or scalar value indicating whether it belongs to a grain or to the surrounding pore space. An example of this kind of representation can be seen in Figure 2. Image-based models are especially useful for capturing highly irregular or porous structures that may be difficult to describe analytically. This representation also lends itself well for intuitively comparing and understanding the structure of porous media. However, the resolution of the voxel grid limits geometric precision, and high-resolution images can quickly lead to large memory requirements.

3.1.2 Explicit Surfaces

In the context of grain representation, explicit surfaces refer to geometric models where the surface of each grain is directly defined using a mesh of polygons, typically triangles. These triangulated mesh-based grains offer a detailed way to describe complex 3D shapes by explicitly listing all vertices and their connectivity. Figure 3 shows how Torskaya et al. [12] convert an image-based representation of grains, acquired through micro-CT scans, into an explicit surface representation by extrapolating the geometry between 62 surface points. One advantage of explicit surface representation is the high level of visual and geometric accuracy it provides, making it well-suited for rendering and surface analysis tasks. However, a major drawback is the computational overhead involved in processing and collision detection, especially when dealing with many irregular grains, such as the ones used in this project, which can significantly hinder performance in packing algorithms and simulations.

3.1.3 Implicit Surfaces

Implicit surfaces represent grain geometry using mathematical functions rather than explicitly defined meshes. Each grain is described by an implicit function, typically a signed distance function, that evaluates a point in space relative to the grain's surface. A point lies inside the grain if the function returns a negative value, outside if positive, and exactly on the surface if the value is zero. A key advantage of implicit surfaces is their flexibility: they allow for smooth, complex geometries and enable efficient overlap detection through simple function evaluations. However, a notable disadvantage is the complexity involved in generating the implicit grain shapes themselves.

In this project, grains are represented using implicit functions. In order to generate shapes defined by a continuous signed distance function, we derive them from real 2D images of sedimentary rock. From these images, we extract the contours as a list of points. These contours are then extruded and tapered to form complex 3D shapes. While implicit functions are used to generate and detect overlap between grains during simulation, an image-based approach is invoked post-simulation to visualize the resulting structure. In addition, in order to compute the porosity, the entire structure is sampled to evaluate the total porous volume. This approach is similar to an image-based approach (voxels) but without rendering an image.

3.2 Packing Algorithm

Two candidate algorithms were considered for the purposes of this project: the generate-settle algorithm and the cooperative rearrangement algorithm. Each has distinct advantages and trade-offs in the context of stochastic grain simulation.

3.2.1 Generate-settle Algorithm

The generate-settle algorithm draws inspiration from natural sedimentary processes, especially deposition. In this approach, grains are introduced into a simulated volume one at a time, typically from the top. Each grain is allowed to fall under gravity and interact with previously placed grains through a simple, or physical contact model. The contact model used by Torskaya et al. [12] is a simple one. Here, when a grain encounters another grain, a random rotation and a random horizontal displacement are repeatedly applied until the grain is able to fall further. The contact model may also be more physics-based such as the one used by Jin et al. [5]. Here, contact forces and friction are simulated to allow grains to slide into a mechanically stable position. This ”settling” process mimics the natural behavior of clastic sedimentation in environments like riverbeds or sedimentary basins.

The main advantage of this approach is that it is intuitive as it closely matches real sedimentary rock formation. This enhances its interpretability and relevance for modeling geological structures. This physical approach also supports the implementation of further physics-based processes such as compaction and cementation as shown by Jin et al. [5]. Finally, this approach is suitable for complex grain shapes as rotation is handled at the same time as displacement.

A notable disadvantage of this approach is its slow convergence, particularly as the volume becomes increasingly occupied. In the early stages, grains can settle quickly due to the small amount of interactions it can experience. But as more and more grains are added, it becomes computationally expensive to find valid positions for new grains as they must interact with many grains to properly settle. Especially when real physics are simulated. If each grain is actively pushing on each neighboring grains, a grain being dropped on top causes a ”ripple” effect of interactions before affecting the bottom grains.

3.2.2 Cooperative Rearrangement Algorithm

The cooperative rearrangement algorithm, originally known as the ”collective rearrangement algorithm” as presented by Clarke and Wiley [2], offers a non-physical, geometric method for achieving dense packing. It begins by randomly placing grains according to a uniform distribution within a bounding volume and then iteratively moves overlapping grains apart while gradually increasing the size of every grain. Grain sizes may be reduced during the process to prevent deadlocks and to optimize packing density. The algorithm emphasizes efficiency and flexibility over physical realism.

Given the need for fast, controllable, and shape-flexible simulations, the cooperative rearrangement algorithm has been selected for this project. It is compatible with implicit grain-shape representation and its ability to stochastically generate dense packings with less computational overhead makes it the most practical choice. While it sacrifices some physical realism compared to the generate-settle algorithm, it is better aligned with the goal of generating large numbers of plausible porous structures.

3.3 Parameters

The simulation is controlled by several user-defined parameters, including the number of grains, grain size, grain size distribution, and target porosity. These parameters influence how grains are initially placed and how they evolve throughout the simulation. For the lithological properties, they directly affect the grains themselves and can be applied upon their initial generation and placement. A key objective of the simulation is to achieve dense packing of grains, where individual particles are positioned as closely as possible without overlapping. As the packing density increases, the total solid volume occupied by the grains grows, resulting in a corresponding reduction in porosity, the fraction of empty space within the bounding volume [2]. A target porosity parameter cannot directly affect the algorithm because the algorithm iteratively adjusts grain positions and sizes to minimize porosity. So instead, a final stage is added to the algorithm. When a dense packing (configuration with minimal porosity) has been achieved, the true porosity is compared against the target porosity. If the target porosity is greater than the true porosity, the grain size is gradually decreased until the true porosity is slightly above the target porosity. This ensures that the final configuration closely matches the desired porosity while preserving the overall structure of the grains.

3.4 Two Models

In this project, two grain packing models are developed: a spherical model and a realistic grain model. The spherical model represents each grain as a perfect sphere and serves as a simplified baseline for simulating packing behavior using the cooperative rearrangement algorithm. Building on this foundation, the realistic grain model introduces more complex grain geometries defined by implicit functions derived from real 2D contours. In order to generate the 3D grain shapes based on 2D contours, an additional parameter called grain length is introduced. Grain size determines the size of the initial contour while grain length determines the magnitude of the extrusion.

3.5 Technology Stack

For the modelling and visualization requirements of this project, the Visualization Toolkit [11] is utilized. This software library, also abbreviated as VTK, is an open-source toolkit that is widely used in scientific research, engineering and data analysis. It provides tools for generating, manipulating and finally rendering grains using built-in abstract interfaces for implicit functions. To handle user interaction, the simulator integrates Qt, a cross-platform GUI framework. It enables dynamic parameter adjustment through a simple graphical interface, allowing users to configure settings such as grain amount, size, and porosity targets before launching a simulation.

The simulator is developed using C++, chosen for its performance advantages and compatibility with VTK. The build system is managed using CMake, and the project is built and tested using Visual Studio on Windows 10. For version control, Git was used throughout the development process to manage code, the branching of the spherical model into the realistic grain model and to track changes.

3.6 Evaluation

In order to answer our research question, a combination of quantitative and qualitative evaluation is necessary. We need to determine whether the cooperative rearrangement algorithm is suitable for stochastically generating, distributing, and positioning grains in a fast manner. We also need to examine the grain geometry of the realistic grain model to determine whether the generated grains are plausibly realistic.

First, the utilization of a literature comparison will serve to validate the simulator. A quantitative comparison is made between the resulting porosity values of the spherical model and those reported in the paper by Jin et al. [5]. By running our simulator with the same parameters as specified in their paper, we can compare the efficacy of the cooperative rearrangement algorithm to that of physical models.

Additionally, we evaluate the realistic grain model qualitatively by visual inspection of the resulting grain geometries. We examine whether the grains exhibit natural variations in size and shape. This is crucial for addressing whether or not the grains are plausibly realistic. We also address artifacts found in the grain shapes as a result of the contour extrusion process.

4 Spherical Model

The spherical model implements the cooperative rearrangement packing algorithm utilizing the vtkSphere class, an implicit function for spheres, while preparing to replace them by more complex grain shapes requiring the more abstract vtkImplicitFunction class.
The collective rearrangement algorithm as originally laid out by Clarke and Wiley [2] generates a dense packing of spheres of two distinct sizes. It does not implement physical processes like gravity or friction. Instead, it allows spheres to overlap and moves them away from each other iteratively while decreasing the tolerance (the amount of overlap allowed) over time.

Figure 4 shows a version of the cooperative rearrangement as used in both the spherical model and the realistic grain model. It is more similar to the approach taken by Mousavi and Bryant [8] as there is no vibration (random displacement) of the grains.

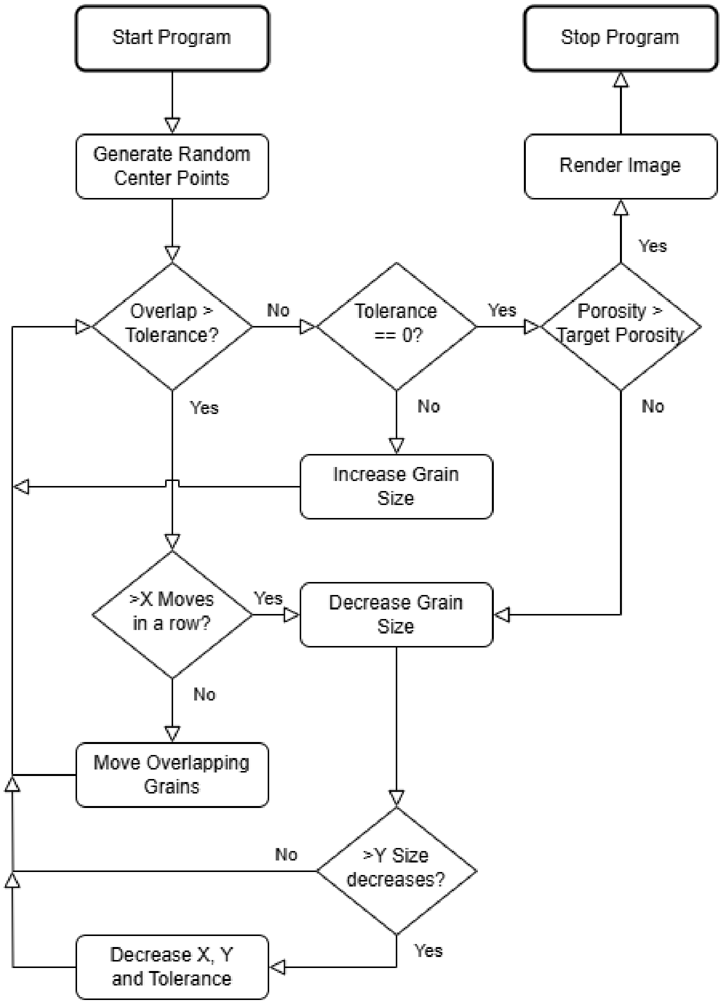

Figure 4: Program Flow Chart

The following parameters are taken from the user before running the program:

- **Grain Amount**: Total number of grains to simulate.
- **Grain Size**: Initial size of the largest grains.
- **Grain Distribution**: Size ratio between large and small grains.
- **Target Porosity**: Desired porosity; controls when the simulation should terminate.

The following parameters are program specific and affect convergence speed and accuracy:

- **Overlap Tolerance**: Threshold for allowable grain overlap before grain repositioning.
- **Maximum Sequential Moves**: Repositioning attempts that may occur before grain size reduction.
- **Maximum Decrease Iterations**: Number of grain size reductions before tightening overlap tolerance.

First, a random center point for each grain is generated to sit within a certain bounding box. These center points are generated by using a uniform distribution, meaning that each point within the bounded volume is equally as likely to be a center point. Then, each pair of neighboring grains is checked to see if their volumes overlap (collision). For any pairs of grains that do overlap, their maximum penetration depth is compared to the overlap tolerance. If out of all neighboring pairs of grains, there exists no overlap that exceeds the overlap tolerance, there is likely to be unoccupied space that the grains could fill up. We allow the grains to do this by proportionally increasing the size of every grain. This way, the grain distribution is preserved.

In addition to maximum penetration depth, a vector pointing towards the center point of the overlapping grain, with magnitude equal to half of the penetration depth, is stored for each collision a grain experiences. If at least one pair of grains overlaps more than the tolerated amount, all overlapping grains, even those below the tolerated threshold, are moved away from one another. For each grain that attempts to move, its overlap vectors are added together, and finally inverted to find the motion vector. The magnitude of the motion is found by multiplying the resulting motion vector by the number of overlaps the grain is experiencing. As a result, an overlapping pair of isolated grains will move apart just enough to separate, while a grain trapped within a group of inter-overlapping grains will shift a greater distance in the direction with the fewest surrounding grains.

After a sufficient number of sequential moves, it is assumed that the grains cannot fit within the allotted space. If a pair of grains overlap beyond the allowed tolerance, but the number of sequential moves has already reached the maximum permitted, the size of each grain is reduced instead. This way, we can attempt to find a dense packing without having any pair of grains overlap more than tolerated.

This process is then repeated: all neighboring grains are checked, and for each detected collision, the maximum penetration depth is compared to the overlap tolerance, and the grains are either moved, grown, or shrunk accordingly (see Figure 4). The goal of the algorithm is to find a configuration such that the grains can be as large as possible without overlapping more than tolerated.

Eventually, the grains will get stuck between growing and shrinking. This means the optimal packing density can be found between the larger size and smaller size that the grains keep switching between. At this stage, we want to decrease the overlap tolerance and start making smaller moves and smaller changes to grain size. This allows the grains to converge to an optimal packing density. To achieve this, the total number of size decreases since the last decrease iteration is counted and when it exceeds the maximum decrease iterations allowed, the counter is reset and overlap tolerance, as well as the magnitude of grain size changes, are decreased.

Finally, the overlap tolerance approaches 0. At this point the changes to grain size and position are small and a dense packing has been achieved. In other words, the algorithm has converged on the minimum porosity. Therefore, if the target porosity is smaller than the achieved porosity, we end the procedure. Otherwise, we gradually decrease the size of the grains until the true porosity is just above the target porosity, and then stop.

5 Realistic Grain Model

Now that we have a working packing algorithm for spherical shapes, we adapt this model into the realistic grain model by augmenting the grains to possess realistic grain shapes.

5.1 Shape Generation

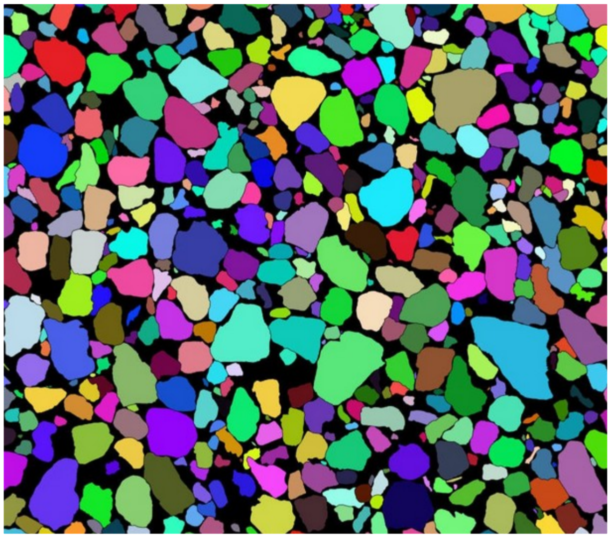

Figure 5: Grain segments

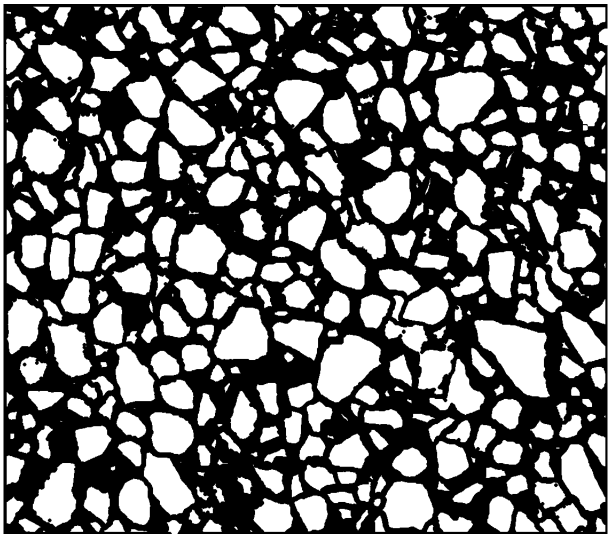

Figure 6: Grain segments separated by a mask

Figure 5 shows a segmented version of a cross-section image of real sedimentary rock. Each colored region represents an individual grain, and the black background represents the pores between the grains. This image, as well as the same grain segments further delineated by a mask as shown in Figure 6 were provided by the supervisors of this project.

The 433 individual grain contours in the mask image were extracted and put into CSV files. In each CSV file, the first column represents the x-coordinate and the second column represents the y-coordinate of the pixels that encompass that specific contour. These contours can be used to extract realistic 2-dimensional shapes but a little more effort is required to generate 3-dimensional shapes.

Instead of utilizing the vtkSphere class, we define and implement the custom ImplicitGrain class which inherits from vtkImplicitFunction. This class has an implicit function evaluation that takes the form of a signed distance function $D(p, S)$ where p is a point in 3D space and S represents the grain geometry such that:

$$\begin{cases} D(p,S) & < 0 \qquad \forall\, p \notin S \\ D(p,S) & > 0 \qquad \forall\, p \in S \\ D(p,S) & = 0 \qquad \forall\, p \cap S \end{cases}$$

Each implicit grain is defined by extruding and tapering its assigned contour represented by a concrete set of points with vtkPolyData. Then, using the vtkImplicitPolyDataDistance class, the signed distance to the grain for any point can be computed. This is crucial for allowing the algorithm to check for grain overlap (collision).

For the realistic grain model the following parameters are taken from the user before running the program:

- **Grain Shape (Contour)**: For each grain: a list of points from a 2D contour which is extruded and tapered to get a 3D shape
- **Grain Amount**: Total number of grains to simulate.
- **Grain Size**: Initial size of the largest grains.
- **Grain Length**: The ratio between height and width as determined by the grain size and contour and its extruded length.
- **Grain Distribution**: Size ratio between large and small grains.
- **Target Porosity**: Desired porosity; controls when the simulation should terminate.

The shape of the grain is determined by its grain size, 2D contour, and by an additional parameter, the grain length. The grain length specifies by how much the 2D contour is extruded. The grain size and contour determine its width in x and height in y, while the grain length determines its length in z.

5.2 Grain Overlap Detection

Efficient and accurate collision detection is critical for the performance of the cooperative rearrangement algorithm, especially when working with complex, non-spherical grain geometries. In this model, grains are defined by implicit functions based on extruded real 2D contours, which means that overlap detection is more computationally expensive than for spherical grains.

To reduce the computational load, a multi-stage collision-checking process is implemented. First, the simulation space within the bounding box is partitioned so that each grain only checks for collisions against nearby neighbors, rather than all other grains. This reduces the number of pairs of grains that need to be compared significantly, especially as the total grain count increases. This is because the program speed scales linearly with the number of grains, as each grain only needs to be compared with a static number of neighbors. [2]

Second, each grain is fitted with a bounding sphere that tightly encloses its full geometry. This allows for a fast preliminary check: if the bounding spheres of two grains do not intersect, then a collision between the actual grain shapes is impossible and further evaluation is skipped. This bounding-sphere check further culls unnecessary comparisons.

Only when two bounding spheres overlap does the algorithm proceed to check for a true collision. This final step involves evaluating the signed distance function along the surface of each grain using the vtkImplicitPolyDataDistance class. If a point on one grain lies within the implicit surface of another, a collision is detected. This way, the maximum penetration depth between the grains can be determined.

6 Results

6.1 Porosity Comparison

First of all, we validate the use of the cooperate rearrangement algorithm by comparing the porosity yielded by our spherical model with the porosity yielded by the depositional model developed by Jin et al. [5]. We do this by running the program 10 times using the same parameters that the depositional model used, which are as follows:

- **Bounding Box Dimensions**: 4 cm × 4 cm × 6.58 cm
- **Grain Amount**: 1100
- **Grain Size**: 2.7 mm
- **Grain Distribution**: 0.666...
- **Target Porosity**: 0

The following values are taken for the program parameters:

- **Overlap Tolerance**: 0.20
- **Maximum Sequential**: Moves 10
- **Maximum Decreases**: Iteration 5

A larger overlap tolerance causes the program to accelerate faster toward a dense packing but the effect saturates around a tolerance of 0.20 according to Clarke and Wiley [2]. For the maximum program parameters, increasing them further than these values does not seem to decrease the final porosity for this configuration. The program is run 10 times and porosity is measured in two ways: by taking the fraction of empty space inside of the entire bounding box, and by taking a 1 cm × 1 cm × 1 cm sub-sample in the center. One representative expression of the simulation is shown in Figure 7.

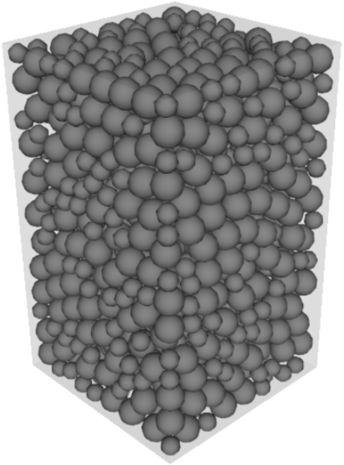

Figure 7: Spherical model with 1100 grains in 4 × 4 × 6.58 bounding box

Table 1: True and sampled porosity values from the spherical model using 1100 spheres

	1	2	3	4	5	6	7	8	9	10
True Porosity	48.1%	47.1%	47.1%	47.3%	48.2%	45.9%	47.0%	47.6%	46.3%	47.3%
Sampled Porosity	42.5%	41.9%	42.0%	41.9%	42.9%	40.9%	41.7%	42.3%	41.3%	41.9%

Table 2: Comparison of spherical model porosity with Jin et al.'s depositional model

	Spherical Model	Jin et al. (deposition)	Jin et al. (compaction)
True Porosity	47.2%	47.5%	40.2%
Sampled Porosity	41.9%	42.4%	36.9%

6.2 Realistic Grain Geometry

Due to computational constraints, the realistic grain model is unable to simulate large-scale structures exceeding approximately 50 grains. As such, porosity analysis for these complex grains is not feasible at the scale required for meaningful porosity analysis. Instead, this section focuses on the geometric characteristics of the individual grain shapes generated using the implicit function approach.

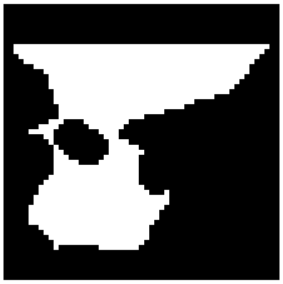

Figure 8: Incomplete contour containing a hole

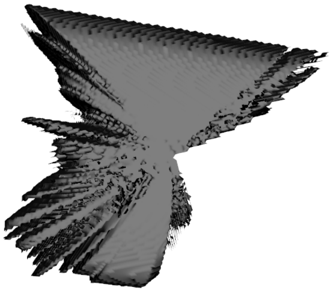

Figure 9: Grain shape extruded from Figure 8

During the generation and rendering of realistic grain shapes, some grains displayed clear visual anomalies or fragmentation, as seen in Figure 9. It is evident that the segment seen in Figure 8 has the same shape as the shape in Figure 9. These artifacts are not due to errors in the rendering pipeline or the implicit function framework itself, but rather originate from the underlying 2D contours used to construct the grains. Through visual inspection of the grain segments in Figure 6. eight of these incomplete contours were identified and removed to prevent the generation of these artifact-prone grain shapes.

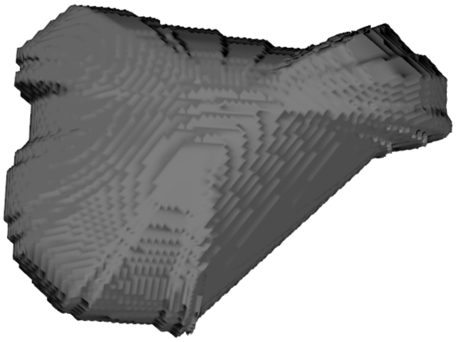

Figure 10: Typical Grain with grain length = 0.7

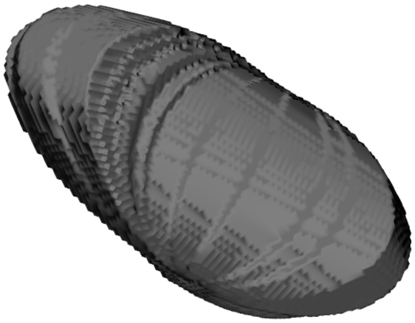

Figure 11: Different angle of the grain in Figure 10

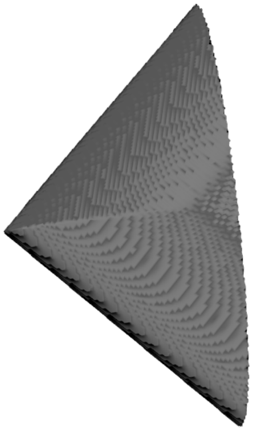

Figure 12: Grain utilizing a tiny contour

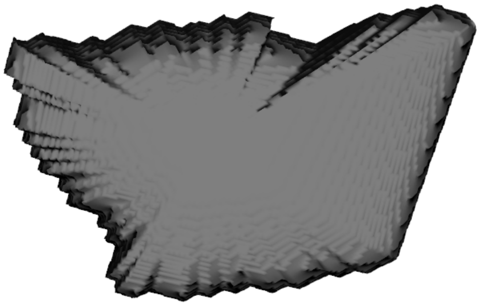

Figure 13: Concave grain with grain length = 0.7

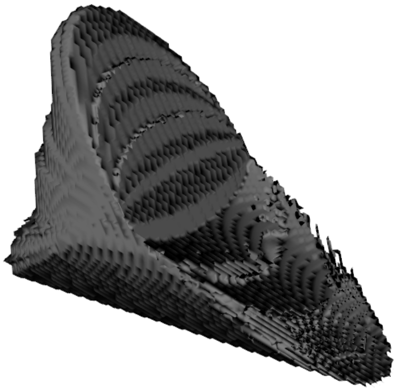

Figure 14: Rendering artifact despite a complete contour

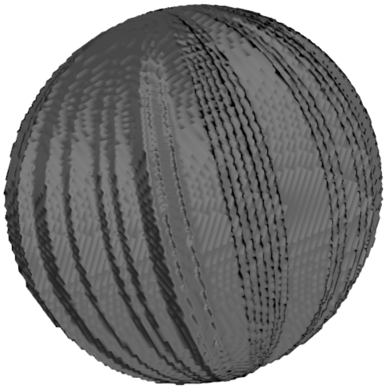

Figure 15: Typical grain with grain length = 2.0

A variety of grain shapes were generated through extrusion and tapering of real 2D grain contours into 3D space. Figures 10-15 show a selection of these grains. The resulting geometries exhibit a wide range of features, including asymmetry, angularity, elongation, and curvature. Some grains resemble naturally weathered or compacted clasts, while others are more geometric or stylized due to the limitations of the contour extraction process.

7 Discussion

We return to the main research question of this project:

How can we stochastically generate, distribute and position plausibly-realistic grains in a fast manner?

While we succeeded in stochastically generating, distributing and positioning complex grains, it is not clear whether those grains can be defined as ”plausibly-realistic”. Furthermore, the presented realistic grain model definitely did not manage to perform in a fast manner. Nonetheless, we validate the use of the cooperative rearrangement algorithm as a packing algorithm.

7.1 Cooperative Rearrangement

By comparing the porosity results from our spherical model with the depositional model by Jin et al. [5], we can see that the porosity of our spherical model is about equivalent to the depositional model after the deposition stage but greater than the depositional model after the compaction stage. Although it should be noted that the compaction stage of the depositional model involves the reduction of the bounding box size to squeeze grains together, this suggests that the cooperative rearrangement model is unable to drive porosity below the deposition threshold.

The slight run-to-run variability in our results (spread of $\geq 2\%$ in true porosity) likely arises from stochastic initial placements and the unpredictable way the grains happen to rearrange themselves in each instance. In practice, increasing the program parameters to allow the program to perform more iterations using smaller moves should allow the simulator to converge closer to a minimal porosity, at the cost of longer runtimes.

7.2 Limitations

The main limitation encountered with our realistic grain model is its computational scalability. While the spherical model can simulate over a thousand grains efficiently, the realistic model becomes unmanageable beyond approximately 50 grains. To understand this limitation, the realistic grain model's performance was analyzed.

By using a program timer, it became apparent that there were two main parts slowing the simulation down. First was the initial grain shape generation and secondly there was the collision detection. When increasing the number of grains, the collision detection used a larger portion of computing time. While the collision detection was designed to scale linearly with increasing grain count, it seems the program cannot handle the number of grains required to start this linear scaling. At low grain counts, increasing the number of grains does increase the number of collision checks the average grain goes through on each iteration.

This performance bottleneck restricts the model from being used for full-scale porosity or permeability analysis, which typically requires hundreds or thousands of grains to produce statistically meaningful results. As a result, the realistic grain model is currently better suited for geometric analysis rather than structural simulations.

While the shapes of the grains reflect the realistic 2D contours when viewed from a certain angle, when viewed from another angle, they appear to have a symmetric, oval-like shape. When the grain length is high such as in Figure 15, the shape becomes effectively ellipsoidal in at least one axis. Furthermore, despite the removal of incomplete contours, there are still some grain shapes that appear to have rendering artifacts such as the shape seen in Figure 14. The cause of these artifacts is not understood.

7.3 Future Work

The cooperative rearrangement algorithm is not well suited for simulating physical processes such as compaction or cementation. However, there are still things that could improve this approach to geometric modelling. First, there is the performance. An alternative method for collision detection could potentially increase performance and allow for the model to generate structures with a higher number of grains. For example, the algorithm presented by Meijster et al. [7] for computing distance transforms in linear time could do exactly that.

Another method to improve the collision detection is by using the Signed Heat Method to approximate the signed distance functions as done by Feng and Crane [3]. The Signed Heat Method is a general-purpose framework for computing signed distances on unreliable or incomplete data. This would allow the collision to work with incomplete shapes containing holes, self-intersections or other artifacts. Additionally, it could also improve performance as detecting a single collision involves solving two linear systems, making it fast and scalable.

In terms of grain shapes, an alternative method for generating 3D grains could involve combining two contours with similar areas. This could be done by positioning one contour perpendicular to the other and extrapolating the shape in 3D space between them. This would result in a grain shape that retains the contour features in more than one axis of rotation.

Another approach that could be utilized is the medial axis transform (MAT). This technique represents a shape not by its boundary, but by the set of points equidistant to two or more points on the shape's boundary. This ”skeleton” captures the core structure of the shape and could be used to generate, or even deform existing grain shapes. Instead of generating implicit functions from the real 2D contours, the medial axis could be derived and then extruded into 3 dimensions. This would lead to smoother and more consistent overall shapes.

7.4 Lessons Learnt

Over the course of this project, one continuous theme was the issue of overscoping. I would often pursue an idea that would lead me to include more project aspects than necessary.

Adjusting my ambitions toward a well-defined and achievable scope was the greatest lesson that this project has granted me. In terms of software engineering, I struggled with the implementation of multi-axis contour extrapolation. Requesting help sooner or at all would have saved me a lot of time and effort. Lastly, my research process should take more from the iterative scientific approach. By making smaller prototypes and iteratively adjusting my goals, this project could have yielded more information regarding the potential of the cooperative rearrangement approach or different approaches to collision detection.

8 Conclusion

The primary goal of this project is the development of a simulation to generate, distribute, and position plausibly-realistic grains efficiently, with control over simulation parameters such as grain shape, grain size, and target porosity. The simulator combines the cooperative rearrangement packing algorithm with realistic grain shapes. For spherical grains, the algorithm proves capable of generating dense packings with porosity results closely matching those from an established depositional model by Jin et al. [5]. However, it does fall short of further reductions achieved through compaction. The stochastic nature of the algorithm introduces slight variability in results, consistent with expectations for randomized initial conditions.

For the realistic grain model, the cooperative rearrangement algorithm was adjusted to facilitate collision detection between general implicit functions. That way, grain shapes generated based on real 2D contours could be included. Performance constraints limited simulations to a low grain count preventing a meaningful porosity analysis. Instead the focus shifted to analyzing the geometry itself. Limitations are mainly tied to collision detection performance and rendering artifacts in some generated shapes. Future improvements could include more efficient collision methods and enhanced 3D shape generation by using multiple 2D contours.

References

[1] Qiyu Chen, Gregoire Mariethoz, Gang Liu, Alessandro Comunian, and Xiaogang Ma. Locality-based 3-d multiple-point statistics reconstruction using 2-d geological cross sections. *Hydrology and Earth System Sciences*, 22(12):6547–6566, 2018.

[2] A. Clarke and J. Wiley. Numerical simulation of the dense random packing of a binary mixture of hard spheres: Amorphous metals. *Physical review b*, 35-14:7350–7356, 1987.

[3] Nicole Feng and Keenan Crane. A heat method for generalized signed distance. *ACM Transactions on Graphics*, 43(4):Article 92, 2024.

[4] Fabio Fragiacomo. Leveraging visual gamification to enhance undergraduate education of geological modelling, 2024.

[5] G. Jin, T. Patzek, and D. Silin. Physics-based reconstruction of sedimentary rocks. In *SPE Western Regional/AAPG Pacific Section Joint Meeting*, 2003.

[6] S. Marshak. *Earth: Portrait of a Planet.* W. W. Norton & Company, 2001.

[7] A. Meijster, J. B. T. M. Roerdink, and W. H. Hesselink. A general algorithm for computing distance transforms in linear time. In J. Goutsias, L. Vincent, and D. S. Bloomberg, editors, *Mathematical Morphology and Its Applications to Image and Signal Processing*, volume 18 of *Computational Imaging and Vision*, pages 331–340. University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science, 2000.

[8] Maryam A. Mousavi and Steven L. Bryant. Geometric models of porosity reduction by ductile grain compaction and cementation. *AAPG bulletin*, 97(12):2129–2148, 2013.

[9] Philip Ringrose and Mark Bentley. *Reservoir Model Design: A Practitioner's Guide.* Springer, 2015.

[10] R. Selley. *Applied Sedimentology.* Academic Press, 2000.

[11] VTK Open Source. Vtk online documentation. https://vtk.org/doc/nightly/html/index.html. Accessed: 2025-06-30.

[12] T. Torskaya, V. Shabro, C. Torres-Verdín, R. Salazar-Tio, and A. Revil. Grain shape effects on permeability, formation factor, and capillary pressure from pore-scale modeling. *Transport in porous media*, 102:71–90, 2013.

[13] William M. Visscher and M. Bolsterli. Random packing of equal and unequal spheres in two and three dimensions. *Nature*, 239(5374):504–507, 1972.