

Generating realistic nanopores using procedural methods*

Anders Hafreager ¹, Nicolaas Groeneboom ¹, and Anders Malthe-Sørensen ¹

¹*Department of Physics - University of Oslo
Sem Sælands vei 24, NO-0316, Oslo, Norway*

(Dated: January 24, 2017)

PACS numbers:

I. INTRODUCTION

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis consequat turpis vel mollis accumsan. Etiam eleifend magna vel quam porta, sed porta nibh placerat. Sed in faucibus risus. Duis suscipit tincidunt leo, nec laoreet leo elementum elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed eros risus, convallis sit amet turpis a, dapibus accumsan quam. Donec fringilla rhoncus turpis, quis ultrices odio vulputate elementum. Cras nec viverra urna. Phasellus laoreet tortor sed est sagittis laoreet. Nullam a volutpat ex.

Mauris vestibulum sit amet libero eget facilisis. Vivamus a nunc sed felis ultrices convallis luctus et mauris. Sed tincidunt mauris nunc, vehicula placerat eros faucibus quis. Morbi egestas, nisi quis luctus dignissim, mauris odio pulvinar tellus, non efficitur risus erat vel nunc. Cras vel felis magna. Mauris rutrum magna a interdum sodales. Nam tempor, diam non sodales tristique, nulla ipsum eleifend dui, non aliquam felis tellus id enim. Donec eget nulla leo.

Ut hendrerit cursus libero, in scelerisque tortor sollicitudin accumsan. Nam eleifend velit metus, quis volutpat justo faucibus ut. Cras ut sem in nunc fringilla egestas vitae sit amet justo. Ut vel condimentum tortor. Morbi in massa in ipsum ultrices tincidunt. Duis feugiat dignissim nisi ac ultricies. Quisque eget risus fermentum, condimentum massa ullamcorper, posuere neque. Nam congue consequat mi, vel commodo mi tempus eget.

Nullam ultrices, velit sed venenatis ultrices, risus urna ultricies magna, at pretium lorem nibh et nunc. Suspendisse sed mattis leo. Nulla facilisi. Praesent finibus lobortis leo, ut porttitor massa tincidunt ut. Vivamus quis orci eu sem sodales ornare. Cras sed faucibus velit. Phasellus scelerisque mauris quis lorem dignissim vehicula. Sed tempor ut urna sed bibendum. Donec et nibh pulvinar, commodo magna ac, interdum nibh. Sed laoreet ipsum quis quam lacinia, in dictum erat laoreet. Etiam rhoncus mauris arcu, ac egestas lacus facilisis nec. Pellentesque a orci vitae felis faucibus ultrices sed non massa. In posuere vel ex in elementum. Aenean ac neque a enim gravida tristique eu ac lacus. Ut tincidunt venenatis lectus, et maximus mauris.

Proin ornare nulla id tristique placerat. Integer nec elementum tellus. Cras at tempus lectus. Cras lorem est, ornare id condimentum vitae, scelerisque eu lectus. Aenean libero arcu, consequat vel sagittis eget, sagittis ut neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nec quam sodales, mollis est sit amet, vulputate turpis. Praesent tempus diam eu sem laoreet finibus. Nulla finibus, sapien vel lobortis ultricies, mauris ipsum aliquet urna, quis rutrum sapien dui id ipsum. Quisque sodales ex sapien, ac egestas lacus suscipit et. Suspendisse accumsan rhoncus tristique. Pellentesque ut eros lectus. Praesent viverra sit amet odio nec vulputate. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nulla molestie metus vitae libero vulputate, at interdum enim tempus.

II. METHOD

A. Procedural noise

Procedural algorithmic methods for creating realistic-looking fractal-like patterns have been around for more than 30 years. The most famous of these algorithms is Perlin noise, first developed by Ken Perlin in 1982 for use in the movie “Tron”. Perlin noise is a type of pseudo-random gradient noise that mimics a Gaussian field, but at low computational cost. An improved version of Perlin noise is called Simplex noise, which has several advantages over standard Perlin noise in terms of scalability and speed.

1. Perlin noise - an overview

An overview of the current state of procedural noise functions can be found in [1]. Perlin noise [2] is a band limited repeatable pseudo-random function from $\mathbb{R}^n \rightarrow \mathbb{R}$ that is computationally fast and has properties which make it suitable for creating patterns that mimic nature. Perlin noise is an approximation to Gaussian filtered noise implemented as a pseudo-random spline. The algorithm uses a predefined grid of gradients pointing in a random direction, where for any point within a grid, we interpolate the gradients from the four corners. The algorithm is as follows:

*Electronic address: anders.hafreager@fys.uio.no

1. For a point, obtain the four closest gradients in the grid.
2. For each of the gradients, calculate the dot product between the gradient and a vector defined by the distance between the point and the current grid corner.
3. Instead of linear interpolation $F(t) = t$, use a fade function to produce smoother interpolated values. Perlin noise typically uses the following fade function: $F(t) = t^3(t(6 - 15) + 10)$.
4. Obtain the final value by linearly interpolating between the x-axis and use this result to linearly interpolate the y-axis.

Simplex noise works in a similar manner, but requires fewer computational steps, scales better in higher (≥ 3) dimensions and has fewer anisotropic properties.

2. Properties of procedural noise

Standard Perlin noise is known for having several anisotropic features [1]. Perlin noise also gives rise to a “textile” pattern in the x and y -directions, which is evident from the upper left part of figure 1, depicting 2D Perlin noise in a single octave. The upper-right part shows the same node for Simplex noise, with less obvious patterns. However, when combining several octaves, as depicted in the two lower parts, these anisotropic patterns tend to cancel out. In any case, for observations of weak gravitational lensing, these anisotropies would occur on much smaller scales than the typical resolution. In addition, we have calculated the statistical distributions for Perlin and simplex noise with $\sigma = 0.2$, and have plotted the results in figure 2. Each procedural noise method has distributions that are close to normal, but with some intrinsically shifted sigma and mean.

B. Multiple octaves: creating patterns

The different octaves from procedural noise functions can be combined to produce a wide range of nature-like patterns. A generic N -dimensional procedural function $\Phi(\vec{x}) : \mathbb{R}^N \rightarrow \mathbb{R}$ could be expressed as

$$\Phi(\vec{x}) = \Theta\left(\sum_k \kappa(k) P(f_s(\vec{x}, k))\right), \quad (1)$$

where $f_s(\vec{x}, k)$ describes the frequency scale modifier (e.g. $f_s(\vec{x}, k) = k\vec{x}$), $\kappa(k)$ is the octave amplitude (eg $\kappa(k) = 1/k$) and $\Theta(x)$ an overall modifier (e.g. $\Theta(x) = x$ or $\Theta(x) = \frac{1}{x}$). In a sense, $\kappa(k)$ is similar to Fourier coefficients. In its simplest case, summing directly over the octaves with amplitude $1/k$ results in

$$\Phi(\vec{x}) = \sum_k \frac{1}{k} P(k\vec{x}), \quad (2)$$

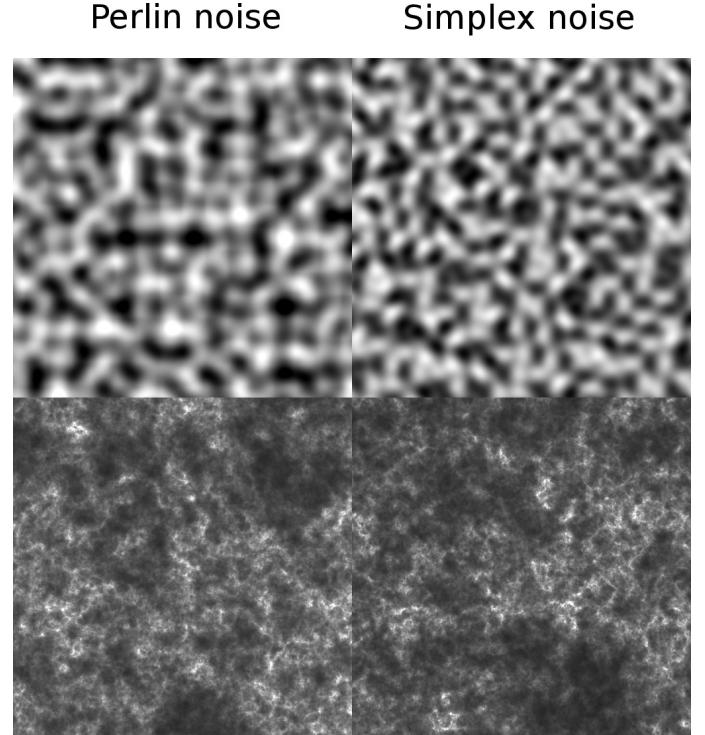


FIG. 1: Perlin noise (left) vs simplex noise (right). Note the straight x-y pattern in the single upper Perlin mode and the less anisotropic simplex mode. Lower part: combining octaves tend to remove these patterns.

and produces an 2D procedural noise pattern depicted in the lower part of figure 1.

C. Model

In this paper, we are interested in developing a model using Simplex noise to reproduce nanoporous structures that are as statistically close to actual simulations of porous media. For a given data set and a model, we wish to estimate the parameters that best represent the data. This is performed through a full likelihood analysis, where we in addition validate the model by comparing the radial pore size distribution $g(r)$, porosity and surface area of the system.

We construct a generic sum over procedural noise octaves with the expression:

$$M_1(\vec{x}) = \sum_{\omega=0}^{\Omega} \psi^{\omega-1} N((\vec{x} + \sigma \cdot \omega) \nu \cdot 2^{\omega-1}), \quad (3)$$

where Ω is the total amount of octaves, ψ the persistence (scale falloff), N a generic N -dimensional noise function, σ a predefined random vector that removes anisotropic artifacts and ν the frequency modifier representing the initial largest scales in the model. For each mode, the frequency is thus multiplied, while the overall amplitude

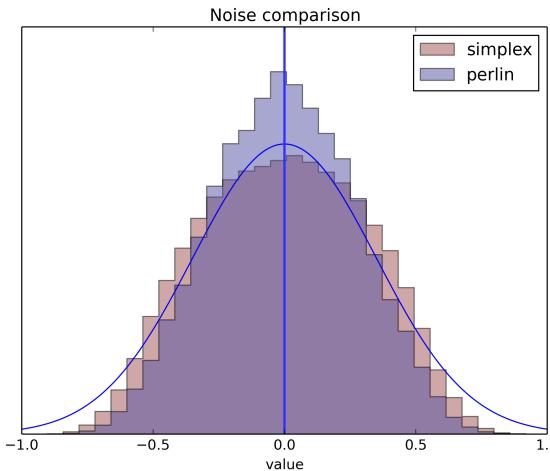


FIG. 2: Normalized distributions of Perlin noise and simplex noise compared with a Gaussian distribution. Note how both noise functions produce near-Gaussian, but slightly different distributions.

is damped by a given persistence. In the end, we simulate a large set of bulk particles without any walls, and use the noise function to remove particles that are not in a procedural void given by a threshold τ . Three examples of this model are depicted in figure 3, where we have used $\Omega = 6$ octaves and the scale is $\nu = 0.01$ and cutoff $\tau = 0$. The left image depicts low persistence ($\psi = 0.0$), where only the first octave dominates (corresponding to large scales). The middle picture shows the same parameters with a higher persistence ($\psi = 0.5$), where smaller scales become more dominating. In the rightmost picture, the effect of the cutoff threshold $\tau = -0.3$ is shown.

D. Model Measure

In order to perform data analysis on the model parameters, we developed a code that uses Monte Carlo Markov chains in order to produce samples for building the likelihood. Before doing so, we need a model measure - a quantitative method that characterizes the internal structure of the porous media. This quantity should be computed for both the original data set and the one produced by our noise model.

Ideally, the model measure should pick up all geometrical properties that are relevant for the physical study of interest. For instance, pore size distribution and porosity are import for studying transport in porous media. Another such measure is the Distance-To-Atom (DTA). The method used for calculating the porosity of a media is defined as in [3], and we obtain the surface area of a porous media by using marching cubes triangulation to sum the surface area of the triangles.

In this paper, we choose to use the radial pore size distribution $g(r)$ as defined and implemented in LAMMPS

[?]. TODO: skriv litt mer om hvorfor vi velger $g(r)$

E. Likelihood

The likelihood \mathcal{L} of parameters θ of a model $M(\theta)$ given outcome x is defined as $\mathcal{L}(M(\theta)|x) = P(x|M(\theta))$, where P is the probability distribution. In our model, we use the χ^2 test to fit our one-dimensional measure from the data \mathbf{d} with our model $M(\theta)$. Pearson's χ^2 is defined as

$$\chi^2 = \sum_i \frac{(d_i - M(\theta)_i)^2}{M(\theta)_i^2}, \quad (4)$$

and the likelihood is thus

$$\mathcal{L} \propto e^{-\chi^2}. \quad (5)$$

For a model with N free parameters, we build the full N -dimensional likelihood by letting random walkers map out the parameter space through the Monte Carlo Markov chain method (MCMC). For each step $M(\theta)_n$ in parameter space, we calculate the likelihood for this parameter configuration. Then, we create a new set of parameters for the next step $M(\theta)_{n+1}$ by choosing new parameters randomly from a N -dimensional sphere around the previous parameters. The model also chooses a new random seed. We calculate the likelihood at the new step and perform the MCMC test: Accept the new step if $\frac{\mathcal{L}_n}{\mathcal{L}_{n+1}} \geq 1$, or reject if the value is lower than a uniform random value $\in [0, 1]$.

III. MODEL VERIFICATION AND DATA ANALYSIS

In order to verify our likelihood code, we perform a model verification test on a mock data set with known input values. In this example, we only use standard simplex noise with three free parameters: the persistence ψ , threshold τ and scale ν . We simulate a data set of 159^3 of 1000000 atoms, and perform the cutoff given input parameters $\Omega = 3$, $\Psi = 0.6$, $\tau = -0.2$, $\nu = 0.03$. The resulting 3D structure is depicted in figure 4.

We continue by performing a full MCMC analysis of the mock data with using a parallelised code running on a 48 core cluster. In order to produce 300 000 samples, about 48 hours was needed. The results from this analysis can be seen in figure 5, where the upper panel depicts the marginalised 1D posteriors while the lower shows the marginalised 2D posteriors. The center and left image shows that the input parameters of $\tau = -0.2$ (threshold), $\psi = 0.6$ (persistence) and $\nu = 0.03$ are all reproduced well within 1σ .

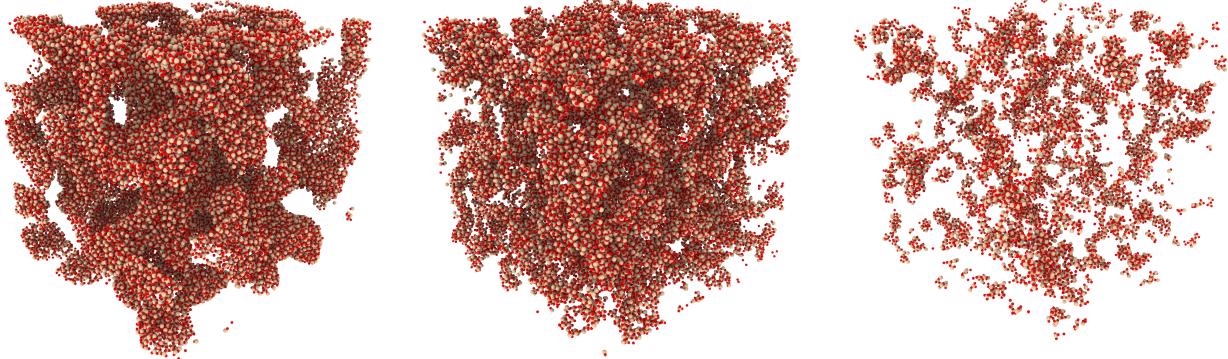


FIG. 3: Examples of 3D simplex noise structures in nanoporous media. Left image: low persistence (large scale dominates), middle image: medium persistence (small scales starts dominating) and right image: high threshold (more atoms are discarded).

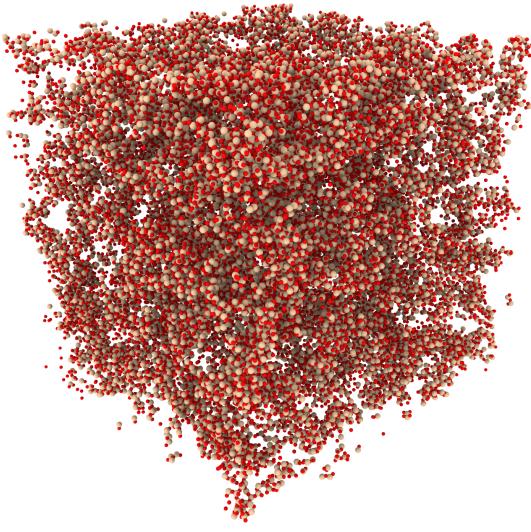


FIG. 4: Simulated mock data used for the initial model testing analysis. Input parameters are $\nu = 0.03$, $\tau = -0.02$ and $\psi = 0.6$.

IV. RESULTS

We now wish to perform a parameter fitting analysis on realistic simulated nanoporous media. Using LAMMPS [?], we simulate a 100^3 Å box of 100K SiO_2 silica particles at an initial temperature of 4500K, using the Vashishta-potential. After an initial period, we expand the box to 159^3 Å and start cooling until quenching is achieved at about 300K. The resulting nanoporous media is depicted in figure ??, with a porosity of 75%.

With the model described in 3, we perform a full likelihood analysis of the data set with the three free parameters τ , ν and ψ using $g(r)$ as the measure. The resulting marginalised posteriors are depicted in figure 6,

with best-fit parameters $\tau = -0.2$, $\psi = 0.8$ and $\nu = 0.3$ is extracted from the full 3D likelihood. We also display the radial pore size distribution $g(r)$ for the input data set, a best-fit parameter data set and a bad-fit data set in figure 7, to illustrated the sensitivity of $g(r)$ given the model parameters. Note that we are not able to accurately reproduce the structures of the SiO_2 $g(r)$ with our current model, but hope that we during future work will be able to produce another model that provides a better fit with this measure.

Finally, we depict a comparison of the actual simulated input SiO_2 data set together with a representation of the best-fit parameters in figure 8

The porosity of the input SiO_2 data set is 75%, while the best-fit data set has porosity $77\% \pm 3\%$, depending on the random seed. In addition, the surface area of the original data set was $\tilde{29000}\text{\AA}^2$, while the best-fit data shows similar values with surface are ranging between $\tilde{28000}\text{\AA}^2$ and $\tilde{30000}\text{\AA}^2$.

A. Some notes on the results

It is worth to note that since we are chiefly interested in obtaining a set of parameters that reproduces a statistical correct representation of the input data, we are relatively free to choose parameters as desired. In the previous analysis, we selected a best fit threshold value of $\tau = -0.2$, but from the width of the likelihood it is clear that nearby parameters would also be suitable. However, the surface area is strongly correlated with this value, so even slight deviations (e.g. $\delta\tau = 0.05$) yields a relatively large change in surface area.

In addition, we do not claim that this model accurately reproduces the pore structure observed in nanoporous simulations, as is evident from the radial pore size distribution shown in figure 7.

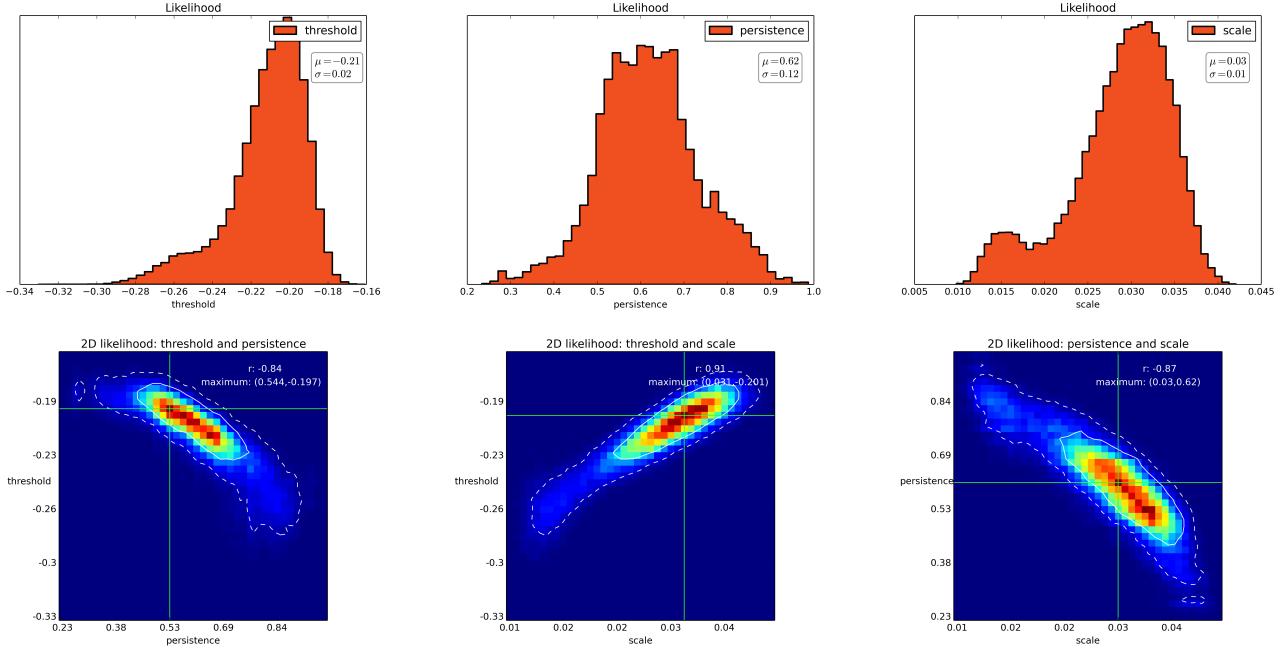


FIG. 5: Results from the model validation simulation. The upper panel displays the marginalised 1D posteriors for each parameter, while the lower panel depicts the marginalised 2D posteriors. Note that all input parameters were reproduced well within 1σ (inner contour line).

V. CONCLUSION

In this paper, we have proposed a novel method for generating a simple kind of nanoporous media using procedural methods, namely Simplex noise. We have developed tools for creating such porous materials, and also estimating model parameters given data. We have shown that it is possible to recreate

A. Future work

Ut hendrerit cursus libero, in scelerisque tortor sollicitudin accumsan. Nam eleifend velit metus, quis volutpat justo faucibus ut. Cras ut sem in nunc fringilla egestas vitae sit amet justo. Ut vel condimentum tortor. Morbi in massa in ipsum ultrices tincidunt. Duis feugiat dignissim nisi ac ultricies. Quisque eget risus fermentum, condimentum massa ullamcorper, posuere neque. Nam congue consequat mi, vel commodo mi tempus eget.

Nullam ultrices, velit sed venenatis ultrices, risus urna ultricies magna, at pretium lorem nibh et nunc. Suspendisse sed mattis leo. Nulla facilisi. Praesent finibus lobortis leo, ut porttitor massa tincidunt ut. Vivamus quis orci eu sem sodales ornare. Cras sed faucibus velit. Phasellus scelerisque mauris quis lorem dignissim vehicula. Sed tempor ut urna sed bibendum. Donec et

nibh pulvinar, commodo magna ac, interdum nibh. Sed laoreet ipsum quis quam lacinia, in dictum erat laoreet. Etiam rhoncus mauris arcu, ac egestas lacus facilisis nec. Pellentesque a orci vitae felis faucibus ultrices sed non massa. In posuere vel ex in elementum. Aenean ac neque a enim gravida tristique eu ac lacus. Ut tincidunt venenatis lectus, et maximus mauris.

Proin ornare nulla id tristique placerat. Integer nec elementum tellus. Cras at tempus lectus. Cras lorem est, ornare id condimentum vitae, scelerisque eu lectus. Aenean libero arcu, consequat vel sagittis eget, sagittis ut neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nec quam sodales, mollis est sit amet, vulputate turpis. Praesent tempus diam eu sem laoreet finibus. Nulla finibus, sapien vel lobortis ultrices, mauris ipsum aliquet urna, quis rutrum sapien dui id ipsum. Quisque sodales ex sapien, ac egestas lacus suscipit et. Suspendisse accumsan rhoncus tristique. Pellentesque ut eros lectus. Praesent viverra sit amet odio nec vulputate. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nulla molestie metus vitae libero vulputate, at interdum enim tempus.

B. Simulated data

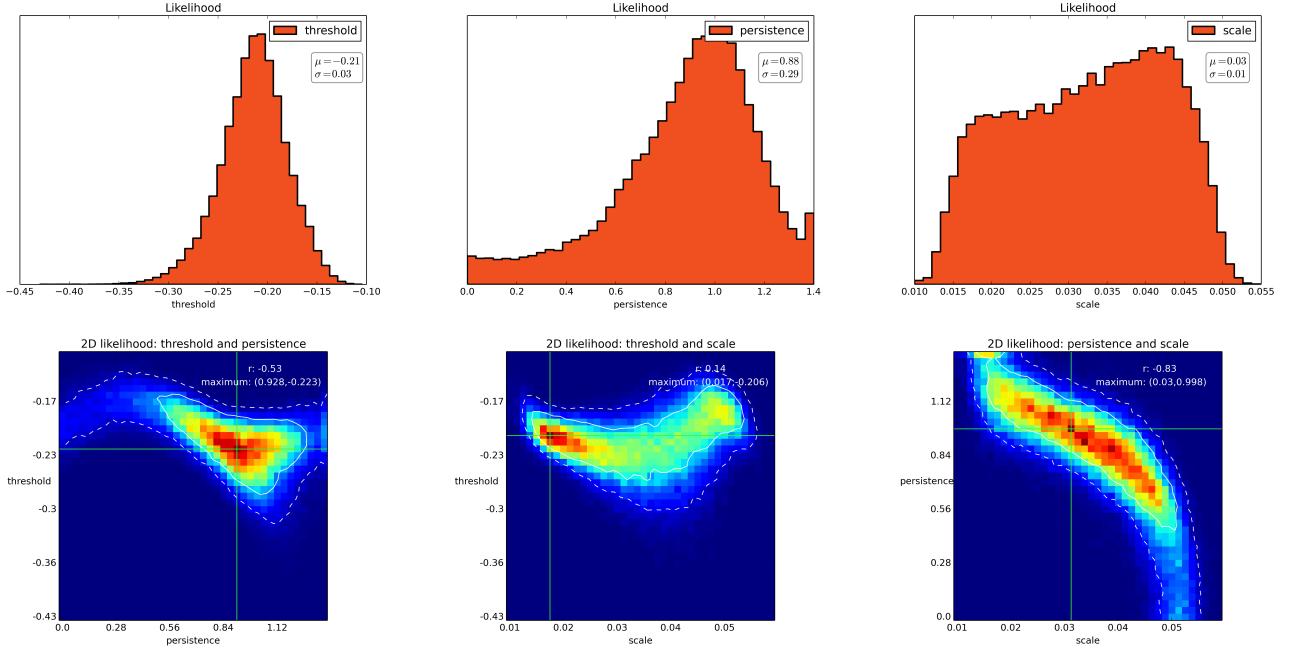


FIG. 6: Results from the analysis of a simulated 159^3 Å SiO^2 nanoporous data set. The upper panel displays the marginalised 1D posteriors for each parameter, while the lower panel depicts the marginalised 2D posteriors. Best fit parameters from the maximum 3D likelihood suggest $\tau = -0.2$, $\psi = 0.8$ and $\nu = 0.3$.

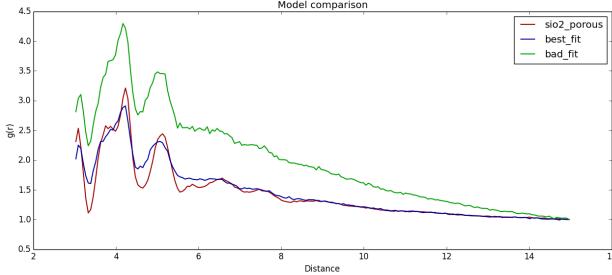


FIG. 7: The radial pore size distribution $g(r)$ for three data sets: the simulated SiO^2 input data set, the best-fit parameter data set and a random low-fit parameter data set.

-
- [1] A. Lagae, S. Lefebvre, R. Cook, T. DeRose, G. Drettakis, D. Ebert, J. Lewis, K. Perlin, and M. Zwicker, Computer Graphics Forum **29**, 2579 (2010), URL <http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2010.01827.x/abstract>.
- [2] K. Perlin, ACM Trans. Graph. **21**, 681 (2002), ISSN 0730-0301, URL <http://doi.acm.org/10.1145/566654.566636>.
- [3] L. D. Gelb and K. Gubbins, Langmuir **14**, 2097 (1998).
- [4] L. D. Gelb and K. Gubbins, Langmuir **15**, 305 (1999).

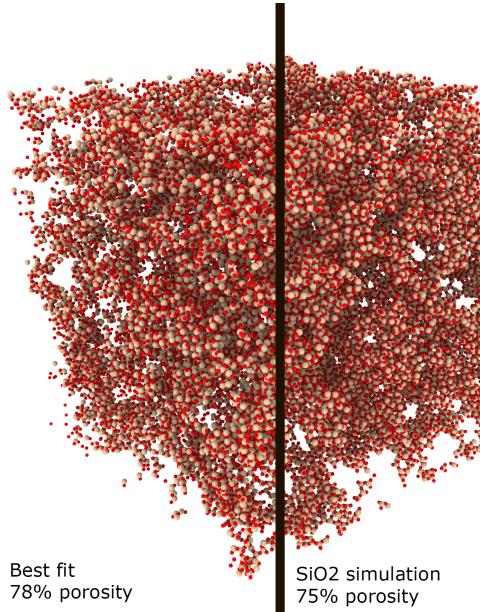


FIG. 8: Right side: A realisation of the best-fit parameters $\tau = -0.2$, $\psi = 0.8$ and $\nu = 0.3$. Left side: Simulated SiO^2 input data set used for parameter estimation.

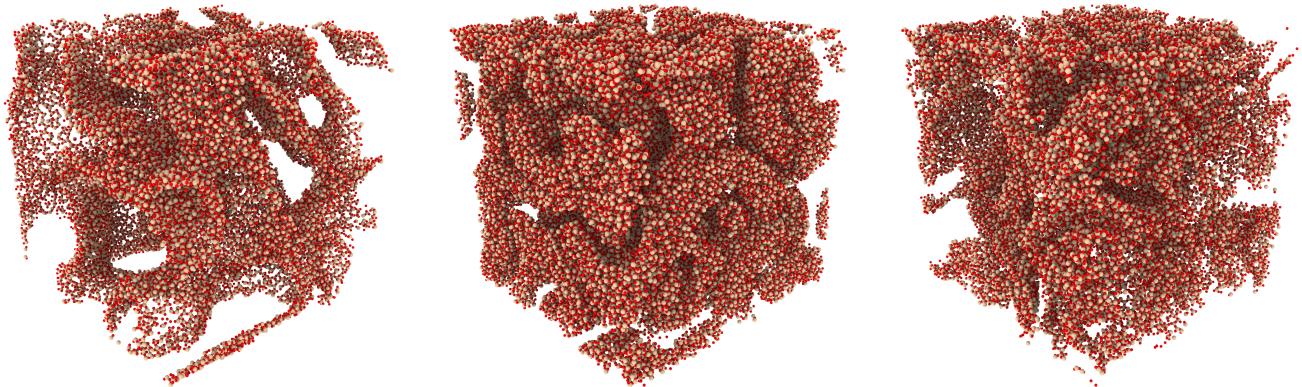


FIG. 9: Example of future models. Left panel: A multifractal model with thin surfaces and large pores. Center panel: a multifractal with cave-like structures. Right panel: A model with strong asymmetric noise.