PoreFlow-Net: a 3D convolutional neural network to predict fluid flow through porous media

Javier E. Santos[1], Duo Xu[2], Honggeun Jo[1], Christopher J. Landry[3], Maša Prodanović[1,3], Michael J. Pyrcz[1,3]

[1]. Hildebrand Department of Petroleum and Geosystems Engineering, The University of Texas at Austin
[2]. Department of Astronomy, The University of Texas at Austin
[3]. Center for Petroleum and Geosystems Engineering, The University of Texas at Austin

Keywords (6):
Fluid flow, porous media, surrogate models, permeability, deep learning, convolutional neural networks

Abstract

We present the PoreFlow-Net, a 3D convolutional neural network architecture that provides fast and accurate fluid flow predictions for 3D digital rock images. We trained our network to extract spatial relationships between the porous medium morphology and the fluid velocity field. Our workflow computes simple geometrical information from 3D binary images to train a deep neural network (the PoreFlow-Net) optimized to generalize the problem of flow through porous materials. Our results show that the extracted information is sufficient to obtain accurate flow field predictions in less than a second, without performing expensive numerical simulations providing a speed-up of several orders of magnitude. We also demonstrate that our model, trained with simple synthetic geometries, is able to provide accurate results in real samples spanning granular rocks, carbonates, and slightly consolidated media from a variety of subsurface formations, which highlights the ability of the model to generalize the porous media flow problem. The workflow presented here shows the successful application of a disruptive technology (physics-based training of machine learning models) to the digital rock physics community.

1. Introduction

Understanding how fluids travel through porous structures of subsurface rock formations is crucial for designing groundwater management, hydrocarbon extraction (Raeini, Blunt, & Bijeljic, 2014), CO_2 sequestration (Chen, Li, Valocchi, & Christensen, 2018), and contaminant remediation projects (Kang, Lichtner, & Zhang, 2007). Currently, most of the energy that we use comes from hydrocarbons extracted from oil and gas reservoirs, most of the water for human consumption travels through underground aquifers, and the first pilot projects of CO_2 sequestration in the subsurface are yielding positive results. For these reasons, it is paramount to accurately describe the flow physics of these fluids to maintain energy security, water availability, and to potentially avoid climate change (Blunt, 2017).

One of the most impactful properties in the decision-making process for the areas mentioned above is the permeability of the underground reservoir of interest. This quantity provides a directional, volume-averaged geometric measure of the ease for a fluid to flow through a specific rock volume. The permeability is determined by the topology of the porous structures of the formation, and it is calculated by computing average velocity (based on the fluid velocity through pore space) and comparing it to Darcy's law (see Equation 1). This quantity is researched primarily to assess preferential flow channels in the subsurface (contaminant tracing, hydrocarbon movement in an oil reservoir), bottlenecks for fluid flow, and to estimate well flow rates (hydrocarbon and water extraction, and CO_2 sequestration). The permeability is shaped by the processes that generate and preserve the rock formation, and the subsequent alterations throughout geological time. Processes such as deposition of grains in a basin, compaction of layers caused by overburden pressure, cementation, recrystallization, and dissolution, change the microscopic structure of the rock, altering the shapes and sizes of the flow paths available. These effects modify the permeability of the rock formation (that can span up to kilometers). Since the behavior of the fluids at the smaller scales is key to make inferences of larger domains, in this paper we are going to focus on the flow of fluids at the microscale.

There are different methods to obtain the flow properties of a rock. Laboratory measurements are able to obtain the average permeability of a sample through direct measurement. Nevertheless, it is not possible to observe the microscopic physics at the pore-scale. These laboratory measurements also tend to take longer times or even fail in tight porous media (lower porosity). On the other hand, there are existing analytical expersions that estimate the permeability of a rock based on fitting parameters that account for the rock type (lithology), grain size distribution, and depositional processes, among others. These require a minimal amount of information, but they are restricted to a particular rock (sometimes even from a specific geographical location) (Xu & Yu, 2008). Finally, there are several numerical simulation methods to reproduce the fluid flow physics (Blunt et al., 2013; Mehmani, Verma, & Prodanović, 2020). Among these, direct simulation methods (DSM) are very attractive because they resolve the flow through irregular geometries, giving the final user a realistic snapshot of how the fluid flows through the pores of subsurface formations. Since the subsurface is highly heterogeneous over multiple scales, direct simulation on a variety of samples at various scales extracted from wells or outcrops of the reservoir or analogous rock of interest provides valuable information to investigate and model subsurface flow for improved subsurface management.

With the rapid development of x-ray scanners and other non-destructive imaging technologies (Mees, Swennen, Geet, & Jacobs, 2003), the simulation of fluid flow through 3D images of porous materials is a topic of increasing interest. The typical workflow for performing direct simulations starts with a gray-scale volume (the output of the x-ray scanner), which is then segmented (to eliminate artifacts and noise) in two phases (binary image) that are discretized into voxels (3D pixels) of solid or space for fluid to flow. These simulations provide an accurate picture (with resolution of micrometers, and even smaller) of how fluid flows through complex geometries. With the advances in computational performance, larger domains are practically simulated. Nevertheless, computing times (even on supercomputer clusters) can be long, and the required computational resources are vast. The computational demand of these methods grows at least at the cube of the side length of the domain for homogeneous cubic samples, so in most cases running direct simulations on a representative elementary volume with typical desktops is unfeasible. Additionally, real materials tend to have pore size distributions that span and vary over a wide range of scales, which increases the size of a representative elementary volume, and thus the computational time to perform the simulations.

There are several numerical methods that are used to obtain flow properties directly from 3D images: the finite volume method (Jenny, Lee, & Tchelepi, 2003), smoothed particle hydrodynamics (Tartakovsky & Meakin, 2005), the finite element method (White, Borja, & Fredrich, 2006), the lattice Boltzmann Method (LBM), among others. A comparison of some of these methods, and their run times can be found in (Yang et al., 2016). In this work we utilize the LBM due to its simplicity for performing simulations in irregular domains, and its well-tested capabilities to simulate flow through porous materials (Pan, Hilpert, & Miller, 2004; Santos, Prodanović, Landry, & Jo, 2018). Although the method is easily parallelizable, its computational time scales increase with domain complexity (Figure 1), which is common to every method that operates on porous materials. We stress, however, that the workflow presented here does not depend on the method chosen to obtain the fluid velocity field.

Recently, deep learning methods have been introduced as a framework for computers to learn from observational data of physical phenomena to predict variables of interest. These methods have been applied to study many problems in image segmentation, pattern recognition and image captioning, and natural language processing. Deep learning methods benefit from benchmark datasets since: (1) supervised deep learning methods require a large amount of validated data to train models; and (2) the capabilities of the trained classifiers must be assessed quantitatively. These algorithms have been applied successfully to digital rock applications like image segmentation (Andrew, 2018; Bihani et al., 2019; Karimpouli & Tahmasebi, 2019b), calculation of wave propagation through a solid matrix (Karimpouli & Tahmasebi, 2019a), 3D rock reconstruction using generative models (Mosser, Dubrule, & Blunt, 2018), and 2D calculations of permeability in small domains (Wu, Yin, & Xiao, 2018). Either segmented real images or porous media reconstructions are required for direct simulation of flow. There are several challenges encountered in applying deep neural networks to predict flow through porous media (or upscaled transport properties of a porous medium). The biggest challenge is the large number of labeled pairs of data (that can come in the form of interpreted seismic cross-sections, segmented images, simulation results, etc.) required to train a model. In addition, performing numerical simulations of porous volumes could require days of computation on hundreds of cores of a supercomputer to converge (Figure 1). Moreover, acquiring the prerequisite many volumes of a similar formation is often challenging, since access to the required imaging technologies (i.e. x-ray scanners) is limited, and finally, given access to a large training set, there is still a memory limitation challenge (more on this in the sections below). To circumvent the above difficulties, we create benchmark datasets reusing images from Digital Rocks Portal (Prodanovic, Esteva, Hanlon, & others, 2015) that are publicly available, and propose a comprehensive workflow to obtain a functional relationship between a 3D binary image and the volumetric solution of the Navier-Stokes equation.

In the context of fluid flow, Carrillo et al (Carrillo, Que, González, & López, 2017) trained an artificial neural network to predict the shape and coordinates of an occlusion blocking a 2D pipe, using only the velocity at points along the horizontal direction (representing sensors) as input data. Moreover, Guo et al (Guo, Li, & Iorio, 2016) trained a convolutional neural network (CNN) to predict velocity fields of a steady state flow with an obstacle (represented by simple geometries) for small domains with closed boundaries, they used the distance transform of the binary image as the model input. For single-phase, time-dependent problems, Hennigh (Hennigh, 2017) proposed the Lat-Net, a convolutional neural network architecture that compressed the output of an LBM simulation (to be memory efficient), and learned the relationship between subsequent (compressed) time steps. Specifically, for porous media applications, Wu (Wu et al., 2018) applied a CNN architecture with a fully connected layer to predict the permeability of 2D images. Sudakov (Sudakov, Burnaev, & Koroteev, 2018) applied simple 2D/3D architectures to predict the absolute permeability a system obtained by a pore-network model (a technique which simplifies the pore space into a network of spheres interconnected by cylinders, losing all the complex features of the image). The authors of this paper (Santos, Prodanović, & Pyrcz, 2018) initially proposed a CNN that used the Euclidean distance as an input to predict the velocity field. Nevertheless, the network was not able to generalize to predict for more heterogeneous pore geometries. Kamrava et al (Kamrava, Tahmasebi, & Sahimi, 2020) showed that by using 3D convolutions their model was able to predict permeability for realistic pore geometries. That paper also provides a detail explanation of all the main components of a convolutional neural network and we refer it to any reader who is not familiar with the basic structure of a neural network. The key difference of our work is that we are able to use large 3D domains with pore geometry that is more complex than in previously published work. Further, compared to other porous media work to date, we are able to predict the fluid 3D velocity field, instead of only trying to predict the permeability value.

In this work, our main contribution is a new 3D deep learning workflow that is able to generalize the single-phase flow of a fluid through granular materials. We show that by combining a feature extraction algorithm, a custom loss function, and a new network architecture, our model can be trained with very simple 3D geometries, and predict accurately in examples of varying sizes and complexity. These predictions require less than a second of computation on a typical desktop computer with a graphics processing unit (GPU), and are comparable in accuracy to the full-physics simulation that might require days of processing on a supercomputer cluster. We will also provide a comprehensive 3D data set that spans a wide range of rock formations all around the globe.

2. Methods

In this section, we present the numerical method used for simulating the flow physics, the morphological feature extraction algorithm, and the architecture of the PoreFlow-Net.

2.1 Velocity field simulation

To simulate the fluid flow through the domains of interest, we selected the lattice Boltzmann method (LBM) (Sukop & Thorne, 2007). Nevertheless, the results of this work are independent of the numerical method used to solve the flow physics. The LBM is one of the most popular methods for performing direct simulation of fluid flow through irregular geometries. This method simulates the streaming and collision of particles on a grid, and it has been demonstrated that is able to recover the full Navier-Stokes equation solution (Frisch, 1991). The advantages of the LBM are that the algorithm is relatively easy to implement, is highly parallelizable, and it can perform direct simulations on images.

We used the same model proposed by Pan (Pan, Luo, & Miller, 2006) with a relaxation time (related to the fluid viscosity) equal to one. It is a slightly compressible model, where a very small pressure gradient (1e-6 lattice units, independent to the permeability of the domain) is applied to drive the fluid forward. All the simulations are in the laminar flow regime (where the Reynolds number is much smaller than one). This is consistent with the typical flow regime through subsurface formations away from fractures or boreholes.

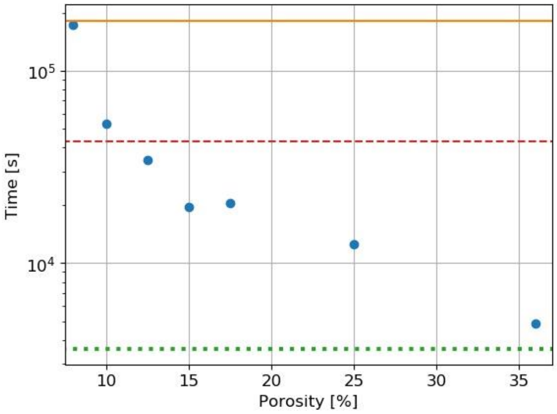

Figure 1: LBM running times (in seconds) for different domains of the same computational size (spherepacks and tight sandstones of 500^3 cells) indicating the impact of porosity (of our particular domains) on computational time. In these domains, the lower porosity samples host more intricate pathways (pore space with higher surface area and higher tortuosity), which increase the number of LBM iterations needed to achieve convergence. We run the simulations in eight Xeon E5-2690 v3 (Haswell) processors totaling 96 computing cores (https://portal.tacc.utexas.edu/user-guides/lonestar5). The dotted horizontal line represents an hour, the dashed line a day, and the solid 2 days of running time. The sample that took the longest had very tight pore throats and a low coordination number between pores (resulting in very poor connectivity), this yields in a large number of iterations for the momentum to equilibrate.

Upon convergence, the LBM simulation outputs the 3D velocity field tensor of the image. To calculate the permeability of the domain, we use Darcy's law (Bear & Bachmat, 1991):

$$k = \frac{\bar{v}}{\mu}\frac{dp}{dz} \quad (1),$$

where $\bar{v}$ represents the mean of the velocity field in the direction of the pressure gradient $\frac{dp}{dz}$, and μ refers to the dynamic viscosity of the fluid. To calculate $\bar{v}$ we calculate the average of the 3D velocity matrix in the direction of flow. The permeability expresses the flow rate as a function of pressure gradient, it has units of length squared, and it is typically expressed in *m*^{2} or in *Darcys*.

2.2 Feature extraction

The typical bottlenecks for deep learning applications are the: (1) vast amount of data required to train a model, and the (2) memory limitations of the computational systems to perform the training of a deep neural network. To overcome these issues, we added to our workflow a pre-training feature extraction step where we extract relevant morphological features of the rock volume.

Since our simulations are time consuming (spanning from hours to days in our cluster), it would be impractical to run domains hosting every possible 3D structure. By adding additional input features to the model, our network is trained to find a more robust functional relationship of the image with the flow field. It is worth noting that these features are computed in seconds, requiring a minimal computational demand compared to the fluid flow simulation. Moreover, since it would be computationally difficult to train the model using the entire simulation domains (>500^3 voxels), we split the input and output images in subsamples to carry out the training process. Since the subsampled volumes are shuffled in a training pool along with other examples from different domains, including information of the boundaries (local with Euclidean distance, and global with the time of flight) gives the model knowledge about the original spatial location of the individual subsample (this process is depicted in Figure 5).

We compute four geometrical features from the binary image (Figure 2). To represent the local characteristics of the binary image, we extracted the Euclidean distance map (also known as the distance transform) of each sample. This is calculated with the following equation:

$$E_{dist} = \left((\mathrm{x}_1 - \mathrm{x}_2)^2 + (\mathrm{y}_1 - \mathrm{y}_2)^2 + (\mathrm{z}_1 - \mathrm{z}_2)^2 \right)^{\frac{1}{2}} \quad (2).$$

Where $\mathrm{x}_{1\ \mathrm{and}\ 2}$, $\mathrm{y}_{1\ \mathrm{and}\ 2}$, $\mathrm{z}_{1\ \mathrm{and}\ 2}$ are the coordinates of each point of the solid and the fluid boundaries respectively. This map provides a compact representation of the distribution of space available for fluid to flow, and the distance to the closest solid (no-flow) boundary. Next, a maximum inscribed sphere (MIS) map in the direction of flow (i.e. an MIS flood) is computed. This map is a simplified and lightweight representation of a non-wetting fluid injection in the direction of flow. Although MIS floods are typically used to describe two-phase flow, here it acts as a measure of geometry (size of pore space) and topology (connectivity to neighboring pore structures to similar size). The MIS map provides information about the local pore space characteristics, as well as the global simulation conditions. It acts as a bridge between the whole domain and its subsamples. Finally, to inform the network about the global conditions of the domain before subsampling it, we employed a detrended time of flight (ToF). We use the fast marching algorithm (Hassouna & Farag, 2007) to compute the shortest distance of all the points of the domain to a point source (in this case, either the XY-plane located at the inlet or the outlet). This method solves the boundary value problem of the Eikonal equation (Hassouna & Farag, 2007) represented by:

$$|\nabla t(x)| = \frac{1}{f(x)}\ , \quad (3)$$

Where *t* represents the time of flight and *f(x)* stands for the speed at every location of the image (a constant in our case). For our case, the speed of the void space is set to one, while the solid matrix is set to zero (impermeable). The result of this operation is a map where each of the voxels of the void space are labeled with a number that depicts the shortest distance (in voxels) to the boundary (the first few layers in the z-coordinate will be given consecutive numbers starting from one, until they find a solid obstacle, then the number sequence will continue around the obstacle). We then subtract the time of flight of the image map without solid obstacles (an image with a porosity of 100%), to calculate a detrended (normalized) map as shown in Figure 02. This feature provides data on tortuosity of the global paths within the domain. In addition, it supplies the model implicit information about the neighboring subsampled blocks. We compute two features using this method. One, where the point source is located at the inlet of the numerical simulation, and the second one where the source is at the outlet (both pressure gradient boundary conditions).

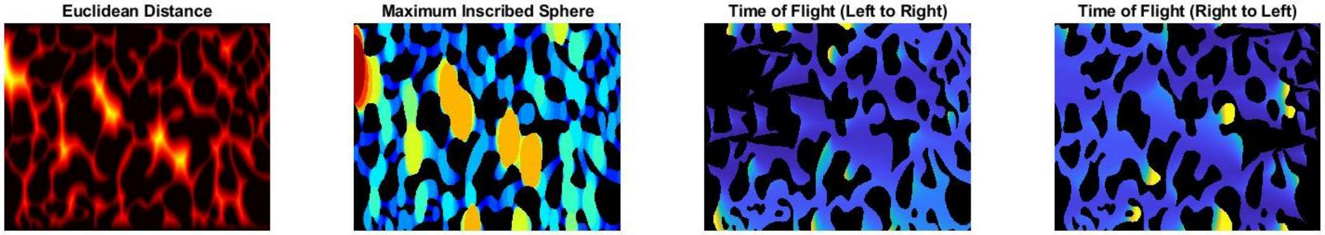
Euclidean Distance
Maximum Inscribed Sphere
Time of Flight (Left to Right)
Time of Flight (Right to Left)

Figure 2: 2D example of the four morphological features (Euclidean distance, maximum inscribed spheres, and time of flight from left and from right, respectively) that we compute from a binary image (Liu, Wang, & Song, 2017) to train the network. The areas where the value of the matrix is zero (i.e. solid phase voxes) are shown in black.

These features have been used in literature to characterize porous materials. Nevertheless, since the relationship of these features with the velocity field is highly non-linear, the selection of the ultimate set of features shown above was a trial-and-error process. These features do not provide an exhaustive description of a 3D porous material. However, they deliver enough information to our model about the local and global boundary conditions of the domain to be able to structure a relationship (in the form of a convolutional neural network model) between these inputs and the Navier-Stokes solution.

2.3 Network

2.3.1 Convolutional neural networks

Convolutional neural networks (CNNs) have excelled in the field of computer vision outperforming classical machine learning methods (Krizhevsky, Sutskever, & Hinton, 2012; LeCun, Bengio, & Hinton, 2015). These models have shown a remarkable capacity to find complex relations in big data sets. By utilizing the discrete convolution operation instead of a regular matrix multiplication (i.e. a fully connected feed-forward network), they generalize local spatial relationships (sparse interactions) across the domain. CNNs utilize filters that are much smaller than the input image, which extract general and meaningful information about the domain in an efficient manner. By stacking convolutional layers, the network extracts features at different levels of abstraction with an increasingly wider receptive field (Figure 03). Finally, the convolution layers are equivariant to translation, which means that if the input feature is shifted, their output will be shifted by the same amount (by creating, in this case, a 3D feature map). This is particularly useful in pattern recognition, because they allow for inputs of variable size. Using this structure, a network can be trained to learn complex, non-linear relationships between inputs and outputs using the backpropagation algorithm.

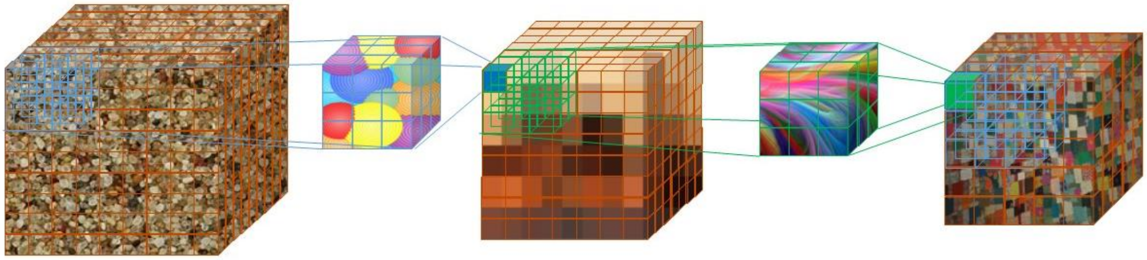

Figure 3: Schematic of three subsequent convolution operations with a 3x3x3 filter and a stride (kernel distance of where the next convolution operation is performed) of two. The network is trained to create a more compact (latent) representation, while retaining relevant features of the original image. Although the image loses the structure to the human eye, it retains the most significant information to the network. This operation allows to capture local and global spatial relationships by convolving over the output of the previous convolutional block. It is also cheaper to train because it has a smaller number of parameters (smaller filters) compared to a fully connected network.

2.3.2 PoreFlow-Net

Recent studies suggest that the performance of a network can benefit from increased depth (longer stack of layers, as described in Section 2.3.1) (Szegedy et al., 2015; Urban et al., 2016). Apart from being computationally more intensive, a deeper network presents issues like vanishing and exploding gradients (Pascanu, Mikolov, & Bengio, 2012), and filter saturation by highly correlated features, making them very hard to train. To improve the gradient propagation and to enhance the training, He et al. (He, Zhang, Ren, & Sun, 2016) proposed the residual network (ResNet). The ResNet concatenates an identity map to the output of a convolutional layer stack (residual unit) to facilitate training. The authors show that the training is eased by targeting this new referenced residual output, avoiding gradient vanishing or saturation. Further, Ronneberger et al. (Ronneberger, Fischer, & Brox, 2015) proposed the UNet. This architecture concatenates feature maps from different layers of the encoding branch to the decoder, improving segmentation accuracy significantly. One of the main advantages of this is that the structure of the network retains high (i.e. lines and edges) and low-level features (i.e. entire objects) to reconstruct the output. They show that the networks train with ease and with fewer parameters due to the better flow of information (both in the forward and backward computations) that the skip connections (direct pathways between the encoding and decoding branch) provide. Building up from these two architectures, Zhang (Zhang, Liu, & Wang, 2018) presented the Deep Residual U-Net (ResUnet) which uses residual units as building blocks and skips connections between them. This network prove to be easy to train (compared to the U-Net that needed extensive data augmentation or a pre-trained model), with an efficient number of parameters and showed accurate results using a small training set.

In this paper, we propose a modification of the ResUnet, which benefits from the information of all the input features by passing them through individual encoding branches (dedicated to each of the extracted features from Section 2.2) with skip connections. We use three residual units for each of the four branches, a bridge, and a single decoder to recover the velocity field. Each of these parts are built with residual units (Figure 04). We use the scaled exponential linear unit

(SeLu) (Klambauer, Unterthiner, Mayr, & Hochreiter, 2017) as the activation function. This is described by the following equation:

$$SeLu(x) = \lambda \begin{cases} x & if\ x > 0 \\ \alpha e^{x} - \alpha & if\ x > 0’ \end{cases} \qquad (4)$$

where the values of α and λ are fixed and provided in the publication. The purpose of this function is to perform additional internal normalization of the inputs, facilitating gradient propagation. According to the derivation of the authors, problems like gradient exploding or vanishing are mathematically infeasible. Moreover, since internal normalization is cheaper, the network converges faster.

Since the velocity distribution spans several orders of magnitude (Figure 6), we use L1 (mean absolute error) as the cost function due to the large number of outliers (velocity tending to zero near the grain boundaries). To increase the attention in tighter geometries we compute the loss as follows:

$$L = \sum\left(\left|y_{true} - y_{pred}\right| * M\right), \qquad (5)$$

Where *M* is a weight vector that accounts for the size of the pores in the direction of flow and $*$ stands for an element-wise multiplication. The algorithm to calculate M can be found in Appendix A1. The loss function (Equation 5) weights the difference between the true values and the predictions so that all the voxels in the training pool have the same relevance (high and low porosity subsamples).

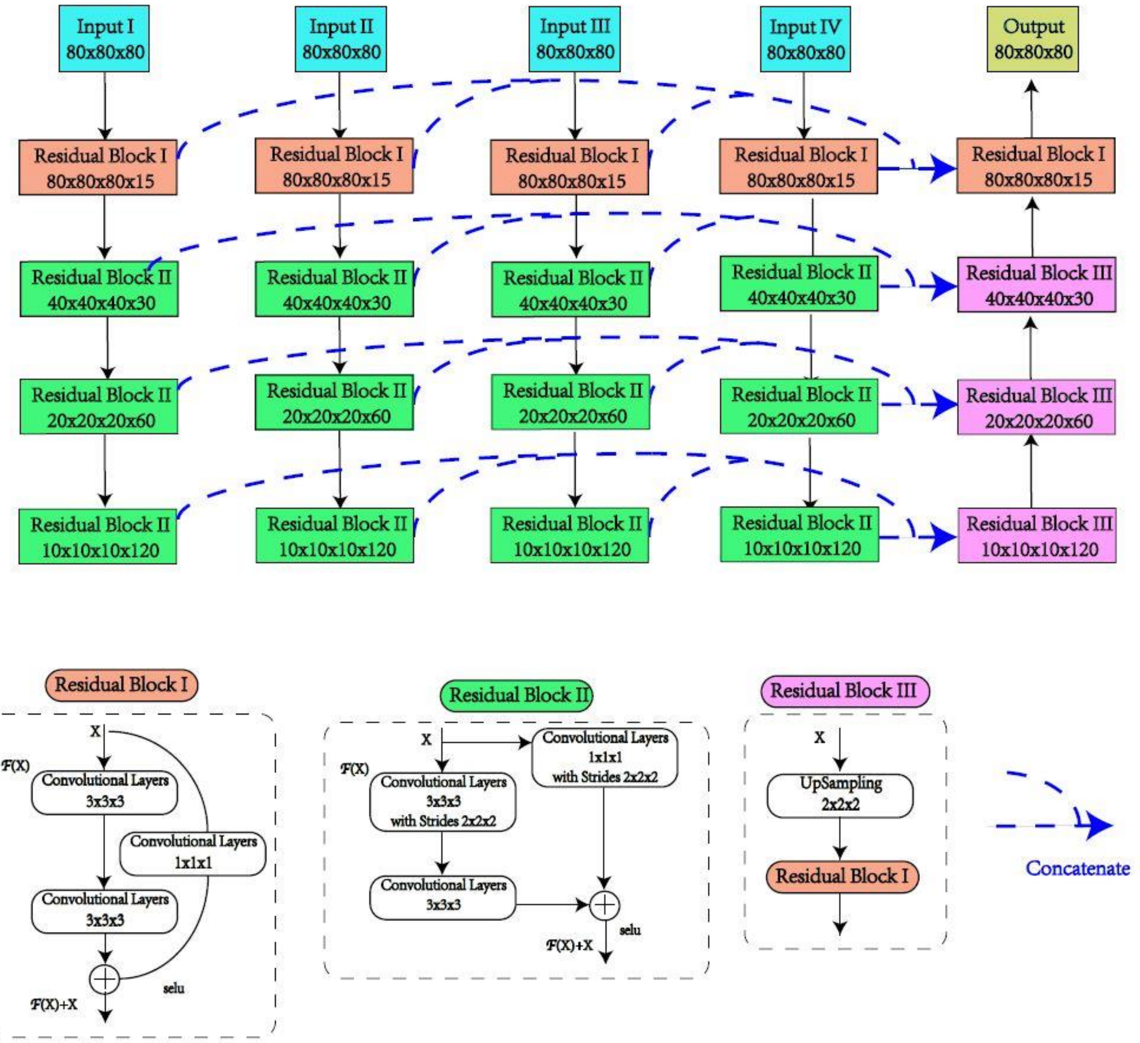

Figure 4: PoreFlow-Net architecture consisting of four input branches and one decoder. A batch normalization layer and an activation follow every convolutional layer. Instead of using deconvolutional layers, the decoder resizes the image using upsampling. It has been shown (Gauthier, 2015) that the pixel overlap in deconvolution layers causes artifacts which increase in higher dimensions. In the case of a 3D deconvolution, some voxels will get six times the number of input information (because the filters visit these locations several times) compared to their neighbors. Since these operations have multiple channels, the network struggles to learn the appropriate weights to reconstruct the output image without artifacts. To avoid these artifacts and increase the memory efficiency, our image is resized using an upsampling layer, which repeats the input by a factor of two in all the coordinate directions (with no trainable parameters, making it cheaper).

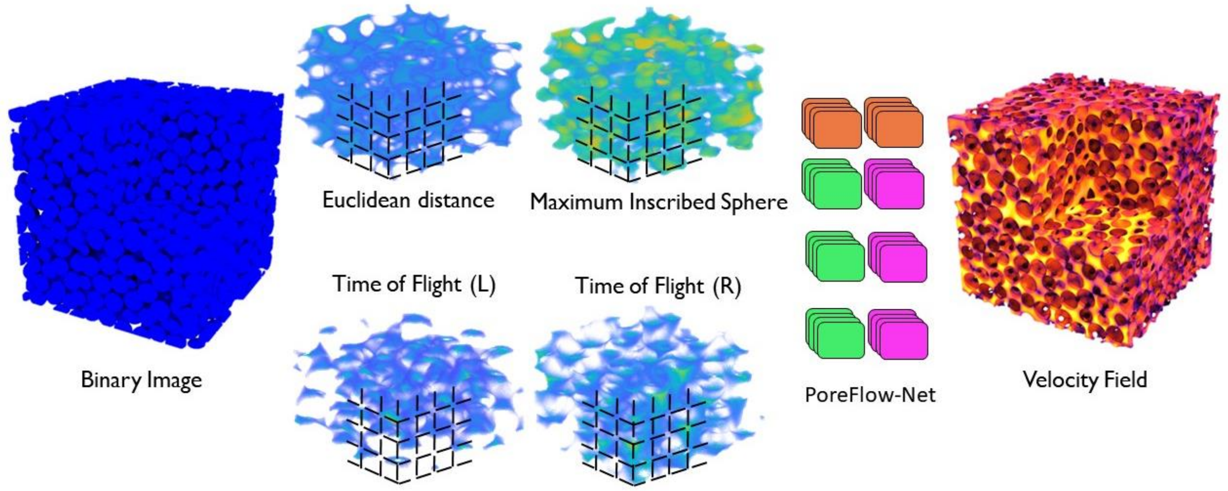

Figure 5: Our workflow. Starting from a binary 3D matrix (left), we compute four geometric features (Section 2.2). The two on the top describe the medium locally, while the two bottom ones provide information about the global domain. These features are computed on the fly for every sample. Then, these features are subsampled (black lines) to train the neural network model. The output is the fluid velocity field in the direction of the pressure gradient. With different colors, we highlighted the different orders of magnitude of the velocity field prediction.

2.4 Training data

2.4.1. Dataset creation

We used a beadpack comprised by a disordered closed pack of spherical grains, originally imaged experimentally by John Finney (Finney & Prodanovic, 2016) as our initial domain. A 500^3 subset of the original spherepack was discretized and segmented to generate training data.

We performed four one-pixel grain dilations to the original sample, where we obtained four images of decreasing porosity (increasingly tighter) that mimic cementation processes in the subsurface, but preserves the simple features of the original spherepack. These samples range from 29.8% to 11% porosity. Finally, we performed a single-phase LBM flow simulation in these four samples where a pressure gradient parallel to the z-coordinate direction was applied with a no-flow boundary condition in the other faces. Since the domains are homogeneous packs of spheres, the simulation converges much faster (in the order of hours) than for real rock x-ray scans. We used these four samples to train the convolutional neural network.

2.4.2 Relationships between inputs and outputs

The lattice Boltzmann simulation outputs a pressure matrix and a velocity tensor in each point of the grid. In this work, we focused on the z-component of the fluid velocity (parallel to the pressure gradient) which determines the permeability. In Figure 6 we show the velocity distribution and its relationship with the morphological features extracted from the binary image.

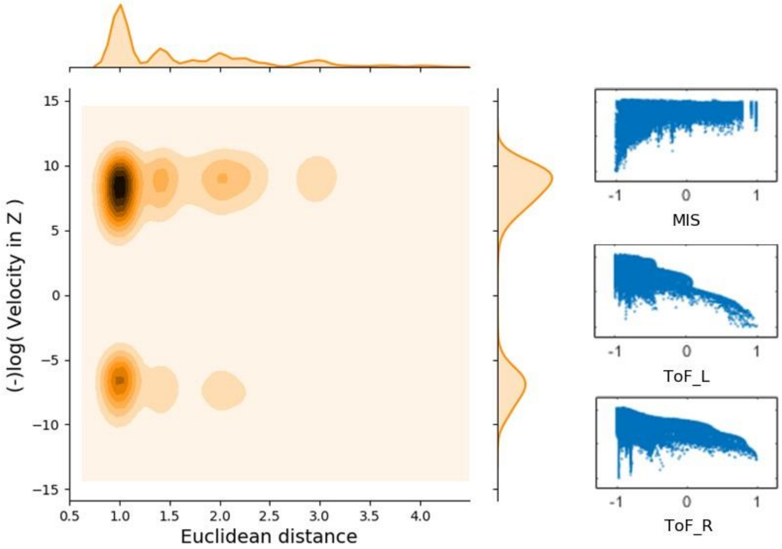

Figure 6: In the left side, a heat map of the signed velocity logarithm (smaller absolute values represent higher velocities, sign represents direction) versus Euclidean distance. The velocity has a bimodal (positive and negative directions) distribution; hence it has a non-unique relationship with the Euclidean distance. Consequently, the plot shows a higher scatter around the small velocities and Euclidean distances. To the right, the scatterplots of maximum inscribed sphere and Time of Flight versus velocity. Heteroscedastic, multimodal behaviors and non-linear correlations are observed.

From the relationships exhibited in Figure 06, we can confirm that traditional machine learning methods would not be able to obtain an accurate model due to the complex, highly non-linear relations between the inputs and the target output.

3. Results

3.1 Impact of the proposed 3D feature extraction

As stated above, it would not be feasible to train our network over the entire simulation domains. Hence, it is necessary to subsample the 3D matrices into smaller volumes to train the model in batches of data. The reasoning behind this is that GPUs have a limited amount of memory, and the model parameters, the inputs and outputs, the gradients, among others must be locally stored. In our experiments, the maximum subset size that conventional hardware could accommodate in memory was no larger than 80^3. The model, as described by Figure 4 requires 2.7 gigabytes (Gb) of memory to be trained with a batch size of one sample. To train the CNN with entire simulation domains one would require approximately 660 Gb of memory available, which greatly exceeds the current capabilities of graphic processing units.

Spatially aware neural networks, benefit greatly from stationary samples because it is easier to find matching patterns in data. In theory, the subset size of the 3D sample should be equal or larger than the representative element volume (REV) (Bachmat & Bear, 1987) to exhibit a stationary behavior of the property of interest (in this case, velocity), impacting in a positive manner the training performance of the neural network. If the subset data size is smaller than the REV, we cannot expect to have a stable measure due to the non-stationarities (in the form of spatial heterogeneities) present in the data.

To show the importance in the training of the network of our proposed feature extraction step, we carried out a moving window analysis to assess the variability of the domains with different volume sizes (Pyrcz & Deutsch, 2014). Using a window of increasing size length, we calculate the coefficient of variation (the ratio of the mean over the standard deviation) of the porosity and fluid velocity within the subset. We carried out this experiment in the original spherepack (36% porosity, before the grain dilations where performed). We executed this procedure iteratively until the variation became not significant. We plotted the results of the moving window analysis in Figure 7. Both of the subplots show the decrease in variability of porosity and velocity respectively with the increase in the size of the window (due to the homogeneity of the sample). The 3D subset size comparison is shown in Figure 8. It is only after 200^3 voxels per side (40% of the image side length) that the velocity field stabilizes (coefficient of variation is less than 1%). This behavior is more significant in tighter and less homogenous samples.

Training the neural network using only the binary image of solid and pore as input will hamper the training process resulting in overfitting (it fails to generalize, causing the training and validation curves to diverge), and poor predictive performance (we carried out this experiment and the results are plotted in Figure 9). Since we are interested in creating a predictive model that is able to perform in different geometries, we show that using the additional inputs (which add additional information about the subvolume as well as how it relates to parts of the image surrounding it, the latter in the form of time of flight) described in Section 2.2, the model increases its training performance, and generalizes enough to predict the flow field (within acceptable error range) in a test set that includes various geometries. In other words, the model is able to find unique patterns to construct a robust function mapping the image with the fluid velocity.

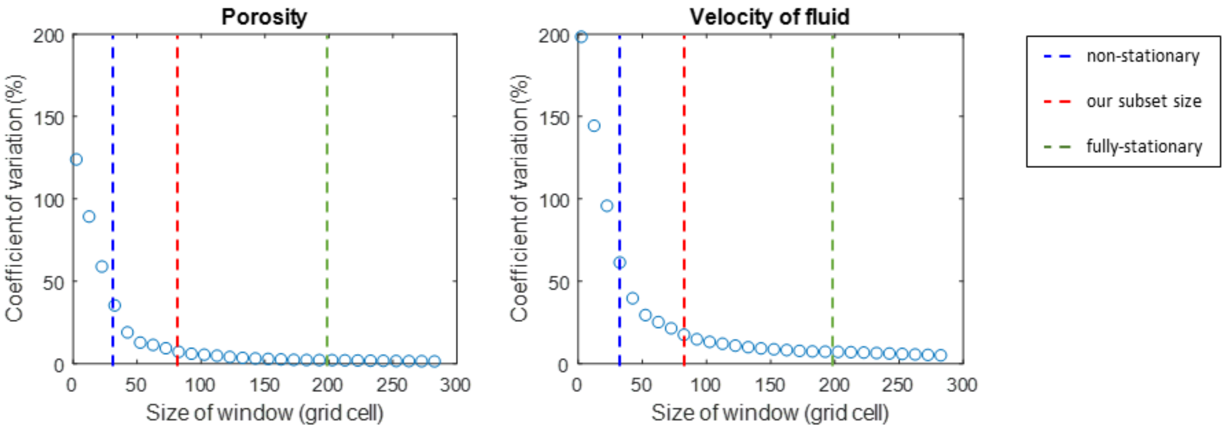

Figure 7: Evaluation of representative element volume with coefficient of variation (CV) for porosity and velocity. The variability of porosity and velocity decreases with the increase in the window size.

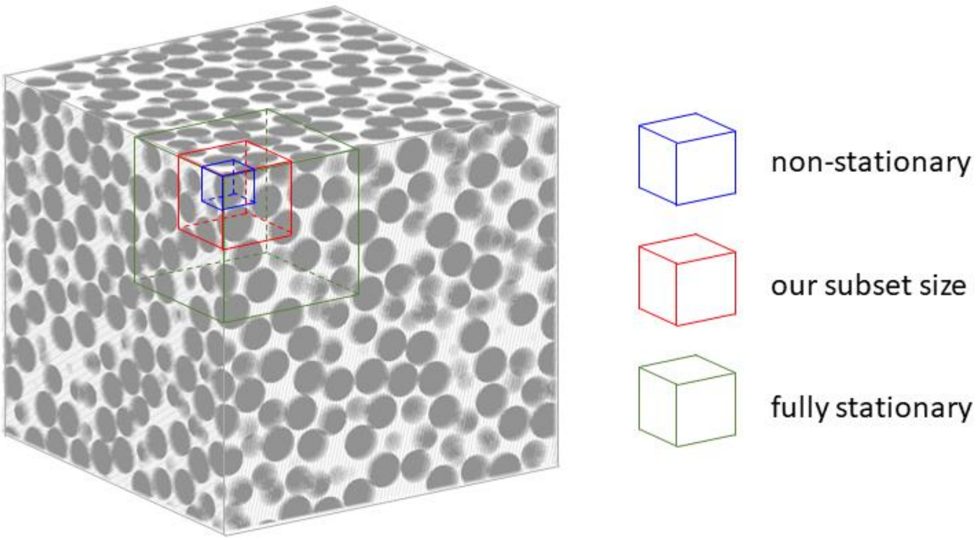

Figure 8: 3D visualization of the moving windows sizes of the stationarity analysis. Gray areas indicate grains whereas white portions represent pores. The blue cube represents a highly non-stationary window size (Figure 7), the red cube is our selected subset size (80^3), and the green cube shows a stationary window size for the pictured domain.

3.2 Model Training

We implemented the model using the Keras python library (Chollet & others, 2015) with TensorFlow (Abadi et al., 2015) as the backend. The model is optimized by minimizing the cost function (Equation 5) using Adam (Kingma & Ba, 2014) with a learning rate of 1e-4. We used four sphere packs (that present four subsequent grain dilations from the original sample, as highlighted in section 2.4.1), and subsampled them into 1080 80^3 cubes for training with a 20% random validation split (216 cubes). The model was trained with a mini-batch size of five, on a desktop with an NVIDIA Quadro M6000 GPU for 140 epochs. This model training process took twelve hours. The inputs and the outputs are transformed using the minmax transform constraining them from minus one to one. A comparison of the performance of three different model setups (training with the binary image only, training with the four proposed geometrical features, and training utilizing the features plus the custom loss function) is plotted in Figure 9. We observed a significant performance increase in the loss value when using the extracted features and the proposed loss.

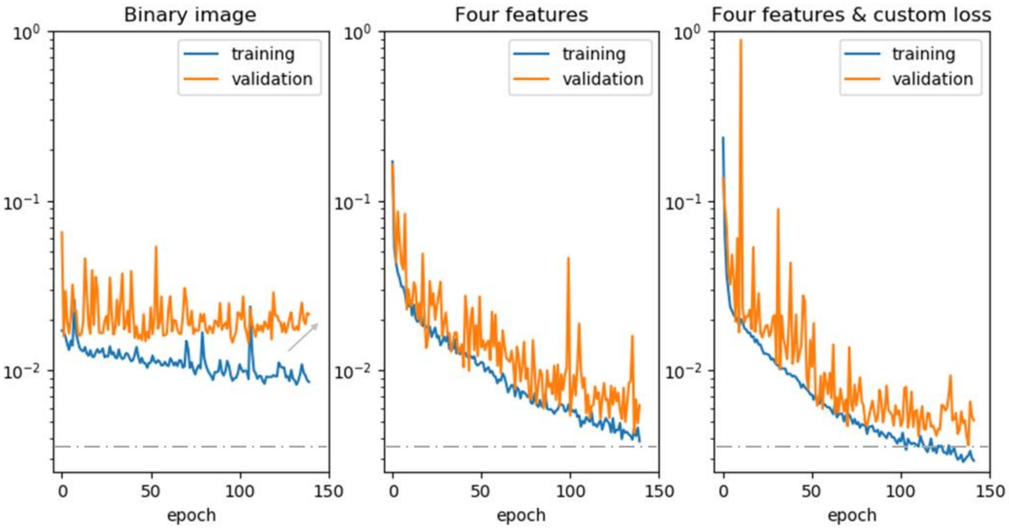

Figure 9: Training and validation loss values for three different cases. The left plot shows the training performed with the binary image only, the middle plot shows where the four features (Euclidean distance, MIS, and the two ToF) are used, and the right plot shows the application of the proposed custom loss function. By using the binary image as the only input, the model overfits (validation curve rising) and its minimum value remains high. By training using the custom loss function, the performance is improved (the slope of the loss is higher) and it reaches a smaller value that the rest (dashed line).

To assess the ability of the three trained models specified above to generalize the training data, we first tested the model using the original sphere-pack (unseen by the models, these were trained using the samples with the dilated grains only). The model trained with the binary input gave a relative error in permeability of several orders of magnitude. Whereas the one using the four input features returned and error 15 %, and finally the model trained using the features plus the custom loss gave a relative error of 13% when compared to the lattice-Boltzmann simulation. We carry out an extensive testing of the latter trained model in the sections below.

3.3 Model Testing

Using the model trained with the four dilated spherepacks (Section 3.2), we tested its capabilities (vs the Navier-Stokes equation solution approximated by the LBM) on domains of different size and complexity.

3.3.1 Fontainebleau sandstone dataset

The first test set was obtained via a simulation of processes that occur during sedimentary rock formation (i.e., sedimentation, compaction, diagenesis, and cementation) to obtain 3D volumes that resemble the Fontainebleau Sandstone formation in France (Berg, 2016). These images are 480^3 voxels, and vary from 8 to 26% porosity. We show a cross section through the middle of four of the samples in Figure 10.

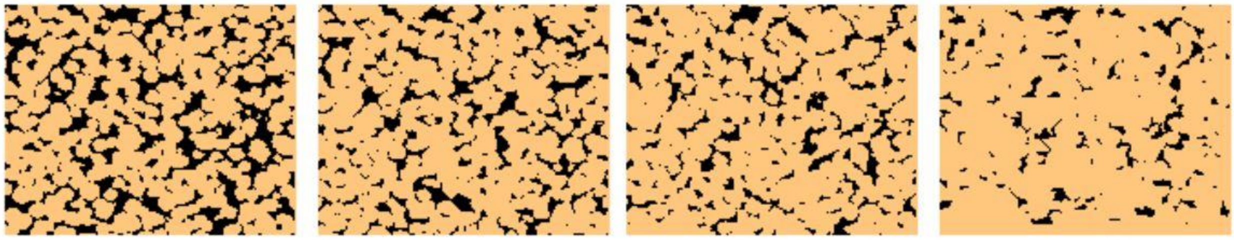

Figure 10: XY-Cross sections of the Fontainebleau sandstones of the test set. The pore space is shown in black. Some of the structures have been disconnected from the bulk during cementation and compaction, making this test set very different from the homogenous spherepacks pore space in our training set.

We present the results in Table 1. These are in very good agreement with the full-physics simulation (carried out to compare the performance of our model). To analyze the error more closely, we selected the worst performing sample (24% porosity sandstone) for further analysis. In Figure 11 we show a visual comparison (cross-section of the 3D volume orthogonal to the flow direction) of the lattice-Boltzmann solution with our model. It is visible that most of the relevant flow features are preserved. A comparison of the velocity histograms is shown in Figure 12. It is worth noting that the flow streamlines are not always continuous, and the 3D solution is not trained to satisfy mass balance (hence the relative error). Additional constraints can be added to honor this, but are out of scope for this work. In this work we are mostly interested in capturing the main flow characteristics (preferred paths and dead-ends) that impact permeability.

Porosity	Average relative error (voxel-wise) in the pore-space	True permeability [m^2]	Predicted permeability [m^2]	Relative error
8%	0.75%	8.76e-14	10.86e-14	24%
9.8	0.80%	2.35e-13	2.44e-13	4%
12.4%	1.04%	4.97e-13	5.34e-13	7.23%

15.2	1.75%	1.47e-12	1.45e-12	1.44%
17.5	2.25%	2.45e-12	2.61e-12	6.73%
24%	4.36%	7.76e-12	8.56e-12	10.32%

Table 1: Comparison of our model performance versus the LBM simulation on the Fontainebleau dataset. The average relative error refers to the error in the velocity parallel to the pressure gradient (z-coordinate). The average voxel-wise error shows a nonlinear relationship with the permeability relative error because the permeability is an average measure of the velocity considering the solid volume. Although the pixel-wise error increases with porosity, the permeability error (calculated using the mean velocity) shows no correlation with porosity. This is due to the fact that the highest errors are present near the grain boundaries (Figure 12), this values do not have a significant impact in permeability.

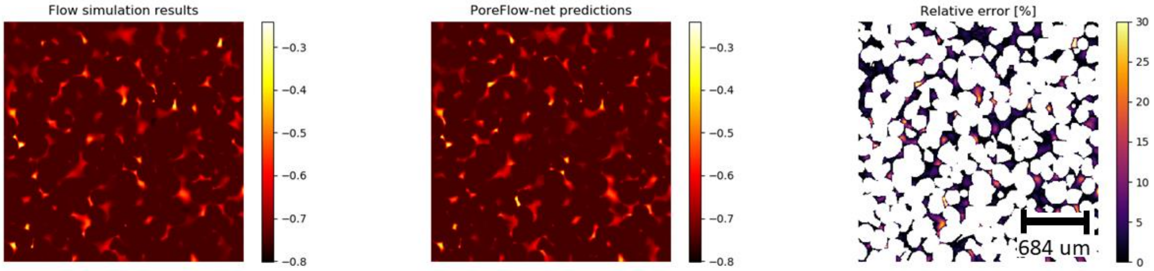

Figure 11: XY Cross-section of the fluid flow simulated velocity in the z-direction (left), PoreFlow-Net prediction (middle) and the relative error between these two (right) for the Fontainebleau sandstone with 24% porosity. The velocity is shown in a dimensionless scale going from one to minus one (minmax transform). The mean average error in the pore space is 4.36% as reported in Table 1. The highest errors (voxel-wise) are in the pore throats (which is also consistent in the velocity histogram in Figure 12).

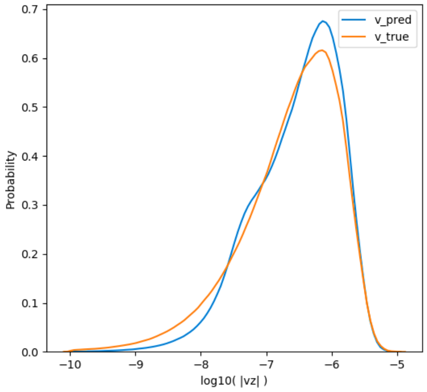

Figure 12: Histogram of true and predicted velocity (480^3 points in lattice units) in the z-direction at every point of the domain. The comparison shows excellent agreement at high velocity, and a

slight disagreement on the lower range. We hypothesize that since the training was performed with spherepacks, by having paths that are more tortuous, hosting a higher amount of dead end pores, and more solid surface area, the network does not perform as accurately. Since the highest orders of magnitude in velocity have a greater effect in permeability, there is good agreement in the permeability magnitude (Table 1).

3.3.2 Tests on different rock types

To further test our model, we predicted the flow field for different rock types available in Digital Rocks Portal (Prodanovic et al., 2015). We first created a sample similar (in shape) to the original training image by performing numerical grain erosion. This creates a sample of larger porosity where grain boundaries are not as restrictive to fluid flow (where the permeability is higher). This case is of interest in irrigation (Garnier et al., 1998). We further tested the original sphere pack (the one that was numerically dilated to generate our training set). Our model yield accurate predictions in these two samples, even when the porosities where larger (where velocities that are also orders of magnitude higher) than the training set. We then tested the trained model on two outcrop sandstones, a limestone, and artificially created multiscale microsand image. In these, the relative error was not higher than 28% (Yang et al., 2016) show that different fluid flow solvers will have a comparable discrepancy among them, even when the same geometry is provided).

We present our results in Table 2, we show the different 3D domains of the test set in Figure 14, and a cross section of the results is shown in Figure 15**.** These geometries have different pore shapes, and in cases of limestone and microsand they have much wider pore size distribution compared to the training set (Figure 13). They also have different absolute volume sizes. While two sandstones have similar absolute volume size (500 voxel on a side), the relative error for prediction is very different (1.06% and 27.30%) likely because they have different grain/pore distribution as well as different number of individual grains per side (which determines how well grain or pores are resolved). Note that our training set as well as the Fontainebleau sandstone test in previous section all have similar level of resolution and hence we saw a very good prediction for all cases in Table 1. Given that the training set was comparatively simple, we find the results in great agreement with the full-physics simulations.

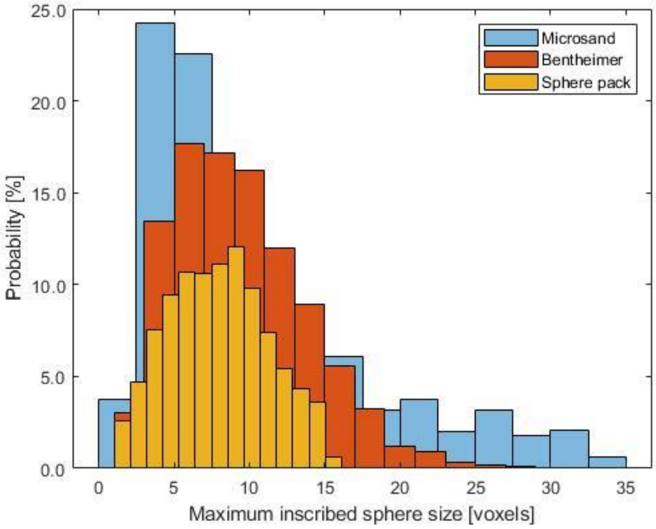

Figure 13: Maximum inscribed sphere distribution for three of our samples. In orange, the training set, where the distribution is Gaussian and relatively narrow. In red and blue, the MIS distribution for the Bentheimer sandstone and microsand respectively. These distributions have a lognormal shape, which is due to the more heterogeneous pore structures.

Sample	Size [voxels^3]	Resolution [m/voxel]	Porosity	True permeability [m^2]	Predicted permeability [m^2]	Relative error
Eroded sphere pack	500	5.7E-6	42%	8.86e-11	6.76e-11	23.53%
Sphere pack	500	5.7E-6	36%	5.26e-11	4.58e-11	12.96%
Estaillades limestone	650	3.3113E-6	11.8%	6.62e-13	6.99e-13	5.45%
Microsand (artificial multiscale sample)	500	3E-6	28.2%	5.64e-12	4.68e-12	17.01%
Castlegate sandstone	512	5.6E-6	20.5%	2.19e-12	2.17e-12	1.06%
Bentheimer sandstone	500	3.0035E-6	20.1%	3.77e-12	2.74e-12	27.30%

Table 2: Results of the additional test set.

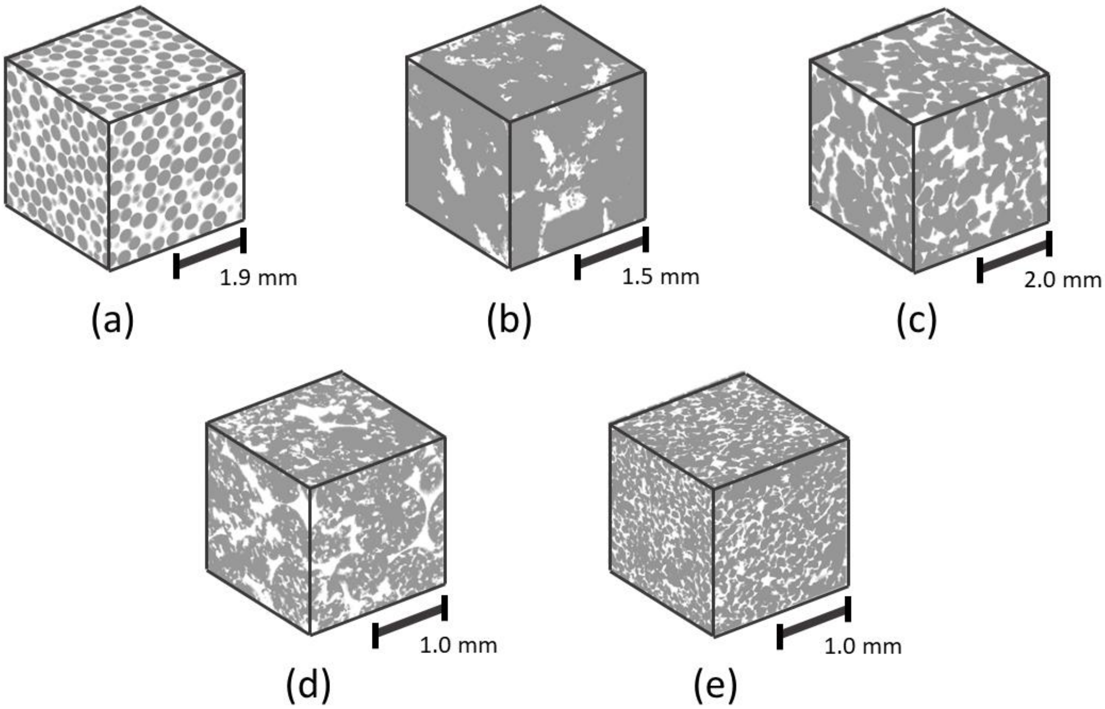

Figure 12: Additional test that includes: a) a slightly consolidated medium, b) Estillades limestone (Muljadi, 2015b), c) Castlegate sandstone (Sheppard & Prodanovic, 2015), d) multiscale microsand (Mohammadmoradi, 2017) and e) Bentheimer sandstone (Muljadi, 2015a), all of which are available at the Digital Rocks Portal.

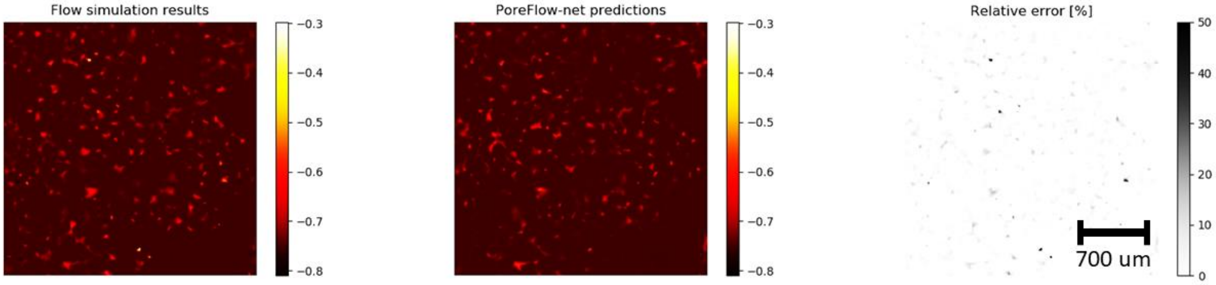

Figure 15: XY Cross-section of the simulated velocity (left), PoreFlow-Net prediction (middle) and the relative error between these two (right) for the Castlegate formation sandstone. The velocity is shown in a dimensionless scale going from one to minus one (minmax transform). The mean average relative error in the pore space only is 1% as reported in Table 2.

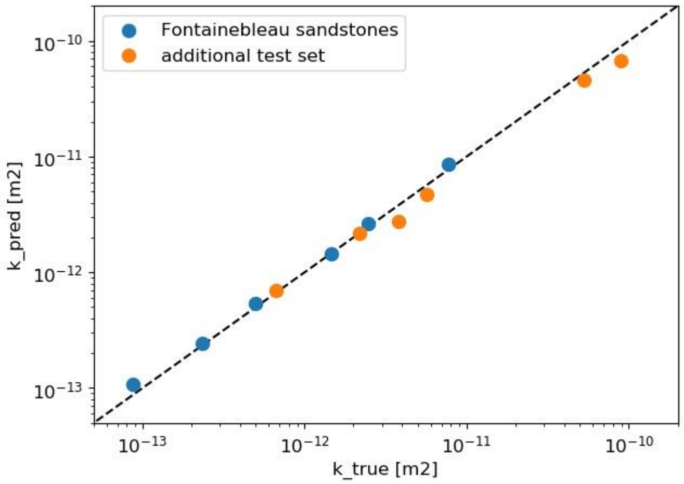

Figure 16: Permeability predictions vs true values. The PoreFlow-Net is able to predict a wide range of orders of magnitude.

4. Conclusions

We train a deep neural network architecture as a fast proxy to predict accurately the 3D physics-based fluid flow velocity fields within digital rock samples. The relationship between details of pore geometry and flow field (with its integral measure of permeability) is complex and not easily predicted based on the geometry statistics alone. Nevertheless, this fundamental relationship allows describing how fluids move through subsurface formations, and is the cornerstone of many research projects in environmental, civil, petroleum engineering as well as in geological sciences.

We demonstrated that our convolutional neural network generalizes the flow problem to predict flow velocity in rocks that host much more complex structures than the original training set. This is attributed to the capacity of the network to model the complicated relationships between pore shape and domain characteristics with the velocity field. The model performs well with rocks of varying types (different lithology), and of different grain distribution and porosities, where the permeability ranged several orders of magnitude (Figure 16). The PoreFlow-Net calculates fluid flow fields in less than a second on a typical desktop, compared with the standard simulation procedure, which takes hours to days in a supercomputer facility (depending on the hardware used as well as complexity of the digitized pore space geometry). Additionally, the model is a lightweight representation (around 25 Mb), whereas the full simulation results takes 20X the hard drive space. The model can be reused in any given geometry, while the simulation has to be run case-by-case. Future work should be focused on finding features that work with fractured domains and ultra-tight rocks.

This method provides a framework for different further applications such as component transport, relative permeability, rock-mechanics applications, formation factor, or resistivity. These models provide a straightforward way to assess important characteristics for improved subsurface management without running expensive physical models and could possibly be a path to data-based upscaling, given the proliferation of digital rock images as evidenced in the Digital Rocks Portal or online data available by different research groups (Blunt, 2015; Dorthe Wildenschild, 2006).

Reproducibility

The code will be publicly available on the author's repository (github.com/je-santos) and all the data used will be posted to Digital Rock Portal upon publication.

Acknowledgements

We would like to thank Risto Miikkulainen and Santiago Gonzalez from the UTCS Neural Networks Research Group for their valuable comments. We would also like to acknowledge Ying Yin and Wenhui Song for their feedback on the flow model. Additionally we would like to thank Renan Rojas, Manish Bhattarai, and Nicholas Lubbers for their suggestions towards the improvement of the neural network model. We gratefully recognize the Texas Advanced Computing Center for their high performance computing resources. M. Pyrcz, J. Santos, and H. Jo acknowledge support from DIRECT Industry Affiliates Program (IAP), and C. Landry and M. Prodanovic acknowledge support from Digital Rock Petrophysics IAP both of The University of Texas. Finally, we thank the four anonymous reviewers for their comments, which greatly improved this paper.

Appendix

A1. Calculating mask for custom loss function

We calculate the weight matrix (Figure 17) using the following pseudocode:

```
image_size = 500                    # length of the volume side
for i in range( 0, image_size ):  # loop along the z-coordinate
        porosity_z = np.sum(binary_im[:,:,i])/image_size^2  # calculate the porosity of the slice
        solid_mask[:,:,i] = (1/porosity_z)*solid_mask[:,:,i]     # multiply by a term that weights lower porosity sections (Figure 17)
    solid_mask[:,:,i][solid_mask[:,:,i]==0]=1  # replace the solids with a 1
        solid_mask[:,:,i] = solid_mask[:,:,i]/sum(solid_mask[:,:,i])*image_size^2 # normalize
```

Where the binary image is composed by a 3D matrix of zeros representing the solids and ones representing the space for fluid to flow.

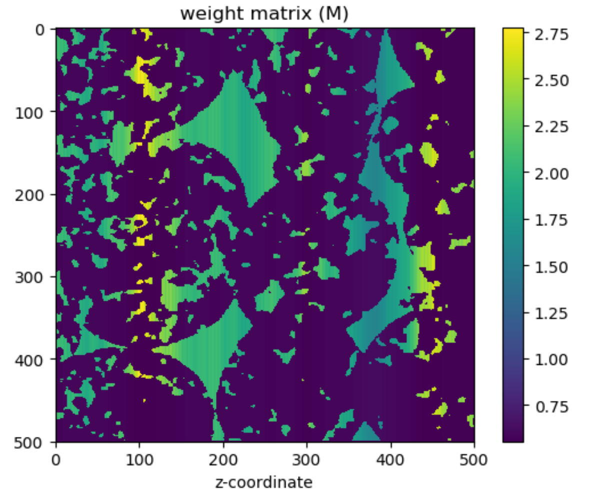

Figure 17: Cross section of the weight matrix (M). Areas with low porosity have a higher weight so that the network 'focuses' in those areas as well as in the bigger channels.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2015). Tensorflow: A system for large-scale machine learning. Retrieved from http://tensorflow.org/

Andrew, M. (2018). A quantified study of segmentation techniques on synthetic geological XRM and FIB-SEM images. *Computational Geosciences*, *22*(6), 1503–1512. https://doi.org/10.1007/s10596-018-9768-y

Bachmat, Y., & Bear, J. (1987). On the concept and size of a representative elementary volume (REV). In *Advances in transport phenomena in porous media* (pp. 3–20). Springer.

Bear, J., & Bachmat, Y. (1991). Introduction to modeling phenomena of transport in porous media. *Theory and Application on Transport Media*, *4*.

Berg, C. F. (2016). Fontainebleau 3D models. Digital Rocks Portal. https://doi.org/doi:10.17612/P75P4P

Bihani, A., Daigle, H., Santos, J. E., Landry, C., Prodanović, M., & Milliken, K. (2019). Insight into the Sealing Capacity of Mudrocks determined using a Digital Rock Physics Workflow. In *TACCSTER 2019 Proceedings*. Austin. https://doi.org/10.26153/tsw/6874

Blunt, M. J. (2015). Micro-CT images of sandstone and carbonate rocks. Retrieved from https://www.imperial.ac.uk/earth-science/research/research-groups/perm/research/pore-scale-modelling/micro-ct-images-and-networks/

Blunt, M. J. (2017). *Multiphase Flow in Permeable Media*. Cambridge: Cambridge University Press. https://doi.org/10.1017/9781316145098

Blunt, M. J., Bijeljic, B., Dong, H., Gharbi, O., Iglauer, S., Mostaghimi, P., … Pentland, C. (2013). Pore-scale imaging and modelling. *Advances in Water Resources*, *51*, 197–216. https://doi.org/10.1016/j.advwatres.2012.03.003

Carrillo, M., Que, U., González, J. A., & López, C. (2017). Recognition of an obstacle in a flow using artificial neural networks. *Physical Review E*, *96*(2), 1–10. https://doi.org/10.1103/PhysRevE.96.023306

Chen, Y., Li, Y., Valocchi, A. J., & Christensen, K. T. (2018). Lattice Boltzmann simulations of liquid CO 2 displacing water in a 2D heterogeneous micromodel at reservoir pressure conditions. *Journal of Contaminant Hydrology*, *212*. https://doi.org/10.1016/j.jconhyd.2017.09.005

Chollet, F., & others. (2015). Keras.

Dorthe Wildenschild. (2006). Multi-Phase Data Exchange. Retrieved from http://research.engr.oregonstate.edu/immiscibles/multi-phase-data-exchange

Finney, J. L., & Prodanovic, M. (2016). Finney Packing of Spheres. http://www.digitalrocksportal.org/projects/47. Digital Rocks Portal. https://doi.org/10.17612/P78G69

Frisch, U. (1991). Relation between the lattice Boltzmann equation and the Navier-stokes equations. *Physica D: Nonlinear Phenomena*, *47*(1–2), 231–232. https://doi.org/10.1016/0167-2789(91)90293-I

Garnier, P., Angulo-Jaramillo, R., DiCarlo, D. A., Bauters, T. W. J., Darnault, C. J. G., Steenhis, T. S., … Baveye, P. (1998). Dual-energy synchrotron X ray measurements of rapid soil density and water content changes in swelling soils during infiltration. *Water Resources Research*, *34*(11), 2837–2842. https://doi.org/10.1029/98WR02367

Gauthier, J. (2015). Conditional generative adversarial nets for convolutional face generation.

Guo, X., Li, W., & Iorio, F. (2016). Convolutional Neural Networks for Steady Flow Approximation. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 481–490. https://doi.org/10.1145/2939672.2939738

Hassouna, M., & Farag, A. (2007). MultiStencils Fast Marching Methods: A Highly Accurate Solution to the Eikonal Equation on Cartesian Domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*, 1563–1574. https://doi.org/10.1109/TPAMI.2007.1154

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016–Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hennigh, O. (2017). Lat-Net: Compressing Lattice Boltzmann Flow Simulations using Deep Neural Networks. Retrieved from http://arxiv.org/abs/1705.09036

Jenny, P., Lee, S. H., & Tchelepi, H. A. (2003). Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, *187*(1), 47–67.

Kamrava, S., Tahmasebi, P., & Sahimi, M. (2020). Linking Morphology of Porous Media to Their Macroscopic Permeability by Deep Learning. *Transport in Porous Media*, *131*(2), 427–448. https://doi.org/10.1007/s11242-019-01352-5

Kang, Q., Lichtner, P. C., & Zhang, D. (2007). An improved lattice Boltzmann model for multicomponent reactive transport in porous media at the pore scale. *Water Resources Research*, *43*(12), 1–12. https://doi.org/10.1029/2006WR005551

Karimpouli, S., & Tahmasebi, P. (2019a). Image-based velocity estimation of rock using Convolutional Neural Networks. *Neural Networks*, *111*, 89–97. https://doi.org/10.1016/j.neunet.2018.12.006

Karimpouli, S., & Tahmasebi, P. (2019b). Segmentation of digital rock images using deep convolutional autoencoder networks. *Computers and Geosciences*, *126*(February), 142–150. https://doi.org/10.1016/j.cageo.2019.02.003

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, *2017–Decem*, 972–981.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Liu, J., Wang, Y., & Song, R. (2017). A Pore Scale Flow Simulation of Reconstructed Model Based on the Micro Seepage Experiment. *Geofluids*, *2017*. https://doi.org/10.1155/2017/7459346

Mees, F., Swennen, R., Geet, M. Van, & Jacobs, P. (2003). Applications of X-ray computed tomography in the geosciences. *Geological Society, London, Special Publications*, *215*(1), 1–6. https://doi.org/10.1144/GSL.SP.2003.215.01.01

Mehmani, A., Verma, R., & Prodanović, M. (2020). Pore-scale modeling of carbonates. *Marine and Petroleum Geology*, *114*(July 2019), 104141. https://doi.org/10.1016/j.marpetgeo.2019.104141

Mohammadmoradi, P. (2017). A Multiscale Sandy Microstructure. Digital Rocks Portal. https://doi.org/doi:10.17612/P7PC7C

Mosser, L., Dubrule, O., & Blunt, M. J. (2018). Stochastic Reconstruction of an Oolitic Limestone by Generative Adversarial Networks. *Transport in Porous Media*, *125*(1), 1–23. https://doi.org/10.1007/s11242-018-1039-9

Muljadi, B. P. (2015a). Bentheimer Sandstone. Digital Rocks Portal. https://doi.org/doi:10.17612/P77P49

Muljadi, B. P. (2015b). Estaillades Carbonate. Digital Rocks Portal. https://doi.org/doi:10.17612/P73W2C

Pan, C., Hilpert, M., & Miller, C. T. (2004). Lattice-Boltzmann simulation of two-phase flow in porous media. *Water Resources Research*, *40*(1), 1–14. https://doi.org/10.1029/2003WR002120

Pan, C., Luo, L.-S., & Miller, C. T. (2006). An evaluation of lattice Boltzmann schemes for porous medium flow simulation. *Computers & Fluids*, *35*(8–9), 898–909. https://doi.org/10.1016/j.compfluid.2005.03.008

Pascanu, R., Mikolov, T., & Bengio, Y. (2012). Understanding the exploding gradient problem. *CoRR, Abs/1211.5063*, *2*.

Prodanovic, M., Esteva, M., Hanlon, M., & others. (2015). Digital rocks portal.

Pyrcz, M. J., & Deutsch, C. V. (2014). *Geostatistical Reservoir Modeling*. Oxford University Press.

Raeini, A. Q., Blunt, M. J., & Bijeljic, B. (2014). Direct simulations of two-phase flow on micro-CT images of porous media and upscaling of pore-scale forces. *Advances in Water Resources*, *74*, 116–126. https://doi.org/10.1016/j.advwatres.2014.08.012

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9351*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Santos, J. E., Prodanović, M., Landry, C. J., & Jo, H. (2018). Determining the Impact of Mineralogy Composition for Multiphase Flow Through Hydraulically Induced Fractures. In *Proceedings of the 6th Unconventional Resources Technology Conference* (pp. 1–15). Tulsa, OK, USA: American Association of Petroleum Geologists. https://doi.org/10.15530/urtec-2018-2902986

Santos, J. E., Prodanović, M., & Pyrcz, M. (2018). Characterizing effective flow units in a multiscale porous medium. In *American Geophysical Union Fall Meeting Abstracts*. https://doi.org/10.1002/essoar.10502121.1

Sheppard, A., & Prodanovic, M. (2015). Network Generation Comparison Forum. Digital Rocks Portal. https://doi.org/doi:10.17612/P7059V

Sudakov, O., Burnaev, E., & Koroteev, D. (2018). Driving Digital Rock towards Machine Learning: predicting permeability with Gradient Boosting and Deep Neural Networks, 1–22. Retrieved from http://arxiv.org/abs/1803.00758

Sukop, M. C., & Thorne, D. T. (2007). *Lattice Boltzmann Modeling*. Springer.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tartakovsky, A., & Meakin, P. (2005). Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *72*(2), 1–9. https://doi.org/10.1103/PhysRevE.72.026301

Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., … Richardson, M. (2016). Do deep convolutional nets really need to be deep and convolutional? *ArXiv Preprint ArXiv:1603.05691*.

White, J. A., Borja, R. I., & Fredrich, J. T. (2006). Calculating the effective permeability of sandstone with multiscale lattice Boltzmann/finite element simulations. *Acta Geotechnica*, *1*(4), 195–209.

Wu, J., Yin, X., & Xiao, H. (2018). Seeing permeability from images: fast prediction with convolutional neural networks. *Science Bulletin*, *63*(18), 1215–1222. https://doi.org/10.1016/j.scib.2018.08.006

Xu, P., & Yu, B. (2008). Developing a new form of permeability and Kozeny-Carman constant for homogeneous porous media by means of fractal geometry. *Advances in Water Resources*, *31*(1), 74–81. https://doi.org/10.1016/j.advwatres.2007.06.003

Yang, X., Mehmani, Y., Perkins, W. A., Pasquali, A., Schönherr, M., Kim, K., … Scheibe, T. D. (2016). Intercomparison of 3D pore-scale flow and solute transport simulation methods. *Advances in Water Resources*, *95*, 176–189. https://doi.org/10.1016/j.advwatres.2015.09.015

Zhang, Z., Liu, Q., & Wang, Y. (2018). Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, *15*(5), 749–753. https://doi.org/10.1109/LGRS.2018.2802944