A prospect for computing in porous materials research: very large fluid flow simulations

Keijo Mattila^{a,b,*}, Tuomas Puurtinen^a, Jari Hyväluoma^c, Rodrigo Surmas^d,
Markko Mylly^a, Tuomas Turpeinen^a, Fredrik Robertsén^e, Jan Westerholm^e,
Jussi Timonen^a

^a*Department of Physics and Nanoscience Center, University of Jyväskylä, P.O. Box 35
(YFL), FI-40014 University of Jyväskylä, Finland*

^b*Department of Physics, Tampere University of Technology, P.O. Box 692, FI-33101
Tampere, Finland*

^c*Natural Resources Institute Finland (Luke), FI-31600 Jokioinen, Finland*

^d*CENPES, Petrobras, 21941-915 Rio de Janeiro, Brazil*

^e*Faculty of Science and Engineering, Åbo Akademi University, Joukahainengatan 3-5,
FI-20520 Åbo, Finland*

Abstract

Properties of porous materials, abundant both in nature and industry, have broad influences on societies via, e.g., oil recovery, erosion, and propagation of pollutants. The internal structure of many porous materials involves multiple scales which hinders research on the relation between structure and transport properties: typically laboratory experiments cannot distinguish contributions from individual scales while computer simulations cannot capture multiple scales due to limited capabilities. Thus the question arises how large domain sizes can in fact be simulated with modern computers. This question is here addressed using a realistic test case; it is demonstrated that current computing capabilities allow the direct pore-scale simulation of fluid flow in porous materials using system sizes far beyond what has been previously reported. The achieved system sizes allow the closing of some particular scale gaps in, e.g., soil and petroleum rock research. Specifically, a full steady-state fluid flow simulation in a porous material, represented with an unprecedented resolution for the given sample size, is reported: the simulation is executed on a CPU-based supercomputer and the 3D geometry involves 16384^3 lattice cells (around 590 billion of them are pore sites). Using half of this sample in a benchmark simulation on a GPU-based system, a sustained computational performance of 1.77 PFLOPS is observed. These advances expose new opportunities in porous materials research. The implementation techniques here utilized are standard except for the tailored high-performance data layouts as well as the indirect addressing scheme with a low memory overhead and the truly asynchronous data communication scheme in the case of CPU and GPU code versions, respectively.

*Corresponding author

Email address: keijo.mattila@jyu.fi (Keijo Mattila)

Keywords: porous material, permeability, fluid flow simulation, lattice Boltzmann method, petascale computing, GPU

1. Introduction

The persistent progress in computational software and hardware offers ever-more powerful tools for research and development. The theoretical peak performances of the top supercomputers are currently measured in tens of petaflops [1], and there are already several scientific softwares which reach a sustained performance of even tens of petaflops (see e.g. Refs. [2, 3, 4, 5, 6, 7, 8, 9]). However, when it comes to a specific research field, the relevance of this immense computational power to solving outstanding research problems is not immediately clear. We must ask ourselves what kind of research tasks can be tackled by fully harnessing the modern computational resources available, or how we can exploit the resources in a meaningful way, and which in fact the ambitious and realistic research questions from purely a computational point of view are. Here we will consider these aspects in connection with soil research and reservoir evaluation.

1.1. The Computational Challenge

In structured heterogeneous soils, water and solutes can be largely transported through macropores, and they can thereby bypass most of the pore matrix. This kind of rapid preferential flow is an important phenomenon in agricultural soils due to its implications on the fast movement of contaminants through the soil profile both in their dissolved and colloid-bound forms [10]. Since macropores are relatively sparsely distributed in soils, quite large soil samples would be necessary in imaging so as to capture a representative volume element (RVE). In order to have a representative sample, typical soil cores used in experiments are ca 20 cm in diameter. At the same time, pores larger than 100 μm have been found to participate in the preferential flows.

To reliably include pores of this size in simulations of fluid flow through soil samples using, e.g., the lattice-Boltzmann method [11, 12], an image resolution of about 20 μm would be needed. So far the largest reported lattice-Boltzmann simulations in 3D soil images, obtained by X-ray microtomography, have used samples with a voxel count of about 1000^3 (see e.g. Refs. [13, 14]). A resolution of 20 μm would have thus allowed the simulation of soil samples with an edge length of 2 cm, which is about ten times bigger than the required size for RVEs. Thus, an order of magnitude increase in the size of the simulation domain would lead to important benefits in this research problem. Obviously such a simulation would still neglect a significant part of the small pores important in many transport processes, but the advance in the state-of-the-art is clear in terms of characterizing the soil macropore networks and studying the preferential and non-equilibrium flows.

Characteristic length scales of an oil reservoir rock, in turn, can vary from kilometers to nanometers depending on the extension of the reservoir and on the

Figure 1: An example of carbonate rock: a sample of dolomite. Images were obtained with X-ray microtomography (*Xradia Ultra*TM, resolution 65 nm). The color-rendered three-dimensional (3D) image on the left shows pore structures and three distinct mineral phases. The planar image on the right shows pore structures in greater detail.

size of the smallest rock structures that can contain oil. In case of a conventional siliciclastic reservoir rock, for example, the mean diameter of pore throats is generally more than $2\ \mu\text{m}$, while in a shale rock these throats can be as small as 5 nm [15]. In carbonate rock, on the other hand, multiple porosity and permeability systems can coexist in a reservoir, both due to the deposition characteristics of the rock and a large variety in the diagenesis processes [16, 17]. Diagenesis, i.e., the chemical and physical changes that occur in a reservoir after the deposition of the rock, can strongly modify its properties by creating in the rock different kinds of microporosity or large structures like vugs and caverns. A visualization by very high resolution X-ray microtomography of carbonate rock is shown in Fig. 1.

Several tools are used to characterize reservoirs: the acquisition of seismic data can cover an area of many square kilometers with a resolution of a few tens of meters, various logging tools are used after the drilling process to characterize the rock at or near the surface of a well with a resolution limited to a few tens of centimeters, and, finally, the rock obtained during the drilling process (i.e., drilling cores, side-wall samples, and/or cuttings) can be investigated in greater detail. The dilemma of course is that, in general, rock properties can be measured with good accuracy from small samples, but, at the same time, the correlation of the observed properties with those of the rest of the reservoir is degraded. Hence, the most important task in everyday reservoir evaluation is to properly up-scale the properties measured at a small scale.

The most commonly used sample for direct, or indirect, laboratory measurements of rock properties before any up-scaling processes is a plug. Plugs are extracted from core samples recovered during drilling, and typically they have a diameter of 3.8 cm and are 5–10 cm long. In the current state-of-the-art digital-rock physics [18, 19, 20], samples of conventional reservoirs with the average

pore throat of the order of $2\text{ }\mu\text{m}$, which thus determines the maximum voxel size, are simulated using a mesh of 2000^3 voxels corresponding to an edge length of 4 mm in the sample. Ten times larger simulation domains are hence required to reach sizes which correspond to the diameter of a plug. Such an advance in the computational capacity would not only increase the representativeness of the simulated properties, but would also help to improve their up-scaling from the plug scale to a whole-core scale.

1.2. A Response to the Challenge: Aims and Means

To summarize, porous materials with complex internal structures, typically involving multiple scales, present serious challenges to computational materials research. In the case of soil research, very large simulation domains are called for in order to reliably capture transport properties of RVEs. In reservoir evaluation the general dilemma is similar: rock properties can be measured accurately for small samples, but then the correlation of the observed properties with those of the rest of the reservoir is compromised. Thus, a fundamental task in everyday reservoir evaluation is to properly up-scale the properties measured at a small scale.

An increase in the size of pore-scale flow simulation domains by an order of magnitude would lead to significant progress both in soil and reservoir rock research. First of all, such *ab initio* simulations can improve our understanding of the fundamental relation between structure and transport properties in heterogeneous materials. Secondly, this progress would benefit more complex multiscale-modelling approaches where the larger scale continuum models require input from the pore scale simulations [21, 22]. For example, the so-called heterogeneous multiscale methods use a general macroscopic model at the system level while the missing constitutive relations and model parameters are locally obtained by solving a more detailed microscopic model [23, 24]. At large enough scales, however, the low resolution and the lack of available data often necessitate rough descriptions of governing processes and parametrizations based on a simplified picture of the pore structure. Therefore, if pore scale modelling would be able to capture the heterogeneity of the porous medium up to the size of RVE, the utilization of advanced multiscale-modelling techniques would become more viable and reliable tools in solving practical problems related to multiscale porous materials including up-scaling transport properties from, e.g., the plug scale to a whole-core scale.

Here we demonstrate that current computing capabilities already allow a direct pore scale simulation of transport phenomena in porous materials using system sizes far beyond what has previously been reported in the literature. The achieved system sizes readily close the particular scale gaps discussed above. In this demonstration, we simulate fluid flow through a very large sandstone sample using the lattice-Boltzmann method. The simulated flow field provides the average flow velocity through the sample which, in turn, allows determination of the sample permeability. In order to better demonstrate the current computing capabilities in porous materials research, we execute simulations with two separate implementations, i.e. with CPU and GPU code versions. The implementation

techniques utilized are standard except for the tailored high-performance data layouts as well as the indirect addressing scheme with a low memory overhead and the truly asynchronous data communication scheme in the case of CPU and GPU code versions, respectively. In our case study we utilize synthetic X-ray tomography images representing microstructure of Fontainebleau sandstone.

We begin by presenting the lattice Boltzmann method together with technical details concerning data layouts and memory addressing schemes in Section 2. Section 3 covers in detail the properties of the porous samples of Fontainebleau sandstone. Section 4 explains the main weak and strong scaling results from simulations on CPU- and GPU-systems, and permeability is used as an example where larger scale computations give qualitatively and quantitatively different results. Results from the fluid flow simulations are explained in Section 4. A general discussion is given in Section 5, and the conclusions are presented in Section 6.

2. Lattice-Boltzmann Method

In this work we will simulate fluid flow through a large porous sample using the lattice-Boltzmann method (LBM) which has emerged, in particular, as a promising alternative for describing complex fluid flows. For example, it has been applied to turbulent flows [25], systems of colloidal particles suspended in a multicomponent fluid [26], and to cardiovascular flows [27]. The first publications on fluid-flow simulations in porous media with LBM date back a quarter of a century, all the way to the lattice-gas automata preceding LBM [28, 29, 30].

The single-particle density distribution function $f_i(\mathbf{r}, t)$, the primary variable in LBM, describes the probability of finding a particle from site $\mathbf{r}$ at instant t traveling with velocity $\mathbf{c}_i$. The distributions are specified only for a finite set of q microscopic velocities. The time evolution of the distributions is governed by the lattice-Boltzmann equation (LBE). Here the standard LBE

$$f_i(\mathbf{r} + \Delta t \mathbf{c}_i, t + \Delta t) = f_i(\mathbf{r}, t) + \Delta t \Omega_i(\mathbf{r}, t) + \Delta t \Gamma_i(\mathbf{r}, t), \quad i = 0, \dots, q-1,$$

where Δt is the discrete time step, Ω_i the collision operator and Γ_i the forcing term, is implemented together with the D3Q19 velocity set [31]. Moreover, the collision operator Ω_i is realized using the two-relaxation-time scheme [32]:

$$\Omega_i = -\lambda_e f_i^{e, neq} - \lambda_o f_i^{o, neq}, \quad f_i^{e, neq} = \frac{1}{2} (f_i^{neq} + f_{-i}^{neq}), \quad f_i^{o, neq} = \frac{1}{2} (f_i^{neq} - f_{-i}^{neq}),$$

where $\mathbf{c}_{-i} = -\mathbf{c}_i$, $f_i^{neq} = f_i - f_i^{eq}$, and the usual second-order isothermal equilibrium function is used,

$$f_i^{eq} = w_i \rho \left[1 + \frac{c_{i\alpha}}{c_T^2} u_\alpha + \frac{1}{2c_T^4} (c_{i\alpha} c_{i\beta} - c_T^2 \delta_{\alpha\beta}) u_\alpha u_\beta \right];$$

w_i denote the velocity-set dependent weight coefficients, $\delta_{\alpha\beta}$ is the Kronecker delta, and the Einstein summation convention is implied with repeated indices.

The thermal velocity $c_T = c_r/a_s$, as well as the speed of sound for isothermal flows, depends on the reference velocity, $c_r = \Delta r/\Delta t$, where Δr is the lattice spacing, and a_s is the velocity-set dependent scaling factor [31, 33]. A relaxation parameter is the inverse of a relaxation time, e.g. $\lambda_e = 1/\tau_e$; for odd moments the relaxation parameter is here assigned according to the so-called magic formula $\lambda_o = 8(2 - \lambda_e)(8 - \lambda_e)$ [32]. The kinematic viscosity is defined by the relaxation time for even moments: $\nu = c_T^2(\tau_e - \Delta t/2)$.

A fluid flow over the porous sample is enforced with an effective pressure gradient, i.e. with a gravity-like external acceleration. The forcing term Γ_i is responsible for the acceleration and it is implemented with a first-order linear expression $\Gamma_i = (1 - \Delta t \lambda_o/2) \rho w_i c_{i\alpha} g_\alpha / c_T^2$ (no summation over i), where $\rho(\mathbf{r}, t) = \sum_i f_i(\mathbf{r}, t)$ is the local density and $\mathbf{g}$ the constant external acceleration. Finally, the local hydrodynamic velocity $\mathbf{u}$ used in the equilibrium function, and extracted from the simulations, is defined according to the expression

$$\rho \mathbf{u}(\mathbf{r}, t) = \sum_i \mathbf{c}_i f_i(\mathbf{r}, t) + \frac{\Delta t}{2} \mathbf{g},$$

which fundamentally emerges from the second-order trapezoidal-rule treatment of the forcing term.

2.1. Boundary conditions

The no-slip condition at walls is implemented with the common halfway-bounceback scheme [34]. The external acceleration driving the flow is imposed in the positive z -direction. Periodic boundary conditions are utilized in the x - and y -directions: the fact that the flow geometry is not periodic in these directions is ignored. In order to conveniently implement outer domain boundary conditions the computational domain is surrounded with a halo layer in our CPU version of the code. The thickness of the halo layer is one lattice site. However, in our GPU code version halo layers are omitted due to computational efficiency reasons (aligned memory accesses, see Ref. [9] for more details about our GPU implementation). We emphasize that exactly the same boundary conditions are implemented in the CPU and GPU code versions; the exclusion of halo layers affects the implementation details, but not the boundary conditions enforced, instead velocity distributions to be communicated are temporarily stored in memory locations for velocity distributions that will receive communicated values.

At the inlet and outlet (the xy -cross sections at the domain boundaries perpendicular to the acceleration) we adopt a strategy where the unknown distributions are decomposed into equilibrium and non-equilibrium parts which are then determined separately by a combination of Dirichlet and Neumann boundary conditions. First, for the equilibrium part of the unknown distributions, we use the constant reference density $\rho_0 \equiv 1$ (given in dimensionless lattice units) and $u_x = u_y = 0$ (we assume a unidirectional flow at the inlet and outlet). The remaining velocity component in the main flow direction, u_z , and the non-equilibrium part of the unknown distributions are determined using a Neumann

boundary condition: we require that the gradient of these variables vanishes in the z -direction. We enforce the Neumann condition with a first-order accurate approximation by simply copying (in the z -direction, not along the characteristics) the known values from the inlet and outlet for the incoming, unknown distributions. These inlet and outlet boundary conditions do not guarantee mass conservation in the system.

2.2. Implementation aspects

The above described lattice-Boltzmann scheme is implemented, both the CPU and GPU code versions, following the AA-pattern algorithm [35]. This algorithm has three major features:

1. during each time step, the distributions can be updated by traversing the lattice sites in an arbitrary order,
2. it is a so-called fused implementation where the relaxation and the propagation steps are executed together for each site, not e.g. by traversing the lattice twice, and
3. only one floating-point value (we use double precision) needs to be allocated per distribution function.

The CPU and GPU code versions are both implemented in C/C++. Furthermore, our GPU code is implemented using double precision floating-point computing and the CUDA API. Note that in our previous work [9], the GPU implementation was based on the Two-lattice algorithm [36] which has the same, attractive main features as the AA-pattern algorithm except that it requires the allocation of two floating-point values per distribution function and thus consumes more memory.

A peculiar feature of the AA-pattern algorithm is that, during an update procedure for a given lattice site and time step, it utilizes the same memory addresses for reading and writing the distribution values. However, these memory addresses are different between even and odd and time steps. There are, at least, two approaches for dealing with this aspect: to write separate update procedures for even and odd times steps, or to write a single update procedure together with an indexing function which acknowledges this feature. Our GPU implementation is based on the former and our CPU implementation follows the latter approach.

2.2.1. Data layout

The distribution functions are stored in the main memory using a non-standard data layout: the collision and stream optimized data layouts are considered standard [37]. Here we instead utilize a kind of compromise between the above two: the distribution functions are first gathered, locally, into several groups according to the microscopic velocity vectors they are associated with. Then a subgroup from each lattice site is collected and stored consecutively into the memory; this is repeated for each kind of subgroup specified. Such an approach has previously been considered e.g. in Refs. [38, 39]. Here we specify

Figure 2: The chosen data layout for the distribution functions related to the D3Q19 velocity set. First three groups are specified according to the z -components of the microscopic velocity vectors (negative, zero, or positive). Then a given group from each lattice site is collected and these groups are stored consecutively in memory.

three subgroups according to the z -component of the microscopic velocity vectors (negative, zero, or positive). This grouping reflects the natural numbering of lattice sites we have chosen: the numbering advances fastest in the x -direction and slowest in the z -direction. The numbering of the lattice sites provides a map from a 3D array into a 1D array. Figure 2 illustrates the adopted data layout.

The GPU code uses basically the same data layout as in the CPU code. However, the layout was slightly modified to suit data access requirements of the GPU: data on the GPU should if possible to be accessed in a coalesced fashion, i.e. in segments of 128 bytes. Therefore, instead of grouping individual distributions from a lattice site, distributions associated with a given microscopic velocity vector are first gathered into blocks of 16 values (from a continuous segment of lattice sites) and then these blocks are grouped in the way described above.

2.2.2. Indirect memory addressing

Furthermore, an indirect memory addressing scheme is adopted. To begin with, all lattice sites belonging to the pore phase are enumerated in the pre-processing stage (using the order defined by the natural numbering mentioned above). The indirect addressing scheme then relies on an additional array storing, for every pore lattice site, the memory addresses of the distribution values propagating from the neighboring pore sites to the given site. This array is assembled by the pre-processor. Here the standard halfway-bounceback operations are resolved already during the pre-processing and they are incorporated

into the stored memory addresses. This approach eliminates all conditional statements from the update procedure for distribution functions, i.e. queries whether a site, or any of its neighbors, belongs to the pore or solid phase are not necessary during the evolution of the distributions.

In practice, we allocate nineteen 32- or 64-bit integers for memory addresses per pore site, depending on the size of the local memory available. These addresses are stored in the collision optimized fashion. Furthermore, we store the index coordinates for each pore site: three 16-bit integers per site. After the pre-processing stage, no data is stored for the lattice sites belonging to the solid phase. The above described scheme is here referred to as the simple or common indirect addressing scheme.

In systems where 32-bit integers are not enough for memory addressing, due to a very large local memory available, instead of using 64-bit integers we here opt to continue with 32-bit integers but now adopting a more complex indirect addressing scheme. This choice is made because our current applications are limited by the total memory available rather than the computational performance. The complex scheme also relies on an additional array, but now this array stores the *relative pore enumeration numbers* of the neighboring sites using 32-bit integers. That is, the absolute pore enumeration numbers of the neighbors can be computed based on this information. Using the computed absolute enumeration numbers, we then further compute, on-the-fly, the memory addresses of the distribution values propagating from the neighboring pore sites to the given site. Enumeration numbers and memory address are both computed using 64-bit integer arithmetics.

A final hurdle remains: the bounceback operations must be executed explicitly in this complex indirect addressing scheme. In other words, the memory access in this approach necessarily involves conditional statements. A compact description of the complex indirect addressing scheme is presented in Algorithm 1. The conditional statements as well as the additional index computations will inflict a penalty on the computational performance. Some penalty figures from benchmark simulations are presented in Section 4.1.

Furthermore, the above complex indirect addressing concerns only our CPU code version as the local memories available in current GPUs are small enough for common indirect addressing with 32-bit integers. We emphasize that a particular CPU code version here utilized uses exclusively one of the three addressing schemes discussed above (the common scheme with 32- or 64-bit integers, or the complex scheme), i.e. no dynamical switching between addressing schemes takes place during code execution.

2.3. Parallel computing

Our CPU code version is based on a hybrid OpenMP/MPI implementation. A target system for such an implementation has a set of interconnected shared memory computing nodes: this is a common configuration on many modern supercomputers. One computing node typically has at least two CPUs, and each CPU includes several computing elements or cores. The strategy then is to assign one MPI process per node, or CPU, and at least one thread per core. In

Algorithm 1 Memory access for distributions f_i propagating to pore lattice site n_f : complex 32-bit indirect addressing scheme in the case of large memories.

Comment: *Relative enumeration numbers of the neighboring pore sites are stored in the array NI (32-bit integers).*

Comment: *Memory addresses m_a for f_i are computed using purely 64-bit integer arithmetics.*

```

for all  $i$  do
   $nn_f = n_f + \text{NI}[n_f][-i]$   //  $-i$  refers to the direction opposite to  $i$ 
  if  $nn_f \neq n_f$  then
     $m_a = \text{MI}(nn_f, i)$       // MI is an index function (or a macro)
  else
     $m_a = \text{MI}(n_f, -i)$       // standard halfway bounceback
  end if
   $f_i = \text{FI}[m_a]$            // array FI stores the distribution values
end for

```

Comment: *For comparison, in the common indirect addressing the array NI stores simply the addresses m_a (either 32- or 64-bit integers).*

The memory access is then straightforward: $f_i = \text{FI}[\text{NI}[n_f][i]]$ and the bounceback scheme is incorporated into the stored m_a .

comparison to a pure MPI implementation the target is to have larger computational subdomains per MPI process. This is desirable because it will improve the ratio between computation and communication – critical for achieving good parallel efficiency. Note that the computation and the communication scale with the volume and surface of the subdomain, respectively.

In the CPU and GPU implementations, simple cartesian and recursive bisection domain decompositions, respectively, are used to assign subdomains for the MPI processes (or the computational nodes). In the case of simple indirect addressing either with 32- or 64-bit integers (see the previous section), the workload balance between the threads of a MPI process or CUDA kernel was measured to be close to ideal. On the other hand, our implementation of the complex 32-bit addressing scheme, according to Algorithm 1, is optimized in such a way that the bounceback branch of the conditional statement involves slightly less index computations than the other branch. Hence, in this case, the workload balance between threads is ideal only with respect to the floating-point computation. Here the relevance of variations in the index computation on the total workload of a thread, including floating-point operations, is not considered further.

In an attempt to overlap communication with computation, the distribution

functions of a subdomain are updated in two steps (this is a standard technique, see e.g. Refs. [36, 40]). First, in each subdomain, the sites of an edge layer are updated. Then non-blocking MPI-routines are called in order to initiate the exchange of data between subdomains. This is immediately followed by an update of the interior sites. Finally, MPI-routines to complete the data exchange are executed. In our CPU code version, we define the edge width as 10% of the subdomain width: roughly speaking, the subdomain sites are divided equally into edge and interior regions. In our GPU code version, the edge width is defined so as to align memory accesses. Furthermore, in the GPU version, the MPI data communication between CPUs can be overlapped with the computation on the GPU side (truly asynchronous data communication), see Ref. [9] for more details about our parallel GPU implementation.

Although not considered in our performance results presented in Section 4, the role of file I/O is pronounced in large scale simulations involving parallel processing. This is acknowledged in our GPU implementation which utilizes parallel MPI-I/O over a Lustre file system [9]. Our CPU implementations, on the other hand, operate with a simple serial output mode and non-blocking, concurrent read for the input.

3. Porous Material

We utilize synthetic X-ray tomography images representing the microstructure of Fontainebleau sandstone: these images are freely available and they are world's largest 3D images of a porous material [41]. The synthetic images are based on a continuum model, which has been geometrically calibrated against true tomographic images of Fontainebleau sandstone [42]. This particular continuum model involved around one million polyhedrons, representing quartz grains, deposited in a manner that mimicked a real sedimentation and cementation process. The average grain size in a Fontainebleau sandstone is about 200–250 μm [43, 44]. There are nine synthetic X-ray images available, each of which is a discrete representation of the continuum model for a given resolution. The sandstone sample, i.e., the continuum model, has a side length of 1.5 cm, while the relative void space of the sample is about 13%. With the best resolution, around 0.46 μm , the largest image has 32768^3 voxels and, with one byte per voxel, requires 32 TB of storage space – each image voxel is given as a gray-scale value. Figure 3 shows individual grains which are identified and color-coded in a small volume by an analysis software.

The modeled sandstone is not really a multi-scale porous material: it is rather homogeneous and involves a single scale in its pore space. However, from a computational point of view, it serves the purpose of demonstrating the power of contemporary hardware solutions and software techniques in porous material research. It is also noteworthy that the continuum-modelling technique utilized in the reconstruction of the present sample was originally developed for multi-scale carbonate rocks [45].

Our largest simulation was executed using the full A1-image [41]. It has the second best resolution of ca 0.92 μm , includes 16384^3 voxels, and requires 4

TB of storage space. We segmented the voxels into pore and solid phases using the gray-scale threshold value 108. The resulting porosity from the A1-binary image, for example, is 0.13435 corresponding to a total of 590 billion fluid or pore sites. The sensitivity of the sample properties, e.g. porosity and specific surface area, to the threshold value have been discussed in Ref. [41]. In general, imaging artefacts, mainly noise and edge softening (caused by the optics and finite size of the X-ray spot), can result some inaccuracy to the image processing. In binary segmentation, where data is divided into solid and pore space, the noise can usually be handled sufficiently by pre- and post-processing algorithms. The edge softening, however, is harder to remove, and can cause error to the pore size distributions by making the pores to appear larger or smaller depending on the selected threshold value. The amount of edge softening is dependent on imaging parameters. The synthetic X-ray images under consideration do not carry these artefacts.

The binary X-ray images are here utilized directly as simulation geometries, i.e. each voxel is mapped into a lattice site. In order to reduce the size of the input images, we developed a custom file format: e.g. the A1-binary image file size with this custom format was around 140 times smaller than with the original *gbd*- or *raw*-format (29 GB instead of 4 TB). Table 1 summarizes the synthetic X-ray images here utilized in simulations.

Table 1: The synthetic X-ray images here utilized in simulations. The pore voxel count is obtained when the voxels are segmented into pore and solid phases using the gray-scale threshold value 108.

Image	Resolution (μm)	Number of voxels	Number of pore voxels
A32	29	512^3	1.66×10^7
A16	15	1024^3	1.42×10^8
A8	7.3	2048^3	1.15×10^9
A4	3.7	4096^3	9.23×10^9
A2	1.8	8192^3	7.39×10^{10}
A1	0.92	16384^3	5.91×10^{11}

3.1. Image-Based Structural Analysis

The characteristic measures of the Fontainebleau sandstone are well documented [44]: the mean length of the pore channels is $130 - 200 \mu\text{m}$, which is comparable to its average grain size $200 - 250 \mu\text{m}$, while the average radius of the pores and the effective pore throats are only 45 and $20 \mu\text{m}$, respectively. In order to verify that the synthetic X-ray images utilized truly represent Fontainebleau sandstone, we carried out a microstructure analysis for a 1024^3 voxel subvolume of the A4-image (the subvolume coined as $x = 0, y = 0, z = 0$ in Ref. [41]).

The separation of individual grains (see Fig. 3) was performed using the Watershed segmentation [46]. At first, an Euclidean distance transform (EDT) was applied to the original data. We assume that contact areas between the

Figure 3: The synthetic X-ray tomography images utilized represent the microstructure of Fontainebleau sandstone, and involve around one million polyhedrons describing the quartz grains. The individual grains are here identified and color-coded in a small volume by an analysis software.

Figure 4: The distributions of the longest and the shortest semi-axes computed for the grains. The average values are $116\mu\text{m}$ and $73\mu\text{m}$ for the longest and the shortest axis, respectively.

grains are small compared to the grain thickness, and thus each grain contains a local EDT maximum. The local maximum is used as a seed point for the Watershed transform resulting in the separation of the grains from the narrowings in between them. As a result we have the individual grains separated. To analyze the grain dimensions we use image moments [47] to define the length of the main axes of the grain (elliptic fitting). The throat areas are computed using the Marching cubes triangulation [48]. The analysis of the pore space is done similarly.

Figure 5: The distributions of longest and shortest semi-axes computed for the pores. The average values are $75\mu\text{m}$ and $35\mu\text{m}$ for the longest and shortest axis, respectively.

Figure 6: The distribution of pore throat radii. The average value is $21\mu\text{m}$.

The distributions of the grain dimensions, i.e. the longest and shortest semi-axes, are presented in Fig. 4. The average values are $116\mu\text{m}$ and $73\mu\text{m}$ for the longest and shortest axis, respectively. Thus, based on the longest axis, the average grain size is approximately $232\mu\text{m}$ which agrees with previously documented values. Furthermore, the pore size distributions are shown in Fig. 5: the average values are $75\mu\text{m}$ and $35\mu\text{m}$ for the longest and shortest axis, respectively. Hence, the computed average pore length $150\mu\text{m}$ is comparable to values reported in the literature, and also the average pore radius $35\mu\text{m}$ is in a reasonable agreement.

Finally, the pore throat radius is approximated from its area by assuming

a circular shape. The resulting distribution of throat radii is shown in Fig. 6. The average radius $21\mu\text{m}$ is in accordance with published values.

4. Computational Experiments

Our aim here is to demonstrate that, using current computing capabilities, direct pore-scale fluid flow simulations are feasible even in the case of very large system sizes which allow the closing of particular scale gaps discussed in the introduction. To this end we conduct computational experiments in order to quantitatively observe, in a realistic test case,

1. how large domain sizes can be simulated with contemporary CPU-systems,
2. how large domain sizes can be simulated with contemporary GPU-systems,
3. the relative computational performance of the CPU- and GPU-systems.

We will use the synthetic X-ray tomography images presented in Sec. 3 for our case studies of fluid flow simulations through porous media.

4.1. Simulation results in a CPU-system

The Archer supercomputer at the Edinburgh Parallel Computing Centre was used to demonstrate the computing capabilities of CPU-systems. It has a total of 3008 compute nodes providing 1.56 Petaflops of theoretical peak performance (phase 1 installation). Each node has two 12-core Ivy Bridge 2.7 GHz CPUs and 64 GB of memory. The code was compiled using the *gcc* compiler version 4.8.1.

To begin with, we benchmarked the computational performance of the Archer nodes by varying the number of MPI processes per node and OpenMP threads per MPI process. As a test geometry we used the full A16-image and two Archer nodes were allocated for the computing. The measured performances are presented in Fig. 7 and reported in Million Fluid Lattice site Updates Per Second (MFLUPS). The utilization of 48 threads per node with hyper-threading (HT) is clearly beneficial: the best performance, 183 MFLUPS, is measured with 4 MPI processes per node, and with 1 MPI process per node the performance is 181 MFLUPS. As the difference between these two cases is marginal, and since we prefer to have large subdomains per MPI process in order to minimize MPI communication, we will use 1 MPI process and 48 OpenMP threads per node in the simulations presented below. The simple 32-bit indirect addressing was used in this benchmark.

Next we evaluate the performance of indirect memory addressing schemes on Archer. Applying the three different addressing schemes presented in Sec. 2.2.2, we ran the CPU code version on sample geometries of various sizes. For each geometry a common domain decomposition was specified. The number of computing nodes was not minimized in these tests which led to a slightly variant per node memory usage (between 16 GB and 27 GB for different geometries). The tests were run for 500 discrete time steps allocating 1 MPI process and 48 OpenMP threads with hyper-threading per node.

Figure 7: Computational performance of Archer nodes in a test case: the full A16-image and two Archer nodes were used while the number of MPI processes per node and OpenMP threads per MPI process were varied. The performance is reported in Million Fluid Lattice site Updates Per Second (MFLUPS), and (HT) refers to hyper-threading. The simple 32-bit indirect addressing was used in this benchmark.

From Table 2 it can be seen that the simple 32-bit addressing scheme is the most efficient, reaching 187.9 MFLUPS per node for the A4 image. Switching to the simple 64-bit addressing reduces the performance about 9-13% while the complex 32-bit addressing inflicts a penalty of around 19-27% on the performance. In order to minimize the memory overhead, instead of optimizing the performance, all subsequent simulations with Archer are done using the complex 32-bit indirect addressing.

Table 2: Computational performance of the CPU code version with the three indirect addressing schemes presented in Sec. 2.2.2. The benchmarks are executed on Archer using four images.

Sample	Cartesian partition	Performance (MFLUPS)		
		Simple 32-bit	Simple 64-bit	Complex 32-bit
A16	$2 \times 1 \times 1$	179.5	155.5	131.4
A8	$5 \times 2 \times 1$	178.2	161.8	141.4
A4	$5 \times 5 \times 4$	187.9	166.7	148.3
A2	$11 \times 11 \times 10$	169.0	152.4	137.7

Figure 8: Weak scaling measured on Archer. The parallel efficiency is reported on the left above the columns (with respect to the simulation with the A8-image). The GFLOPS results are estimated using 213 floating-point instructions per fluid site update.

4.1.1. Weak scaling, CPU

In order to measure the parallel efficiency of our CPU implementation we carried out a weak scaling experiment. Table 3 lists the images used in the simulations and the corresponding cartesian domain partitions. In each weak scaling step the total workload increases by a factor of 8. The maximum and minimum workload, i.e. the number of fluid or pore sites in a subdomain assigned to an MPI process, are also reported in Table 3. The workloads are well balanced except in the largest simulation with the A1-image.

Table 3: The images used in a weak scaling experiment with the CPU code and the corresponding cartesian domain partitions. The maximum and minimum workload, i.e. the number of fluid or pore sites in a subdomain assigned to a MPI process, is also reported.

Sample	Cartesian partition	Workload (million pore sites)		
		Maximum	Minimum	Ratio
A8	$5 \times 1 \times 1$	232	229	1.01
A4	$5 \times 4 \times 2$	236	226	1.04
A2	$8 \times 8 \times 5$	246	218	1.13
A1	$16 \times 15 \times 12$	261	168	1.56

Results from the weak scaling experiment are presented in Fig. 8. The parallel efficiency is very good, even in the largest case with unbalanced workloads. The GFLOPS results are estimated using 213 floating-point instructions per fluid site update (instructions counted manually from the source code). The

Figure 9: Evolution of computed permeabilities in simulations when using A32- and A16-images and various values for the relaxation time $\tau^* \equiv \tau_e^*$.

largest simulation with the A1-image was executed using 138240 threads on 2880 nodes (96% of the total node count). Steady-state was reached after 20 000 discrete time steps requiring approximately 10 hours of computing: 89% of the total computing time was spent in the simulation kernel, the rest in file input and output operations. The simulation geometry based on the binary A1-image includes around 590 billion fluid lattice sites. The computational performance was 78.6 Teraflops, i.e. 5% of the theoretical peak performance, and this performance measure was obtained by ignoring the file I/O-operations. The parallel efficiency was 0.91 for the largest simulation. The above computational performance results are on a par with results reported for similar parallel lattice-Boltzmann implementations (cf. Ref. [49]). The maximum memory consumption by a node was 59 GB (92% of the available memory).

4.1.2. Computed permeability

The above weak scaling measurement was done based on full-fledged, steady-state fluid flow simulations. The relevant simulation parameters are here given in dimensionless (lattice) units indicated by the superscript *. A small external acceleration $g_z^* = 10^{-6}$ was enforced in order to simulate flows in the low Reynolds number regime: Darcys law is applied in the computation of permeability which assumes a Stokes flow. First, we observed the effect of relaxation time on the computed permeability. Simulations on A32- and A16-images were carried out with various values for $\tau^* \equiv \tau_e^*$ and the results are presented in Fig. 9. It is immediately clear that in the case of the A32-image the computed permeability values depend strongly on the relaxation time. On the other hand, when using the A16-image, permeabilities converge to a common value independent of the relaxation time.

Flow simulations with the A16-image (resolution $14.7 \mu\text{m}$) give a permeabil-

Figure 10: Evolution of computed permeabilities in simulations with the images A16, A8, A4, A2, and A1 ($\tau^* = 0.65$). The measured Reynolds number as well as the maximum and average dimensionless velocity are presented as a function of image resolution. The Reynolds number is computed using the pore diameter $69 \mu\text{m}$ (from Sec. 3.1) as a characteristic length, and the measured average velocity as the characteristic velocity.

ity value of 1.24 D; the different values obtained with the A32-image (resolution $29.3 \mu\text{m}$) are well below this. This can be explained by the characteristic measures of Fontainebleau sandstone [44]: the mean length of pore channels is $130 - 200 \mu\text{m}$, which is comparable to its average grain size, while the average values for the effective pore throat and pore radius are only 20 and $45 \mu\text{m}$, respectively. In Sec. 3.1 it was shown, using an image-based structural analysis, that the synthetic X-ray tomography images here utilized respect these characteristic measures. Since the lower resolution of the A32-image was greater than the smallest characteristic size, i.e., the average radius of pore throats, the digital image could not capture properly the critical features of the sandstone, and the poor representation of the true sample was reflected in the simulated values of permeability.

The convergence towards steady-state for the smaller images was fastest with $\tau^* = 0.65$ corresponding to the kinematic viscosity $\nu^* = 0.05$. Hence, only this parameter value was used in simulations with the images, A8, A4, A2, and A1: the corresponding results are presented in Table 4 and Fig. 10. As the resolution is increased, the permeability values clearly approach the value 1.16 D obtained with the A1-image. Table 4 reports also the absolute values of error in permeability computed relative to the reference value obtained with the A1-image. Note that the simulations are run for various numbers of time steps,

Table 4: The number of time steps executed in the steady-state simulations ($\tau^* = 0.65$) for each sample geometry as well as the extracted permeability value. The error in permeability (absolute value) is computed relative to the reference value obtained with the A1-image.

Sample	Time steps	Permeability	Relative error
A16	20000	1.2410	6.767×10^{-2}
A8	10000	1.1889	2.281×10^{-2}
A4	5000	1.1693	5.981×10^{-3}
A2	10000	1.1607	1.481×10^{-3}
A1	20000	1.1624	—
Convergence rate of error			$(\Delta r)^{1.85}$

due to limited access to computing resources, and hence the deviation from the steady-state varies between simulations. The number of discrete time steps required to reach a steady-state depends in a complex manner on the resolution and the relaxation time. Here this dependence is not considered further.

Nevertheless, the observed rate of convergence for the relative error is $(\Delta r)^{1.85}$. This is noteworthy because, in general and from a theoretical perspective, the halfway-bounceback boundary scheme degrades LB implementations into only first-order accurate with respect to lattice spacing. However, for the purpose of approximating permeability, which is a system quantity depending on the average of flow field, LB implementations relying on the halfway-bounceback scheme appear to be effectively second-order accurate.

Furthermore, Fig. 10 shows the measured Reynolds number as well as the maximum and average dimensionless velocity as a function of image resolution. The Reynolds number is computed using the pore diameter $69 \mu\text{m}$ (from Sec. 3.1) as a characteristic length, and the measured average velocity as the characteristic velocity. The measured dimensionless velocities are proportional to Δr^{-2} which is the expected scaling with constant ν^* and g_z^* . Furthermore, the Reynolds number scales with Δr^{-3} .

In addition, the measured permeability values agrees well with the previously reported value, 1.18 D, computed for a small subdomain of 300^3 voxels with a resolution of $7.5 \mu\text{m}$ [42]. This kind of agreement is expected since Fontainebleau sandstone is homogeneous, and its RVE is small: an edge length of 0.7 mm, just a few grains, has been proposed for the RVE [50]. This is certainly smaller than the subdomain utilized in Ref. 27. To conclude, the small span of length scales presented above, $20 - 700 \mu\text{m}$, reflects the single-scale nature of Fontainebleau sandstone, and it is thus easily represented in numerical simulations. Truly multi-scale porous materials require large system sizes and hence pose computationally far more difficult problems. Figure 11 visualizes a small part of the simulated flow field in the A1-image.

Finally, from the simulation with the A1-image, the tortuosity value 1.78 was measured using the approximation presented in Ref. [51]. The average and maximum velocities were $u_{ave}^* = 3.6 \times 10^{-4}$ and $u_{max}^* = 1.7 \times 10^{-2}$, respectively.

Figure 11: a) Simulated flow field in the A1-image, which is shown for a small sub-volume only. Red and blue arrows indicate fast and slow local flow velocities, respectively, while yellow and green represent intermediate velocities. The main flow direction is from below to above the presented subsample. b) The relative size of the shown sub-volume in comparison with the full computational domain. c) A further illustration of the proportions using a cross-section of the segmented 3D binary image that describes the computational domain.

The ratio between the maximum and the minimum local density was 1.006. In comparison to the initial state, the relative change of the total system mass at the steady-state was about 2×10^{-6} . This negligible change is due to the inlet and outlet boundary conditions utilized which do not conserve mass.

4.2. Benchmark results for a GPU-system

To estimate the computing capabilities of modern GPU-based systems in porous materials research we carried out parallel efficiency benchmarks on the Titan supercomputer (Oak Ridge National Laboratory) ranked 2nd among supercomputers [1]. It has 18688 computing nodes with AMD Opteron 6274 16-core CPU and one Nvidia Tesla K20X GPU per node; each GPU has 6 GB of memory. The theoretical peak performance of the system is 27 PFLOPS.

Our GPU code was implemented using CUDA 5.5 toolkit and compiled with the *gcc* compiler, version 4.8.2. Furthermore, double precision floating-point numbers, recursive bisection domain decomposition, and asynchronous MPI data communication (CPU-CPU) were utilized. The domain partition was such that each subdomain fitted into the memory of a single GPU. The results from the weak scaling test are presented in Fig. 12. The parallel efficiency, relative to the performance measured with the A16-image, is excellent even in the case of the largest simulation with half of the A1-image. Table 5 shows the workload balance for the different weak scaling cases. Here the recursive bisection

Figure 12: Weak scaling measured on Titan. The parallel efficiency is reported on the left above the columns (with respect to the simulation with the A16-image). The TFLOPS results are estimated using 275 floating-point instructions per fluid site update.

Table 5: The images used in a weak scaling experiment with the GPU code and the corresponding domain partitions. The maximum and minimum workload, i.e. the number of fluid or pore sites in a subdomain assigned to a GPU, is also reported.

Sample	Partition	Workload (million pore sites)		
		Maximum	Minimum	Ratio
A16	$2 \times 2 \times 2$	17.6	17.4	1.008
A8	$4 \times 4 \times 4$	17.9	17.8	1.011
A4	$8 \times 8 \times 8$	18.1	17.8	1.012
A2	$16 \times 16 \times 16$	18.1	17.9	1.009
A1 (half)	$32 \times 32 \times 16$	18.2	17.9	1.018

domain decomposition leads to an almost perfect workload balance between the subdomains.

The TFLOPS results are estimated using 275 floating-point instructions per fluid site update; the instruction count is obtained using the NVIDIA Visual profiler. On Titan the largest simulation was executed with one half of the A1-image due to the limited memory on GPUs: using 16384 computational nodes or 88% of the total node count, a computational performance of 1.77 PFLOPS was measured, 6.5% of the theoretical peak performance. Based on the profiler data from a simulation with a single GPU, we estimate that the code uses over 85% of the available memory bandwidth on the GPUs and is thus memory bandwidth limited.

Figure 13 shows results from strong scaling benchmarks on Titan. The ideal scaling presented is based on the performance measured with the A16 image and

Figure 13: Strong scaling measured on Titan. The ideal scaling presented is based on the performance measured with the A16 image and 8 computing nodes. The largest simulations are run using 16384 computing nodes.

8 computing nodes. In general, good strong scaling is observed. For example, the computational performance with the A4 and A8 images scale well up to 4096 compute nodes. On the other hand, the scaling issues observed with more than 4096 compute nodes stem from the communications network on Titan, i.e. the performance becomes communication bound. Part of the communication problems are attributed to the fact that the data transfer between the GPUs need the additional step over the PCI-e bus on each node for the messages to get from the network adapter to the device. The topology of the interconnect, as well as the fact that the system is a shared resource, also affects the performance of the interconnect system.

5. Discussion

The reported, fully resolved steady-state fluid flow simulation with the A1-image has an unprecedented resolution for the given sample size. Simulation with around 590 billion pore sites and 20000 discrete time steps required approximately 9 hours of computing and practically all computational resources available on the Archer supercomputer ranked number 19 among supercomputers [1]. Furthermore, the largest benchmark simulations on Titan, ranked 2nd among supercomputers [1], delivered performance beyond one PFLOPS when using 88% of the computational resources.

At the same time, the top position is currently held by the Tianhe-2 super-computer, developed and hosted by National University of Defense Technology, China, which already has a theoretical peak performance of 54.9 Petaflops. In

addition, even the pessimistic estimates on the performance development of supercomputers promise significant improvements in computational capabilities – especially if measured by the average performance of a supercomputer. Thus, the extreme simulation reported above will be feasible for a wider community in the near future. The hardware configuration with a large memory on each computing node appears to be well suited for steady-state fluid-flow simulations in porous media using LBM: it allows for large subdomains per node thus improving the ratio of computation to intra-node communication. Finally, the measured computational performances are currently limited by the memory bandwidth.

In the above simulations we utilized synthetic X-ray-tomography images that represent the microstructure of Fontainebleau sandstone: these images are freely available and they are the world’s largest 3D images of a porous material [41]. The largest synthetic image available is 32768^3 voxels in size, which is currently unattainable by direct X-ray tomography. In X-ray tomography, the 3D image size of a scan is limited by the number of pixels in the detector. For example, Bruker Corporation has recently introduced the SkyScan™ 2211, a multi-scale high-resolution X-ray nanotomograph, capable of producing an $8000 \times 8000 \times 2300$ voxel image with a single scan [52]. The image size can be increased by using a montage of several reconstructed 3D volumes [53] or using an extended field of view in the individual 2D projection images by merging information from various detector positions. Another technique to acquire multi-scale 3D data sets is serial sectioning [54]. The method is destructive and based on high-resolution 2D imaging of these sections. 2D structural information can be acquired either from individual sections or directly from a block face. Individual sections can be prepared using microtome methods, or a block face can be precisely milled using a focused ion beam [55]. A multi-scale imaging technique, where images of varying size and resolution, possibly produced by different imaging techniques, are combined, is still under development [17, 56, 57, 58, 59].

The most commonly used sample for direct, or indirect, laboratory measurements of rock properties before any up-scaling processes is a plug. Plugs are extracted from core samples recovered during drilling, and typically they have a diameter of 3.8 cm and are 5–10 cm long. Drilling-core-scale samples, with a diameter of about 10 cm and about a meter long, are not commonly used in laboratory measurements due to the difficulty in handling them. Nevertheless, laboratory measurements even for plugs can be very difficult to perform, e.g., when the relative permeability is measured in reservoir conditions. Moreover, such measurements can be very slow: in some cases they may take even months to complete. As an alternative, digital rock physics or digital core analysis [18, 19, 20] uses digital 3D representations of rock samples obtained (typically) by X-ray microtomography. These 3D images represent the internal structure of rock, and they can be used for, e.g., direct pore-scale simulations of physical phenomena. Such simulations allow, for example, a numerical determination of the permeability of the rock. Furthermore, such simulations can speed-up reservoir evaluations and enable the determination of rock properties in conditions

that are difficult, or even impossible, to mimic in the laboratory. For example, simulations can enable the evaluation of wettability on the relative permeability [60, 61].

The situation is quite similar in soil research. First of all, structure is a key factor that affects the functioning of soil. Important related processes include water storage and supply to plants, aeration of soils, and water infiltration and drainage. Thus there is a great need to understand the connection between the structure and the transport properties of soil. However, soils inherently involve multiple scales, which hampers such research. Clay soils in particular include an enormous range of length scales. They are formed primarily of clay particles which have dimension of the order of 100 nm. These mineral particles form compound particles, clusters, aggregates and clods. Soil properties may vary significantly in the field scale, i.e., over a distance of, say, 1000 m. We thus end up with at least 10 orders of magnitude of length scales, all of which are relevant when considering the functioning of soils. The complementary picture of soil structure is given by considering instead of solids the pore space, which comprises intra- and inter-aggregate voids, root channels, earthworm burrows, and shrinkage cracks. Thus clay soils are truly multi-scale porous media, whose functioning cannot be understood without considering many phenomena that take place at various length scales.

As a consequence of this, there is currently no way to model on a detailed level the field-scale flow phenomena in soils, such that they would resolve phenomena that occur on a pore scale. Thereby continuum-level modelling will be needed in, e.g., environmental load assessment in the foreseeable future. Continuum models need, however, accurate and physically sound descriptions of the relevant soil processes and structures, which are difficult to obtain by traditional experimental techniques. To this end, 3D imaging and image-based flow simulations have recently been increasingly used to study transport processes on a pore scale. The rapid development of non-destructive 3D imaging techniques, especially the X-ray tomography, has to some extent allowed a direct observation of the multi-scale geometry of soil pore-network systems [62]. Combining such imaging with pore-scale simulation techniques, such as, e.g., the lattice-Boltzmann method [11, 12], provides then direct means to quantify the effects on the macroscopic constitutive hydraulic properties of soil and pore structure and pore-scale flow phenomena. Nevertheless, the generalization and up-scaling of pore-scale simulation results for continuum-modelling purposes is not a straightforward task, and any progress in bridging the existing gaps in length scales would be of fundamental importance.

Furthermore, as X-ray tomography is a non-destructive technique, exactly the same samples could be used in the experiments and imaging. Imaging and image-based simulations would thus provide us with direct information of the flow processes responsible for the experimental results, i.e., they would provide new means to understand and interpret the experiments. The increased image size in pore scale simulations would also be important for continuum modelling. Presently the pore and field-scale models are separated as there is a clear scale gap between the length scales that can, respectively, be reached within these

approaches. Bringing the pore scale simulations to totally new sample sizes (while keeping the resolution fixed), would allow at least a partial bridging of this scale gap. These simulations could thus directly provide us with information needed to develop more realistic parametrizations for both pore and field-scale models, as well as to extract values for the related parameters.

6. Conclusions

The constant progress in computational software and hardware offers powerful tools for research and development. From a computational performance point of view, the petaflops regime has already been conquered and currently there is a rush towards exascale. Here we addressed the question how to utilize this immense power in a meaningful way. We concentrated on the porous materials research field and demonstrated the current computing capabilities for very large fluid flow simulations.

We utilized synthetic X-ray tomography images representing the microstructure of Fontainebleau sandstone as test geometries: these are the world's largest 3D images of a porous material. The microstructure of these samples was first computationally examined with image-based analysis. Fluid flow simulations through these samples were then executed using the lattice-Boltzmann method. Based on the results, the image-based structural analysis and LBM are both reliable tools as they capture the material properties in a consistent way. In particular, the minimum resolution required for LBM to produce consistent results correlates well with the smallest features present in the porous sample.

Among the presented results, the highlights include the full steady-state flow simulation on a 3D geometry involving 16384^3 lattice cells with around 590 billion pore sites and, using half of this sample in a benchmark simulation on a GPU-based system, a sustained computational performance of 1.77 PFLOPS. These advancements expose new opportunities in porous materials research. For example, bringing the pore-scale simulations to totally new sample sizes, while keeping the resolution fixed, allows the partial bridging of some scale gaps currently present in soil research and reservoir evaluation.

Here the test sample utilized was a homogeneous material. In order to simulate fluid flows in large, heterogeneous or multi-scale systems, balancing the computational workload in parallel processing becomes essential. In addition, the image-based structural analysis was here carried out for a small subvolume only. The treatment of very large images requires high-performance, inherently parallel implementations of these computational tools as well. In fact, in order to carry out very large computational experiments on a particular supercomputer, it might become necessary, or at least convenient, to integrate the structural analysis tools and the simulation software. The same applies to post-processing tools as the results from very large simulations become inconvenient or even impossible to analyse on a desktop or a workstation.

7. Acknowledgments

We acknowledge the financial support from the European Community's Seventh Framework programmes ICT-2011.9.13 and NMP.2013.1.4-1 under Grant Agreements no. 287703 and no. 604005, respectively. We are also grateful for the computational resources provided by CSC IT Center for Science Ltd (Finland) and Edinburgh Parallel Computing Centre (UK). This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Furthermore, we would like to thank R. Hilfer from University of Stuttgart, Institute for Computational Physics, for providing us access to the very large synthetic X-ray tomography images as well as Jyrki Hokkanen, CSC IT Center for Science Ltd, for the visualization of the simulated flow field. Finally, we appreciate the figures of X-ray tomography images provided by Petrobras (Brazil).

References

- [1] TOP500, Supercomputer Sites: Lists of November 2013 and June 2015, <http://www.top500.org/> (26th October 2015).
- [2] Y. Hasegawa et al., First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer, in: Proceedings of the ACM/IEEE SC'11 conference, Seattle, WA, USA, 12–18th November, 2011, pp. 1–11. doi:10.1145/2063384.2063386.
- [3] T. Ishiyama, , K. Nitadori, J. Makino, 4.45 Pflops astrophysical N-body simulation on K computer: The gravitational trillion-body problem, in: Proceedings of the ACM/IEEE SC'12 conference, Salt Lake City, UT, USA, 10–16th November, 2012, pp. 1–10.
- [4] D. Jun, Peta-scale Lattice Quantum Chromodynamics on a Blue Gene/Q supercomputer, in: Proceedings of the ACM/IEEE SC'12 conference, Salt Lake City, UT, USA, 10–16th November, 2012, pp. 1–10. doi:10.1109/SC.2012.96.
- [5] D. Rossinelli et al., 11 PFLOP/s simulations of cloud cavitation collapse, in: Proceedings of the ACM/IEEE SC'13 conference, Denver, CO, USA, 17–22th November, 2013, pp. 1–13. doi:10.1145/2503210.2504565.
- [6] P. Staar et al., Taking a quantum leap in time to solution for simulations of high-Tc superconductors, in: Proceedings of the ACM/IEEE SC'13 conference, Denver, CO, USA, 17–22th November, 2013, pp. 1–11. doi:10.1145/2503210.2503282.
- [7] J. Bédorf et al., 24.77 Pflops on a gravitational tree-code to simulate the milky way galaxy with 18600 GPUs, in: Proceedings of the ACM/IEEE SC'14 conference, New Orleans, LA, USA, 16–21th November, 2014, pp. 54–65. doi:10.1109/SC.2014.10.

- [8] A. Heinecke et al., Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers, in: Proceedings of the ACM/IEEE SC'14 conference, New Orleans, LA, USA, 16–21th November, 2014, pp. 3–14. doi:10.1109/SC.2014.6.
- [9] F. Robertsén, J. Westerholm, K. Mattila, Lattice Boltzmann simulations at petascale on multi-GPU systems with asynchronous data transfer and strictly enforced memory read alignment, in: Proceedings of the Euromicro PDP'15 conference, Turku, Finland, 4–6th March, 2015, pp. 604–609. doi:10.1109/PDP.2015.71.
- [10] N. Jarvis, A review of non-equilibrium water flow and solute transport in soil macropores: principles, controlling factors and consequences for water quality, *European Journal of Soil Science* 58 (3) (2007) 523–546. doi:10.1111/j.1365-2389.2007.00915.x.
- [11] R. Benzi, S. Succi, M. Vergassola, The lattice Boltzmann equation: theory and applications, *Physics Reports* 222 (3) (1992) 145–197. doi:10.1016/0370-1573(92)90090-M.
- [12] C. Aidun, J. Clausen, Lattice-Boltzmann method for complex flows, *Annual Review of Fluid Mechanics* 42 (2010) 439–472. doi:10.1146/annurev-fluid-121108-145519.
- [13] F. Khan, F. Enzmann, M. Kersten, A. Wiegmann, K. Steiner, 3D simulation of the permeability tensor in a soil aggregate on basis of nanotomographic imaging and LBE solver, *Journal of Soils and Sediments* 12 (1) (2012) 86–96. doi:10.1007/s11368-011-0435-3.
- [14] J. Hyväluoma et al., Using microtomography, image analysis and flow simulations to characterize soil surface seals, *Computers & Geosciences* 48 (2012) 93–101. doi:10.1016/j.cageo.2012.05.009.
- [15] P. Nelson, Pore-throat sizes in sandstones, tight sandstones, and shales, *AAPG Bulletin* 93 (3) (2009) 329–340. doi:10.1306/10240808059.
- [16] Y.-Q. Song, S. Ryu, P. Sen, Determining multiple length scales in rocks, *Nature* 406 (6792) (2000) 178–181. doi:10.1038/35018057.
- [17] A. Grader, A. Clark, T. Al-Dayyani, A. Nur, Computations of porosity and permeability of sparic carbonate using multi-scale CT images, in: Proceedings of the SCA'09 symposium, Noordwijk aan Zee, The Netherlands, 27–30th September, 2009, pp. 1–10.
- [18] H. Andrä et al., Digital rock physics benchmarks—Part I: Imaging and segmentation, *Computers & Geosciences* 50 (2013) 25–32. doi:10.1016/j.cageo.2012.09.005.

- [19] H. Andrä et al., Digital rock physics benchmarks–Part II: Computing effective properties, *Computers & Geosciences* 50 (2013) 33–43. doi:10.1016/j.cageo.2012.09.008.
- [20] M. Blunt et al., Pore-scale imaging and modelling, *Advances in Water Resources* 51 (2013) 197–216. doi:10.1016/j.advwatres.2012.03.003.
- [21] M. Balhoff, K. Thompson, M. Hjortsø, Coupling pore-scale networks to continuum-scale models of porous media, *Computers & Geosciences* 33 (3) (2007) 393–410. doi:10.1016/j.cageo.2006.05.012.
- [22] J. Chu, B. Engquist, M. Prodanović, R. Tsai, A multiscale method coupling network and continuum models in porous media II – Single- and two-phase flows, in: R. Melnik, I. Kotsireas (Eds.), *Advances in Applied Mathematics, Modeling, and Computational Science*, Vol. 66 of Fields Institute Communications, Springer US, 2013, pp. 161–185. doi:10.1007/978-1-4614-5389-5_7.
- [23] W. E, B. Engquist, The heterogenous multiscale methods, *Communications in Mathematical Sciences* 1 (1) (2003) 87–132.
- [24] W. E, B. Engquist, X. Li, W. Ren, E. Vanden-Eijnden, Heterogeneous multiscale methods: A review, *Communications in Computational Physics* 2 (3) (2007) 367–450.
- [25] H. Chen et al., Extended Boltzmann kinetic equation for turbulent flows, *Science* 301 (5633) (2003) 633–636. doi:DOI:10.1126/science.1085048.
- [26] K. Stratford, R. Adhikari, I. Pagonabarraga, J.-C. Desplat, M. Cates, Colloidal jamming at interfaces: A route to fluid-bicontinuous gels, *Science* 309 (5744) (2005) 2198–2201. doi:10.1126/science.1116589.
- [27] A. Peters et al., Multiscale simulation of cardiovascular flows on the IBM Blue Gene/P: full heart-circulation system at near red-blood cell resolution, in: *Proceedings of the ACM/IEEE SC’10 conference*, New Orleans, LA, USA, 13–19th November, 2010, pp. 1–10. doi:10.1109/SC.2010.33.
- [28] D. Rothman, Cellular-automaton fluids: A model for flow in porous media, *Geophysics* 53 (4) (1998) 509–518. doi:10.1190/1.1442482.
- [29] S. Succi, E. Foti, F. Higuera, Three-dimensional flows in complex geometries with the lattice Boltzmann method, *Europhysics Letters* 10 (5) (1989) 433–438. doi:10.1209/0295-5075/10/5/008.
- [30] A. Cancelliere, C. Chang, E. Foti, D. Rothman, S. Succi, The permeability of a random medium: Comparison of simulation with theory, *Physics of Fluids A* 2 (12) (1990) 2085–2088. doi:10.1063/1.857793.
- [31] Y. Qian, D. d’Humières, P. Lallemand, Lattice BGK models for Navier-Stokes equation, *Europhysics Letters* 17 (6) (1992) 479–484. doi:10.1209/0295-5075/17/6/001.

- [32] I. Ginzburg, D. d’Humières, Multireflection boundary conditions for lattice Boltzmann models, *Physical Review E* 68 (6) (2003) 066614. doi:10.1103/PhysRevE.68.066614.
- [33] P. Philippi, L. Hegele Jr, L. Emerich dos Santos, R. Surmas, From the continuous to the lattice Boltzmann equation: The discretization problem and thermal models, *Physical Review E* 73 (5) (2006) 056702. doi:10.1103/PhysRevE.73.056702.
- [34] R. Cornubert, D. d’Humières, D. Levermore, A knudsen layer theory for lattice gases, *Physica D: Nonlinear Phenomena* 47 (1–2) (1991) 241–259. doi:10.1016/0167-2789(91)90295-K.
- [35] P. Bailey, J. Myre, S. Walsh, D. Lilja, M. Saar, Accelerating lattice Boltzmann fluid flow simulations using graphics processors, in: *Proceedings of the ICPP’09 conference, Vienna, Austria, 22–25th September, 2009*, pp. 550–557. doi:10.1109/ICPP.2009.38.
- [36] M. Schulz, M. Krafczyk, J. Tölke, E. Rank, Parallelization strategies and efficiency of CFD computations in complex geometries using lattice Boltzmann methods on high-performance computers, in: M. Breuer, F. Durst, C. Zenger (Eds.), *High Performance Scientific And Engineering Computing*, Vol. 21 of *Lecture Notes in Computational Science and Engineering*, Springer Berlin Heidelberg, 2002, pp. 115–122. doi:10.1007/978-3-642-55919-8_13.
- [37] G. Wellein, T. Zeiser, G. Hager, S. Donath, On the single processor performance of simple lattice Boltzmann kernels, *Computers & Fluids* 35 (8–9) (2006) 910–919. doi:10.1016/j.compfluid.2005.02.008.
- [38] K. Mattila, J. Hyväluoma, J. Timonen, T. Rossi, Comparison of implementations of the lattice-Boltzmann method, *Computers & Mathematics with Applications* 55 (7) (2008) 1514–1524. doi:10.1016/j.camwa.2007.08.001.
- [39] A. Shet et al., Data structure and movement for lattice-based simulations, *Physical Review E* 88 (1) (2013) 013314. doi:10.1103/PhysRevE.88.013314.
- [40] T. Pohl et al., Performance evaluation of parallel large-scale lattice boltzmann applications on three supercomputing architectures, in: *Proceedings of the ACM/IEEE SC’04 conference, Pittsburgh, PA, USA, 6–12th November, 2004*, pp. 21–. doi:10.1109/SC.2004.37.
- [41] R. Hilfer, T. Zauner, High-precision synthetic computed tomography of reconstructed porous media, *Physical Review E* 84 (6) (2011) 062301. doi:10.1103/PhysRevE.84.062301.

- [42] F. Latief, B. Biswal, U. Fauzi, R. Hilfer, Continuum reconstruction of the pore scale microstructure for Fontainebleau sandstone, *Physica A: Statistical Mechanics and its Applications* 389 (8) (2010) 1607–1618. doi:10.1016/j.physa.2009.12.006.
- [43] B. Biswal, C. Manwart, R. Hilfer, S. Bakke, P.-E. Øren, Quantitative analysis of experimental and synthetic microstructures for sedimentary rock, *Physica A: Statistical Mechanics and its Applications* 273 (3) (1999) 452–475. doi:10.1016/S0378-4371(99)00248-4.
- [44] W. Lindquist, A. Venkatarangan, J. Dunsmuir, T.-F. Wong, Pore and throat size distributions measured from synchrotron X-ray tomographic images of Fontainebleau sandstones, *Journal of Geophysical Research: Solid Earth* 105 (B9) (2000) 21509–21527. doi:10.1029/2000JB900208.
- [45] B. Biswal, P.-E. Øren, R. Held, S. Bakke, R. Hilfer, Stochastic multi-scale model for carbonate rocks, *Physical Review E* 75 (6) (2007) 061303. doi:10.1103/PhysRevE.75.061303.
- [46] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (6) (1991) 583–598. doi:10.1109/34.87344.
- [47] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 2nd Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [48] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, *ACM SIGGRAPH Computer Graphics* 21 (4) (1987) 163–169. doi:10.1145/37402.37422.
- [49] C. Godenschwager, F. Schornbaum, M. Bauer, H. Köstler, U. Rüde, A framework for hybrid parallel flow simulations with a trillion cells in complex geometries, in: *Proceedings of the ACM/IEEE SC’13 conference*, Denver, CO, USA, 17–22th November, 2013, pp. 1–12. doi:10.1145/2503210.2503273.
- [50] Richa, Preservation of transport properties trend: computational rock physics approach, Ph.D. thesis, Stanford University, Stanford, CA, USA (2010).
- [51] A. Koponen, M. Kataja, J. Timonen, Tortuous flow in porous media, *Physical Review E* 54 (1) (1996) 406–410. doi:10.1103/PhysRevE.54.406.
- [52] Bruker Corporation, Skyscan 2211: multi-scale X-ray nano-CT system, <http://www.bruker-microct.com/products/2211.htm> (8th June 2015).
- [53] M. Uchida et al., Soft x-ray tomography of phenotypic switching and the cellular response to antifungal peptoids in *Candida albicans*, *Proceedings of the National Academy of Sciences of the United States of America* 106 (46) (2009) 19375–19380. doi:10.1073/pnas.0906145106.

- [54] M. Uchic, Serial sectioning methods for generating 3D characterization data of grain- and precipitate-scale microstructures, in: S. Ghosh, D. Dimiduk (Eds.), Computational Methods for Microstructure-Property Relationships, Springer US, New York, USA, 2011, pp. 31–52. doi:10.1007/978-1-4419-0643-4_2.
- [55] L. Holzer, M. Cantoni, Review of FIB-tomography, in: I. Utke, S. Moshkalev, P. Russell (Eds.), Nanofabrication using focused ion and electron beams: principles and applications, Oxford University Press, New York, USA, 2012, Ch. 11, pp. 410–435.
- [56] R. Sok et al., Pore scale characterization of carbonates at multiple scales: Integration of MicroCT, BSEM and FIBSEM, in: Proceedings of the SCA’09 symposium, Noordwijk aan Zee, The Netherlands, 27–30th September, 2009, pp. 1–12.
- [57] D. Wildenschild, A. Sheppard, X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems, Advances in Water Resources 51 (2013) 217–246. doi:10.1016/j.advwatres.2012.07.018.
- [58] J. Wilson et al., Three-dimensional reconstruction of a solid-oxide fuel-cell anode, Nature Materials 5 (7) (2006) 541–544. doi:10.1038/nmat1668.
- [59] M. Puhka, M. Joensuu, H. Vihinen, I. Belevich, E. Jokitalo, Progressive sheet-to-tubule transformation is a general mechanism for endoplasmic reticulum partitioning in dividing mammalian cells, Molecular Biology of the Cell 23 (13) (2012) 2424–2432. doi:10.1091/mbc.E10-12-0950.
- [60] C. Ping, T. Guo, D. Mingzhe, Z. Yihua, Effects of wettability alternation simulation by lattice Boltzmann in porous media, in: Proceedings of the SCA’12 symposium, Aberdeen, Scotland, UK, 27–30th August, 2012.
- [61] C. Landry, Z. Karpyn, O. Ayala, Relative permeability of homogenous-wet and mixed-wet porous media as determined by pore-scale lattice Boltzmann modeling, Water Resources Research 50 (5) (2014) 3672–3689. doi:10.1002/2013WR015148.
- [62] V. Cnudde, M. Boone, High-resolution X-ray computed tomography in geosciences: A review of the current technology and applications, Earth Science Reviews 123 (2013) 1–17. doi:10.1016/j.earscirev.2013.04.003.