Parallel Computation Particle Methods for Multi-Phase Fluid Flow with Application Oil Reservoir Characterization

John R. Williams, David Holmes and Peter Tilke

Abstract This contribution presents a strategy for programming mechanics simulations including particle methods on multi-core shared memory machines.

1 Introduction

1.1 Oil Reservoir Characterization

Estimation of porous media properties such as absolute and relative permeability are key to managing oil and gas recovery. Understanding the behavior of fluids as they flow through porous media is important to a variety of contemporary problems in earth science and engineering. To complement traditional displacement type experiments on rock core samples [2–6], numerical techniques are used widely for both explicit parameter determination, and as research tools to probe complex physical phenomena, not easily observed in experiments. Here we describe an SPH formulation and validation tests, which can model multi-phase fluid flow through the rock matrix at the pore scale.

Early work into reservoir simulation involved numerical tests on idealized and statistical reconstructions of reservoir rock [7–13], but later, application of X-ray

John R. Williams
Massachusetts Institute of Technology, 77 Massachusetts Av., Cambridge, MA 02139, USA;
e-mail: jrw@mit.edu

David Holmes
James Cook University, Angus Smith Drive, Douglas, Queensland 4811, Australia;
e-mail: dholmes@mit.edu

Peter Tilke
Schlumberger-Doll Research Center, 1 Hampshire St, Cambridge, MA 02139-1578, USA;
e-mail: tilke@slb.com

E. Oñate and R. Owen (eds.), *Particle-Based Methods*, Computational Methods in Applied Sciences 25, DOI 10.1007/978-94-007-0735-1_4,

micro-tomography on core samples [14–17] has provided researchers with voxelized representations of actual rock geometries on which to base more highly accurate and case specific models. Examples of numerical techniques used to determine properties from X-ray CT images include the random walk method (determining permeability from its relationship to diffusion [18–22]), the finite difference method (both fluid flow and electrical diffusion [23]), the finite element method (both fluid flow and electrical diffusion [24–26]) pore network models developed with realistic dimensions and connectivity (single-phase [22–27], two-phase [28, 29]), and the lattice-Boltzmann method (single-phase [19–20, 26, 30, 31], multi-phase [30, 32–34]). Due to its ability to explicitly represent multi-phase wettability and capillary forces, the lattice-Boltzmann method [35, 36] provides the most detail on grain scale flow of conventional numerical methods. There are, however, limitations to lattice-Boltzmann regarding solution robustness (related to statistical 'tuning' parameters) and the method's inability to account for electrical and chemical phenomena that can have important cross-relationships with flow. Instead, we favor an alternative particle based method.

Smooth particle hydrodynamics (SPH) is a mesh-free Lagrangian particle method first proposed for astrophysical problems by Lucy [37] and Gingold and Monaghan [38] and now widely applied to fluid mechanics problems [39–44] and continuum problems involving large deformation [44, 45] or brittle fracture [46]. As a Lagrangian particle method (see also dissipative particle dynamics (DPD) [47, 48]), fluid mass in SPH is advected with each particle. In multi-phase problems, phase interfaces are addressed intrinsically by this mass advection and properties like surface tension, wettability and capillary forces can be included using pair-wise inter-particle forces, analogous to the molecular forces driving such phenomena in reality [43, 49–52]. In our experience, SPH is less sensitive to small corrections in model parameters than lattice-Boltzmann, in-part due to the inherent robustness of a method with direct analogy to molecular physics. Additionally, it has been shown that the generality of the SPH formulation accommodates the inclusion of a variety of physical phenomena with a minimum of effort (miscible flows, chemical transport and precipitation [43, 52–54], thermal problems [39, 42, 55–59] and electrical/magnetic fields [39, 60, 61]).

There is a computational price for managing free particles when compared to grid based alternatives. However, in many circumstances this expense can be justified by the versatility with which such a variety of multi-physics phenomena can be included. Additionally, new parallel hardware architectures such as multi-core [62] are removing many of the barriers which have traditionally limited the practicality of high resolution numerical techniques like SPH.

1.2 Multi-Core Parallel Computing

Multi-core machines can increase the speed at which applications execute. In particular, on board data access is more than 10,000 times faster than cross machine

access. However, new parallel programming challenges are introduced because each core can address all of the main memory, leading to potential memory access conflicts, such as race conditions and deadlock. Some software architects address this by using a process on each core leveraging the operating system which guarantees each process runs in its own isolated memory space. The penalty for this isolation is the time consuming task of cross process communication, which requires object marshalling and then re-instantiation of the object in the new memory space.

Here we show that a large class of computational physics problems, including "particle" simulations, can be decomposed into orthogonal compute tasks that can be executed safely in parallel threads within a single process on multi-core machines. A new task management algorithm called H-Dispatch [62] is developed that allows optimal use of memory by matching the task size to the available L3 cache, while optimizing the CPU usage by employing a "hungry" task pull strategy rather than the common push strategy.

The technique is demonstrated on SPH problems and it is shown that an optimal task size exists. If the task size is too small adding more cores can actually slow down execution because the problem becomes dominated by messaging latency. However, when the task size is increased an optimal speedup is attained. It is shown that a near linear speedup is attained on a 24-core machine. It is noted that the algorithm is quite general and can be a applied to a wide class of computational tasks on heterogeneous architectures involving multi-core and GPGPU hardware. One solution that ensures memory isolation is to run a separate MPI [63] process on each core. The operating system then ensures an isolated memory space for each process. Data is then shared across processes(cores) by sending MPI message requiring object marshalling and un-marshalling. The problem of memory conflicts is avoided but the cross-core/cross-process communication overhead is significant, on the order of 10,000 machine cycles. So if the time to access a variable in memory is say A cycles then we will now incur A + 10,000 cycles to access that same variable in another process. We note that not all data needs to be communicated in this way, and in computational mechanics problems only "ghost region data" is shared across process boundaries. However, in 3D calculations the ghost regions can be roughly 50% of the unknowns. In essence, the MPI strategy turns each core into an information island, with information transfer being limited by the speed at which MPI messages can be marshaled and delivered across processes. While this is around 100 times faster than cross-machine MPI messages, this is still relatively slow compared to sharing main memory between the cores.

An alternative strategy, which allows memory sharing across cores, is to share a single process across all cores, but use separate threads of execution on each core. In order to avoid memory contention "thread safety" must now be managed explicitly by the programmer. "Thread safe" programming can be complex even for the best programmers and the non-deterministic nature of running multiple threads makes detection of race conditions difficult. However, there are specific classes of problem where thread safety can be guaranteed. Indeed, this is the basis of OpenMP [64] and Cilk++ [65] that break "for loops" into parallel execution.

We show below that in a large class of computational physics problems, including "particle" simulations, we can decompose the problem into orthogonal compute task that share memory but execute "safely" in parallel on multi-core machines.

In the next sections we detail an SPH formulation for fluid flow, its validation and testing and its implementation on a multi-core architecture. The problem of managing 3D space to ensure orthogonal compute tasks and the problem of Ghost Regions are addressed.

2 Computational Physics Using Particle Methods

2.1 Overview

Mesh based numerical methods have been the cornerstone of computational physics for decades. Here, integration points are positioned according to some topological connectivity or mesh to ensure compatibility of the numerical interpolation. Examples of Eulerian mesh based methods include finite difference (FD) [66] and the lattice Boltzmann method (LBM) [30, 34–36, 67], while Lagrangian examples include the finite element method (FEM) [68]. While powerful for a wide range of problems, mesh limitations for problems involving large deformation and complex material interfaces has led to significant developments in meshless and particle based methodologies [44–62, 69]. For such methods, integration points are positioned freely in space, capable of advection with material in a Lagrangian sense. For methods like molecular dynamics (MD) [70] and the discrete element method (DEM), such points represent literal particles, atoms and molecules for MD and discrete grains for DEM [71–73], while for methods like dissipative particle dynamics (DPD) [74] and smooth particle hydrodynamics the particle analogy is largely figurative. For such methods, particles provide positions at which to enforce a partition of unity (Figure 1). By partitioning unity across the particles, continuity can be imposed without a defined mesh, allowing such methods to represent a continuum in a generalized way.

In Eulerian mesh based methods like FD and LBM, continuity is inherently provided by the static mesh, while for Lagrangian mesh based approaches like FEM, continuity is enforced through the use of element shape functions. The partition of unity imposed on mesh-free particle methods can be seen to be a generalization of shape functions for arbitrary integration point arrangements. From Li and Liu [44]:

> ... meshfree methods are the natural extension of finite element methods, they provide a perfect habitat for a more general and more appealing computational paradigm – the partition of unity.

The advantage of partition of unity methods is that any expression related to a field quantity can be imposed on the continuum. Where for a bounded domain – in Euclidean space, a set of nonnegative compactly supported functions, $\phi(x_j)$, sums to unity (Figure 1), i.e.

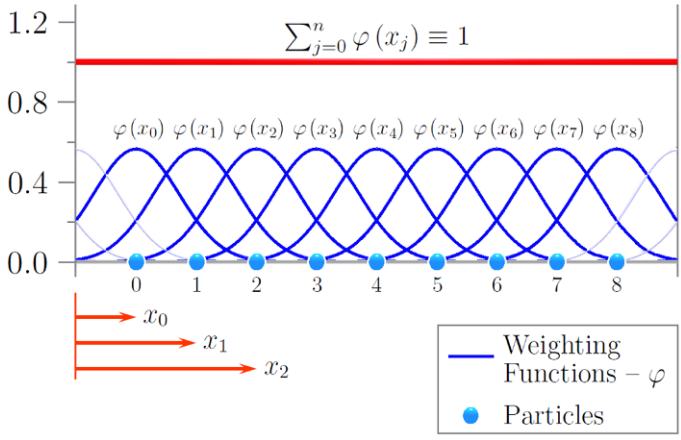

Fig. 1 Partition of unity constructed from basis functions

$$\sum_{j=0}^{n} \phi(x_j) \equiv 1 \text{ on } \Omega \tag{1}$$

Correspondingly, the value of some field function, $f(x_i)$, at the point x_i in space can be determined from its value at all other points via

$$f(x_i) = \sum_{j=0}^{n} \phi(x_j) f(x_j) \tag{2}$$

The function $f(x_i)$, can be related to any physical field expression; hydrodynamic, mechanical, electrical, chemical, magnetic etc. Such versatility is a key advantage of meshfree particle methods.

We shall now derive the equations for fluid flow.

2.2 SPH for Fluid Flow

By discretizing the fluid volume into a finite number of disordered integration points or 'particles', any function, such as density or velocity, can be approximated by the summation interpolant

$$f_i = \sum_{j=0}^{n} \frac{m_j}{\rho_j} f_j \phi(r_i - r_j, h) \tag{3}$$

where smoothing length h is generally set as the initial particle spacing, m_j and ρ_j are the mass and density of particle j at position r_j, and the fraction m_j/ρ_j ac-

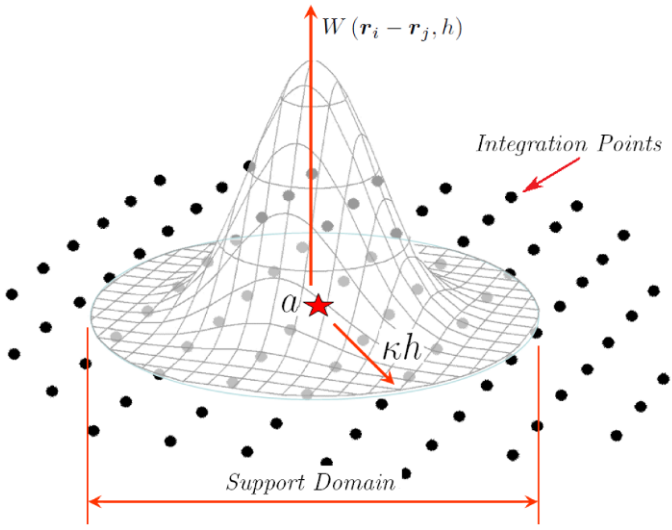

Fig. 2 The support domain and smoothing function in 2 dimension for some particle a

counts for the approximate volume of space each particle represents so as to maintain consistency between the discrete expression (5), and the continuous field that it represents. Correspondingly, the gradient of f is given

$$\nabla f_i = \sum_{j=0}^{n} \frac{m_j}{\rho_j} f_j \nabla_i \phi(r_i - r_j, h) \tag{4}$$

Figure 2 illustrates a smoothing function for a single integration point in space, a.

Authors such as Tartakovsky and Meakin [43, 50] and Hu and Adams [51] have suggested a variation to (5) and (6) where a particle number density term, n_i is used where $n_i = \rho_i / m_i$ and then

$$f_i = \sum_{j=0}^{n} \frac{f_j}{n_j} \phi(r_i - r_j, h) \tag{5}$$

$$\nabla f_i = \sum_{j=0}^{n} \frac{f_j}{n_j} \nabla_i \phi(r_i - r_j, h) \tag{6}$$

Applying this to the particle number density itself

$$n_i = \sum_{j=0}^{n} \phi(r_i - r_j, h) \tag{7}$$

and similarly, mass density of each particle is given by

$$\rho_i = m_i n_i = m_i \sum_{j=0}^{n} \phi(r_i - r_j, h) \tag{8}$$

This expression conserves mass exactly, much like the summation density approach of conventional SPH [19]. Use of a particle number density variant of the SPH formulation is typically motivated by the need to accommodate multiple fluid phases of significantly differing densities. Use of (7) and (8) eliminates the artificial surface tension effects observed by Hoover [40] and removes density discrepancies which would otherwise manifest at phase interfaces. The multi-phase formulation used in this chapter follows that presented by Tartakovsky and Meakin [43, 50].

Determination of particle velocity is achieved through discretization of the Navier–Stokes conservation of linear momentum equation. In this work, a modified version of the expression provided by Morris et al. [41] and used by Tartakovsky and Meakin [43] has been used, where

$$\frac{dv_i^\alpha}{dt} = -\frac{1}{m_i} \sum_{j=0}^{n} \left(\frac{P_i}{n_i^2} + \frac{P_j}{n_j^2} \right) \frac{\partial \phi_{ij}}{\partial r_i^\alpha} + \frac{1}{m_i} \sum_{j=0}^{n} \left(\frac{\mu_i + \mu_j}{n_i n_j} \right) (v_i^\alpha - v_j^\alpha) \frac{r_i^\beta - r_j^\beta}{|r_i^\beta - r_j^\beta|^2} \cdot \frac{\partial \varphi_{ij}}{\partial r_i^\beta} + F_i^\alpha \tag{9}$$

where P_i is the pressure, μ_i is the dynamic viscosity, v_i is the particle velocity and F_i is the body force applied on particle i. Indices α and β refer to vector components and, corresponds to an Einstein's summation on the right of the expression. An equation of state proposed by Morris and co-workers [41] has been used to determine particle pressure at each time step via

$$P_i = c^2(\rho_i - \rho_0) \tag{10}$$

where ρ_0 is the fluid reference density while c is the artificial sound speed. Following Morris et al. [41], the artificial sound speed term, c, should be chosen according to

$$c^2 \approx \text{Max}\left(\frac{\rho_0 V_0^2}{\Delta\rho}, \frac{\rho_0 \nu V_0}{L_0 \Delta\rho}, \frac{\rho_0 L_0 |F|}{\Delta\rho} \right) \tag{11}$$

where ν is the kinematic viscosity $\nu = \mu/\rho_0$, V_0 and L_0 are the velocity and length scales and $|F|$ is the magnitude of body force per unit mass, and $\Delta\rho$ is the maximum allowed amount of density fluctuation (generally chosen as being around 1%) meaning that c will scale with the degree of incompressibility of the system.

In this work, we integrate the differential rate equation (9) using a conventional Leapfrog [35] numerical integration scheme. A stable solution can be achieved by enforcing the following conditions on the time step length [19, 16, 36]

$$\Delta t \leq 0.125 \frac{h^2}{\nu}, \quad \Delta t \leq 0.25 \frac{h}{3c}, \quad \Delta t \leq 0.25 \min(h/(3|F_i|))^{1/2} \tag{12}$$

where $|F_i|$ is the magnitude of the force on a particle.

We use a quintic spline kernel function following Morris [41] such that, given $R = |r_i - r_j|/h$, then

$$W(R,h) = \alpha_d \times \begin{cases} (3-R)^5 - 6(2-R)^5 + 15(1-R)^5 & 0 \le R < 1 \\ (3-R)^5 - 6(2-R)^5 & 1 \le R < 2 \\ (3-R)^5 & 2 \le R < 3 \\ 0 & 3 \le R \end{cases} \quad \text{for} \tag{13}$$

where $\alpha_d = 120/h, \alpha_d = 7/478\pi h^2$, $\alpha_d = 3/359\pi h^3$ in 1, 2 and 3 dimensions respectively.

2.3 Testing and Verification

To verify the accuracy of the developed SPH code, simulations of several well defined one, two and three-dimensional flow problems were carried out. Results from these simulations are detailed in Holmes et al. [62]. As an example we briefly describe a three dimensional test that has direct application to fluid flow through rock cores.

Ordered sphere packings have been used extensively within the literature as an idealized three-dimensional porous medium. Authors such as Hasimoto [75], Zick and Homsy [76] and Sangani and Acrivos [77] have each presented well verified results for flows through simple cubic, body- and face-centered cubic arrays of spheres with porosities ranging up to the close-touching limits of the spheres. In the case of spheres fluid flow will continue in sphere packs well past the point where sphere radii exceed the close touching limit (Figure 3). Authors such as Larson and Higdon [78] and Roberts and Schwartz [79] have used such model geometries to represent consolidated porous media. In this work, we have tested the performance of the developed SPH code for three-dimensional flow using a simple cubic array of spheres with sphere radii up to, and past, the sphere close touching limits as per Larson and Higdon [78] (Figure 3). Again, symmetry of the periodic system facilitated the reduction of the model to the representative three-dimensional unit cell shown in the right part of Figure 3. Periodic boundaries were enforced in all three model dimensions and a center-to-center sphere distance, d, of 1×10^{-3} m was chosen. The fluid was assigned the properties of water ($\rho = 10^3$ kgm^3, $\nu = 10^{-6}$ m^2 s^{-1}).

A variety of solid volume fractions, ϕ, were used in the simulations following the work of Larson and Higdon [78], where

$$\phi = \frac{V_{\text{solid}}}{V_{\text{cell}}} \tag{14}$$

In all simulations, flow was driven from rest by a constant body force of $F = 0.049$ ms^{-2} and an artificial sound speed of $c = 0.07$ ms^{-1} was chosen. During

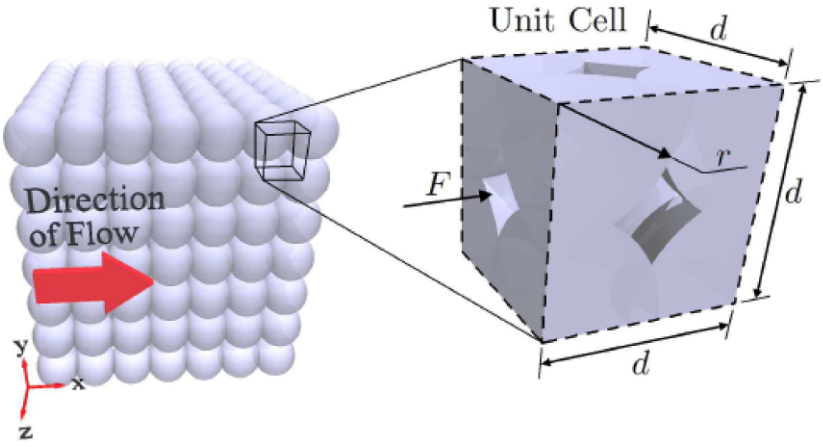

Fig. 3 Cubic packing of overlapping spheres.

model definition, SPH particles were initially arranged on a uniform hexagonal close packed grid and particles positioned inside the bounds of solid spherical grains were designated as being boundary particles.

2.4 Characterization of Flow

Flow through the cubic array of consolidated spheres has been characterized in terms of a friction coefficient, K, and the permeability, k. For the case of a spherical grain in infinite dilution, the drag force, F_d, can be expressed via the Stokes–Einstein form, $F_d = 6\pi\mu r U$. Using this term to non-dimensionalize the sphere pack drag, we establish the friction coefficient used by authors such as Zick and Homsy [76] and Larson and Higdon [78] as

$$K = \frac{F_d}{6\pi\mu r U} \tag{15}$$

The force and velocity terms were determined from simulation results and permeability was determined through Darcy's law with the dimensionless form,

$$\frac{k}{d^2} = \frac{1}{K}\left(\frac{V_{\text{cell}}}{6\pi r d^2}\right) \tag{16}$$

Tests of the mesh sensitivity are reported by Holmes et al. [62] and it was found that approximately 30 particles should span a circular pore throat. The results for dimensionless permeability are plotted below and show good agreement with Sangani and Acrivos.

Table 1 Friction coefficient for various values of solid volume fraction for flow through a cubic array of consolidated spheres

Solid Volume Fraction	Friction Coefficient	
	SPH Results	Larson and Higdon [4]
0.001	1.21	1.21
0.008	1.54	1.52
0.027	1.95	2.01
0.064	2.78	2.81
0.125	4.25	4.29
0.216	7.45	7.44
0.343	15.61	15.40
0.450	27.92	28.09
0.5236	41.95	41.99
0.53	43.78	43.60
0.55	47.93	48.80
0.60	66.41	66.10
0.65	93.29	93.36
0.70	137.37	139.8
0.75	233.48	228.3
0.80	423.10	426.9
0.85	1003	1020
0.90	4366	4290

Table 1 shows the results for the friction coefficient for various volume fractions and shows excellent agreement with those predicted by Larson and Higdon even up to high solid volume fractions, where the pore throats to pore volume ratio is large. Friction coefficient results are plotted in Figure 4, while results for dimensionless permeability are plotted in Figure 5.

3 Application of SPH to Pore Scale Physics

Oil reservoirs are extremely difficult to characterize because in a reservoir extending kilometers only a tiny volume of the rock can be sampled. Furthermore, if the rock is sampled directly by taking a core it is very difficult to maintain the in-situ conditions when the sample is retrieved. Re-creation of in-situ conditions in the laboratory is both time consuming and expensive. Indirect sampling of the rock, using seismic, electromagnetic, acoustic and other means can complement the laboratory tests but often they too are inconclusive. Thus, many researchers believe numerical modeling is an essential tool to give us a better understanding of rock physics. The goal is to predict from digital rock images macroscopic properties such as porosity, absolute permeability, relative permeability, electrical conductivity and elastic properties.

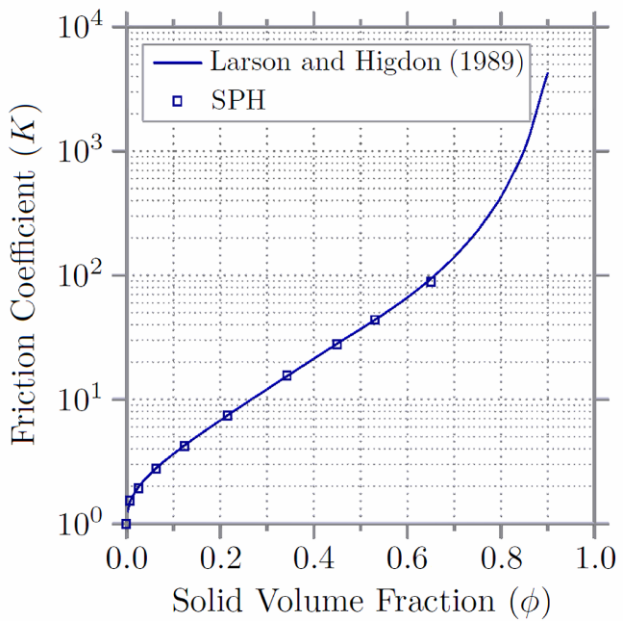

Fig. 4 Friction coefficient

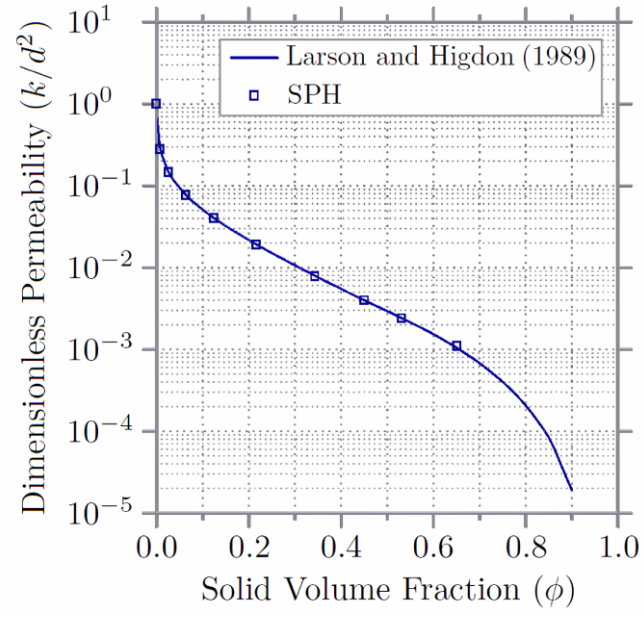

Fig. 5 Dimensionless permeability

Given micro-CT scans of a rock sample, the digital image data is first segmented to detect the boundaries between the rock matrix and the pore space as shown in Figure 5. We note that we can use our partition of unity approach to segment the image. Since the digital image data gives us a voxelized sample of the rock we can directly use the same SPH code to interpolate the data and determine the boundary between rock and pore space.

$$f(x) = \sum_{k=0}^{n} a_k \phi(x - k) \tag{17}$$

where a_k is the voxel value at $x = k$. This process is illustrated in Figure 6 and the workflow for generating a numerical model in Figure 7.

Fluid flow in the range of low Reynolds numbers experiences a no-slip flow condition at solid boundary surfaces. For the permeable rock applications of interest in this paper, flow will occur in this low Reynolds number range and so no-slip boundary conditions must be enforced to accurately reproduce the appropriate flow profiles. The method developed in our previous work by Holmes et al. [80] uses imposed artificial velocities at boundary particles to create antisymmetry in the velocity field at boundary surfaces. This allows complex pore geometries to be simulated as shown below.

The developed SPH simulator has been used to analyze flow though model geometries derived from X-ray CT images of 23.6% porosity Berea sandstone as shown in Figure 6.

Figures 8 and 9 show typical results of multi-phase flow applied to pore scale analysis of an idealized oil reservoir; tests that can be repeated on the digital rock geometries.

4 Parallel Computation on Multi-Core

4.1 Parallel Algorithms and the Ghost Region Issue

In computational mechanics problems involving parallel processing we divide the problem into a number of tasks which can be "scattered" out to the various processors and executed simultaneously. In cross machine computing we divide the unknowns into non-overlapping domains. However, there is spatial coupling of unknowns across domains so each domain must keep a copy of the "ghost region" belonging to its neighboring domains (Figure 10). Updating unknowns from time step N to step $N + 1$ within a domain then proceeds in parallel. Only unknowns "belonging" to the domain are updated at this stage. Once the time step is complete the "ghost regions" are then updated by sending MPI messages from one machine to the other. We note that there needs to be a synchronization point that ensures all machines have finished the update on their own domains before messages updating

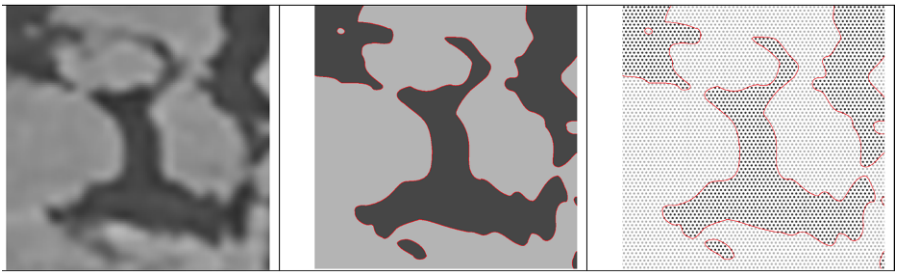

Fig. 6 Digital image segmentation using particles as interpolation functions to directly generate SPH particles

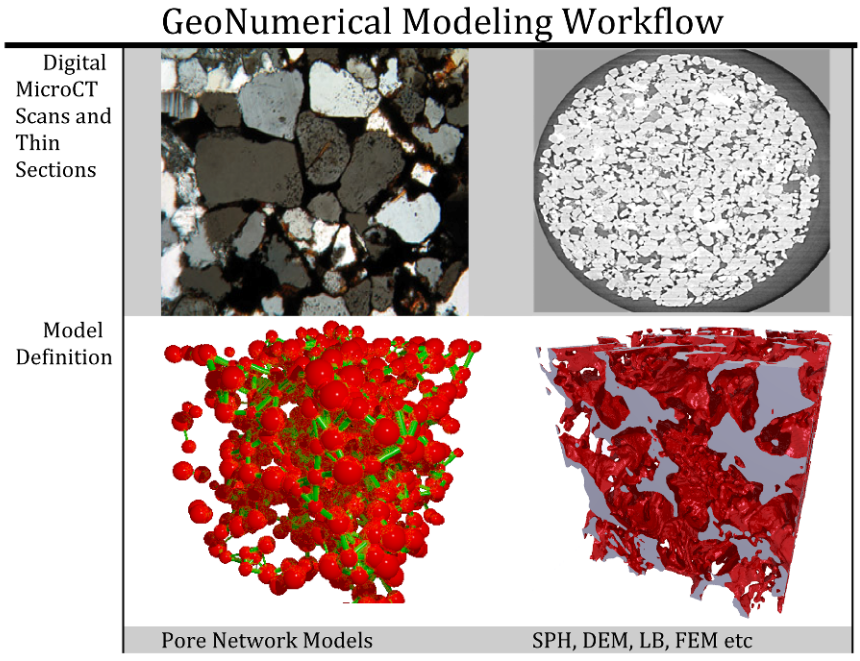

Fig. 7 Workflow for numerical model generation

the "ghost regions" are sent. Such synchronizations generally mean that computations on every machine must halt. Synchronization points in coordinating parallel computing tasks are critical as we shall see below.

If an MPI process is launched on each core then the computational process is almost identical to that described above for cross machine computation. The only difference is that the MPI messages can be optimized for in-machine communication. In the Microsoft.NET, environment marshalling and un-marshalling objects across AppDomain boundaries (somewhat equivalent to Linux process boundaries) allows approximately 100,000 messages per second, so that each message takes

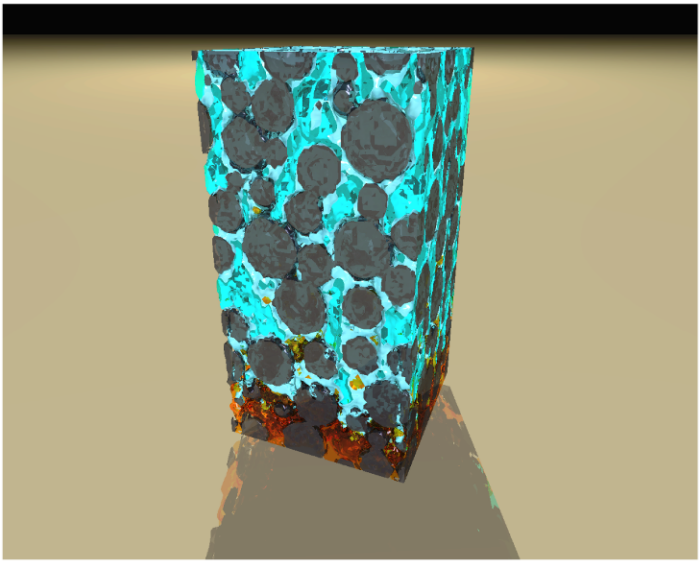

Fig. 8 SPH simulation of flushing of oil from rock pores

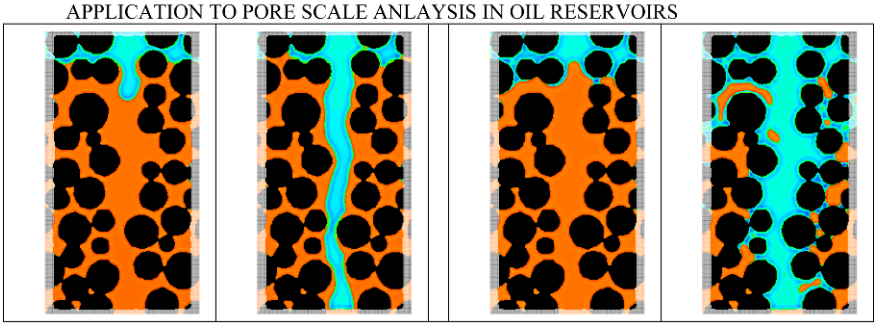

Fig. 9 SPH multi-phase fluid simulation (a) water-rock non-wetting, (b) water-rock wetting

roughly 10,000 machine cycles. Additionally, problem size must be very large to reduce the fraction of ghost points needing to be communicated to manageable levels (Figure 11).

In the case of shared memory there are no "ghost regions" since any unknowns from neighboring domains may be read directly from memory. However, we must now devise a strategy for ensuring that writing does not corrupt data being read. One method is to provide two memory slots for each variable, one for v_n and one for v_{n+1}. Using this strategy only one synchronization point is necessary at the end of the time step.

Typically, the domain boundaries are minimized so that the "ghost regions" are as small as possible and message passing is minimized (see relationship in Figure 11).

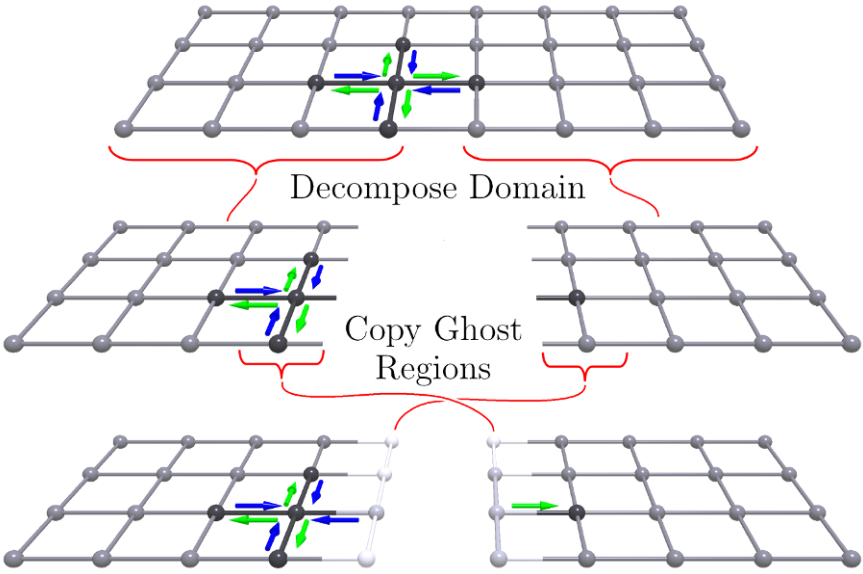

Fig. 10 Illustration of ghost regions in numerical simulation

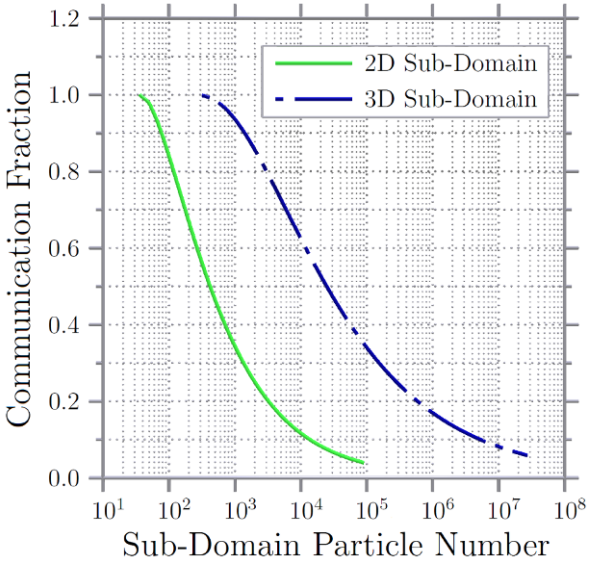

Fig. 11 Data communication fraction in typical cluster computing problem

However, using our shared memory strategy there is no penalty involved in dividing the problem up into smaller domains (Figure 12). Indeed, there is a benefit in doing this because we can now optimize the task size to match the underlying hardware,

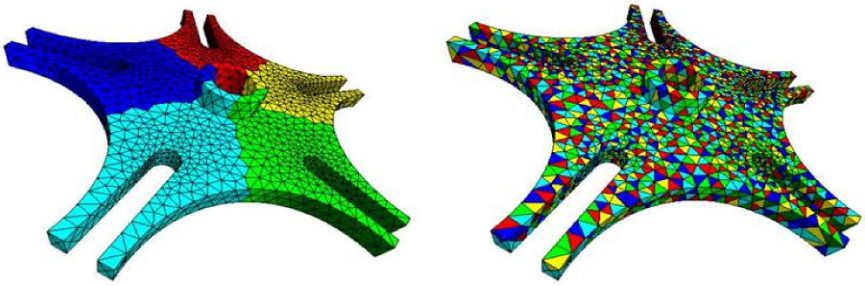

Fig. 12 Contrast of domain decompositon in cluster and multcore computing

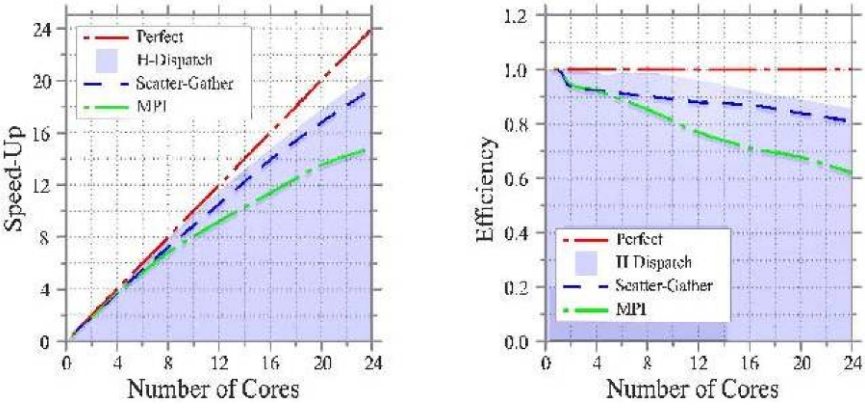

Fig. 13 (a) Speed-up versus number of cores, (b) efficiency versus number of cores

particularly the L3 cache size. On a 24 core machine we have shown that we get better efficiency in breaking the problem into hundreds of smaller domains (see Figure 12). Figure 13 shows the performance of the H-Dispatch strategy compared to MPI and traditional scatter-gather.

When using modern languages such as Java and C#, we need also to minimize garbage collection because all cores must be stopped while the heap is re-mapped. We achieve this by essentially managing memory on each core explicitly. We allocate a block of memory for each core at the start of the computation and hold it until the end of the computation. This is achieved by allocating a master thread on each core. The thread "pulls" tasks from a single dispatcher queue as fast as it can. The task size is such that it can be mapped into the memory allocated. Furthermore, load balancing across cores is ensured no matter if one core runs slower than another. Indeed, even if a core "fails" by not responding within some given period of time, we can resubmit the task to the queue and it will execute on another core. The end of the time-step occurs when there are no more tasks in the queue.

4.2 Spatial Hashing in Particle Methods

The central idea behind spatial hashing is to overlay some regular spatial structure over the randomly positioned particles e.g. an array of equally sized cells. We can then perform spatial reasoning on the cells rather than on the particles themselves. Spatial hashing assigns particles to cells or 'bins' based on a hash of particle coordinates. The numerical expense of such an algorithm is $O(N)$ where N is particle number [15, 16]. A variety of programmatic implementations of hash cell methods have been used (for example linked list). In this work a dictionary hash table is used where a generic list of particles is stored for each cell and indexed based on an integer key unique to that cell, i.e. $key = (k \times ny + j) \times nx + i$ where nx an ny are the total number of cells in the x and y dimensions and i, j and k are the integer cell coordinates in the x, y and z dimensions.

The cells also provide a means for defining task packages to the cores. By assigning a number of cells to each core we assure "task orthogonality" in that each core is operating on different set of particles. Traditional software applications for shared memory parallel architectures have utilized locks to avoid thread contention. We note that a core may "read" the memory of particles belonging to surrounding cells but may not update them. We execute a single loop in which global memory for both previous and current field values (v_{t_n} and $v_{t_{n+1}}$) is stored for each particle. Gradient terms can then be calculated as functions of values in previous memory, while updates are written to the current value memory, in the same loop. In parallel, minimizing the frequency of so called synchronization points has advantages for performance and we utilize a "rolling memory algorithm" that allows such previous and updated terms to be maintained without needing to replace the former with the latter at the end of each step and thus, ensuring a single synchronization per step.

In a typical SPH simulator, two operations must be done on particles in each cell per step, separated by a synchronization, the first to determine the particle number density of each particle, and the second to perform the field variable updates. Interacting particles must be known for each of these two stages. Using a standard SPH formulation, the performance of two structure variations can be compared in a case study. In structure A, interacting particles are determined in the first stage for all cells and stored in a global list for use in the second. In structure B, interacting particles are determined as needed in each stage for each cell, i.e. twice per cell per time step. Recalculation of interacting particles means they need only be kept in a local thread list that is overwritten with each newly dispatched cell. While differences in execution memory are to be expected of the two code versions, the differences in execution time are more surprising. For low core counts (< 10 cores), as would be expected, the single search variant (A) solves more quickly than the double (B) due to less computations. After this point, however, the double search (B) is shown to provide marked improvements in speed over (A) (up to 50%). This can be attributed to better cache blocking of the second approach and the significantly smaller amount of data experiencing latency when being loaded from RAM to cache. The fact that such performance gains only manifest when more than 10 cores

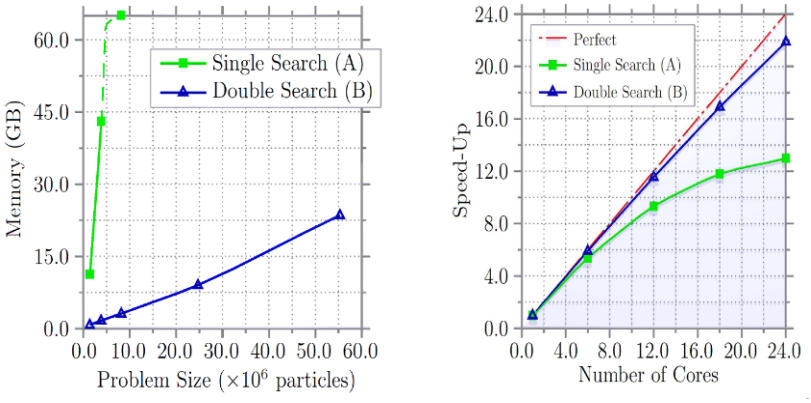

Fig. 14 (a) Memory vs. number of particles, (b) speed-up vs. number of cores

are used, suggests that for less than 10 cores, RAM pipeline bandwidth is sufficient to handle a global interaction list.

5 Conclusions

The verification of SPH as an accurate analysis tool for single-phase flows has been detailed and its extension to multi-phase flows and complex pore geometries demonstrated. The ability of SPH to simulate multiple fluid phases with accurate expression of surface tension and interfacial properties such as wettability and contact angle, make the method a powerful numerical tool for geo-numerics problems. Since flow in reservoir rock typically occurs in the range of low Reynolds number, the enforcement of no-slip boundary conditions is an important factor in simulation. Using the no-slip boundary conditions we show that SPH can handle the degree of complexity of boundary surfaces characteristic of an actual permeable rock sample.

We present a parallel numerical simulation framework, which allows parallel implementation of a wide range of numerical methods in a multi-core, shared memory, environment. We use a novel domain decomposition methodology that takes optimal advantage of the shared memory architecture and allows for dynamic load balancing of the cores.

References

1. Tiab, D., Donaldson, E.C., *Petrophysics: Theory and Practice of Measuring Reservoir Rock and Fluid Transport Properties*, 2nd ed., Elsevier, San Diego, CA, 2003.
2. Johnson, E.F., Bossler, D.P., Naumann, V.O., Calculation of relative permeability from displacement experiments, petroleum transactions. *AIME* 216:370–372.8, 1959.

3. Taber, J.J., Dynamic and static forces required to remove a discontinuous oil phase from porous media containing both oil and water. *SPE Journal* 9:3–12, 1969.
4. Jones, S.C., Roszelle, W.O., Graphical techniques for determining relative permeability from displacement experiments. *Journal of Petroleum Technology* 30:807–817, 1978.
5. Kerig, P.D., Watson, A.T., Relative-permeability estimation from displacement experiments: An error analysis. *SPE Reservoir Engineering* 1:175–182, 1986.
6. Teige, G.M.G., Hermanrud, C., Thomas, W.H., Wilson, O.B., Nordgard Bolas, H.M., Capillary resistance and trapping of hydrocarbons: A laboratory experiment. *Petroleum Geoscience* 11:125–129, 2005.
7. Roberts, J.N., Schwartz, L.M., Grain consolidation and electrical conductivity in porous media. *Physical Review B* 31:5990–5997, 1985.
8. Schwartz, L.M., Martys, N., Bentz, D.P., Garboczi, E.J., Torquato, S., Cross-property relations and permeability estimation in model porous media. *Physical Review E* 48:4584–4591, 1993.
9. Adler, P.M., Jacquin, C.G., Quiblier, J.A., Flow in simulated porous media. *International Journal of Multiphase Flow* 16:691–712, 1990.
10. Adler, P.M., Jacquin, C.G., Thovert, J.F., The formation factor of reconstructed porous media. *Water Resources Research* 28:1571–1576, 1992.
11. Hazlett, R.D., Statistical characterization and stochastic modeling of pore networks in relation to fluid flow. *Mathematical Geology* 29:801–822, 1997.
12. Yeong, C.L.Y., Torquato, S., Reconstructing random media. *Physical Review E* 57:495–506, 1998.
13. Yeong, C.L.Y., Torquato, S., Reconstructing random media. II. Three-dimensional media from two-dimensional cuts. *Physical Review E* 58:224–233, 1998.
14. Flannery, B.P., Deckman, H.W., Roberge, W.G., D'Amico, K.L., Three-dimensional x-ray microtomography. *Science* 237:1439–1444, 1987.
15. Dunsmuir, J.H., Ferguson, S.R., D'Amico, K.L., Stokes, J.P., X-ray microtomography: A new tool for the characterization of porous media. In: *Proceedings of the SPE Annual Technical Conference and Exhibition*, Society of Petroleum Engineers, pp. 22860-MS, 1991.
16. Spanne, P., Thovert, J.F., Jacquin, C.J., Lindquist, W.B., Jones, K.W., Adler, P.M., Synchrotron computed microtomography of porous media: Topology and transports. *Physical Review Letters* 73:2001–2004, 1994.
17. Coles, M.E., Hazlett, R.D., Spanne, P., Soll, W.E., Muegge, E.L., Jones, K.W., Pore level imaging of fluid transport using synchrotron x-ray microtomography. *Journal of Petroleum Science and Engineering* 19:55–63, 1998.
18. Schwartz, L.M., Auzerais, F., Dunsmuir, J., Martys, N., Bentz, D.P., Torquato, S., Transport and diffusion in three-dimensional composite media. *Physics A* 207:28–36, 1994.
19. Arns, C.H., Sheppard, A.P., Sok, R.M., Knackstedt, M.A., NMR petrophysical predictions on digitized core images. In: *SPWLA 46th Annual Logging Symposium*, Society of Petrophysicists and Well Log Analysts, p. MMM, 2005.
20. Arns, C.H., Sheppard, A.P., Saadatfar, M., Knackstedt, M.A., Prediction of permeability from NMR response: Surface relaxivity heterogeneity. In: *SPWLA 47th Annual Logging Symposium*, Society of Petrophysicists and Well Log Analysts, p. GG, 2006.
21. Auzerais, F.M., Dunsmuir, J., Ferréol, B.B., Martys, N., Olson, J., Ramakrishnan, T.S., Rothman, D.H., Schwartz, L.M., Transport in sandstone: A study based on three dimensional microtomography. *Geophysical Research Letters* 23:705–708, 1996.
22. Ryu, S., Zhao, W., Leu, G., Singer, P.M., Cho, H.J., Keehm, Y., Numerical modeling of complex porous media for borehole applications. ArXiv e-prints. [Online] Available: http://adsabs.harvard.edu/abs/2009arXiv0908.1962R, 2009.
23. Zhan, X., Schwartz, L., Morgan, D., Smith, W., Toksöz, N., Numerical modeling of transport properties and comparison to laboratory measurements. Technical Report, Massachusetts Institute of Technology. [Online] Available: http://www-eaps.mit.edu/erl/Zhan_2008_final.pdf, 2008.
24. Arns, C.H., Knackstedt, M.A., Pinczewski, W.V., Garboczi, E.J., Computation of linear elastic properties from microtomographic images: Methodology and agreement between theory and experiment *Geophysics* 67:1396–1405, 2002.

25. Knackstedt, M.A., Arns, C.H., Sok, R.M., Sheppard, A.P., 3D pore scale characterization of carbonate core: Relating pore types and interconnectivity to petrophysical and multiphase flow properties. In: *International Petroleum Technology Conference*, pp. 11775-MS.
26. Knackstedt, M.A., Arns, C.H., Sheppard, A.P., Senden, T.J., Sok, R.M., Cinar, Y., Pinczewski, W.V., Ioannidis, M., Padhy, G.S., Archie's exponents in complex lithologies derived from 3D digital core analysis. In: *SPWLA 48th Annual Logging Symposium*, Society of Petrophysicists and Well Log Analysts, p. UU, 2007.
27. Zhao, W., Picard, G., Leu, G., Singer, P.M., Characterization of single-phase flow through carbonate rocks: Quantitative comparison of NMR flow propagator measurements with a realistic pore network model. *Transport in Porous Media* 81:305–315, 2010.
28. Blunt, M.J., Jackson, M.D., Piri, M., Valvatne, P.H., Detailed physics, predictive capabilities and macroscopic consequences for pore-network models of multiphase flow. *Advances in Water Resources* 25:1069–1089, 2002.
29. Sok, R.M., Arns, C.H., Knackstedt, M.A., Senden, T.J., Sheppard, A.P., Averdunk, H., Pinczewski, W.V., Okabe, H., Estimation of petrophysical parameters from 3D images of carbonate core. In: *SPWLA Middle East Regional Symposium*, Society of Petrophysicists and Well Log Analysts, 2007.
30. Ferréol, B., Rothman, D.H., Lattice-Boltzmann simulations of flow through Fontainebleau sandstone. *Transport in Porous Media* 20:3–20, 1995.
31. Kameda, A., Dvorkin, J., Keehm, Y., Nur, A., Bosl, W., Permeability-porosity transforms from small sandstone fragments. *Geophysics* 71:N11–N19, 2006.
32. Hazlett, R.D., Coles, M.E., Jones, K.W., Andrews, B., Dowd, B., Siddons, P., Peskin, A., Developments in synchrotron X-ray microtomography for application to flow in porous media. In: *Proceedings of the 1996 Annual Technical Conference of the Society of Core Analysists*, p. 9630, 1996.
33. Hazlett, R.D., Chen, S.Y., Soll, W.E., Wettability and rate effects on immiscible displacement: Lattice Boltzmann simulation in microtomographic images of reservoir rocks. *Journal of Petroleum Science and Engineering* 20:167–175, 1998.
34. Chen, S., Doolen, G.D., Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics* 30:329–364, 1998.
35. Zheng, H.W., Shu, C., Chew, Y.T., A lattice Boltzmann model for multiphase flows with large density ratio. *Journal of Computational Physics* 218:353–371, 2006.
36. Huang, J.J., Shu, C., Chew, Y.T., Lattice Boltzmann study of droplet motion inside a grooved channel. *Physics of Fluids* 21:022103, 2009.
37. Lucy, L.B., A numerical approach to the testing of the fusion hypothesis. *Astronomical Journal* 82:1013–1024, 1977.
38. Gingold, R.A., Monaghan, J.J., Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181:375–389, 1977.
39. Monaghan, J.J., Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30:543–574, 1992.
40. Hoover, W.G., Isomorphism linking smooth particles and embedded atoms. *Physica A* 260:244–254, doi: 10.1016/S0378-4371(98)00357-4, 1998.
41. Morris, J.P., Fox, P.J., Zhu, Y., Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics* 136:214–226, doi: 10.1006/jcph.1997.5776, 1997.
42. Monaghan, J.J., Smoothed particle hydrodynamics. *Reports on Progress in Physics* 68:1703–1759, 2005.
43. Tartakovsky, A.M., Meakin, P., Pore scale modeling of immiscible and miscible fluid flows using smoothed particle hydrodynamics. *Advances in Water Resources* 29:1464–1478, 2006.
44. Liu, G.R., Liu, M.B., *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*, World Scientific, Singapore, 2007.
45. Bui, H.H., Fukagawa, R., Sako, K., Ohno, S., Lagrangian meshfree particles method (SPH) for large deformation and failure flows of geomaterial using elastic-plastic soil constitutive model. *International Journal for Numerical and Analytical Methods in Geomechanics* 32:1537–1570, 2008.

46. Randles, P.W., Libersky, L.D., Smoothed particle hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering* 139:375–408, 1996.
47. Liu, M., Meakin, P., Huang, H., Dissipative particle dynamics simulation of pore-scale multiphase fluid flow. *Water Resources Research* 43:W04411, 2007.
48. Liu, M., Meakin, P., Huang, H., Dissipative particle dynamics simulations of multiphase fluid flow in microchannels and microchannel networks. *Physics of Fluids* 19:033302, 2007.
49. Tartakovsky, A.M., Meakin, P., Simulation of unsaturated flow in complex fractures using smoothed particle hydrodynamics. *Vandose Zone Journal* 4:848–855, 2005.
50. Tartakovsky, A.M., Meakin, P., Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E* 72:1–9, 2005.
51. Hu, X.Y., Adams, N.A., A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics* 213:844–861, 2006.
52. Tartakovsky, A.M., Meakin, P., Ward, A.L., Smoothed particle hydrodynamics model of non-aqueous phase liquid flow and dissolution. *Transport in Porous Media* 76:11–34, 2009.
53. Tartakovsky, A.M., Meakin, P., A smoothed particle hydrodynamics model for miscible flow in three-dimensional fractures and the two-dimensional Rayleigh–Taylor instability. *Journal of Computational Physics* 207:610–624, 2005.
54. Tartakovsky, A.M., Meakin, P., Scheibe, T.D., West, R.M.E., Simulations of reactive transport and precipitation with smoothed particle hydrodynamics. *Journal of Computational Physics* 222:654–672, 2007.
55. Monaghan, J.J., Kocharyan, A., SPH simulations of multi-phase flow. *Computer Physics Communications* 87:225–235, 1995.
56. Cleary, P.W., Modelling confined multi-material heat and mass flows using SPH. *Applied Mathematical Modelling* 22:981–993, 1999.
57. Jiang, F., Sousa, A.C.M., SPH numerical modeling for ballistic-diffusive heat conduction. *Numerical Heat Transfer, Part B. Fundamentals* 50:499–515, 2006.
58. Rook, R., Yildiz, M., Dost, S., Modeling transient heat transfer using SPH and implicit time integration, *Numerical Heat Transfer, Part B. Fundamentals* 51:1–23, 2007.
59. Price, D.J., Modelling discontinuities and Kelvin–Helmholtz instabilities in SPH. *Journal of Computational Physics* 227:10040–10057, 2008.
60. Dolag, K., Bartelmann, M., Lesch, H., SPH simulations of magnetic fields in galaxy clusters. *Astronomy and Astrophysics* 348:351–363, 1999.
61. Borve, S., Omang, M., Trulsen, J., Regularized smoothed particle hydrodynamics: A new approach to simulating magneto-hydrodynamic shocks. *The Astrophysical Journal* 561:82–93, 2001.
62. Holmes, D.W., Williams, J.R., Tilke, P., An events based algorithm for distributing concurrent tasks on multi-core architectures. *Computer Physics Communications* 181:341–354, 2010.
63. Gropp, W., Lusk, E., Skjellum, A., *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press, Cambridge, 1999.
64. Eadline, D., MPI on multicore, an OpenMP alternative? *Linux Magazine*. [Online] Available: http://www.linux-mag.com/id/4608, 2007.
65. Leiserson, C.E., Mirman, I.B., *How to Survive the Multicore Software Revolution (or at Least Survive the Hype)*, Cilk Arts, Cambridge, 2008.
66. Strikwerda, J.C., *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brooks/Cole, Pacific Grove, CA, 1989.
67. Chen, S., Doolen, G.D., Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics* 30:329–364, 1998.
68. Zienkiewicz, O.C., Taylor, R.L, *The Finite Element Method*, 4th ed., McGraw-Hill, London, 1991.
69. Liu, H., Shi, P., Meshfree particle method. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)*, Vol. 1, IEEE Computer Society, Los Alamitos, CA, USA, pp. 289–296, 2003.
70. Koplik, J., Banavar, J.R., Willemsen, J.F., Molecular dynamics of fluid flow at solid surfaces, *Physics of Fluids A. Fluid Dynamics* 1:781–794, 1989.

71. Cundall, P.A., Strack, O.D.L., A discrete numerical model for granular assemblies. *Geotechnique* 29:47–65, 1979.
72. Williams, J.R., O'Connor, R., Discrete element simulation and the contact problem. *Archives of Computational Methods in Engineering* 6(4):279–304, 1999.
73. Williams, J.R., Perkins, E., Cook, B.K., A contact algorithm for partitioning N arbitrary sized objects, *International Journal of Computer Aided Methods in Engineering – Engineering Computations* 21(2–4):235–248, 2004.
74. Keaveny, E.E., Pivkin, I.V., Maxey, M., Karniadakis, G.E., A comparative study between dissipative particle dynamics and molecular dynamics for simple- and complex-geometry flows. *The Journal of Chemical Physics* 123:104107, 2005.
75. Hasimoto, H., On the periodic fundamental solutions of the Stokes equations and their application to viscous flow past a cubic array of spheres. *Journal of Fluid Mechanics* 5(2):317–328, doi: 10.1017/S0022112059000222, 1959.
76. Zick, A.A., Homsy, G.M., Stokes flow through periodic arrays of spheres. *Journal of Fluid Mechanics* 115:13–26, doi: 10.1017/S0022112082000627, 1982.
77. Sangani, A.S., Acrivos, A., Slow flow through a periodic arrays of spheres. *International Journal of Multiphase Flow* 8(4):343–360, doi: 10.1016/0301-9322(82)90047-7, 1982.
78. Larson, R.E., Higdon, J.J.L., A periodic grain consolidation model of porous media. *Physics of Fluids A* 1(1):38–46, doi: 10.1063/1.857545, 1989.
79. Schwartz, L.M., Martys, N., Bentz, D.P., Garboczi, E.J., Torquato, S., Cross-property relations and permeability estimation in model porous media. *Physical Review E* 48(6):4584–4591, doi: 10.1103/PhysRevE.48.4584, 1993.
80. Holmes, D.W., Williams, J.R., Tilke, P., Smooth particle hydrodynamics simulations of low Reynolds number flows through porous media. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2010.