Accepted Manuscript

The transport of nanoparticles in subsurface with fractured, anisotropic porous media: Numerical simulations and parallelization

Meng-Huo Chen, Amgad Salama, Shuyu Sun

PII: S0377-0427(18)30564-8
DOI: https://doi.org/10.1016/j.cam.2018.09.019
Reference: CAM 11914

To appear in: *Journal of Computational and Applied Mathematics*

Received date : 17 November 2017
Revised date : 3 September 2018

Please cite this article as: M.-H. Chen, et al., The transport of nanoparticles in subsurface with fractured, anisotropic porous media: Numerical simulations and parallelization, *Journal of Computational and Applied Mathematics* (2018), https://doi.org/10.1016/j.cam.2018.09.019

The transport of nanoparticles in subsurface with fractured, anisotropic porous media: numerical simulations and parallelization

Meng-Huo Chen[a], Amgad Salama[b], Shuyu Sun[c,*]

[a]School of Computing, University of Leeds, EC Stoner Building, Leeds LS2 9JT, UK
[b]Faculty of Engineering, University of Regina, Regina, Sk., Canada
[c]Physical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, KSA

Abstract

The flow of fluids through fractured porous media has been an important topic in the research of subsurface flow. The several orders of magnitude in size between the fractures and the rock matrix causes difficulties for simulating such flow scenario. The fluid velocities in fractures are also several orders of magnitude higher than that in the rock matrix due to high permeability and porosity. If there exists pollutant such as nanoparticles in the fluids, the pollutant may be transported rapidly and the rock matrix's properties near the fractures are hence changed. In this research, we simulate the transport phenomena of nanoparticles in the fluid flow through fractured porous media. The permeability fields which contain different anisotropy angles are considered in the simulation. Fractures are represented explicitly by volumetric grid cells and the numerical algorithm is parallelized in order to reduce the simulation time. We investigate the effect of the appearance of fractures and rotated anisotropy on the transport of nanoparticles, particles deposition, entrapment and detachment. The results show that flow directions are affected by the direction of anisotropy and the transport of nanoparticles in the fractures is significantly faster than that in rock matrix due to high fluid velocities. The direction of anisotropy distorted the pressure field and changed the fluid flow directions, which determined the time needed for the pollutant front to reach the fractures. The parallel efficiency of the overall algorithm is also discussed and the experimental results show that it is deeply affected by the performance of the multigrid solver.

Keywords: Multipoint flux approximation, nanoparticle transport, algebraic multigrid, parallel computation

1. Introduction

Nanoparticles are particles that are between 1 and 100 nanometers in size. These materials are the basic building block of nanotechnology and are intensively used in electronic, biomedical, cosmetic, automotive products [1] and others. Recently, nanomaterials have been proposed to be used in petroleum-related industries. There have been ideas to using nanoparticles in various activities such as reservoir exploration, drilling and completion, production and improved oil recovery as well as refinery [2]. The growing demand on nanomaterials has increased their production and consequently their inevitable discharged into the environment. Nanoparticles have several favourable properties including their effectiveness in surface phenomena because of the extremely larger surface area they provide. Some nanomaterials provide bactericidal and antimicrobial effect such as silver, and zinc. They can, therefore, serve as a versatile bactericide with a wide range of uses such as in fabrics, filtration membranes, and surgical instruments [3]. Due to their high surface to volume ratio, nanoparticles become highly reactive or catalytic and may impose a negative impact on the environment. It is noticed that even exposure of living organism to ZnO nanoparticles may be toxic to many species including plants, animals and useful bacteria [4]. All these situations raise the concern about the fate of these materials in their local environment and motivate the research on the transport of nanoparticle in the environment. Several research works have been conducted to study the transport of nanoparticles in subsurface groundwater reservoirs. Most of the research works conducted on the transport of nanoparticles in the subsurface have considered only isotropic and homogeneous reservoirs. Recently, Salama et al. [2; 5; 6; 7] considered more elaborate scenarios to account for anisotropy in permeability field. As transported nanoparticles deposit, they deteriorate both the open space for the flow and reduce the permeability. The results of

*Corresponding author

numerical simulations showed that the pressure, velocities and porosity fields are distorted by the direction of the anisotropy. With large anisotropy angle, the deposition of the nanomaterials seemed to be severe and the minimum permeability of the domain is reduced at a higher rate. In all the aforementioned work, the permeability fields were generally uniform and didn't consider heterogeneity caused, for example, by the existence of fractures in the porous media.

Fractured porous media are in fact very common in consolidated rock systems with the fractures forming networks of interconnected and/or isolated sets of fractures. Most natural porous media contain fractures at various scales and they have strong influence on the flow system because most of the flow takes place along them. Numerical simulation of the flow in fractured porous media is a challenge task on account to the various length scales and the fluid velocities within the rock matrix and inside the fractures, which differ by several orders of magnitude. A higher resolution mesh is required in order to resolve the steep variations induced by the flow field and mass transfer between rock matrix and the fractures accurately.

In general, there are two major approaches to simulate the flow and transport in the fracture-matrix system based on the fracture intensity. In cases when the fracture intensity is relatively small such that it is feasible to consider every single fracture as an individual entity, discrete fracture models (DFM) have been developed. In these models, the fractures are considered as either part of the domain that are assigned much higher permeability or as a lower dimensionality region in the study domain. When the intensity of the fractures, on the other hand, are higher such that considering individual fractures becomes very expensive or infeasible, a multi-continuum description may be considered. In this approach, all the complex features of the network of fractures are homogenised and a dual- continuum is constructed, one for the matrix and one for the fractures. Each of the two continua has their own porosity and/or permeability. It is, therefore, called dual porosity/permeability models. In this model, most of the fluid storage is considered in the matrix with the large-scale flow occurring through the fracture continuum. The interactions between the flows in the matrix and the fracture continua in dual continua models is described by a mass transfer function. Such mass transfer functions are obtained through analytical expressions or using empiricism involving some ad-hoc shape factors. While this may be easy to obtain for simple cases, other cases involving, for example, compositional and compressible flows, may be difficult to establish such shape factors. The interesting thing about the dual continua models, however, is that they require less computational efforts compared with discrete fracture models. In the discrete fracture modelling approach, the flow problem is explicitly modelled in each fracture and in the matrix using Darcy's law. In the grid meshing, each fracture is represented explicitly and individually using highly resolved Cartesian or unstructured grids, hence the expensive computational costs.

In this work, we study the problem of flow and transport of nanoparticles in anisotropic porous medium domain containing a set of discrete fractures. Each fracture is represented by volumetric cells. Several scenarios are considered such as different angles of anisotropy and the location of the fractures. In order to handle the anisotropy which does not align with the coordinate axes, the spatial variables are discretized using the multipoint flux approximation (MPFA) as in [1]. The algorithm is implemented in parallel using FORTRAN and the linear system arising from the pressure equation is solved by the aggregation-based algebraic multigrid solver in [8; 9]. The rest of the paper is organized as follows: In Section 2 we describe the transport models of nanoparticles, where the governing equations for the transport and fluid flow are listed and reviewed. In Section 3 we discuss the numerical algorithm for solving the governing equations, including some issues regarding the linear system solver and the parallel implementation and processing of the numerical scheme. Section 4 describes the setting of the numerical examples and the simulation results and observations are discussed in the final section.

2. Transport models of nanoparticles

Flow and transport in porous media are, usually, described within the framework of the continuum hypothesis in which variables pertinent to porous media exist everywhere and behave in a continuous fashion in space and in time. This framework allowed the description of several phenomena in porous media in the form of partial differential equations. One of these equations is that which describes the movement of materials with the flow in porous media. The what is called advection-dispersion equation is a second order partial differential equation

which takes the form:

$$\frac{\partial \phi C}{\partial t} + \nabla \cdot (\mathbf{u}C - \phi \mathbf{D} \nabla C) = q_c + R \tag{1}$$

where ϕ is the porosity, C is the concentration per unit volume of the fluid, $\mathbf{u}$ is Darcy velocity, $\mathbf{D}$ is the dispersion tensor, q_c is a source/sink term and R is a generation/depletion term. This equation describes wide spectrum of phenomena in porous media involving chemical reactions, adsorption and desorption, deposition and remobilization and several other interesting processes. While the terms in the left-hand side exist in almost all processes, the terms in the right-hand side are exclusively what differentiate the different physical and chemical processes. In this work we are mainly interested in the transport of nanomaterials, it is important to understand the physics involved and how they may be incorporated in the above transport equation.

2.1. Filtration theory

When nanoparticles are transported with water in porous media in the form of colloids dispersed in the water phase, they interact with each other due to collisions. If such collisions result in the aggregation of the nanoparticles, deposition occur. If, on the other hand, the collisions among nanoparticles result in more dispersion, then stabilized colloidal system is obtained. According to the DLVO theory, between every two nanoparticles there exist attractions due to the Van der Waals forces and repulsions due to electrostatic effects. The overall behaviour of the colloidal system depends on the resultant effect of these two forces. These forces in general may lead to deposition of nanoparticles. Other forces may exist between nanoparticles and the porous skeleton which derives adsorption/desorption processes. Hydrodynamic effects may also exist due to the flow velocity and could lead to remobilization of deposited nanoparticles. Several models exist that describe the transport of nanoparticles in porous media ([10; 11; 12; 13; 14; 15; 16; 17]). As discussed earlier the modelling of the generation term is important so that the transport equation can be solved. Gargiulo et al.[14] and Goldberg et al.[15] have recently reviewed a number of these models. The mass balance of nanoparticles may be described as

$$\frac{\partial \phi C + S}{\partial t} + \nabla \cdot (\mathbf{u}C - \phi \mathbf{D} \nabla C) = q_c + R \tag{2}$$

where C is the concentration of nanoparticles in the water per unit volume of the water, ϕ is the porosity, q_c is a source/sink term, and S is the concentration of deposited nanoparticles per unit volume of the porous medium. Within the framework of the filtration theory, the rate of deposition of nanomaterials may be proposed to depend on the flux of nanomaterials. That is

$$\frac{\partial S}{\partial t} = \gamma |J| \tag{3}$$

where γ is a proportionality coefficient (1/L) and J is the advection flux of nanoparticles. This equation can, therefore, be written as

$$\frac{\partial S}{\partial t} = \gamma_d |u_p| C - \gamma_r |u_p - u_r| S \tag{4}$$

where u_p is Darcy velocity, u_r is the critical velocity to entrain particles ([18]), γ_d and γ_r are coefficients for deposition and remobilization, respectively. The complete model and the governing equations of flow and transport are given in the next section.

2.2. The complete model

In porous media flows, the permeability is a measure of the ability of the medium to conduct fluids through it. Most carbonate reservoirs are naturally fractured and the permeability in fractures is generally several orders of magnitude larger than that in the rock matrix. On the other hand, anisotropy can be observed in almost all subsurface formation due to the several physiochemical and mechanical processes that took place over the longer geologic time scale. Cullen et al. [19] investigated the transport of carbon nanoparticles in a two-dimensional heterogeneous anisotropic porous medium rectangular domain. The principle direction of the anisotropy is aligned with the coordinate axes. In this case, two points flux approximation (TPFA) is sufficient to solve the system.

However, when the anisotropy direction is not aligned with the coordinate axes, TPFA fails to account for the fact that pressure gradient in one direction can cause flow in the other directions. In the case of the presence of full permeability tensor, the more involved multipoint flux approximation (MPFA) methods (as will be explained later) is needed in order to obtain the correct discretization as has been done by Salama et al. [1]. In this work, we extend this framework to account for the effect of a set of discrete fractures on the transport of nanoparticles in anisotropic medium.

The governing equations of fluid flow in porous media are given by the mass conservation equation and the Darcys law. In the absence of source/sink term and in the case of incompressible fluids, the principle of mass conservation assumes that the mass inflow and outflow are equal when fluid flow crosses a certain region. Therefore, we have

$$\nabla \cdot \mathbf{u} = 0 \tag{5}$$

where $\mathbf{u}$ is the velocity obeying the Darcy's law:

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu}\left(\nabla p - \rho \mathbf{g}\right). \tag{6}$$

In equation (6), $\mathbf{K}$ is the permeability tensor, μ is the fluid viscosity, ρ is the fluid density and $\mathbf{g}$ is the gravity. Combining Eqs. (5) and (6), an equation in the pressure only is obtained, therefore

$$\nabla \cdot \frac{\mathbf{K}}{\mu}\left(\nabla p - \rho \mathbf{g}\right) = 0 \tag{7}$$

The transport of nanoparticles is described by Eq. 2 in which the dispersion tensor $\mathbf{D}$ is calculated as

$$\mathbf{D} = \mathbf{D}^{\mathrm{disp}} + \mathbf{D}_i^{\mathrm{Br}}, \tag{8}$$

where

$$\mathbf{D}^{\mathrm{disp}} = d_{t,w}|\mathbf{u}|\mathbf{I} + \left(d_{l,w} - d_{c,w}\right)\frac{\mathbf{uu}}{|\mathbf{u}|} \tag{9}$$

and

$$\mathbf{D}_i^{\mathrm{Br}} = \frac{k_B T}{3\pi\mu d_{p,i}}. \tag{10}$$

For simplicity we consider single size nanoparticles. In this work, we use the model in Gruesbeck and Collin [20] which describes the process of deposition as a consequence of two interacting mechanisms, one at the pore surface and one at pore throats, such that

$$R_i = \frac{\partial s_{1i}}{\partial t} + \frac{\partial s_{2i}}{\partial t}, \tag{11}$$

where

$$\frac{\partial s_{1i}}{\partial t} = \begin{cases} \gamma_{di}|\mathbf{u}|c_i, & |\mathbf{u}| \le \mathrm{u}_c \\ \gamma_{di}|\mathbf{u}|c_i - \gamma_{ei}(|\mathbf{u}| - \mathrm{u}_{ci})s_{1i}, & |\mathbf{u}| > \mathrm{u}_c \end{cases}. \tag{12}$$

and

$$\frac{\partial s_{2i}}{\partial t} = \gamma_{pt,i}|\mathbf{u}|c. \tag{13}$$

In Eq. (12) γ_{di} and γ_{ei} are rate coefficients for surface retention and entrainment of nanoparticles in interval i, respectively. u_c is the critical value for the magnitude of the velocity. With equations (12) and (13), the porosity and permeability are updated:

$$\phi = \phi_0 - \sum \delta\phi_i = \phi_0 - (s_{1i} + s_{2i}) \tag{14}$$

$$\mathbf{K} = \mathbf{K}_0 \Big[(1 - f_r) k_{f_r} + f_r \frac{\phi}{\phi_0} \Big]^l. \tag{15}$$

From equations (12) - (15) s_1 and s_2 are the concentration of nanoparticles at the surfaces of the grains and entrapped in pore throats, respectively. Furthermore, $\mathbf{K}_0$ and ϕ_0 are initial permeability and porosity, k_{f_r} is a constant for fluid seepage allowed by plugged pores. Finally, f_r is the fraction of the original cross sectional area open to flow and is calculated as

$$f_r = 1 - \gamma_{f_r} s_2, \tag{16}$$

where γ_{f_r} is a constant.

Note that as the test problems of this work, the above simple model serves our purpose for assessing issues regarding numerical algorithm for solving flow/transport problem in anisotropic porous media with fractures using parallel processing. In future work we will assess similar problems on a yet more complicated example, such as nanoparticle transport in shale reservoirs ([21]), which involves two-phase (aqueous and gaseous) flow. Furthermore, this work can be extend 3D flow/transport problems ([22]).

3. Numerical algorithm

3.1. Multipoint flux approximation

As mentioned in the previous subsection, we use the multipoint flux approximation (MPFA) method to discretize the spatial variables on staggered grid in order to accurately describe the flow behaviour in the porous media whose permeability is anisotropic, in which case the permeability tensor is full. For staggered grid in 2-D space, pressure values are defined at the centre of the cells and velocity components are located on edges (see Fig. 1).

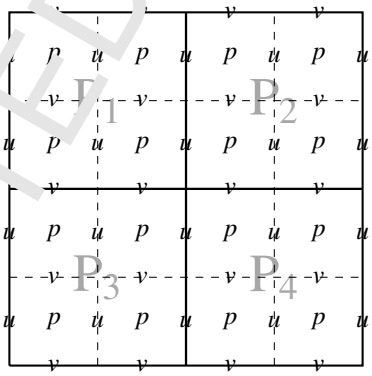
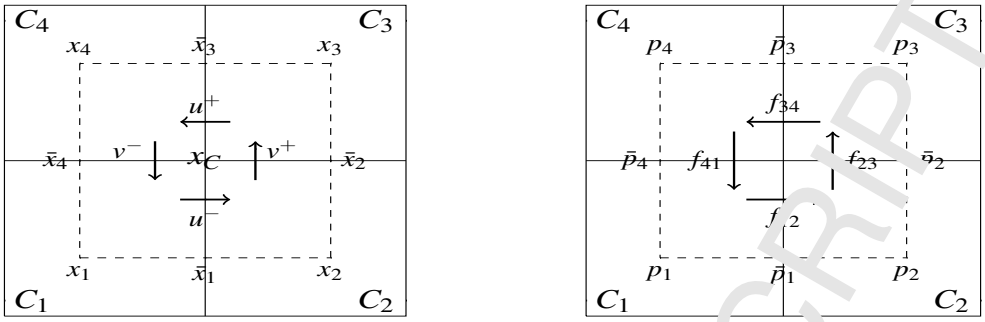


Figure 1: Staggered grid divided by 4 processes (cores).

Consider the four quadrilateral cells C_l, $l = 1,...,4$ in Fig. 2a with a common vertex at x_C. MPFA method builds the interaction region (dashed line in Fig. 2) that is centred on the four adjacent cells and there are four fluxes that need to be calculated in each interaction region. Denote the cell centres by x_k, and the edge midpoints by $\bar{x}_k$, where $k = 1,2,3,4$. Lines are drawn between the cell centres and the midpoints of the edges (shown as dashed lines in the figure). These lines bound an area around each vertex which is called an interaction volume. As seen from the figure, there are four half cell edges (solid lines) in the interaction volume. To discretize the PDEs (5) - (6) using MPFA, we first compute the flux of potential. The flux of a phase α through half cell edge S in an interaction volume can be computed by

$$f^\alpha = -\int_S (\mathbf{K} \nabla \Phi^\alpha) \cdot \hat{\mathbf{n}} \, dS \tag{17}$$



(a) velocities and positions

(b) Pressure and fluxes

Figure 2: Cells and interaction volumes

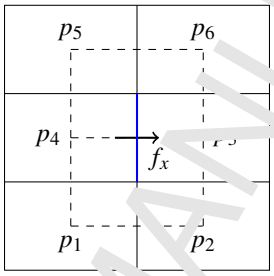


Figure 3: Six pressure values contribute to flux calculations in multipoint point flux approximation.

where ϕ is the phase potential (which is pressure in our case), $\mathbf{K}$ is the transmissibility tensor (permeability in our example), and $\hat{\mathbf{n}}$ is the unit normal vector to the surface. For convenience we drop the phase index from now on and the above formula is reduced to

$$f = -\int_S (\mathbf{K}\nabla\Phi)\cdot\hat{\mathbf{n}}\,\mathrm{d}S \tag{18}$$

The integral on the right hand side of (18) can be approximated by

$$f \approx \sum_{l\in\Omega} t_l\Phi_l \tag{19}$$

where t_l and Φ_l are the transmissibility coefficient and the potential gradient at the centre of the cell-l, respectively. The fluxes are conserved locally by assuming the inflow and outflow fluxes are equal. From Fig. 2, the flux f_{12} is given by

$$\begin{aligned} f_{12} &\approx -t_1^{xx}(\bar{p}_1 - p_1) - t_1^{xy}(\bar{p}_4 - p_1) \\ &\approx -t_2^{xx}(p_2 - \bar{p}_1) - t_2^{xy}(\bar{p}_2 - p_2). \end{aligned} \tag{20}$$

The fluxes f_{23}, f_{34} and f_{41} are obtained in a similar fashion. Notice that these fluxes contribute only half of the flux calculations on the edges where they are located since the full flux calculations on an edge involve two adjacent interaction regions to be considered as expressed in equation (19). From each interaction region, we would obtain four systems of equations that need to be solved locally. Each full flux involves two interaction regions and hence six pressure variables in neighbouring cells (see Fig. 3). It is clear that the construction of MPFA stencil is quite complicated. The x-component of the flux, for example, may be expressed as

$$f_x = c_1p_1 + c_2p_2 + c_3p_3 + c_4p_4 + c_5p_5 + c_6p_6 \tag{21}$$

where the coefficients c_q, $q = 1,...,6$ in the above equation include both the discretization and the permeability information. The derivations of these coefficients require significant amount of computations. To resolve this

difficulty Sun et al. ([23; 24]) developed experimenting field method to reduce the cumber[illegible]e calculations. In experimenting field approach, the matrix coefficients are obtained by solving several local problems which involves only neighbouring variables. The linear systems arising from local problems are smaller and much easier to solve. The predefined experimenting pressure fields are designed and operated over the local problems such that the global matrix coefficients are obtained. The linear system associated with the local problem for each node (i, j) of the Cartesian grid shown in Fig. 2 is therefore

$$A[u^- \quad v^+ \quad u^+ \quad v^-]^T = B[p_1 \quad p_2 \quad p_3 \quad p_4]^T \tag{22}$$

$$A_{i,j} = \frac{1}{4}\begin{bmatrix} \frac{k_{yy}^{C_1}}{\det(C_1)}A^{C_1} + \frac{k_{yy}^{C_2}}{\det(C_2)}A^{C_2} & -\frac{k_{xy}^{C_1}}{\det(C_2)}A^{C_2} & [illegible] & -\frac{k_{xy}^{C_1}}{\det(C_1)}A^{C_1} \\ -\frac{k_{xy}^{C_2}}{\det(C_2)}A^{C_2} & \frac{k_{xx}^{C_2}}{\det(C_2)}A^{C_2} + \frac{k_{xx}^{C_3}}{\det(C_3)}A^{C_3} & -\frac{k_{yy}^{C_3}}{\det(C_3)}A^{C_3} & 0 \\ 0 & -\frac{k_{xy}^{C_3}}{\det(C_3)}A^{E_3} & \frac{k_{yy}^{C_3}}{\det(C_3)}A^{C_3} + \frac{k_{yy}^{C_4}}{\det(C_4)}A^{C_4} & -\frac{k_{xy}^{C_3}}{\det(C_3)}A^{C_3} \\ -\frac{k_{xy}^{C_1}}{\det(C_1)}A^{C_1} & 0 & -\frac{k_{xy}^{C_4}}{\det(C_4)}A^{C_4} & \frac{k_{xx}^{C_1}}{\det(C_1)}A^{C_1} + \frac{k_{xx}^{C_4}}{\det(C_4)}A^{C_4} \end{bmatrix}, \tag{23}$$

and

$$B_{i,j} = \frac{1}{2}\begin{bmatrix} \Delta y_{C_1} & -\Delta y_{C_2} & 0 & 0 \\ 0 & \Delta x_{C_2} & -\Delta x_{C_3} & 0 \\ 0 & 0 & -\Delta y_{C_3} & \Delta y_{C_3} \\ \Delta x_{C_4} & [illegible] & 0 & -\Delta x_{C_4} \end{bmatrix}. \tag{24}$$

Here A^{C_i}, Δx_{C_i} and Δy_{C_i}, $i = 1, ..., 4$, are the area, the horizontal and vertical length of cell C_i, respectively, and $\det(C_i)$ is the determinant of the hydraulic conductivity tensor of each cell. From (23) - (24) we obtain a system of algebraic equations for solving the pressure variables

$$C\mathbf{p} = \mathbf{b} \tag{25}$$

where C is the matrix of coefficients, $\mathbf{p}$ is the vector of the unknown pressure field and $\mathbf{b}$ is the right hand side vector which includes the source term and the boundary condition. For simulations with high resolution, matrix C is sparse symmetric positive definite and the dimension is large. To solve a large sparse linear system, iterative methods are better options than direct methods (generally using Gauss eliminations). Furthermore, iterative methods are generally easier to parallelize. Among iterative methods, multigrid methods have been proved to be one of the most efficient algorithms to solve the elliptic type partial differential equations numerically. For solving the linear system in (25), we employ the aggregation-based algebraic multigrid (AGMG) method in [8; 9].

It is worth noting that multipoint flux mixed finite element (MFMFE) method is closely related to the MPFA method. In fact, one can view the O-version of the MPFA method (the one considered and presented in this work) as a special case of the multipoint flux mixed finite element (MFMFE) method with a certain quadrature rule. In this paper, we restrict our attention to MPFA for two reasons. Firstly, MPFA is more widely used than MFMFE in engineering applications, and secondly, aggregation-based algebraic multigrid is more natural when applying to MPFA as compared with MFMFE. A separate work (still in progress) will be devoted to aggregation-based algebraic multigrid for MFMFE in near future and it will appear in another paper.

3.2. Aggregation-based algebraic multigrid

Consider the linear system $Ax = b$ on a fine grid, where A represents the matrix and b the right-hand side vector, and x is the solution vector that one seeks. In iterative algorithms we let $x^{(i)}$ denote the approximate solution to the linear system at the ith iteration and decompose the corresponding error $e^{(i)} = x^{(i)} - x$ into its Fourier components. Classical relaxation schemes, such as weighted Jacobi or Gauss-Seidel methods, can quickly damp the high frequency components of the error. For this reason these methods are called smoothers (see step 1 in Fig. 4 where x_0^l, r^l and S_1^l are the input vector, residual and pre-smoothing operator, respectively). Notice that the superscript l in

Fig. 4 represents the multigird level. As the high frequency components are damped, the further smoothing iterations on this level hardly attenuate the low-frequency components. To solve this, the multigrid methods use coarse grid correction, that is, the low-frequency components of residual (step 2, Fig. 4) are restricted to a sufficiently small grid (step 3, Fig. 4, where R^l is the restriction operator). On the coarser level few (one or two) relaxations (step 1 at the next level, Fig. 4) can effectively reduce the error because the smoother components projected from the fine level appear more oscillatory. The smoothing-restriction procedure can be carried out until the coarsest grid is reached where the corresponding residual equation is inexpensive to solve directly by Gaussian eliminations (step 4 at the coarsest level, Fig. 4). The coarsest grid solution ($\tilde{x}_2^l$ at the coarsest level) is then interpolated back to the finer levels (step 5, Fig. 4, where P^l is the interpolation operator) where further relaxation sweeps (step 6, Fig. 4, where S_2^l is the post-smoothing operator) may be applied. Recursive application of the above procedure leads to a multigrid V-cycle implementation, as shown in Fig. 4. Note that in Fig 4 the approximation in step 4 becomes exact on the coarsest level where the system is directly solved.

1. Pre-smooth $\quad x_1^l = S_1^l(x_0, r^l)$
2. Residual $\quad r_1^l = r^l - A^l x_1^l$
3. Restriction $\quad \tilde{r}_1^l = R^l r_1^l$
4. Next level $\quad \tilde{A}^l \tilde{x}_2 \approx \tilde{r}_1^l$
5. Prolongation $\quad x_2^l = P^l \tilde{x}_2^l$
6. Correction $\quad x_3^l = x_1^l + x_2^l$
7. Post-smooth $\quad x_0^l = S_2^l(x_3^l, r^l)$

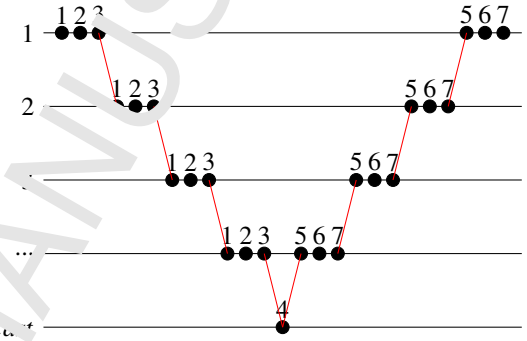


Figure 4: Multigrid algorithm (left) and illustration of V-cycle (right). S_1 , S_2^l, P^l and R^l are pre-smoothing, post-smoothing restricting and prolongation operators at level l, respectively.

For coarse grid construction in AGMG algorithm the (say N) unknowns are subdivided into (say N_c) disjoint small groups (aggregates) I_k, $k = 1,...,N_c$ which represent unknowns on next coarser level. In each aggregate, the prolongation values are assumed to be uniform, that is, functions are piecewise constant. Therefore the prolongation operator P is of the form

$$P_{ik} = \begin{cases} 1 & \text{if } i \in I_k, \\ 0 & \text{otherwise} \end{cases} \qquad (1 \le i \le N, 1 \le k \le N_c). \tag{26}$$

P is an $N \times N_c$ matrix with exactly one nonzero entry per row [9]. Consequently, the restriction operator is $R = P^T$ and the coarse grid matrix A_c is obtained by Galerkin formula

$$A_c = P^T A P \tag{27}$$

where the entry $(A_c)_{ij}$ is calculated by

$$(A_c)_{ij} = \sum_{k \in I_i} \sum_{l \in I_j} a_{kl}. \tag{28}$$

Piecewise constant prolongation may attenuate the convergence property of the algorithm and a remedy is proposed in [8], in which K-cycle instead of V-cycle is used in cycle strategy. In K-cycle the approximate solution $\tilde{x}_2$ in Fig. 4 is obtained by one or two multigrid preconditioned Krylov subspace iterations (preconditioned conjugate gradient or GMRES), where the multigrid preconditioner is the K-cycle implementation on the next coarser level. The K-cycle strategy of the AGMG is described in detail in [8] and [9], respectively.

As one of the multigrid methods, aggregation-based algebraic multigrid can be very efficient because of its relatively low setup time. However, even with K-cycle strategy it may still suffer slow convergence in some complex cases. Specifically, the PDEs whose dispersion tensor is rotated anisotropic, appear to be a challenging problem for aggregation-based algebraic multigrid methods. This is due to that for some rotation angle (e.g., $\pi/6$, $\pi/3$) the

original aggregation algorithm generates parallelogram aggregates (4 nodes form an aggregate in 2D) rather than preferred line aggregates. For this type of problems, Chen et. al. [25] proposed an improved aggregation strategy to improve the convergence. In [25] a new automatic aggregation algorithm determines appropriate aggregates based on eigenvectors associated with small eigenvalues of a block splitting matrix. The generated aggregates by this approach mostly are lines and aligned with the direction of the rotated anisotropy. In this work the improved aggregation algorithm is used in the simulation code.

3.3. Parallel implementation of the Solution Algorithm

In our simulation the space domain is divided into $m \times m = m^2$ blocks (m is a positive integer), each block contains a fixed number of grid cells, as shown in Fig. 1 (where the domain is divided into 4 subdomains). In each subdomain variables (velocities, pressure and porosity) are computed or updated by one process. For convenience, we divide the domain along the horizontal and vertical edges. In this format each process needs to share variables at the internal boundary with its neighbouring processes. For example, in Fig. 1 processes P_1 and P_2 share u while P_1 share v with P_3 at the internal boundary (solid lines). In each time step the variables at cell centres (pressure, permeability , hydraulic conductivity and concentration) next to the internal boundaries are communicated between processes using MPI in order to compute the solution correctly. Furthermore, the construction of transmissiblity matrix in experimenting field approach ([23]) also needs to communicate the residual values (at cell centres) between processes.

In the above computations only local communications with neighbours rather than global communications are needed for computing the variables at the internal boundary. This reduces the impact of communication cost to the parallel efficiency. After computing the transmissibility matrix, the resulted linear system in equation (25) is solved by parallel aggregation-based algebraic multigrid to obtain the pressure field. The porosity and the permeability are then updated.

However, there are several issues affect the parallel efficiency of multigird methods. These issues are parallel coarsening in setup phase, parallel smoothing and coarsest grid solving ([26; 27]). In parallel AGMG the aggregation algorithm aggregates unknowns locally without information from neighbouring processes, therefore the resulting prolongation operator, the associated coarse grid matrix and the smoother are different from their counterpart in sequential AGMG. Furthermore, the parallel pre- and post-smoothing, which may involve inversion of matrices, ignore the connection information with the variables which are not in current process. These factors may affect the convergence of AGMG. To assess the impact of the above issues on linear system solving arising from our numerical examples (described in the next section) we carried out some preliminary parallel run with different number of processes (cores) to observe the convergence of AGMG multigrid solver in the first 1000 time steps. We record the average multigrid cycles needed for solving the pressure field at each time step. The results of the runs are shown in Table 1. From the table we see that the convergence of AGMG is not affected significantly by the aforementioned first two issues for the numerical examples we studied in this work.

Table 1: Effect of parallelization on the convergence of AGMG. 1600^2 unknowns.

No. of processes (cores)	# level	AGMG iterations			
		$\theta = 0^\circ$	$\theta = 30^\circ$	$\theta = 45^\circ$	$\theta = 60^\circ$
1	7	14	14	14	14
4	7	14	14	14	14
16	7	14	15	14	15
64	7	14	15	15	15
256	7	15	15	15	15
1024	7	15	15	15	15

The coarsest grid solving is a critical step affecting the parallel efficiency [26] when the number of processes (cores) is large (1024 in this research). The size of the portions of the operator stored in each processes at this level is generally small, and the time required for communication may be higher than the time required to perform the calculations on a single process. Furthermore, the coarsest grid operator may couple all pieces of the global problem (i.e., it is dense, or nearly dense), and thus global communication of the right-hand side or other data may

be necessary. To avoid excessive communication at the coarsest level, we use Jacobi preconditioned conjugate gradient (PCG) to solve the coarsest grid system instead of Gauss eliminations (direct solving). Compared with Gauss eliminations, the CG algorithms can be parallelized easily and the implementation only need few global communications.

In all the coarsest grid solving of AGMG cycles, the PCG iterations stop when the 2-norm of the relative residual is less than a threshold. The computation of the threshold is as follow: For each visit of the coarsest grid, one may expect that a defect reduction of $\rho^{1/\kappa}$ will be sufficient, where ρ denotes the expected multigrid convergence factor and κ is the number of coarsest grid visits [27]. If the expected multigrid convergence factor is 0.15 and the number of coarse grid visits is 192, the the threshold can be computed as $\sqrt[192]{0.15} \approx 0.99$. Generally this threshold can be reached easily by one or two preconditioned CG iterations, while the overall AGMG convergence was not affected. Therefore in the actual implementation we let the threshold be 0.99.

In summary, the solution steps of the numerical algorithm (experimenting pressure field approach, parallel implementation) described in this section are shown in Fig. 5.

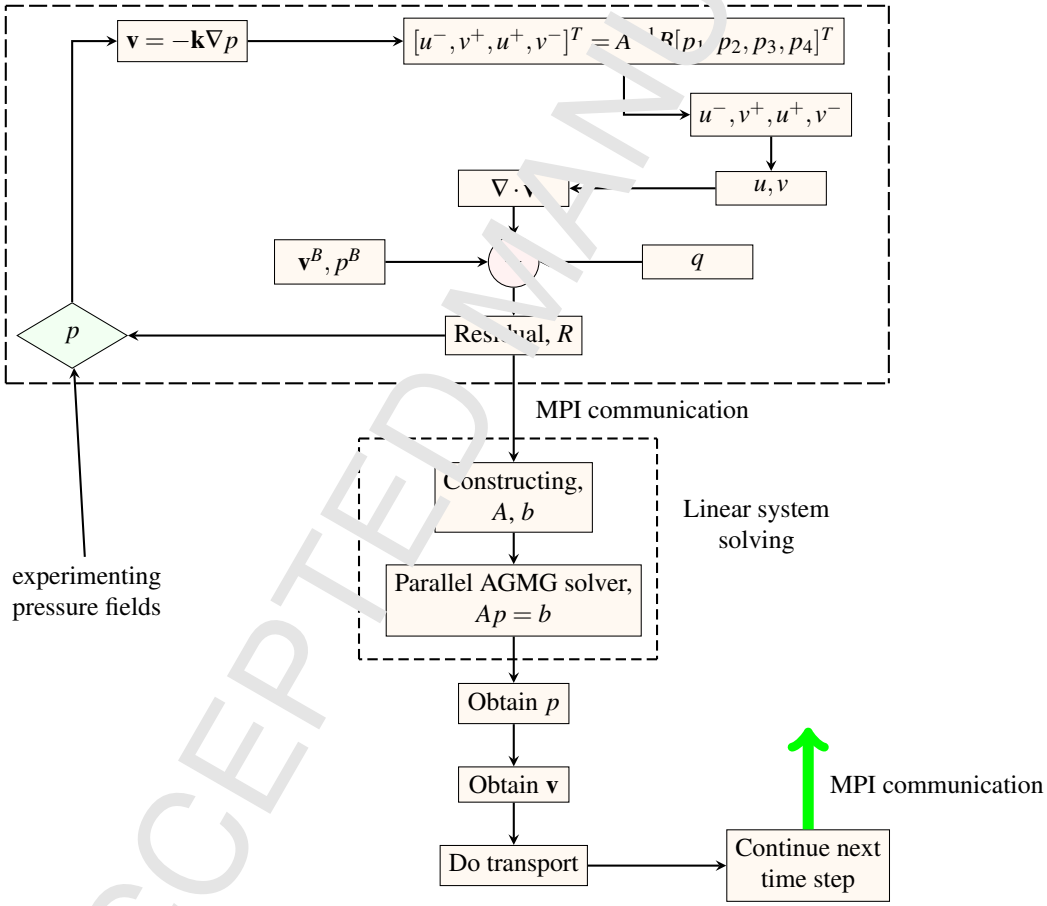


Figure 5: Flow chart of the solution using the experimenting pressure field approach to calculate the velocity field.

4. Numerical example

In this work, we consider a set of 2D numerical experiments to investigate the transport of nanoparticles in heterogeneous media which contains fractures. The physical domain is a square with side length of 3.2 meters and consists of three zones, where the permeability in the middle zone is rotated anisotropic (lightly hatched region in Fig.6). The whole domain is resolved by a mesh with size $n_x \times n_y = 6400 \times 6400$. The size of each cell is 0.0005 m

$\times$ 0.0005 m. In an attempt to investigate the parallel efficiency, the unknowns are divided in various of numbers of groups, with each number is an even power of 2. The largest number of processes used in this research is 1024.

The anisotropy is defined by two parameters. The first parameter is the angle that the principle direction of anisotropy is making with the x-axis. The other is the anisotropy ratio, which is the ratio between the components in the principle directions. The angles of anisotropy considered in this work are 0°, 30°, 45° and 60°. The anisotropy ratio is set to be $\varepsilon = 0.1$. Consequently the components of the initial permeability tensor in rock matrix is given by the formula

$$\begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix} = \begin{bmatrix} K_0(\cos^2(\theta) + \varepsilon \sin^2(\theta)) & K_0(1-\varepsilon)\cos(\theta)\sin(\theta) \\ K_0(1-\varepsilon)\cos(\theta)\sin(\theta) & K_0(\sin^2(\theta) + \varepsilon\cos^2(\theta)) \end{bmatrix}, \tag{29}$$

where K_0 is a constant given by 9.869×10^{-14}. The domain contains three fractures whose positions and lengths are shown in Fig.6. The width of each fracture is resolved by 2 volumetric cells, which makes the fracture width to be 0.001 m.

At the centre of the domain, a 0.46 m $\times$ 0.46 m square region (cross hatched region in Fig. 6) serves as a source of nanoparticle pollutant. The nanoparticles are emitted at an intensity 2×10^{-5} L/s. The fluid flow transports these particles and certain amount of them deposit in the media. In the region where nanoparticles are injected, the deposition of nanoparticles changes the homogeneity of the media so that the porosity reduces with respect to time. All the simulations stop when the minimum porosity of the domain has reached 0.10 and the numerical data are saved/recorded. Note that the value 0.10 is enough for us to observe the significant differences in fields (pressure, porosity and velocity) for various cases, however there is nothing particularly significant about this value and we are able to proceed further in time. For the boundary conditions, we impose velocity boundary condition of 2×10^{-6} m/s on the left edge of the domain. Moreover at the same place the pressure is set to 1 atmosphere (1.01325 bar). Initially, the pressure field is 0 except on the left side of the domain. No flow boundary condition is assumed on the top and bottom of the domain.

The parameters for the transport model described in the previous section are listed in Table 2. For convenience we made few assumptions. Firstly the nanoparticles are assumed not affecting the flow field. This assumption allows the decoupling of flow and transport equations and simplifies the calculations. Secondly, we assume that the permeability and the porosity are larger in the fractures. The permeability in the fracture is considered, approximately, 3 orders of magnitude than it is in the rock matrix. The initial porosity in fractures is assumed 0.9, while in the matrix the porosity is considered 0.3. The larger permeability in fractures makes the entrapment of the nanoparticles more difficult. We also assume that surface retention and entrainment of nanoparticles in fractures to be smaller than that in the rock matrix. All the simulations use a fixed time step size of $\Delta t = 3.25 \times 10^{-4}$ s.

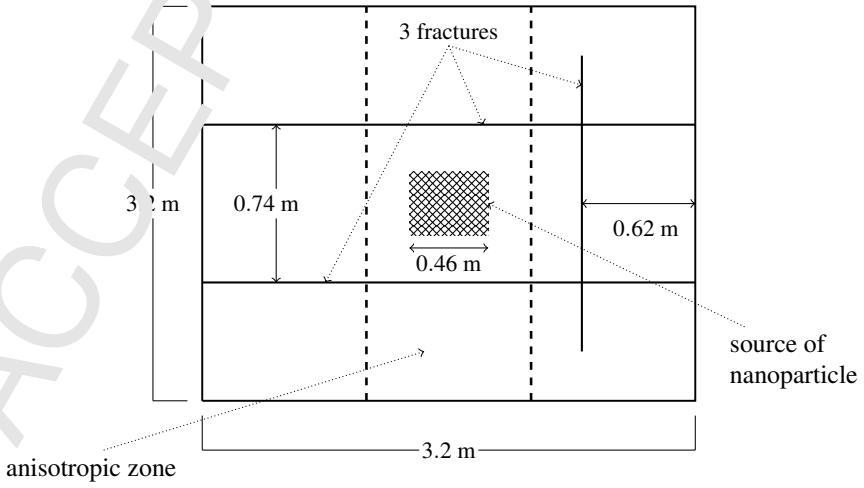


Figure 6: Schematic of the computational domain with the fracture system.

Parameters	Values
Initial permeability in rock matrix	$\mathbf{K_0}$
Isotropic layer	$\begin{bmatrix} 9.869 & 0 \\ 0 & 9.869 \end{bmatrix} \times 10^{-14}$ m^2
Anisotropic layer	
0°	$\begin{bmatrix} 9.869 & 0 \\ 0 & 0.987 \end{bmatrix} \times 10^{-14}$ m^2
30°	$\begin{bmatrix} 7.649 & 3.846 \\ 3.846 & 3.208 \end{bmatrix} \times 10^{-14}$ m^2
45°	$\begin{bmatrix} 5.428 & 4.441 \\ 4.441 & 5.428 \end{bmatrix} \times 10^{-14}$ m^2
60°	$\begin{bmatrix} 3.208 & 3.846 \\ 3.846 & 7.649 \end{bmatrix} \times 10^{-14}$ m^2
Initial permeability in fracture	$750 \times 9.896 \times 10^{-14}$ m^2

Parameters	Values
$L_x \times L_y$	3.2 m × 3.2 m
$n_x \times n_y$	6[illegible]6400
ϕ_0	[illegible]3
ϕ_0 (in fracture)	0.9
D^{Br}	2×10^{-9} m^{2}s^{-1}
μ	10^{-3} Pa s
γ_d	5 m^{-1}
γ_d (in fracture)	0.025 m^{-1}
γ_t	1 m^{-1}
γ_{pt}(in fracture)	0.005 m^{-1}
γ_e	10 m^{-1}
γ_e (in fracture)	0.05 m^{-1}
γ_{f_r}	0.001 m^{-1}
γ_{f_r} (in fracture)	5×10^{-6} m^{-1}
α_f	0.6
c_0	0
c_{s1}	0
c_{s2}	0
q_c	10^{-6} s^{-1}
Δ_t	0.02 s

Table 2: Parameters for single [illegible]no-size[illegible]article transport simulations.

5. Simulation results

In this section we first investigate the physics delivered from the simulation results for the nanoparticle transport in fractured media. The parallel simulations were carried out using 1024 cores. Secondly we assess the issue for parallel efficiency by measuring the running time for the simulations with 1000 time steps using various number (power of 2: 1, 4, 16, 64, 256 and 1024) of cores. All simulations were carried out on Shaheen II, a Cray XC40 delivering over 7.2 Pflop/s of theoretical peak performance. The system has 6,174 dual sockets compute nodes based on 16 core Intel Haswell processors running at 2.3GHz. Each node has 128GB of DDR4 memory running at 2300MHz. Overall the system has a total of 197,568 processor cores and 790TB of aggregate memory. At the end of 24 hours maximum execution time the restart mechanism implemented in the code saves the current numerical data which is used for the next restarting simulation.

Anisotropy angles	0°	30°	45°	60°
No. of time steps	1940000	1810000	1850000	1820000

Table 3: Total number of time steps for the simulations to reach the minimum porosity threshold 0.10.

5.1. The evolution of minimum porosity

The change of the minimum porosity versus time for different scenarios are shown in Fig. 7. The number of time steps to reach the minimum porosity threshold for each anisotropy angle are listed in Table 3. In all scenarios the minimum porosity decreases with time as a consequence of the increase of deposition of nanoparticles. The decreasing of minimum porosity becomes significantly after the time 1000 second. In the case for $\theta = 0°$ the minimum porosity decreases slower than the others do. The decreasing rate is the largest for the cases $\theta = 30°$ and $\theta = 60°$, where their time evolution profiles are pretty close. As we shall see later, the porosity profiles (Figs. 16) exhibit that the minimum porosity happened inside the (square) nanoparticle emission region. The anisotropy angles 30° and 60° exhibit a geometric symmetry to the square. The accumulation of the nanoparticles acts similarly

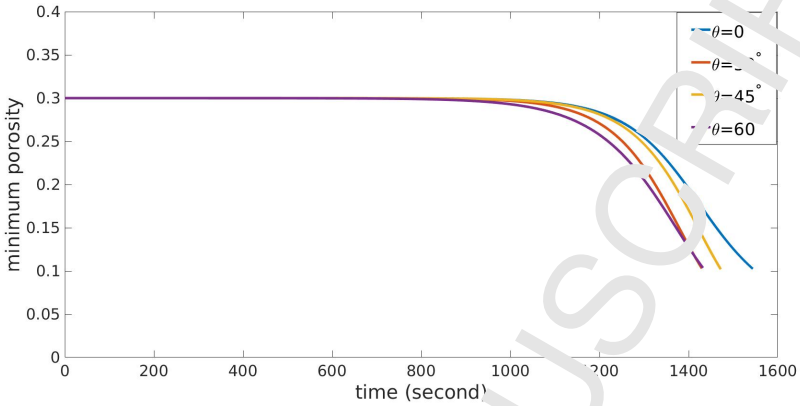
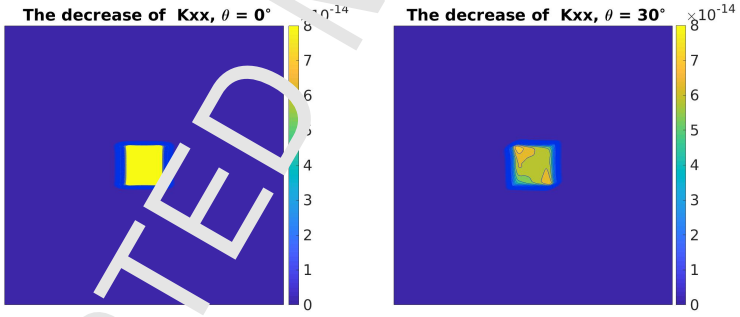
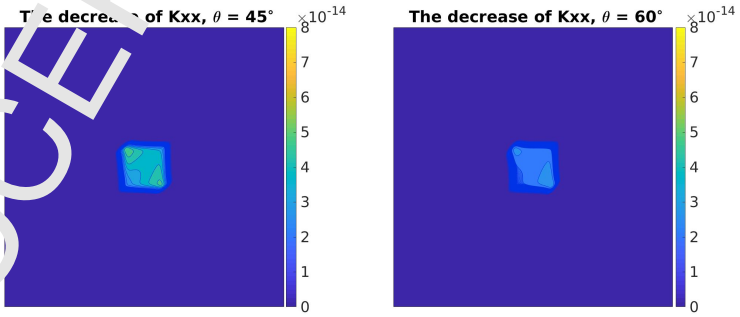


Figure 7: Evolution of minimum porosity.



(a) 0° (b) 30°



(c) 45° (d) 60°

Figure 8: The decrease of the permeability component Kxx at the end of simulation.

in these two cases.

Time evolution of the minimum values for each component of the permeability tensor $[K_{xx}K_{xy};K_{xy}K_{yy}]$ shows almost the same as the porosity plot in Fig. 7 for various anisotropy angle. In particular, the minimum of the permeability components decrease with time as a consequence of the increase of deposition of nanoparticles. Like the minimum porosity, the decrease of the minimum permeability components becomes significant after the time 1000 second. The decrease of the permeability tensor component K_{xx} at the end of the simulation is shown at Fig. 8 (the figures of other permeability tensor components and their time evolution are not shown here due to limitation of space). In the case for $\theta = 0^\circ$ the minimum permeability components K_{xx}, K_{xy} and K_{yy} decrease more than the others do. This may attribute to longer simulation time than the other cases.

5.2. Speed

The speed profiles sampled along the middle lines of the entire domain parallel to the coordinate axis for various cases are shown in Figs. 9 - 12. Because the speeds at the fractures ($\approx 10^{-3}$) are several orders of magnitudes larger than that ($\approx 10^{-5}$) in rock matrix, the y-axis is presented in terms of logarithmic scale. This should not be surprising as the permeability in fractures is three orders of magnitude larger than that in the rock matrix.

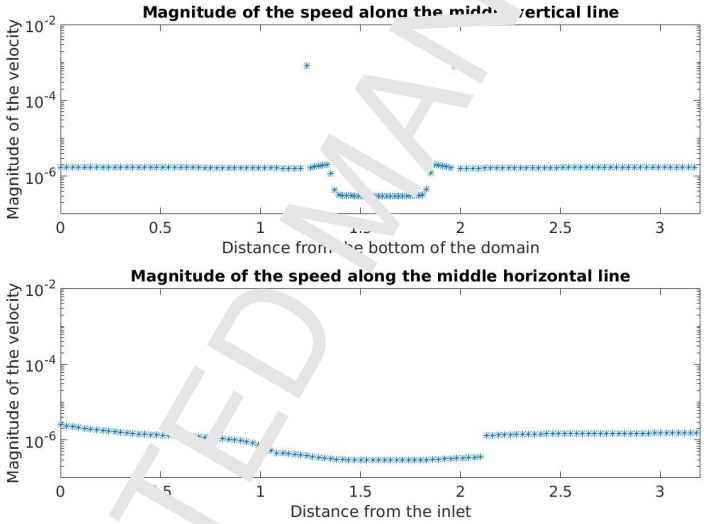


Figure 9: Speed profile along the middle vertical and middle horizontal lines of the entire domain for $\theta = 0^\circ$.

As seen from the figures, for all the anisotropy angles the speed profiles along the vertical middle line of the entire domain exhibit two significant large values at two horizontal fractures, roughly 1000 times larger than that at the rock matrix. On the other hand, along the middle horizontal line the fluid speed at the vertical fracture is 10 times larger than that in rock matrix. The speeds profiles also show that in the nanoparticle emission region (see Fig. 6) the fluid flow slows down, correspond to the relatively significant decreasing of the permeability of that area.

5.3. Pressure, Porosity and concentration fields

In this work the primary concern is on how the fractures and the anisotropy of the middle region affect the different variables such as pressure, porosity, flow velocities and concentration distribution of nanoparticles. In this study, we consider the scenarios in which the deposition of nanoparticles changes the properties of porous media. In particular, the deposition decreases the volume of the void space available for the flow and therefore, the permeability also decreases. Since the velocity is set to be constant at the left boundary, the average pressure changes as a consequence of the decrease of both the porosity and the permeability with time. In this section we highlight these effects as per the different scenarios.

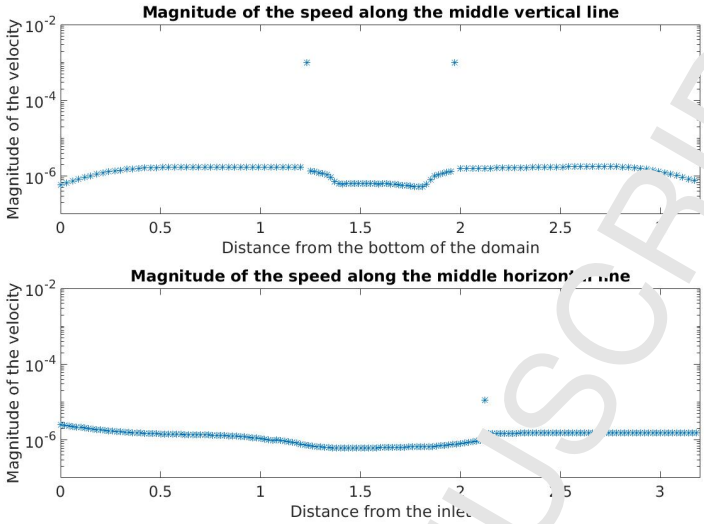


Figure 10: Speed profile along the middle vertical and middle horizonta lines of the entire domain for $\theta = 30^\circ$.

Figs 15 - 17 show the update of the various variable after the minimum porosity reached 0.10. First, we investigate the pressure field for each anisotropy scenario. From Fig. 15 one notices that the pressure field distorts with the direction of anisotropy. As the anisotropy angle increases, the gradient of the pressure profiles along the x-direction increases (Fig. 13). At the middle y-line of the domain, the pressure in the two horizontal fractures is smaller than that in the rock matrix (Fig. 14). This is understandable since it is easier for the fluids to pass through the fractures than they do in the rock matrix. Furthermore, Figs 13 and 14 show that the magnitude of the maximum pressure at the left hand side of the domain increases as the anisotropy angle increases, which indicates that large anisotropy angle hinders the fluid flow more significantly than small anisotropy angle does. Also notice that the pressure is larger in the upper half domain than that in lower half domain when the anisotropy angle is not zero, which again indicate the influence of the anisotropy angle on the pressure distribution.

The profiles of media's porosity at the end of the simulation in all cases are shown in Figs 16. From the area where the porosity is reduced, we see that the anisotropy affects the spread of the nanoparticles significantly. In the area surrounding the injection region of the nanoparticles, the porosity fields are different in these four cases. For the case $\theta = 0^\circ$ the area with minimum porosity is rectangular (Fig. 16a) otherwise it rotates according to the direction of anisotropy (Fig. 16b - 16d). Notice that the porosity in the fractures does not change significantly during the simulation in all cases. This may be because the velocity in the fracture is large compared with that in the rock matrix and therefore the rate of detachment is increased. One may also observe that nanoparticles spread towards the two horizontal fractures for the cases $\theta = 30^\circ$, $\theta = 45^\circ$ and $\theta = 60^\circ$. In these three cases the spread of nanoparticles along the horizontal fractures reaches the fracture intersections. It is not surprising that the horizontal fractures transport nanoparticles more rapidly than the rock matrix does. In the case $\theta = 30^\circ$, the spread of nanoparticles just barely reach the horizontal fractures when the simulation ends, so the transportation of nanoparticles in fractures didn't proceed long enough to see the similar effect appeared in the cases $\theta = 45^\circ$ and $\theta = 60^\circ$.

Fig. 17 exhibits the concentration contours for different cases. Comparing with Fig. 16 the profiles of the concentration fields in all four cases match well with their corresponding porosity fields. From Fig. 17 the nanoparticles spread out in broader range for the case $\theta = 60^\circ$ than the other scenarios do. It can be seen that nanoparticles are transported all the way to the fractures. After the pollutant is transported into the fractures, the concentration near the area around the fractures increases gradually as the time evolves.

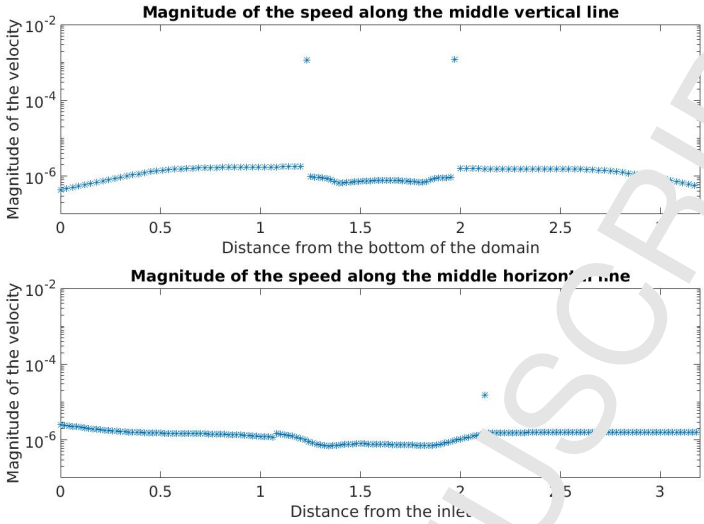


Figure 11: Speed profile along the middle vertical and middle horizontal lines of the entire domain for $\theta = 45°$.

5.4. Parallel efficiency

In this section we assess the performance and parallel efficiency issues for the simulations. In each case the execution times for linear system solving and the overall simulation are recorded. The core numbers used in simulation for assessing parallel efficiency issue are 1, 4, 16, 64, 256 and 1024. The problem size is fixed (6400×6400 unknowns) and all the anisotropy angles (0°,30°,45°,60°) are considered. The simulation times for executing 1000 time steps from the restart file at a fixed time are measured. As mentioned in Section 3, the numerical algorithm of simulation involves three parts: the construction of the matrix arising from MPFA discretization, solving the resulted linear system for pressure field and the update of the variables such as velocity, porosity and concentration. In attempt to assess the issue for the impact of the AGMG's performance on the parallel efficiency of the overall algorithm, the running times for both the overall simulation and linear system solving are compared.

The timing results are listed in Tables 4 (for the whole simulation) - 5 (for linear system solving). The corresponding parallel efficiency plots are shown in Figs. 18 - 19. The results in the tables show that the linear system solving step takes from $\sim 5\%$ (1 core) to $\sim 30\%$ (1024 cores) of the overall simulation time for all the anisotropy angles. The execution time for the linear system solving in the case 0° is the smallest among all scenarios. This can be verified by the number of unknowns on each multigrid levels, which indicates that for the case 0° the reduction ratio from fine level to the next coarser level is the largest and therefore has the least computational cost for smoothing and interpolation/prolongation among these cases.

As seen from Fig 18 the parallel efficiencies, either for the entire simulation or for linear system solving, do not differ significantly in all anisotropy angles. For large number of cores (≥ 256) the parallel efficiency decreases to below 50% for the overall simulation (see Fig. 18). This may attribute to the performance of linear system solver. As seen from Fig. 19, the parallel efficiency for the linear system solving step reduces significantly in all anisotropy angles for large number of cores (≥ 256).

6. Conclusion

In this work we have simulated the flow and transport of nanoparticles in porous media. The domain is composed of three vertical layers. The middle layer is assumed anisotropic with respect to permeability. A simple system of fractures is also considered. Several scenarios of anisotropy of the permeability of the middle layer are considered. Spatial variables were discretized using the multipoint flux approximation (MPFA) method. The resulting linear system for pressure field is solved by aggregation-based algebraic multigrid (AGMG). We assume

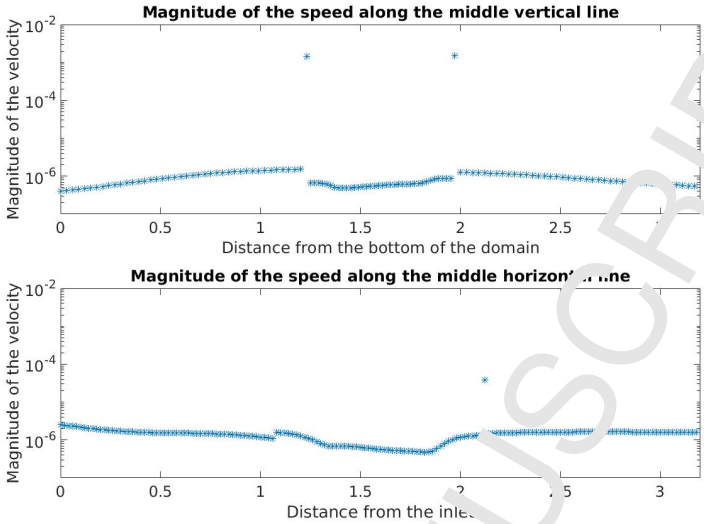


Figure 12: Speed profile along the middle vertical and middle horizonta lines of the entire domain for $\theta = 60°$.

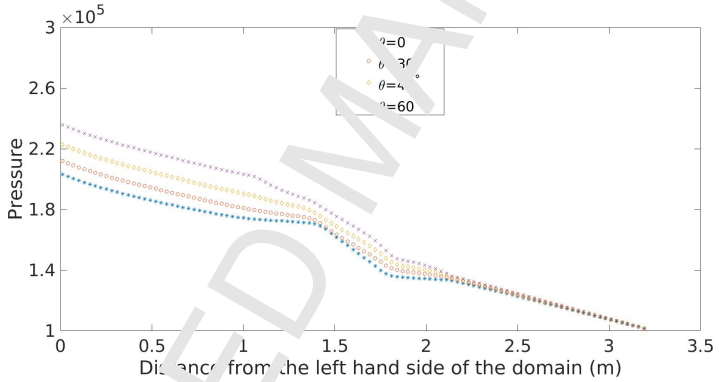


Figure 13: Pressure profile along the middle x-line of the domain.

that the transportation of nanoparticles is less hindered in fractures hence the rate of particle retention and deposition is small. The widths of the fractures are 1 mm. In order to represent the fractures by volumetric cells we use high resolution mesh (6400 × 6400). The solution algorithm is parallelized so that the results can be obtained in reasonable length of time. The fields for pressure, porosity, concentration and velocity are distorted by the anisotropy. The direction of distortion follows the anisotropy angles. The fluid speed in fractures is several order of magnitude larger than that in rock matrix. The fractures act as conduits which convey the nanoparticles much faster than the rock matrix does. In this model the deposition of pollutant is not severe in the fractures due to the high fluid velocity and hence increased rate of detachment.

The issues for parallel efficiency of the simulations are assessed. The efficiency is not affected significantly by the anisotropy angle, which is expectable since the intensity of anisotropy in our study is relatively mild. The performance and speed up of linear system solving (for pressure field) affect the overall parallel efficiency of the simulation significantly. For large number of cores the linear system solving step becomes the bottleneck of the performance and efficiency, as seen from the test results for AGMG's parallel efficiency. This may attribute to the communication cost for solving the coarsest grid problem (visiting 192 times in each multigrid cycle) and the overhead arising from the setup phase in parallel AGMG.

To the best of our knowledge, this is the first attempt using parallel MPFA discretization and AGMG to simulate

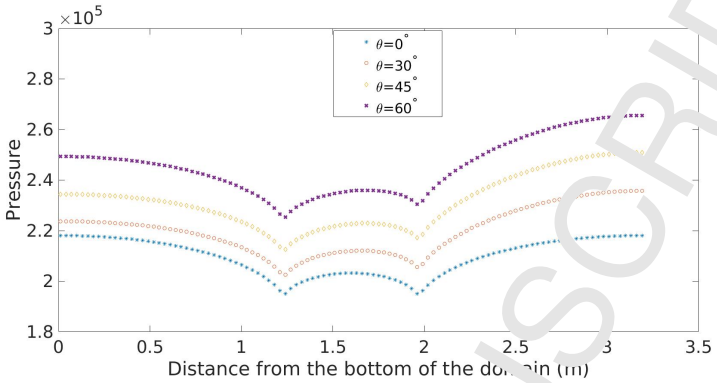
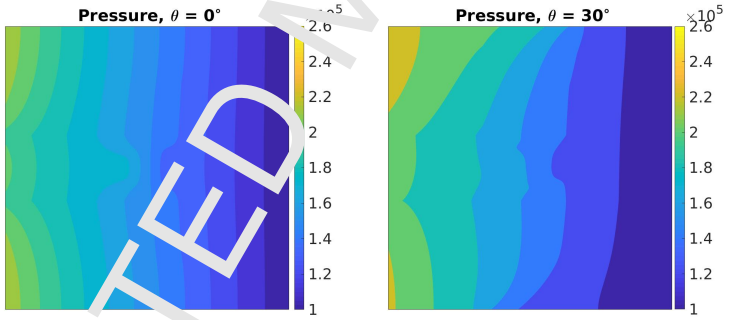
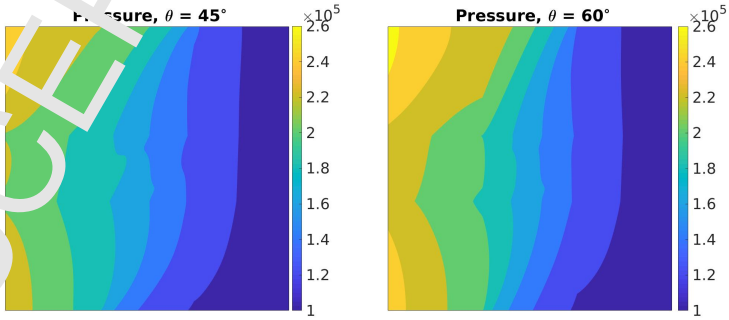


Figure 14: Pressure profile along the length of the recta ular doma n (the middle y-line).

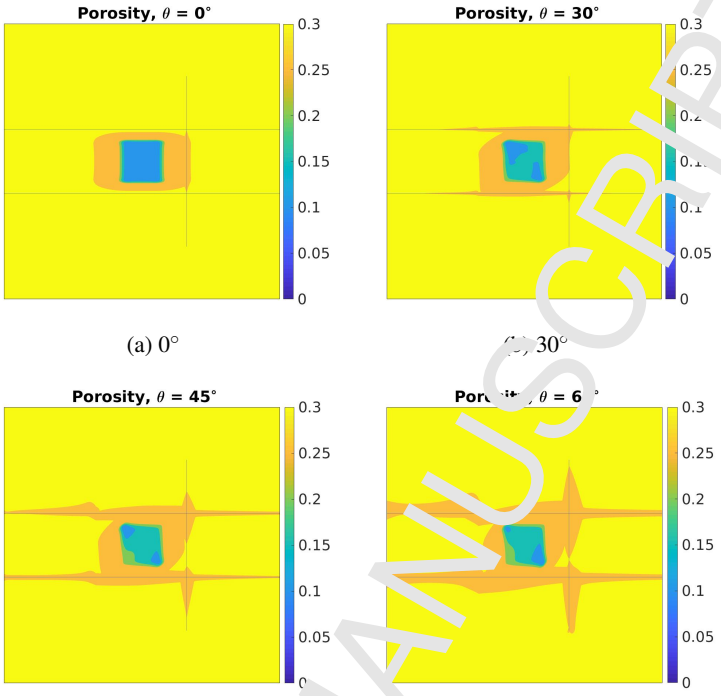


(a) 0° (b) 30°



(c) 45° (d) 60°

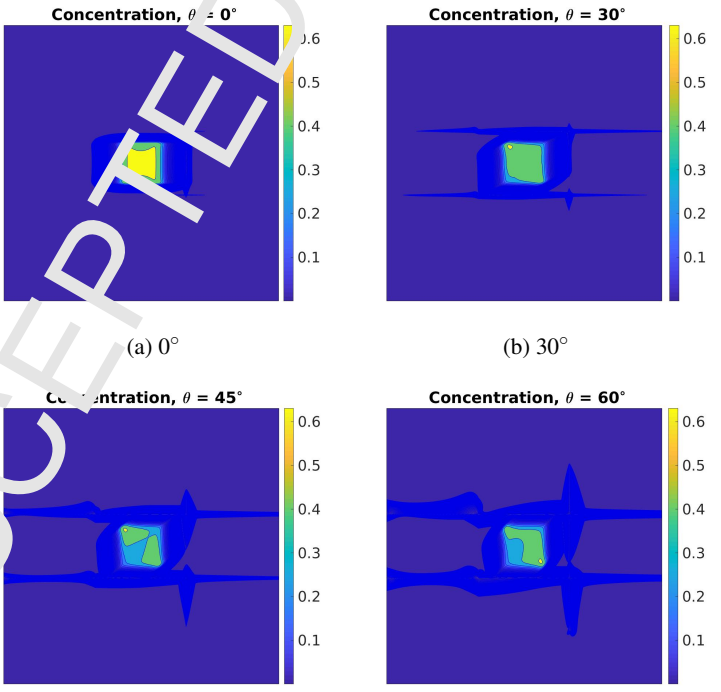
Figure 15: Pressure fields at the end of simulation.



(a) 0° (b) 30°

(c) 45° (d) 60°

Figure 16: Porosity fields at the end of simulation.



(a) 0° (b) 30°

(c) 45° (d) 60°

Figure 17: Concentration fields at the end of simulation.

	0°		30°		45°		60°	
No.of processes (cores)	time	speedup	time	speedup	time	speedup	time	speedup
1	468524		479235		481930		478444	
4	123520	3.79	124610	3.85	125542	3.84	124320	3.85
16	31120	14.10	32164	14.90	32333	14.9	31873	15.01
64	8433	55.55	8603	55.71	8572	55.22	8671	55.52
256	3058	153.21	3125	153.36	3128	154.07	3074	155.64
1024	952	495,15	1068	448.72	1047	460.31	1101	434.55

Table 4: The overall simulation time (in seconds) and speedup for running 1000 time steps.

	0°		30°		45°		60°	
No.of processes (cores)	time	speedup	time	speedup	time	speedup	time	speedup
1	20497		22872		21871		23020	
4	6160	3.33	6590	3.47	6371	3.42	6523	3.53
16	1846	11.11	2090	10.94	1950	11.18	2108	10.92
64	580	35.34	678	33.73	612	35.64	686	33.55
256	334	61.37	365	62.66	362	60.25	371	62.05
1024	263	77.93	309	74.02	298	73.19	304	75.72

Table 5: Execution time (in seconds) and speedup for linear system solving (for pressure field) in 1000 time steps.

and investigate the nanoparticle transport in porous media with fractures. The current work shed light on the new approaches to tackle the complicated systems of nanoparticle transport with fractures and full-tensor permeability. This work can be extended to multiphase flow in a rock containing fractures of various scales, which is a topic we plan to work on in near future.

Acknowledgments

The research reported in this publication was supported in part by funding from King Abdullah University of Science and Technology (KAUST) through the grant BAS/1/1351-01-01. Also the authors would like to thank Dr. P. K. Jimack for constructive criticism of the manuscript.

References

[1] D. Lin, X. Tian, F. Wu, B. Xing, *Fate and transport of engineered nanomaterials in the environment*. J. Environ. Qual. 39(2010), issue 6, pp. 1896-1908.

[2] A. Salama, A. Negara, M. El-Amin, S. Sun, *Numerical investigation of nanoparticles transport in anisotropic porous media.* J. Contam. Hyro., 267 (2014), pp. 117-130.

[3] S. Lin, Y. Cheng, Y. Bedrocombe, K. L. Jones, J. Liu, and M. R. Wiesner, *Deposition of Silver Nanoparticles in Geochemically Heterogeneous Porous Media: Predicting Affinity from Surface Composition Analysis*

[4] H. Ma, L. L. Williams, S. A. Diamond, 2013. *Ecotoxicity of manufactured ZnO nanoparticles - a review*. Environ. Pollut. 172 (2013), pp. 76-85.

[5] S. Sun, A. Salama, M. F. El-Amin, *Matrix-oriented implementation for the numerical solution of the partial differential equations governing flows and transport in porous media.* Computer & Fluids, 68(2012), pp.38-46.

[6] M. F. El-Amin, S. Sun, A. Salama, *Modeling and Simulation of Nanoparticle Transport in Multiphase Flows in Porous Media: CO2 Sequestration.* SPE-163089. Mathematical Methods in Fluid Dynamics and Simulation of Giant Oil and Gas Reservoirs 2012.

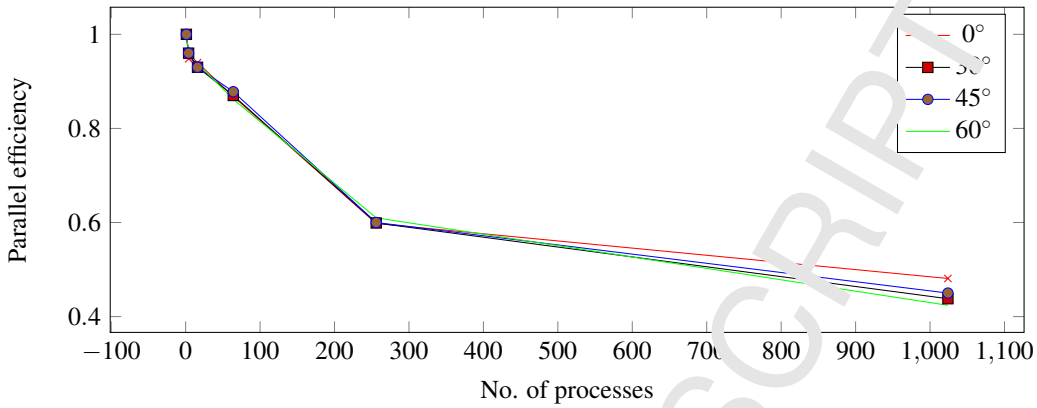


Figure 18: Strong scaling: parallel efficiency for the overall simulation.

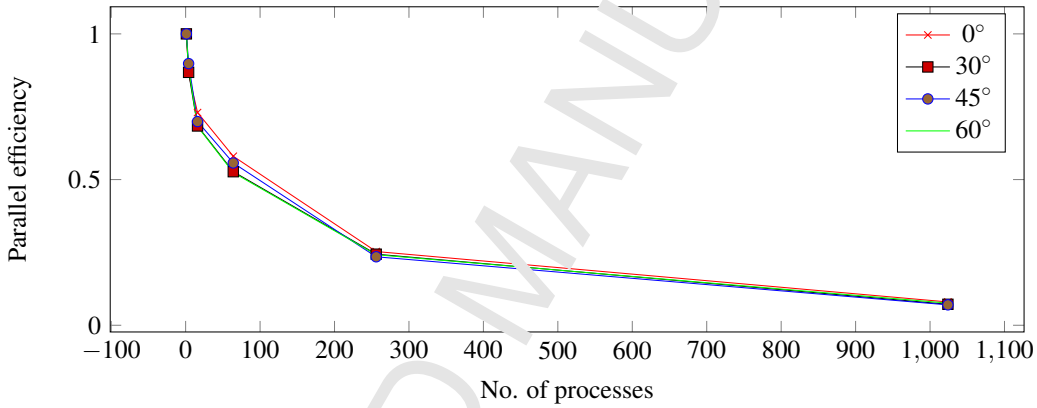


Figure 19: Strong scaling: parallel efficiency for the linear system solver.

[7] M. F. El-Amin, A. Salama, S. Sun, *Numerical and dimensional investigation of two-phase counter current imbibition in porous media*. J. of Comput. Appl. & Math., 242(2013), pp. 285-296.

[8] Y. Notay, *An Aggregation-based Algebraic Multigrid Method.* Elect. Trans. on Numer. Analy., 37 (2010), pp. 123-146

[9] A. Napov, Y. Notay *An algebraic multigrid method with guaranteed convergence rate.* SIAM J. Sci. Comput., 34, No. 2, A1079-A1109.

[10] P. R. Johnson, M. Elimelech, *Dynamics of colloid deposition in porous media: blocking based on random sequential adsorption*. Langmuir 11 (1995), pp. 801-812.

[11] N. Tufenkji, M. Elimelech, *Deviation from the classical colloid filtration theory in the presence of repulsive DLVO interactions* Langmuir 20 (2004), pp. 10818-10828.

[12] J. F. Schijven, S. M. Hassanizadeh, R. H. de Bruin, *Two-site kinetic modeling of bacteriophages transport through columns of saturated dune sand*. J. Contam. Hydrol. 57 (2002), pp. 259-279.

[13] S. A. Bradford, S. R. Yates, M. Bettahar, J. Simunek, 2002. *Physical factors affecting the transport and fate of colloids in saturated porous media.* Water Resour. Res. 38 (2002), pp. 1327-1338

[14] G. Gargiulo, S. Bradford, J. Simunek, P. Ustohal, H. Vereecken, E. Klumpp, *Bacteria transport and deposition under unsaturated conditions: the role of the matrix grain size and the bacteria surface protein*. J. Contam. Hydrol. 92 (2007), pp. 255-273

[15] E. Goldberg, M. Scheringer, T. D. Bucheli, K. Hungerbuhler, *Critical assessment of models for transport of engineered nanoparticles in saturated porous media.* Environ. Sci. Technol. 48 (21), 2014, pp. 12732-12741.

[16] P. Bedrikovetsky, A. Santos, A. Siqueira, A. L. Souza, F. Shecaira, *A stochastic model for deep bed filtration and well impairment.* SPE 82230, 2003, European Formation Damage Conference in The Hague, The Netherlands.

[17] A. Benamar, N. D. Ahfir, H. Q. Wang, A. Alem, *Particle transport in a saturated porous medium: pore structure effects.* C. R. Geosci. 339 (2007), pp. 674-681

[18] B. Ju and T. Fan,*Experimental study and mathematical model of nanoparticle transport in porous media.* Powder Technology, 192 (2009), issue 2, pp. 195-202

[19] E. Cullen, D. M. O'Carroll, E. K. Yanful, B. Sleep, *Simulation of the subsurface mobility of carbon nanoparticles at the field scale*. Adv. Water Resour., 33 (2010), pp. 361-371.

[20] C. Gruesbeck, R. E. Collin, *Entrainment and deposition of fine particles in porous media.* Soc. Pet. Eng. J. 24 (1982), pp. 847-855.

[21] C. An, A. Masoud, B. Yan, J. E. Killough, *A new study of magnetic nanoparticle transport and quantifying magnetization analysis in fractured shale reservoir using numerical modeling.* J. Nat. Gas Sci. and Eng., 28 (2016), pp. 502-521.

[22] C. An, B. Yan, A. Masoud, L. Mi, J. E. Killough, Z. Heidan, *Estimating spatial distribution of natural fractures by changing NMR T_2 relaxation with magnetic nanoparticles.* J. Pet. Sci. and Eng., 157 (2017), pp. 273-287.

[23] S. Sun, A. Salama, M. F. El-Amin, *An equation-type approach for the numerical solution of the partial differential equations governing transport phenomena in porous media.* Procedia Comput. Sci., 9 (2012), pp. 661-669.

[24] A. Salama, S. Sun, M. Wheeler, *Solving global problem by considering multitude of local problems: application to flow in anisotropic porous media using the multipoint flux approximation.* J. Comput. Appl. Math. 267 (2014), pp. 117-130.

[25] M. Chen, A. Greenbaum, *Analysis of an aggregation-based algebraic two-grid method for a rotated anisotropic diffusion problem.* Numer. Lin. Alg. with Appl., 22 (2015), Issue 4, pp. 681-701.

[26] E. Chow, R. D. Falgout, J. J. Hu, R. S. Tuminaro and U. M. Yang *A survey of parallelization techniques for multigrid solvers*, Parallel Processing For Scientific Computing, SIAM, series on Software, Environments, and Tools (2006).

[27] J. Linden, G. Lonsdale, H. Ritzdorf, A. Schuller, *Scalability aspects of parallel multigrid*. Future Generation Computer Systems 10 (1994), pp. 429-439.

[28] F. He, M. Zhang, T. Qian, D. Zhao, *Transport of carboxymethyl cellulose stabilized iron nanoparticles in porous media: column experiments and modeling.* J. Colloid Interface Sci. 334 (2009) , pp. 96-102.