Parallel algorithm for fractal model of creation of soil structure using particles-size distributions

V Roubtsova, B Morin, M Chekired
Institut de recherche d’Hydro-Quebec, 1740 boul. Lionel-Boulet, Varennes (Quebec) J3X 1S1
e-mail: roubtsova.varvara@ireq.ca

Abstract. To simulate the soil in 2-D we use a fractal scaling of the soil particle-size distribution. The sequential algorithm of this model is based on an 11 levels deep nested loop in which the inner limits are dependent on the previous iteration. In this article we describe the parallel algorithm for this type of problem which avoids the dependent recursion by using a flattened generator.

1. Introduction
Each year, many disasters happen in the word because of interaction of water with soils particles. Dam failures, blurring of road coating, destruction of houses caused by the liquefaction of foundations, all are the sources of not only of physical damages but also of the loss of human lives. The cause of such catastrophes at a macro level which can be found at the micro level is very difficult to observe.

The main objective of our project is to provide simulation tools for both engineers and researchers to freely explore an area of research that would otherwise be very difficult to achieve. We are thus developing a virtual laboratory for geotechnical research. The advantages of this laboratory are (i) access to parameters for which measurement in real experiments are impossible, (ii) the possibility to make experiments which would otherwise be difficult or impossible to recreate in laboratories (either by being too expensive, too long, too difficult to measure or too difficult to recreate), (iii) errors made by engineers or researchers can be corrected at only the cost time.

The Discrete Element Method (DEM) was used to model the soil. This Lagrangian method takes into account all sorts of interaction forces between the particles. We can watch an individual particle: we know its velocity, its location and the forces acting on it at each moment. The fluid motion is based on the Navier-Stokes equations using Marker-and-Cells (MAC) method. This Eulerian approach gives access to information about the pressure and real velocity at each point in a pore-channel.

The visualization plays an important role in the validation of numerical algorithms. Using Alioscopy technology, we were able to project eight different perspectives on a screen covered by a lenticular array, resulting is seven simultaneously observable angles of vision. This technology allows multiple observers to interact with a simulation, and permits more complex systems to be viewed. The OpenGL (Open Graphics Library) API (Application Programming Interface) was used to compute both the flat projection, as well as the different views required by the Alioscopy technology in a portable manner. GLSL (Graphics Library Shading Language) shaders were used to combine the
views into a single image projected on the screen, which is then redecomposed into individual images by the lenticular lens.

One of the possible applications for this virtual laboratory is the computation of soil permeability. Soil permeability is one of the most important parameter and is used to preview the soil behavior under water flow. It is dependent only on the network of the pore channels. Our laboratory can helps to determine this parameter by using only the particles-size distribution. The random distribution of particles, with both diameter and quantity corresponding to the particle-size curve, guarantees a good fit with actual soil in 3-D. We have the opposite situation in 2-D, where we need to create an artificial model of pore channels network. There are a lot of different models of pores channels [1, 2, 3, 4], we chose the pore-soil fractal approach [4]. This model offers a wider scope for modeling of soil.

2. Fractal model
The fundamentals of fractal geometry are described by Mandelbrot [5]. Fractals are very popular in soil research as a means of characterizing various properties of porous media. For examples: the fractal pore-size distribution was presented by Friesen and Mikula [6], the Sierpinski carpet was used to model the soil water retaining curve by Soto and Vilar [7], the pore-solid-fractal (3G-PSF) approach accounts for a wide range of particle-size distribution. More information about the different applications of the fractal in soil science is presented in [10].

Initially, soil is represented as a random matrix composed of both particles and pores. In the 2-D model, we need to distribute particles within the matrix in such a way as to obtain the same porosity and conductivity as in an actual 3-D soil.

A detailed study of the Pore Solid Fractal (PSF) model has been conducted by Perrier et al. [4]. The generation of a PSF starts with the initiator, which consists of a region that forms a square surface of length L in a 2-D configuration. This region is divided into R² subregions with length L/R. The subregions represent the pores, the particles, and the smaller subregions that will be divided in the next iteration step.

This structure can be described by a generator that includes the scaling factor R, the number of pores P, and the number of particles (solids) S. The number of unspecified elements F (fractals) is equal to R³ – P – S.

This model was extended by combining three fractal processes (generators) in series by Lehmann et al. [9]. Each generator was applied for a certain number of iterations. The resulting three-generator-pore-solid-fractal (3G-PSF) approach accounts for a wide range of particle-size distribution.

We used porosity as the marker of good agreement with natural pore channel network because this parameter represent a more natural fit that the pore-size distribution. For example there is a big difference between definition of pore-size distribution in [11] and [12].

To obtain a more realistic model, we replaced the solid square subregions by subregions containing a circular particle surrounded by an empty area. After some iterations, the porosity is found as follows:

\[
\phi_{n+2n+3} = \frac{P_{1} + (1-0.25 \times \pi)S_{1}}{P_{1} + S_{1}} \left( 1 - \left( \frac{F_{1}}{R_{1}^{2}} \right)^{n} \right) + \frac{P_{2} + (1-0.25 \times \pi)S_{2}}{P_{2} + S_{2}} \left( 1 - \left( \frac{F_{2}}{R_{2}^{2}} \right)^{n} \right) + \frac{P_{3} + (1-0.25 \times \pi)S_{3}}{P_{3} + S_{3}} \left( 1 - \left( \frac{F_{3}}{R_{3}^{2}} \right)^{n} \right)
\]

the indices 1, 2, and 3 correspond to the first, second, and third generators respectively end n is the number of iterations. All unspecified elements are considered to be pores for the last iteration.

In the cases of low porosity we don’t have the pore subregions but we have the pore channels that assure the passing of water through soil matrix. The last one is very important for compacting permeable soils.

Finally, the best approximation of the particle size distribution curves and the porosity are determined by the parameters of the generators R₁, P₁, S₁, n₁, R₂, P₂, S₂, n₂, R₃, P₃, S₃, n₃.
Looking at figure 1 we can see the steps of the development of a fractal porous medium that corresponds to a typical particle-size distribution. The first generator \((R_1 = 3, S_1 = 3, P_1 = 4)\) was used for the first three steps and it approximate the section of the curve for coarse particles. Using the second generator \((R_2 = 4, S_2 = 5, P_2 = 7)\) for the next two steps, we approximate the middle section of the curve. Finally we used the third generator \((R_3 = 3, S_3 = 2, P_3 = 2)\) to approximate the last part of the curve, representing fine particles.

The location of particles into pattern was determined by the following conditions: (i) the fractal structure must have more possible channels from left to right, (ii) the channels mustn’t be straight. It should be noted that different location patterns exist for the same parameters. Figure 2 shows examples of soil pattern for different maximal scaling factors.

**Figure 1. Construction of a 3G-PSF with round particles.**

*For the first generator: \(R_1 = 3, S_1 = 3, P_1 = 4, \) and \(n_1 = 3; \) for the second generator: \(R_2 = 4, S_2 = 5, P_2 = 7, \) and \(n_2 = 2; \) and for the third generator: \(R_3 = 3, S_3 = 2, P_3 = 2, \) and \(n_3 = 3.\)*
The results obtained by this model have been presented by Chekired and Roubtsova [13]. A summary of key achievements are listed below.

A comparison of the particle size distribution of various soil samples with those obtained by fractal modeling is shown in Figure 3.

The water flow through a pore channel is described by the Navier-Stokes equations for viscous-incompressible liquids, and is expressed by:

\[
\rho \left[ \frac{\partial \vec{v}}{\partial t} + (\vec{v} \nabla)\vec{v} \right] = -\nabla p + \eta \nabla^2 \vec{v} + \vec{f}
\]

\[
\nabla \cdot \vec{v} = 0
\]

where \( \rho \) is the density of the liquid (kg/m\(^3\)), \( t \) is the time (s), \( \vec{v} \) is the flow velocity (m/s), \( p \) is the water pressure (Pa), \( \eta \) is the dynamic viscosity (Pa s), and \( \vec{f} \) is the force of gravity (N). Water flows through the pattern from left to right by applying a pressure differential as boundary conditions. Hydrodynamic flows through four different soil samples generated by the fractal model are presented in Figure 4.
The coefficient of permeability (m/s) for this soil sample is (Scheidegger [14])

\[ K = \frac{\langle v \rangle s \rho g}{\Delta p} \]

where \( \langle v \rangle \) is the average velocity in the soil (m/s), \( s \) is the length of the sample in the direction of flow (m), and \( \Delta p \) is the pressure difference at the boundary (Pa).

A comparison of porosity and permeability obtained from the fractal model and from empirical formula for different soils is presented in Tables 1 and 2 respectively.

**Table 1. Comparison of porosity**

|        | Silt   | Sand   | Fine Gravel | Coarse Gravel |
|--------|--------|--------|-------------|---------------|
| Empirical | 0.406  | 0.303  | 0.261       | 0.268         |
| Fractal  | 0.405  | 0.314  | 0.265       | 0.279         |

**Table 2. Comparison of coefficient of permeability (m/s)**

|        | Silt       | Sand       | Fine Gravel | Coarse Gravel |
|--------|------------|------------|-------------|---------------|
| Empirical | 1.07 \(10^{-4}\)  | 2.03 \(10^{-4}\)  | 2.17 \(10^{-4}\)  | 5.09 \(10^{-3}\)  |
| Fractal Min  | 1.61 \(10^{-4}\)  | 2.18 \(10^{-4}\)  | 8.61 \(10^{-5}\)  | 9.68 \(10^{-4}\)  |
| Fractal Max  | 1.94 \(10^{-4}\)  | 2.65 \(10^{-4}\)  | 3.97 \(10^{-4}\)  | 3.44 \(10^{-3}\)  |

The empirical porosity of the soil was calculated using \( \phi = 0.255[1 + 0.83^{U}] \) [15], where \( U = d_{60}/d_{10} \) is the coefficient of grain uniformity and where \( d_{60} \) and \( d_{10} \) represent the grain diameter in mm at which 60% and 10% grains are finer respectively. For all type of soil, six different pattern where created and the different values for the coefficients of permeability were obtained. We present the both minimum and maximum values for the coefficient of permeability computed during simulations for various particle positions. It is not proven that we take in account all possible placements of particles.

The empirical coefficient of permeability was calculated using the Pavchich formula [11] expressed as follows:

\[ K = \frac{8}{\nu} \cdot 10^{-3} B \sqrt{U} \frac{\phi^3}{(1-\phi)^2} d_{i7} \]
where \( g \) is the acceleration due to gravity, \( \nu \) is the kinematic viscosity of water (m\(^2\)/s), \( B \) is equal to 1 for sand and gravel and 0.35-0.4 for pebbles, and \( d_{17} \) represents the grain diameter in mm for which 17% of the sample are fine then. This formula has been found to be satisfactory for a wide range of soils.

These results indicate that the proposed 2-D model can virtually generate different kinds of soils successfully using only grain size distribution curves, and can estimate soil characteristics, such as the coefficient of permeability.

3. Parallel algorithm to create the soil pattern

As notice early we need to check all possible combinations of 12 variables (\( R_1, P_1, S_1, n_1, R_2, P_2, S_2, n_2, R_3, P_3, S_3, n_3 \)) to create the soil pattern with the best approximation for particle-size curve and porosity. The user can select only the maximum scaling factor \( R_{\text{max}} \). The numbers of possible combinations for other parameters are determined by this factor and the particle-size distribution.

The pseudo code for sequential algorithm is:

```plaintext
for \( R_1 = 2 \) to \( R_{\text{max}} \) 
  for \( n_1 = 0 \) to \( n_{1\text{max}} \) (\( n_{1\text{max}} = f(R_1) \)) 
    for \( R_2 = 2 \) to \( R_{\text{max}} \) 
      for \( n_2 = 0 \) to \( n_{2\text{max}} \) (\( n_{2\text{max}} = f(n_{1\text{max}}, R_1, R_2) \)) 
        for \( S_1 = 0 \) to \( R_1^2 - 1 \) 
          for \( P_1 = 0 \) to \( R_1^2 - S_1 \) 
            for \( S_2 = 0 \) to \( R_2^2 - 1 \) 
              for \( P_2 = 0 \) to \( R_2^2 - S_2 - 1 \) 
                for \( R_3 = 2 \) to \( R_{\text{max}} \) 
                  for \( S_3 = 0 \) to \( R_3^2 - 1 \) do 
                    for \( P_3 = 0 \) to \( R_3^2 - S_3 - 1 \) 
                      {Computation distribution and porosity}
```

Where:

\[
n_{1\text{max}} = -\frac{\log\left(\frac{d_{\text{min}}}{L_1}\right)}{\log(R_1)} + 1
\]

\( L_1 = d_{\text{max}} R_1 \) is the length of matrix;

\[
n_{2\text{max}} = \left[-\frac{\log\left(\frac{d_{\text{min}}}{L_2}\right)}{\log(R_2)} + 1\right] \times \left(\frac{R_1^2 + R_2^2}{4}\right)
\]

\( L_2 = \frac{L_1}{R_1^n} \) is the length of matrix for second generator;

\( d_{\text{max}} \) is the maximal size of soil particle;

\( d_{\text{min}} \) is the minimal size of soil particle.

The twelfth variable \( n_3 \) only has a single value for each combination of the other variables. It is thus the dependent parameter.

We can note that with \( R_{\text{max}} \) only equal to three, we obtain more than ten millions possible combinations. It is thus important to decrease the time to parse the results.
Using this algorithm, the computation of porosity and particle-size distribution are independent for each combination of fractal parameters while using the same instructions for all combinations. This type of problem is ideal candidate for parallelization. The main difficulty in parallelizing this algorithm lies in the fact that the inner limits are dependent on the previous iteration values. Our algorithm avoids this recursive dependency by using a flattened generator.

Firstly, we need to know the total number of threads. It is:

$$k = k_1 \cdot k_2 \cdot k_3$$

where

$$k_1 = \sum_{i=2}^{n_{1\text{max}}} n_{1\text{max}}(R_1)$$ \text{ number of iterations for loops 1-3 in the sequential algorithm;}

$$k_2 = \sum_{i=0}^{n_{2\text{max}}} n_{2\text{max}}(R_1, R_2) \frac{(R_1^2 + R_1^4)(R_2^2 + R_2^4)}{4}$$ \text{ number of iterations for loops 4-8;}

$$k_3 = \left( \frac{2R_1^3 + 3R_2^2 + R_{\text{max}}^2 + 6R_{\text{max}}^5 + 15R_{\text{max}}^4 + 10R_{\text{max}}^3 - R_{\text{max}}^2}{30} - 2 \right) / 2$$ \text{ number of iterations for loops 9-11.}

Now, we need to know the values of all parameters for each thread. The special preparation was made before to run the main kernel where we calculate the porosity and particle-size distribution. We created the arrays of values $R_1, n_1, R_2, n_2$ for each work-item for the main kernel.

All the remaining variables are determined in each work-item.

The number of solids for the first generator (loop 5) is:

$$S_1 = \frac{- (2R_1^2 + 1) - \sqrt{(2R_1^2 + 1)^2 - \frac{4c}{k_3(R_2^2 + R_2^4)}}}{2}$$

c is the remainder from dividing the total number of possible combinations $k$ to number of possible combinations of variable $(R_1, n_1, R_2, n_2)$.

The number of pores for first generator (loop 6) is:

$$P_1 = c - \frac{S_1(2R_1^2 - S_1 + 1)k_3(R_2^2 + R_2^4)}{2k_3(R_2^2 + R_2^4)}$$

The number of solids for second generator (loop 7):

$$S_2 = \frac{- (2R_2^2 + 1) - \sqrt{(2R_2^2 + 1)^2 - \frac{8c}{k_3}}}{2}$$

The number of pores for second generator (loop 8):

$$P_2 = c - \frac{S_2(2R_1^2 - S_1 + 1)k_3}{2k_3}$$
The number of subdivisions for third generator (loop 9):

\[ R_3 = \left( \frac{2(R_3^3 + 3R_3^2 + R_3)}{6} + \frac{6R_3^5 + 15R_3^4 + 10R_3^3 - R_3}{30} - 2 \right) / 2 \]

The number of solids for third generator (loop 10):

\[ S_3 = -\left(2R_3^2 + 1\right) - \sqrt{\left(2R_3^2 + 1\right)^2 - \frac{8c}{k_3}} \]

The number of pores for third generator (loop 11):

\[ P_3 = c - S_3\left(2R_3^2 - S_3 + 1\right) \]

4. Results and discussion

This parallel algorithm was tested on a 6 GB, 448 cores Tesla C2070 GPU (CUDA SDK 4.0) from Nvidia [16]. For portability, we chose the OpenCL (Open Computing Language) framework [17], which offers an abstract view of the parallel architecture used. This allowed us to use both the CPU (Central Processing Units) and the GPU backend. Comparison of computation time for CPU and GPU is presented in table 3. The CPU used is an Intel Xeon E5506 processor running at 2.13GHz. The computation time for GPU includes the runtime kernel compilation time.

| \( R_{max} \) | Number the patterns | Time CPU | Time GPU |
|---------------|---------------------|----------|----------|
| 3             | 10943625            | 8 sec    | 1.5 sec  |
| 4             | 310388561           | 18 min   | 4.5 sec  |

Using this algorithm, we achieved a significant reduction of computation time. It is now possible to increase the maximal scaling factor \( R_{max} \) beyond these values and to explore even more combinations in order to select the best one within a reasonable amount of time. This was our first attempt at adapting our code to innovative parallel architectures such as GPU. We are satisfied with the results we obtained and we look for others perspectives.

A soil sample can contains billions of particles. At every time step, all of these particles interact with each other with water flowing between them. The parameters that characterize this flow are also constantly changing in response to these interactions. Thus, the number of operations needed to compute one simulated second for a single cubic centimeter of soil is immense. Our tests have shown that the method proposed here offer a good potential to adapting to parallel architectures.

Use of simulation environments such as our virtual laboratory opens new and interesting possibilities. We are currently investigating the possibility of (i) computing the motion of particles under water flow in real time, (ii) increasing of the number of particles or including finer particles, (iii) increasing the accuracy of the results for the Navier-Stokes equations, (iv) adding new physical phenomena in our model such as the thermo effects.

References
[1] Indraratna B., and Vafai F (1997) “Analytical Model for Particle Migration Within Base Soil – Filter System”, Jour. Geotech. & GeoEnv. Eng., ASCE, vol 123, No 2, pp. 100-109
[2] Witt K. (1993) “Reliability Study of Granular Filters”, Filters in Geotech. & Hydr. Eng., Braun, Heibaum &Schuler (eds.) 1993 Balkema, Rotterdam, pp. 35-41
[3] T.C. Kenney, R.Chanal, E. Chiu, G.I. Ofoegbu, G.N. Omange and C.A. Ume (1985) “Controlling constriction size of granular filters”, Can. Geotech. J, 22, pp. 32-43.
[4] Perrier, E., Bird, N., and Rieu, M. (1999) “Generalizing the fractal model of soil structure: the pore-solid fractal approach.” Geoderma, Vol. 88: pp. 137-164.
[5] Mandelbrot, B.B. 1983 the fractal geometry of nature, W.H. Freeman, New York
[6] Friesen, W.I., Mikula, R.J., (1987) fractal dimensions of coal particles. J. Colloid interf. 120, pp 263-271
[7] M.A. Soto and O.M. Vilar “Evaluation of a Pore Fractal Model for the Prediction of Soil water retention Curve”, unsaturated Soils (2006), pp. 2441-2452
[8] Toledo, P.G., Novy, R.A., davis, H.T., Scriven, L.E., 1990. Hydraulic conductivity of porous media at low content. Soil Sci. Soc. Am. J. 54, 673-679
[9] Lehmann, P., Stähli, M., Papritz, A., Gygi, A., and Flühler, H. (2003). "A Fractal Approach to Model Soil Structure and to calculate thermal conductivity of soils," Transport in porous media, Vol. 52: pp. 313-332.
[10] Edited by Y. Pachepsky, J.W. Crawford, W.J. Rawls “Fractals in soil sciences” 2000, ELSEVIER, 295 p.
[11] Goldin, A. L. and Rasskazov, L. N. (1992). Earth Dam Design, Geotechnika 2. Balkema, Rotterdam, 328 p.
[12] L.M. Arya, F.J. Leij, P.J. Shouse and M.Th. van Genuchten “relation between the Hydraulic Conductivity function and the particle-Size distribution” (1999), Soil Sci. Soc. Am J 63: 1063-1070
[13] Chekired M. and Roubtsova V. (2010) “2-D pore-particle scale model of the erosion at the boundary of two soils under horizontal groundwater flow”, 5th International Conference on Scour and Erosion ICSE-5
[14] Scheidegger, A. E. (1963). "The Physics of Flow Through Porous Media," University of Toronto Press, Toronto.
[15] Vukovic, M. and Soro, A. (1992). "Determination of Hydraulic Conductivity of Porous Media from Grain-Size Composition." Water Resources Publications, Littleton, Colorado, CO, 83p.
[16] NVIDIA Corporation (2009) NVIDIA’s Next Generation CUDA Compute Architecture: Fermi, NVIDIA Corporation.
[17] Munshi, A. (2011) The OpenCL Specification Version 1.1, The Khronos Group. 5. Mark, S. and Akeley, K., (2010) The OpenGL Graphics System: A SpecificationVersion 4.1 (Core Profile), The Khronos Group.