



Article

A Stochastic Filling and Modeling Algorithm of Non-Equal Diameter Particles with Specified Probability Density for Porous Permeable Materials

Wei Zhang ¹, Lile He ¹, Fazhan Wang ^{1,*} and Guangyong Zhang ²

¹ School of Mechanical and Electrical Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China; zhangweixjd@xauat.edu.cn (W.Z.); hllnh2013@163.com (L.H.)

² State Key Laboratory of Metal Extrusion and Forging Equipment Technology, China National Heavy Machinery Research Institute Co., Ltd., Xi'an 710018, China; zhanggy137@163.com

* Correspondence: wangfz1_01@163.com

Abstract: In this paper, a model generation algorithm for non-equal diameter particles with a specified probability density distribution is proposed. Based on considering the randomness of the size and distribution of the particles, the compact stacking of the particles is realized by the compactness algorithm, and then the spatial distribution of the tightly compacted particles is made to meet the random distribution of the specified probability density and the specified volume fraction by the filtering algorithm. The computational efficiency and effectiveness of the algorithm are verified, and the effects of the particle size and volume fraction on the distribution are analyzed. Finally, the proposed model has been used to study the permeability of a titanium porous filter cartridge. The results show that the size and location of the particle samples that are generated by the proposed algorithm follow specified probability distributions according to the requirements, and the volume fraction can be adjusted. Compared with the traditional algorithm, the computational effort and complexity are reduced. The resultant model can be used to study the permeability of porous materials and provide modeling support for structural optimization and further simulation of porous materials.

Keywords: non-equal diameter particles; specified probability density; random filling; permeability; porous materials



Citation: Zhang, W.; He, L.; Wang, F.; Zhang, G. A Stochastic Filling and Modeling Algorithm of Non-Equal Diameter Particles with Specified Probability Density for Porous Permeable Materials. *Materials* **2022**, *15*, 4733. <https://doi.org/10.3390/ma15144733>

Academic Editors: Miguel Jorge, Moises Luzia Pinto and Carlos Manuel Silva

Received: 13 May 2022

Accepted: 3 July 2022

Published: 6 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many new functional materials are bonded and compressed by granular materials with a pore structure that follows a certain probability distribution, such as titanium porous filter cartridge and metal-matrix composites. In order to obtain a mesoscopic mechanism for materials with a specific random distribution of particles, it is necessary to study the relationship between the properties of granular materials and the microstructure morphological characteristics of the formed materials, so as to carry out an effective structural performance design of the material [1,2]. However, because of the complex mesoscopic pore structure, it is difficult to realize the visualization of mesoscopic mechanism, for example, the seepage properties, through experiments [3]. Therefore, a meso-model with a specified probability density distribution is built to provide reference for improving the accuracy and efficiency of porous media meso-modeling.

As for the generation algorithm of particle distribution, Primera et al. [4] proposed the method of triangulation to describe the pore size, but this method was completed through two-dimensional slices and could not fully represent the three-dimensional morphology of the pores. Tory E M et al. [5] continuously filled the fixed space with the sequence addition method and completed the sequence filling of the powder layer based on considering the stability of the powder particles and the interaction force between the

particles. Jodrey W S et al. [6] used the non-sequence rearrangement algorithm to start position allocation with large-diameter particles first, allocating the position with small-diameter particles, and then reducing the overlap degree by moving particles. Cundall P A et al. [7] made the stacking process of particles similar to the flow process of dynamic fluid based on the discrete element method and realized the stacking of powder particles. A. Bezrukov et al. [8] describes two algorithms for the generation of random packings of spheres with arbitrary diameter distribution. The first algorithm is the force-biased algorithm. It produces isotropic packings of a very high density. The second algorithm is the Jodrey–Tory sedimentation algorithm, which simulates the successive packing of a container with spheres following gravitation. It yields packings of a lower density and of weak anisotropy. Although the above algorithms could achieve the stacking of powder particles, there existed some shortcomings in the randomness of spatial distribution, and the filling efficiency was low. At the same time, the above algorithms inevitably need to constantly detect the interference between particles, which means tedious calculation.

In addition, Bailakanavar M et al. [9] used the random order adsorption algorithm to determine whether the current particles interfered with the existing particles, retained the data of the non-interfered particles, and then established the relevant model. Xin Zhenyang et al. [10] used the perturbation algorithm to assign regular distribution positions to all particles and applied the perturbation to the amount of random distance between each particle. Through interference judgment, the position information of particles was retained. Note that through the above, the generation algorithms could ensure the randomness of the space to a certain extent. Due to the need to compare the position relationship between the specified particles and all the particles in the distribution process, the computation burden is somewhat high, which affects the efficiency of the method.

The transmission characteristics of spherical packing include electrical conductivity and permeability, among which permeability is one of the most studied properties [11–13]. Experimental and theoretical studies on permeability have achieved certain results, such as Darcy’s Rule for fluid flow in porous media and the empirical formula Kozeny relation for permeability [14]. As the pore structure of sphere accumulation is complex like that of other porous media, it is difficult to study fluid flow in porous media by traditional methods [15].

To solve the above problems, this paper proposes a method to generate the meso-model of non-equal diameter particles with a specified probability density distribution. The filling method of particles avoids the tedious calculation that is caused by interference detection, and since the movement of particles is only related to a small number of particles in the neighborhood, the modeling efficiency is further improved to a significant degree. Through the comparison of computation with the traditional methods and a hypothesis test, the efficiency and effectiveness of the algorithm are verified, illustrating that the proposed generation algorithm could effectively build a meso-model for porous media and titanium foam with a specified probability density distribution. The lattice Boltzmann method is used to calculate the permeability of porous material with different probability density distributions. The results show that the porous material meso-model that is established by the proposed algorithm in this paper can provide a basis for further numerical simulation of fluid permeability.

2. Generation Algorithm for Meso-Distribution Particles

Considering the randomness of the particles in size and distribution, Python is employed to build the algorithm, and gives the generation steps of the particles a meso-model, according to the specified probability density. The proposed algorithm consists of two parts. The first part is the compactness algorithm, which causes the particles with the size of a specified probability density distribution to stack closely together. The second part is the filtering algorithm. By using this method, the distribution of the particles in the spatial position according to the specified probability density can be realized, and the specified volume fraction can be achieved based on the requirements.

2.1. Compactness Algorithm for Particles

Let i represent the direction of the x axis, j represent the direction of the y axis, and k represent the direction of the z axis, respectively, corresponding to columns, rows and layers. The particles of random sizes are generated in the cube area according to the direction of the column, row and layer successively, and the tight compactness is realized by adjusting the position between the particles.

2.1.1. Formation of Non-Equal Diameter Particles

According to the actual demand, the particle size is randomly distributed according to the specified probability density by defining an appropriate sphere diameter range.

The expectation of X_r is

$$E(X_r) = \int_A \mu f d\mu$$

In practice, the algorithm generates particles with a radius that is randomly distributed on the interval $[a, b]$; the average of the radius is

$$r_{avg} = E(X_r) = \int_a^b xf(x)dx$$

2.1.2. Determine Initial Position

At this step, the algorithm places the random sized particles roughly in a 3D matrix pattern, with particle positions vibrated by random noises. To place the first particle with radius r_1 near coordinates $(0, 0, 0)$, the algorithm considers the particle radius, the cube area boundary restriction and the random noise. The resulting center coordinates of the particle are noted as (x_1, y_1, z_1) , where $x_1 = r_1 + d_{noise}$, $y_1 = r_1 + d_{noise}$ and $z_1 = r_1 + d_{noise}$; the second particle with radius r_2 is placed at coordinates (x_2, y_2, z_2) , where $x_2 = x_1 + r_1 + r_2 + d_{noise}$, $y_2 = r_2 + d_{noise}$, $z_2 = r_2 + d_{noise}$. the third particle with radius r_3 is placed at $(x_2 + r_2 + r_3 + d_{noise}, r_3 + d_{noise}, r_3 + d_{noise})$, etc. The notation d_{noise} stands for the random noise that the algorithm introduces; it takes a different value each time it appears. The particle placement process continues until the first row on the xy -plane is filled. Next, the algorithm places the second row close to the first row on the xy -plane and the row placement process repeats until the first xy -layer is filled. Afterwards, the second particle layer is placed closely above the first layer and the layer placement process repeats until there are enough layers. Prior to the placement of any particles, the algorithm estimates the number of particles to generate by considering the scenario where the cube area is filled with particles of equal radius r_{avg} . Assume the length of the edges of the cube along the x, y, z -axis are L_x, L_y and L_z , respectively; the number of particles in each row, the number of rows in each layer and the number of layers are estimated as $N_x = L_x / (2r_{avg} + \rho)$, $N_y = L_y / (2r_{avg} + \rho)$ and $N_z = L_z / (2r_{avg} + \rho)$, respectively. The notation ρ stands for the upper bound of the gap distance between adjacent particles and is enforced by the compactness step (Section 2.1.3). Since the density of the particles is increased by the compactness step and the goal is to distribute the particles in the entire cube area following a given probability density, the estimated values N_x, N_y and N_z are further scaled up by an overfill factor $f_{ov} > 1$. After the compactness step, the excessive particles that remain outside of the cube boundaries are easily pruned.

To ensure that each newly generated particle and the last generated particle are adjacent in the same row and layer, the adjacent scales vibrate within a very small range. The advantage of this method is that the detection of particle interference can be omitted, and thus the randomness of the position can be retained, which significantly simplifies the computational effort of the algorithm. The logical steps are as follows:

(1) By specifying a minimal disturbance $init_noise$ to add a segment of white noise, its power spectral density will be constant in the whole frequency domain, as shown in Figure 1 (when $i = 1$, the x coordinate of the center is evaluated on $[x_min, x_min + init_noise]$ in a uniformly distributed scale, where x_min is simply the radius of the particle and $init_noise$

is a preselected constant to control the amplitude of the noise). Otherwise, make the minimum coordinate x_{min} of the newly generated particles in row j of the k th layer equal to the *previous maximum x* coordinate $prev_x_max$, plus the radius of the new particle, before taking the corresponding disturbance distance.

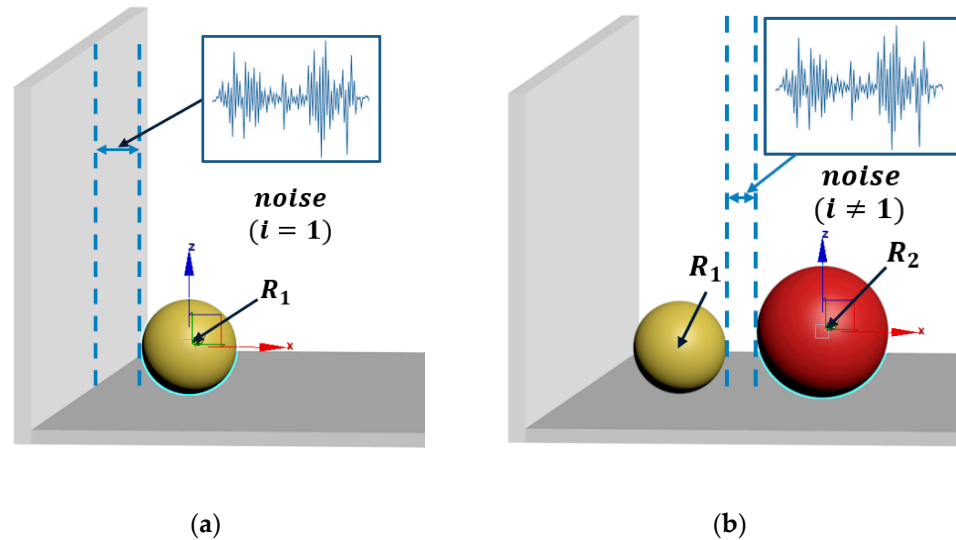


Figure 1. Principle of initial position determination. (a) The position of new particle generated ($i = 1$). (b) The position of new particle generated ($i \neq 1$).

(2) Similarly, when $j = 1$, the y coordinate of the center is evaluated at a uniformly distributed scale on $[y_{min}, y_{min} + init_noise]$, where y_{min} is simply the particle radius. Otherwise, since the $(j - 1)$ th row particles in layer k have been determined, the minimum coordinate y_{min} of the newly generated particles in layer k should be equal to the maximum y coordinate $prev_y_max$ of the $(j - 1)$ th row, plus the radius of the new particle, before taking the corresponding disturbance distance.

(3) Similarly, when $k = 1$, the z coordinate of the center is evaluated on $[z_{min}, z_{min} + init_noise]$, where z_{min} is simply the radius of the particle. Otherwise, since the particles at the $(k - 1)$ th layer have been determined, the minimum coordinate z_{min} of the newly generated particles should be equal to the maximum z coordinate $prev_z_max$ ($k - 1$)th layer, plus the radius of the new particle, before taking the corresponding disturbance distance.

If the amplitude of the noise is 0.1 and the mean value is 0.05, when the new particle has been generated 1000 times, the graph of the noise can be obtained, as shown in Figure 2.

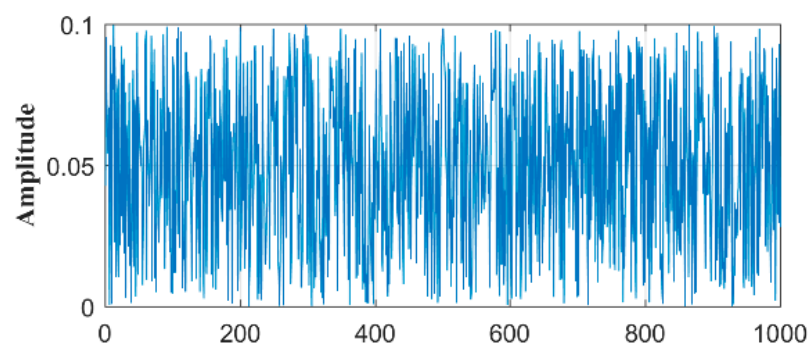


Figure 2. White noise waveform.

The pseudocode of the placement process is as follows in Algorithm 1.

Algorithm 1 Algorithm of initial position determination

```

for k in [1, fov*Nz]
  z_max =  $-\infty$ 
  for j in [1, fov*Ny]
    y_max =  $-\infty$ 
    prev_x_max =  $-\infty$ 
    for i in [1, fov*Nx]
      generate particle with random size r
      if i = 1 then
        x_min = r
      else
        x_min = prev_x_max + r
      if j = 1 then
        y_min = r
      else
        y_min = prev_y_max + r
      if k = 1 then
        z_min = r
      else
        z_min = prev_z_max + r
      x_new = u(x_min, x_min+init_noise)
      y_new = u(y_min, y_min+init_noise)
      z_new = u(z_min, z_min+init_noise)
      place the newly generated particle at (x_new, y_new, z_new)
      prev_x_max = max(prev_x_max, x_new+r)
      y_max = max(y_max, y_new+r)
      z_max = max(z_max, z_new+r)
    prev_y_max = y_max
  prev_z_max = z_max

```

In the pseudo code, $u(a, b)$ generates a uniformly distributed random value on the interval $[a, b]$.

2.1.3. Compactness of Particles

Assume that the randomly generated particles are stored in a 3-dimensional array M , whose size is $I_{max} \times J_{max} \times K_{max}$, $n_{neighbor}$ is a small positive integer, and ρ is a small positive value. The position of the particles will be repositioned by coordinate translation.

By calculating the distance dz between the particle and the $n_{neighbor}$ neighboring particles in layer $(k - 1)$, local neighborhood compression can be achieved to speed up the iterative process and more tightly compact the particles. While dy is the distance between the particles and the particles in the $n_{neighbor}$ range that are located in row $(j - 1)$ of layer k , dx is the distance between the particle and the particles in the $n_{neighbor}$ range that are located in row j , column $(i - 1)$ of layer k . Let $dw = \text{Max}(dx, dy, dz)$ ($w \in x, y, z$). Move the particle by $dw/3$ in the w direction. Repeat this step until $dx, dy, dz < \rho$.

The center coordinate of the particle $P = (x, y, z)$ is reduced by $1/3dw$, $dw = \text{Max}(dx, dy, dz)$ ($w \in x, y, z$), and a new position $P' = (x', y', z')$ is obtained. The coordinates of the new position are derived as follows.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & x \\ 0 & -1 & 0 & y \\ 0 & 0 & -1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{3} \begin{bmatrix} dw_x \\ dw_y \\ dw_z \\ 3 \end{bmatrix}$$

According to the aforementioned methods, the particles at the initial position can be further compressed in distance to be tightly compacted.

The pseudocode of the compactness process is as follows in Algorithm 2.

Algorithm 2 Compactness algorithm

```

for  $k$  in  $[0, K_{max}-1]$ 
  for  $j$  in  $[0, J_{max}-1]$ 
    for  $i$  in  $[0, I_{max}-1]$ 
       $x_{boundary} = \max(x \mid x \text{ is the } x \text{ coordinate of points on the particles at } M[i-1, j, k])$ 
       $y_{boundary} = \max(y \mid y \text{ is the } y \text{ coordinate of points on the particles at } M[i, j, k],$ 
         $\text{where } j_n \text{ in } [j-n_{neighbor}, j+n_{neighbor}])$ 
       $z_{boundary} = \max(z \mid z \text{ is the } z \text{ coordinate of points on the particles at } M[i, j, k_n],$ 
         $\text{where } j_n \text{ in } [j-n_{neighbor}, j+n_{neighbor}] \text{ and } k_n \text{ in } [k-n_{neighbor}, k+n_{neighbor}])$ 

    Assume:
      Plane  $P_x$  parallel to the  $yz$ -plane intersects the  $x$ -axis at  $x_{boundary}$ .
      Plane  $P_y$  parallel to the  $xz$ -plane intersects the  $y$ -axis at  $y_{boundary}$ .
      Plane  $P_z$  parallel to the  $xy$ -plane intersects the  $z$ -axis at  $z_{boundary}$ .
       $d_x = \text{distance between } M[i, j, k] \text{ and } P_x$ 
       $d_y = \text{distance between } M[i, j, k] \text{ and } P_y$ 
       $d_z = \text{distance between } M[i, j, k] \text{ and } P_z$ 
      while  $\max(d_x, d_y, d_z) \geq \rho$ 
        if  $d_x = \max(d_x, d_y, d_z)$ 
          subtract  $d_x/3$  from the  $x$  coordinate of  $M[i, j, k]$ 
        else if  $d_y = \max(d_x, d_y, d_z)$ 
          subtract  $d_y/3$  from the  $y$  coordinate of  $M[i, j, k]$ 
        else  $d_z = \max(d_x, d_y, d_z)$ 
          subtract  $d_z/3$  from the  $z$  coordinate of  $M[i, j, k]$ 
        Recalculate  $d_x, d_y, d_z$ 
      if loop has executed more than 1000 times:
        Report exception and quit
  
```

Figure 3 is the compactness model diagram that is formed after the implementation of the compactness algorithm. The particles whose size complies with the specified distribution law are packed tightly in the cube region.

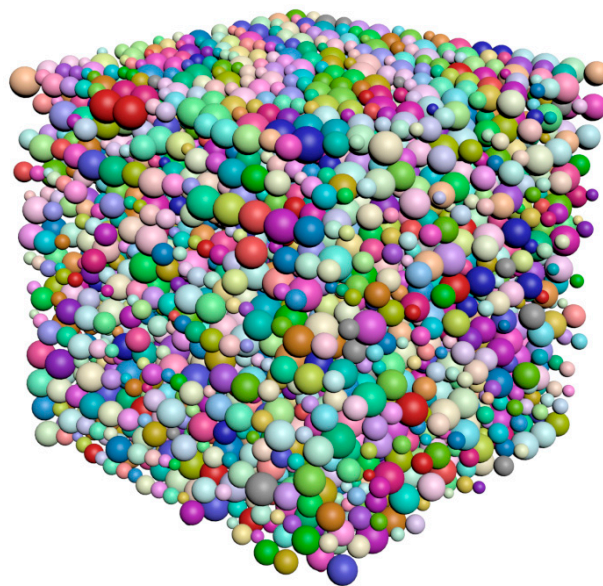


Figure 3. Results of compactness algorithm execution.

2.2. Filtering Algorithm

After the compactness algorithm, the particles with a random size distribution can be tightly compacted and a higher volume fraction of bulk density can be obtained. However, the spatial distribution cannot meet the random distribution of the assigned probability density, and the volume fraction cannot be adjusted according to the demand. Filtering algorithms help with this process. This paper makes the following assumptions:

(1) Assuming that the volume of the cube in which the particles are located is v , such that the particles are uniformly distributed, the probability density is the same throughout the cube;

$$u_{x,y,z} = 1/v$$

(2) After the execution of the compactness algorithm and before the execution of the filtering algorithm, the total volume of the particles is v_g ;

(3) If the target volume fraction is α , the total volume of the particles corresponding to this volume fraction is αv ;

(4) The target probability density function is $f_{x,y,z}$.

Given that a small zone with volume dv is centered at (x, y, z) in the cube, the probability of retaining the particles near (x, y, z) by the filtering algorithm is

$$s_{x,y,z} = \frac{f_{x,y,z} \alpha v dv}{u_{x,y,z} v_g dv}$$

Substitute the hypothesis, simplify it, and we can achieve

$$s_{x,y,z} = \frac{\alpha v^2}{v_g} f_{x,y,z}$$

Assuming that the tightly packed particles are stored in a three-dimensional array $I_{max} \times J_{max} \times K_{max}$, after using the aforementioned methods, the particles that do not conform to the given probability density distribution in space are filtered out.

The pseudocode is as follows in Algorithm 3.

Algorithm 3 Filtering algorithm

```

for  $k$  in  $[0, K_{max}-1]$ 
  for  $j$  in  $[0, J_{max}-1]$ 
    for  $l$  in  $[0, I_{max}-1]$ 
      Let  $x,y,z$  be the coordinates of the object  $M[i,j,k]$ 
      if  $s > \alpha v^2 / v_g \cdot f_{x,y,z}$ 
        Delete  $M[i,j,k]$ 

```

Figure 4 shows examples of common probability density functions (uniform, normal and exponential), as generated by the filtering algorithm.

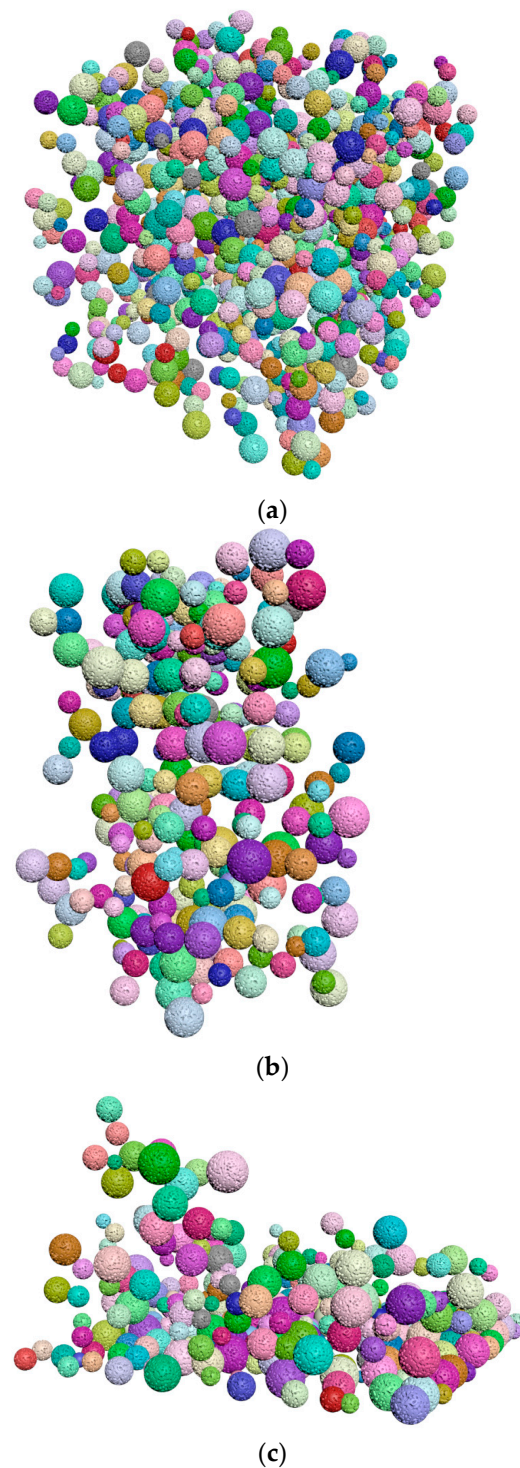


Figure 4. The result of the filtering algorithm execution. (a) Uniform distribution along x , y , and z , (b) Normal distribution along the x direction, (c) Exponential distribution along the z direction.

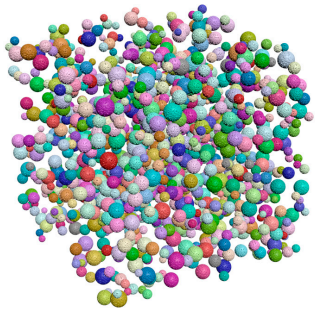
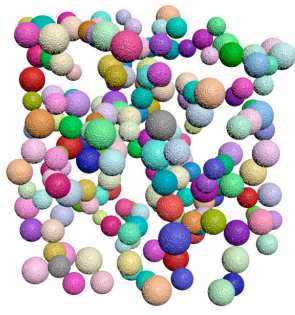

3. Influence of Parameters on the Algorithm

The parameters of the algorithm directly affect its performance. In this section, the influence of the design parameters on particle filling and distribution effect in the application of this algorithm are discussed. The design parameters mainly include the particle size and the volume fraction. This section presents the distribution and filling effect and summarizes the related influence.

3.1. Distribution of Particles with Different Size Ranges with Same Probability Density

Taking the uniform distribution in the x , y and z directions as an example, the distribution of the particles with different size ranges is simulated by using the proposed algorithm. The volume fraction is 5%, while the volume of the cube is 125 mm³. The total volume of the particles is equal. The distribution results are shown in Table 1.

Table 1. Distribution effects of particles with different size ranges with the same probability density.

Range of Particle Sizes	The Distribution of Particles
R = 0.05–0.15 mm	
R = 0.15–0.25 mm	
R = 0.25–0.35 mm	

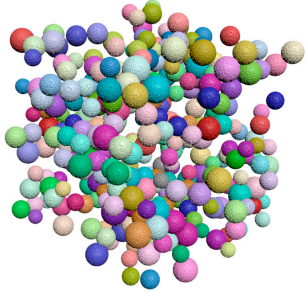
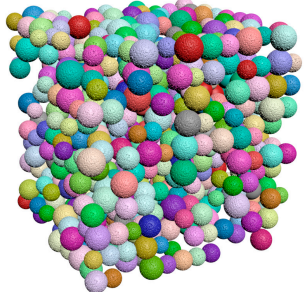
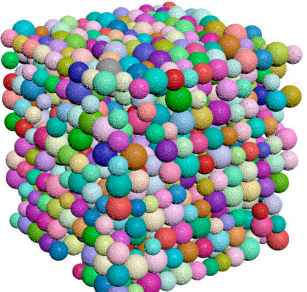
As can be seen from Table 1, when the volume fraction remains unchanged, the number of particles decreases significantly with the increase in particle size, and the distribution becomes sparser.

3.2. Particles Distribution with Same Probability Density and Different Volume Fraction

In this section, the uniform distribution in the x , y and z directions is taken as an example to simulate the distribution of particles in different volume fraction ranges by using the proposed algorithm. The radius sizes of the particles are all between 0.15–0.25 mm. The total volume of the cube is 125 mm³. The distribution results are shown in Table 2.

As can be seen from Table 2, when the radius size of the particles remains unchanged, the volume of the particles increases with the increase in volume fraction, but the maximum volume fraction of the particles cannot exceed the volume fraction of the tightest compactness that is formed by the compactness algorithm.

Table 2. Distribution effects of particles with the same probability density and different volume fractions.

Volume Fraction	Total Volume (mm ³)	The Distribution of Particles
10%	12.550	
20%	25.504	
30%	37.743	

4. Algorithm Analysis

The effectiveness of the algorithm is analyzed in two regards, namely, the efficiency and the randomness. First, the efficiency of the algorithm is demonstrated from the perspective of computational complexity. Secondly, the sample that is generated by the algorithm is used to verify the randomness of the random algorithm in particle size and location by the statistical hypothesis testing method.

4.1. Computational Efficiency

Taking the uniform distribution as an example, it is assumed that n particles obey the random distribution, and the efficiency analysis of the take-and-place algorithm and the algorithm proposed in this paper is as follows:

(1) The major steps of the take-and-place algorithm are as follows: randomly generate the position of the new particle and determine whether there is any interference between the current particle and the existing particles. If there is no interference, the relevant data of the current particle are saved, and the particle is accepted; otherwise, the algorithm needs to try new locations again and again. Therefore, the take-and-place algorithm not only reduces the randomness of sample generation, but also comes with the computational complexity of $c_1 (m_1 + 2m_2 + \dots + (k-1)m_k + \dots + (n-1)m_n)$ that is far greater than $O(n^2)$. To clarify, m_k is the number of repeated generation positions when filling the k th particle and a positive constant c_1 is independent of n ;

(2) As mentioned in Section 2, the algorithm that is proposed in this paper consists of two parts. In the first part of the algorithm, the position of each particle is only related to a small number of particles in its neighborhood when the initial position is generated or the tightness of the particles is adjusted. Therefore, the computational complexity is $c_1' n_1$, where n_1 is the number of particles and a positive constant c_1' is independent of n_1 . In the second part of the algorithm, each particle is independently processed, and the complexity is $c_2' n_1$, where c_2' is a constant independent of n_1 . Accordingly, the overall computational complexity of our algorithm is $O(n)$.

The computational complexity of the particle random generation algorithm that is proposed in this paper is significantly lower than that of the traditional algorithm, and it is highly efficient.

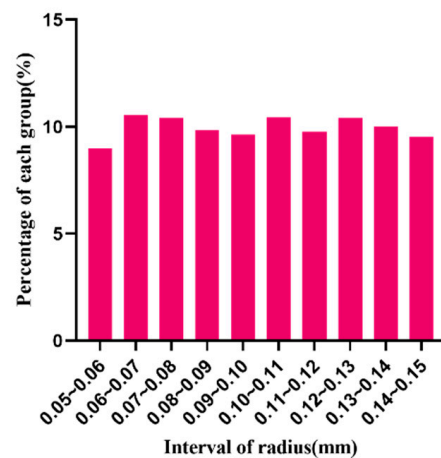
4.2. The Randomness of Generated Samples

4.2.1. Random Test of Particles Size

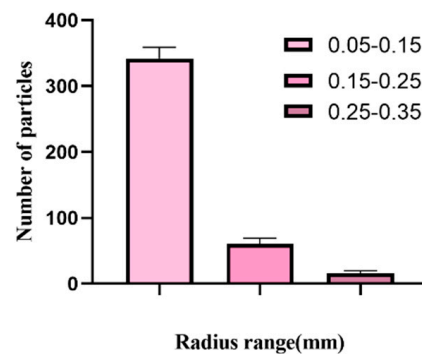
At present, most of the simulations focus on the case of uniform and random distribution, so the random characteristics of the samples with uniform and random distribution are studied, and an χ^2 test [16] is used to evaluate the uniformity of the samples. The statistic test formula is

$$\chi^2 = \frac{m}{\epsilon} \sum_{i=1}^m \left(\epsilon_i - \frac{\epsilon}{m} \right)^2$$

where ϵ is the number of all random numbers, m is the number of intervals, and ϵ_i is the number of the i th interval. In the uniformity test of the algorithm, we generated 3485 samples whose radii are on the interval (0.05, 0.15). The percentages of each radius are shown in Figure 5.



(a)



(b)

Figure 5. Distribution of particle radius in each interval. (a) Percentage of particles in each radius, (b) The number and standard deviation of particles within each radius.

The particles sizes are evenly distributed in each interval. Assuming that the degree of freedom is 9, the calculated χ^2 value of radius R is 11.72, and the asymptotic significance is 0.230, greater than 0.05. According to statistic theory, the χ^2 distribution table shows that the distribution of particle radius R that is obtained by the algorithm can be considered as an ideal uniform random distribution. The test statistical scale is shown in Table 3.

Table 3. χ^2 test statistical table.

Test Statistics	
Statistic	Value
χ^2	11.720
Degrees of freedom	9
Asymptotic significance	0.230

4.2.2. Random Test of Particle Position

Taking normal distribution as an example, the Shapiro–Wilk test method [17] and the Kolmogorov–Smirnov test method [18,19] are used to test the normality of the samples that are generated by the random algorithms, using the compactness algorithm to make the spherical center coordinates of 301 particles normally distributed along the x-axis. Table 4 is the analysis result of the normality test.

Table 4. Normal test analysis results.

Sample Size	Average	Standard Deviation	Partial Degrees	Kurtosis	Kolmogorov–Smirnov Test		Shapiro–Wilk Test	
					The Statistic D Value	p	The Statistic W Value	p
301	3.014	0.511	−0.028	−0.219	0.029	0.742	0.996	0.610

As can be seen from Table 4, the sample data did not show statistical significance ($p > 0.05$), which means that the hypothesis (hypothesis: the data are normally distributed) is accepted and the sample has the characteristic of normality. The histogram of the sample distribution along the x-axis is shown in Figure 6.

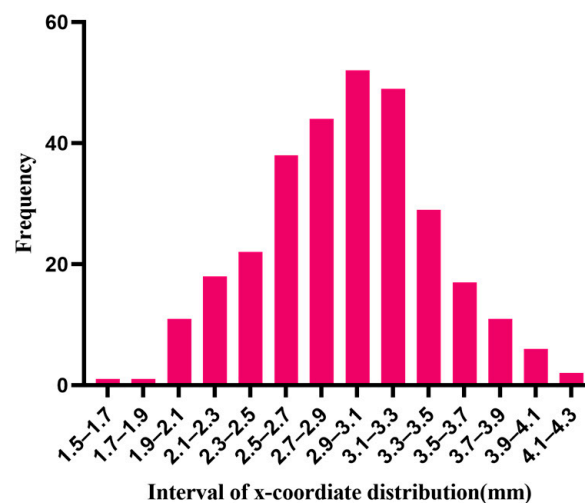


Figure 6. Particle coordinate distribution along the x direction.

5. Permeability of Porous Media

Titanium porous filter cartridge is made of porous titanium metal filter material by the powder metallurgy method. Its internal pores are curved and crisscross, and the filtering

mechanism is typical deep filtration. Permeability is the ability of a porous medium to allow fluid matter to pass through. In this section, the lattice Boltzmann method is used to study the flow of fluid under the condition of mass particle filling and the influence of initial particle distribution on fluid permeability by using the meso-model of porous media that is constructed by the above algorithm.

5.1. Lattice Boltzmann Method

The motion of a fluid can be described by a set of partial differential equations, such as the Navier–Stokes equations, which are highly nonlinear in most cases and find it very difficult to obtain analytical solutions. The Lattice Boltzmann method is used to solve the numerical solution of the fluid motion equation by means of the discrete method. The lattice Boltzmann method can be regarded as a special discrete scheme of a continuous Boltzmann equation, as shown in the following formula.

$$g_i(x + e_i\delta_t, t + \delta_t) - g_i(x, t) = \Omega_i(x, t)$$

where g is the discrete distribution function, e is the velocity space, i is the type of velocity, δ_t is the discrete time step, t is the current time step, x is a point on the grid, and Ω is the change caused by collision.

According to the operator that is proposed by Bhatnagar, Gross, and Krook [20], $\Omega_i(x, t)$ can be replaced, as shown in the following formula.

$$g_i(x + e_i\delta_t, t + \delta_t) - g_i(x, t) = \frac{1}{\tau} (g_i^{eq} - g_i)$$

where g_i^{eq} is an equilibrium distribution function to be determined; τ is the relaxation time.

The D_dQ_m model that is proposed by Qian et al. [21] is the basic model of the lattice Boltzmann model, where d represents d -dimensional space and m represents the number of discrete velocities of the lattice. In this paper, the velocity is discretized into 19 directions in three-dimensional space, as shown in Figure 7.

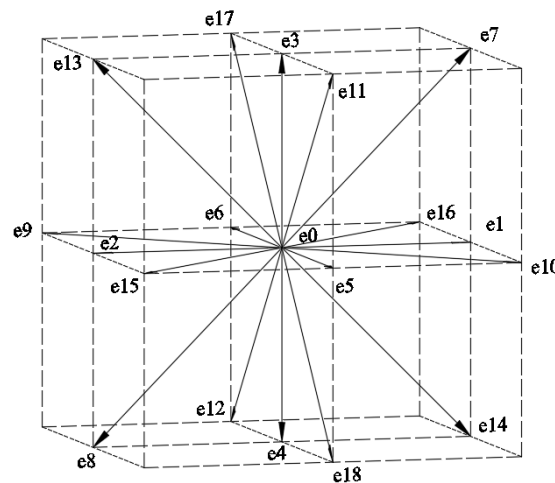


Figure 7. Discrete velocity diagram of lattice points.

The discrete component of the velocity is

$$E = [e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7, \dots, e_{18}]$$

The equilibrium equation [21] of this model is

$$g_i^{eq} = \rho_m \omega_i \left[1 + \frac{e_i \cdot \mu}{c_s^2} + \frac{(e_i \cdot \hat{\sigma})^2}{2c_s^4} - \frac{\hat{\sigma}^2}{2c_s^2} \right] \quad i = 0, 1, 2, \dots, 18$$

where C_s is the lattice speed, ω is the weight coefficient, $\hat{\sigma}$ is the fluid velocity, and ρ_m is the fluid density. The lattice velocity C_s is as follows.

$$C_s = \frac{c}{\sqrt{3}}$$

where c is the ratio of grid step to time step, and its value is 1. The calculation of ω_i is shown in the following formula.

$$\omega_i = \begin{cases} 1/3 & e_i^2 = 0 \\ 1/18 & e_i^2 = c^2 \\ 1/36 & e_i^2 = 2c^2 \end{cases}$$

Let $g_i(x, t)$ be the distribution function of time t , at lattice point x , velocity e , then the evolution equation of the distribution function is

$$\begin{aligned} & g_i(x + e_i \delta_t, t + \delta_t) - g_i(x, t) \\ &= \frac{1}{\tau} \left(g_i^{eq}(x, t) - g_i(x, t) \right) \end{aligned}$$

5.2. Darcy's Law

When single-phase flow flows in porous media at a low Reynolds number, it follows Darcy's Law, also known as the Darcy model, which is one of the most basic and commonly used mathematical models for macroscopic seepage, as shown in the following equation

$$\sigma = -\frac{h}{\nu} \nabla p_l$$

where σ is the Darcy velocity, h is the permeability of porous media, ν is the universality coefficient of the fluid, p_l is the fluid pressure, and ∇ is the Hamiltonian operator.

According to the distribution function, the macroscopic parameters of the fluid density ρ_l and fluid velocity σ can be obtained from formula

$$\rho_l = \sum_i g_i \sigma = \frac{1}{\rho_l} \sum_i g_i \cdot e_i$$

The permeability of porous media can be calculated according to Darcy's formula [22].

$$h = \frac{\nu \bar{\sigma}}{\Delta p_l}$$

where h is the permeability of the porous media, ν is the coefficient of motion universality, Δp_l is the pressure difference, and $\bar{\sigma}$ is the average speed.

5.3. Lattice Boltzmann Method Simulation Procedures

The lattice Boltzmann method simulation program structure is collision-migration. The specific process is as follows:

- (1) Setting of initial conditions;
- (2) Execute collision at time t ;
- (3) Boundary processing;
- (4) Calculate macroscopic quantities;
- (5) Check whether convergence exists. If not, return to Step 2. Otherwise, go to the next step;
- (6) Output the result.

Suppose the fluid flows in a $4.5 \times 4.5 \times 4.5$ mm cube tube, as shown in Figure 8. The fluid is the single-phase flow of a low Reynolds number and is incompressible. The same

number of lattice points are used in the x , y and z directions in the grid division of the stacking space. The flow direction of the fluid is in the z direction, the top and bottom surfaces are the fluid outlet and inlet surfaces, respectively, and the other surfaces are boundary surfaces. When the relaxation coefficient is fixed at 1, the universality coefficient is 10^{-6} . At the beginning, the velocity in the whole flow field is set at 0 and the density is 1. The method to judge the convergence is that in 50 time steps; if the change of the sum of the absolute values of the velocity along the direction of fluid flow is less than 0.0001, convergence is realized.

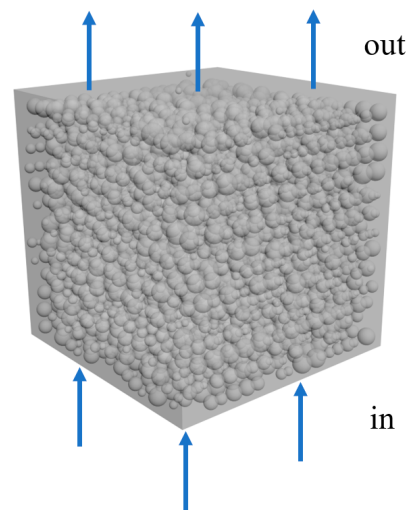


Figure 8. Diagram of flow field.

5.4. Simulation Results

5.4.1. The Influence of the Distribution Law

The model is divided into 1283 grid points, and each grid point has 19 velocity directions. The algorithm that is proposed in this paper is used to generate uniform distribution, normal distribution (mean is 3, and the standard deviation is 0.5) and exponential distribution, respectively (rate parameter λ is 2). The simulation results are shown in Table 5.

Table 5. Permeability and average velocity of different distribution.

Distribution	Average Velocity (m/s)	Permeability k (m ²)
Uniform	0.586×10^{-3}	5.204×10^{-10}
Normal	0.533×10^{-3}	5.335×10^{-10}
Exponential	0.681×10^{-3}	6.757×10^{-10}

The above table shows that the average velocity and permeability of exponential distribution are the largest. There is little difference between a uniform distribution and a normal distribution. It should be noted that this example only calculates permeability under different probability density distributions to verify the feasibility of the modeling algorithm. The parameters that are related to the probability density function, such as the direction of the distribution law and the parameters of the distribution function, have significant effects on permeability, and their effects are often coupled with each other. Therefore, further research is needed to establish a general strong law before these questions can be resolved. This modeling algorithm provides a model foundation and technical support for further study of these laws.

5.4.2. The Influence of the Numbers of Particles and Grids

In the case of the lattice Boltzmann method, both the number of lattice points and the volume fraction of the particles become significant influencing factors. On the premise that the particle distribution law is determined as uniform distribution, the permeability of the model is simulated for the following two cases: (1) the volume fraction of the particles is 0.55, 0.57, 0.61, 0.63, respectively, and the number of grid points stays the same as 1283; (2) the volume fraction is 0.60 and the number of grid points is 643, 1283, 2563, 5123, respectively. The simulation results are shown in Figures 9 and 10.

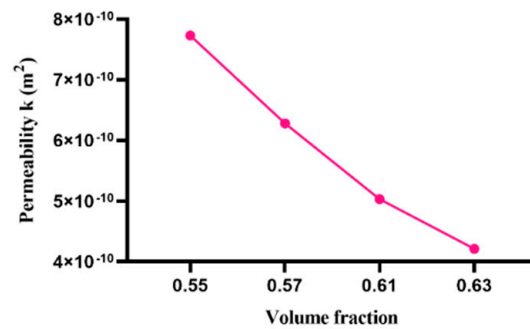


Figure 9. Effect of volume fraction on fluid permeability.

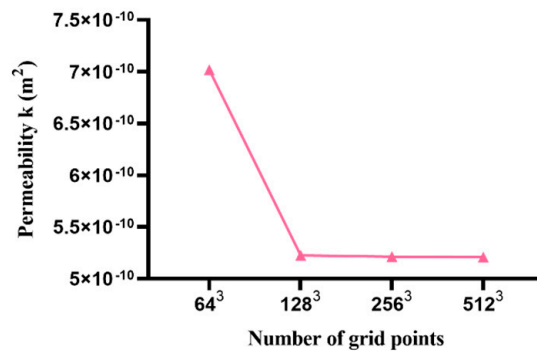


Figure 10. Effect of grid number on fluid permeability.

Through analytical methods [23], the relationship between the permeability and volume fraction of porous materials can be predicted by specific formulas. The permeability of porous materials is related to the minimum and maximum radius of the cross-section through the transport phase, the shortest path of transport and the volume fraction of the pores. The specific formula is as follows.

$$\hat{k} = (0.94 r_{min} + 0.06 r_{max})^2 \varepsilon^{2.14} \tau^{-2.44} / 8$$

where \hat{k} is the predicted value of permeability, r_{min} and r_{max} are the minimum and maximum radius of the cross-section through the transport phase, ε is the volume fraction of the pores, and τ is the shortest path of transportation.

It can be seen from the formula that the permeability of porous materials increases with the increase in the volume fraction of the pores. It is worth noting that the volume fraction of pore ε and the volume fraction of particle α are complementary, and the sum is 1. This is consistent with the change trend that is shown by the simulation.

As shown in Figure 9, the simulation results show that with the same distribution law, the permeability of the fluid through this distribution decreases with the increase in the volume fraction. Figure 10 shows that under the same distribution law, the permeability of the fluid through the same amount of particle decreases with the increase in the number of lattice points. When the lattice points increase to 1283 or higher, the permeability is almost unchanged.

6. Conclusions

In this paper, an efficient algorithm is proposed to generate the particle meso-model by specifying the probability density distribution. The following conclusions and innovations can be obtained:

1. The filling process avoids the huge calculation burden that is caused by the continuous iterative interference detection;
2. In the process of generating the initial position of the particles or realizing the compact stacking, the changes of each particle are only related to its small neighborhood, so it has a high compactness efficiency;
3. The computational complexity of the algorithm is first order, while that of the traditional algorithm is much larger than second order, which illustrates the computational efficiency of the proposed algorithm;
4. With this algorithm, the size and position of the particles can be distributed according to the arbitrary probability density based on the requirements, and the specified volume fraction can also be realized according to the requirements.

The results provide a new method for the space filling and modeling of porous materials and provide technical support for 3D printing when optimizing porous material structures. This method provides a modeling basis for further exploring the influence of porous materials with a specified probability density on the mesoscopic and multi-field simulation research of sound absorption, heat absorption, radiation resistance, seepage and extrusion resistance, which are difficult to observe by experimental methods.

Author Contributions: Methodology, W.Z. and F.W.; Software, W.Z. and L.H.; Data Analysis, L.H. and G.Z.; Supervision, L.H. and G.Z.; Writing—original draft, W.Z.; Writing—review & editing, F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Key Research and Development Projects [grant numbers 2022GY-399] of Shaanxi Province, China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Symbols and Abbreviations

f	probability density
M	3-dimensional array
ρ	a small positive value
$n_{neighbor}$	a small positive integer
v	volume of the cube
u	probability density
v_g	total volume of the particles
α	volume fraction
n	particle number
M_k	number of repeated generation position
c_1	a positive constant
ϵ	number of all random numbers
g	discrete distribution function
e	velocity space
δ_t	discrete time step

t	current time step
Ω	change caused by collision
C_s	lattice velocity
ω	weight coefficient
$\hat{\sigma}$	fluid velocity
σ	Darcy velocity
h	permeability of porous media
p_l	fluid pressure
∇	Hamiltonian operator
$\hat{\kappa}$	predicted value of permeability
λ	rate parameter of the exponential distribution

References

- Noubactep, C.; Caré, S.; Togue-Kamga, F.; Schöner, A.; Woafu, P. Extending Service Life of Household Water Filters by Mixing Metallic Iron with Sand. *Clean Soil Air Water* **2010**, *38*, 951–959. [[CrossRef](#)]
- Wang, W.; Parker, K.H. Effect of deformable porous surface layers on the motion of a sphere in a narrow cylindrical tube. *J. Fluid Mech.* **1995**, *283*, 287–305. [[CrossRef](#)]
- Sobieski, W. Waterfall Algorithm as a tool of investigation the geometrical features of granular porous media. *Comput. Part. Mech.* **2021**, *9*, 551–567. [[CrossRef](#)]
- Primera, J.; Hasmy, A.; Woignier, T. Numerical study of pore sizes distribution in gels. *J. Sol-Gel Sci. Technol.* **2003**, *26*, 671–675. [[CrossRef](#)]
- Tory, E.M.; Cochrane, N.A.; Waddell, S.R. Anisotropy in Simulated Random Packing of Equal Spheres. *Nature* **1986**, *220*, 1023–1024. [[CrossRef](#)]
- Jodrey, W.S.; Tory, E.M. Computer Simulation of Isotropic, Homogeneous, Dense Random Packing of Equal Spheres. *Powder Technol.* **1981**, *30*, 111–118. [[CrossRef](#)]
- Cundall, P.; Strack, O. Discussion: A discrete numerical model for granular assemblies. *Geotechnique* **1980**, *30*, 331–336. [[CrossRef](#)]
- Bezrukov, A.; Bargiel, M.; Stoyan, D. Statistical analysis of simulated random packings of spheres. *Part. Part. Syst. Charact. Meas. Descr. Part. Prop. Behav. Powders Other Disperse Syst.* **2002**, *19*, 111–118. [[CrossRef](#)]
- Bailakanavar, M.; Liu, Y.; Fish, J.; Zheng, Y. Automated modeling of random inclusion composites. *Eng. Comput.* **2012**, *30*, 609–625. [[CrossRef](#)]
- Xin, Z.; Miao, W.; Wang, Y.; Chen, H. Simulation of Tensile Fracture of ZrO₂ Toughened Al₂O₃ Particles Reinforced Fe₄₅ Composites by Finite-Discrete Element Coupling Method. *Acta Mater. Sin.* **2019**, *36*, 1471–1479. (In Chinese)
- Stevnl, B.; Peter, R.K.; David, M. Network Model Evaluation of Permeability and Spatial Correlation in a Real Random Sphere Packing. *Transp. Porous Media* **1993**, *11*, 53–70.
- Heijs, A.W.J. Numerical evaluation of the permeability and the Kozeny constant for two types of porous media. *Phys. Rev. E* **1995**, *51*, 4346–4353. [[CrossRef](#)] [[PubMed](#)]
- Bryant, S.; Blunt, M. Prediction of relative permeability in simple porous media. *Phys. Rev. A* **1992**, *46*, 2004–2011. [[CrossRef](#)] [[PubMed](#)]
- Raats, P.A.C. Dynamics of Fluids in Porous Media. *Soil Sci. Soc. Am. J.* **1973**, *37*, vi. [[CrossRef](#)]
- Zhu, J.; Xi, Z.; Tang, H.; Tan, P. Characterization of porous structures and fractal theory. *Rare Met. Mater. Eng.* **2006**, *35*, 452–456.
- Hosmane, B.S. Improved likelihood ratio tests and Pearson chi-squared tests for independence in two dimensional contingency tables. *Commun. Stat.* **1986**, *16*, 1875–1888. [[CrossRef](#)]
- Kline, R.; Kline, R.B.; Kline, R. Principles and Practice of Structural Equation Modelling. *J. Am. Stat. Assoc.* **2005**, *4*, 1941–1947.
- Drezner, Z.; Zerom, O.T.D. A Modified Kolmogorov-Smirnov Test for Normality. *Commun. Stat.—Simul. Comput.* **2010**, *39*, 693–704. [[CrossRef](#)]
- Lilliefors, H.W. On The Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown. *J. Am. Stat. Assoc.* **1967**, *62*, 399–402. [[CrossRef](#)]
- Bhatnagar, P.L.; Gross, E.P.; Krook, M. A model for collision processes in gases: I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.* **1953**, *94*, 511–525. [[CrossRef](#)]
- Qian, Y.H.; D’Humières, D.; Lallemand, P. Lattice BGK Models for Navier-Stokes Equation. *EPL (Europhys. Lett.)* **1991**, *17*, 479–484. [[CrossRef](#)]
- Maier, R.S.; Kroll, D.M.; Kutsovsky, Y.E.; Davis, H.T.; Bernard, R.S. Simulation of flow through bead packs using the lattice Boltzmann method. *Phys. Fluids* **1998**, *10*, 60–74. [[CrossRef](#)]
- Neumann, M.; Stenzel, O.; Willot, F.; Holzer, L.; Schmidt, V. Quantifying the influence of microstructure on effective conductivity and permeability: Virtual materials testing. *Int. J. Solid Struct.* **2020**, *184*, 211–220. [[CrossRef](#)]