Comparative Analysis of Porosity Coarse-Graining Techniques for Discrete Element Simulations of Dense Particulate Systems

Moris Kalderon[1], Edward Smith[2] and Catherine O'Sullivan[3]

[1]*National Technical University of Athens, School of Mechanical Engineering, Section of Mechanical Design & Automatic Control, Dynamics & Structures Laboratory, Heroon Polytechniou 9, 15780 Zografou Athens, Greece,*

mail: moriska@mail.ntua.gr, ORCID iD: 0000-0002-1593-9892

[2]*Brunel University London, Department of Mechanical and Aerospace Engineering, Uxbridge, Middlesex UB8 3PH, UK,*

mail: Edward.Smith@brunel.ac.uk, ORCID iD: 0000-0002-7434-5912

[3]*Imperial College London London, Department of Civil and Environmental Engineering, Skempton Building, London, SW7 2AZ, UK*

mail: cath.osullivan@imperial.ac.uk, ORCID iD: 0000-0002-0935-1910

Corresponding author: Moris Kalderon: mail: moriska@mail.ntua.gr

Acknowledgements

Dr. Edward Smith was funded by EPSRC grant EP/P010393/1 as well as by the embedded CSE programme of the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk). He would also like to acknowledge support under both the Distributed and Embedded Computational Science and Engineering program (dCSE and eCSE respectively).

Abstract

The Discrete Element Method (DEM) is a well-established approach to study granular materials in numerous fields of application; modelling each granular particle individually to predict the overall behavior. This behavior can be then extracted by averaging, or coarse graining, the sample using a suitable method. The choice of appropriate coarse-graining method entails a compromise between accuracy and computational cost, especially in the large-scale simulation typically required by industry. A number of coarse-graining methods have been proposed in the literature, these are reviewed and categorized in this work. Within this contribution two novel porosity coarse-graining strategies are proposed including a Voxel method where a secondary dense grid of “pixel-cells” is implemented adopting a binary logic for the coarse graining and a Hybrid method where both analytical formulas and pixels are utilized. The proposed methods are compared with four coarse-graining schemes that have been documented in the literature, including the Particle Centroid Method (PCM), an Analytical method, a method which solves the diffusion equation and an approach which employs averaging using kernels. The novel methods are validated for problems in both two and three dimensions through comparison with the “accurate” Analytical method. It is shown that, once validated, both the proposed schemes can approximate the exact solutions quite accurately, however there is a high computational cost associated with the Voxel method. The accuracy of both methods can be adjusted allowing the user to decide between accuracy and computational time. A detailed comparison is then presented for all six schemes considering “accuracy”, “smoothness” and “computational cost”. Optimal parameters are obtained for all six methods and recommendations for coarse graining DEM samples are discussed.

KEY WORDS: granular materials; discrete element method; homogenization; numerical simulations; coarse graining

Nomenclature

a, b, c	distances between the particle centroid and each bin face
A_{cap}	area of disk cap
A_i	disk segment area i
b	bandwidth (width of averaging kernel)
CFD	Computational Fluid Dynamics
Cu	Coefficient of Uniformity
D	diffusion coefficient
DEM	Discrete Element Method
d_p	particle diameter
d_{coarse}	coarse particle diameter
d_{fine}	fine particle diameter
F_{fine}	fines content (%)
G	shear modulus
h	disk segment height
n	porosity
$No_{Eulerian-cells}$	number of Eulerian cells
N_p	number of particles
PCM	Particle Centroid Method
Pi	number of pixels along each direction of the sample
PPM	Particle Meshing Method
r_p	particle radius
t	Time
V_{cap}	spherical cap volume
V_{cell}	Eulerian cell volume

V_{edge}	overlapping edge volume
V_p	particle volume
V_{segm}	particle segment volume
X	multiplier that controls the Eulerian cell size
$\boldsymbol{x}$	vector containing coordinates of Eulerian cell centroids
x_c, y_c	coordinates of particle centroid
$\boldsymbol{x}_{c,k}$	vector containing coordinates of particle centroid
a	magnitude of diffusion coefficient
Δs	fluid-cell cell size
Δx	fluid-cell cell size (x-dimension)
Δy	fluid-cell cell size (y-dimension)
Δz	fluid-cell cell size (z-dimension)
ε^p	particle volume fraction
$\zeta_{i,cell}$	coefficient of volume fraction
μ	interparticle friction coefficient
ν	Poisson's ratio
ρ	particle density

1 Introduction

There are many reasons why one might want to upscale, homogenize or coarse-grain, the particle information available from a Discrete Element Method (DEM) simulation to obtain a continuum field. These include a desire to interpret the simulation data within the continuum mechanics framework in which much of our understanding of material behaviour has evolved [1], the need to upscale data for use in multi-scale modelling approaches [2] or where a DEM code is coupled with a Computational Fluid Dynamics (CFD) solver [3]. This last application, coupling DEM-CFD simulations, is the motivation for this study. Specifically this contribution is relevant to the unresolved coupling approach where the fluid is modelled at a scale much larger than individual particles so the particle information is treated in an average manner [4].

In the case of unresolved DEM-CFD simulations there is no strict consensus on the appropriate size of the Eulerian cells used to discretize the fluid field, however Itasca Consulting Group Inc. [5] suggest $\Delta s/\mathrm{d_p} > 5$ where Δs is the length of the side of the CFD cell and d_p is the diameter of the considered DEM particles. In CFD-DEM the calculation of porosity (n) within the Eulerian cells is a key step in the calculation process, shown in **Fig. 1**. One challenge posed is the need to balance the accuracy of the porosity calculations with computational efficiency and at the same time conserve the total mass of the solid phase [6]. In addition, averaged DEM fields are directly used in the numerical solution of the CFD, where large oscillations ("noise") can cause problems with the numerical solution [7]. Kloss et al. [8] , Link et al. [2], Zhao and Shan [9] studied several calculation methods to improve smoothness and hence robustness of the pressure-velocity coupling, e.g. by smearing out each particle's region of influence or by additional spatial smoothing of the void fraction field.

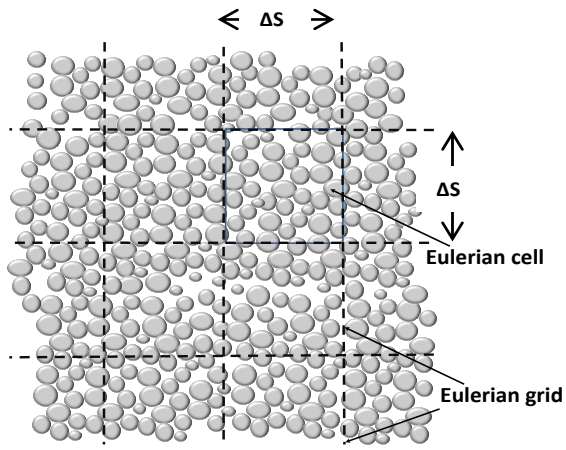

Fig. 1 Schematic showing a Eulerian grid for an unresolved coupled DEM-CFD simulation.

This manuscript critically assesses existing methods that have been developed to calculate porosity in CFD-DEM and proposes two novel algorithms. The existing and proposed methods are applied to dense assemblies of polydisperse disks and spheres and a comparison of the accuracy, computational cost and smoothness of the various methods is presented. The advantages and disadvantages of each method are listed, highlighting the trade-off between the need for more accurate methodologies and available computational power.

The paper is organized as follows. In Section 2 the existing coarse-grained algorithms documented in the literature are reviewed, compared, and evaluated. Section 0 presents the averaging algorithms implemented in the current study and demonstrates the rationale behind them in two and three dimensions. Section 4 describes the calibration of each algorithm, and presents the results of the prior tests. Section 5 identifies the advantages and limitations of each method by directly comparing them in terms of accuracy, smoothness and computational cost. Finally, conclusions are provided in Section 6.

2 Review of existing methods

The documented coarse graining approaches can be broadly classified as "binning" and "smoothing" methods. In the binning methods the particle volumes are assigned to the Eulerian cells (bins) and the porosity calculation is determined in each bin separately. In contrast, smoothing methods are based on the generation of a Eulerian field, from multiplication of particle positions by a weighting, generated using a Gaussian or Lucy function, so the particle density is distributed or "smoothed" over the support of the weighting function.

Binning methods: The simplest and most commonly used binning method is depicted in **Fig. 2a** and is known as the Particle Centroid Method (PCM). According to this method the entire volume of each particle is attributed to the Eulerian cell in which its centroid is located, even where the particle is shared between two or more cells [10–12]. Porosity fields in simulations using this method can display unrealistic results due to particle crossing the Eulerian cell boundaries, especially when the cells are not much larger than the particle diameter. To reduce this adverse effect the so-called approximate methods were developed. The "Divided Particle Volume Methods" (DPVP) approximate the shape of the particle by some simple geometric approach such as cube and then portions of the particle volume are assigned to neighbouring cells. Darmana et al. [13] following the work of Tomiyama et al. [14] introduced a DPVM where the particles (described as bubbles), can be either smaller or bigger than the Eulerian grid size. The particle volume fraction ε^p in a computational cell was calculated from the volume occupied by the particles (approximated as cuboids) present in the cell under consideration, multiplied with ζ_{cell}^{i}, a factor that depends on the particle shape, see **Fig. 2b**. Similarly, Khawaja et al. [15] assumed cubic particles (**Fig. 2c**) and applied corrections to approximate the solution for the case of spherical particles, aiming to reduce the computational effort of the exact solution. However, the problem of the approximation error remained, especially when the sphere is intersected by three planes, i.e. the sphere is divided to four segments while the cube is divided to eight. In their 2D study of a ring axi-symmetric Couette shear-cell, Lätzel et al. [16], proposed a slicing method for calculating the particle areas that belong to each ring wall with radius R_i (**Fig. 2d**), assuming that the boundaries are straight and the particles are cut in slices. Satellite point methods (SPM) [17–23] approximate the particle as collection of smaller pseudo-particles. Porosity is then determined by the amount of these pseudo-particles within each cell. Recently, such methods were employed to volume averaging of non-spherical particles [24–27] as they are significantly more accurate compared to the PCM. Peng et al. [18] introduced the Particle meshing method (PMM) which belongs to the SPM family. This method relies on a versatile meshing technique, by which the particles are meshed into several small particle grids. The solid volume in a fluid cell is calculated by counting the number of particle grids in the cell and adding up their volumes with the advantage that the particle volume can be allocated to different cells (**Fig. 2e**).

To obtain accurate results some researchers have focused on the development of so called "analytical" methods. These aim to get the exact fraction of the spherical particles located in each of the cells using analytical formula for the geometry intersection. Wu et al. [28] developed an accurate analytical

method developed for cells of different shapes (wedged, tetrahedral, or hexahedral) and for different particle–cell intersections (node, edge or face) (**Fig. 2f**). Bnà et al. [29] used a double Gauss–Legendre integration and a large number of nodes to calculate the intercepted sphere volumes. Similarly, Freireich et al. [30] assumed spherical particles, cuboidal measurement bins and a Cartesian coordinate system, and derived their solution by integrating the overlap volumes.

Smoothing methods: Unlike binning methods which extract average values in small volumes, smoothing techniques evaluate continuum fields as a function of time and space. The resulting fields satisfy the conservation equations of continuum mechanics, the particles are not assumed rigid or spherical and the spatial and temporal averaging scales are well-defined [31, 32]. In the smoothing methods a kernel, or wavelet, is applied to the center of each particle and porosity is distributed according to the kernel's shape. A similar procedure is used for filters in digital image processing applications. As a result, the solid volume fraction at each point is calculated as the summation of the distributed volume of each particle. Goldhirsch et al. [33, 34] studied the effect of the application of statistical kernels to calculate stresses in rapid granular flows. They noted that the classical statistical mechanical theories do not take into account the effect of resolution on the constitutive relationships and the fluctuations can be strong, revealing unrepresentative averaging. Weinhart et al. [35–37] chose a Gaussian function to establish the macroscopic velocity and stress fields while delivering smooth results; they applied cut-off limits to reduce the computational cost. They also mentioned that coarse-graining functions such as Heaviside or Lucy polynomials can be used instead, however the coarse-grained fields depend only weakly on the choice of the function. This approach has been implemented to consider chute flows [35, 36], silo flows [37], dosing of cohesive powders [38], ring shear cells [39], granular avalanches and more, using the open-source software MercuryDPM [40]. Zhu and Yu [41] developed an averaging method to derive the average balance equations for granular materials. In their work they described the characteristics of a proper weighting technique and special treatment was provided for the particles located near the external boundaries by introducing image particles (**Fig. 2g**). Sun and Xiao [42] applied the Gaussian distribution to calculate porosity and validated their results by comparison with other coarse graining methods, while Ries et al. [43] studied three different methods to tackle the challenge posed by applying statistical methods near the physical boundaries. Simonsen et al. [44] proposed a new weight for distributing the discrete phase exchange rates to the continuous phase cells. This is done by analytically integrating any distribution over the entirety of the particle trajectory, and only requires the probability and cumulative density functions to be specified. An equivalent to the kernel method approach, named the Diffusion Based method, was proposed by Sun and Xiao [42, 45]. This approach uses the field from the PCM method as an initial condition for a diffusive solver, which distributes the measured quantities over a wider area. Kloss et al. [8], Jing et al. [46] and Link et al. [47] identified limitations on the applicability of the binning methods (referred to as the divided void fraction method [9]) as erroneous results arise when the maximum particle size gets close to the fluid cell size, because the discrete particles would cause discontinuities to the porosity field in the fluid domain. To overcome this obstacle, they treated the particles as porous cubes [47] or porous spheres, [44]. According to this approach the actual particle volume is equally distributed to the porous particle allowing different porous particles to overlap in one Eulerian cell (see **Fig. 2h**).

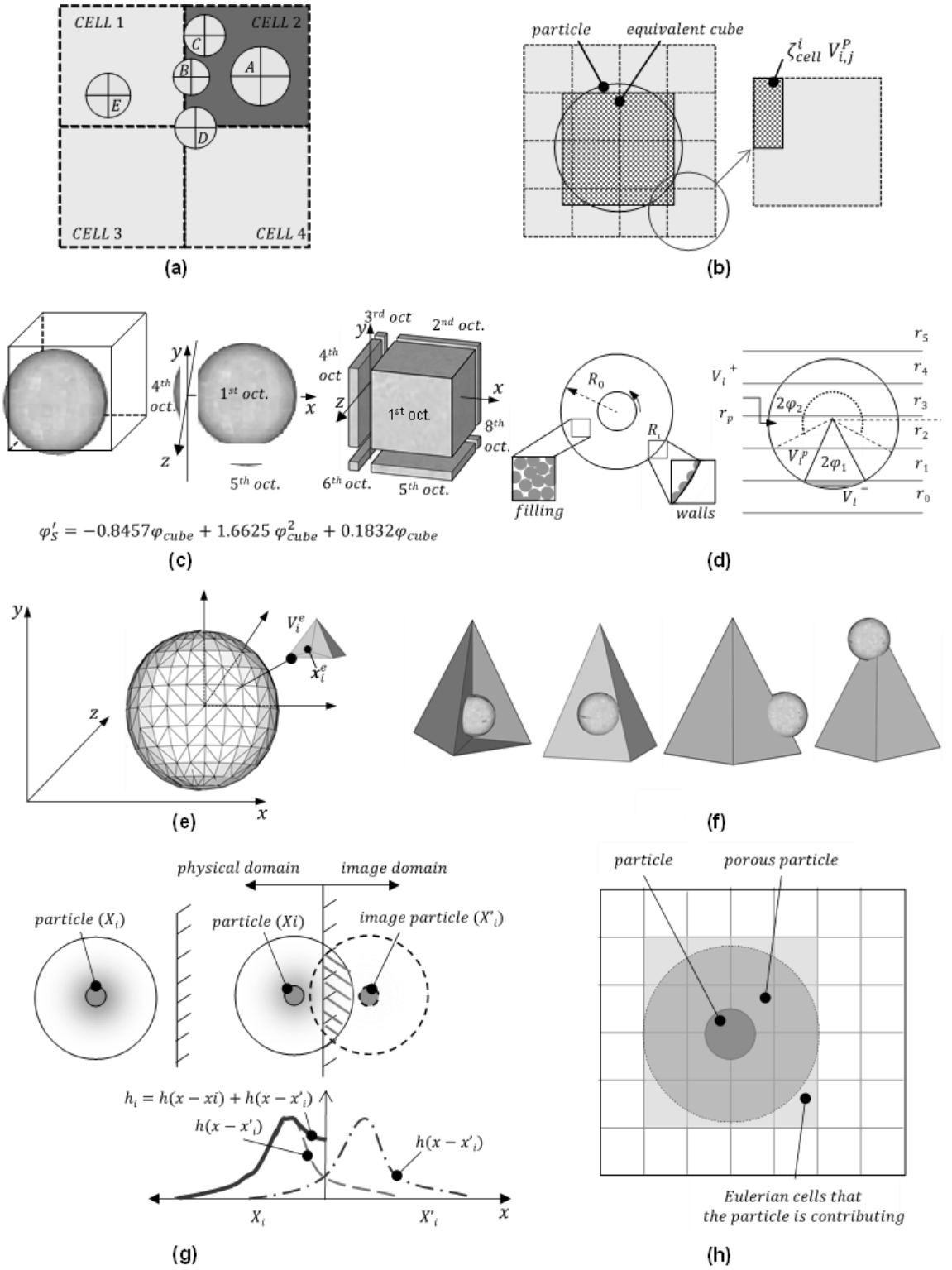

Fig. 2 Graphical illustration describing the binning and smoothing approaches applied in the literature. **(a)** PCM: The volume of particles A, B, C, D is allocated to $Cell\ 2$ and the volume of particle E to $Cell\ 1$. **(b)** Darmana et al. [13] DPVM: The spherical particles are approximated by cubes and corrections are applied to increase accuracy. **(c)** Khawaja et al. [15] DPVM corrected Cube method: Comparison between a sphere and a cubic particle at the same location, intersected by the same planes and the cube to sphere correlation. **(d)** Latzel et al model [16]: Plan view and schematic plot of a particle at radial position r_p which is cut into pieces by the boundaries R_l. **(e)** PMM [18]: Acquisition of centroid coordinate and volume of particle grids of each particle, where $\boldsymbol{x}_i^e$, V_i^e are the meshed particle coordinates and volume. **(f)** Wu et al. [28] analytical model: different cases of particle- positions/unstructured lattice. **(g)** PPM: Configuration of the porous particle representation,

assuming a porous sphere. **(h)** Statistical kernel method: Gaussian kernel averaging for a particle volume near a physical boundary. The kernel function $h(x - x_i)$ for the physical particle located at x_i is shown on the physical domain (left to the boundary); the kernel function $h(x - x_i)$ for the image particle located at x_i' is shown on the image domain (right to the boundary). The kernel function based on the superposition of the two functions above are shown on the physical domain in the top panel with shade and in the bottom panel with solid line

The most commonly employed coarse graining methods are summarized **Table 1** and are assessed based on the implementation complexity, use with unstructured mesh, parallel coding capabilities and difficulties in treating the exterior boundaries. Additionally, the smoothness of solution in small cells is stated.

Table 1 Summary of advantages and disadvantages of aggregation methods in the literature, with the names of the studied methods in this work in bold followed by an asterisk.

Coarse graining method	Use with unstructured mesh	Complexity to implement	Implementation in parallel code	Smooth field on small cells	Treatment of exterior boundaries	Variations of method	References
Particle centroid method (PCM)*	Easy	Easy	Easy	No	Easy	Classic	[10–12]
						2D to 3D scaling	[12, 48, 49]
Approximate method	Difficult	Difficult	Difficult	Depend on approximations	Easy	Equivalent cube method	[13–15, 28]
						Slicing method	[16]
						Particle meshing method (PPM)	[18]
						Satellite point methods	[17, 19–27]
Exact methods*	Difficult	Moderate	Difficult	Yes (if $\Delta x > \sim 2d_p$)	Easy	Divided particle volume method	[28]
						Numerical integration of Volume of Fluid function	[29]
						Analytical method for determining local solid fractions	[30]
						Analytical calculation of the overlap volume of spheres and mesh elements	[50]
Porous particle methods	Moderate	Moderate	Moderate	Yes	Moderate	Big particle (CFDEM software)	[8]
						Porous sphere method	[46]
						Porous cube method	[47]
Kernel method*	Moderate	Moderate	Moderate	Yes	Difficult	Statistical functions applied to stress fields	[33–40]
						Averaging method for granular material	[41]

Coarse graining method	Use with unstructured mesh	Complexity to implement	Implementation in parallel code	Smooth field on small cells	Treatment of exterior boundaries	Variations of method	References
						Kernel-based interpolation procedure between Eulerian and Lagrangian fields	[51]
						Statistical kernel method	[42]
						Coarse graining strategies at walls	[31, 43]
Diffusion-based method*	Easy	easy	Easy	Yes	Moderate	-	[42, 45]

3 Description of the porosity Coarse Graining Approaches

In the current study, six coarse graining methods are assessed. These comprise two binning methods; *the Particle centroid method* (PCM) and an *Analytical method* as well as two new approaches, a *Voxel-based method* and a *Hybrid method* where porosity is calculated combining analytical and a pixel/voxel grid. Additionally, two smoothing approaches, the *Diffusion-based* and *Kernel-based* methods are considered. A diagram, illustrating and classifying the considered methods is presented in **Fig. 3**.

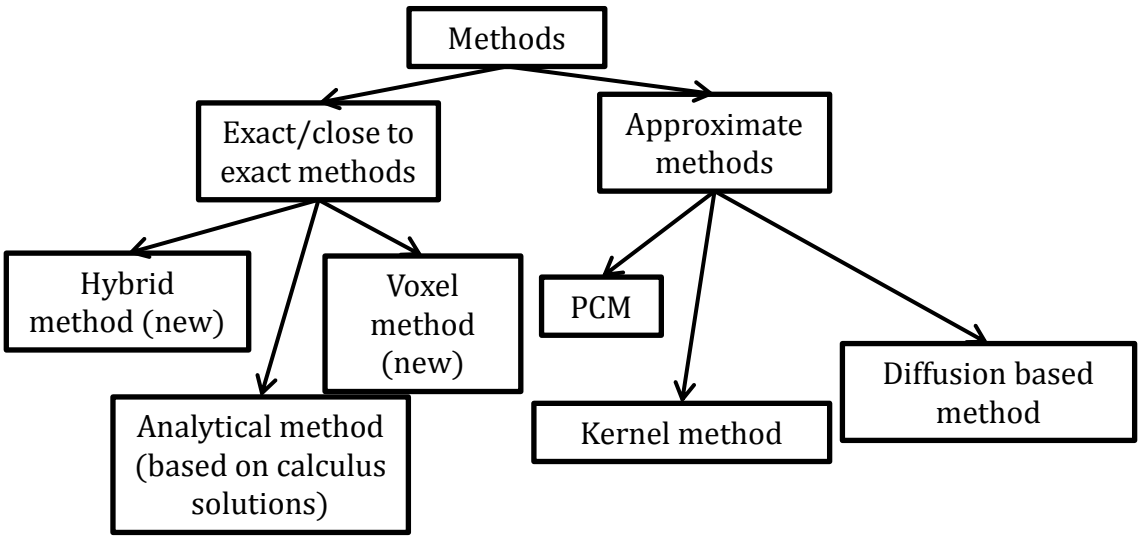

Fig. 3 Diagram of coarse graining methods analyzed here

3.1 General methodology

Any coarse-graining algorithm must consider particles moving freely in a specific domain, so that their centroid can be located at any random position within each Eulerian cell. Four different cases can be identified. In Category A particles are completely contained within the cell; in this case the whole area / volume of the particle is attributed to the specific cell. Category B considers the particles that intercept the boundary between two cells and as a result the particles are divided into two segments. For Category C the particles overlap the intersection of 4 (2D) or 8 (3D) Eulerian cells. The fourth category refers to the case where particles intersect the external boundaries of the sample (Category D). The applications considered in this study used periodic boundaries, consequently the portion of the particle located outside the sample boundaries is essentially attributed to the opposite Eulerian cell.

3.2 Particle centroid method (PCM)

This is the simplest but crudest approach to obtain a porosity coarse-grained field. The whole particle area is attributed to the cell containing the particle's centroid even when the particle extends beyond the cell boundaries. The porosity of a Eulerian cell using the PCM is as follows:

$$n = \frac{V_{cell} - \sum_{i=1}^{N_p} V_i^p}{V_{cell}} \tag{1}$$

where V_{cell} is the volume of the Eulerian cell, V_i^p is the volume of particle i and N_p is the number of particles contributing to the cell considered.

3.3 Analytical method

This method calculates the exact overlap volume (or area in 2D) of a particle and a cuboid bin by integrating the sphere's (or circle's in 2D) equation. Referring to **Fig. 4(left)** the integral limits are defined as the intersection of the planes defining the bin boundaries and the particle. This method assumes that the smallest dimension of the bin is larger than the circle's diameter and the implication of this assumption is that a 2D particle (circle) can be in contact with at most two mutually perpendicular bin walls or planes and a sphere with at most three. The mathematical formulations used are based on Strobl et al. [50], Richards [52] and Freireich et al. [30]. The overlapping area A_{cap} is calculated from **Eq. (2)** [53] and the segment A_4 by **Eq. (3)**:

$$A_{cap} = \mathrm{r}_p{}^2 \cos^{-1}\left(\frac{\mathrm{r}_p - h}{\mathrm{r}_p}\right) - (\mathrm{r}_p - \mathrm{h})\sqrt{2\mathrm{r}_p\mathrm{h} - \mathrm{h}^2} \tag{2}$$

$$A_4 = \int_{L1}^{L2} \left(\left(\sqrt{\mathrm{r}_p{}^2 - (x - \boldsymbol{x}_c)^2} \right) + \boldsymbol{y}_c \right) dx \tag{3}$$

where r_p is the particle radius, h is the disk segment height, x_c, y_c are the particle centroid coordinates and L_1, L_2 are the limits of the integral along the $x - axis$. The remaining segments are calculated by simple subtractions and the portions are allocated to the corresponding bins.

In three dimensions the complexity of the algorithm increases as the sphere can be divided in up to eight segments and the solution of the integrals is more computationally demanding. **Fig. 4(right)** illustrates the decomposition of a particle intercepted by a bin.

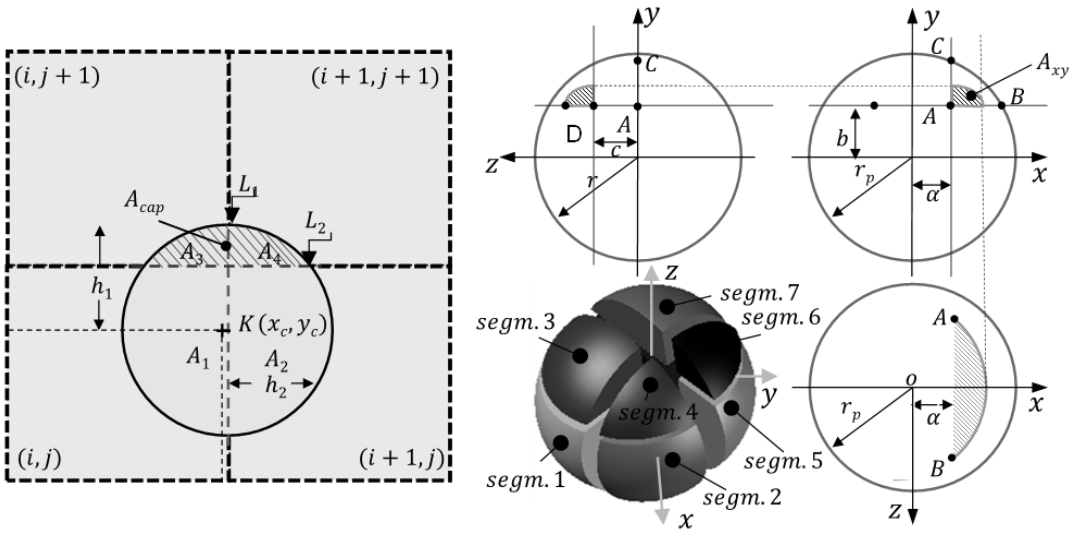

Fig. 4 (left) Indicative segment segregation and depiction of the integration limits in the 2D Analytical method. **(right)** Schematic showing the segment segregation (3D Analytical method) when the particle is intercepted in with a bin corner. The volumes in the Analytical method are calculated by double integrating to the depicted limits.

The first step to estimate each partial volume is to calculate the total sphere volume and the spherical cap volume, see **Eq. (4)**. Next the volumes due to edge overlap, see **Eq. (5)** and corner overlap, see **Eq. (6)** are determined. Finally, after all the portions are known the volumes are allocated to the relevant bin.

$$V_{cap} = \frac{1}{3}\pi h^2(3\mathrm{r}_p - h) \tag{4}$$

$$V_{edge} = \frac{2}{3}ab\sqrt{\mathrm{r}_p{}^2 - a^2 - b^2} + 2\mathrm{r}_p{}^3\tan^{-1}\left(\frac{b\sqrt{\mathrm{r}_p{}^2-a^2-b^2}}{\mathrm{r}_p a}\right) + \frac{2}{3}\mathrm{r}_p{}^3\tan^{-1}\left(\frac{a\sqrt{\mathrm{r}_p{}^2-a^2-b^2}}{\mathrm{r}_p b}\right) - \frac{1}{3}\left(3b\mathrm{r}_p{}^2 - b^3\right)\tan^{-1}\left(\frac{\sqrt{\mathrm{r}_p{}^2-a^2-b^2}}{a}\right) - \frac{1}{3}\left(3a\mathrm{r}_p{}^3 - a^3\right)\tan^{-1}\left(\frac{\sqrt{\mathrm{r}_p{}^2-a^2-b^2}}{b}\right) \tag{5}$$

$$V_{corner} = V_{segm.7} = \frac{1}{2}V_{edge} - \frac{1}{6}\Bigg[6abc - 2ac\sqrt{\mathrm{r}_p{}^2 - a^2 - c^2} - 2bc\sqrt{\mathrm{r}_p{}^2 - b^2 - c^2} - \left(3a\mathrm{r}_p{}^2 - a^3\right)\tan^{-1}\left(\frac{c}{\sqrt{\mathrm{r}_p{}^2-a^2-c^2}}\right) + 2\mathrm{r}_p{}^3\tan^{-1}\left(\frac{bc}{\mathrm{r}_p\sqrt{\mathrm{r}_p{}^2-b^2-c^2}}\right) + \left(c^3 - 3c\mathrm{r}_p{}^2\right)\tan^{-1}\left(\frac{b}{\sqrt{\mathrm{r}_p{}^2-b^2-c^2}}\right) - \left(3b\mathrm{r}_p{}^2 - b^3\right)\tan^{-1}\left(\frac{c}{\sqrt{R^2-b^2-c^2}}\right) + 2\mathrm{r}_p{}^3\tan^{-1}\left(\frac{ac}{\mathrm{r}_p\sqrt{\mathrm{r}_p{}^2-a^2-c^2}}\right) + \left(3c\mathrm{r}_p{}^2 - c^3\right)\tan^{-1}\left(\frac{\sqrt{\mathrm{r}_p{}^2-a^2-c^2}}{a}\right)\Bigg] \tag{6}$$

where r_p is the particle radius and a, b, c are the distances between the particle centroid and the corresponding bin face. The detailed descriptions of each scenario and the derivation of the equations are presented in [54].

3.4 Voxel method

The Voxel method was developed in the current study as an alternative to the Analytical method. The algorithm is based on the discretization of the whole sample domain introducing a dense secondary grid of small regular "pixel" cells. The density of the grid can be user-defined to control the resolution so that accuracy increases as the voxel size reduces. Two loops are used to scan the 2D sample in both horizontal and vertical direction in order to define the location of each voxel in respect to the particle's location. Similarly, in 3D samples the same procedure is applied with an additional loop to cover the extra dimension. Voxels that are covered by particles (particle voxels) are assigned with one and those which are covered by fluid (fluid voxels) are assigned with zero. The application of this method requires the secondary mesh to be few times smaller than the mean particle diameter. **Fig. 5** illustrates with "$\boldsymbol{x}$" and "**+**" symbols the voxel cells centroids of the Eulerian cell $(i+1, j+1)$, where crosses denote the fluid voxels and exes the Particle voxels. In other words**,** a binary logic is applied to determine whether a specific voxel centroid is covered by a particle. The distance of each voxel to the

nearest cell corner is then calculated and the volume of the particle voxel is added to the corresponding Eulerian cell.

3.5 Hybrid method

The Hybrid method can be considered as a special case of the Voxel method, where the Analytical method is employed for trivial calculations. Specifically, the area/volume of the particles is calculated analytically only when the circle/sphere overlaps one cell face or is entirely inside one Eulerian cell. Otherwise, when the sphere overlaps more than two cell faces, and the analytical approach becomes more complex, the calculation is achieved by pixel/voxel summation. In two dimensions the area of the circular particle is approximated by applying a second dense square grid extending one diameter from the particle's centre and the area is calculated by summing the area of the boxes that are shaped between the grid lines and are located within the circle's area. Similarly, in three dimensions the volume of the spherical particle is approximated by applying a cuboid grid and the volume is calculated by summing the volume of the corresponding boxes. An example of the method is described in **Fig. 6.**, where the volume of particle voxel 1 is allocated in Eulerian cell $(i, j+1)$, the volume of particle voxel 2 in Eulerian cell $(i+1, j+1)$, the volume of particle voxel 3 in Eulerian cell (i, j) and the volume of particle voxel 4 in Eulerian cell $(i+1, j)$.

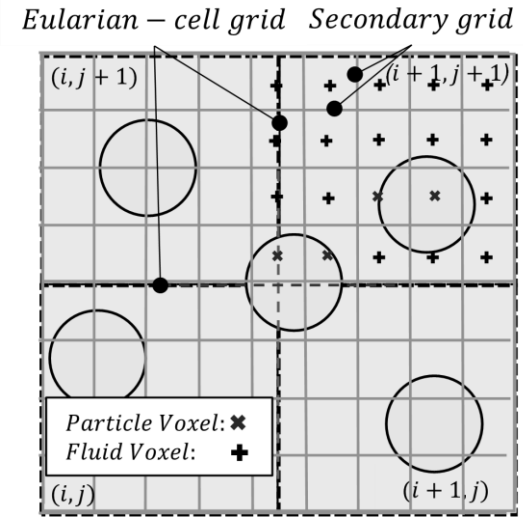

Fig. 5 Depiction of Voxel method double grid. The voxels that their centre is located in the particle area are counted and then allocated to the corresponding bins.

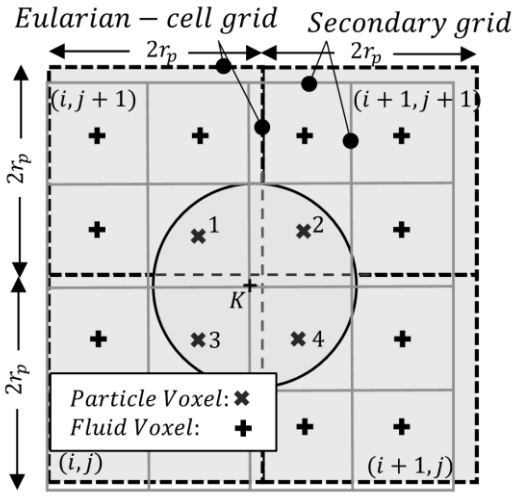

Fig. 6 Shematic of Hybrid method function when the particle is intercepted by more than two fluid cells.

3.6 Kernel method

This method differs from the previous methods as porosity is calculated indirectly and a heuristic approach is taken to explain the procedure. The current naïve MATLAB implementation will be less efficient than implementation of each method in a low-level language optimized for efficiency [40]. The volume fraction can be calculated in arbitrary positions on the sample's domain as a summation of the particles' distributed volumes included in the zone of influence of the kernel. Here, a Gaussian kernel is allocated to each particle and porosity of each Eulerian cell is estimated based on the contribution of each particle volume. Consequently, the smeared volume is a function of the distance between the particle and the Eulerian cell, as well as the shape of the kernel. **Fig. 7** shows schematically the process that porosity of Eulerian cell $(i, j+1)$ is calculated. The contribution of

particles 1, 2, 3 to the volume fraction ($V_{i,j+1}^{cell}$) depends on the distance (r) between the particle and the cell centroid, which should be less than the kernel's bandwidth (b_k) ($r_k \leq b_k$). The contribution of a particle to a cell is given by **Eq. (7)** for the 2D case and **Eq. (8)** for the 3D case, assuming a Gaussian kernel in both cases.

$$V_i^{cell,k} = \frac{V_i^p}{\pi b^2} e^{-\frac{(\boldsymbol{x}-\boldsymbol{x}_{c,k})^T(\boldsymbol{x}-\boldsymbol{x}_{c,k})}{b^2}} \tag{7}$$

$$V_i^{cell,k} = \frac{V_i^p}{(\pi b^2)^{3/2}} e^{-\frac{(\boldsymbol{x}-\boldsymbol{x}_{c,k})^T(\boldsymbol{x}-\boldsymbol{x}_{c,k})}{b^2}} \tag{8}$$

where, $V_i^{cell,k}$ is the volume "contributed" by particle k to Eulerian cell i, V_i^p is the volume of particle k, b is the kernel bandwidth, $\boldsymbol{x}$ a vector of the Eulerian cells coordinates and $\boldsymbol{x}_{c,k}$ a vector of particle coordinates.

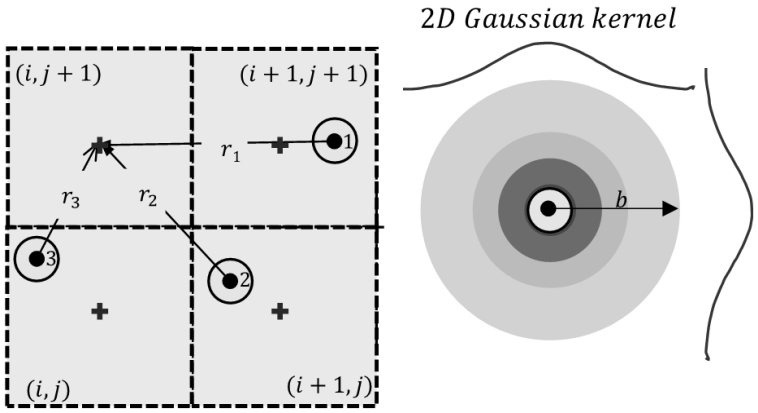

Fig. 7 Illustration of distributed volumes in Kernel method. Porosity of Eulerian cell $(i, j+1)$ is calculated by summation of the smeared volumes of particles 1, 2, 3.

3.7 Diffusion based method

This method was established to smooth the porosity coarse grained field that is produced by the PCM method and increase accuracy. Porosity essentially is redistributed by solving the diffusion equation [55] with initial conditions being the Eulerian cell porosity values derived from the PCM. The initial conditions correspond to time t, where porosity redistribution has not occurred yet and the final conditions correspond to time $t + \Delta t$, i.e after diffusion is completed, see **Fig. 8**. Time is introduced to describe a pseudo-timestep between the initial and final conditions and is used only to solve diffusion equation.

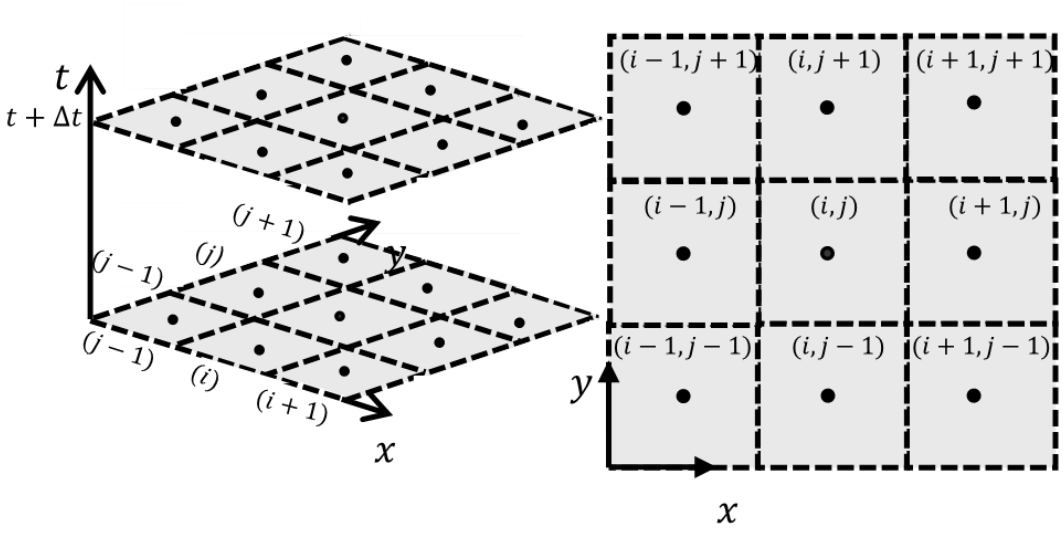

Fig. 8 Schematic explaination of diffusion prossess.

The numerical finite difference method (using a simple first order forward Euler in time and central-difference in space) is adopted to solve the diffusion equation, formulated as follows:

$$\frac{\partial n(r,t)}{\partial t} = D\,\nabla^2 n(\boldsymbol{x},t) \tag{9}$$

Where, n is the diffused quantity at location $\boldsymbol{x}$ and time t and D is the diffusion coefficient. In two dimensions the solution is as follows:

$$n_t \rightarrow \frac{n_{i,j}^{t+\Delta t} - n_{i,j}^{t}}{\Delta t} \tag{10}$$

$$n_x \rightarrow \frac{n_{i+1,j}^{t} - 2n_{i,j}^{t} + n_{i-1,j}^{t}}{\Delta x^2} \tag{11}$$

$$n_y \rightarrow \frac{n_{i,j+1}^{t} - 2n_{i,j}^{t} + n_{i,j-1}^{t}}{\Delta y^2} \tag{12}$$

By substituting **Eq. (10)**, **Eq. (11)** and **Eq. (12)** into **Eq. (9)** we have:

$$n_t = D\left(\mathrm{n_x} + \mathrm{n_y}\right) \tag{13}$$

For $\Delta x = \Delta y = \Delta s$ (square Eulerian cells) and $a = 2D\,\frac{\Delta t}{\Delta s^2}$

$$n_{i,j}^{t+\Delta t} = a\left(n_{i+1,j}^{t} + n_{i-1,j}^{t} + n_{i,j+1}^{t} + n_{i,j-1}^{t} - 4n_{i,j}^{t}\right) + n_{i,j}^{t} \tag{14}$$

In three dimensions the solution is as follows:

$$n_t \to \frac{n_{i,j,k}^{t+\Delta t} - n_{i,j,k}^{t}}{\Delta t} \tag{15}$$

$$n_x \to \frac{n_{i+1,j,k}^{t} - 2n_{i,j,k}^{t} + n_{i-1,j,k}^{t}}{\Delta x^2} \tag{16}$$

$$n_y \to \frac{n_{i,j+1,k}^{t} - 2n_{i,j,k}^{t} + n_{i,j-1,k}^{t}}{\Delta y^2} \tag{17}$$

$$n_z \to \frac{n_{i,j,k+1}^{t} - 2n_{i,j,k}^{t} + n_{i,j,k-1}^{t}}{\Delta z^2} \tag{18}$$

By substituting **Eq. (15)**, **Eq. (16)**, **Eq. (17)** and **Eq. (18)** to **Eq. (9)** we have,

$$n_t = D\left(n_x + n_y + n_z\right) \tag{19}$$

For $\Delta x = \Delta y = \Delta z = \Delta s$ (square Eulerian cells) and $a = D\frac{\Delta t}{\Delta s^2}$

$$n_{i,j,k}^{t+\Delta t} = a\left(n_{i+1,j,k}^{t} + n_{i-1,j,k}^{t} + n_{i,j+1,k}^{t} + n_{i,j-1,k}^{t} + n_{i,j,k+1}^{t} + n_{i,j,k-1}^{t} - 6n_{i,j,k}^{t}\right) + n_{i,j,k}^{t} \tag{20}$$

The diffusion coefficient a describes the rate of diffusion and is used to in place of the constants D, Δt and Δs, which have no physical meaning here. Selecting a suitable value for coefficient a is the main challenge associated with using this approach. The rationale for selection of a representative value is discussed in section **5**.

4 Parametric investigation of the proposed algorithms

In order to fairly compare the methods, the optimal performance of each method must be determined by tuning the parameters such as voxel size, kernel bandwidth or diffusion coefficient. This can be thought of as a calibration of the various methods to ensure a fair comparison and achieve optimal accuracy, smoothness, and computational cost. Computational cost is determined by implementation of all the methods in MATLAB® [56], an interpreted language which allows us to focus on just algorithmic cost with no compiler optimization. Naturally, some bias may be introduced by the author as the algorithms have not been optimized, however all approaches have been implemented using the same programming language and operating system.

The accuracy of the results of each implementation is judged relative to the Analytical method, which calculates the exact fraction of each sphere inside a cubic cell. The relative error between the two porosity fields is defined as follows:

$$Error\ (\%) = \frac{1}{No_{Eulerian-cells}} \sum_{i=1}^{No\ cells} \frac{\left|n_{cell,i}^{examined} - n_{cell,i}^{analytical}\right|}{n_{cell,i}^{analytical}} * 100 \tag{21}$$

where $n_{cell,i}^{examined}$ is the porosity value of the Eulerian cell i calculated by the examined method, $n_{cell,i}^{analytical}$ is the porosity value of the Eulerian cell i calculated by the Analytical method (reference method) and $No_{Eulerian-cells}$ is the total number of cells considered in the sample.

As stated in the introduction (Section 1) achieving a "smooth" averaged field from the DEM is an important factor to achieve a numerically stable CFD simulation. Smoothness is measured as the gradient of porosity between the adjacent cells as follows:

$$smoothness = \frac{1}{No\ cells} \sum_{i=1}^{No\ cells} \nabla n(\boldsymbol{x})_i \tag{22}$$

where $\boldsymbol{x} = x, y$ in two dimensions and $\boldsymbol{x} = x, y, z$ in three dimensions. The computational cost is assessed as the CPU time required to obtain the porosity field for an identical system run on the same computer with timing obtained as the average of several repeats.

The Kernel, Diffusion, Voxel and Hybrid methods have user-specified parameters which determine how well they perform based on competing accuracy, smoothness, and computational cost considerations. For the Kernel methods this is the adjustable bandwidth "b", while the Diffusion method has diffusivity coefficient included in the parameter α, see **Eq. (20)**. Similarly, the number of pixels can be varied in the Voxel and Hybrid methods along with the number of Eulerian cells selected for the sample. According to this calibration process the number of Eulerian cells is studied indirectly and is defined as $No_{Eulerian-cells} = (\frac{Sample\ size}{X*2r_p^{max}})^d$, where r_p^{max} is the largest particle radius, $Sample\ size$ is the width of the Sample (square or cubic samples are considered), d is the number of dimensions (2D versus 3D) and X is a multiplier that controls the size of the cells. To this end, the $No_{Eulerian-cells}$ is different for each sample but the multiplier X allows us to directly compare samples of different size. A multiplier X greater than 1 is proposed in order to avoid Eulerian cells fully occupied by particles, therefore the following five multipliers were adopted in this study: X =2, X =3, X =4, X =5 and X =6.

Note that the present analysis is performed under static conditions for simplicity. Instead of running the algorithms in each time step, as the particle positions are updated, time frames have been extracted from the DEM-CFD analysis and porosity is evaluated based on the particle's positions in the specific frame. In practice, coarse graining is applied to dynamic systems which increases the complexity of the performed calculations. However, porosity coarse-grained fields calculated statically can provide a basis for representative comparisons, as long as the same assumptions apply.

4.1 Assessment of the Implementations

The 2D implementations were calibrated based on five samples of disks generated by Summersgill [57] using the two-dimensional DEM code, PFC-2D [5]. The simulation parameters are particle density ρ=2600kg/m^3. The Gaussian distribution generates discs using the Gaussian curve with the minimum radius 1m (one standard deviation from the average) and absolute minimum radius 0.5m. **Fig. 9** gives visual illustrations of the examined 2D samples, where both Gaussian and uniform distributions of particle sizes were considered. The simulation parameters are summarized in **Table 3.** Where the simulation is named, the notation which identifies its variables is as explained in the following example (**Table 2**).

Table 2 Notation for 2D simulations

Simulation	Maximum number of discs	Distribution g=Gaussian, u=Uniform	Radius	Range of radii, p is decimal point	Friction coefficient of ball (aka disc)	Friction coefficient	Simulation identification (letter)
10000gr12p5fb0a	10000	g	r	12p5	Fb	0	a

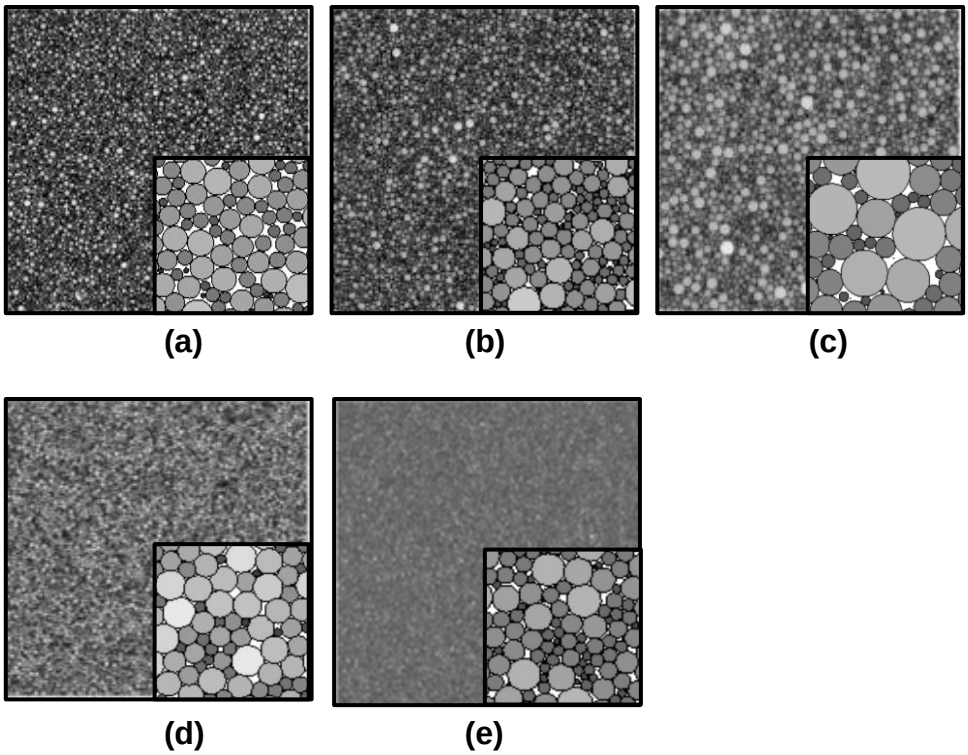

Fig. 9 Images of the 2D samples considered for the calibration. **(a)** 10000gr12p5fb0a, **(b)** 10000gr15fb1b, **(c)** 10000gr120fb0a, **(d)** 10000ur15fb0b, **(e)** 100000gr15fb0p5b. The particles are shaded by increasing radius.

Table 3 Summary of 2D DEM samples considered

Sample	Maximum number of discs	Distribution for Disc generation	Maximum radius (m)	Wall length (m)	Disc Friction Coefficient
10000gr12p5fb0a	10000	Gaussian	2.5	500	0
10000gr15fb1b	10000	Gaussian	5	1000	1
10000gr120fb0a	10000	Gaussian	20	4000	0
10000ur15fb0b	10000	Uniform	5	850	0
100000gr15fb0p5b	100000	Gaussian	5	3300	0.5

The 3D DEM data were generated using a modified version of LAMMPS [58] by Shire [59]. The eight cubic samples were created within periodic boundaries and the Hertz-Mindlin contact model [29,30] was adopted. The simulation parameters are Poisson ratio ν=0.3, shear modulus G=27GPa and particle density ρ=2670kg/m^3. **Table 4** summarizes samples with linear Particle Size Distributions (PSD) and different Coefficients of Uniformity ($Cu = d_{60}/d_{10}$, term d_{60} is grain diameter below which 60% of particles exist, and term d_{10} is grain diameter below which 10% of particles exists.); sample density was controlled by the coefficient of friction. As detailed in **Table 5**, bimodal samples with different particle size ratios $x = d_{coarse}/d_{fine}$ and fines content F_{fine}(%) are also studied, where d_{coarse} is the diameter of a coarse particle and d_{fine} is the diameter of a fine particle. A similar notation strategy to the 2D samples was also followed here. For example, sample L1.2a is a linearly graded sample with $Cu = 1.2$, where L stands for Linearly Graded (or B for bimodal) and 1.2 is Cu in case of a Linearly graded sample or size ratio in case of a bimodal. The inclusion of a or b is a simulation identification letter. Pictures of the 3D samples are given in **Fig. 10**. Hereinafter, the results of the parametric analyses that were conducted both on two and three dimensional implementations are presented.

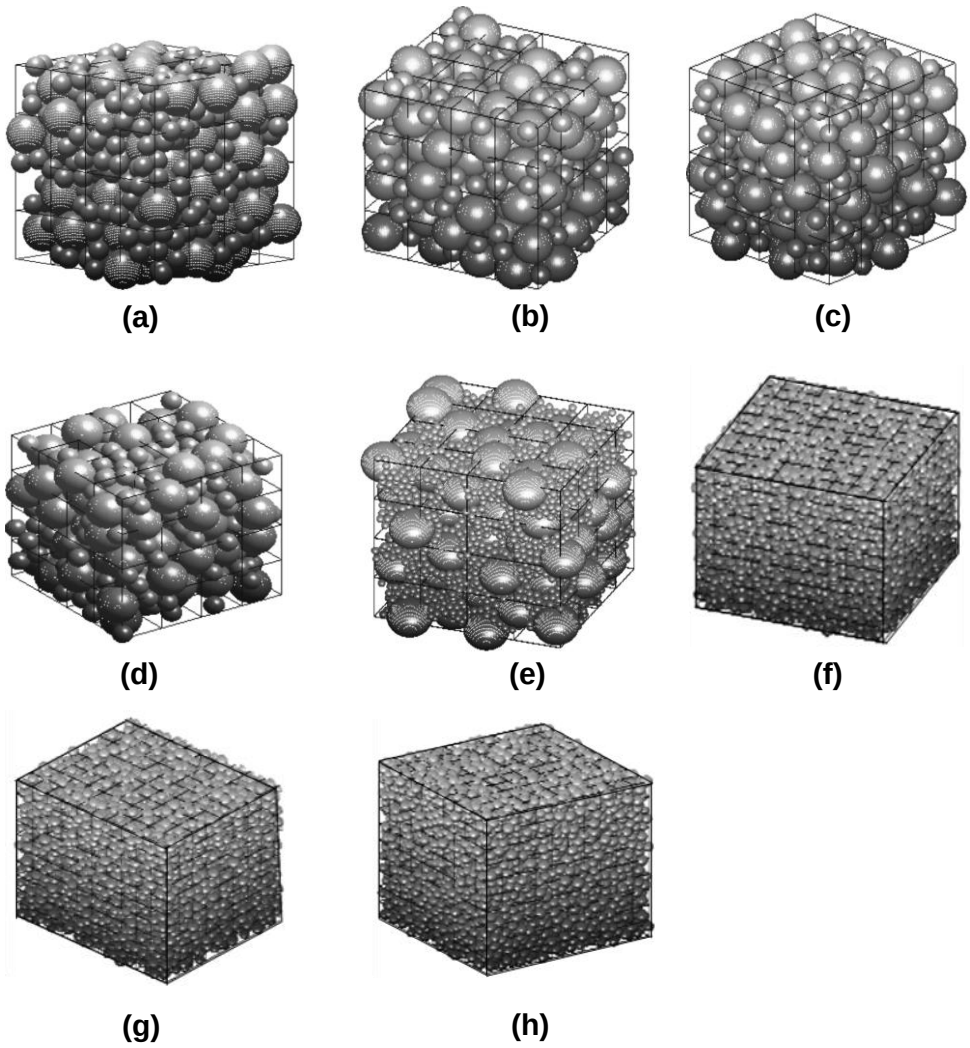

Fig. 10 Images of the 3D samples considered for the calibration: **(a)** B2a, **(b)** B2b, **(c)** B2c, **(d)** B2d. **(e)** B6a, **(f)** L1.2a, **(g)** L1.2b, **(h)** L1.2c.

Table 4 Summary of the 3D linearly graded samples.

Sample	Coefficient of uniformity (Cu)	Interparticle friction coefficient (μ)	Sample characterisation	Number of particles	Average particle radius (m)	Wall length (m)
L1.2a	1.2	0	Dense	8262	0.000117	0.00223
L1.2b		0.1	Medium			
L1.2c		0.3	Loose			
L3a	3	0	Dense	22600	0.000148	0.0049
L3b		0.1	Medium			
L3c		0.3	Loose			
L6a	6	0	Dense	59183	0.00015	0.0076
L6b		0.1	Medium			
L6c		0.3	Loose			

Table 5 Summary of the 3D bimodal samples.

Sample	Size ratio, d_{coarse}/d_{fine}	Fines content, F_{fine}(%)	Number of particles	Average particle radius (m)	Wall length (m)
B2a	2	20	307	0.000666	0.0046
B2b		25	367	0.000636	0.0047
B2c		30	443	0.000613	0.0048
B2d		35	531	0.000594	0.0049
B6a	6	20	5588	0.000182	0.0043
B10a	10	20	25376	0.000103	0.0043

4.2 Hybrid method

The performance of Hybrid method in two dimensions was assessed for the four different resolutions depicted in **Fig. 11**. From visual observation, it is apparent that when the coarse grid is utilized only a crude approximation of the particle is obtained, while a more accurate representation of the particle shape is available when more than 50x50 pixels are used. **Fig. 12a** shows the method's accuracy and computational time as an average of the studied samples for Eulerian cell size multiplier values of X=2 to 6. In other words, every sample has been tested for different number of Eulerian cells and accuracy is depicted as the average error of these tests, whilst the maximum and minimum deviation is presented with error bars. A fitting curve has been added to provide an estimation of the expected error, see **Eq. (23)**, where Pi is the number of Pixels along each direction of the sample ($Pi = \sqrt[2]{Pixels}$). For a resolution of Pi=5 the error is 6% and is reduced to 2.7% when Pi=10, while further increasing resolution to Pi =50 the error drops to 0.44%. Any additional increase of the resolution only provides a relatively minor reduction in error. For example, with Pi=100 the error is 0.34%. An indication of the expected computational time can be found by **Eq. (24).**

$$Error\ (\%) = 27.829 * Pi^{-2} \quad (23)$$

$$Comp.time\ (sec.) = 4 * 10^{-5} Pi^2 - 0.0005 * Pi + 2.0524 \quad (24)$$

The 3D Hybrid method was studied for three different resolutions (5x5x5, 20x20x20, 50x50x50) presented in **Fig. 12b**. When the method's resolution increases from Pi=5 to Pi=20 the average error reduces from 2% to 1.2 %, following with an increase in computational time from 0.2 to 0.9. When a higher resolution is selected, i.e. Pi=50 the error is estimated 0.4%, yet the computational time is one order of magnitude higher than before. The approximate relationship between error and number of pixels, where $Pi = \sqrt[3]{Pixels}$ is described by **Eq. (25)** and the relationship between the computational time and the number of pixels is given by **Eq. (26)**. Consequently, a resolution of Pi=50 is considered the most suitable selection for the 2D samples and Pi=20 for the 3D samples. These will be used as the calibrated values in section **5**.

$$Error\ (\%) = 7.0052 * Pi^{-0.702} \quad (25)$$

$$Comp.time\ (sec.) = 0.0037 * Pi^2 - 0.0477 * Pi + 0.3697 \quad (26)$$

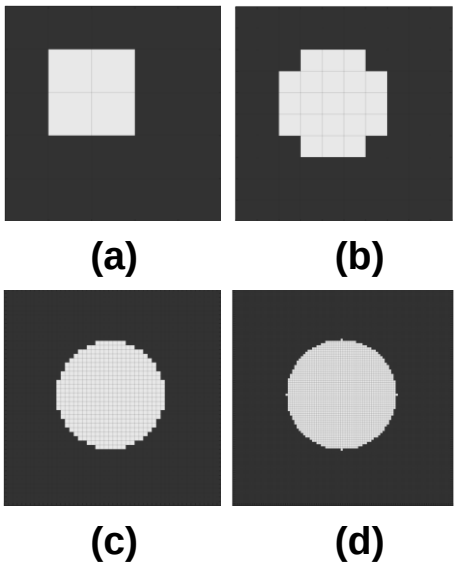

Fig. 11 Illustration of particle images produced by the Hybrid method for resolution **(a)** Pi =5 **(**5x5 pixels), **(b)** Pi =10 (10x10 pixels), **(c)** Pi =50 (50x50 pixels) **(d)** Pi =100 (100x100 pixels).

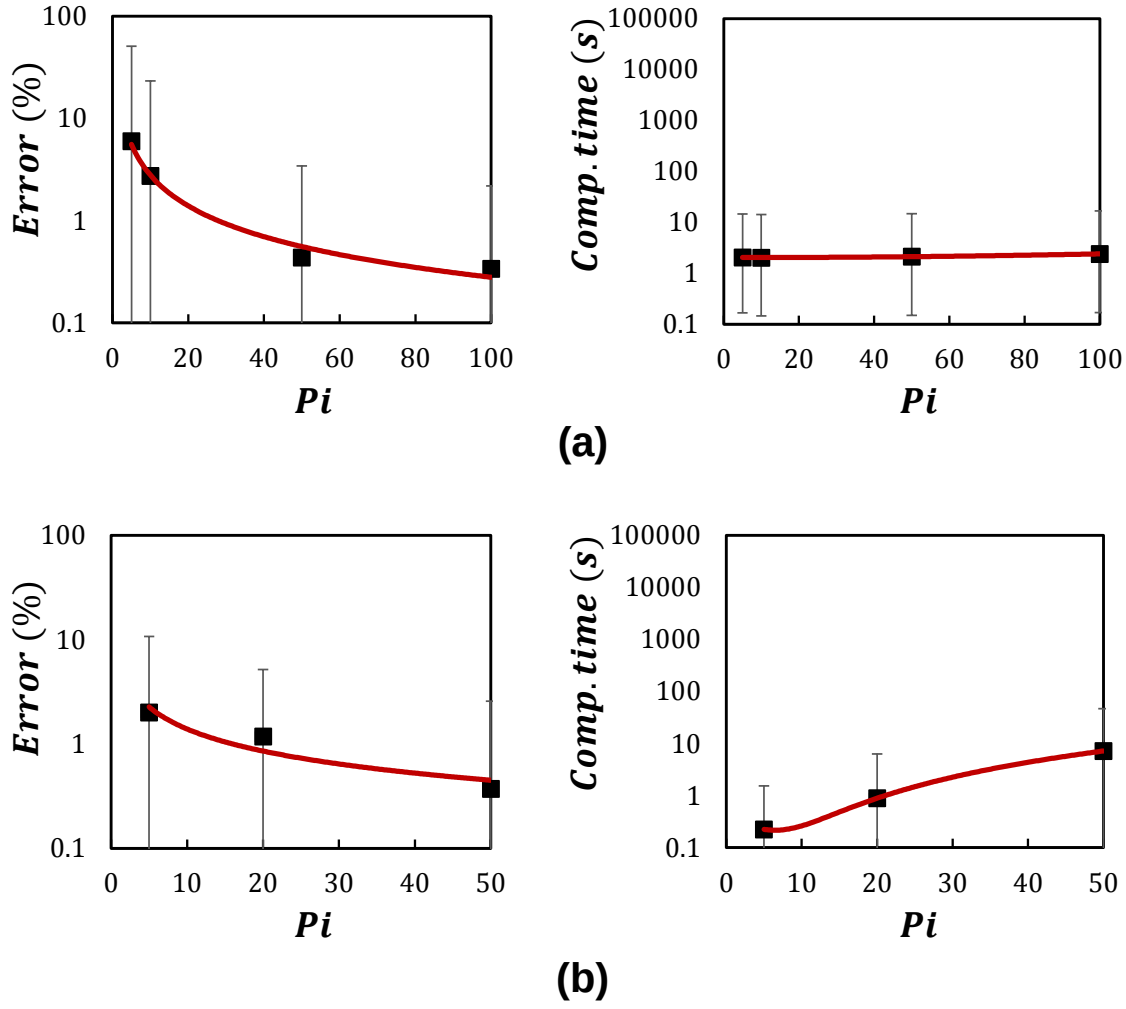

Fig. 12 Calibration of Hybrid method (**left**) Error (%) vs Number of Pi**,** (**right**) Computational time vs Number of Pi **(a)** 2D **(b)** 3D.

4.3 Voxel method

The same procedure as used when considering the Hybrid method was followed for the calibration of the Voxel method. In two dimensions four different resolutions (Pi =100, Pi =300, Pi =500 and Pi =1000) were studied per sample. **Fig. 13a** shows that when the resolution is low, for example Pi =100 the average error is 53%, although the computational time is just 6 sec. On the other hand, when resolution is increased to Pi =300 or Pi =500 the average error reduces to 15% and 10%, respectively. Unfortunately, the computational cost increases approximately two order of magnitude compared with the low-resolution case, reaching 140 sec. for Pi =300 and 590 sec. for Pi =500. A final high-resolution case of Pi =1000 was also studied to highlight the potential of the method in terms of accuracy. For this resolution, the average error drops to less than 5%. **Equation (27)** describes the expected average error based on the selected resolution and **Eq. (28)** provides an estimation of the expected computational time in seconds.

$$Error\ (\%) = 6130.8 * Pi - 1.038 \quad (27)$$

$$Comp.time\ (sec.) = 10^{-5} * Pi^3 - 0.0055 * Pi^2 + 1.5032 * Pi - 99.286 \quad (28)$$

For the 3D Voxel method four different resolutions were tested, namely Pi =20, Pi =50, Pi =100 and Pi =200. Of course, the lowest resolution of Pi=20 which corresponds to 8000 pixels cannot accurately represent the porosity coarse grained field (average error=24%), although it looks appealing showing tolerable computational cost (8 sec.), see **Fig. 13b**. A large error reduction is observed when the resolution increases to Pi =50 where the average error is 13% and the average computational time is 70 sec., while looking at the two highest resolutions a further improvement in accuracy (error= 10% and 9%) occurs, which is accompanied with an almost prohibitive increase in computational time; 450 sec. and 3000 sec., respectively. **Eq.(29)** and **Eq.(30)** express the relationship of resolution with accuracy and computational time.

Here, the target is to keep a balance between accuracy and computational time. Thus, a resolution of Pi =300 for the 2D samples and Pi =50 for the 3D samples is suggested.

$$Error\ (\%) = 77.814 * Pi^{-0.429} \quad (29)$$

$$Comp.time\ (sec.) = 0.0003 * Pi^3 + 0.0217 * Pi^2 - 0.5481 * Pi + 8.1295 \quad (30)$$

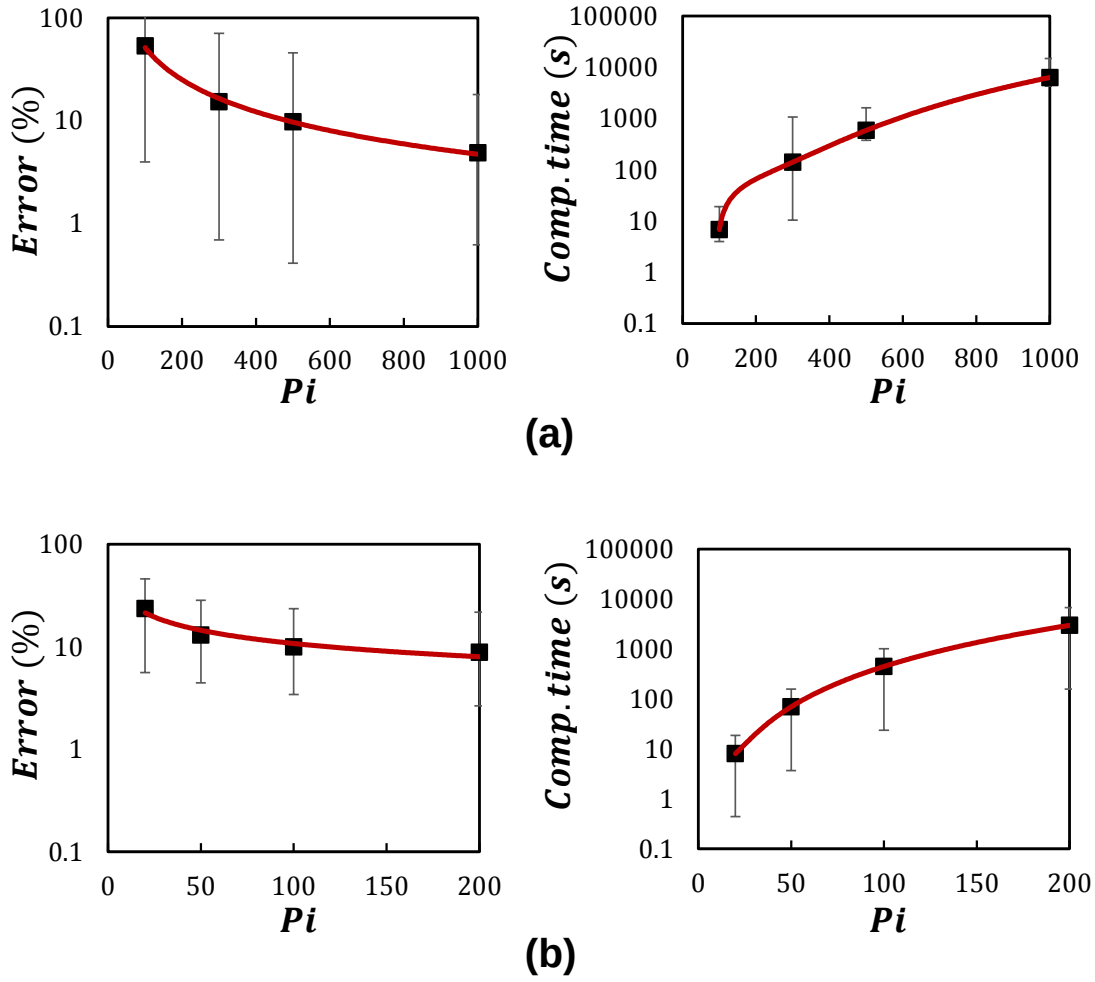

Fig. 13 Calibration of Voxel method (**left**) Error (%) vs Number of Pi (**right**) Computational cost vs Number of Pi **(a)** 2D **(b)** 3D.

4.4 Kernel method

When using the Kernel method the appropriate bandwidth (b) was selected by introducing a normalized bandwidth ($b/\Delta s$), where $\Delta s = X * 2r_p^{max}$ is the Eulerian cell size. Accuracy, which is translated to minimum error compared to the Analytical method, is again the criterion here for the selection of the optimum bandwidth. Specifically, a parametric analysis has been conducted for each sample to reveal the normalized bandwidth $(b/\Delta s)$ that demonstrates the minimum error. The optimization algorithm searched for minimum error for $b/\Delta s$ between 0.1 and 2, ignoring any local minima and the investigation was repeated for all five multipliers X, which as was stated previously, control the size of the cells. Hereinafter, the normalized bandwidth values that yielded the minimum error are plotted against the multiplier X, see **Fig. 14.**

The 2D investigation results show little variation in optimum $b/\Delta s$. Thus, a trendline is introduced in both plots, as optimum, that describes the relationship between $b/\Delta s$ and X multiplier based on the average values defined in each case, see **Eq. (31)** and **Eq. (32)**. Furthermore, the data suggest that an

average normalized bandwidth $b/\Delta s$ = 1 may be used across a range of 2D samples. Likewise, **Fig. 14(right)** indicates that a universal bandwidth $b/\Delta s$ = 0.4 can be used across the 3D samples without severely compromising the method's accuracy.

$$(\mathrm{b}/\Delta\mathrm{s})_{2D} = -0.038 * X + 1.112 \quad (31)$$

$$(\mathrm{b}/\Delta\mathrm{s})_{3D} = -0.0075 * X + 0.409 \quad (32)$$

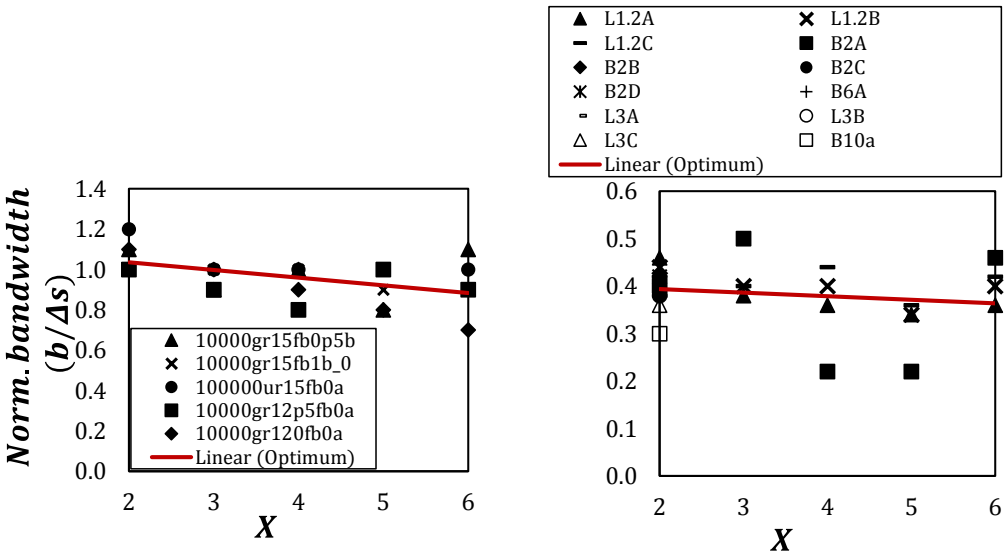

Fig. 14 Selection of optimal user-specified parameters in case of the Kernel method for **(a)** 2D implementations **(b)** 3D implementations.

4.5 Diffusion based method

Following the same logic as used in the calibration of the Kernel method, the porosity of every sample was initially estimated for several values of the smearing parameter, α, with range 0.01 to 0.3. Then the parameter α that revealed the minimum error for each studied case was plotted against the multiplier X. As seen in **Fig. 15**, the data accumulate around specific values of α for both the 2D **(left)** and the 3D **(right)** implementations plots irrespectively of the X multiplier. Specifically, in case of the 2D samples the average accuracy improves for α values between 0.15 and 0.22 and in case of the 3D samples between 0.10 and 0.15. A trendline is introduced in both plots, as optimum, that describes the relationship between α and X multiplier based on the average values defined in each case. This relationship is expressed in **Eq. (33)** for the 2D approach and in **Eq. (34)** for the 3D approach. Finally, it is observed that the selection of a suitable α is not severely affected by the X multiplier, and crudely a universal parameter α = 0.17 can be used across the 2D samples and α=0.12 across the 3D samples.

$$a_{2D} = -0.0042 * X + 0.211 \quad (33)$$

$$a_{3D} = -0.003 * X + 0.1327 \quad (34)$$

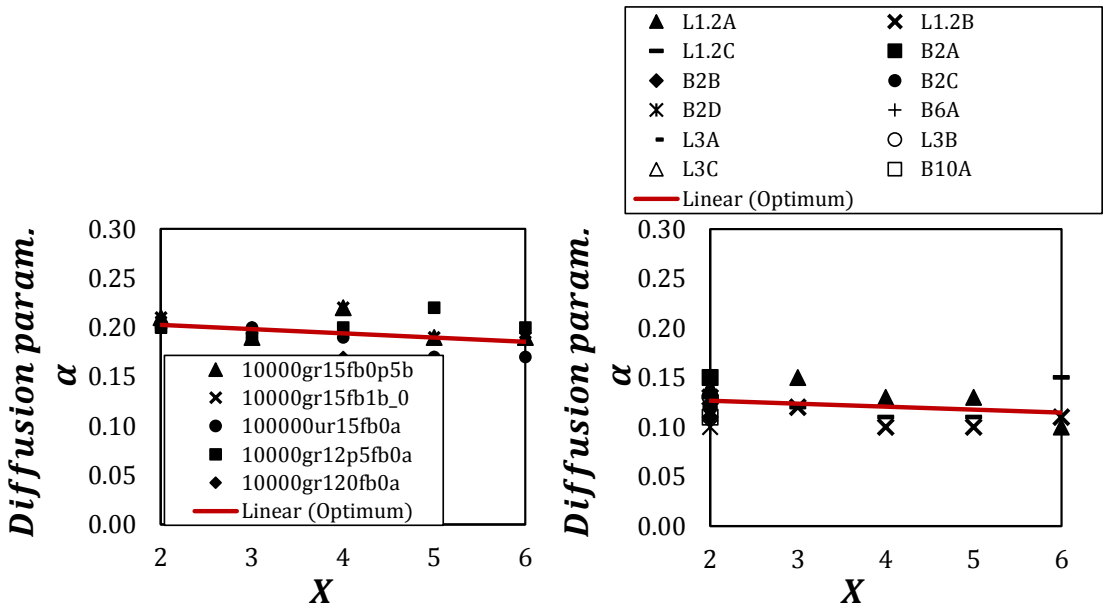

Fig. 15 Selection of optimal user-specified parameters in case of the Diffusion based method for **(left)** 2D implementations **(right)** 3D implementations.

5 Comparison – Discussion

In order to understand and identify the differences in the 2D and 3D porosity methods, each method was applied to one 2D sample and one 3D sample. Having established the parameters and characteristics of each algorithm, the comparison is conducted in terms of error (%), computational cost and smoothness. The smoothness metric quantifies how much noise is introduced by the DEM course-graining process.

Sample 10000gr12p5fb0a is used to compare the 2D algorithms and the comparison is based on the optimal parameters (see **Table 6**) derived in section 4. For this example, a Eulerian cell size multiplier X =3 is selected which corresponds to 144 Eulerian cells.

Table 6 Summary of the parameters (sample - 10000gr12p5fb0a) used in the comparison of the 2D methods

Kernel method $(b/\Delta s)$	**Diffusion based method** (α)	**Hybrid method (pixel grid)**	**Voxel method (pixel grid)**
1	0.17	50x50	300x300

Fig. 16 is a depiction of the sample's coarse-grained fields. The variations in cell shading show that the Analytical and the Hybrid methods yield almost identical porosity fields, which is not surprising, due to their common characteristics. The Voxel method follows the exact methods with a visually similar field, although there are still regions resembling to the PCM field. The PCM field reveals substantial variations among neighbouring cells, whereas after the application of diffusion the field is much smoother and closer to that of the Analytical method. The Kernel method has blurred out some local features of the porosity map, yet it is still comparable to that produced by the Diffusion and Analytical method.

A direct comparison among the methods is presented in the bar chart of **Fig. 17**. The PCM was found the least accurate (approximately 17% error) and the Hybrid (approximately 0.3% error) together with the Analytical the most accurate methods. The Diffusion based method manages to increase the PCM accuracy by almost 3 times, giving an error of 6.1%, while the Kernel method performed even better reporting 3.8% error. The Voxel method is ranked fourth in terms of accuracy, presenting 8.8% error.

The computational cost varies significantly amongst the implemented methods. The most cost-efficient methods are the PCM and Analytical method, while the Voxel method is over 3,000 times more computationally demanding. The Diffusion based method is 3 times slower and both the Hybrid and the Kernel 26 times slower than the PCM.

Regarding the smoothness metric PCM demonstrates the highest gradients (worst case) and the Kernel method the lowest (best case). The Analytical and Hybrid methods show comparable smoothness values which are 70% reduced compared with the PCM. The Diffusion method is the second-best method in terms of the smoothness metric reducing the PCM average gradient to more than half but still is not as efficient as the Kernel method which yielded only the 20% of the average PCM gradient.

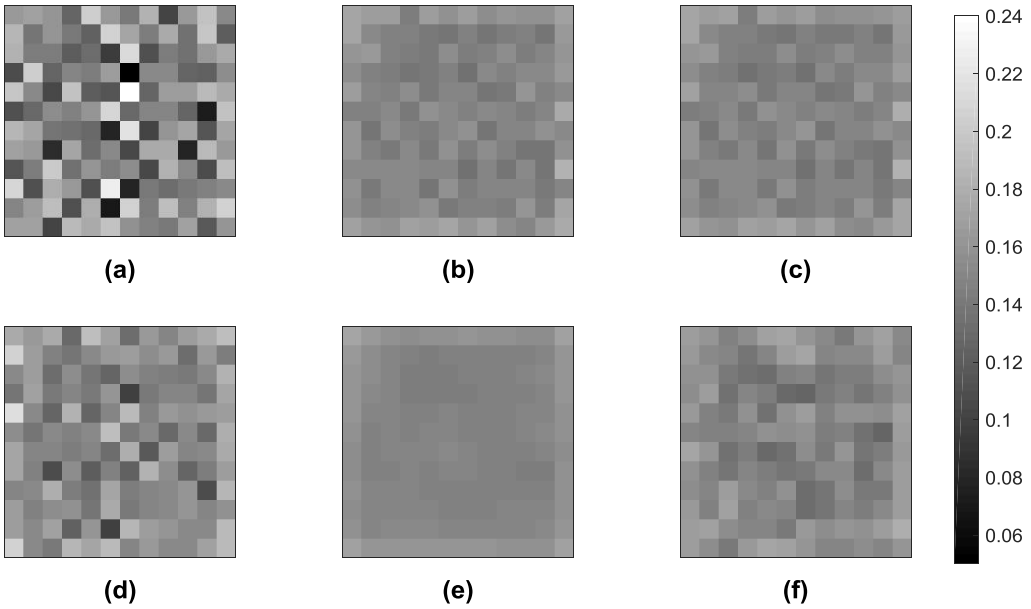

Fig. 16 Graphical presentation of sample 10000gr120fb0a by shading the Eulerian cells by porosity using **(a)** the PCM, **(b)** the Analytical, **(c)** the Hybrid, **(d)** the Voxel, **(e)** the Kernel and **(f)** the Diffusion based method.

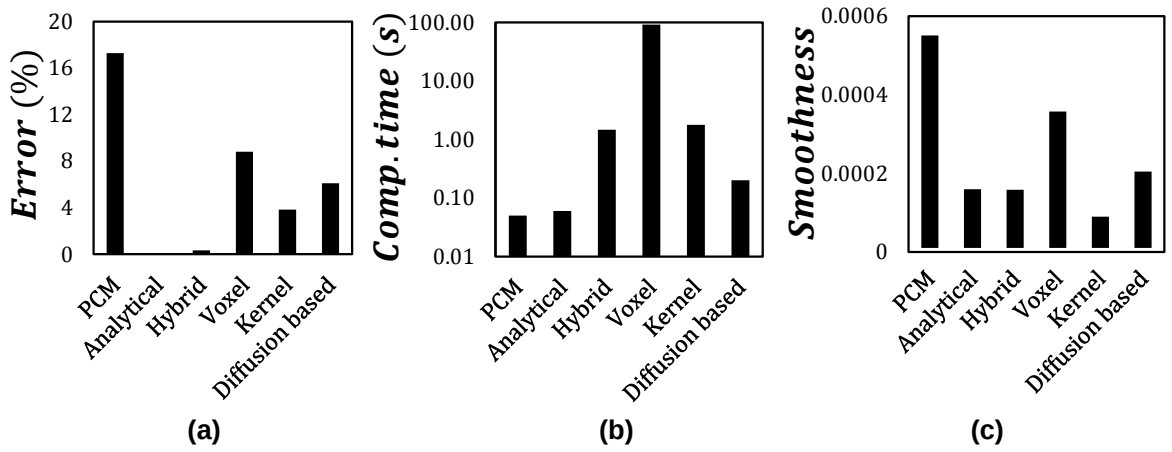

Fig. 17 Illustration of **(a)** the Error (%) **(b**) the computational time (s) and **(c)** smoothness of each method when applied to the 2-D sample 10000gr12p5fb0a.

The comparison among the 3D algorithms was carried out based on sample L1.2a and a Eulerian cell size multiplier X=3 which corresponds to 125 Eulerian cells. The calibration parameters are presented in **Table 7**. The bar charts in **Fig. 18** show that the Voxel method yielded the highest error, approximately 8%, which is attributed to the low resolution that was selected. In contrast, the Hybrid method displays the greatest accuracy, 1.4% error, while the performance of the Kernel and Diffusion methods is again remarkable displaying 1.7% and 2.5% error, respectively. In terms of computational time the Voxel method was again the most demanding, requiring almost 7,000 times more computational time compared with the PCM and Analytical method. The Hybrid method is the second most demanding, cost wise, however is still two orders of magnitude faster compared with the Voxel. All methods managed to reduce the high gradients observed at the PCM smoothness tests, with the Kernel method to be proven the most appealing as an average reduction of 76% is attained.

Table 7 Summary of the parameters (sample – L1.2a) used in the assessment of the 3D methods

Kernel method $(b/\Delta s)$	**Diffusion based method** (α)	**Hybrid method (pixel grid)**	**Voxel method (pixel grid)**
0.4	0.12	20x20x20	50x50x50

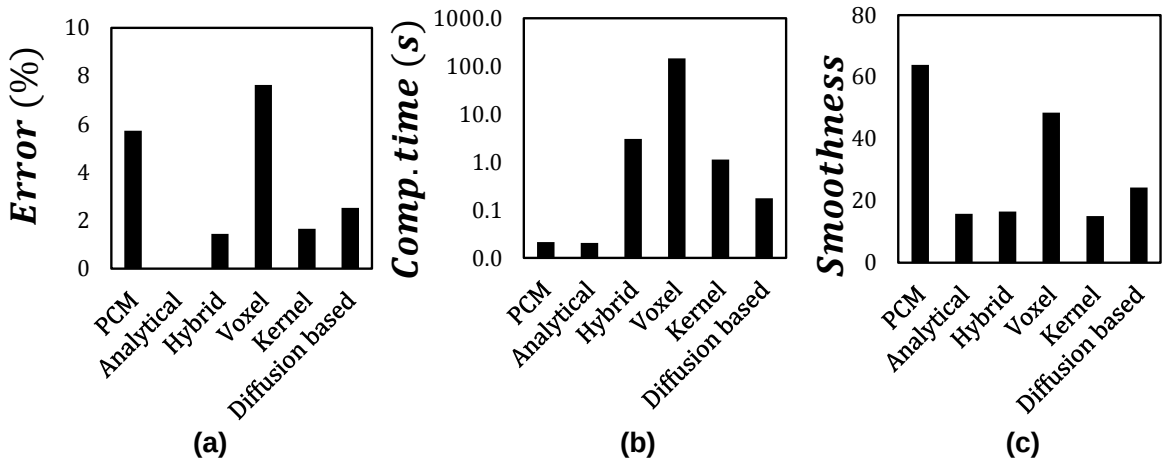

Fig. 18 Illustration of the **(a)** Error (%) **(b)** computational time (s) and **(c)** smoothness of each method when applied to the 3D sample L1.2a.

6 Conclusions

In this study, six different porosity coarse graining schemes were reviewed and applied to 2D and 3D systems of circular / spherical particles with periodic boundaries. This work was motivated by the inability of the most commonly used Particle centroid Method to accurately reproduce the required coarse graining of porosity and similar fields. The schemes studied were the Particle centroid method, an Analytical method, a Kernel based method, a Diffusion based method, together with two novel methods called the Voxel and Hybrid methods. The foundations and the mathematical background of each method were explored from a theoretical perspective outlining the rationale behind each approach. The optimal user-defined parameters needed to apply the methods were selected by considering “accuracy”, “smoothness” and “computational cost” based on available two and three-dimensional DEM-CFD samples. Note that the analysis was conducted under static conditions and some assumptions may have to be reconsidered for realistic, dynamic systems, however the comparison clearly illustrates the major characteristics of each method.

In conclusion, it was shown that the Kernel and Diffusion based methods required calibration in terms of the kernel width and diffusion magnitude, before any further evaluation is conducted, but once tuned they can provide cost-effective and reasonable representations of coarse-grained fields for the studied samples. The proposed Voxel and Hybrid methods showed high accuracy, giving the user control over accuracy against computational cost, an advantage that is not included in any other method found in Literature. The main drawback identified is the high computational resources required in case of the Voxel method, especially when 3D implementations are assessed. A wider range of samples could be examined in order to confirm a universal validity of the suggested parameters. All studied schemes demonstrated a more accurate depiction of the calculated fields compared to the PCM with varying degrees of smoothness and ease of implementation in CFD solvers.

Declarations

Compliance with ethical standards

The authors declare that they have no conflict of interest.

Availability of data and material

Supporting data is available on request: please contact moriska@mail.ntua.gr.

References

1. O'Sullivan C, D. Bray J, Li S (2003) A new approach for calculating strain for particulate media. Int J Numer Anal Methods Geomech 27:859–877. https://doi.org/10.1002/nag.304

2. Guo N, Zhao J (2016) 3D multiscale modeling of strain localization in granular media. Comput Geotech 80:360–372. https://doi.org/10.1016/j.compgeo.2016.01.020

3. Kafui KD, Thornton C, Adams M (2002) Discrete Particle-Continuum Fluid Modelling for Gas–Solid Fluidised Beds. Chem Eng Sci 57:2395–2410. https://doi.org/10.1016/S0009-2509(02)00140-9

4. Xu BH, Yu AB (1997) Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics. Chem Eng Sci 52:2785–2809. https://doi.org/10.1016/S0009-2509(97)00081-X

5. Itasca Consulting Group Inc. (2008) Particle Flow Code in 3 Dimensions CCFD Add-on. Minneapolis, MN, USA.

6. Smith ER, Heyes DM, Dini D, Zaki TA (2012) Control-volume representation of molecular dynamics. Phys Rev E 85:56705. https://doi.org/10.1103/PhysRevE.85.056705

7. Hirsch C (2007) Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics. Numer Comput Intern Extern Flows Fundam Comput Fluid Dyn. https://doi.org/10.1016/B978-0-7506-6594-0.X5037-1

8. Kloss C, Goniva C, Hager A, et al (2012) Models, algorithms and validation for opensource DEM and CFD-DEM. Prog Comput Fluid Dyn 12:140–152. https://doi.org/10.1504/PCFD.2012.047457

9. Zhao J, Shan T (2013) Coupled CFD–DEM simulation of fluid–particle interaction in geomechanics. Powder Technol 239:248–258. https://doi.org/10.1016/j.powtec.2013.02.003

10. Kawaguchi T, Sakamoto M, Tanaka T, Tsuji Y (2000) Quasi-three-dimensional numerical simulation of spouted beds in cylinder. Powder Technol 109:3–12. https://doi.org/10.1016/S0032-5910(99)00222-3

11. Tsuji Y, Kawaguchi T, Tanaka T (1993) Discrete particle simulation of two-dimensional fluidized bed. Powder Technol 77:79–87. https://doi.org/10.1016/0032-5910(93)85010-7

12. Hoomans BPB, Kuipers JAM, Briels WJ, van Swaaij WPM (1996) Discrete particle simulation of bubble and slug formation in a two-dimensional gas-fluidised bed: A hard-sphere approach. Chem Eng Sci 51:99–118. https://doi.org/10.1016/0009-2509(95)00271-5

13. Darmana D, Deen NG, Kuipers JAM (2005) Detailed modeling of hydrodynamics, mass transfer and chemical reactions in a bubble column using a discrete bubble model. Chem Eng Sci 60:3383–3404. https://doi.org/10.1016/j.ces.2005.01.025

14. Tomiyama A, Žun I, Higaki H, et al (1997) A three-dimensional particle tracking method for bubbly flow simulation. Nucl Eng Des 175:77–86. https://doi.org/10.1016/S0029-5493(97)00164-7

15. Khawaja HA, Scott SA, Virk MS, Moatamedi M (2012) Quantitative Analysis of Accuracy of Voidage Computations in CFD-DEM Simulations. J Comput Multiph Flows 4:183–192. https://doi.org/10.1260/1757-482X.4.2.183

16. Lätzel M, Luding S, Herrmann HJ (2000) Macroscopic material properties from quasi-static, microscopic simulations of a two-dimensional shear-cell. Granul Matter 2:123–135. https://doi.org//10.1007/s100350000048

17. Fries L, Antonyuk S, Heinrich S, Palzer S (2011) DEM–CFD modeling of a fluidized bed spray granulator. Chem Eng Sci 66:2340–2355. https://doi.org/10.1016/j.ces.2011.02.038

18. Peng Z, Moghtaderi B, Doroodchi E (2016) A modified direct method for void fraction calculation in CFD–DEM simulations. Adv Powder Technol 27:19–32. https://doi.org/10.1016/j.apt.2015.10.021

19. Gui N, Fan J, Luo K (2008) DEM–LES study of 3-D bubbling fluidized bed with immersed tubes. Chem Eng Sci 63:3654–3663. https://doi.org/10.1016/j.ces.2008.04.038

20. Radl S, Gonzales B, Goniva C, Pirker S (2014) State of the Art in Mapping Schemes for Dilute and Dense Euler-Lagrange Simulations

21. Hobbs A (2009) Simulation of an aggregate dryer using coupled CFD and DEM methods. Int J Comut Fluid Dyn 23:199–207. https://doi.org/https://doi.org/10.1080/10618560802680971

22. Boyce CM, Holland DJ, Scott SA, Dennis JS (2014) Novel fluid grid and voidage calculation techniques for a discrete element model of a 3D cylindrical fluidized bed. Comput Chem Eng 65:18–27. https://doi.org/10.1016/j.compchemeng.2014.02.019

23. Clarke DA, Sederman AJ, Gladden LF, Holland DJ (2018) Investigation of Void Fraction Schemes for Use with CFD-DEM Simulations of Fluidized Beds. Ind Eng Chem Res 57:3002–3013. https://doi.org/10.1021/acs.iecr.7b04638

24. Fitzgerald BW, Zarghami A, Mahajan V V, et al (2019) Multiscale simulation of elongated particles in fluidised beds. Chem Eng Sci X 2:100019. https://doi.org/https://doi.org/10.1016/j.cesx.2019.100019

25. Mahajan V V, Nijssen TMJ, Kuipers JAM, Padding JT (2018) Non-spherical particles in a pseudo-2D fluidised bed: Modelling study. Chem Eng Sci 192:1105–1123. https://doi.org/10.1016/j.ces.2018.08.041

26. Mahajan V V, Padding JT, Nijssen TMJ, et al (2018) Nonspherical particles in a pseudo-2D fluidized bed: Experimental study. AIChE J 64:1573–1590. https://doi.org/doi:10.1002/aic.16078

27. Mema I, Mahajan V V, Fitzgerald BW, Padding JT (2019) Effect of lift force and hydrodynamic torque on fluidisation of non-spherical particles. Chem Eng Sci 195:642–656. https://doi.org/10.1016/j.ces.2018.10.009

28. Wu CL, Zhan JM, Li YS, et al (2009) Accurate void fraction calculation for three-dimensional discrete particle model on unstructured mesh. Chem Eng Sci 64:1260–1266. https://doi.org/10.1016/j.ces.2008.11.014

29. Bnà S, Manservisi S, Scardovelli R, et al (2015) Numerical integration of implicit functions for the initialization of the VOF function. Comput Fluids 113:42–52. https://doi.org/10.1016/j.compfluid.2014.04.010

30. Freireich B, Kodam M, Wassgren C (2010) An exact method for determining local solid fractions in discrete element method simulations. AIChE J 56:3036–3048. https://doi.org/10.1002/aic.12223

31. Weinhart T, Thornton AR, Luding S, Bokhove O (2012) From discrete particles to continuum fields near a boundary. Granul Matter 14:289–294. https://doi.org/10.1007/s10035-012-0317-4

32. Babic M (1997) Average balance equations for granular materials. Int J Eng Sci 35:523–548. https://doi.org/10.1016/S0020-7225(96)00094-8

33. Glasser BJ, Goldhirsch I (2001) Scale dependence, correlations, and fluctuations of stresses in rapid granular flows. Phys Fluids 13:407–420. https://doi.org/10.1063/1.1338543

34. Goldhirsch I (2010) Stress, stress asymmetry and couple stress: from discrete particles to continuous fields. Granul Matter 12:239–252. https://doi.org/10.1007/s10035-010-0181-z

35. Weinhart T, Thornton AR, Luding S, Bokhove O (2012) Closure relations for shallow granular flows from particle simulations. Granul Matter 14:531–552. https://doi.org/https://doi.org/10.1007/s10035-012-

0355-y

36. Weinhart T, Hartkamp R, Thornton AR, Luding S (2013) Coarse-grained local and objective continuum description of three-dimensional granular flows down an inclined surface. Phys Fluids 25:70605. https://doi.org/https://doi.org/10.1063/1.4812809

37. Weinhart T, Labra C, Luding S, Ooi JY (2016) Influence of coarse-graining parameters on the analysis of DEM simulations of silo flow. Powder Technol 293:138–148. https://doi.org/10.1016/j.powtec.2015.11.052

38. Imole OI, Krijgsman D, Weinhart T, et al (2016) Reprint of "Experiments and discrete element simulation of the dosing of cohesive powders in a simplified geometry." Powder Technol 293:69–81. https://doi.org/10.1016/j.powtec.2015.07.052

39. Roy S, Scheper BJ, Polman H, et al (2019) Surface flow profiles for dry and wet granular materials by Particle Tracking Velocimetry; the effect of wall roughness. Eur Phys J E 42:14. https://doi.org/10.1140/epje/i2019-11778-x

40. Weinhart T, Orefice L, Post M, et al (2020) Fast, flexible particle simulations — An introduction to MercuryDPM. Comput Phys Commun 249:107129. https://doi.org/https://doi.org/10.1016/j.cpc.2019.107129

41. Zhu HP, Yu AB (2002) Averaging method of granular materials. Phys Rev E 66:021302. https://doi.org/10.1103/PhysRevE.66.021302

42. Sun R, Xiao H (2015) Diffusion-based coarse graining in hybrid continuum–discrete solvers: Applications in CFD–DEM. Int J Multiph Flow 72:233–247. https://doi.org/10.1016/j.ijmultiphaseflow.2015.02.014

43. Ries A, Brendel L, Wolf DE (2014) Coarse graining strategies at walls. Comput Part Mech 1:177–190. https://doi.org/10.1007/s40571-014-0023-6

44. Simonsen AS, Deen NG, Sørensen K, et al (2019) Trajectory integrated smoothening of exchange fields for discrete phase simulations. Comput Fluids 186:15–23. https://doi.org/10.1016/j.compfluid.2019.03.015

45. Sun R, Xiao H (2015) Diffusion-based coarse graining in hybrid continuum–discrete solvers: Theoretical formulation and a priori tests. Int J Multiph Flow 77:142–157. https://doi.org/10.1016/j.ijmultiphaseflow.2015.08.014

46. Jing L, Kwok CY, Leung YF, Sobral YD (2016) Extended CFD-DEM for free-surface flow with multi-size granules. Int J Numer Anal Methods Geomech 40:62–79. https://doi.org/10.1002/nag.2387

47. Link JM, Cuypers LA, Deen NG, Kuipers JAM (2005) Flow regimes in a spout–fluid bed: A combined experimental and simulation study. Chem Eng Sci 60:3425–3442. https://doi.org/10.1016/j.ces.2005.01.027

48. Ouyang J, Li J (1999) Particle-motion-resolved discrete model for simulating gas–solid fluidization. Chem Eng Sci 54:2077–2083. https://doi.org/10.1016/S0009-2509(98)00413-8

49. van Wachem BGM, van der Schaaf J, Schouten JC, et al (2001) Experimental validation of Lagrangian–Eulerian simulations of fluidized beds. Powder Technol 116:155–165. https://doi.org/10.1016/S0032-5910(00)00389-2

50. Strobl S, Formella A, Pöschel T (2016) Exact calculation of the overlap volume of spheres and mesh elements. J Comput Phys 311:158–172. https://doi.org/10.1016/j.jcp.2016.02.003

51. Xiao H, Sun J (2011) Algorithms in a Robust Hybrid CFD-DEM Solver for Particle-Laden Flows. Commun Comput Phys 9:297–323. https://doi.org/10.4208/cicp.260509.230210a

52. Richards WA (1995) The common content of a sphere and a regular tetrahedron having a common centre. Int J Math Educ Sci Technol 26:257–266. https://doi.org/10.1080/0020739950260211

53. Harris J, Stoecker H (1998) The Handbook of Mathematics and Computational Science

54. Kalderon M (2017) A Fundamental Assessment of Computational Porosity. MSc Thesis, Imp Coll London. https://doi.org/10.5281/zenodo.4436747

55. Fourier JBJ (1822) Théorie analytique de la chaleur. Chez Firmin Didot, père et fils

56. MATLAB (2010) version 7.10.0 (R2010a). The MathWorks Inc., Natick, Massachusetts

57. Summersgill F (2009) The Use of Particulate Discrete Element Modelling to Assess the Vulnerability of Soils to Suffusion. M.Eng Thesis, Imperial College London

58. Plimpton S (1995) Fast Parallel Algorithms for Short-Range Molecular Dynamics. J Comput Phys 117:1–19. https://doi.org/10.1006/JCPH.1995.1039

59. Shire T (2014) Micro-scale modelling of granular filters. PhD Thesis, Imperial College London

60. Hertz H (1882) Ueber die Berührung fester elastischer Körper. J für die reine und Angew Math 92:156-71

61. Mindlin R (1949) Compliance of elastic bodies in contact. Trans ASME, J Appl Mech 16:259-68