

We thank the referee 1 for their positive comments and thoughtful suggestions below. Each point has been addressed individually, with changes to the manuscript text shown in **orange text**. Unchanged manuscript associated with these changes is shown in **blue text**.

Comments from Referee 1

This contribution presents Pysammos, a software that can be used to post-process the results of any Discrete Element Modelizations (DEM). Indeed, the discrete particle data obtained with the DEM can be transformed into continuous fields, more accessible for visualization purposes and deeper analysis. The architecture, the abilities, the validity, and the performances of Pysammos are depicted in this paper. This method appears user-friendly, computationally efficient, and flexible to the DEM software used.

The article is well written and well structured.

Even if I am positive about this article, I have a few queries to consider during this minor revision.

- Comment 1: L. 55: I would change “force displacement law” to “contact law”.

Response: Thank you for having raised such an important point. We believe that using “force displacement law” remains more faithful to the theory explained in the seminal Discrete Element Method (DEM) papers. For example, in Cundall & Strack (1979) the calculation cycle of the DEM is summarised as follows: “The calculations performed in the distinct element method alternate between the application of Newton’s second law to the discs and a force-displacement law at the contacts. Newton’s second law gives the motion of a particle resulting from the forces acting on it. The force displacement law is used to find contact forces from displacements. [...] The relative displacements occurring at contacts A and C at time $t_1=t_0 + \Delta t$ (see Fig. 1(b)) are used in a force-displacement law for the calculation of contact forces.” Force displacement laws relate the generalised internal force to the relative motion at the contacts. Although they can be used in a context where contact occurs, they are not inherently about the nature of a contact. They focus on the behaviour of individual elastic elements under applied force, regardless of whether they are in contact or not with another object, and regardless of the nature of the contact law, which describes the conditions under which two elements interact physically (e.g., Hertzian contact law).

Nonetheless, we have clarified the sentence to be more accurate and avoid confusion:

L54: The DEM explicitly **applies** Newton’s second law of motion **to the particle centres**, **and the force displacement law at the contacts**, to compute trajectories of individual granular particles from the forces exerted on particles, and vice versa, over time periods in which accelerations are assumed to be constant (Cundall & Strack, 1979).

- Comment 2: L. 116: *There is no verb in “Similarly, to enhance DEM analysis by extracting continuum fields without the need to handle inner-level source code”. Could you reformulate this sentence to ease the reading?*

Response: Thank you for pointing this out. We agree with your comment. We think this sentence should be better tied to the previous sentence. Therefore, we have reworded the point that explains the motivation behind the user-friendly interface enabled by the modular structure:

L115: Its modular design enhances DEM analysis across users with varying expertise, as it does not require access to inner-level source code. This is intended to encourage a broader use of DEM in geoscientific research, where many observed processes involving discrete elements are currently modelled as a continuum but could benefit from a discrete insight.

We have also moved the sentence on algorithmic complexity before the point about the user-friendly design justification:

L113: This code package provides flexibility in the output variable selection, mesh parametrisation, and data analysis of the obtained results. The efficient algorithmic complexity exhibited by Pysammos avoids the requirement of extensive computational resources, making it a program that can be run at the same time as other processes.

- Comment 3: L. 149: *I would not refer to Eq. 9 solely, as you use multiple functions for the coarse-graining function Ψ . Is a reference to Section 3.4.3 more accurate*

Response: We agree with your suggestion: referring to the section rather to just the Lucy function equations remarks that different weighting functions can be used. Therefore, we have made the following modification to the sentence:

L147: across a volume of radius c and effective smoothing width w (see Sect. 3.5.3).

Note that further modifications to the sentence were made according to the response to comment 7.

- Comment 4: Eq. 5: *Could you check this Equation? In the current version, an individual contact is computed twice. I guess that the second sum should start at $j=i+1$, in agreement with the formulation depicted in Weinhart et al. (2012).*

Response: Thank you for pointing this out. Indeed, the formulation $j=1, j \neq i$ to $N+N_b$ was convoluted and incomplete. We agree with your suggestion and have therefore made the appropriate modification to Eq. 5 and the corresponding symbol description. In the correction below you will also find that we have addressed Comment 13 about the wall particles.

$$\boldsymbol{\sigma}^c(\mathbf{r}, t) = \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^{C_{ij}} \mathbf{f}_{ij}^k \mathbf{a}_{ij}^k \int_0^1 \Psi(\mathbf{r} - \mathbf{r}_i + s \mathbf{a}_{ij}^k) ds,$$

L163: where N is the number of particles in the system, and C_{ij} is the maximum number of contact points between two particles.

We have also made the same correction to the fabric tensor (Eq. (C4)):

$$\mathbf{F}(\mathbf{r}, t) = \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^{C_{ij}} \nu_i \mathbf{n}_{ij}^k \mathbf{n}_{ij}^k \int_0^1 \Psi(\mathbf{r} - \mathbf{r}_i + s \mathbf{a}_{ij}^k) ds,$$

- Comment 5: L. 170: *It appears in L. 385 that you made a different choice than other D2C software (MercuryDPM, for instance) for the particle search (particle vs contact point). A justification of this assumption can be relevant here.*

Response: Thank you for pointing this out. In the original manuscript, we suggested that the slight disagreement in contact stress tensor between MercuryDPM and Pysammos could be due a difference choice in particle search. We decided to perform the particle search with respect to the particle centroid because the weights in the interparticle interaction terms are calculated with respect the centroids of the particles, according to the seminal paper of Babic 1997. However, the choice of the element search in the code workflow is not specified by the mathematic formulae and therefore it is a choice of the developer. We agree it is worth providing a justification of this choice, but it is more appropriate to do so in section “3. Methodology: code structure and algorithm steps”, rather than in section “2. Theory”. In section “3.5.2 Particle Search” we have included the following text to clarify that the search is done with respect to particle centroids for all fields, for consistency and to improve computational efficiency:

L328: Continuum macroscopic fields consist of the spatial weighted average of microscopic particle quantities within a finite volume around any point in space (Fig. 5). The particles within this volume are identified based on the distances between particle centres and grid points, using a k-d tree-based nearest-neighbour search, which efficiently identifies, for all grid points in a single query, the particle centres

within a cut-off distance of each point (Appendix B2). To improve computational efficiency, the search is performed only once every time step, and the resulting neighbour information is employed in all subsequent calculations of fields involving either individual particles or interparticle interactions.

- Comment 6: L. 264: *I would start with the Section “Data reading and handling” as the format of the input data is described. Please consider reorganizing your Sections.*

Response: Thank you for raising this point. The original structure of the Methodology section intended to describe two main aspects of the code: the code structure, through section 3.1 Software design, and the main algorithm steps, through sections 3.2 Data inspection, 3.3 Mesh generation and 3.4 Coarse-graining steps. The algorithm steps were put in that order because they follow the workflow of the coarse-graining encompassed in the CoarseGraining class (Fig. 3). We described the nature of the DEM data in the section “Data reading and handling” with the purpose of having a compact data section. However, we agree with you that describing the input data earlier on in the Methodology section would be helpful for the reader. Hence, we have added a section after section 3.1 Software design, and before the sections corresponding to the principal algorithm steps, called “Input DEM data”. In this new section we detail the nature of the DEM data required in Pysammos and how to supply it to the code.

3. Methodology: code structure and algorithm steps

L218: This section firstly describes how the architecture has been designed to facilitate maintenance, and to cater for the performance considerations inherent to DEM data processing. Subsequently, it provides a description of the input DEM data, and an overview of the algorithm steps of the CG workflow embedded and handled from the user-facing class.

[...]

3.2 Input DEM data

L263: The paths to the DEM data files and the corresponding variable names are supplied by the user either through the configuration file (Appendix A), or directly to the user-facing CoarseGraining class.

The input DEM data required for Pysammos consist of particle data and contact data, which are stored in separate files. The particle data include particle position, particle ID, velocity, diameter (or radius), density, volume, mass, and optionally coordination number. The contact data include the particle IDs of the two particles involved in each recorded contact, the force acting on the first particle, and the contact point

position. If contact point positions are unavailable (e.g., when using earlier MFiX releases), Pysammos can recalculate them for spherical particles.

MFiX records single-sided contacts, providing the force vector acting on the first particle, and the contact point position. Since the contact point position is explicitly available, reconstruction of the full bidirectional contact list is valid regardless of particle size distribution — the branch vectors (a_{ij} in Eq.(5)) are computed directly from particle centre positions and the contact point, rather than being inferred from particle geometry. The reconstruction procedure is described in Sect. 3.5.1 and the Appendices referenced therein.

In addition, we modified the heading of the sections that correspond to algorithm steps to include the word “stage”, to emphasise the nature of their content.

3.3 Data inspection stage

[...]

3.4 Mesh generation stage

[...]

3.5 Coarse-graining stage

[...]

The “Data reading and handling” section remains as it was but without the information about the data from the MFiX files.

3.5.1 Data reading and handling

L315: The particle and contact data described in Sect. 3.2 are retrieved from the MFiX .vtp file using the variable names specified in the configuration file (Appendix A).

Regarding contact data, Pysammos reconstructs the full bidirectional contact list by duplicating each particle pair, applying the opposite force to the second particle, and assigning the same contact point to both entries. For concave particles, multiple contact points between the same particle pair are explicitly reported and are handled as distinct contacts in our analysis. For convex particles, in contrast, the contact is represented by a single point at the centre of the overlapping volume.

Several data mapping structures and algorithms are used in the pre-processing stage prior to applying CG. Following the data ingestion, DEM data are sorted by particle ID, facilitating the mapping of contact data onto the global particle data (Appendix D1), so that branch vectors between particle pairs can be computed. Subsequently, pre-processing steps are applied to ensure data quality, including the

identification and removal of duplicated contacts written during parallel execution (Appendix D3). During branch vector computation, periodic boundaries are accounted for by evaluating displacements to the nearest periodic equivalent (Appendix D2).

- Comment 7: *Figure 6: Could you check this plot? Indeed, the Heavyside function does not verify the formulation detailed in Eq. 11. The function should be =0 for $w < r$, while the plot shows a non-null value for $w < r < 2w$. This verification should be conducted for the Lucy and Gaussian functions.*

Response: Thank you for raising this issue, indeed, this point is not clear enough in the text. The kernels in Figure 6 are shown in w -scale, which is the smoothing parameter that controls the spread of the kernel and determines the variance. Moreover, the kernels have been scaled such that their variance matches. That way, the differences in Figure 6 highlight the differences in shape given by the function, instead of the w or c . In the original manuscript, only the following sentence addressed this point:

Figure 6 provides a comparison of the three CG functions by shape, as the scale has been normalised by imposing equal variance.

Therefore, we have included a more complete description of the parameters c and w , and the construction of Figure 6. In section “3.5.3. Weight calculation” we added the following text to make the definition consistent and more transparent:

L336: The macroscopic continuum fields result from convolution of a microscopic discrete field with a CG (smoothing) function, Ψ , over the model space, thus performing a spatial weighted average of particle quantities. The smoothing function is defined by two parameters: the spatial domain, c , over which Ψ has non-zero values; and the effective smoothing width, w , which determines the spread or variance of Ψ , thereby controlling how particle contributions are spatially distributed within the volume of radius c .

L341: Various kernel functions that may be employed in coarse-graining are implemented in Pysammos: the Lucy (Eq. (9)), Gaussian (Eq. (10)), and Heaviside (Eq. (11)) functions. Different kernel forms relate c and w with different scaling factors. To provide a consistent comparison across kernels in Figure (6), all functions are rescaled so that they share the same variance. This ensures that the differences between kernels are due to the function shape and not the difference in effective smoothing width w or the kernel spatial domain c .

We also modified Eq. (11) to be with respect to the spatial domain measure (c) instead of the variance (w), for consistency.

$$\Psi(\mathbf{r}) = \begin{cases} \frac{1}{\varphi}, & r \leq c, \\ 0, & r > c, \end{cases}$$

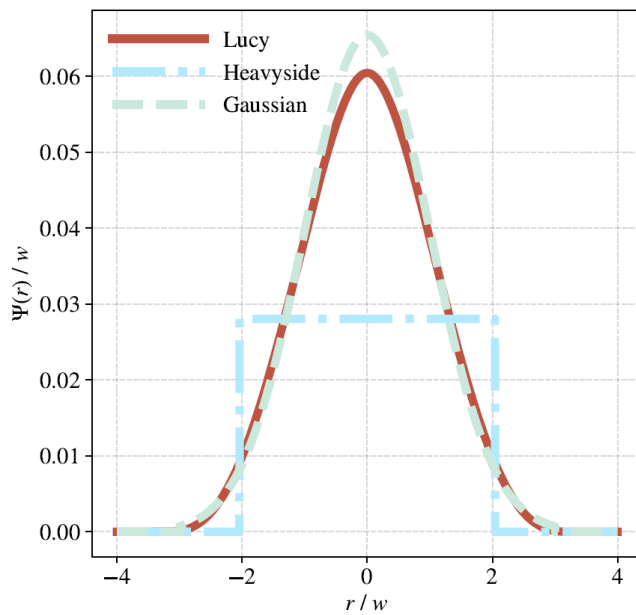
L352: where $\varphi = 4/3 \pi c^3$ is the volume of a sphere with radius c .

Additionally, in the “2. Theory” section, we modified the wording of the following sentence:

L147: The net effect of Ψ is the smearing of the Dirac delta function of Eq. (2) across a volume of radius c and effective smoothing width w (see Sect. 3.5.3).

- Comment 8: Figure 6: I guess that $W(r)$ should be $\Psi(r)$. If not, could you specify the definition of $W(r)$?

Response: Thank you for pointing this out. We agree that, indeed, the symbol for the coarse-graining function should be consistent across the text and figures of the article. Therefore, we have modified the $W(r)$ for a $\Psi(r)$.



L367: Figure 6: Visualisation of the three implemented coarse-graining weighting functions, $\Psi(r)$, in one dimension (r) normalised by the effective smoothing width (w). Note that the functions have been plotted with equal variance in order to ensure that any differences are due to kernel shape, not scale.

- Comment 9: *Figure B2: Same remark as Figure 6: I guess that $W(r)$ should be $\Psi(r)$. If not, could you specify the definition of $W(r)$?*

Response: Thank you for pointing this out. We agree that, indeed, the symbol for the coarse-graining function should be consistent across the text and figures of the article. Therefore, we have modified the $W(r)$ for a $\Psi(r)$.

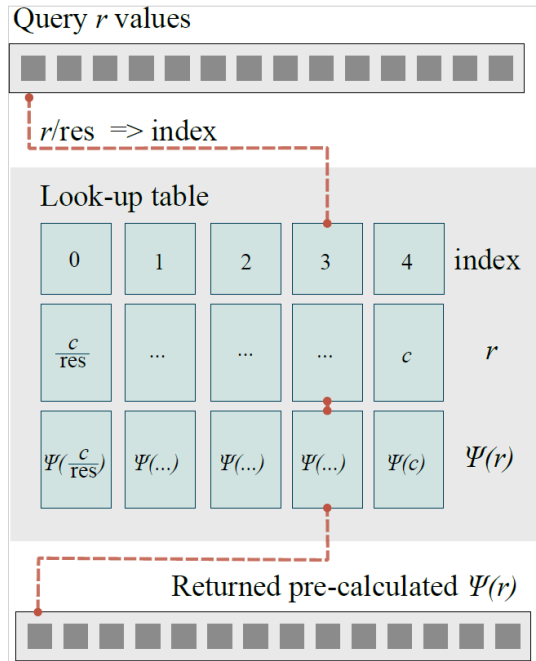


Figure B2: Schematic diagram of the lookup table algorithm implemented in Pysammos. r corresponds to the distance between a particle and a grid point. The resolution of the lookup table is res . $\Psi(r)$ is the coarse-graining function, and c corresponds to its cut-off distance (i.e., $\Psi(c)=0$).

- Comment 10: *L. 529: This depth-independence of the vertical pressure is controversial. I agree with the Authors that Fig. 13 does not show a clear increase in the pressure. However, the relation $P=\rho \cdot g \cdot h$ is well established in the literature. I guess this relation is retrieved for the configuration depicted if a larger height h or a larger density ρ is considered. Could you add limitations to the conclusion raised concerning the homogeneous vertical pressure with depth?*

Response: Thank you for bringing up this important point. We agree that $P = \rho gh$ is very well established for hydrostatic and hydrostatic-like granular pressure profiles, and that our result should not be read as contradicting it in general — only as reflecting the specific, small-scale, wall-confined configuration simulated here. We have revised the text to make this limitation explicit. We have added a clarifying sentence to note that the observed depth-independence is a feature of this specific

geometry rather than a general property of granular piles. In the new version, we now state:

L571: This is visible in Fig. 13e, where in the wall-confined region (heights of 0.005–0.03 m), horizontal stress increases with depth while vertical stress remains constant — indicating that the walls are supporting part of the pile's weight. We note that this apparent depth-independence of vertical stress is not expected to persist as the pile height grows, given that the wall-supported fraction of the total weight would saturate, while the unsupported fraction below saturation depth would grow with height approximately following a hydrostatic profile. Finally, we find that the different phases identified by grain diameter contribute differently to the bulk stress at different heights, although the relatively low polydispersity of this mixture means all phases contribute roughly equally to the bulk field along the profile.

We have also plotted the horizontal stresses along the centreline (grey line in Fig. 13.e), to better explain the Janssen-like effect that occurs in wall-confined granular piles, whereby vertical stress saturates with depth as it is redistributed laterally through friction at the side walls.

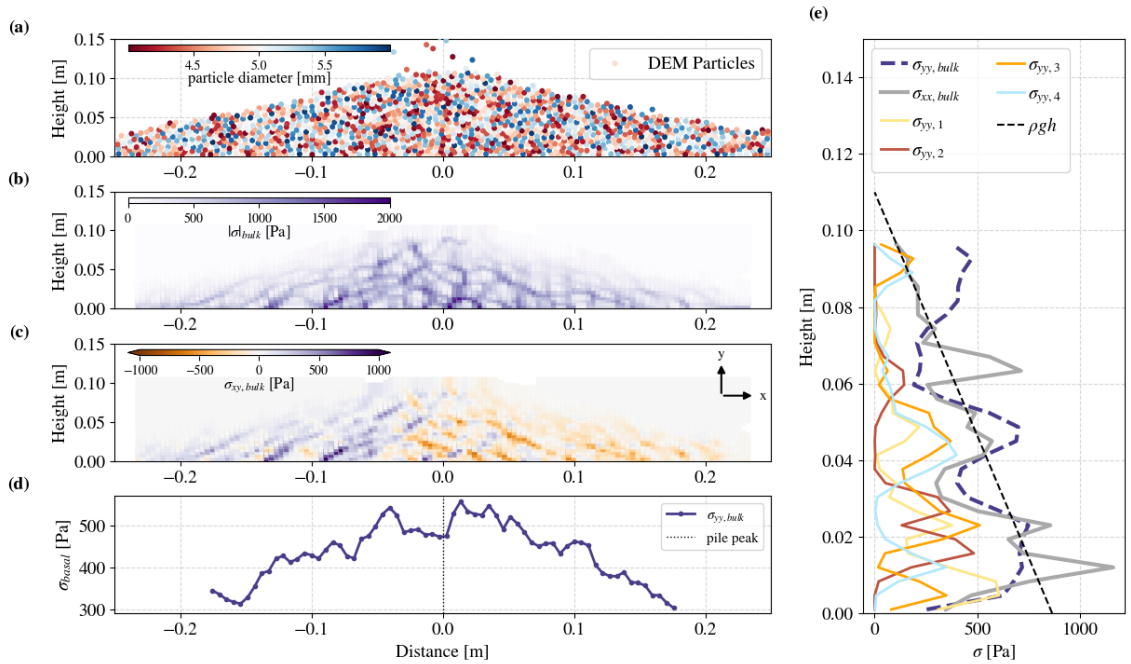
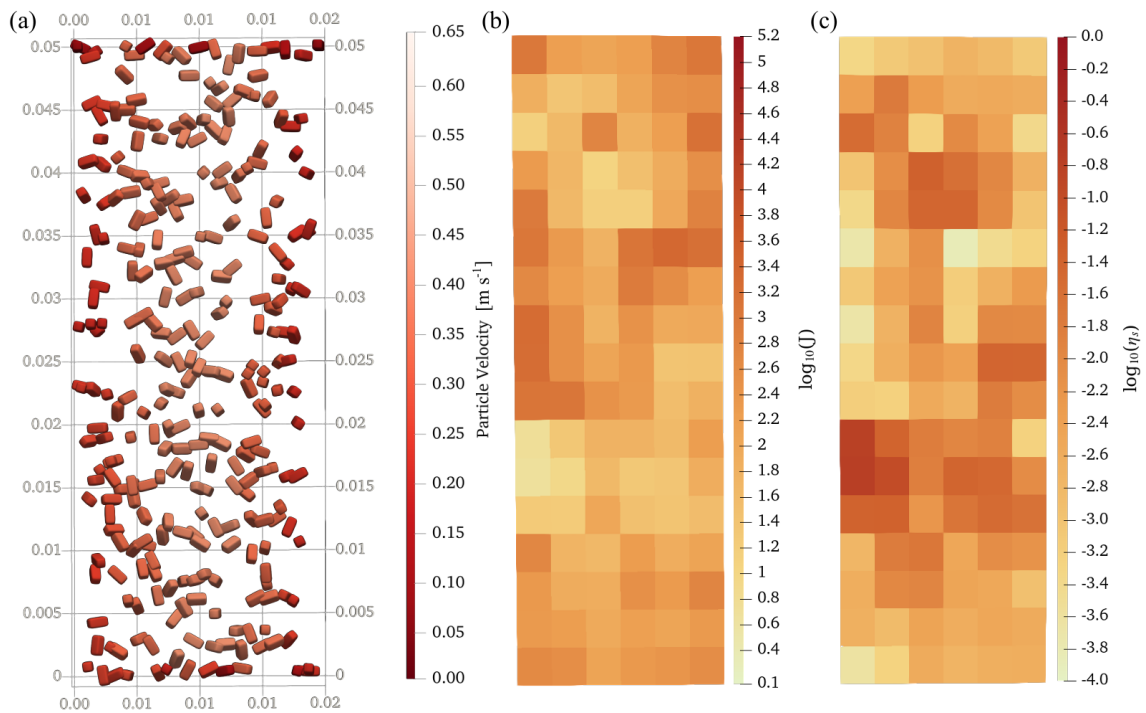


Figure 13: Analysis of a 2D DEM simulation of a growing granular pile carried out with Pysammos's Python-supported visualisation. Simulation particles coloured by size (a), coarse-grained magnitude of the total stress tensor (b) and shear stress (c) across the granular mixture. Coarse-grained basal vertical stresses, taken from the lowest row of CG cells, as a function of horizontal distance, testing for stress arching beneath the pile crest (dotted black line, $x=0$) (d). Vertical profiles, taken at the pile centre ($x=0$) in (d), of the horizontal stress of the mixture (grey solid line), vertical stress of the mixture (dashed blue line), and vertical stress of each of the phases

with diameters 5.7mm, 4.7mm, 5.1mm, and 4.2mm, in yellow, red, orange and blue, respectively (e).

- Comment 11: Figure 12: Could you consider modifying the scale used for the right map? The color assigned to the value -6.6 does not seem to be employed in the map. The range used for the scale can be shorter.

Response: Thank you for pointing this out, it had escaped our attention. We agree that the colour map and the corresponding colour bar should be coherent. Previously we had used the option to scale the colour map by the range of all the simulation time steps, hence the greater range in the colour map than in the plot. We have now changed the colour range to correspond only to the data at time step shown in Figure 12:



- Comment 12: L. 574: As discussed with Section 4.4.3, the default value of w ($=0.75*d_{43}$) may not be the accurate choice to make. Maybe a default value of w that depends on the mean distance of the particles is a more resilient selection. In particular, if this code is used by a standard user (see the definition of the profile in Section 3.1.3). This remark is more a hint to the Authors, the text of the article does not have to be changed. I have not tried; maybe it is a bad idea.

Response: Thank you for this suggestion. Indeed, the discussion we advise the use of a larger smoothing width than the default ($w = 0.75*d_{43}$) for dilute granular mixtures. The default w we propose simply falls within the range that other studies

have reported to be stable coarse-graining scale (Goldenberg et al., 2006, Weinhart et al., 2013). We explicitly encourage the readers to choose and validate the smoothing width with caution, according to their specific application. Your recommendation, namely relating the smoothing width to the mean interparticle distance, is another reasonable and potentially more general estimate. We appreciate this valuable suggestion and will consider exploring and evaluating this approach in future work. Since this comment is intended as a general recommendation rather than a correction to the current methodology, we have not modified the manuscript.

- Comment 13: L. 683: *The consideration of the particle-wall interactions seems in Equation 5 to be already conducted (with the particles between $N+1$ and $N+N_b$). If it is finally not the case, could you delete this part of the Equation?*

Response: We agree with your comment. We initially included the formulation for particle-wall interactions because it has been done in the literature and this is section about the general theory of coarse-graining. However, we agree that it is misleading to include the particle-wall interaction in Eq. 5 if we have not implemented it in the code. Therefore, we have removed it from Eq. 5 and clarified that it is an addition that has been implemented in the literature. Note that the correction shown below also includes the modifications originated from Comment 4:

$$\sigma^c(\mathbf{r}, t) = \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^{C_{ij}} \mathbf{f}_{ij}^k \mathbf{a}_{ij}^k \int_0^1 \Psi(\mathbf{r} - \mathbf{r}_i + s \mathbf{a}_{ij}^k) ds,$$

where N is the number of particles in the system, and C_{ij} is the maximum number of contact points between two particles. [...]

L167: Note that Eq. (5) does not account for wall-particle interactions; this extension is discussed in detail in Weinhart et al. (2012).