

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.

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

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.

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

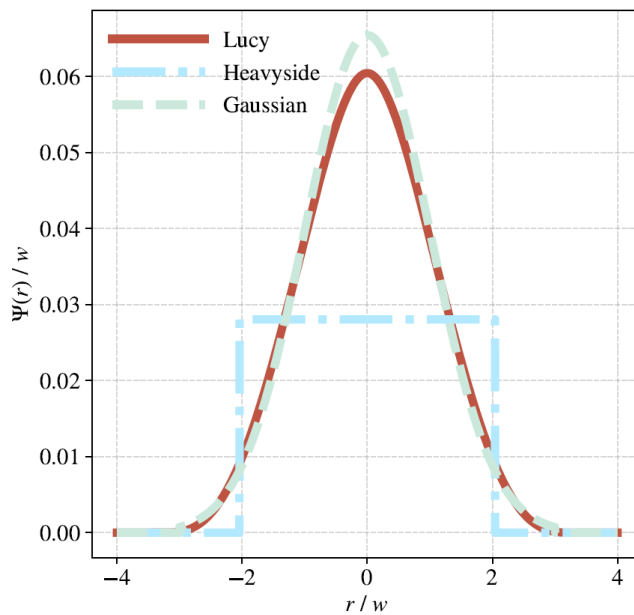
L352: where $\varphi = 4/3 \pi c^3$ is the volume of a sphere with radius c .

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 $\Psi(r)$.

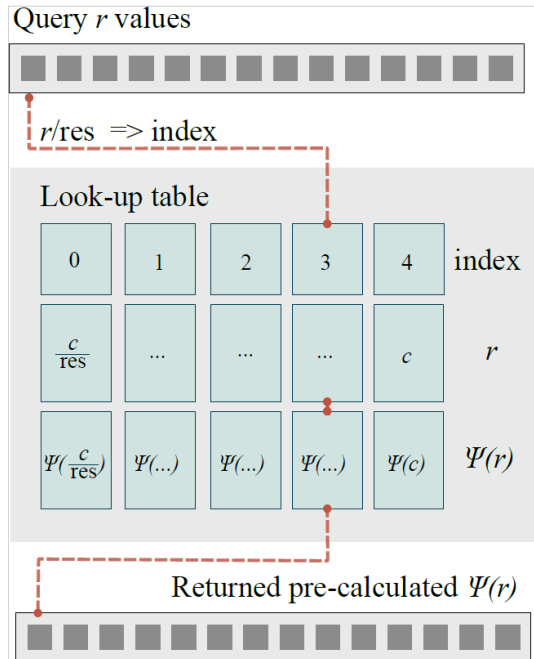


Figure B2: Schematic diagram of the lookup table algorithm implemented in Pysammos. r corresponds to the distance between a particle and a grid point. The resolution of the lookup table is res . $\Psi(r)$ is the coarse-graining function, and c corresponds to its cut-off distance (i.e., $\Psi(c)=0$).

- 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.

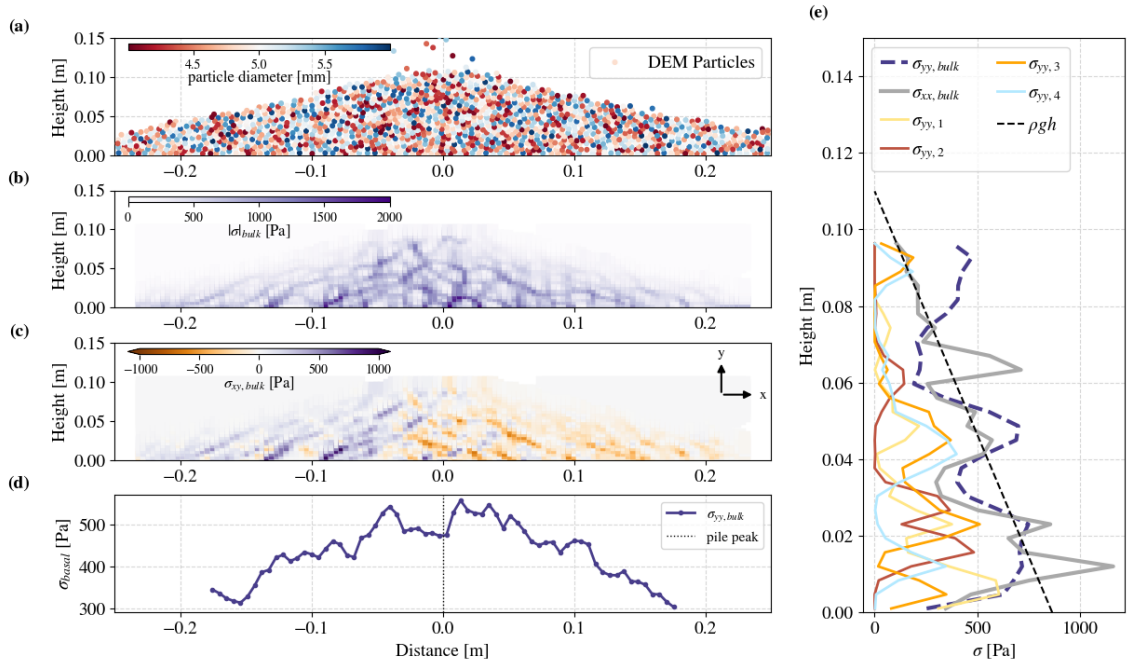
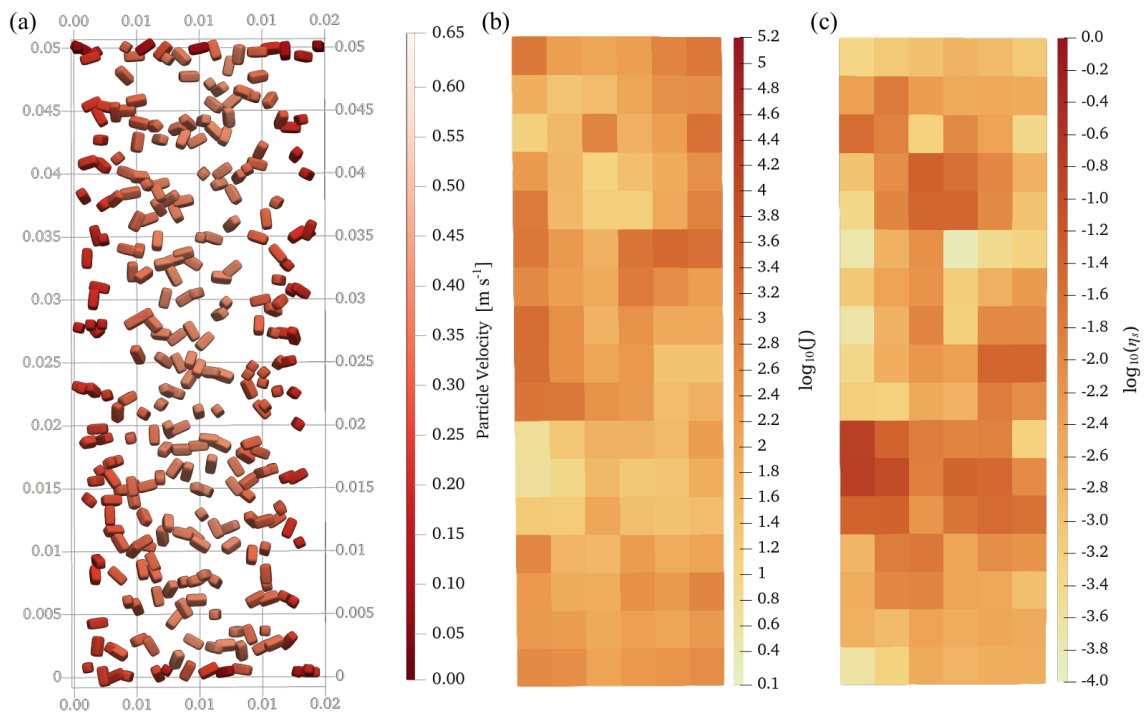


Figure 13: Analysis of a 2D DEM simulation of a growing granular pile carried out with Pysammos's Python-supported visualisation. Simulation particles coloured by size (a), coarse-grained magnitude of the total stress tensor (b) and shear stress (c) across the granular mixture. Coarse-grained basal vertical 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).

We thank the referee 2 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 2

This paper describes a coarse-graining software, Pysammos, dedicated to the spatial averaging of discrete element modelling data, in particular the one produces by the software MFIX-DEM. The principle of the method and software are explained, as well as its specificities, and its performance

are compared with other coarse-graining codes and with different geometries and simulation conditions.

The paper is overall well written and detailed, and can be useful as an entry point to coarse-graining, while proposing additional features not often present in already existing solutions. Some of the specific aspects, such as the phase identification and the validation with non-spherical particles, can be useful in a broad range of contexts. I only have some minor comments and questions below.

- Comment 1: In the simulations of particles suspended in a magma, it is not clear how the flow is created, and if it is a one-way or two-ways interaction between the fluid and particles. It would be interesting as well to provide the volume fraction field, which may have some spatial variations. Also, the simulations are indicated as 2D, but the particles seem to be three-dimensional?

Response: Thank you for raising this point. Indeed, further information about the simulation set up would be beneficial for this showcase example. Therefore, we have addressed these details in the results section where the example is introduced:

L533: Hence, we present a DEM-CFD pseudo-2D (i.e., with a thickness of one cell) numerical simulation representing non-spherical crystals suspended in magma flowing cyclically in the positive vertical direction through a narrow conduit with frictional walls (Fig. 12.a). The particles are fully three-dimensional and interact with the fluid through a two-way coupling via drag and pressure forces. The flow is driven by a constant flux, causing the particles to accelerate until they eventually reach steady state. Given the sparsity of the solid particles, we chose a smoothing function half-width four times larger than the default option, $0.75 \times d_{43}$.

We agree that we should provide the volume fraction field, since we claim that the average is below 0.1 in line 517. Hence, we have added it to the Supplementary Material, as Figure S8 and cross-referenced it in the text (line ??).

L546: Therefore, for the low concentration of suspended particles ($\phi \ll 0.1$) used in this example (see Fig. S9 of the Supplementary Material), very low effective shear viscosity is to be expected.

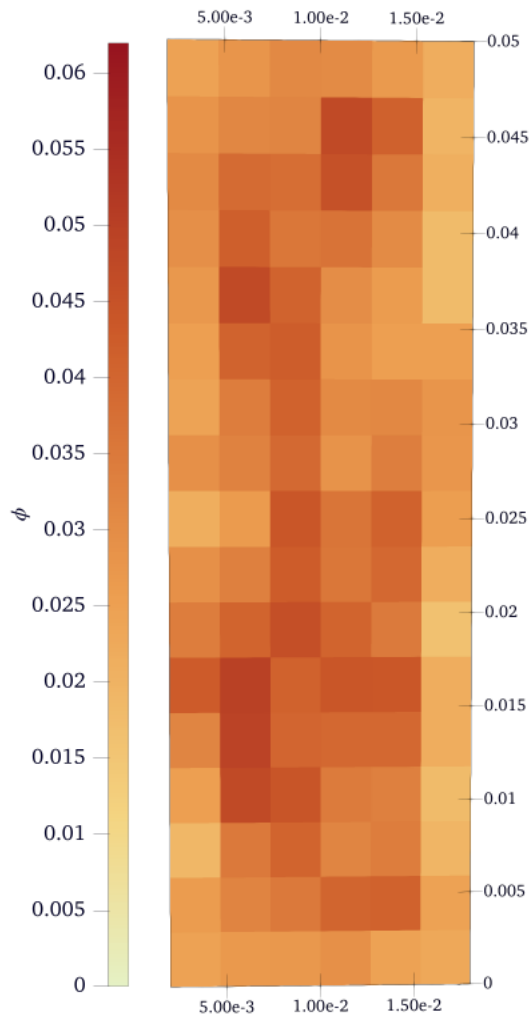


Figure S9: Visualisation of the volume fraction field that corresponds to a DEM-CFD pseudo-2D numerical simulation representing non-spherical crystals suspended in magma.

- Comment 2: *The results on the stress under the pile (4.4.4) are quite surprising. Experimental observations show essentially hydrostatic pressure, sometimes with a dip of pressure below the centre, but both appear quite different from the results reported. Even if frictional redistribution of the stress occurs (similar to the Janssen effect), the weight of the grains still needs to be supported somehow. It would be good to also look at the shear stresses to assess whether this frictional redistribution does occur. Also, Fig 13a may be better plotted with a linear scale, since again the expectation would be a linear stress increase with depth on average.*

Response: Thank you for raising these points, which we address in turn. We agree that a linear scale of Fig. 13.a (now 13.b) enables a more accurate visualisation of the stress distribution across the pile, and hence, we have modified it. Further, to improve clarity, we have plotted the particle position on a different plot (now Fig. 13.a). We agree that this example would benefit from a more robust and clear display of stress distribution and redistribution. Hence, we have added the map

of the total stress tensor magnitude (Fig. 13.b), the map of the shear stress (Fig. 13.c) and the horizontal profile of basal vertical stress (Fig. 13.d). We hope that the combination of these three quantities yield a more diagnostic stress redistribution. The total stress tensor magnitude shows the force chain arching, whereas the basal vertical stress shows a distinct dip beneath the apex of the pile. Notably, peak shear stress occurs not immediately at the walls but within the lower-mid region of the pile, consistent with force chains curving from the crest through the granular bulk before reaching the walls, where the resulting lateral thrust is partially supported, in a Janssen-like manner. These points have been added to the corresponding section of the manuscript:

L549: This example highlights the package's ability to (1) analyse and visualise Pysammos results entirely in Python, (2) resolve individual phases within a polydisperse mixture, and (3) straightforwardly assess each phase's contribution to the bulk properties of the mixture. DEM can also be used to study vertical load redistribution in loose sediment — for instance, in debris-flow deposits posing a risk to nearby communities and infrastructure. Here, we present a 2D DEM simulation of a growing granular pile, in which grains of different sizes are poured from above into a confined space between two walls 50 cm apart (Fig. 13a). This model captures two combined effects: force-chain arching in a free-surface granular pile (Brockbank et al., 1997), and the Janssen effect, whereby vertical stress saturates with depth as it is redistributed laterally through friction at the side walls (Windows-Yule et al., 2019).

L557: Through this example we see that, near the peak of the pile, basal pressure exhibits a local minimum, rather than the maximum that a hydrostatic gradient of a fluid would predict directly beneath the peak (Fig. 13.d) — a phenomenon that has been documented experimentally (Brockbank et al., 1997). This occurs because granular assemblies transmit stress anisotropically along chains of contacting grains, rather than isotropically as in fluids, redirecting stress sideways by "arching" it toward the flanks of the pile rather than accumulating it directly underneath. These arching structures are visible as high-magnitude bands in the total stress tensor magnitude field (Fig. 13b). This effect is even more pronounced in this 2D configuration, compared to a 3D pile.

The mechanism behind this stress redistribution is more explicitly shown by the shear stress field (Fig. 13.c). Unlike the stress magnitude, which illustrates where stress locally intensifies, the sign of the shear stress specifies the direction in which load is being carried: shear stress is close to zero along the vertical centreline and grows with opposite sign toward each flank. This confirms that weight is indeed transported away from the pile's apex in both lateral directions, rather than straight downward. The largest shear stress magnitudes do not occur at the side walls, but midway between the peak and the walls. This is consistent with force chain curving

outward and downward through the pile before reaching the walls. This suggests force chain arching is probably the dominant redistribution mechanism here, with the walls providing at least a partial frictional contribution to support, consistent with a weakly-developed Janssen effect. 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.

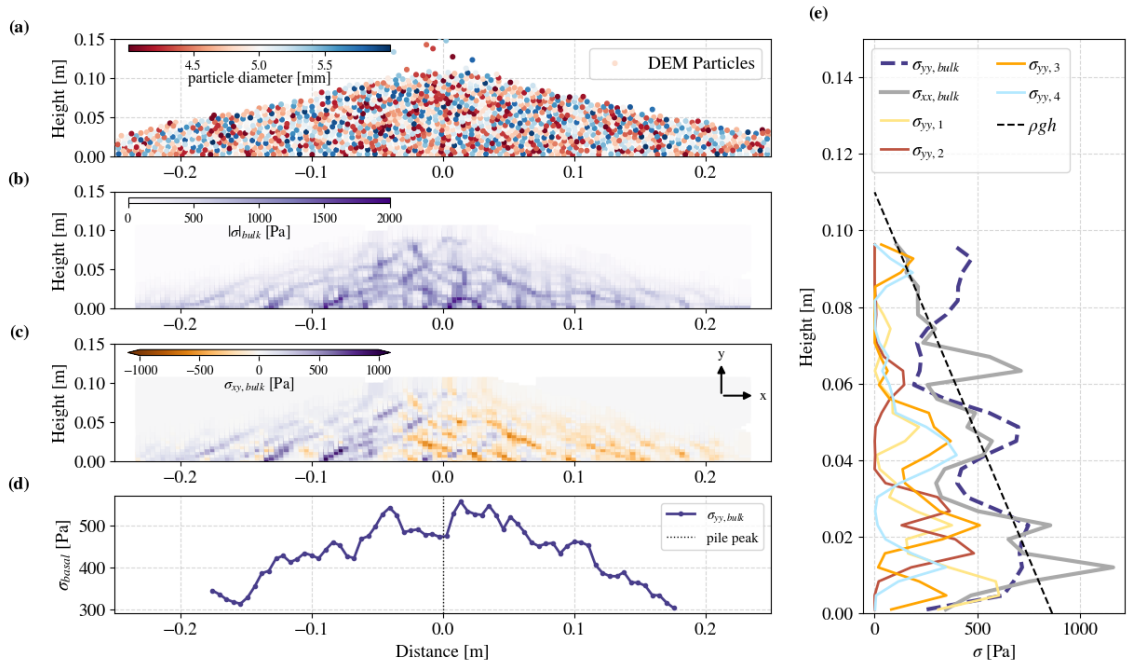


Figure 13: Analysis of a 2D DEM simulation of a growing granular pile carried out with Pysammos's Python-supported visualisation. Simulation particles coloured by size (a), coarse-grained magnitude of the total stress tensor (b) and shear stress (c) across the granular mixture. Coarse-grained basal vertical 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 3: *The calculation of the contact stress involves integrating the windowing function over the branch vector, as shown in equation 5. Is this approximated through averaging the values at the centre, as is commonly done, or is the integral numerically solved? How? This may explain some of the differences pointed out in σ_c later on in the paper.*

Response: Thank you for raising this point, which was indeed underspecified in the manuscript. The line integral in Eq. 5 is not approximated by evaluating the windowing function at the branch-vector midpoint. Instead, Pysammos solves it via trapezoidal numerical integration, sampling the weight function at multiple points along the branch vector between the particle centre and the contact point, and integrating those samples numerically. We have added this explanation to Section 3.5.3:

The weights are pre-calculated for a range of particle-grid point distances, [...] lookup table algorithm, details of which can be found in Appendix C.

L355: The calculation of contact-related fields (e.g., the contact stress tensor, Eq. (5)) requires the additional line integral of Ψ along the branch vector connecting a particle centre to a contact point. This integral could, in principle, be solved in analytically for each kernel individually (Weinhart et al., 2012). In Pysammos, we instead evaluate it numerically: the branch vector is discretised into $N=10$ points, sampled through the same lookup table employed for particle weights, and integrating with the trapezoidal rule. This provides a single, kernel-independent implementation of the integral. Increasing N beyond 10, up to 10^5 , produced no discernible change in the resulting contact stress fields (Fig. S7 of the Supplementary Material), confirming that the trapezoidal integration is already converged at this resolution.

We investigated the effect that the number of trapezoidal integration bins, N , has on the contact stress results. To assess this, we carried out a test that compared results yielded by $N=10$ and $N=10^5$, at low w/d , the range that shows larger discrepancy with MercuryCG. This was added to the Supplementary Material and referenced in the Section 3.4.3, to ensure the reader that the choice of N does not significantly compromise the results.

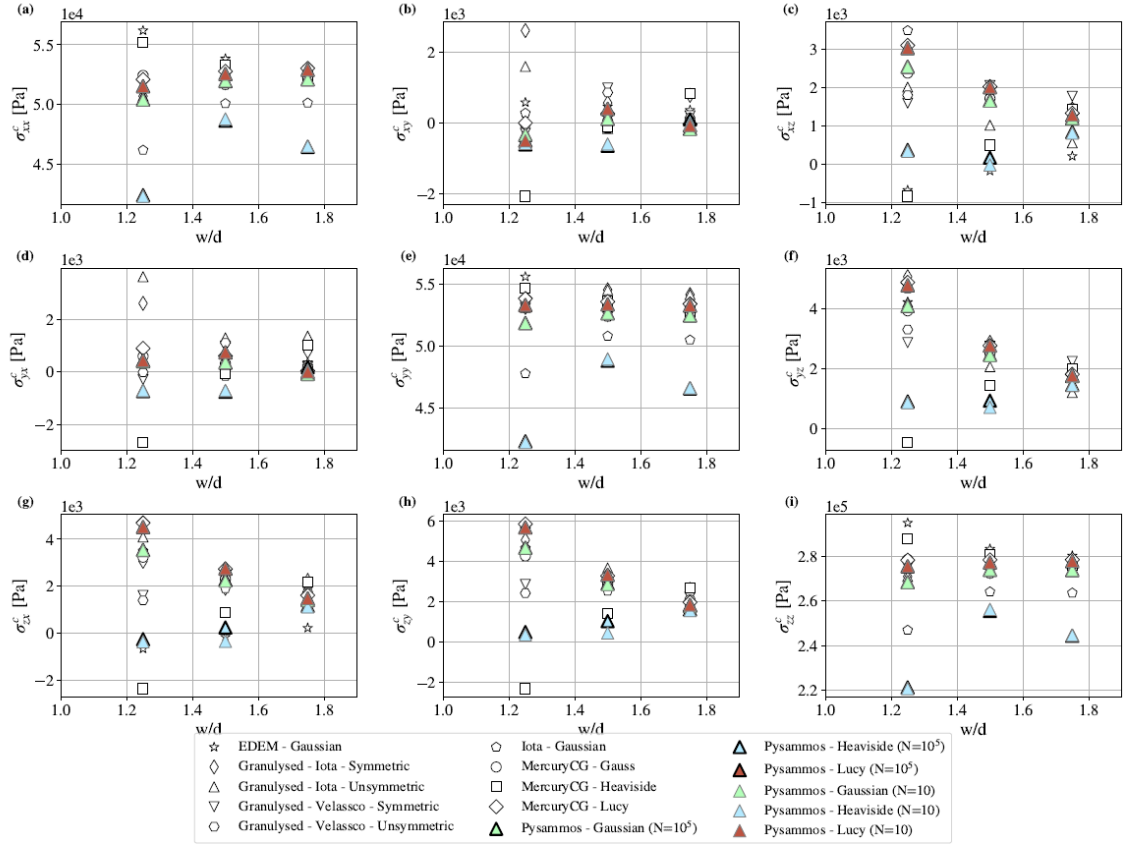


Figure S7: Benchmark of the contact stress for a monodisperse cubic lattice of grains with a diameter of 0.02m and a density of 2500kgm⁻³ (g), evaluated at the point $P = (0, 0, 0.42725)$ m. The stress was coarse-grained using Pysammos with trapezoidal integration and 10 and 105 bins (coloured triangles with black and grey edges, respectively), and using other coarse-graining software (open symbols): EDEM, Granulysed, Iota, and MercuryCG. Where available, results obtained using different smoothing functions (Gaussian, Heaviside, and Lucy) are compared over a range of smoothing half-widths, expressed as multiples of the characteristic particle size (w/d). Only values of $w/d < 2$ are shown, as this is the range where the largest discrepancies between models occur. Panels (a)–(i) show the components of σ_{ij} arranged in matrix form, with rows corresponding to $i = x, y, z$ and columns to $j = x, y, z$.

The contact stress discrepancies we report between Pysammos and MercuryCG at low w/d (Section 4.1) are therefore likely to reflect the particle-search volume being centred on particle i rather than on the contact point, which we already identify as a contributing factor. We have honed the Results section, by providing a more precise reason for the discrepancy in the σ_c field at low w/d values.

L405: The contact tensor field (Fig. 7.d) shows a broadly constant mean value across the analysed w/d range (see Fig. S6.e for enlarged view), with a greater scatter at the low end. Pysammos and MercuryCG show a degree of discrepancy at lower w/d . This could be attributed to the difference in the particle-search

criterion between the two codes: in Pysammos, a contact is included in the averaging volume based on the distance from the grid point to the centre of *particle i*, whereas MercuryCG's determines inclusion based on the minimum distance from the grid point to the branch vector itself (Weinhart et al., 2020). This more general criterion allows MercuryCG to include contacts whose branch vector is positioned near the grid point even when the particle centre itself lies outside the cutoff radius, an effect expected to be more pronounced at low w/d , where the search radius is comparable to the branch-vector length. We note that this discrepancy is unlikely to stem from the numerical evaluation of the branch-vector line integral in Pysammos (Sect. 3.5.3), as the trapezoidal integration is already converged at $N=10$ sampling points (Fig. S7 of the Supplementary Material) and is not expected to be a significant source of error even at low w/d . Further, the Heaviside function in Pysammos shows larger disagreement with that of MercuryCG across the whole w/d range and especially in the diagonal components of [...]

- Comment 4: *There are many different clustering algorithms that exist, how was the particular one used chosen?*

Response: We agree that specifying the clustering algorithm employed in the phase detection is very relevant. In line 278, we already mention that “In Pysammos the phase detection is carried out by clustering particle data using the k-means algorithm, explained in detail in Appendix B1”, and subsequently explain the k-means algorithm in depth in Appendix B1.

- Comment 5: *line 144: the mathematical detail of the Dirac delta function is inaccurate. Technically, the Dirac function is infinity at the particle centre, and 0 otherwise, such as its integral over the whole space is 1.*

Response: Thank you for pointing this out. We agree that the mathematical detail of the Dirac delta function is inaccurate. Therefore, we have modified this sentence to:

L141: Note that δ is zero everywhere except at the particle centres, where it is infinitely high, and has a total integral of one.

- Comment 6: *Equations 5 and C4: the tensor product, or Kronecker product that is used for the vector in those equations, should be specified. It is usually denoted with $\otimes$.*

Response: Thank you for bringing this up — the notation was indeed ambiguous as written. Because we use component-free (bold tensor) notation throughout, the juxtaposition of two vectors does not carry a single unambiguous meaning the

way it does in index notation, where two free indices (e.g. $f_p a_q$) automatically identify a rank-2 tensor. In bold notation, a reader could plausibly interpret the juxtaposition as an inner product, which would incorrectly yield a scalar rather than the rank-2 tensor we intend. To remove this ambiguity, we have introduced the $\otimes$ symbol to denote the dyadic (outer) product explicitly. Equations 5 and C4 have been updated accordingly, and we have checked the rest of the manuscript for consistency with this notation:

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

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

- Comment 7: *line 193: the use of only equilibrium states to justify the replacement of ensemble averaging with time averaging appears quite restrictive. Most granular systems are out of equilibrium, and time averaging can still be meaningfully performed if the systems are at steady state.*

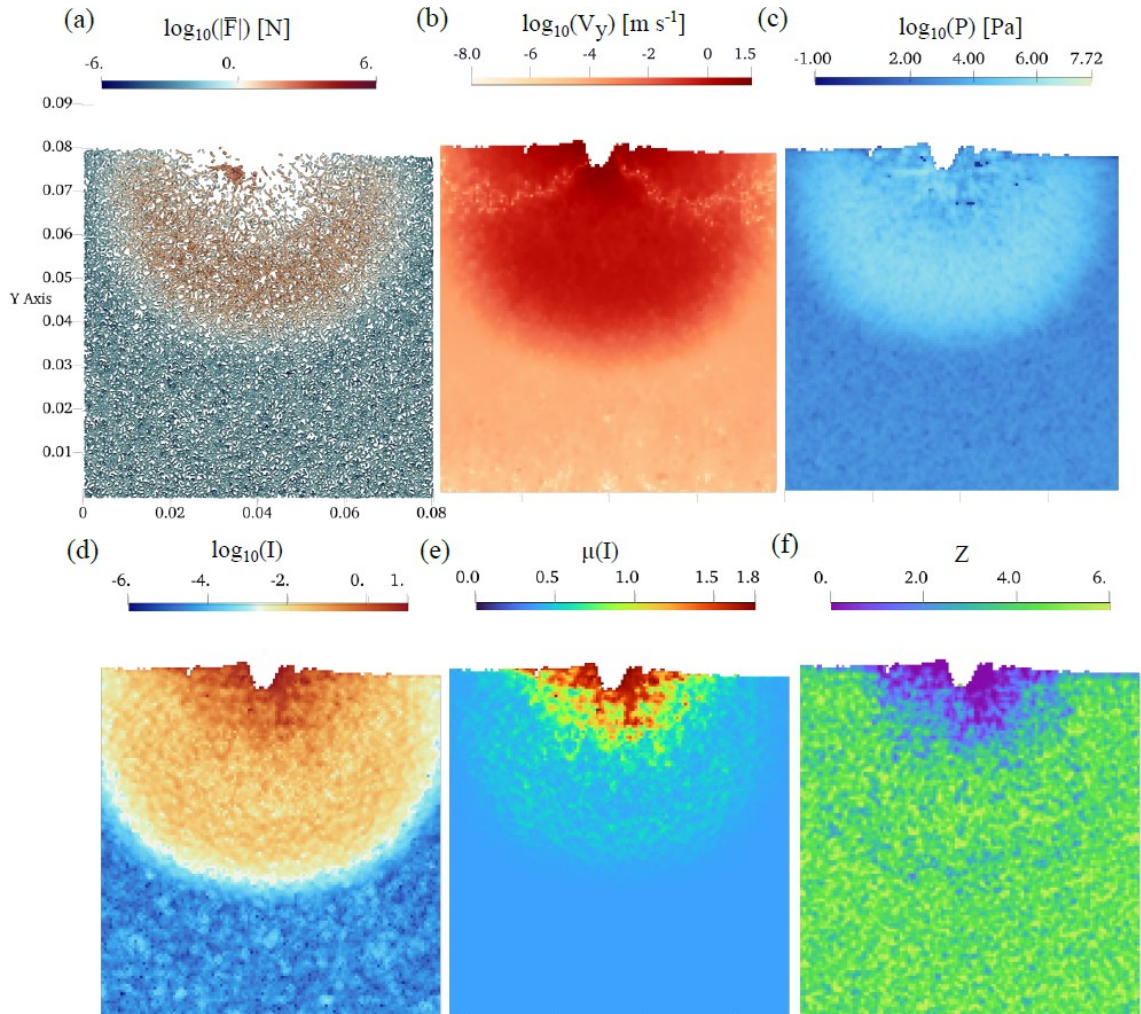
Response: Thank you for this comment. We agree that framing the ensemble-time averaging equivalence purely in terms of equilibrium states is too restrictive, since granular flows are inherently dissipative. We have revised the text to clarify that the relevant condition for this equivalence is statistical stationarity (steady state) rather than equilibrium in the strict thermodynamic sense. We have added a citation to Babic (1997) in support of this point. The manuscript has been modified accordingly, as shown below:

L189: Traditional statistical mechanics considers multiple realisations, or probabilistic copies of a system, over which the corresponding macroscopic quantities are averaged (Jaynes, 1957). This conceptual tool is equivalent to averaging over the total time range, an equivalence classically derived for ergodic systems in equilibrium (Huang, 1987), but which extends to systems at (non-equilibrium) steady state, where the relevant statistics are stationary in time (Babic, 1997). This is advantageous for systems that are complex to simulate over large time scales. In the case of granular media we are able to simulate complex granular systems for longer time scales with DEM. Hence, in the field of computational granular physics, ensemble averaging is not required (Glasser, 2001), and in fact could be problematic due to the inherent lack of scale separation. Instead, the analytical expressions for continuum fields tend to include a time-averaging, as well as a space-averaging, which acts as a low-pass filter on the inherent fluctuations of granular flows, to obtain steady-state solutions (Babic, 1997; Glasser, 2001). Nevertheless, the time averaging is not strictly required, and in fact, even moving window Heaviside temporal averages

could obscure time-dependent trends in highly transient flows, as pointed out by Weinhart et al. (2012). Hence, in this study only spatial averaging is considered for the analytical equations of coarse-graining.

- Comment 8: *Fig. 11b: the scale shows velocities going up to 10^{30} m/s, which is much higher than the speed of light. There is probably an issue with the scale.*

Response: Thank you for spotting this, as it was clearly wrong. We corrected the colour scale range in ParaView and now it presents values that are more reasonable, with an upper bound of $\log_{10}(V_y)=1.5$.

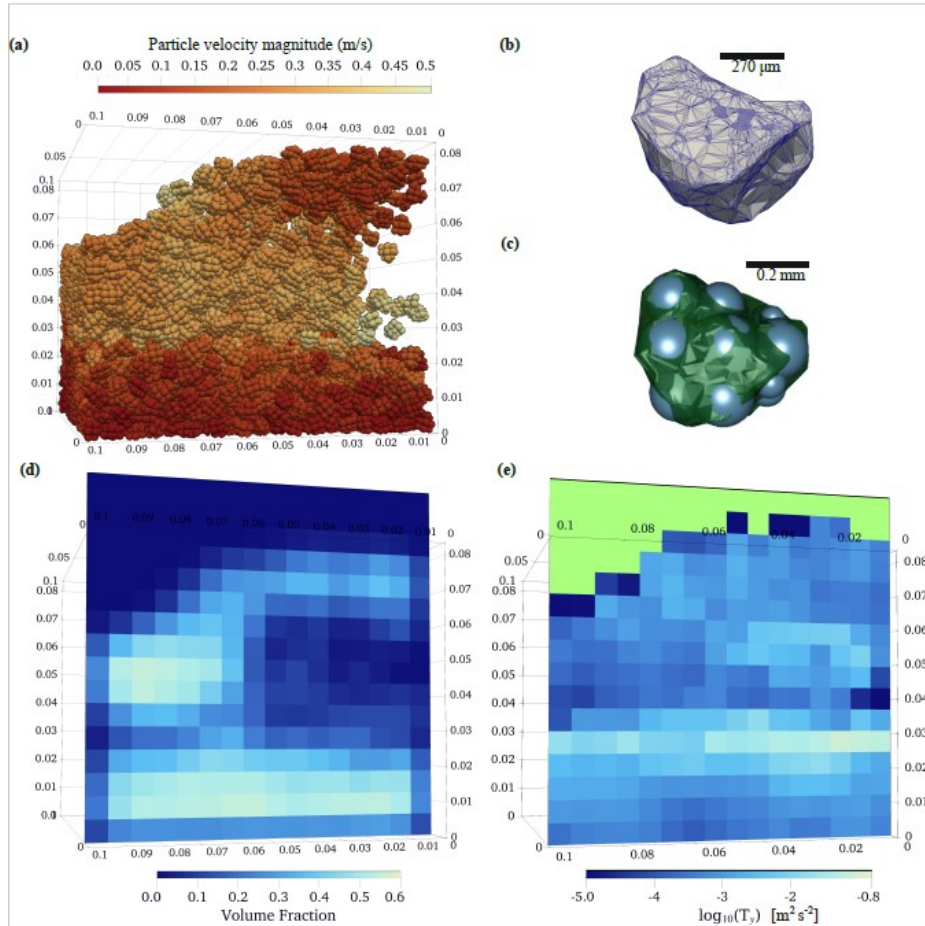


- Comment 9: *Fig 15 caption: the explanation of the "greyish" colour is incorrect. T_y is not zero at these locations, but undefined, since the volume fraction is zero (no particles). As indicated by equation C13, this would correspond to dividing a zero kinetic stress tensor trace by the zero density at that location, which numerically should result in a Not a Number.*

Response: Thank you for bringing this up, as this was unclear. Indeed, in the regions where the mixture density is zero, T should be Not a Number (NaN), owing to the indeterminate division of zero by zero. Nevertheless, we force T to zero when the denominator in Eq. C13 (i.e., the mixture density) is zero. This is a developer choice and it was done with numerical safety in mind, as downstream code may expect finite numbers — for example, the calculator filter in ParaView. Thus, this safety-over-accuracy trade-off avoids NaNs propagating through arrays, plots, and filters. However, we agree that this needs to be explained in the text. We have therefore added an explanation of this design choice to the Appendix section where granular temperature is introduced:

L927: Granular temperature, T , measures the level of fluctuation [...] It is important to note that in regions with no particle influence (i.e., $\rho(r, t) = 0$), T is set to zero, to avoid the undefined division of 0 by 0, and subsequent NaN propagation. Although a NaN would be more mathematically accurate, we have opted for this common numerical-safety practice.

However, it is true that $\log_{10}(0)$ is also NaN, and therefore it should be made clear in Fig. 15. Previously the NaN values were substituted by a 0 in ParaView, which was very misleading. We have modified Figure 15 to show NaN values in a clearly different colour (pale green).



We have accordingly tweaked the caption of Figure 15 explain that $T = 0$ is forced, as well as indicating that $\log_{10}$ of $T = 0$ are coloured pale green, because they are NaN.

L717: Figure 15: DEM simulation of a granular bed, fluidised with gas [...] Coarse-grained fields of the volume fraction (d) and vertical granular temperature (e) corresponding to the simulation time step shown in (a). Note that the pale green colour in (e) indicates NaN. This occurs at locations where T_y is set to $0 \text{ m}^2\text{s}^{-2}$ by convention (see Appendix C). Consequently, $\log_{10}(0)$ is undefined and is represented as NaN in ParaView.

- Comment 10: line 885: it would be good to define D right after eq. C13, rather than on line 895.

Response: Thank you for this suggestion. We agree that the definition of D is more appropriate after the equation that introduces granular temperature T (C13), rather than after the deviatoric stress equation (C15), later on in the text. Hence, we have moved the definition of D closer to Eq. C13:

L921: Granular temperature, T , measures the level of fluctuation of particle velocity about the macroscopic velocity of the granular mixture (Goldhirsch, 2008). It is calculated as a spatial average of the velocity fluctuations squared, since by definition the average of fluctuating velocity (not squared) is zero and hence avoid the cancellation of fluctuating velocities with the same magnitude but opposite sign,

$$T(\mathbf{r}, t) := \frac{1}{D\rho(\mathbf{r}, t)} \text{tr}(\boldsymbol{\sigma}^{k'}(\mathbf{r}, t)),$$

where D is the number of spatial dimensions considered. Note that the scale-independent term of the kinetic tensor must be used ($\hat{\sigma}_k$), as T focuses on the local velocity fluctuations, V_i (Eq. (6) and 7).

- Comment 11: *eq. C19: the equation uses $\bar{d}$, which is not defined. Is it the same as D_{43} (eq. C6)?*

Response: Thank you for this comment. The representative particle size $\bar{d}$ is defined in section 3.4. Mesh generation:

L296: The mesh is created with a resolution w (Eq. (9)) specified in `.set_resolution()`. It is calculated as $w = m_w \bar{d}$, where m_w is a scaling and $\bar{d}$ is a representative particle size. By default, m_w is 0.75 (Weinhart et al., 2016). In polydisperse granular media the choice of $\bar{d}$ is not trivial (Polanía et al., 2025), hence, we provide different metrics for the user to choose from, such as the median diameter or d_{50} , the root mean squared diameter (d_{rms}), volume-weighted average (d_{43}) and area-weighted average or Sauter diameter (d_{32}).

The reference to Appendix C regarding the rest of continuum fields is made in section 3.5.4 Macroscopic fields. However, we have added a sentence after Eq. C19, redirecting the reader to section 3.4:

L952: where $\bar{d}$ is the representative particle size (Sect. 3.4). The inertial number is a measure of the dynamic state of a granular medium. It is also understood as the ratio of (1) the microscopic timescale (or scale of rearrangement), representing the time a particle takes to fall in an a slit of a given diameter under a given pressure; and (2) the macroscopic time scale, linked to the average deformation (Midi, 2004).