DISCRETE ELEMENT SIMULATIONS TO LEARN CLOSURE MODELS FOR GRANULAR FLOWS

SURF 2025 REPORT #1

Amitesh Pandey*
Engineering & Applied Sciences
Caltech
Pasadena
amitesh@caltech.edu

Harkirat Singh†
Engineering & Applied Sciences
Caltech
Pasadena

Lianghao Cao†
Computing & Mathematical Sciences
Caltech
Pasadena

Kaushik Bhattacharya†
Engineering & Applied Science
Caltech
Pasadena

Feburary 22, 2025

ABSTRACT

Discrete Element Method (DEM) simulations are a high-fidelity computational technique used to model granular materials, consisting of discrete particles, to uncover the underlying physics governing various processes. However, DEM scales poorly with system size, making it computationally intractable for simulating real-world systems. Instead, these simulations can be leveraged to inform macro-scale models, which offer a more efficient approach to studying large-scale systems. In this work, we plan to run DEM simulations across a range of cases, varying particle properties, grain-level interactions, flow geometries, and system sizes to explore their effects. We will employ advanced tools to extract meaningful coarse-scale properties from grain-scale simulations, enabling the development of accurate macro-scale models and advancing the understanding of granular material behavior.

Keywords Discrete Element Method (DEM) · Granular Materials · Coarse Graining · Closure Models

1 Introduction and Methodology

1.1 Granular Materials

A granular material is used to refer to a collection of grains that interact with each other primarily through mutual contact forces (Clement [1999]) and are under the influence of more general external forces (like gravity). These systems are widespread in our physical world and of significance in a diverse array of domains. A cohesive model for the flow of granular materials would be of great benefit to industries ranging from agriculture (in understanding how soil particles interact under different environmental conditions), geophysics (through the modeling of snow particles for forecasting landslides or avalanches), and pharmaceuticals (for the industrial optimization of the transport of medicinal pills).

These materials can be described by a range of variable parameters, such as the sizes and densities (together called the phases) of the constituent particles, the boundaries of the material, the applied external forces, the fluid medium of dispersal, and the initial schematic configuration. To understand the complex macroscopic behaviour of granular materials and then control their flow, we simulate the behaviour of individual particles using a famous simulation technique known as the Discrete Element Method (DEM) (Cundall and Strack [1979]).

*Student
†Mentor

1.2 Discrete Element Method (DEM) Simulations

1.2.1 Balance Laws

These simulations model the movement of particles primarily in accordance with Newton's Second Law (and its rotational analogue) (Guo and Curtis [2015]). For a dry material, for any particle i, in translational motion,

$$m_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \mathbf{F}_i^{\mathrm{cf}} + m_i\mathbf{g} + \cancel{\mathbf{F}_i^{\mathrm{fp}}}^{\nearrow 0}$$

and in the rotational motion case, with the inertia tensor $\mathbf{I}_i$,

$$\mathbf{I}_i \frac{\mathrm{d}\omega_i}{\mathrm{d}t} - (\mathbf{I}_i \cdot \omega_\mathbf{i}) \times \omega_i = \boldsymbol{\tau}_i$$

are the laws followed. Here $\mathbf{v}_i$ and ω_i are the translational and angular velocities, respectively. m_i is i's mass, and $\mathbf{g}$ is gravitational acceleration as usual. $\mathbf{F}_i^{\mathrm{cf}}$ is the net *contact force* between i and surrounding particles, and $\mathbf{F}_i^{\mathrm{fp}}$ is the *fluid-particle* contact force, and finally the torque applied that induces rotational motion in i is $\boldsymbol{\tau}_i$. In the DEM technique, the progression of some of these physical quantities (in particular, the velocities and contact forces) is traceable through the time integration of the equations of motion using the computational sequence produced in the simulation. For the DEM distribution we use, `LAMMPS` (Thompson et al. [2022]), the velocities and contact forces are approximated using current & historic positions with the *Verlet* scheme (Plimpton [1995]), discussed in Appendix A.1.

1.2.2 Contact Force Model

The total contact force $\mathbf{F}_i^{\mathrm{cf}}$ acting on particle i due to its interactions with other particles is decomposed into normal and tangential components arising from pairwise contacts. In our `LAMMPS` implementation, these are calculated according to a linear spring-dashpot model with frictional history (through the `pair_style gran/hooke/history` flag):

$$\mathbf{F}_i^{\mathrm{cf}} = \sum_{j \neq i} \left(\mathbf{F}_{ij}^n + \mathbf{F}_{ij}^t\right)$$

Here, the normal ($\mathbf{F}_{ij}^n$) and tangential ($\mathbf{F}_{ij}^t$) contact forces between particles i and j are calculated as

$$\mathbf{F}_{ij}^n = k_n \delta_{ij} \mathbf{n}_{ij} - \gamma_n \mathbf{v}_{ij}^n,$$
$$\mathbf{F}_{ij}^t = -k_t \boldsymbol{\xi}_{ij} - \gamma_t \mathbf{v}_{ij}^t,$$

where k_n and k_t are the normal and tangential stiffness coefficients, γ_n and γ_t are the corresponding damping coefficients, δ_{ij}, the overlap distance, is a measure of the extent of the two particles' surface penetration. For i at position $\mathbf{x}_i$ with radius R_i and j at $\mathbf{x}_j$ with radius R_j, $\delta_{ij} = (R_i + R_j) - ||\mathbf{x_i} - \mathbf{x_j}||$, $\mathbf{n}_{ij}$ is the unit vector along the line joining the centers of i and j, $\mathbf{v}_{ij}^n$ and $\mathbf{v}_{ij}^t$ are the relative normal and tangential velocities, and $\boldsymbol{\xi}_{ij}$ is the tangential displacement accumulated during the contact. In the exact (and continuous) form, at current time $\mathrm{t} = \mathrm{t}_c$, for an impact beginning at $\mathrm{t} = \mathrm{t}_0$, for the accumulated tangential displacement we have:

$$\boldsymbol{\xi}_{ij} = \int_{\mathrm{t}_0}^{\mathrm{t}_c} \mathbf{v}_{ij}^t(\mathrm{t})\mathrm{dt}$$

The discretisation scheme for this integral follows a simple forward Euler sum discussed in Appendix A.1.

The tangential force is capped by a Coulomb friction criterion:

$$|\mathbf{F}_{ij}^t| \leq \mu |\mathbf{F}_{ij}^n|,$$

where μ is the coefficient of friction between i and j, beyond which sliding occurs, and $\mathbf{F}_{ij}^t$ is projected back onto the Coulomb limit as detailed (see Mindlin and Deresiewicz [1953]). The torque on particle i from tangential forces is

$$\boldsymbol{\tau}_i = \sum_{j \neq i} R_i \, \mathbf{n}_{ij} \times \mathbf{F}_{ij}^t,$$

where R_i is the radius of particle i.

1.3 DEM Output Data

These simulations provide us with the evolution of individual atoms' positions, velocities, and their *Virial Stress* tensor (Bagi [1999]) throughout the flow. Briefly, in discrete element analysis, the virial stress tensor measures the stress faced by individual particles that arises from the pairwise sum of the particle's interactions with surrounding particles. For particle i, at $\mathbf{r}_i$ and any j at $\mathbf{r}_j$, we have

$$\mathbf{s}_i^{\text{virial}} = -\frac{1}{2V_i}\left(\sum_{j\neq i}(\mathbf{r}_i - \mathbf{r}_j) \otimes \mathbf{F}_{ij}^{\text{cf}}\right)$$

where $\otimes$ is the standard outerproduct that induces a tensor and $\mathbf{F}_{ij}^{\text{cf}} = (\mathbf{F}_{ij}^{n} + \mathbf{F}_{ij}^{t})$. A snapshot of the format of the `dump` files from `LAMMPS` is provided in Appendix A.2. Note, however, that data on individual particles' physical properties provide us with little insight into the general behaviour of the flow. We undertake the averaging procedure of *coarse-graining* the particulate data to produce data for the physical fields inside the container (Shaebani et al. [2011]).

1.4 Coarse Graining DEM Data

The synergistic application of DEM and coarse-graining (together, and more generally, the DEM-CFD routine) has given rise to a very rich subset of literature. Coarse-graining is a modelling procedure aimed at extracting continuum-level fields from the discretised data generated from DEM simulations. The essence of the technique is to consolidate (or *bin*) a large number of small particles (say N particles) over some volume (or region) into a smaller number of *grains*. With a smoothing kernel φ (typically Gaussian or Lucy), say the center of some bin is z_k with volume V_{bin}, then for any material property field A, the total sum of convolution integrals:

$$A(z_k) = \frac{1}{\mathcal{N}(z_k)}\sum_{i=1}^{N}\int A_i(z')\varphi(z_k - z')\mathrm{d}z'$$

Here $\mathcal{N}(z_k)$ is the normalisation term (dependent on whether the material field is intensive or extensive). In practice, the integral is approximated by discretising the bins into grids and evaluating the total sum across all grid points. Then, for m grid points, for a volume-weighted field,

$$A(z_k) = \frac{1}{V_{\text{bin}}}\sum_{i=1}^{N}\sum_{j=1}^{m}\left(\frac{V_{ij}}{V_i}\right)A_i\varphi(z_k - z_j)$$

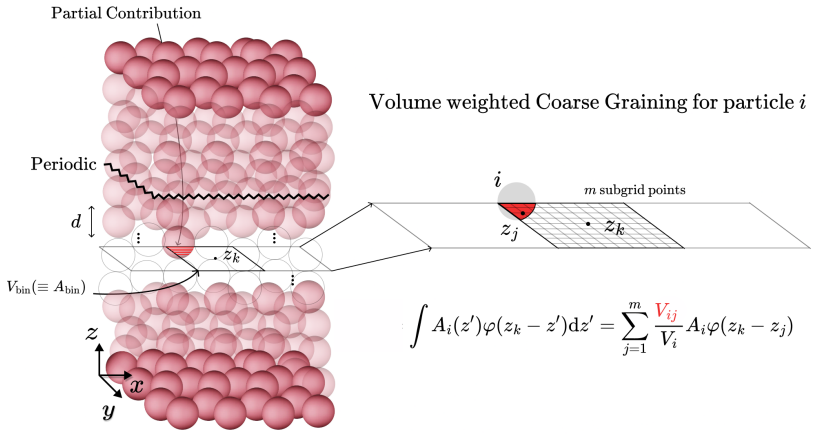
An important volume weighted field is the *Cauchy*-stress tensor $\sigma_{\alpha\beta}(\mathbf{r})$ along the $\alpha, \beta \in \{x, y, z\}$ components of stress, with $\mathbf{r}$ being the position vector for any point. In our coarse-graining procedure, this field is defined as

$$\sigma_{\alpha\beta}(\mathbf{r}_k) = \frac{1}{2V_{\text{bin}}}\left(\sum_{i=1}^{N}\sum_{j=1}^{m}\mathbf{s}_{i,\alpha\beta}^{\text{virial, eff}}\varphi(\mathbf{r}_k - \mathbf{r}_j)\right) = \frac{1}{2V_{\text{bin}}}\left(\sum_{i=1}^{N}\sum_{j=1}^{m}\left(\frac{V_{ij}}{V_i}\right)\mathbf{F}_{ij}^{\text{cf},\alpha}\mathbf{r}_{ij}^{\beta}\varphi(\mathbf{r}_k - \mathbf{r}_j)\right)$$

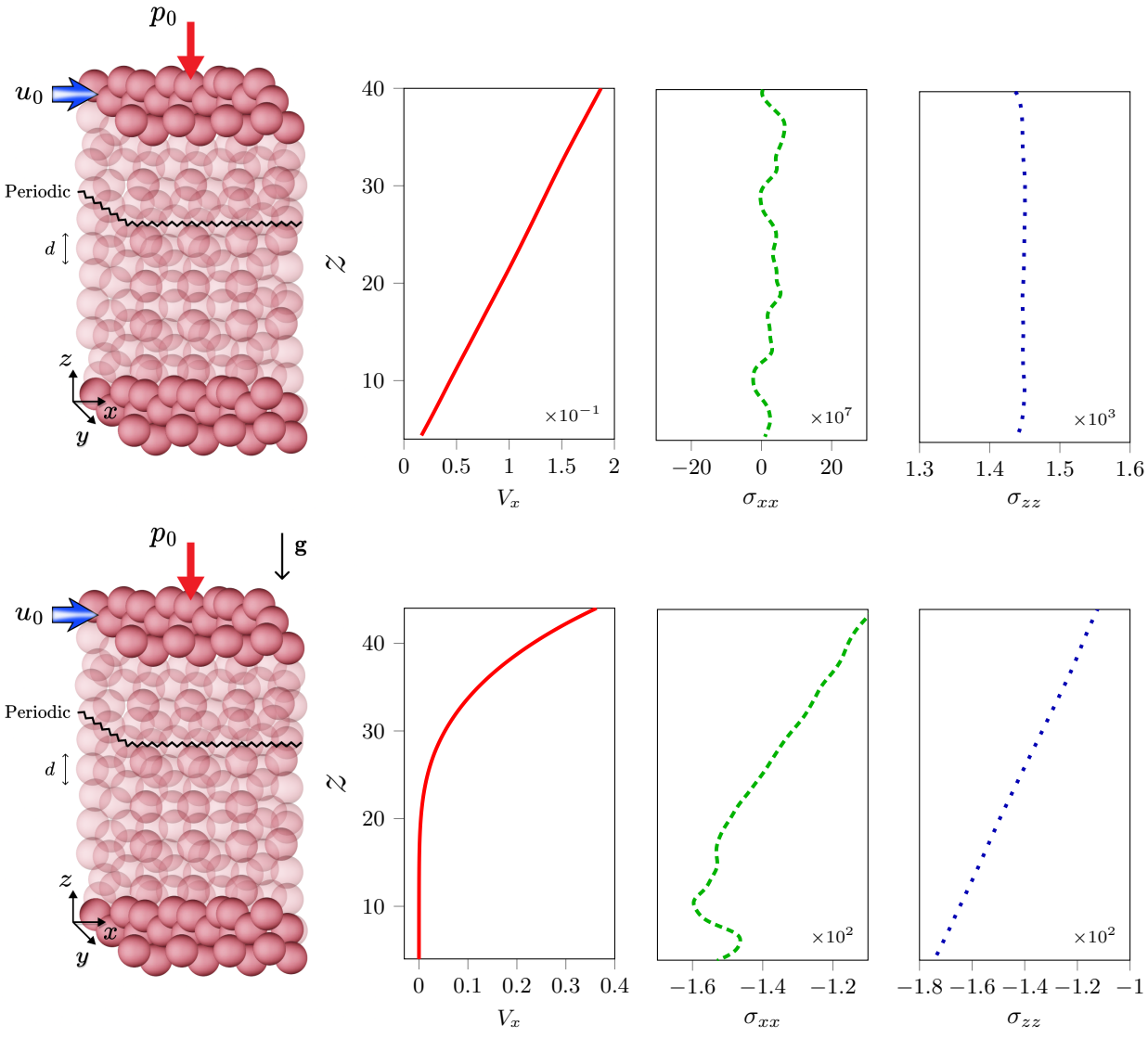
We assume the boundaries along x and y axes to be periodic (see schematic). This enforces statistical homogenization along these axes, so we may resolve to the z axis only Goldhirsch [2010]. Then, the expression for the equivalent stress field becomes

$$\sigma_{\alpha\beta}(z) = \frac{1}{2A_{\text{bin}}}\left(\sum_{i=1}^{N}\sum_{j=1}^{m}\mathbf{F}_{ij}^{\text{cf},\alpha}\mathbf{r}_{ij}^{\beta}\varphi(z_k - z_j)\right)$$

See Appendix A.3. for more details on the exact coarse-graining expressions used for different fields, and also a proof of why the smoothing kernel is required to preserve properties (for example, conservation of mass).


Partial Contribution
Volume weighted Coarse Graining for particle i
Periodic
d
$V_{\text{bin}} (\equiv A_{\text{bin}})$
z_k
i
m subgrid points
z_j
z_k
$\int A_i(z')\varphi(z_k - z')\mathrm{d}z' = \sum_{j=1}^{m} \frac{V_{ij}}{V_i} A_i \varphi(z_k - z_j)$
z
x
y

2 Results


p_0
u_0
Periodic
d
z
x
y
$\mathbf{g}$
z
V_x
σ_{xx}
σ_{zz}
$\times 10^{-1}$
$\times 10^{7}$
$\times 10^{3}$
$\times 10^{2}$

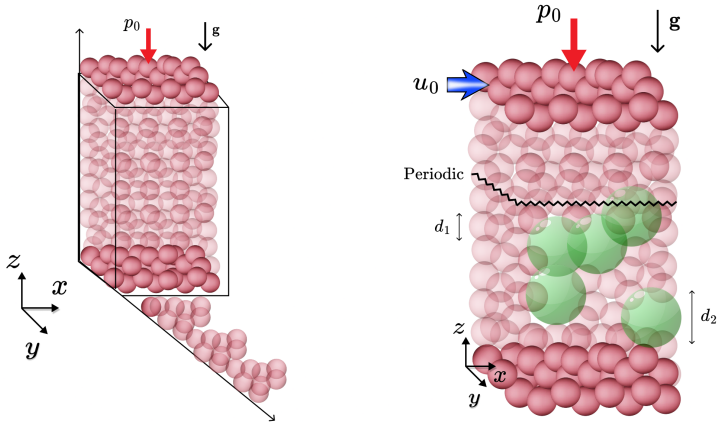
In the first flow, we can see V_x increase linearly as we approach the top (the source of the shear). Additionally, we see that σ_{zz} is nearly constant (in the absence of gravity). Also, σ_{xx} has a very high value relative to σ_{zz} (and along other stress components), this is obviously expected given the shear is applied along the x-axis. For the same reason, in Appendix A.4., we can see that V_y, V_z are effectively zero whereas V_x is orders of magnitude larger throughout the top half of the container. However, as we approach the bottom (lower values of z), V_x approaches zero linearly.

In the second flow, we can see that the reduction in velocity as we go deeper into the container is no longer linear. Beyond a depth of just 10m, it's as if the particles at the bottom have no information about the shear applied on the top. This time, due to the presence of gravity, we see a linear (and increasing in magnitude) σ_{zz} with increasing depth, whereas σ_{xx} shows a jump at the bottom.

3 Future Work

3.1 More Granular Flows

In the coming weeks, we will simulate and coarse-grain two more flows. One is when the particles are simply under the action of gravity on an incline, instead of having a lower boundary. The other is similar to our second flow in the results, except that there are now two different types of particles with varying sizes.



(a) Planar Incline Flow

(b) Bidisperse Shear Flow

3.2 Neural Operators

A neural operator is designed to learn mappings between function spaces ($\mathcal{G} : \mathcal{A} \to \mathcal{B}$), such as those arising in physical systems, where $\mathcal{A}$ and $\mathcal{B}$ are spaces of functions (e.g., velocity fields, stress fields, etc.). Given a dataset of input-output pairs $\{(a_i, b_i)\}_{i=1}^N$ constructed through the coarse graining process, with $a_i \in \mathcal{A}$ and $b_i \in \mathcal{B}$, the neural operator approximates the true physical operator (underlying constitutive model) $\mathcal{G}^*$ via a parameterized model $\mathcal{G}_\theta$:

$$\mathcal{G}_\theta : a \mapsto b$$

The learning objective is to minimise the expected loss between predicted and true output fields:

$$\mathcal{L}(\theta) = \mathbb{E}_{(a,b)\sim\mathcal{D}} \left[\ell \left(\mathcal{G}_\theta(a),\ b \right) \right]$$

where $\ell(\cdot, \cdot)$ is a suitable loss function (e.g., mean squared error), and $\mathcal{D}$ is the data distribution (Li et al. [2021]).

References

Eric Clement. Rheology of granular media. *Current Opinion in Colloid Interface Science*, 4(4):294–299, 1999. ISSN 1359-0294. doi:https://doi.org/10.1016/S1359-0294(99)90004-3. URL https://www.sciencedirect.com/science/article/pii/S1359029499900043.

Peter A Cundall and Otto DL Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1):47–65, 1979.

Yu Guo and Jennifer Sinclair Curtis. Discrete element method simulations for complex granular flows. *Annual Review of Fluid Mechanics*, 47:21–46, January 2015. doi:10.1146/annurev-fluid-010814-014644.

Aidan P. Thompson, H. Metin Aktulga, Richard Berger, Dan S. Bolintineanu, W. Michael Brown, Paul S. Crozier, Pieter J. in 't Veld, Axel Kohlmeyer, Stan G. Moore, Trung Dac Nguyen, Ray Shan, Mark J. Stevens, Julien Tranchida, Christian Trott, and Steven J. Plimpton. Lammps - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271:108171, 2022. ISSN 0010-4655. doi:https://doi.org/10.1016/j.cpc.2021.108171. URL https://www.sciencedirect.com/science/article/pii/S0010465521002836.

Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117 (1):1–19, 1995.

R D Mindlin and H Deresiewicz. Elasticspheres in contact undergoing tangential forces. *Journal of Applied Mechanics*, 20:327–344, 1953.

Katalin Bagi. Microstructural stress tensor of granular assemblies with volume forces. *Journal of Applied Mechanics*, 66(3):934–940, 1999. doi:10.1115/1.2791800.

M Reza Shaebani, Mahyar Madadi, Stefan Luding, and Dietrich E Wolf. Influence of polydispersity on micromechanics of granular materials. *arXiv preprint arXiv:1107.2069*, 2011.

Isaac Goldhirsch. Stress, stress asymmetry and couple stress: from discrete particles to continuous fields. *Granular Matter*, 12(3):239–252, 2010. doi:10.1007/s10035-010-0181-z.

Zongyi Li, Nikola B Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.08895.

Appendix

A.1. Integration Schemes for different physical quantities

The equations of motion for both translation and rotation are integrated using the Verlet algorithm, as implemented in `LAMMPS` via the `fix nve/sphere` command. For each timestep Δt, the algorithm proceeds as follows:

1. Translational and Angular velocity **half step** update:

$$\mathbf{v}_i\left(t+\frac{\Delta t}{2}\right) = \mathbf{v}_i(t) + \frac{\Delta t}{2m_i}\,\mathbf{F}_i^{\mathrm{tot}}(t),$$
$$\boldsymbol{\omega}_i\left(t+\frac{\Delta t}{2}\right) = \boldsymbol{\omega}_i(t) + \frac{\Delta t}{2}\,\mathbf{I}_i^{-1}\boldsymbol{\tau}_i(t),$$

   where $\mathbf{F}_i^{\mathrm{tot}}(t)$ in our case is just contact (and gravity).

2. Position and angular orientation update:

$$\mathbf{x}_i(t+\Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i\left(t+\frac{\Delta t}{2}\right)\Delta t,$$
$$\boldsymbol{\theta}_i(t+\Delta t) = \boldsymbol{\theta}_i(t) + \boldsymbol{\omega}_i\left(t+\frac{\Delta t}{2}\right)\Delta t.$$

   The contact forces $\mathbf{F}_i^{\mathrm{cf}}(t+\Delta t)$ and torques $\boldsymbol{\tau}_i(t+\Delta t)$ at the next time step are calculated using then expressions in the previous section with these new approximate (full-step) positions and (half-step) velocities. Finally,

3. **Full-step** translational and angular velocity update:

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i\left(t+\frac{\Delta t}{2}\right) + \frac{\Delta t}{2m_i}\,\mathbf{F}_i^{\mathrm{tot}}(t+\Delta t),$$
$$\boldsymbol{\omega}_i(t+\Delta t) = \boldsymbol{\omega}_i\left(t+\frac{\Delta t}{2}\right) + \frac{\Delta t}{2}\,\mathbf{I}_i^{-1}\boldsymbol{\tau}_i(t+\Delta t).$$

This explicit, second-order accurate scheme ensures stable and efficient integration of the particle trajectories and rotations, provided the timestep Δt is chosen to be much smaller than the characteristic collision time of the particles. The tangential displacement vector between contacting particles i and j at time $t+\Delta t$ is updated using the forward Euler scheme as follows:

$$\boldsymbol{\xi}_{ij}(t+\Delta t) = \boldsymbol{\xi}_{ij}(t) + \mathbf{v}_{ij}^t\,\Delta t$$

A.2. DEM Data Format

ID	Type	x	y	z	v_x	v_y	v_z	Diam.	Mass	σ_{xx}	σ_{yy}	σ_{zz}	σ_{yz}	σ_{zx}	σ_{xy}
1	2	0.00048	-0.00363	0.00048	0	0	0	0.00096	1.13e-6	-3.87e-7	-1.59e-6	-1.53e-6	-6.14e-8	3.11e-8	…
2	2	0.00141	-0.00391	0.00040	0	0	0	0.00080	6.69e-7	-2.95e-8	-2.34e-7	-8.79e-8	7.60e-8	-4.63e-8	…
3	2	0.00210	-0.00334	0.00046	0	0	0	0.00092	1.00e-6	-1.24e-6	-8.01e-7	-2.16e-6	1.60e-7	-3.89e-7	…
4	2	0.00302	-0.00345	0.00047	0	0	0	0.00094	1.05e-6	-1.69e-7	-6.78e-7	9.00e-9	4.12e-8	-5.78e-9	…
5	2	0.00393	-0.00375	0.00046	0	0	0	0.00093	1.03e-6	-5.56e-7	-5.03e-7	-4.86e-7	3.51e-7	2.10e-7	…
6	2	0.00491	-0.00335	0.00043	0	0	0	0.00086	8.07e-7	-6.34e-7	-1.23e-6	-8.63e-7	1.71e-7	-5.99e-8	…
7	2	0.00612	-0.00358	0.00039	0	0	0	0.00077	5.93e-7	-5.63e-7	-2.83e-7	-1.81e-7	-1.53e-7	3.30e-8	…
10	2	0.00734	-0.00371	0.00036	0	0	0	0.00071	4.68e-7	-3.29e-7	-2.45e-7	1.67e-8	-2.29e-7	-1.53e-8	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

A.3. Coarse Graining for different property fields and Kernel

Smoothing Kernel

Let the coarse-grained mass density field $\rho(\mathbf{r})$ be defined as:

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} m_i \, \varphi(\mathbf{r} - \mathbf{r}_i)$$

where $\varphi(\mathbf{r})$ is a smoothing kernel satisfying:

$$\iiint \varphi(\mathbf{r}) \, \mathrm{d}V = 1$$

Then, the total mass in the domain is given by:

$$M = \iiint \rho(\mathbf{r}) \, \mathrm{d}V = \iiint \left(\sum_{i=1}^{N} m_i \, \varphi(\mathbf{r} - \mathbf{r}_i) \right) \mathrm{d}V$$

Switching the summation and integration:

$$M = \sum_{i=1}^{N} m_i \left(\iiint \varphi(\mathbf{r} - \mathbf{r}_i) \, \mathrm{d}V \right)$$

By a change of variables $\mathbf{r}' = \mathbf{r} - \mathbf{r}_i$, the integral becomes:

$$\iiint \varphi(\mathbf{r} - \mathbf{r}_i) \, \mathrm{d}V = \iiint \varphi(\mathbf{r}') \, \mathrm{d}V = 1$$

Hence:

$$M = \sum_{i=1}^{N} m_i$$

The total coarse-grained mass equals the sum of particle masses. Thus, mass is conserved *iff* the kernel is normalised:

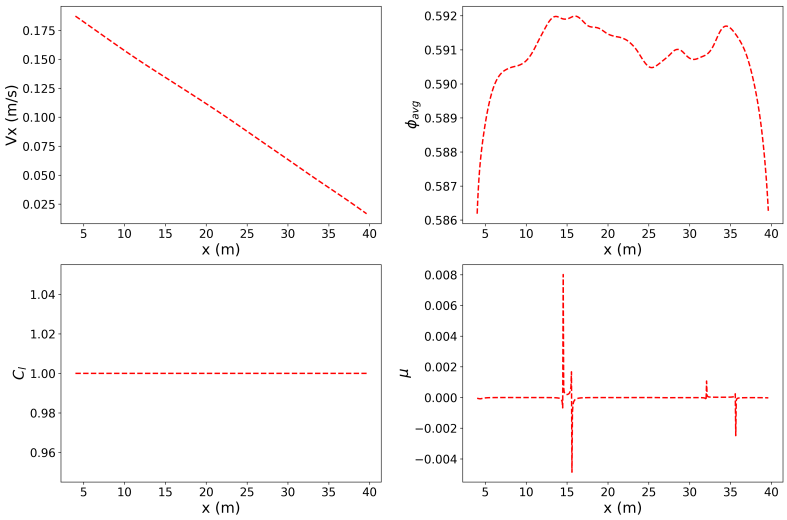
$$\iiint \varphi(\mathbf{r}) \, \mathrm{d}V = 1$$

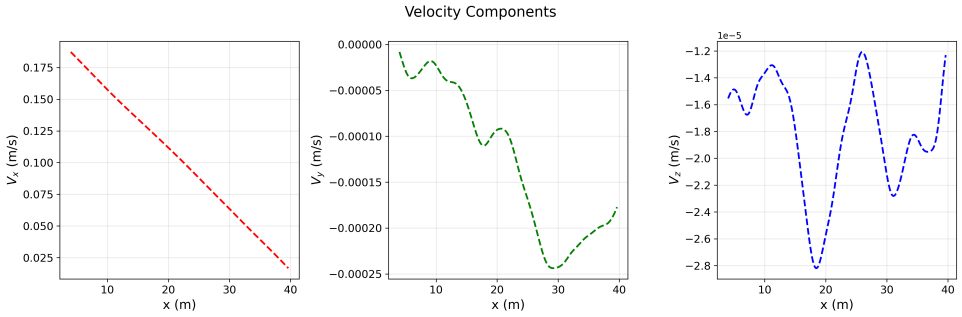
Summary Table

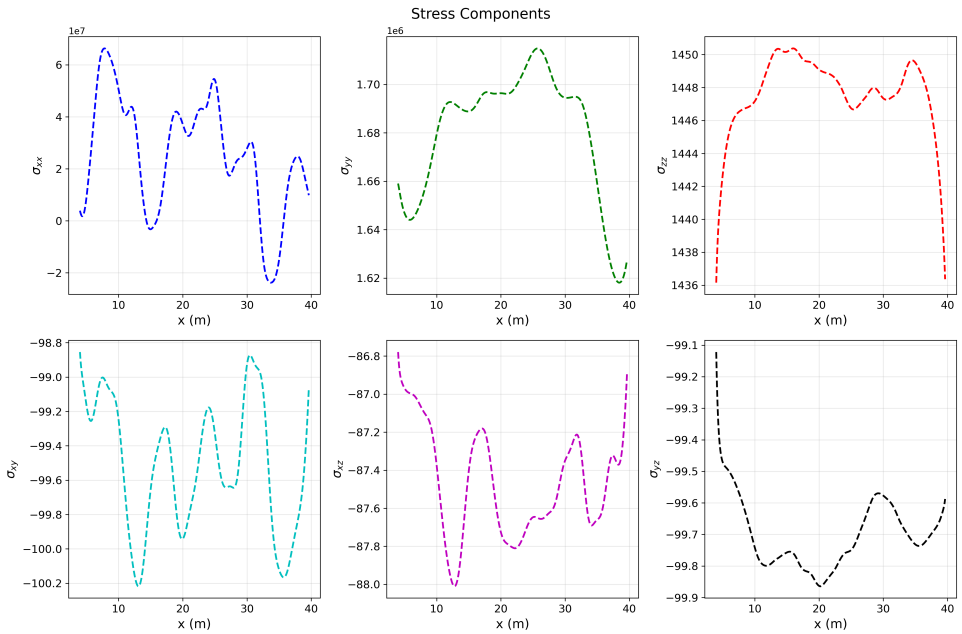
Field	Coarse Graining Procedure
$\phi(z_k)$	$\dfrac{1}{V_{\text{bin}}} \sum_{i,j} V_{ij} \varphi(z_k - z_j)$
$\mathbf{v}(z_k)$	$\dfrac{\sum_{i,j} \left(\frac{V_{ij}}{V_i} \right) \mathbf{v}_i \varphi(z_k - z_j)}{\sum_{i,j} \left(\frac{V_{ij}}{V_i} \right) \varphi(z_k - z_j)}$
$\sigma_{\alpha\beta}(z_k)$	$\dfrac{1}{2V_{\text{bin}}} \sum_{i,j} \left(\dfrac{V_{ij}}{V_i} \right) \mathbf{F}_{ij}^{\text{cf},\alpha} \, \mathbf{r}_{ij}^{\beta} \, \varphi(z_k - z_j)$
$P(z_k)$	$-\dfrac{1}{3} \left(\sigma_{xx}(z_k) + \sigma_{yy}(z_k) + \sigma_{zz}(z_k) \right)$

A.4. Other Plots

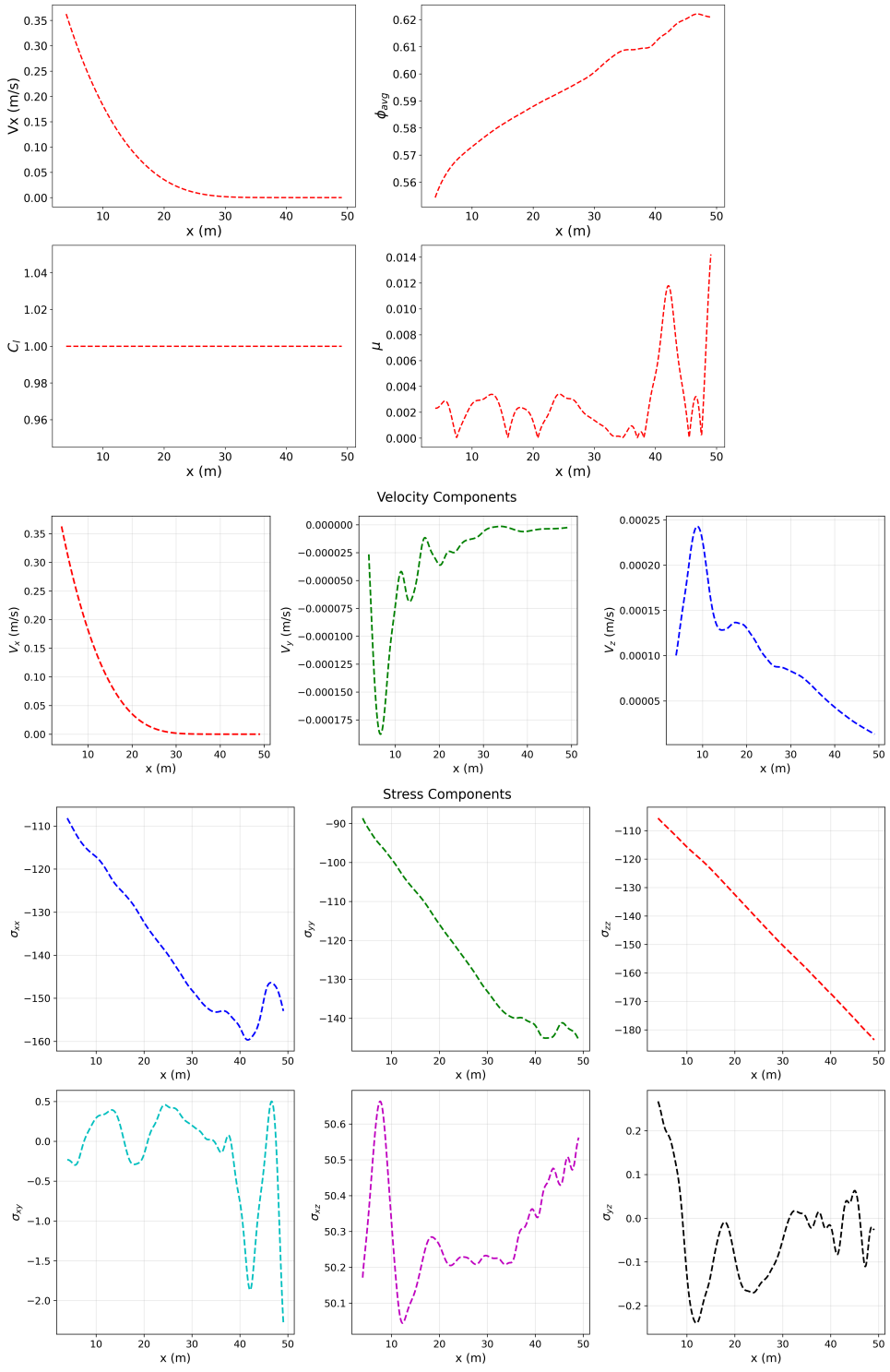
Flow 1







Flow 2



NOTE: In these preliminary plots, "$x(\mathrm{m})$" on the x-axis is really just $40 - z$ (i.e. these have the origin translated to the top of the container).