Lees-Edwards boundary conditions for the multi-sphere discrete element method

Citation for published version:

Digital Object Identifier (DOI):
10.1016/j.powtec.2021.05.025

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Powder Technology

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Lees-Edwards Boundary Conditions for the Multi-Sphere Discrete Element Method

Nathan Berry\textsuperscript{a}, Yonghao Zhang\textsuperscript{b}, Sina Haeri\textsuperscript{a,∗}

\textsuperscript{a}Institute for Materials and Processes, School of Engineering, Sanderson Building, King’s Buildings, Robert Stevenson Road, Edinburgh, EH9 3FB

\textsuperscript{b}Institute for Multiscale Thermofluids, School of Engineering, University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Road, Kings Buildings, Edinburgh, EH9 3FD

Abstract

A consistent implementation of Lees-Edwards boundary conditions is proposed for the Multi-Sphere Discrete Element Method, which can mitigate various unphysical effects at the bulk and micro-structural levels. These effects include non-linear velocity profiles and inhomogeneous particle distributions, which result in significant errors with respect to pressure and granular temperature. In order to allow for a fair assessment of different implementations, a novel compound sphere particle shape is devised for comparison to reliable benchmark data generated from systems of spherical particles. The Multi-Sphere Discrete Element Method is utilised to examine two implementations of these conditions. The commonly used Naive approach results in the aforementioned unphysical effects, which are numerical artefacts causing deviations from the benchmark results of up to one order of magnitude. Meanwhile, the proposed consistent implementation fulfils the fundamental requirements of Lees-Edwards boundary conditions and produces data which are in excellent agreement with the benchmark results, as well as the available literature. Comparing the aforementioned implementations, general principles are developed for implementing Lees-Edwards boundary conditions for the Multi-Sphere Discrete Element Method.

Keywords: Simple shear; Rheology; Constitutive modelling; Non-spherical particles; Particle based methods; Multi-Sphere Discrete Element Method

∗Corresponding author

Email address: s.haeri@ed.ac.uk (Sina Haeri)
1. Introduction

Lees-Edwards boundary conditions (LEBCs), initially developed for molecular dynamics simulations, can induce simple shear flow in a material while utilising periodic boundary conditions [1]. LEBCs also have desirable ability of maintaining homogeneous deformations over arbitrarily large strain scales. From these simple shear flow simulations, the stress response of a given material can be calculated, from which its viscosity can be obtained. As a result of their success, LEBCs have been extended beyond pure molecular dynamic simulations and have been implemented to several particle based numerical techniques [2–4].

LEBCs have also recently been used extensively for fundamental studies of granular materials including jamming transition phenomena [5, 6] and kinetic theory based continuum modelling [7]. In such fundamental studies, LEBCs are advantageous as they eliminate complex boundary effects induced by walls [8] and reduce the computational cost allowing for extensive parametric studies. Furthermore, simple shear flow simulations generated with LEBCs have also been pivotal for significant advances in constitutive modelling of spherical particulate systems [9, 10].

Recent literature indicates that constitutive models developed for spherical particles [11–13], appear amenable for extension to aspherical particles [14–18]. Moreover, it is now well established that particle morphology causes complex microstructural behaviour [19–23] and significantly influences the properties [24–28] of granular materials. Therefore, LEBCs in conjunction with Discrete Element Modelling (DEM) are indispensable for further development of more advanced constitutive models. Additionally, these techniques will also provide fundamental insights into the microstructural behaviour of aspherical particle assemblies, helping address the high demand for improving the efficiency of granular systems. However, the study of granular materials utilising LEBCs have largely been limited to systems of spherical particles, despite the significant progress that has been made in advancing DEM simulations of aspherical particulate systems.

There are now several numerical methods readily available to simulate granular materials consisting of aspherical particles. Many of the methods available utilise
the soft-sphere principle initially devised for spherical particles used in DEM [29].
Such aspherical particle methods include the use of superquadrics [30], ellipsoids [31], polygons and polyhedra [32, 33], basis-splines [34] and spherical harmonics [35].
Several non-exhaustive reviews of such techniques have appeared recently in the literature [36–38]. Analogous methodologies for simulating aspherical particles are also available using the contact-dynamics method [39, 40]. However, arguably the most popular approach for simulating aspherical particles [37], is the Multi-Sphere Discrete Element Method (MS-DEM) [41] which is the focus of this work.

The MS-DEM method, or the glued-sphere discrete element method, fixes individual spherical DEM particles together to define a new rigid particle. This method has three main advantages compared to other techniques for simulating aspherical particles. Firstly, particles of arbitrary shape can be simulated flexibly unlike analytical approaches including ellipsoids, superquadrics and spherical harmonics. Secondly, the algorithm used to detect particle contacts is the simplest of all the aforementioned methods as it is identical to that for spheres. This minimises the computational costs and allows for efficient handling of static friction forces that persist during the inter-particle contact [16]. Thirdly, reliable contact force models that have been developed and validated for spheres can be used with the MS-DEM, unlike those employed for polyhedra and polygons whose accuracy is still open for debate [30]. Despite MS-DEM being one of the most common methods for simulating aspherical particles and the broad applicability of LEBCs, there is currently no literature available on the consistency of LEBCs with MS-DEM and its serious implications on quality of the simulations both at microstructural and macroscopic levels.

To address this gap in the current literature, the implementation of LEBCs for the MS-DEM is thoroughly characterised in this work. Two possible implementation styles are considered and the microstructural and macroscopic properties of systems for different particle shapes, across a wide range of volume fractions and shear rates are systematically analysed. The results are carefully compared to a range of well-defined benchmark problems to document various numerical artefacts resulting from a naive implementation of LEBCs, leading to the development of consistent LEBCs for MS-DEM. The MS-DEM algorithm has a similar structure to that of other DEM
techniques used to model aspherical particles \[42, 43\] as well as other particle-based simulation techniques \[39, 38\]. Therefore, it is believed that this work will have a significant impact on the wider community of users of particle-based methods.

This manuscript is organised as follows. First a brief introduction to the MS-DEM as well as a high level overview of the algorithm that is utilised to perform a MS-DEM timestep, are given in Section 2. Following this, in Section 3.1, a short discussion on the classical implementation of LEBCs for DEM simulations of a system of spherical particles is provided. The two possible implementations of LEBCs for the MS-DEM that are investigated are outlined in Section 3.2 and the proposed numerical experiments to compare the two implementations are detailed in Section 4. The general principles that must be followed to successfully implement LEBCs for the MS-DEM are discussed in Section 5.3. All the algorithms are developed within the open-source framework LAMMPS \[44\].

2. The Multi-Sphere Discrete Element Method

From an algorithmic standpoint, it is instructive to think of an MS-DEM particle as being made up of two separate computational entities. The first of which is called the rigid-body information object (RBIO) and the second, simply, constituent spheres. The RBIO contains all of the rigid-body information used for updating trajectories of the MS-DEM particle, including centre of mass (COM), position, velocity, inertia tensor and quaternions or rotation tensor. Whilst the constituent sphere object, contains all the information about the spheres that make up a given MS-DEM particle such as their positions, velocities and diameters. It should also be noted that constituent spheres are used purely for contact detection, and force calculations. The algorithm used here, as shown in Figure 1, is a slightly modified velocity Verlet algorithm \[45\] consisting of three stages. The first stage of the algorithm is to update the RBIO, which updates the particles COM position by a full timestep, $\Delta t$, as well as the COM velocity by a half timestep as follows:

$$V_\ell \left( t + \frac{1}{2} \Delta t \right) = V_\ell (t) + \frac{F_\ell (t) \Delta t}{2M},$$

(1)
In Eqs. (1) and (2), \( F_{\ell}, V_{\ell} \) and \( X_{\ell} \) are the force, translational velocity and position respectively, of the COM held in the RBIO at a time \( t \). The subscript \( \ell \in \{1, 2, 3\} \) corresponds to the \( \{x, y, z\} \) directions in a Cartesian coordinate system.

The orientation of a given particle can be defined using the directions of the principle moments of inertia of the particle, from which an orthogonal rotation matrix \( Q_{m\ell}(t) \) can be constructed. With the aid of this rotation matrix, the following procedure can be used to update the positions and velocities of the constituent spheres:

\[
\Lambda_\ell = Q_{m\ell}(t) \left( x_m(t) - X_m(t) \right),
\]

\[
x_\ell(t + \Delta t) = Q_{m\ell}(t + \Delta t) \Lambda_m + X_\ell(t + \Delta t),
\]

\[
v_\ell \left( t + \frac{1}{2} \Delta t \right) = \left[ \epsilon_{\ell mn} \Omega_m \left( t + \frac{1}{2} \Delta t \right) \left( x_n(t + \Delta t) - X_n(t + \Delta t) \right) \right] + V_\ell(t + \frac{1}{2} \Delta t).
\]

In Eqs. (3) to (5) a tensor notation is used. Furthermore, \( x_\ell \) and \( v_\ell \) are the vector positions of a given constituent sphere and its corresponding translational velocity. The term \( \Omega_\ell \) is the angular velocity and \( \Lambda_\ell \) is by definition a constant for a rigid body, which can be determined at the beginning of a simulation to save computational time. Within LAMMPS, both \( Q_{m\ell} \) and \( \Omega_\ell \) are updated using quaternions. A discussion of these operations is omitted for brevity as, LEBCs only affect translational degrees of freedom, however, the interested reader may consult [30, 37] for more detail.

The second stage of the algorithm is to apply any boundary conditions, including LEBCs, which are discussed in Section 2. Additionally, contact detection is performed at this stage using the constituent spheres and with the aid of an appropriate contact model the resulting forces acting on each constituent sphere are calculated [46, 49].

Finally, in stage 3, all the forces acting on a given MS-DEM particle are summed...
and the velocity of the COM is updated in the RBIO by a remaining half time step:

\[ V_\ell(t + \Delta t) = V_\ell \left( t + \frac{1}{2} \Delta t \right) + \frac{F_\ell(t + \Delta t) \Delta t}{2 \mathcal{M}}. \]  

(6)

With the additional update of \( \Omega_\ell \) by the remaining half timestep, the constituent spheres have their velocities updated accordingly:

\[ v_\ell(t + \Delta t) = \epsilon_{\ell_{mn}} \Omega_m \left( t + \Delta t \right) \left( x_n(t + \Delta t) - X_n(t + \Delta t) \right) + V_\ell(t + \Delta t). \]  

(7)

3. Lees-Edwards Boundary Conditions

3.1. LEBCs for Spherical Particles

In this section a general LEBC procedure that results in a linear velocity profile is described for spherical particles. It is assumed the 3D computational domain with dimensions \( L_\ell \) in Cartesian coordinates, is filled with mono-dispersed spherical particles characterised by a diameter \( d \) with position, \( x_\ell \), and velocity, \( v_\ell \), vectors given at time \( t \) for each particle \( i \), \( i \in 1 \cdots N_{\text{tot}} \), where \( N_{\text{tot}} \) is the total number of particles in the assembly. Without loss of generality, it is assumed that the velocity profile is given by

\[ u_\ell = \dot{\gamma} y(e_x \cdot e_\ell), \]  

(8)

where \( u_\ell \) is the velocity vector, \( e_x \) is the unit vector in the x-direction, \( \dot{\gamma} \) is the applied shear rate and \( y \) is the position in the y-direction. Furthermore, the subscript min (e.g. \( L_{\ell,\text{min}} \)) means the lower bound of the domain in \( \ell \)-direction and similarly a subscript “max” is used to represent the upper bound.

All particles are subject to classical periodic boundary conditions, fulfilled by remapping particle positions as follows

\[ x_\ell' = \begin{cases} 
    x_\ell - L_\ell, & \text{if } x_\ell > L_{k,\text{max}} \\
    x_\ell + L_\ell, & \text{if } x_\ell < L_{k,\text{min}} \\
    x_\ell, & \text{otherwise,}
\end{cases} \]  

(9)
where, "′", indicates a mapped variable. However, for particles passing through a boundary perpendicular to the direction of the shear flow, LEBCs are simultaneously applied through additional velocity and position remapping. Thus, for shear acting on the \( xy \)-plane the particle velocities and positions are remapped according to

\[
v'_x = \begin{cases} 
  v_x - \dot{\gamma}L_y, & \text{if } x_y > L_{y,\text{max}} \\
  v_x + \dot{\gamma}L_y, & \text{if } x_y < L_{y,\text{min}} \\
  v_x, & \text{otherwise.}
\end{cases}
\]  

(10)

and

\[
x'_x = \begin{cases} 
  x_x - \Delta t\dot{\gamma}L_y, & x_y > L_{y,\text{max}} \\
  x_x + \Delta t\dot{\gamma}L_y, & x_y < L_{y,\text{min}}.
\end{cases}
\]  

(11)

Furthermore, if a particle has partially crossed a boundary it is identified as “close” to a boundary and will require special treatment. For example, a spherical particle with diameter \( d \) is defined as close to a boundary if either

\[
L_{\ell,\text{max}} - x_\ell < \frac{1}{2}d,
\]  

(12)

or

\[
L_{k,\text{min}} + x_\ell < \frac{1}{2}d,
\]  

(13)

holds. For particles close to a boundary, periodic images must be defined to ensure that the particles interact correctly and with the accurate relative velocities.

3.2. Implementation styles of LEBCs for the MS-DEM

In this section two different styles for implementing LEBCs for MS-DEM are discussed. The first style is an intuitive and highly convenient extension of the LEBC implementation for spherical particles as discussed in Section 3.1. Here, LEBCs as described with Eqs. (9) to (13) are applied to the the constituent spheres at stage 2 of the MS-DEM algorithm as is shown in Figure 2. This approach is straightforward since an available implementation of LEBCs for purely spherical particle systems can be
directly used with the MS-DEM algorithm. However, it will be shown in Section 5 that this implementation produces numerical artefacts which severely impact the quality of the results and hence it has been called the “Naive” approach.

To resolve the issues with the Naive approach, a second algorithm is proposed in this paper for the first time, which we refer to as the Consistent approach. This implementation is described in Figure 2 and begins by applying Eqs. (9) to (11) directly to the RBIO, where \( x_\ell \) and \( v_\ell \) are replaced with their RBIO equivalents \( X_\ell \) and \( V_\ell \). Following this, the Constituent spheres are mapped onto the new RBIO positions and velocities as described with Eqs. (3) to (5). Finally, the constituent spheres are remapped with Eqs. (9) to (13). This remapping of the constituent spheres is used to incorporate the effect of having an MS-DEM particle close to the boundary. This is analogous to use of the conditions in Eqs. (12) and (13) for determining the application of periodic domain images for spherical DEM particles as is explained in Section 3.1.

4. Simulation Setup and Parameters

4.1. Particle Shapes

To compare the Naive and Consistent implementations, a series of simulations are carried out that utilise four different particle shapes shown in Figure 3. The first shape presented in Figure 3a is a normal spherical particle which is used for benchmarking.

The second shape presented in Figure 3b is a compound sphere made up of a small massless spherical particle embedded within a larger one. Although this shape requires utilisation of the MS-DEM, the small embedded spheres cannot interact with any other particles and should have no impact on the physics obtained from the system. Therefore, it is expected that results generated with compound spheres using a successful implementation of LEBCs for the MS-DEM, would be identical to the produced benchmark results for spherical particles (Figure 3a) as well as the independent earlier works by Chialvo and Sundaresan [7], Chialvo et al. [10].

The third shape considered here is a simple dimer or dumbbell particle, formed by fixing two identical spheres together with no overlap, see Figure 3c. This elongated
particle shape was chosen so that qualitative comparisons can be made to data provided in the available literature Nath and Heussinger [14], Nagy et al. [15], Salerno et al. [16], Nagy et al. [18].

The final shape under consideration, is that of an ellipsoid or prolate spheroid. To construct this particle, the algorithm for filling an ellipse with spheres as proposed by Markauskas et al. [50] is utilised. All ellipsoids have an aspect ratio of $\alpha = 1.5$ and are approximated with 5 overlapping spheres of varying diameter as shown in Figure 3d. The aspect ratio is defined as $\alpha = r_{max}/r_{min}$, where $r_{max}$ and $r_{min}$ are the lengths of semi-major and semi-minor axes of the ellipsoid respectively. This shape with $\alpha = 1.5$ was also chosen so that quantitative comparisons can be made to Campbell’s data [17], which are obtained by using analytically defined ellipsoidal particles in their simulations with the same aspect ratio, as opposed to the MS-DEM approach in this work.

All particles utilising the MS-DEM algorithm, were constructed using the fix rigid command/package in LAMMPS.

4.2. Force models and material parameters

All simulations use a simple linear spring-dashpot model where the normal and tangential forces acting on particle $i$ as a result of contact with particle $j$ are described by

$$F_{n,i,j} = k_n \delta_{ij} n_{\ell,i,j} - \gamma_n m_{eff} v_{\ell,i,j}^n,$$

and

$$F_{t,i,j} = -k_t \Delta S_{\ell,-} - \gamma_t m_{eff} v_{\ell,i,j}^t,$$

respectively. In Eqs. (14) and (15) a tensor notation is used. Moreover, $\delta_{ij}$ is the overlap between particles $i$ and $j$; $n_{\ell,i,j}$ is the unit normal vector in the direction of the line connecting the centres of particle $i$ to $j$; and $k_n$ and $k_t$ are spring stiffness constants acting in the normal and tangential directions respectively. The normal and tangential damping constants are given by $\gamma_n$ and $\gamma_t$ respectively. Furthermore,
$m_{\text{eff}}$ is the reduced mass of the interacting particles which is defined as $m_{\text{eff}} = m_i m_j / (m_i + m_j)$. The normal and tangential relative velocities are denoted by $v_{n,ij}^\ell$ and $v_{t,ij}^\ell$ and $\Delta S_\ell$ is the elastic shear displacement. A Coulomb friction coefficient $\mu$ is utilised to model a stick and slip behaviour, by requiring an effective tangential force $F_{t,ij}^\ell = \min\{|F_{n,ij}^\ell|, \mu|F_{n,ij}^\ell|\} t_\ell$, where $t_\ell = F_{t,ij}^\ell / \|F_{t,ij}^\ell\|$. The coefficient of restitution, $e$, resulting from the linear spring-dashpot model can be calculated as $e = \exp(-\gamma_n \pi / \sqrt{4k_n / m_{\text{eff}} - \gamma_n^2})$.

All assemblies were composed of mono-dispersed and mono-shaped particles at six different volume fractions $\phi = [0.1, 0.25, 0.35, 0.45, 0.55, 0.6]$, spanning dilute to dense systems. For each simulation the volume fraction is fixed. Then the system is subjected to five different shear rates, $\dot{\gamma}$, which are scaled using the following equation:

$$\dot{\gamma}^* = \dot{\gamma} d / \sqrt{k_n / \rho d}, \quad (16)$$

where $\rho$ is the particle density. For cases with aspherical particles, the same scaling forms are used with $d$ replaced by an equivalent diameter, which is defined as the diameter of a sphere that has the same volume as the given particle. All parameters are chosen to yield identical scaled shear rates. The contact model parameters are also selected such that the coefficient of restitution of a collision between two particles is 0.9 and the tangential stiffness coefficient is set as $k_t = 2k_n / 7$. The compound spheres and dumbbell simulations contain 2000 particles, whilst only 1000 ellipsoidal particles are used – identical to the system considered in literature[17]. Each simulation domain is prepared by giving each particle an initial random velocity, followed by an isotropic compression to the desired volume fraction.

The full simulation matrix is summarised in Table 1 which is designed to investigate the effectiveness of the proposed LEBC algorithm for MS-DEM for all granular regimes comprising quasi-static, inertia and intermediate [10].

A snapshot of the simulations for three different particle shapes are presented in Figure 4 using both Naive and Consistent implementations. The microstructural difference between the two implementation styles are clearly visible in this figure, in terms of both of particle number density distributions and the average particle velocity.
Table 1: Table containing the simulation parameters used for this work.

<table>
<thead>
<tr>
<th>φ</th>
<th>( \dot{\gamma}^* )</th>
<th>Shape and Particle number</th>
<th>( e )</th>
<th>( \mu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>( 10^{-1} )</td>
<td>sphere : 2000</td>
<td>0.9</td>
<td>0.5</td>
</tr>
<tr>
<td>0.55</td>
<td>( 10^{-3/2} )</td>
<td>compound sphere: 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.45</td>
<td>( 10^{-2} )</td>
<td>dumbell: 2000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>( 10^{-5/2} )</td>
<td>ellipse: 1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total number of simulations: 168</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

profiles along the y-axis. These will be discussed in detail in Section 5 alongside the associated bulk properties showing the superior performance of the Consistent implementation.

4.3. Data acquisition protocols

All of the data presented is ensemble averaged over 2000 snapshots of the simulation. These snapshots are taken at equidistant points selected between the strain values, \( 50 < \dot{\gamma}t < 150 \). Representative curves for \( T^* \) – as defined by Eq. (22) – versus \( \dot{\gamma}t \) are provided Figure 5 and the range of \( \dot{\gamma}t \) values over which the data is acquired is highlighted in this figure. By investigating the history data similar to Figure 5, all the simulations were found to have reached a stationary state for \( \dot{\gamma}t > 50 \) and were independent of the initial configuration confirming the validity of the averaging procedure.

In this work, the macroscopic quantities considered are the stress and granular temperature calculated for all the simulations. The stress tensor, \( \sigma_{\ell m} \), is given by

\[
\sigma_{\ell m} = \frac{1}{V} \left\langle \sum_i^{N_{\text{tot}}} \left[ \sum_{j \neq i}^{N_{\text{tot}}-1} \frac{1}{2} r_{\ell,ij} F_{m,ij} + m_i (v'_{m,i})(v'_{m,i}) \right] \right\rangle , \tag{17}
\]

where \( V \) is the computational domain volume; \( r_{\ell,ij} \) is the vector connecting the centre of mass of particle \( i \) to that of particle \( j \); \( m_i \) is the mass of particle \( i \); and \( v'_{\ell,i} \),
is the fluctuating velocity of particle $i$ defined as:

$$ v'_{\ell,i} = v_{\ell,i} - \hat{v}_{x,\text{bin}} (e_x \cdot e_\ell), $$

(18)

where $\hat{v}_{x,\text{bin}}$ is an instantaneous spatial average which will be discussed later.

The scalar pressure, $P$, is defined as $\frac{1}{3} \text{tr}(\sigma_{\ell m})$ and the granular temperature, $T$, is calculated as given by [51]:

$$ T = \frac{1}{3N_{tot}} \left\langle \sum_{i=1}^{N_{tot}} v'_{\ell,i} v'_{\ell,i} \right\rangle. $$

(19)

Furthermore, in this paper, the pressure and temperature are presented in non-dimensional form defined as follows:

$$ P^* = \frac{P_d}{k_n}, $$

(20)

$$ P_2^* = \frac{P}{\rho(\dot{\gamma} d)^2}, $$

(21)

$$ T^* = \frac{T}{(\dot{\gamma} d)^2}. $$

(22)

Particle velocity and number density profiles are used to quantify the quality of the proposed LEBCs algorithms in relation to the expected microstructures of the system. Both of these are found by assigning particles to one of ten equidistant spatial bins defined along the y-axis. The average velocity of the particles in each spatial bin, is calculated according to

$$ \hat{v}_{x,\text{bin}} = \frac{1}{N_{p,\text{bin}}} \sum_{i=1}^{N_{p,\text{bin}}} v_{x,i}, $$

(23)

where, $N_{p,\text{bin}}$ is the instantaneous number of particles in that bin. Here, only the velocity component parallel to the flow (x-direction) is of interest. Furthermore, to calculate the velocity, $\dot{v}_x$, and number density, $N_p$, profiles in section Section 5.1, $\hat{v}_{x,\text{bin}}$ and $N_{p,\text{bin}}$ are ensemble averaged similar to Eqs. (17) and (19).
5. Results and Discussion

5.1. Microstructural Properties

Quantifying the quality of the velocity and particle number density profiles produced from LEBCs is essential in order to verify that the conditions have been implemented properly. By definition [1], LEBC should result in linear velocities described with Eq. (8), as well as produce homogenous number density profiles. Thus, for the velocity profiles, any error resulting from a given implementation can be quantified with respect to Eq. (8). Here, the final percentage error, $\epsilon_v$, resulting from a particular set of simulation parameters is defined by,

$$\epsilon_v = \frac{1}{10} \sum_{bin=1}^{10} \frac{\left| \hat{v}_{x,bin} - \hat{y}_{ybin} \right|}{\hat{y}_{ybin}} \times 100,$$

(24)

where, $y_{bin}$ is the central position of a spatial bin. Furthermore, the percentage error resulting from a given system deviating from a homogeneous number density profile is quantified using:

$$\epsilon_n = \frac{1}{10} \sum_{bin=1}^{10} \frac{\left| N_{tot} - 10 N_{p,bin} \right|}{N_{tot}} \times 100,$$

(25)

where, $N_{p,bin}$ is the number density in a given spatial bin.

In order to concisely discuss the effect of different parameters on $\epsilon_n$ or $\epsilon_v$, further averaging on these error values is performed and explained here for the case of $\epsilon_n$ (an identical notation is used for $\epsilon_v$). The symbol $\bar{\epsilon}_n$ represents the average value of $\epsilon_n$ taken across all values of $\dot{\gamma}^*$ for a given particle shape and LEBC implementation. Meanwhile for values taken across $\phi$, $\tilde{\epsilon}_n$ is used. The symbol, $\hat{\epsilon}_n$ represents the average value of $\epsilon_n$ taken across all values of $\dot{\gamma}^*$ and $\phi$ for a given particle shape and LEBC implementation. Finally $\check{\epsilon}_n$ is used to average across $\dot{\gamma}^*$, $\phi$ and particle shape for a given implementation.

For comparison, dropping the subscript $bin$, the velocity profiles are non-dimensionalised using $(\dot{\gamma} L_y)$ as the reference velocity: $v^*_x = \hat{v}_x/(\dot{\gamma} L_y)$. Similarly, the number of particles in each bin is normalised as $N^*_p = N_p/(N_{tot}/10)$, such that a homogenous particle
distribution will result in $N_p^* = 1$. The position perpendicular to shear flow in a given domain is also non-dimensionalised as $y^* = 2y/L_y$.

5.1.1. Compound Sphere

Figure 6a shows the errors in the number density profiles for the Naive and Consistent approaches. The Naive implementation generates large discrepancies in terms of number density profile, with an average $\epsilon_n$ across all values of $\phi$ and $\dot{\gamma}^*$ of $\bar{\epsilon}_n = 12.67\%$. Meanwhile, for the consistent approach, $\epsilon_n = 0.43\%$ showing a relative improvement of 96%. For the benchmark results generated using spherical particles and traditional LEBCs, $\epsilon_n = 0.72\%$ which is 11.95% lower than the errors observed for the Naive approach but is compatible with the value observed for the Consistent implementations further indicating the expected equivalence between the two cases.

The Consistent implementation, shows a slight improvement in $\epsilon_n$ for $\phi = 0.6$ where $\bar{\epsilon}_n = 0.22\%$, but otherwise appears independent of $\dot{\gamma}^*$ and $\phi$. However, for the Naive implementation, $\epsilon_n$ demonstrates a strong dependence on both $\phi$ and $\dot{\gamma}^*$. An interesting feature of these dependencies is that the value of $\bar{\epsilon}_n$ at $\dot{\gamma}^* = 10^{-1}$ remains small at 0.28%, but for $\dot{\gamma}^* < 10^{-1}$, the errors remain small only for the minimum and maximum $\phi - \bar{\epsilon}_n = 0.88\%$ and $\bar{\epsilon}_n = 4.9\%$ respectively – whereas sharp increases in the values of $\epsilon_n$ at $0.15 < \phi < 0.45$ are observed with errors reaching as high as $\epsilon_n = 35\%$. The cause of these high $\epsilon_n$ values, as observable in Figures 4b and 6c, is a clustering of particles towards the centre of the domain which is clearly a numerical artefact.

For the Naive implementation all velocity profiles, as shown in Figure 6b, are highly erroneous with respect to Eq. (8), with $\dot{\epsilon}_v = 82.04\%$. Conversely, the Consistent approach generates profiles in excellent agreement with Eq. (8), with an average error across all volume fractions and applied shear rates of $\dot{\epsilon}_v = 8.31\%$. Thus, utilising the consistent approach improves the observed error by over 73%.

Similar results are obtained with comparison to the benchmark data. For the Naive implementation, the average error across all volume fractions and shear rates, $\dot{\epsilon}_v$, is 73.82% higher than the value obtained from the benchmark simulations, $\dot{\epsilon}_v = 8.22\%$. Whilst the (absolute) overall difference in $\dot{\epsilon}_v$ between Consistent implementation and
benchmark simulations is below 0.1%. This comparison further confirms the equivalence between the spherical benchmark results produced with the traditional LEBCs and the results produced with MS-DEM technique in conjunction with the newly developed Consistent LEBCs.

The velocity profiles for the Consistent approach show no clear dependence on $\phi$ and $\dot{\gamma}^*$. Whilst results from the Naive implementation appear to show a weak dependence on $\phi$ and $\dot{\gamma}^*$, with error generally decreasing as $\phi$ increases and increasing as $\dot{\gamma}^*$ increases. The high values of $\epsilon_v$ obtained from the Naive implementation are physically manifested as non-linear velocity profiles as shown in Figures 4b and 6d. It should also be noted that the velocity profiles are not just non-linear but are, in several cases such as at $\phi = 0.15$ and $y^* = -1$, an order of magnitude below their ideal value given by Eq. (8). Meanwhile the velocity profiles obtained for the Consistent implementation produce linear profiles, with negligible deviation from their prescribed value.

5.1.2. Dumbell

As with the compound spheres, the Naive approach, as shown in Figure 7a, produces highly erroneous number density profiles resulting in $\hat{\epsilon}_n = 29.17\%$. In comparison, the consistent implementation provides a relative decrease in error of 97.8\%, with $\hat{\epsilon}_n = 0.64\%$. For the consistent implementation, no dependence on $\dot{\gamma}^*$ is observed although as $\phi$ increases $\epsilon_n$ decreases. For the Naive implementation, with the exception of $\dot{\gamma}^* = 10^{-1}$ which shows a maximum value at $\phi = 0.35$, $\bar{\epsilon}_n$ decreases significantly with respect to $\phi$, from 43.93\% at $\phi = 0.15$ to 6.64\% at $\phi = 0.6$. The large $\epsilon_n$ values resulting from the Naive implementation are caused by significant particle clustering in the centre of the computational domain, as evinced by Figures 4d and 7c.

Additionally, the Naive approach produces highly erroneous velocity profiles as shown in Figure 7b, with the averaged $\hat{\epsilon}_v = 84.49\%$. Meanwhile, the Consistent implementation provides relative decrease in error of 89\%, with $\hat{\epsilon}_v = 9.29\%$. For the velocity profiles produced from the Consistent implementation, no clear dependence on $\dot{\gamma}^*$ is observed, however, $\epsilon_v$ generally increases as $\phi$ increases, from an average $\bar{\epsilon}_v = 7.36\%$ at $\phi = 0.15$, to $\bar{\epsilon}_v = 13.79\%$ at $\phi = 0.6$. The velocity profiles obtained
from the Naive implementation, show significantly higher values of \( \epsilon_v \) at \( \dot{\gamma}^* = 10^{-1} \) and in general error increases as \( \phi \) increases. As with the compound spheres, the high \( \epsilon_v \) values are a consequence of highly non-linear velocity profiles, a numerical artefact, that deviate significantly from Eq. (8), see Figures 4d and 7d.

5.1.3. Ellipse

As shown in Figure 8a, the number density profiles obtained with the Naive implementation are also erroneous with \( \bar{\epsilon}_n = 22.71\% \). The Consistent implementation, again, significantly improves these results, with a relative decrease in error of 97.16\%, resulting in an \( \bar{\epsilon}_n \) value of 0.65\%. The values of \( \epsilon_n \) obtained from the Consistent implementation show no clear dependence on \( \dot{\gamma}^* \), although an overall decreasing trend with increasing \( \phi \) is observed from \( \epsilon_n = 1.29\% \) at \( \phi = 0.15 \) to \( \epsilon_n = 0.2\% \) at \( \phi = 0.6 \). For the Naive implementation, the dependence of \( \epsilon_n \) on \( \phi \) is similar to that of the compound sphere, with the minimum \( \bar{\epsilon}_n \) values occurring at the minimum and maximum values of \( \phi \), with \( \bar{\epsilon} = 8.91\% \) at \( \phi = 0.15 \) and \( \bar{\epsilon}_n = 8.93\% \) at \( \phi = 0.60 \). No clear dependence on \( \dot{\gamma}^* \) is observed for the Naive implementation with respect to \( \epsilon_n \). As with the other particle shapes, the large \( \epsilon_n \) values resulting from the Naive implementation are caused by significant particle clustering in the centre of the computational domain, as evinced by Figures 4f and 8c.

The velocity profile errors, provided in Figure 8b, show that, as with the other particle shapes, the velocity profiles produced by the Naive implementation are highly erroneous, \( \bar{\epsilon}_v = 72.12\% \). This error is decreased significantly by the Consistent implementation, which results in \( \bar{\epsilon}_v = 8.38\% \). With respect to \( \epsilon_v \), the Consistent implementation shows no clear dependence on \( \dot{\gamma}^* \). Also, with a maximum value of \( \epsilon_v = 11.36\% \) at \( \phi = 0.6 \), \( \epsilon_v \) generally increases as \( \phi \) increases. For the Naive implementation, the maximum \( \epsilon_v \) values are observed at \( \dot{\gamma}^* = 10^{-1} \). For all values of \( \dot{\gamma}^* \), the maximum \( \epsilon_n \) values occur at \( \phi = 0.15 \). As with the other shapes, the high \( \epsilon_v \) values are a result of highly non-linear velocity profiles, see Figures 4f and 8d.
5.1.4. General discussion and summary

For all particle shapes, the Consistent LEBC implementation is highly successful in minimizing numerical errors and producing velocity and density profiles in agreement with the expected ideal values. As can be seen in Figures 6d, 7d and 8d, the velocity profiles are linear, with an average deviation, across all shapes, from the idealised velocity profiles of only \( \hat{\epsilon}_v = 8.66\% \). The number density profiles, Figures 6c, 7c and 8c, obtained from the Consistent implementation are also highly successful for all shapes, with an average deviation from homogeneity of just \( \hat{\epsilon}_n = 0.57\% \). These small deviations are contrasted by the large errors, \( \hat{\epsilon}_n = 79.55\% \) and \( \hat{\epsilon}_v = 21.34\% \), generated using the Naive implementation.

Across all shapes, as well as for both implementations, the effect of \( \dot{\gamma}^* \) on the velocity and number density profiles is typically weak. However, for the Naive implementation, \( \epsilon_v \) is generally higher at \( \dot{\gamma}^* = 10^{-1} \). With the exclusion of Naive implementation for the compound sphere at high \( \dot{\gamma}^* \), for all cases it is found that in dense packings, \( \epsilon_n \) is typically lower than the overall average. This is not surprising, given that as \( \phi \to 1 \), \( \epsilon_n \to 0 \). However this result is important to highlight, as dense systems understudy may, in effect, hide the fact that LEBCs have been incorrectly implemented.

These results show that the Consistent implementation successfully applies LEBCs for the MS-DEM. However, it remains pertinent to understand the influence that these two implementation styles have on the macroscopic properties obtained from the system.

5.2. Macroscopic Properties

5.2.1. Compound Sphere

Figure 9a shows the \( P^* \) values for dense granular assemblies utilising the Naive implementation, where significant deviation from the independent data obtained by Chialvo et al. \[10\] is observed with an average absolute difference of 190.46\%. The maximum pressure deviation resulting from the Naive implementation is observed at \( \dot{\gamma}^* = 10^{-1} \) where the pressure is overestimated by 525.01\% and 740.27\% at \( \phi = 0.55 \) and \( \phi = 0.6 \) respectively. The Naive implementation produces similarly erroneous
results with respect to the benchmark data with an overall average pressure deviation of 184.13%. Meanwhile, as shown in Figure 9a, the Consistent implementation is in excellent agreement with independent work and with spherical benchmark data, with an average pressure deviation of 5.20% and 0.58% respectively. Despite the large quantitative differences, qualitatively the pressure response of the dense systems from both implementations are similar.

The pressure deviation accrued from the Naive implementation is more significant for dilute systems with respect to the spherical benchmark data, as can be seen in Figure 9b, with an average pressure deviation from spherical benchmark data of 1335.6%. A simple linear interpolation of the Chialvo and Sundaresan [7] data presented in Figure 9c is used for direct comparisons at each $\phi$ value. For example, the $P^*_2$ value at $\phi = 0.15$, is approximated as the average of the values obtained at $\phi = 0.1$ and $\phi = 0.2$. With comparison to these interpolated data points, the Naive implementation shows an average pressure deviation of 1840.6%. This extreme deviation is similar to that obtained when comparing the Naive implementation with respect to the spherical benchmark data shown in Figure 9c with an average increase in pressure of 1267.9%.

In comparison, the Consistent implementation produces excellent results, with negligible deviations of 0.35% and 0.37% with respect to the spherical benchmark data shown in Figures 9b and 9c respectively. Similarly, the comparison of the results obtained from the Consistent implementation to the interpolated data shown in Figure 9c is significantly improved, with an average difference in pressure across volume fractions of 16.54%. It should be noted that additional error is naturally introduced here due to the required interpolations. Unlike the dense systems, the dilute systems obtained from each implementation show qualitatively very different behaviour. Interestingly, the Naive implementation, does not even generate a qualitatively correct behaviour. The pressure as shown in Figure 9c initially decreases with respect to $\phi$ before increasing again beyond $\phi = 0.35$ whereas, over the given range of $\phi$, a strictly increasing behaviour is expected [52, 54].

The temperatures obtained using the Naive implementation are presented in Figure 9d, which show a significant overestimation of 1497% compared to the interpolated
independent data [7]. A significant increase in comparison to the spherical benchmark results are also observed with an average increase of 2412.3%. As with the pressures, the temperatures obtained from the Consistent implementation are in excellent agreement with the spherical benchmark results, with an average difference of just 0.46% and a relatively good agreement with the interpolated independent data with an average deviation of 21.3%.

5.2.2. Dumbell

As can be seen in Figure 10a, the pressures obtained from the dense systems, \( \phi > 0.45 \), using the Naive implementation have an average difference of 43.17% with respect to the Consistent implementation. At high \( \dot{\gamma}^* \), the difference in pressure between the two implementations is small, but gets significant as \( \dot{\gamma}^* \) decreases. Qualitatively, both implementations result in similar behaviour with pressure increasing with \( \phi \).

For the dilute systems, \( \phi < 0.55 \), a larger difference in pressure is observed from the two implementations, with the Naive implementation producing results 495.46% higher than their equivalents from the Consistent implementation. Furthermore, the qualitative behaviour is significantly different between the two implementations. The Naive implementation shows no clear dependence on volume fraction, such unusual dependences on volume fraction have not been observed in the current literature [14–16, 18]. Whilst the Consistent implementation clearly shows that as \( \phi \) increases, \( P^* \) increases as would be expected.

As with the compound spheres, Section 5.2.1, the temperatures obtained from the Naive approach are significantly higher than those obtained with the Consistent implementation, with an average increase in temperature of 1289%.

5.2.3. Ellipse

As shown in Figure 11a for dense systems, \( \phi > 0.45 \), the pressure resulting from the Naive implementation deviates from independent work by Campbell [17], in particular for \( \phi = 0.6 \), with the overall average difference in pressure of 76.15%. An improvement with respect to the independent work is observed from the Consistent
implementation with an average difference in pressure of 45.89%. Qualitatively, both the implementation styles result in pressures behaving similarly to the literature [17].

For dilute systems, $\phi < 0.55$, Figure 11a, the deviation from independent work incurred from the Naive implementation is significantly higher than for the dense systems, with an average increase in pressure of 504.82%. In contrast, the Consistent implementation produces more accurate results than for dense systems with respect to the independent data [17], with an average pressure difference of 19.26%. As with the other particle shapes, discussed in Sections 5.2.1 and 5.2.2, the pressure results obtained from the Naive implementation show little dependence on $\phi$. The Consistent approach does, however, show an expected dependence on $\phi$, with increasing $\phi$ resulting in an increase in $P^*$. The temperature results in Figure 11b, show that the Naive implementation, as with the other shapes Sections 5.2.1 and 5.2.2, are significantly higher than those obtained from the Consistent approach, with an increase of 889.69%.

5.2.4. General discussion and summary

For all particle shapes, several commonalities exist between all of the pressure and temperature data obtained from our simulations. All pressures obtained from dense systems utilising the Naive implementation are erroneous, with overall average pressure differences of more than 75% observed when compared to the literature and benchmark data. For all particle shapes, the Consistent approach is typically in excellent qualitative and quantitative agreement with benchmark data and the literature. Some deviation does occur for the ellipsoid particles at high $\dot{\gamma}^*$ for the consistent implementation. The reason for this relatively large error obtained from the Consistent approach for the ellipsoid particles, is due to differences in contact number and force calculations as a result of the MS-DEM, as an actual (analytical) ellipse was used in Campbell [17]. These slight differences were to be expected, as they have been previously reported [55, 57]. However, it is clear that the Consistent implementation generates significantly better results than what is obtained from the Naive implementation.

Despite having highly erroneous number density and velocity profiles, resulting in
erroneous pressures, reasonable qualitative agreement in pressure produced from the Naive implementation in dense systems is obtained. This is likely due to the fact that when within or approaching the quasistatic flow regime, the pressure is independent of \( \dot{\gamma} \). Therefore, for dense systems the effect of producing erroneous velocity profiles is minimised. As such, when implementing LEBCs, it is recommended to test dilute systems as macroscopic errors will be more clear.

For the compound sphere and ellipsoidal particle shapes, utilising the Naive implementation in dilute systems, \( \phi < 0.55 \), results in pressures significantly higher, by several hundred percent, than would be expected from the literature [7, 17] as well as benchmark data. These high pressures are also accompanied with either no correlation or in some cases an unusual correlation with respect to \( \phi \). This qualitative behaviour is inconsistent with both the behaviour observed in the literature [7, 17], as well as the benchmark simulations. Despite the fact that no fair quantitative comparison could be made with benchmark data or data available in the literature, the dumbell particles also clearly show the same highly erroneous behaviour in dilute systems. In contrast, utilising the Consistent approach for all particle shapes, produces excellent results with respect to both the benchmark data and independent work, and is qualitatively consistent with behaviour observed in the literature.

The erroneous dilute pressure behaviour obtained from the Naive implementation, can be directly explained by the erroneous velocity and number density profiles described in Section 5.1. In particular, the erroneous number densities result in large clusters in the centre of the computational domain Figure 4, increasing contact numbers and in turn increasing stress Eq. (17). Similarly, the erroneous velocity profiles, characterised by a reduction in velocity, result in an increase in temperature defined by Eq. (19), which increases pressure as defined by Eq. (17). Finally, the erroneous pressure variations observed with respect to \( \phi \), are accompanied with significant differences in \( \epsilon_n \) with respect to \( \phi \) as explained in Section 5.1.

The most striking commonality between all particle shapes is that, for the Naive implementation, the granular temperature is significantly higher than the results of the consistent implementation typically by over 1000%. For the Compound sphere, the resulting temperatures are also significantly higher than what would be expected.
from the literature and benchmark data as discussed earlier Section 5.2.1. Given the similar results obtained from the other particle shapes, for which a significant deviation in temperature would not be expected, the Naive implementation should be considered to result in highly erroneous temperature behaviour. For all cases utilising the Naive implementation, the high temperatures can be explained from the low velocities produced from the Naive approach Section 5.1, which causes deviation from the prescribed flow direction, increasing granular temperature of the systems Eq. (19). In contrast, the temperatures obtained using the Consistent approach, are in excellent agreement with the benchmark results and work obtained in the literature for the Compound sphere. Since a significant deviation in temperature would not be expected between each particle shape, the Consistent approach is considered highly effective in obtaining the correct granular temperature.

5.3. Summary of the key findings

In the Naive implementation, LEBCs are only applied to the constituent spheres at stage two of the MS-DEM algorithm, see Section 2 and Figure 1. Then at stage three the RBIOs are used to update the constituent sphere’s velocities and position. Since the RBIOs have not actually undergone LEBCs, the boundary conditions applied at stage two of the MS-DEM algorithm are overridden at stage three. This withdraws kinetic energy from the system, which is evidenced by the increase in granular temperature as the particle velocities deviate from simple shear flow, as well as the maximum average velocities in the system being an order of magnitude less than that of their prescribed value. This causes the non-linear velocity profiles and inhomogeneous particle distributions. Conversely, the success of the Consistent implementation is due to explicitly applying LEBCs to the RBIOs as shown in Figure 2 and then the crucial step of remapping the constituent spheres back onto the RBIOs before applying LEBCs to the constituent spheres.

6. Conclusions

A Consistent algorithm for applying the LEBCs to aspherical particle using MS-DEM techniques is presented. Irrespective of particle shape, the Consistent imple-
mentation proposed here, produces linear velocity profiles and homogeneous particle distributions, from dilute to dense systems and across a wide range of shear rates. The resulting pressures and granular temperatures are shown to be in excellent qualitative and quantitative agreement with both benchmark data produced in this work for spherical particles and data available in independent literature. A thorough evaluation of the effects of an unsuccessful implementation are provided, highlighting non-physical artefacts to be expected from an erroneous implementation. A detailed guide of how to implement LEBCs for the MS-DEM using the Consistent implementation is provided, for which the importance of remapping the centre of mass of each MS-DEM particles is essential to its success.

Acknowledgements

The authors gratefully acknowledge the support of EPSRC (Engineering and Physical Sciences Research Council) for the present work under EP/T009128/1.

References


[45] L. Verlet, Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules, Physical review 159 (1967) 98.


Update RBIO velocities by a half time step, update positions and orientations by a full time step

Map constituent spheres onto the RBIOs, updating their velocities and positions

Apply boundary conditions and perform contact detection

Stage 1

Stage 2

Stage 3

Sum the forces acting on the constituent spheres and update the RBIO velocities by the remaining half time step

Update the velocities of the constituent spheres by remaining half timestep

Figure 1: Full flowchart of the algorithm used to complete an MS-DEM time step.
Stage 1

Apply LEBCs to the constituent spheres

Perform contact detection and force calculations using the constituent spheres

Stage 2:
Naive

Stage 3

Apply LEBCs to the RBIOs

Map constituent spheres on to the RBIOs

Stage 2:
Consistent

Apply LEBCs to the constituent spheres

Perform contact detection and force calculations using the constituent spheres

Stage 3

Figure 2: Direct comparison between the Naive (red) and Consistent (green) LEBC algorithms as described in Section 3.2. Stages 1 and 3 are explained in Section 2 and depicted in Figure 1.
Figure 3: Particle shapes utilised: a) sphere, b) compound sphere, c) dumbbell and d) ellipsoid. Images are not to scale.
Figure 4: Snapshots of the xy-planes of computational domains, at $\dot{\gamma}^* = 10^{-2}$ and $\phi = 0.25$. With Figures (a), (c) and (e) taken using the Consistent implementation for the compound sphere, dumbbell and ellipsoidal shaped particles respectively. The results from the Naive implementation for the compound sphere, dumbbell and ellipsoidal shaped particles are shown in Figures (b), (d) and (f) respectively. The colour legend is calibrated to correspond the maximum, $+Vx$, and minimum, $-Vx$, velocities in the domain obtained using the Consistent implementation. Images are not to scale.
Figure 5: Representative temperature history plot for spherical particles with $\dot{\gamma}^* = 10^{-3/2}$. Only a few data points on each curve are identified with symbols for clarity.
Figure 6: The microstructural analysis of the compound sphere particle assemblies under shear and the resulting numerical errors for the Naive and Consistent implementation styles. (a) The number density errors and (b) velocity profile errors are given in with respect to $\phi$. Both (a) and (b) share a legend within which the abbreviations C, N and B are used for the Consistent implementation, Naive implementation and the spherical Benchmark data, generated in this work, respectively. (c) The dimensionless number density profiles and (d) dimensionless velocity profiles with respect to $y^*$ are given, both share a legend. The data in (c) and (d) was taken at $\dot{\gamma}^* = 10^{-3/2}$. The velocity and number density profiles obtained from the remaining values of $\dot{\gamma}^*$ appear very similar to those presented here and are provided in Appendix A.
Figure 7: The microstructural analysis of the dumbbell particle assemblies under shear and the resulting numerical errors for the Naive and Consistent implementation styles. (a) The number density errors and (b) velocity profile errors are given with respect to $\phi$. Both (a) and (b) share a legend within which the abbreviations C and N are used for the Consistent and Naive approach respectively. (c) The dimensionless number density profiles and (d) dimensionless velocity profiles with respect to $y^*$, both share a legend. The data in (c) and (d) was taken at $\dot{\gamma}^* = 10^{-3/2}$. The velocity and number density profiles obtained from the remaining values of $\dot{\gamma}^*$ appear very similar to those presented here and are provided in Appendix A.
Figure 8: The microstructural analysis of ellipsoid particle assemblies under shear and the resulting numerical errors for the Naive and Consistent implementation styles. (a) Number density errors and (b) velocity profile errors resulting from the ellipsoid particle shape are given with respect to $\phi$. Both (a) and (b) share a legend within which the abbreviations C and N are used for the Consistent and Naive approach respectively. (c) The dimensionless number density and (d) velocity profiles with respect to $y^*$, both share a legend. The data in (c) and (d) was taken at $\dot{\gamma}^* = 10^{-3/2}$. The velocity and number density profiles obtained from the remaining values of $\dot{\gamma}^*$ appear very similar to those presented here and are provided in Appendix A.
Figure 9: Dimensionless pressure and temperature results obtained for the compound sphere particle shape. (a) The pressure results with respect to $\dot{\gamma}^*$ obtained from dense systems, $\phi > 0.45$, is presented along side independent data [10]. The spherical benchmark data generated in this work was omitted in (a) for clarity but is in excellent agreement with the Consistent approach as discussed in Section 5.2.1. (b) The pressure results obtained from dilute systems $\phi < 0.55$. (c) A comparison is made for the pressure scaling $P^*$ against $\phi$ at $\dot{\gamma}^* = 10^{-3/2}$. (d) The scaled temperature versus $\phi$ at $\dot{\gamma}^* = 10^{-3/2}$, with (c) and (d) sharing a legend. Within (c) and (d) a comparison to independent work is provided [7]. Due to the nature of the scaling used in (c) and (d), minimal differences were observed with varying $\dot{\gamma}^*$ and are omitted for clarity. For all legends the abbreviations C, N and B are used for the Consistent implementation, Naive implementation and the spherical benchmark data respectively.
Figure 10: Dimensionless pressure and temperature results obtained for the dumbbell particle shape. (a) Pressure results with respect to $\phi$. (b) The scaled temperature versus $\phi$ is given in at $\dot{\gamma}^* = 10^{-3/2}$. Due to the nature of the scaling used in (b) minimal differences were observed with varying $\dot{\gamma}^*$. For all legends the abbreviations C and N are used for the Consistent implementation and Naive implementation respectively.
Figure 11: Dimensionless pressure and temperature results obtained for the ellipsoid particle shape. (a) Pressure results with respect to $\dot{\gamma}^*$, are presented along side independent data [17]. (b) The scaled temperature versus $\phi$ is given at $\dot{\gamma}^* = 10^{-3/2}$. Due to the nature of the scaling used in (b) minimal differences were observed with varying $\dot{\gamma}^*$. For all legends, the abbreviations C and N are used for the Consistent implementation and Naive implementation respectively.
Appendix A. Additional Velocity and Number Density Profiles

Additional velocity profiles and number for each particle shape as discussed in Section 5.1 are provided. All the data presented was acquired and processed as detailed in Section 4.3.
Figure A.12: Velocity and number density profile results for the compound sphere particle assemblies under shear. (a) and (b) are taken at $\dot{\gamma}^* = 10^{-1}$, (c) and (d) are taken at $\dot{\gamma}^* = 10^{-2}$, while (e) and (f) are taken at $\dot{\gamma}^* = 10^{-5/2}$. All figures use the same legend as Figure 6d.
Figure A.13: Velocity and number density profile results for the dumbbell particle assemblies under shear. (a) and (b) are taken at \( \dot{\gamma}^* = 10^{-1} \), (c) and (d) are taken at \( \dot{\gamma}^* = 10^{-2} \), while (e) and (f) are taken at \( \dot{\gamma}^* = 10^{-5/2} \). All figures use the same legend as Figure 7d.
Figure A.14: Velocity and number density profile results for the ellipsoid particle assemblies under shear. (a) and (b) are taken at $\dot{\gamma}^* = 10^{-1}$, (c) and (d) are taken at $\dot{\gamma}^* = 10^{-2}$, while (e) and (f) are taken at $\dot{\gamma}^* = 10^{-5/2}$. All figures use the same legend as Figure 8d.