LIGGGHTS is a simulation code for conducting Discrete Element Method (DEM) simulations of granular materials. Unlike some commercial codes, LIGGGHTS is not driven by a graphical user interface. Rather, the user drives the simulation by assembling a text-based input deck consisting of a series of commands to conduct the simulation. The input deck is read sequentially, so ordering of statements is important.

A LIGGGHTS input script typically has four parts:

1. Initialization
   - Set the parameters that need to be defined before particles are created
2. Setup
   - Define the material properties, particles, geometry, and particle generation
3. Detailed Settings
   - Define some settings that correspond to speed and memory utilization, output options, etc.
4. Execution
   - The actual *run* command that executes the simulation

Note that there are two basic types of statements in a LIGGGHTS input deck – individual commands and fixes. The individual commands establish basic settings for the simulation, while the fixes are used to set particular aspects of the simulation. Generally, fixes (and a few commands, e.g. region) follow a common structure of

```
fix ID group-ID style args
```

- ID = user assigned name for the fix
- group-ID = ID of the group of atoms to which the fix is applied
- style = type of fix being applied
- args = arguments used by a particular style; often consists of a keyword and corresponding value

In this tutorial, we will step through a few input decks to describe the commands and fixes to run each simulation. This tutorial is not intended to cover all possible situations and commands, but rather, to try to describe some common elements in a LIGGGHTS simulation and to provide a starting point for users to develop their own simulations. For additional or more specific information about options for each statement, please see the documentation.

Finally, at the end of this report, we provide some ‘best practices’ that have been collected over the past few years to try to help the reader avoid some common mistakes and to achieve the maximum performance out of the simulation.

Example 1 – Bin flow

A cylindrical hopper was initially filled with a set of spherical particles which were allowed to settle under the influence of gravity. The bottom of the hopper was then opened and the material was allowed to pour from the hopper.

```
### Bin Flow Simulation

### This simulation involves inserting a set of particles into a cylindrical hopper,
### allowing them to settle under gravity, and then suddenly opening a stopper at the
### bottom of the hopper and allowing the particles to pour out.  We track the
### flowrate of material from the hopper by tracking how many particles remain in the
### system as a function of time.



### Initialization

# Preliminaries
units          si
atom_style     sphere
atom_modify    map array
boundary       f f f
newton         off
communicate    single vel yes
#processors    2 2 3

# Declare domain
region         domain block -0.138 0.138 -0.138 0.138 -0.0045 0.43 units box
create_box     2 domain

# Neighbor listing
neighbor       0.003 bin
neigh_modify   every 1 check no



### Setup

# Material and interaction properties required
fix            m1 all property/global youngsModulus peratomtype 2.5e7 2.5e7
fix            m2 all property/global poissonsRatio peratomtype 0.25 0.25
fix            m3 all property/global coefficientRestitution peratomtypepair 2 0.5 0.5 0.5
fix            m4 all property/global coefficientFriction peratomtypepair 2 0.2 0.175 0.175
0.5

# Particle insertion
fix             pts all particletemplate/sphere 1 atom_type 1 density constant 1000 &
               radius constant 0.0015
fix            pdd all particledistribution/discrete 63243 1 pts 1.0
region         factory cylinder z 0 0 0.10 0.4 0.41 units box
fix             ins all insert/rate/region seed 123481 distributiontemplate pdd &
               nparticles 150000 particlerate 150000 insert_every 1000 &
               overlapcheck yes vel constant 0. 0. -1.0 region factory ntry_mc 10000

# Import mesh from cad:
fix            cad1 all mesh/surface file hopper.stl type 2 scale 0.001

# Use the imported mesh as granular wall
fix            geometry all wall/gran/hertz/history mesh n_meshes 1 meshes cad1

# Create stopper for funnel
fix            stopper all wall/gran/hertz/history primitive type 2 zplane 0.0

# Define the physics
pair_style     gran/hertz/history
pair_coeff     * *



### Detailed settings
```

Comment [A1]: Describes the boundaries of the domain as fixed (f) or periodic (p)

Comment [A2]: Determines whether or not to calculate pairwise interactions on each processor; best to leave off for a DEM simulation where only short-range interactions are presnet

Comment [A3]: Specifies how to decompose domain for parallelization; default values aren't always optimal from a simulation speed standpoint

Comment [A4]: Specifies a region called 'domain' that describes the bounds of the domain

Comment [A5]: This simulation uses two different material types; in this case, one was used for the particles and one for the bin

Comment [A6]: Neighbor statements describe how large neighbor lists will be and how often to recalculate; can affect speed and memory utilization; Default value for SI units is 0.001 m; we've enlarged it here to a value equal to the particle diameter to improve performance

Comment [A7]: When using multiple materials, the properties for each material are specified sequentially

Comment [A8]: Statements to specify material and interaction properties

Comment [A9]: When using N materials, the interactions are specified by the value N followed by a symmetric NxN matrix; note that the particle-particle and particle-wall friction coefficients are different

Comment [A10]: Describe the particles to be inserted (material type, radius, and density)

Comment [A11]: Describe the composition of the particle distribution

Comment [A12]: Describe the particle insertion; primary focus is on the number of particles, the insertion rate, and the initial velocity; This example uses the insert/rate/region method in which all particles are placed inside a specified primitive region.

Comment [A13]: Declare the geometry files to be read in to the simulation; various options for scaling, rotation, and translation are possible

Comment [A14]: Declare which geometry elements should be used as actual walls in the simulation

Comment [A15]: Use of a primitive wall; primitive options include planes and cylinders

Comment [A16]: Define the physics model to be used; pair_coeff statements can be used to describe different interactions between different particle types

```
# Integrator
fix           integrate all nve/sphere

# Gravity
fix           grav all gravity 9.81 vector 0.0 0.0 -1.0

# Time step
timestep      0.00001

# Thermodynamic output settings
thermo_style         custom step atoms ke cpu
thermo               10000
thermo_modify        norm no lost ignore

# Check time step and initialize dump file
fix           ctg all check/timestep/gran 1 0.01 0.01
run           1
unfix         ctg

# Create imaging information
dump          dumpstl all stl 10000 dump*.stl
dump          dmp all custom 10000 dump.1 id type type x y z ix iy iz vx vy vz &
              fx fy fz omegax omegay omegaz radius


### Execution and further settings

# Fill the hopper
run           150000 upto

# Pull the stopper
unfix         stopper

run           300000
```

Comment [A17]: Declaration of the integrator to use; always nve/sphere unless using multisphere particles

Comment [A18]: Establish the gravity vector

Comment [A19]: Time step for the simulation. Typically try to keep this less than or equal to 20% of the Rayleigh time.

Comment [A20]: ; Describe which quantities are to be printed in the logfile

Comment [A21]: How often to write thermodynamic quantities

Comment [A22]: Set options for thermodynamics output; 'norm' normalizes output on a per particle basis ; 'lost ignore' allows the simulation to continue running if a particle is lost from the simulation, which occurs when a particle travels outside the specified domain limits; often, this is a bad thing, as it indicates a system is losing mass incorrectly; however, for this simulation (and continuous flow processes in general), once we start discharging particles from the hopper, we want to eliminate the particles from the simulation in order to speed it up

Comment [A23]: Run the time step check over 1 time step

Comment [A24]: Generates a set of 'dump' files that contain information for imaging the system; Caution – this can produce huge files

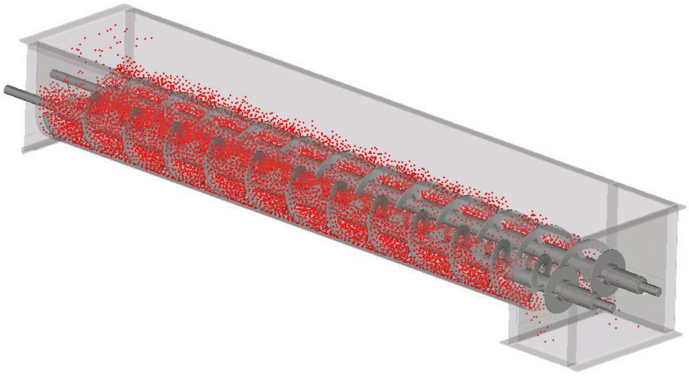
Comment [A25]: Note that these commands must be preceded by a 'run' statement to ensure that the initial dump isn't empty

Comment [A26]: Execution statement; 'upto' option causes simulation to advance up to this net timestep from current status

Comment [A27]: For this simulation, we remove a fix via the 'unfix' command to suddenly erase the stopper and initiate the flow of particles from the simulation

Example 2 – Continuous Blending Mixer

A continuous blending mixer is simulated in which a stream of particles enters at one end and is continuously blended as it travels down the length of the bed before discharging through a chute at the other end of the unit. We consider different feed rates and different impeller rotation rates. Material is fed into the simulation via a constant mass flowrate stream at one end of the unit, and is removed through a chute at the other end.



```
### Continuous Blending Mixer Simulation

### This simulation involves inserting a stream of particles at one end of a continuous
### blending mixer, conveying and mixing the material along the bed of the mixer, and then
### discharging the material through a chute at the other end of the unit.



### Initialization

# Preliminaries
units         si
atom_style    sphere
boundary      f f f
newton        off
communicate   single vel yes

# Declare domain
region        reg block -0.320 0.320 -1.960 1.460 -0.290 0.335 units box
create_box    1 reg



### Setup

# Material and interaction properties required
fix           m1 all property/global youngsModulus peratomtype 2.5e7
fix           m2 all property/global poissonsRatio peratomtype 0.25
fix           m3 all property/global coefficientRestitution peratomtypepair 1 0.5
fix           m4 all property/global coefficientFriction peratomtypepair 1 0.5
fix           m5 all property/global coefficientRollingFriction peratomtypepair 1 0.1

# Particle insertion
region        factory block -0.225 0.225 -1.650 -1.450 0.3 0.33 units box
fix           pts all particletemplate/sphere 1 atom_type 1 density constant 1000 &
              radius constant 0.005
fix           pdd all particledistribution/discrete 14127 1 pts 1.0
fix           ins all insert/rate/region seed 132412 distributiontemplate pdd &
              nparticles 10000000 massrate 1.66666667 insert_every 1000 &
              overlapcheck yes vel constant 0. 0. -1. region factory ntry_mc 10000

# Import mesh from cad:
fix           cad1 all mesh/surface file trough2.stl type 1 scale 0.001 curvature 1e-5
fix           cad2 all mesh/surface file left_shaft2.stl type 1 scale 0.001 curvature 1e-5
fix           cad3 all mesh/surface file right_shaft2.stl type 1 scale 0.001 curvature 1e-5

# Use the imported mesh as granular wall
fix           mixer all wall/gran/hertz/history mesh n_meshes 3 meshes cad1 cad2 cad3 &
              rolling_friction cdt


# Define the physics
pair_style    gran/hertz/history rolling_friction cdt
pair_coeff          * *


### Detailed settings

# Integrator
fix           integrate all nve/sphere

# Gravity
fix           grav all gravity 9.81 vector 0.0 0.0 -1.0
```

Comment [A28]: Rolling friction must also be listed below in statements declaring contact physics and granular wall to be enabled

Comment [A29]: # of particles is set very high to ensure that particles will continue to be generated throughout the duration of the simulation

Comment [A30]: Rolling friction declaration for particle-wall collisions

Comment [A31]: Use multiple CAD components; alternatively, each CAD geometry can be enabled via a separate wall/gran/hertz/history statement, which can facilitate later addition/removal of geometry

Comment [A32]: Rolling friction declaration for particle-particle collisions

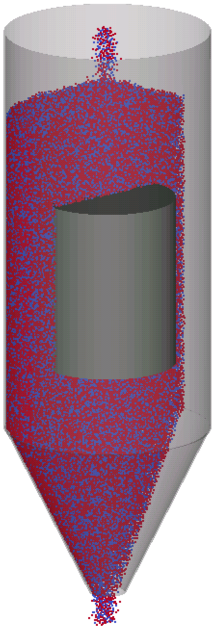
```
# Timestep (keep < 20% T_Rayleigh)
timestep     0.00003125

# Thermodynamic output settings
thermo_style        custom step atoms ke cpu
thermo              1600
thermo_modify       lost ignore norm no

# Rotate the shafts
fix          movecad1 all move/mesh mesh cad2 rotate origin -0.1369 0. -0.0462 &
             axis 0. 1. 0. period 1.5   # 40 RPM
fix          movecad2 all move/mesh mesh cad3 rotate origin  0.1369 0. -0.0462 &
             axis 0. 1. 0. period -1.5  # 40 RPM
```

Comment [A33]: Enable moving mesh commands which allow for motion of the imported geometric bodies. Motions include translations, rotations, vibrating translations, and vibrating rotations.

```
# Check time step and initialize dump file
fix          ctg all check/timestep/gran 1 0.01 0.01
run          1
unfix ctg

# Create imaging information
dump          dumpstl all stl 10000 dump*.stl
dump          dmp all custom 10000 dump.1 id type type x y z ix iy iz vx vy vz &
              fx fy fz omegax omegay omegaz radius



### Execution and further settings

# Run to 300 sec to equilibrate system
run          3840000 upto
```

Example 3 – Couette cylinder

A Couette cylinder with axial flow, based on work by Kheiripour et al[1], is depicted in the figure below. The device is intended to look at the rheological behaviour of a granular flow and to permit study of the tendency for material to segregated based on particle size. In these simulations, we have randomly inserted two different particle sizes (particles are otherwise identical) over a one second interval. The stopper at the bottom of the hopper is then pulled and material flows through a periodic z-boundary condition to flow back into the top of the unit. In doing so, the net mass in the system remains constant. Simultaneously, the drum in the center of the unit rotates about its axis, setting up a shearing action inside the cylinder.



[1] *Study of powder flow patterns in a Couette cell with axial flow using tracers and solid fraction measurements*, M. Kheiripour Langroudi, P. R. Mort, Gi. I. Tardos, Granular Matter, Oct 2011, Vol 13, Iss 5, pp 541-552

```
### Couette cylinder simulation

### This simulation first inserts a set of particles into the Couette cylinder and allows
### them to settle.  Then the interior cylinder is set to rotate at a constant rate and the
### bottom of the hopper is opened to allow material to flow in the vertical direction.  A
### periodic boundary is used in the z-direction to reinsert the flow of material and
### maintain a constant mass of material in the apparatus.



### Initialization

# Preliminaries
units         si
atom_style    sphere
boundary      f f p
newton        off
communicate   single vel yes

# Declare domain
region        reg block -0.078 0.078 -0.078 0.078 -0.1524 0.306 units box
create_box    2 reg



### Setup

# Material properties and interactions
fix           m1 all property/global youngsModulus peratomtype 2.5e7 2.5e7
fix           m2 all property/global poissonsRatio peratomtype 0.25 0.25
fix           m3 all property/global coefficientRestitution peratomtypepair 2 0.5 0.5 0.5 0.5
fix           m4 all property/global coefficientFriction peratomtypepair 2 0.5 0.5 0.5 0.5
fix           m5 all property/global coefficientRollingFriction peratomtypepair 2 0.1 0.1 0.1
0.1

# Particle insertion
fix           pts1 all particletemplate/sphere 1 atom_type 1 density constant 1000 &
              radius constant 0.00125
fix           pts2 all particletemplate/sphere 1 atom_type 2 density constant 1000 &
              radius constant 0.00100
fix           pdd all particledistribution/discrete 12345 2 pts1 0.5 pts2 0.5
fix           ins_mesh all mesh/surface file factory.stl type 1 scale 0.001
fix           ins all insert/stream seed 123481 distributiontemplate pdd &
              nparticles 450000 particlerate 900000 overlapcheck yes &
              vel constant 0. 0. -3.0 insertion_face ins_mesh extrude_length 0.02

# Import mesh from cad:
fix           cad1 all mesh/surface file outer_cylinder.stl type 1 scale 0.001 curvature 1e-5
fix           cad2 all mesh/surface file inner_cylinder.stl type 1 scale 0.001 curvature 1e-5
fix           cad3 all mesh/surface file funnel.stl type 1 scale 0.001 curvature 1e-5
fix           cad4 all mesh/surface file plate.stl type 1 scale 0.001 curvature 1e-5

# Use the imported mesh as granular wall
fix           geometry all wall/gran/hertz/history mesh n_meshes 4 meshes cad1 cad2 cad3 &
              cad4 rolling_friction cdt

# Define the physics
pair_style    gran/hertz/history rolling_friction cdt
pair_coeff    * *



### Detailed settings

# Integrator
fix           integrate all nve/sphere
```

Comment [A34]: Using a periodic boundary in the z-direction so we can pour material back and forth between two vessels while minimizing the amount of extraneous vessel motion

Comment [A35]: 3 different materials used representing two particle types and one for the walls of the geometry. Note that all material properties are identical; the separation by type facilitates grouping and tracking for determining how well-mixed a system is

Comment [A36]: Inserting multiple particle types within a single factory

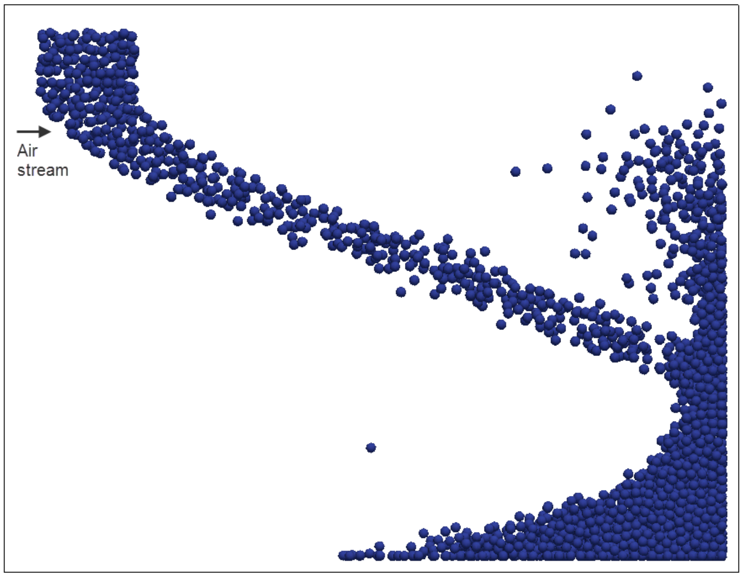
Comment [A37]: Import the CAD that will define the insertion volume ; CAD image must be 2-dimensional

Comment [A38]: Using insert/stream for particle insertion; note that the extrude_length defines the rate at which particles are inserted

```
# Gravity
fix          grav all gravity 9.81 vector 0.0 0.0 -1.0

# Timestep
timestep            0.00000625

# Thermodynamic output settings
thermo_style        custom step atoms ke cpu
thermo              8000
thermo_modify       lost ignore norm no

# Check timestep
fix          timecheck all check/timestep/gran 1 0.01 0.01
run          1
unfix        timecheck

# Dump output
#dump        dmp all custom 16000 dump.1 id type type x y z ix iy iz vx vy vz fx fy fz &
             omegax omegay omegaz radius
#dump        dumpstl all stl 16000 dump*.stl
#dump        dmp2 all custom 16000 dump.txt type x y z radius



### Execution and further settings

# Run 1.0 sec to insert and settle particles
run          160000 upto

# Remove the stopper and start the rotation
unfix        geometry
fix          geometry all wall/gran/hertz/history mesh n_meshes 3 meshes cad1 cad2 cad3 &
             rolling_friction cdt
fix          movecad all move/mesh mesh cad2 rotate origin 0. 0. 0. axis 0. 0. 1. period 1.

# Run 30 sec
run          4800000
```

Example 4 – 1-Way Flow Coupling

A narrow slot is filled by a stream of particles falling near one side, with the flow passing across a narrow jet of air. The air stream causes the particles to blow towards the opposite end of the slot and creates a pile against the far wall.



In this example, we've designed the stream force to be added via the 'fix addforce' command. Using this command, we have designated a small region of space immediately below the inlet stream of particles that causes all particles that enter the region to experience a body force that pushes them laterally. While we've used a simple constant force in this example, it is also possible to use variable quantities for the force to represent more complicated flow fields. In addition, the force can be applied to certain groups of particles.

While this flow treatment is obviously reduced in fidelity from a truly coupled CFD-DEM simulation, it does provide a simple way to at least include some influence of an external flow field. Thus, for situations in which the particles do not significantly alter the flow field, we can use CFD to determine the flow field in the absence of the solid phase, and then apply this flow field to the DEM simulations to better understand the actual particle dynamics. One could envision using this to model the segregation due to drag forces acting on different sized particles.

```
### 1-way flow coupling

### This simulation involves inserting particles into slot across a horizontal air
### stream that applies a force to the particles and pushes them sideways, creating
### a pile-up at the far end of the slot.  The focus here is on the use of the applied
### body force to act as a weak 1-way coupled flow field.  While not a general substitute
### for the full 2-way coupling, it is useful in some circumstances.


### Initialization

units           si
atom_style      sphere
boundary        f f f
newton          off
communicate     single vel yes

# Declare domain
region          reg block -0.155 0.155 -0.015 0.015 -0.005 0.235 units box
create_box      2 reg


### Setup

# Material and interaction properties
fix             m1 all property/global youngsModulus peratomtype 2.5e7
fix             m2 all property/global poissonsRatio peratomtype 0.25
fix             m3 all property/global coefficientRestitution peratomtypepair 1 0.5
fix             m4 all property/global coefficientFriction peratomtypepair 1 0.5

# Physics
pair_style      gran/hertz/history
pair_coeff      * *

# Particle insertion
fix             pts all particletemplate/sphere 1 atom_type 1 density constant 1000 &
                radius constant 0.002
fix             pdd all particledistribution/discrete 34234 1 pts 1.0
region          factory block -0.14 -0.10 -0.006 0.006 0.215 0.225 units box
fix             ins all insert/rate/region seed 54325 distributiontemplate pdd &
                nparticles 10000 particlerate 10000 insert_every 500 overlapcheck yes &
                vel constant 0. 0. -2.0 region factory ntry_mc 10000

# Geometry
fix             wall1 all wall/gran/hertz/history primitive type 1 zplane 0.0
fix             wall2 all wall/gran/hertz/history primitive type 1 xplane -0.15
fix             wall3 all wall/gran/hertz/history primitive type 1 xplane 0.15
fix             wall4 all wall/gran/hertz/history primitive type 1 yplane -0.01
fix             wall5 all wall/gran/hertz/history primitive type 1 yplane 0.01

# Applied flow field
region          spray block -0.15 -0.05 -0.01 0.01 0.1 0.2 units box
fix             push all addforce 0.01 0.0 0.0 region spray
```

Comment [A39]: Define the region where the flow field is acting

Comment [A40]: Define the vector representing the force acting in the above region

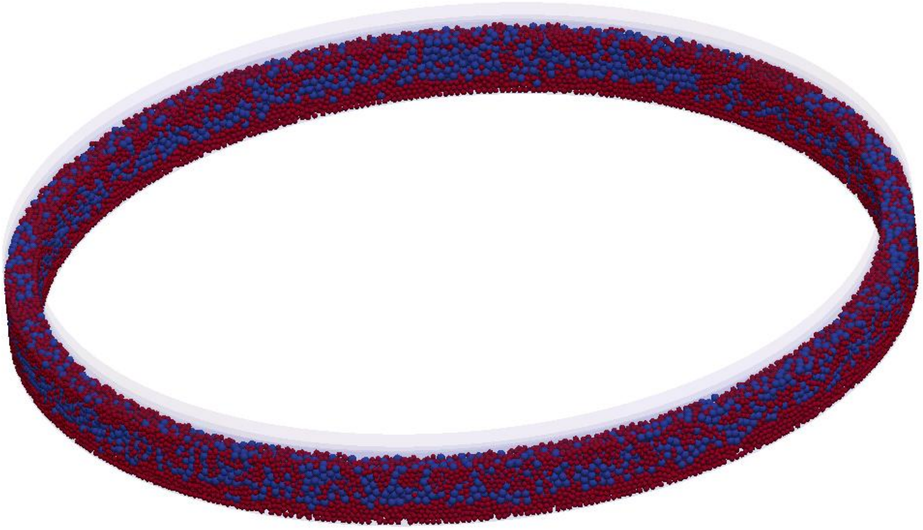
```
### Detailed settings

# Integration
fix             integrate all nve/sphere

# Gravity
fix             grav all gravity 9.81 vector 0.0 0.0 -1.0

# Timestep
```

```
timestep        0.00001

# Thermo settings
thermo_style    custom step atoms ke cpu
thermo          10000
thermo_modify   lost ignore norm no
compute_modify  thermo_temp dynamic yes

# Run 1 step to check timestep and initialize
fix             ctg all check/timestep/gran 1 0.01 0.01
run             1
unfix           ctg

# Dump particle positions
dump            dmp all custom 100 dump.1 id type type x y z ix iy iz vx vy vz fx fy fz &
                omegax omegay omegaz radius



### Execution and further settings

# Run remainder
run             200000 upto
```

Example 5 – Shear Cell

This simulation is designed to highlight the use of the servo-controlled wall, in which a geometric construct can move in response to a collision with a particle. Typical walls in a DEM simulation are assumed to be immune to the influence of particle impacts, and so wall motions must be completely prescribed. Using the servo wall allows us to consider additional systems, such as the shear cell shown here, in which the wall position is determined by a combination of a subjected load and impacts with the particles.

An annular ring was initially filled with two sets of spherical particles (red = 4 mm diameter and blue = 6 mm) which were allowed to settle under the influence of gravity. The particles were filled sequentially with the larger blue particles on the bottom, followed by the red particles on top. After the cell is filled, an annular lid is lowered onto the top of the cell and held with a load on top. The lid is allowed to move in the z-direction in response to stresses imparted by the particles and the load on top. Once the lid has been placed, the bottom of the ring is slowly rotated to create a shear field and instigate segregation.



Once the shearing action starts, the smaller particles percolate down through the layer of larger particles, causing the material to vertically segregate.

```
# Shear Cell example

### Initialization

# Preliminaries
units           si
atom_style      sphere
boundary        f f f
newton          off
communicate     single vel yes

# Declare domain
region          reg block -0.325 0.325 -0.325 0.325 -0.001 0.081 units box
create_box      2 reg



### Setup

# Material and interaction properties
fix        m1 all property/global youngsModulus peratomtype 1.0e7 1.0e7
fix        m2 all property/global poissonsRatio peratomtype 0.25 0.25
fix        m3 all property/global coefficientRestitution peratomtypepair 2 0.5 0.5 0.5 0.5
fix        m4 all property/global coefficientFriction peratomtypepair 2 0.5 0.5 0.5 0.5

# Physics
pair_style      gran/hertz/history
pair_coeff      * *

# Particle setup - insertions to be handled below
fix        pts1 all particletemplate/sphere 1 atom_type 1 density constant 1000 &
           radius constant 0.003
fix        pdd1 all particledistribution/discrete 63243 1 pts1 1.0
fix        pts2 all particletemplate/sphere 1 atom_type 2 density constant 1000 &
           radius constant 0.002
fix        pdd2 all particledistribution/discrete 45324 1 pts2 1.0

# Geometry
fix        cad1 all mesh/surface/stress file shear_cell.stl type 1 scale 0.001 stress on
fix        cad2 all mesh/surface/stress file bottom_plate.stl type 1 scale 0.001 stress on
fix        ins_mesh all mesh/surface/planar file factory.stl type 1 scale 0.001
fix        geometry all wall/gran/hertz/history mesh n_meshes 2 meshes cad1 cad2
```

Comment [A41]: Insert some of the geometry components. A loaded top plate will be added later once particles have been inserted.

```
### Detailed settings

# Integration
fix             integrator all nve/sphere

# Gravity
fix             grav all gravity 9.81 vector 0.0 0.0 -1.0

# Timestep
timestep        0.00002

# Thermodynamic output
thermo_style          custom step atoms ke cpu
thermo                10000
thermo_modify         lost ignore norm no



### Execution and further settings
```

```
# Particle insertion - large particles
fix         ins1 all insert/stream seed 123481 distributiontemplate pdd1 nparticles 5000 &
            particlerate 10000 overlapcheck yes vel constant 0. 0. -2. &
            all_in yes insertion_face ins_mesh extrude_length 0.01
run         50000

# Particle insertion - small particles
fix         ins2 all insert/stream seed 431524 distributiontemplate pdd2 nparticles 16875 &
            particlerate 33750 overlapcheck yes vel constant 0. 0. -2. &
            all_in yes insertion_face ins_mesh extrude_length 0.01
run         50000


fix         cad3 all mesh/surface/stress/servo file top_plate.stl type 1 scale 0.001 &
            stress on com 0 0 0.06 dim z ctrlPV force target_val -10.0 vel_max 1.0
fix         geometry2 all wall/gran/hertz/history mesh n_meshes 1 meshes cad3

# Update the thermodynamic output
thermo_style     custom step atoms ke cpu f_cad3[1] f_cad3[2] f_cad3[3]
thermo           50000

# Settle the plate on top of the particles
run         50000

# Set the dumps
dump        dmp all custom 50000 dump.1 id type type x y z ix iy iz vx vy vz fx fy fz &
            omegax omegay omegaz radius
dump        dmpstl1 all mesh/stl 50000 cell*.stl cad1 cad2
dump        dmpstl2 all mesh/stl 50000 lid*.stl cad3

# Describe shearing action
fix         movecad all move/mesh mesh cad2 rotate origin 0. 0. 0. axis 0. 0. 1. period 20.0

# Shear
run         30000000
```

Comment [A42]: Particle insertion steps were moved here as they will be staged to create the two particle layers

Comment [A43]: Insert plate, restrained in the z-direction, and loaded with a force of 10 N, on top of the particle bed

Comment [A44]: Enable contacts with the new wall

Comment [A45]: Added the stress components for the top plate to the thermodynamic output

Comment [A46]: Dump commands were moved here as the insertion and settling were not of interest

Comment [A47]: Split the geometry dump into separate files to make visualization easier.

Comment [A48]: Rotate the bottom plate to create a shear field

Best Practices

Stability

- Keep the timestep below 20% TR for stability. Use the fix_check_timestep_gran to ensure that your timestep is not too large.
- Use 'thermo modify lost ignore' for continuous processes in order to allow the simulation to ignore particles that leave the simulation domain. Otherwise, the simulation will shut down once the first particle is lost.
- In conjunction with the previous note, do not ignore lost particles for batch processes as a lost particle is an indication that the simulation isn't stable.
- Pay attention to the kinetic energy and cpu time in your thermo output.
    - If the kinetic energy is jumping about violently, it is an indication that the timestep may be too large. This could result in a particle deeply penetrating another body during the course of one timestep, resulting in a huge repulsive force and high corresponding kinetic energy in the next step.
    - The cpu time per step should remain fairly constant when at equilibrium, and should follow a fairly smooth progression when filling a system. If the time/step suddenly starts to increase dramatically during filling, it is an indication that something is blocking the feed zone and preventing you from achieving the desired fill rate. This may also manifest as warnings that fewer particles were inserted than desired.

Particle generation

- If you see warnings that the simulation was unable to insert the desired amount of particles, it is an indicator that you are trying to feed too aggressively. To counter this, either (1) decrease the feed rate, (2) increase the size of the insertion zone, or (3) increase the initial velocity of the particles so they clear the zone more quickly.
    - The insertion zone is often limited by the size of the geometry. In cases where it is not, it can sometimes be tempting to enlarge the zone dramatically to ease particle insertion. However, this can cause two problems. First, if the zone is large in the direction of gravity, particles may accelerate to sufficiently high velocities so as to cause stability problems where a particle will impact another body at too high of a speed to be handled by the base timestep. Second, it may force you to expand the domain of the simulation, which can slow the simulation down (depends on processor assignment).
    - Setting the initial velocity too high can lead to stability problems as described above.
- If the simulation initially inserts particles without issue, but then starts to suffer problems where it is unable to insert the desired number of particles, it is an indication that something is now obstructing the insertion zone. Most likely is that the previously inserted mass has backed up or formed a pile that is now intruding into the insertion zone.
- It is generally a good idea to make sure a batch system is fully enclosed so as to avoid losing particles during the simulation. However, the presence of a lid can limit the ability to insert particles. If the lid is inserted as a separate object, it can be moved about before and after the particle insertion to facilitate the filling operation, or simply added to the system after all particles have been added.
- Using different material types for different groups of particles, even if all of the material properties are identical, can be a convenient way to identify particles and groups of particles during post-processing. This is especially true for mixing problems.
- Of the three insertion methods (pack, rate/region, and stream), it is recommended to use the rate/region or stream methods. The pack method relies on being able to fill a volume with a large number of particles at one time via random sequential addition, which will typically fill a volume to no more than 30%. This can make it difficult to achieve the desired number of particles and will often leave a large empty space in the simulation domain that reduces the efficiency of parallelization. Using the

rate/region or stream options allows you to use a much smaller insertion volume and insert material over time. Rate/region is preferred in general over stream solely because the user can define the insertion region from within the input deck and does not require the user to create and import some external CAD. That said, sometimes (e.g. filling an annular ring) is accomplished far more efficiently with the rate/stream option.

Geometry

- Use a simple mesh, even w/ high-aspect ratio triangles. Tests thus far indicate no difference in accuracy of results (**disclamer – more testing is needed here, especially for stress calculations!**) between using a coarse mesh as generated directly from a CAD program, and using a fine mesh as generated by a tool such as gmsh.
    - The high-aspect ratio triangles may diminish the ability to parallelize the geometry, but if there are significantly fewer facets than a fine description, you may end up saving more time in the long run. In many simulations, the number of particles will far exceed the number of facets in the geometry, and so parallelizing the geometry isn't as critical.
- Simplify the geometry to avoid including parts that will never touch the particles (e.g. the motor that drives a mixer).
- Simplify geometry to avoid very small curved surfaces if possible. These may end up using a lot of facets that aren't really that important.
- Keep in mind that gravity doesn't always have to point down.

Speed-ups

- Keep the domain size as small as possible without losing geometry. This will avoid wasting processing power on empty space.
- Pay attention to the processor distribution. For many systems, you can take advantage of knowing where your particles will be during the simulation to assign processors in an advantageous way and enhance the speed of the simulation. For example, if you have a batch mixer where the particles are gently mixed and generally occupy only the lower half of the mixer volume, using two processors in the z-direction results in the top-most layer of processors doing very little work and not being used effectively. Using a single layer in the z-direction instead will lead to a significantly faster simulation.
- In the absence of testing, set the skin size for neighbor list calculations to ~ 1 particle diameter. Setting the neighbor list skin size too small leads to overly frequent recalculations of the neighbor list, while setting it too large leads to huge memory requirements. The optimal setting is a balance between the two and depends on the size of the particles, the density of the system, and the speed at which particles are moving about. A single guideline for all cases is thus difficult to set, and ideally some testing should be done to optimize the simulation.
- Adjust physical properties as possible for speed. For example, modulus can often be set much lower than realistic values, yielding a significant speedup, without impacting the overall simulation results. Typically, setting the elastic modulus to ~10^7 Pa is sufficiently high so as to capture the basic particle dynamics. Increasing the density also has a positive effect by allowing for a larger timestep, though typically we are not able to alter density by as much as an order of magnitude.

Visualization/post-processing

- You can export the geometry in multiple sets of dump files to facilitate visualization by allowing you to set the transparency on a per-body basis.
- You can use multiple particle types, even if they all have the same material properties, to help with post-processing elements. For example, when considering the performance of a mixer, having two particle types, with one type fed to each side of the mixer, it becomes trivial to visualize mixing performance as you can color by particle type. You can also calculate quantities like a mixing index based on particle type.