IJIRST –International Journal for Innovative Research in Science & Technology| Volume 2 | Issue 02 | July 2015
ISSN (online): 2349-6010

Study of Interaction between Discrete Granular Particles "An Effort to Simulate a Few Basic Granular Flow Problems"

Shivam Bahuguna
B. Tech Student
Department of Mechanical Engineering
National Institute of Technology, Warangal

Abstract

The field of granular flows has become an important area for researchers due to its application in various industrial areas. One vital reason for recent activity hike in this region is due to the reliance of various industries on transporting and storing granular materials, including pharmaceutical industry which relies on processing of powders and pills, agriculture and food processing industry where seeds, grains and foodstuffs are transported and manipulated, as well as all construction based industries. Additional manufacturing processes, like in automotive industry, rely on casting large metal parts in carefully packed beds of sand. Yet the technology for handling and controlling granular materials is not yet fully developed. Estimates show that about 60% of the capacity of many industrial plants is being wasted due to problems related to the transport of these materials from part of factory floor to another. Hence even a small improvement in the understanding of how granular media behave should have large impact on industry. Initially, one challenge was the realistic simulation of granular materials, like sand or powders, consisting of millions of particles. As an effect, granular flow was extensively taken by research scholars in institutes, which resulted in the development of YADE framework for granular flow simulations. The following paper describes the use of YADE in modelling a few selected problems to show how YADE can be effective to simulate other real life problems related to granular flows.

Keywords: DEM, YADE, Granular Particle, Simulation, FEM, Engine, Navier-Stokes Equation

I. INTRODUCTION

The approach towards the microscopic understanding of macroscopic particulate material behavior is the modelling of particles using so-called discrete element methods (DEM). Even though millions of particles can be simulated, the possible length of such a particle system is in general too small in order to regard it as macroscopic. These "microscopic" simulations of a small sample can be used to derive macroscopic constitutive relations needed to describe the material within the framework of a macroscopic continuum theory. For granular materials, as an example, particle properties and interaction laws are inserted into DEM and lead to collective behavior of dissipative many-particle system. From a particle simulation, one can extract, *e.g.,* the pressure of the system as function of density. The equation of state allows a macroscopic description of material, which can be viewed as compressible, non-Newtonian complex fluid, including a fluid-solid phase transition. [1]

In the following paper, an idea of how DEM is used to simulate simple phenomena of particle interaction is conveyed.

A. Insight into DEM:

Abbreviation of Discrete Element Method, it is simulation technique widely used for study of granular material. DEM is a numerical method used to compute the stresses and displacements in a volume containing large number of particles such as grains of sand. The granular material is modelled as an assembly of rigid particles and the interaction between each particle is explicitly considered. The particle shapes and geometries are specified by user. Spheres or ellipsoids are commonly used. DEM has already been extensively used to replicate soil particulate nature [2]. The DEM method was initially developed for the analysis of rocks. DEM has rightly proven to be a substitute for Continuum as Continuum Material behavior descriptions combine multiple processes into complex functions that are too difficult to apply to variable and inhomogeneous materials. DEM explicitly describes the dynamics of assemblies of particles and micro-mechanical interaction processes between grains – including inhomogeneity. Interactions are at grain scale, explicit algorithms are present for separate physical dynamic or quasi static processes, complex behavior is captured through the separately acting physical process algorithms and these algorithms can also be combined with any other numerical technique to solve potential problems. But, computationally, DEM is expensive and construction of realistic DEM particle bed is difficult. DEM differs from FEM as FEM models the media as continuum and describes the soil by point wise mathematical expressions, i.e. stress in a point x: sigma(x). [3]

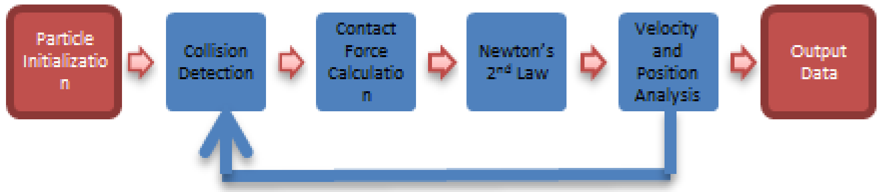


Fig. 1: Process flow in DEM

The figure clearly shows the working of DEM algorithm. After particle has been identified, and whenever a collision is detected, contact forces are calculated. This data is used to calculate velocity and position of the particle using Newton's second law of motion and data is fed for analysis. Now this force calculation undergoes an iterative process, calculating new forces at each time step and modifying the results.

B. Insight into Granular Materials:

Granular materials can be considered as large conglomerations of discrete macroscopic particles. Sand is a typical example of granular particles. The size of Granular materials ranges from 10 micrometers to 3 millimeters, classified further as granular powders (10 micrometers-100 micrometers) and granular solids (100 micrometers-3 millimeters). They can be deformed as solid bodies or soil, they may have flow ability like liquids and compressibility like gases. These can be stated as a separate form of matter from solid, liquid and gas because of their peculiar behavior.

If the inter-granular forces are mostly non-corrosive, then the shape of material is mostly decided by the gravity and the external boundary conditions. In case they are dry, the role of inter-particle fluid can be neglected in determining few properties of the granular flow, but not all. The behavior of these was tried to relate to dense gaseous flows as gases too comprise of large distant molecular particles, but, studies revealed that unlike gases, KT plays no role in granular flows. Instead, potential energy is defined by $m * g * d$, mass of granular flow raised to its diameter against gravity, which is way too large than KT. Hence, efforts have been made to study granular flow behavior which ultimately became the base of present research and simulation processes of granular flows. Various theories have been put for asserting the interactions between granular particles. [4]

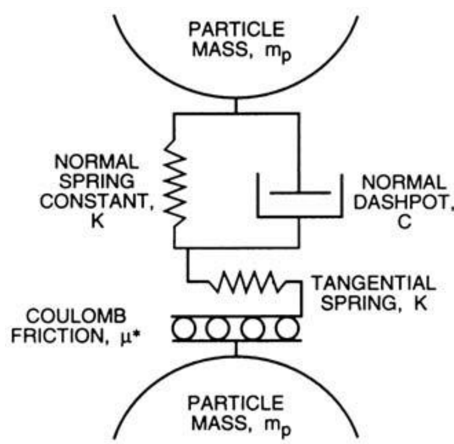


Fig. 2: Schematic of soft particle Interaction model of particle interaction[5]

C. Insight into YADE:

Abbreviation for Yet Another Dynamic Engine, YADE is an open-source framework used extensively for discrete numerical modelling. It makes use of C++ and Python scripts for computational and simulative modelling. It was developed as part of a project involving study of granular flows and is widely used these days in research fields for simulation involving discrete modelling of granular particles. Its capability to import CAD files and allowance to make changes in structure and meshing of the imported geometry makes it a very flexible tool for DEM simulations.

The paper aims at using physical DEM approach to simulate the experiments which define grain-grain interactions by specifying mechanical properties, particle contact mechanisms and the physical properties in the code algorithms.

II. EVALUATION OF FORCES AND TORQUE ALGORITHMS

The force evaluation algorithm in DEM requires the exact degree of overlap, velocities at the contact and previous time step forces at the contact and then the new forces computed are needed to be stored in the variable storage, which will again undergo iteration in the next time step.

III. CONTACT DETECTION MODELS

DEM is all about specifying contact interactions between the bodies, computation of all the forces on a body and then the integration of all the forces to evaluate various derived parameters like acceleration, velocity, position, etc.

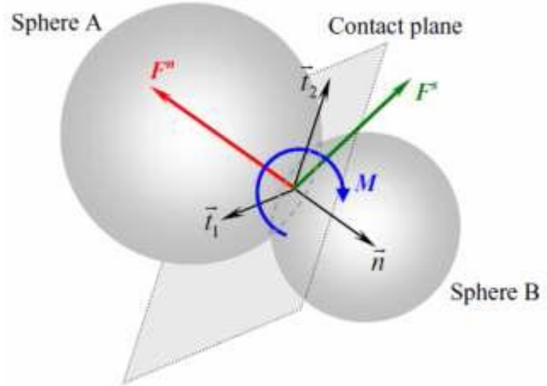


Fig 3: Force generation in particle interaction[6]

Various theories have been put to specify the contact models like the "Linear contact model", "Hertz contact model" etc., each having its own peculiarity. [7]

-->Linear model

$$f_{(hys)} = K1 * d \text{ (loading/reloading)}$$

-->Hertz model

$$f_{(hys)} = K1 * D^{1.5} \text{ (loading/reloading)}$$

For the formulation of simulations described here, basic linear model for the computation has been used. The method is simple and easy to implement, being linear.

IV. CODE ANALYSES

YADE algorithms use components commonly used by various simulation types (DEM, FEM, Lattice or SPH), like:

- Newton's law or Hooke's law
- Time integration algorithms
- Damping methods
- Collision detection algorithms
- Boundary conditions
- Data classes (to store information about bodies or interactions)
- Common OpenGL methods (to draw popular geometries)

Considered that the simulation involves bodies between which interactions occur. These interactions can be detected and processed by certain computational algorithms and physical rules (engines in YADE). Results of these algorithms can be moment, force, displacement, etc., which in general produce a response that affects body state. All bodies, interactions and the simulation loop that processes them (engines) are stored inside the World class, which is called each time we need to run the simulation. It is to be noted at this point of time that all algorithms are engines in YADE. [8]

Data is distinguished in YADE as:

- Bodies
- Interactions

A. Data classes:

The objects of data classes cannot move or interact within themselves, as they only contain data. Their movement and interaction are handled by the engine classes. The body is represented by six data classes:

- BodySate
- BodyStateConstraints
- BodyConstitutiveParameters
- BodyShape
- BodySimplified-Shape
- BodyBoundingVolume

B. Engine Classes:
Every operation concerning data is performed by a dedicated Engine. Applying boundary conditions, moving, creating, modifying, destroying, displaying, loading, saving, calculating, converting, interpreting — all these functions are performed by some specific Engine class. The engine classes used in the simulations descried here are-

- ForceResetter- Resets all forces stored in Scene. (O.forces in python), so that fresh calculations can be carried out at every new time step.
  - Typically, this is the first engine to be run at every step.
- InsertionSortCollider- invokes the desired collision detection model in algorithm.
- InteractionLoop- Unified dispatcher for handling interaction loop at every step, for parallel performance reasons.
- NewtonIntegrator- integrates Newtonian motion equations
- PyRunner- saves data accessed. It inherits periodicy control from PeriodicEngine.

V. SIMULATION OVERVIEW

The Omega class is a top–level object representing the simulated world, containing both data and engines operating on it. The engines are executed by calling sequentially the isActivated() method for each Engine, and if the answer was positive, then calling action(). It is up to user to specify what engines are inside the simulation loop. When top–level Omega class is loaded from file, its initializers are invoked, one after another. Usually a BodySimplifiedShape is generated from provided BodyShape. BodyBoundingVolume is generated from BodySimplifiedShape. Even BodyShape can be generated here according to some algorithm, or by loading it from another file written in different format (eg. exported from netgen, gmsh or some other program that can perform model discretization, as has been done in this project). When the simulation is started, engines stored in simulation Loop are executed sequentially. Usually this involves detecting interactions, solving them, applying solution results to bodies and saving some data to disk.

The computational parts are available in C++ and the files can be easily used to inherit the methods and classes required. The simulation script is written in Python and is run in YADE. Various parameters like material, position, and other mechanical and physical parameters were set inside the python script. Suitable engines were invoked and various modules were imported from the open source YADE.

VI. "THE BOUNCING SPHERE"

A. Project Description:
The project aims at simulation of a case where a sphere of fixed dimensions is allowed to collide with another sphere of same dimensions under the effect of gravity. DEM concepts have been implemented and the process is simulated in YADE. Two separate cases of simulation of a single pair of spheres and two spheres of spheres have been considered.

B. Physical Parameters:
The physical parameters in DEM refer to the particle size, shape, specific gravity, contact area radius (or friction angle) and the dilating sphere radius.

- Particle Size: 20 microns
- Particle Shape: spherical shaped balls
- Specific Gravity : 2
- (density of particles with respect to water)
- Friction angle: 26.56°
- Acceleration due to gravity, g: 9.81 m/s2

C. Mechanical Parameters:
The mechanical parameters in DEM refer to contact friction coefficient, normal contact stiffness and the contact tensile strength, creep viscosity, normal viscosity (for bonded particles).

- Contact time, tc: 0.0001 seconds
- Contact Friction Coefficient: 0.5
- Coefficient of restitution in normal direction, en: 0.7
- Coefficient of restitution in tangential direction, et: 0.7

The above properties were stored globally in the global variables so that any class can inherit these entities with a little improvisation to include new materials. The Simulation shows two granular particles, spherical in shape, with one in a fixed mode at coordinate [0,0,0] with the properties assigned as shown below:

- Radius=20 microns
- Friction angle=26.56°

- Contact time , Tc=tc
- Normal coefficient of restitution, En=en
- Tangential coefficient of restitution, Et=et

The second granular ball is located at the coordinate [0,0,(5 times the radius)] and is in a free state(FIXED=FALSE). The properties assigned to this ball are as follows:

- Radius=20 microns
- Friction angle=26.56°
- contact time , Tc=tc
- Normal coefficient of restitution, En1=en/2
- Tangential coefficient of restitution, Et1=et/2

The material properties of both the balls are stored in a global variable, mat1 and mat2 which were called at the time of invoking the variables defining the properties of each grain.

D. Results and Interpretations:

1) Test for Position Evaluation At Various Time-Steps For A Single Pair Bouncing Sphere Problem

"Graph for Position v/s Time at various time steps"

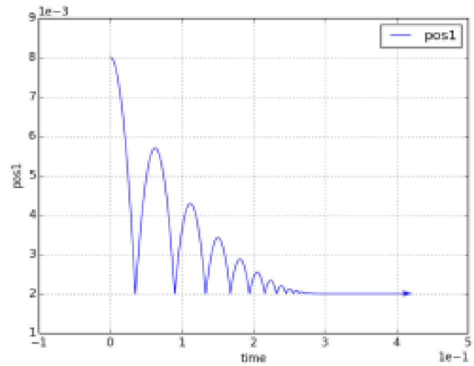


Fig. 4: Graphs for Position with Time at few steps

The graphs depict the variation in relative position of the two grains. The graph is self-explanatory in the sense that it depicts the fall in position of the sphere till the time the spheres collide but due to impact, the free sphere bounces back, but with a less initial velocity and this goes until the bounce velocity becomes 0, and the free sphere comes to rest.

"Exact Granular Distance at a few time-steps"

POS	***TIME***
0.00799979787476	*0.0002*
0.00799920338876	*0.0004*
0.00799821650276	*0.0006*
0.00799683721676	*0.0008*
0.00797619501476	*0.0022*
0.00797167652876	*0.0024*
0.00796676564276	*0.0026*
0.00796146235676	*0.0028*

Table 1: Exact Granular Distance at a few time-steps

"Graph for Velocity v/s Time at various time steps"

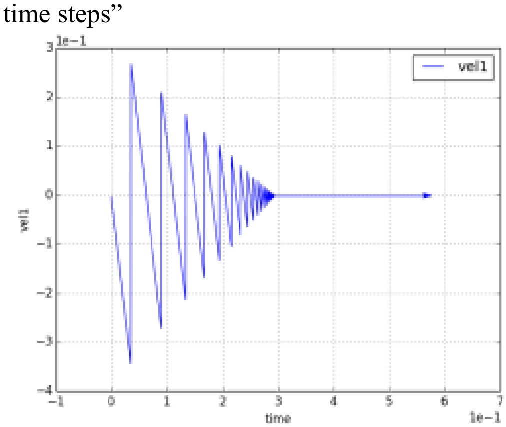


Fig. 5: Graph for Velocity with Time at few steps

The above graphical variations show the change in velocity of the free grain as it comes in contact with the fixed grain and then behaves according to the coefficient of restitution and other physical and mechanical parameters. One grain is fixed and hence, the velocity variation of the free body along one axis becomes the variation in absolute velocity of that grain under the effect of gravity. Alternate change in direction of the free sphere is obvious as the free sphere bounces back after hitting the fixed sphere until it ultimately comes to a halt after multiple time steps.

"Exact Granular Velocities at a few time-steps"

Table – 2
Exact Granular Velocities at a few time-steps

VEL	***TIME***
0.00799979787476	*0.0002*
0.00799920338876	*0.0004*
0.00799821650276	*0.0006*
0.00799683721676	*0.0008*
0.00799506553076	*0.001*
0.00797619501476	*0.0022*
0.00797167652876	*0.0024*
0.00796676564276	*0.0026*
0.00796146235676	*0.0028*
0.00795576667076	*0.003*

The velocities shown in the above table are actually for initial stages and hence, no negative velocities are being depicted but eventually, as the direction changes, the velocity becomes negative, as direction changes at each collision with the fixed sphere.

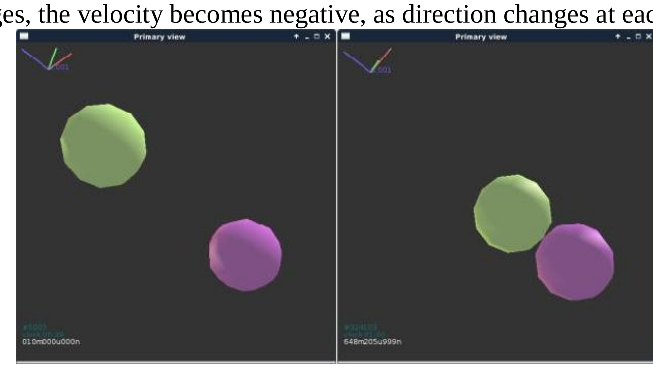


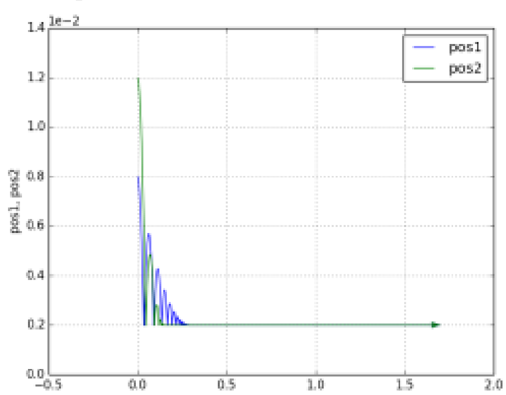
Fig. 6: Before starting simulation   After starting simulation

The pink sphere is fixed and linear 1 degree of freedom has been given to the green sphere. Simulation showed that the free sphere ultimately came to rest, due to zero bounce velocity which restricted the free sphere to bounce back further against the gravity.

2) Test for Position Evaluation At Various Time-Steps For A Double Pair Bouncing Sphere Problem

In this problem, two pairs of spheres have been placed at certain distance from each other. Each pair has a fixed sphere and a free sphere. The free sphere interacts with the fixed sphere I the effect of gravity and as an effect of which, variation in position, velocity occurs.

"Graph for Position v/s Time at various time steps"



Separate graphs for clarity--

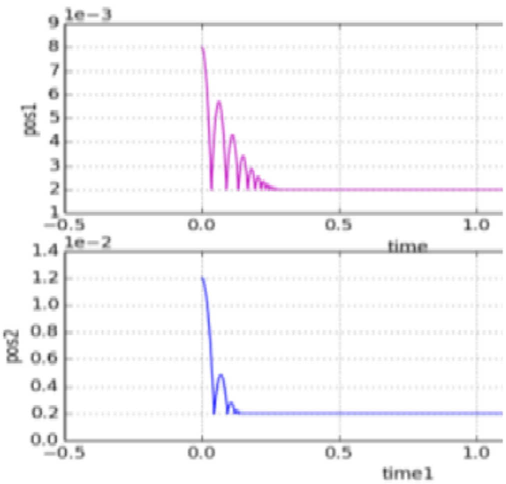


Fig. 7: Graph for Position and Time at various time steps

"Exact position of free spheres at various time steps"

Table – 3
Exact position of free spheres at various time steps

POS1	*POS2*	*TIME*
0.0079997978747	*0.011999797874*	*0.0002*
0.0079992033887	*0.011999203388*	*0.0004*
0.0079982165027	*0.011998216502*	*0.0006*
0.0079968372167	*0.011996837216*	*0.0008*
0.0079950655307	*0.011995065530*	*0.001*

"Graph for Velocity v/s Time at various time steps"

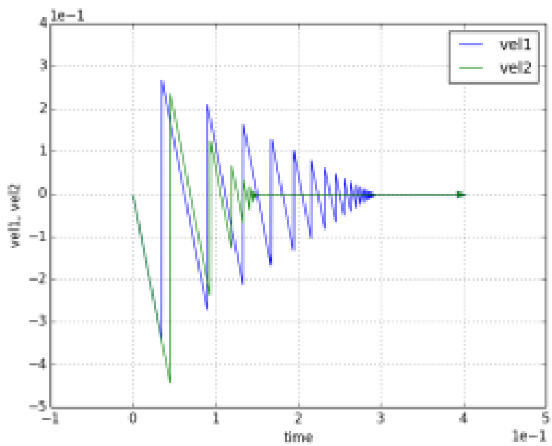


Fig. 8: Graph for Velocity v/s Time at various time steps

As obvious, the nature of graphs (both position and velocity variations) is similar to that observed in case of a single pair of sphere simulation.

"Exact velocities of free spheres at various time steps"

Table – 4
Exact velocities of free spheres at various time steps

VEL1	*VEL2*	*TIME*
-0.004	*-0.004*	*0.0002*
-0.0059	*-0.0059*	*0.0004*
-0.0079	*-0.0079*	*0.0006*
-0.0099	*-0.0099*	*0.0008*
-0.0118	*-0.0118*	*0.001*

The table shows the velocity values at the initial stages of simulation, as understood from the fact that the velocity seems to constantly increase in the negative direction, referring to the condition that collision has not occurred yet.

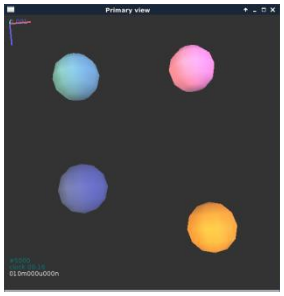


Fig. 9: Before starting simulation


After simulation

It can be noticed with ease that the sky blue sphere and the pink sphere are fixed and the dark blue sphere and the orange sphere are free. The initial and final simulation results have been shown above.

VII. "THE ROTATING DRUM"

A. Project Description:

The project aims at simulation of a large package of granular particles falling all over in a rotating drum. The idea is to simulate the phenomena like mixing of cement particles in the mixing machines at the construction sites, etc.

The cads of the drum and the grains were made in .gmsh [9] format and were imported in the python script.

1) Physical Parameters & Mechanical Parameters:

- Density=2400 Kg/m^3
- Friction angle=0°
- Contact time, Tc=0.001 seconds
- Normal coefficient of restitution, En=0.3
- Tangential coefficient of restitution Et=0.3
  - (For both spheres and the drum)
- Sphere radius=0.15 cm

2) Sphere Packing:

- Packing type: regular hexa
- Sphere gap: 0.3 cm

The material properties of both the balls are stored in a global variable, mat1 and mat2 which were called at the time of invoking the variables defining the properties of each grain.

B. Results and Interpretations:

1) Test for Position & Velocity Evaluation At Various Time-Steps For A Grain In The Package Within The Drum

"Graph for Position v/s Time at various time steps"

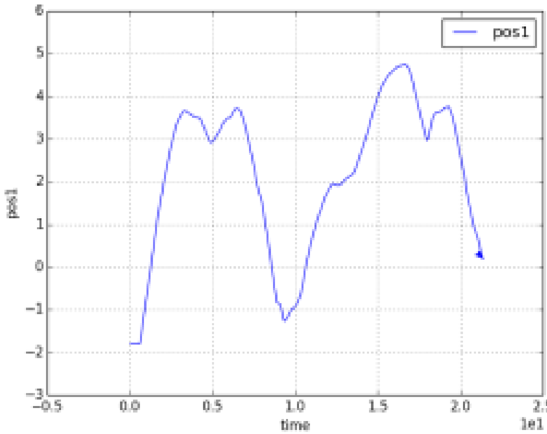


Fig. 10: "Graph for Position and Time at various time steps"

The above graphical variations show the change in position of one of the free grains in the direction of gravity as it comes in contact with the rotating drum and then behaves according to the set physical and mechanical parameters.

"Graph for Velocity v/s Time at various time steps"

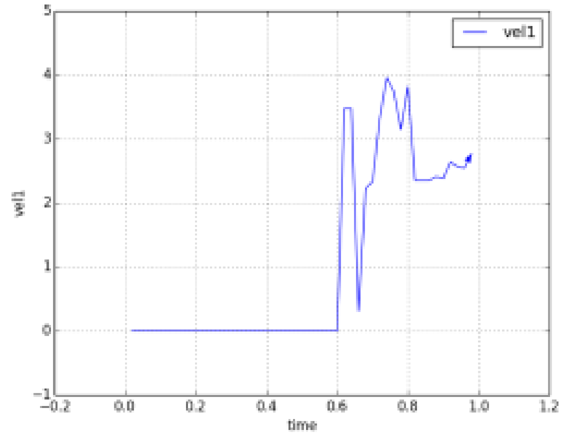


Fig. 11: "Graph for Velocity with Time at various time steps"

Figure shows changes in velocity of one of the free grains as and before it comes in contact with rotating drum. The random nature of the graph can be explained keeping in mind the fact that the grain package falls over an already rotating drum about its own axis. So initially there is a linear change in one direction alone, but as the impact occurs, motion becomes two dimensional as the motion is now governed by both the gravity and the drum surface friction. This random nature of graph becomes much clearer to analyze when the actual simulation is viewed in YADE.

"Exact Velocity of descending particle in drum"

Table – 5
Exact Velocity of descending particle at few time steps

VEL	*TIME*
3.47568321294	*0.62*
3.47568321294	*0.64*
0.312015206477	*0.66*
2.22235784945	*0.68*
3.73627733185	*0.76*
3.15222413253	*0.78*

"Graph for Torque v/s Time at various time steps"

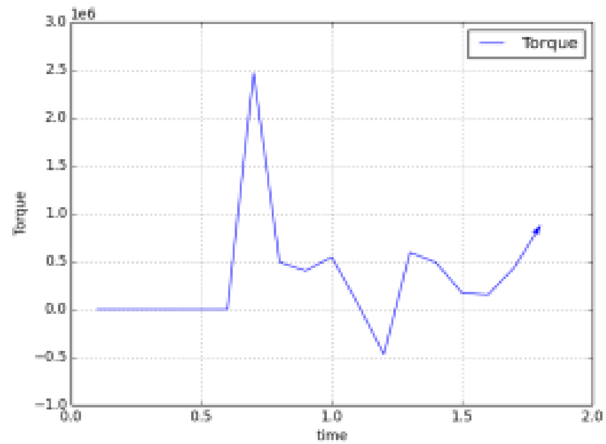


Fig. 12: Graph for Torque with Time at few time steps

Above graphical variations show the change in torque required to rotate the drum at the same velocity when subsequently the package of the free grains comes in contact with the rotating drum and then behaves according to the set physical and mechanical parameters. As expected, extra torque required to rotate the drum in the initial stages is zero until the package of spheres comes in contact with the drum. Subsequently, torque required first increases as a lot of force is required to force the particles up along the drum geometry while it rotates against gravity and as the top most point is achieved, package starts to slide along the geometry under the effect of gravity which reduces the torque required momentarily.

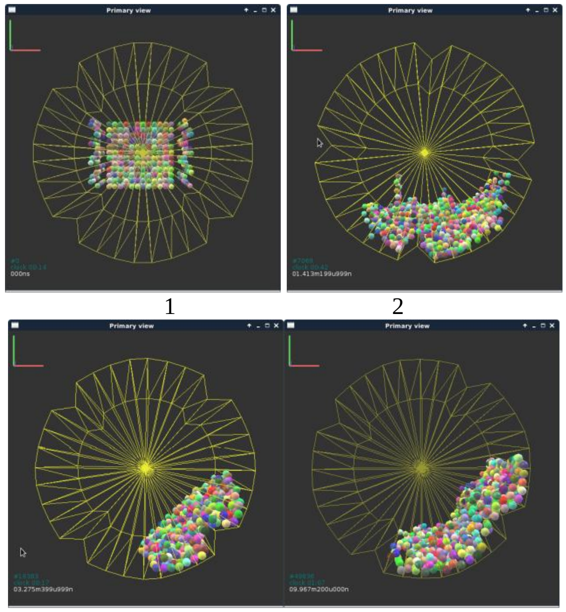


Fig. 13: Simulations during various time steps

VIII. "THE MIXER"

A. Project Description:

The project aims at simulation of a discrete flow of granular particles within a cylindrical geometry and there passage through smooth blades. The idea is to simulate the phenomena like mixing of grains in a mixer grinder, and to examine the extend of actual homogenous mixing in such kind of industrial operations.

The cads of the mixer and the grains were made in .gmsh and were imported in the python script.

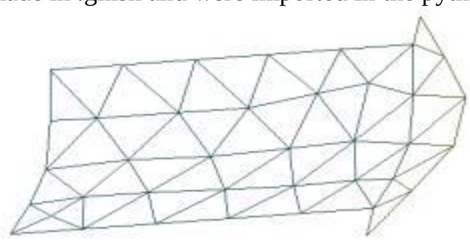


Fig. 14: Meshing of Showel

1) Physical Parameters & Mechanical Parameters:

- Density=2400 Kg/m^3
- Friction angle=35 radians
- Contact time, Tc=0.001 seconds
- Normal coefficient of restitution, En=0.3
- Tangential coefficient of restitution, Et=0.3
  - (For both spheres and the drum)
- Rotational velocity of blade=10 rpm
- No. of blades = 4

The material properties of both the balls are stored in a global variable, mat1 and mat2 which were called at the time of invoking the variables defining the properties of each grain.

B. Results and Interpretations:

1) Test For Torque Evaluation On Blades Of Mixer At Various Time-Steps

"Graph for torque v/s time at various time steps"

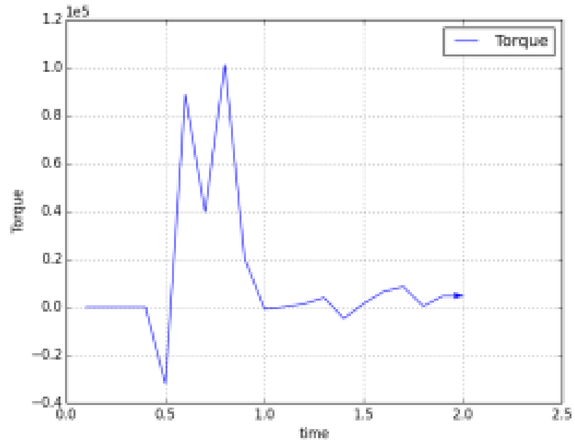


Fig. 15: Graphs for torque with time at few time steps

The above graphical variations show the change in torque applied by the blades when subsequently free grains discretely come in contact with the blades and then behaves according to the set physical and mechanical parameters. The above results were for the simulation of initial stages when the drum was getting filled. After the mixer was filled with particles, the blades were set into rotation and the homogenous mixing of the grains was simulated and analyzed.

C. Torque v/s Time for 3 different Showel Rotational Speeds---

1) @ RPM=50

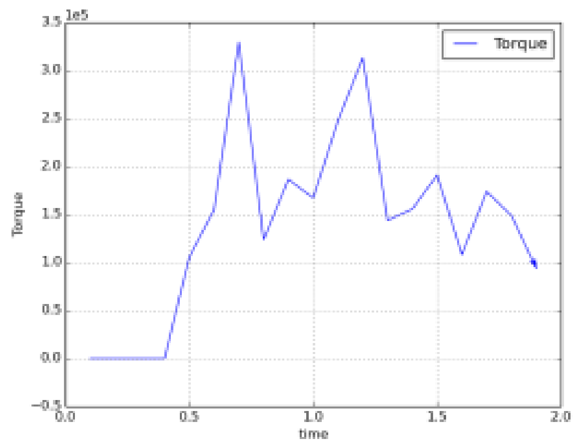


Fig. 16: Torque variation with time @ RPM=50

2) @ RPM=100

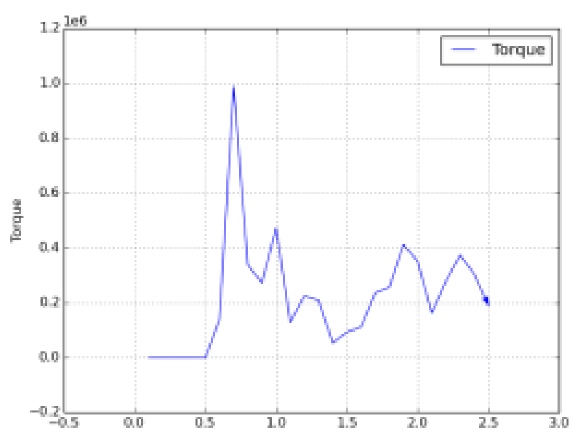


Fig. 17: Torque variation with time @ RPM=100

3) @ RPM=150

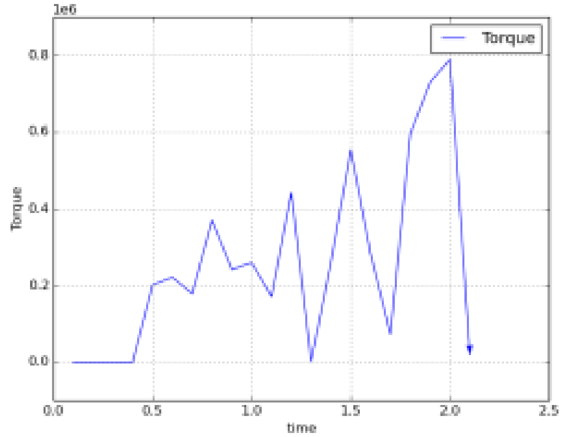


Fig. 18: Torque variation with time @ RPM=150

Changing the value of rotational speed of the showel every time, the resulted change in position as a function of time has been analyzed above. It was observed that the mixing phenomenon became unstable at higher rotational speeds of showel as that led to spilling of the particles out of the mixer. Hence the speed was restricted up to 150 rpm for simulation.

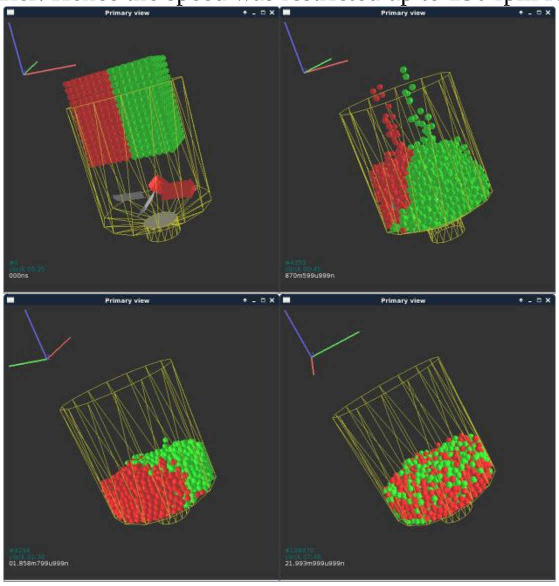


Fig. 19: Simulation steps

IX. Conclusion

The present study is a summary of the most important details about soft particle molecular dynamics (MD), widely referred to as discrete element methods (DEM) in engineering. In conclusion, discrete element methods have proven a helpful tool for the understanding of many granular systems. The qualitative approach of the early years has now developed into the attempt of a quantitative predictive modeling of the diverse modes of complex behavior in granular media. To achieve this goal will be a research challenge for the next decades, involving enhanced kinetic theories for dense collisional flows and elaborate constitutive models for quasi-static, dense systems with shear band localization.

X. Future Scope

Development of a physical DEM requires coordinated experiments and simulations to derive reasonable DEM physical parameters. Physical DEM has the potential to accurately simulate various industrial real time operations, for example, this technique is widely used to simulate concrete flow for civil applications. Hence, in order to increase the efficiency of the simulation and to reduce the bridge between the actual effects and the simulative effects, work is needed to improve the quality of physical parameters (Density, particle shape and size distribution, contact mechanics etc.). This calls upon improvement in computational codes to enhance computational speed and lower the computational costs. Hence, a lot is to be done with collaboration of experts to conduct regular micro/macro-scale tests, and interpret the results. This will lead to development of algorithms which will be computationally developed using DEM methods and simulations. At the end, there is a need to develop computational hardware and software solution to increase computational power.

ACKNOWLEDGEMENT

The above presented work on DEM is a result of extensive training sessions conducted as part of the research and intern training in YADE at Technische Universitat Bergakademie Freiberg, Germany. The compilation of the results wouldn't have been possible without the help of program mentors and guides. Author wishes to thank all those who made this project possible from the core of his heart.

REFERENCES

[1] Hi Jun Choe, Hyunseok Kim(2003), Strong Solutions of the Navier-Stokes Equations for isentropic compressible fluids. In Journal of Differential Equations, 190 (2003) 504-523
[2] Stefan Luding(2008), Introduction to Discrete Element Method. In Discrete Modelling of Geomaterials EJECE – 12/2008, pages 785-826
[3] Ante Munjiza(2004), The Combined FEM-DEM. In John Wiley & Sons, Ltd
[4] Sebastian Schmidt(2009), Thesis on Numerical Simulation of Granular Flow. In Technische Universitat Kaiserslautern
[5] B.K. Mishra, R.K. Rajamani(1992), The Discrete Element Method for the Simulation of Ball Mills In Applied Mathematical Modelling, Volume 16,Issue 11, November 1992, Pages 598-604
[6] J Kozicki and F.V Donze(2008), YADE-OPEN DEM: An open-source software using a discrete element method to simulate granular materials Page 8, Figure 5
[7] Viktor Mechtcherine, Annika Gram, Claudia Bellmann, Sergiy Shyshko(2014), Simulation of Fresh Concrete Flow using DEM In Springer 2014, ISBN 978-94-017-8883-0
[8] Official documentation at www.yade.dem.org
[9] Christophe Geuzaine, Jean-Francois Remacle(2009), Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities In International Journal for Numerical Methods in Engineering.