Rigid clumps in MercuryDPM

Igor Ostanin [1], Anthony Thornton, Thomas Weinhart, and other MercuryDPM developers

Multi-Scale Mechanics (MSM), Faculty of Engineering Technology, MESA+, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

Abstract

The paper details novel implementation of dynamics engine for rigid clumps - assemblies of spherical particles, approximating a given nonspherical shape. We detail the pre-processing algorithms, as well as the algorithms of contact detection and time integration. The capabilities of our implementation are illustrated with a number of examples.

---

1. Introduction

Rigid clumps of spherical particles is an important tool to analyse the behavior of granular materials consisting of particles of irregular shapes with the discrete element method (DEM). Commercial DEM codes provide the necessary functional to model complex-shaped particles, however, as will be demonstrated below, the implementation of rigid clumps in DEM introduces lots of ambiguities that are often hard to interpret when the source code and exact implementation details are unavailable. Our work seeks to fill this gap, presenting fully functional, well-documented and completely open source implementation of this functional.

Our paper is organized as follows. Theoretical aspects of the problem are considered in chapter I. Peculiarities of our implementation are discussed in Chapter II. Chapter III provides the set of simple validation examples, followed by the advanced study cases.

2. Theoretical Background

2.1. The notion of a rigid clump

By *rigid clump*, or just *clump* we will imply the aggregate of N rigid spherical particles of a given density, that are rigidly linked to each other at a given relative translational and rotational positions. The constituent particles of a clump will be referred to as *pebbles*. The number of non-trivial constraints that are implicitly introduced on relative translational and rotational positions of particles is $6(N-1)$. The clump is therefore a rigid body posessing only 6 degrees of freedom.

The pebbles may (or may not) have overlaps, introducing volumes within a clump that belong to more than one pebble. The number of such volumes is bound from above by $2^N - 1$. Therefore, in the general case of all different densities ρ_i of the pebbles, this may introduce up to $2^N - 1$ complex - shaped regions with distinct densities. For most of the practical applications this is an inconvenience, since the clumps are used to represent the complex-shape, constant-density particles of the same density. In this case we assume that all the regions have the same density ρ. In such a case the possibilities of computing the inertial properties of particles by analytical summation over pebbles are limited. The analytical treatment is possible in case of absent overlaps (direct summation over pebbles) and overlaps between no more than two spherical pebbles (summation over pebbles and subtraction of "cap" segments) [].

[1] Corresponding author, e-mail:i.ostanin@utwente.nl.

2.2. Equations of motion of a rigid clump

In case of spherical tensor of inertia, the rotational dynamics of a particle reduces to a set of Newton-like equations (here and below the indicial notation/Einstein convention is used, $i = x, y, z$ are the axes of the global Cartesian frame):

$$I\dot{\omega}_i = M_i \tag{1}$$

For the spherical particle of mass M and radius R, $I = 0.4MR^2$. It worth noting here that in general case the spherical tensor of inertia does not imply spherical symmetry of particle mass distribution - the dynamics of a wide class of non-symmetric rigid mass distributions can be described with (1). In the general case when the tensor of inertia is non-spherical (the principal moments of inertia are not equal) the dynamics is described by somewhat more complex set of equations:

$$I_{ii}\dot{\omega}_i - I_{ij}\dot{\omega}_j + \epsilon_{ijk}\omega_j(I_{kk}\omega_k - I_{kl}\omega_l)) = M_i \tag{2}$$

The non-spherical tensor of inertia I_{ij} is computed based on one of the algorithms discussed below.

2.3. Time integration of the EoM of a rigid clump

2.4. Computing inertial properties of a clump

2.4.1. Moment of inertia of a body bound by a triangulated surface

The center of mass of a tetrahedron j with the vertices $[a_j^1, a_j^2, a_j^3, a_j^4]$ is given by [1]:

$$c_j = \frac{a_j^1 + a_j^2 + a_j^3 + a_j^4}{4} \tag{3}$$

Volume of a tetrahedron j is given by

$$V_j = \frac{1}{6}\begin{vmatrix} (a_j^1)_x & (a_j^1)_y & (a_j^1)_z & 1 \\ (a_j^2)_x & (a_j^2)_y & (a_j^2)_z & 1 \\ (a_j^3)_x & (a_j^3)_y & (a_j^3)_z & 1 \\ (a_j^4)_x & (a_j^4)_y & (a_j^4)_z & 1 \end{vmatrix} \tag{4}$$

.

Here the volume V_j comes with the the sign that depends on the order of vertices.

Given arbitrary volume bounded by a triangulated surface Γ consisting of a set of triangles s_j, and an arbitrary point O, one can compute the center of mass of the volume as:

$$c = \sum c_j V_j \tag{5}$$

where V_j and c_j are the volume and center of mass of the tetrahedron $[a_1, a_2, a_3, a_4] = [O, s_j^1, s_j^2, s_j^3]$

One can further compute mass and tensor of inertia of the body with respect to its center of mass as the sum of masses and moments of inertia of constituent tetrahedrons:

$$\begin{aligned} M &= \rho \sum V_j \\ I &= \sum I_j \end{aligned} \tag{6}$$

The tensor of inertia of a tetrahedron is computed according to [2]:

$$I_j = \rho \begin{pmatrix} a & -b' & -c' \\ -b' & b & -a' \\ -c' & -a' & c \end{pmatrix} \tag{7}$$

where

$$a = \int_D (y^2 + z^2) dD, \qquad b = \int_D (x^2 + z^2) dD, \qquad c = \int_D (x^2 + y^2) dD, \tag{8}$$

$$a' = \int_D yz dD, \qquad b' = \int_D xz dD, \qquad c' = \int_D xy dD, \tag{9}$$

$$\tag{10}$$

where D is the tetrahedral domain. Denoting $[a_1, a_2, a_3, a_4] = [(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)]$, the integrals 8 are solved explicitly as:

$$\begin{aligned} a = V_j(&y_1^2 + y_1y_2 + y_2^2 + y_1y_3 + y_2y_3 + y_3^2 + y_1y_4 + y_2y_4 + y_3y_4 + y_4^2 \\ &+ z_1^2 + z_1z_2 + z_2^2 + z_1z_3 + z_2z_3 + z_3^2 + z_1z_4 + z_2z_4 + z_3z_4 + z_4^2)/10 \end{aligned} \tag{11}$$

$$\begin{aligned} b = V_j(&x_1^2 + x_1x_2 + x_2^2 + x_1x_3 + x_2x_3 + x_3^2 + x_1x_4 + x_2x_4 + x_3x_4 + x_4^2 \\ &+ z_1^2 + z_1z_2 + z_2^2 + z_1z_3 + z_2z_3 + z_3^2 + z_1z_4 + z_2z_4 + z_3z_4 + z_4^2)/10 \end{aligned} \tag{12}$$

$$\begin{aligned} c = V_j(&x_1^2 + x_1x_2 + x_2^2 + x_1x_3 + x_2x_3 + x_3^2 + x_1x_4 + x_2x_4 + x_3x_4 + x_4^2 \\ &+ y_1^2 + y_1y_2 + y_2^2 + y_1y_3 + y_2y_3 + y_3^2 + y_1y_4 + y_2y_4 + y_3y_4 + y_4^2)/10 \end{aligned} \tag{13}$$

$$\begin{aligned} a' = V_j(&2y_1z_1 + y_2z_1 + y_3z_1 + y_4z_1 + y_1z_2 + 2y_2z_2 + y_3z_2 + y_4z_2 + y_1z_3 \\ &+ y_2z_3 + 2y_3z_3 + y_4z_3 + y_1z_4 + y_2z_4 + y_3z_4 + 2y_4z_4)/20 \end{aligned} \tag{14}$$

$$\begin{aligned} b' = V_j(&2x_1z_1 + x_2z_1 + x_3z_1 + x_4z_1 + x_1z_2 + 2x_2z_2 + x_3z_2 + x_4z_2 + x_1z_3 \\ &+ x_2z_3 + 2x_3z_3 + x_4z_3 + x_1z_4 + x_2z_4 + x_3z_4 + 2x_4z_4)/20 \end{aligned} \tag{15}$$

$$\begin{aligned} c' = V_j(&2x_1y_1 + x_2y_1 + x_3y_1 + x_4y_1 + x_1y_2 + 2x_2y_2 + x_3y_2 + x_4y_2 + x_1y_3 \\ &+ x_2y_3 + 2x_3y_3 + x_4y_3 + x_1y_4 + x_2y_4 + x_3y_4 + 2x_4y_4)/20 \end{aligned} \tag{16}$$

Principal axes of inertia e_1, e_2, e_3 are found as eigenvectors of I:

$$Ie_i = \lambda e_i \tag{17}$$

In case of spherical tensor of inertia eigendirections can be assigned arbitrarily - we assign global Cartesian axes as eigendirections.

2.4.2. Moment of inertia of a rigid clump - summation over pebbles

This method of computation works if we the pebbles do not overlap or we presume that the inertial properties of a clump are defined by the total mass of the pebbles. In this case the total mass and tensor of inertia can be directly summed over the spherical pebbles using Steiner's theorem:

$$M = \sum m_i = \sum \frac{4}{3} \pi R_i^3 \rho_i \tag{18}$$

2.4.3. Moment of inertia of a rigid clump - voxels

2.4.4. Moment of inertia of a rigid clump - Monte-Carlo approach

2.5. approximation of a body bound by triangulated surface with the set of pebbles

3. Examples

3.1. Computation of tensors of inertia

The computation of the tensor of inertia is easy to validate, starting from the simplest analytical examples. Here we compare three different algorithms of computations of tensor of inertia, based on few basic examples. The results are summarized in the table 1.

3.2. Dynamics of a single particle - Dzhanibekov effect

3.3. Rolling of a Gomboc

Gomboc is the simply connected body that, being put on the flat surface, has one point of stable and on point of unstable equilibrium. Arbitrarily oriented at the inital moment, the Gomboc finally arrives to its stable equilibrium point. Our simulations demonstrate that this property is nicely modeled with our rigid clump model.

3.4. Domino effect

Domino effect is well know to be quite non-trivial benchmark example for DEM simulation with non-spherical particles. MercuryDPM provides a driver file designed for parameteric studies of a domino effect. Dominoes are equispaced along the straight line. At the initial moment the domino 1 is given an inital push.

3.5. bulk material of nonspherical particles

4. Conclusions

In this work we have presented the rigid clump feature of MercuryDPM particle dynamics code. For the first time, full functionality of modeling nonspherical particles as rigid clumps is provided within an open source code. The code can be downloaded at

Acknowledgments

References

[1] 2010. URL `https://people.sc.fsu.edu/~jburkardt/presentations/cg_lab_tetrahedrons.pdf`.

[2] F. Tonon. Explicit exact formulas for the 3-d tetrahedron inertia tensor in terms of its vertex coordinates.