Fifth International Conference on CFD in the Process Industries
CSIRO, Melbourne, Australia
13-15 December 2006

CLUSTER INTEGRATION METHOD FOR NON-SPHERICAL GRANULAR PARTICLES

Roar MELAND[1] , Inge R. GRAN[1], Jens A. MELHEIM[2], Svend T. MUNKEJORD[1] and Nils E. HAUGEN[1]

[1] SINTEF Energy Research, Division of Energy Processes, 7465 Trondheim, NORWAY
[2] GexCon AS, 5892 Bergen, NORWAY

ABSTRACT

A time consuming part of any discrete element method (DEM) simulation is the calculation of the forces. In conventional DEM, all particles are integrated with the same small time-step, which is due to the fact that somewhere in the system there are almost always 2 particles in contact. In a recent article, the cluster integration method (CIM) was introduced for spherical granular particles (Melheim, Computer Physics Communications 171 (2005) 155-161). Here we extend the method to non-spherical particles. We use a cell list to construct a Verlet neighbor list, in which also wall segments are included. From the neighbor list we assemble clusters of close particles. A particle can only be part of one cluster. The clusters are integrated separately during a period of time determined from the validity of the neighbor list, using a Runge-Kutta method with error control. During the integration of a cluster, we go through the neighbor lists of all particles in the cluster and calculate the forces. The basic idea is to use as long time-steps as possible for the various clusters, hence keep expensive force calculations to a minimum. After the global time-step, the clusters are rebuilt. The cluster integration method is ideal for dilute or mixed dilute/dense systems of stiff particles, e.g. as found in fluidized beds. DEM has problems for realistic values of the stiffness of the particles, giving small contact times, leading many researchers to use artificially soft particles, but with CIM realistic values can be used while still having acceptable CPU time. Here we use a simple example, a dilute system of elastic spherocylinders in vacuum, using energy conservation for validating that the integration is correct, demonstrating that CIM is more advantageous compared to DEM the stiffer the particles are.

NOMENCLATURE

A	rotation matrix
I	moment of inertia
l	length of spherocylinder shaft
N	torque
$\mathbf{n}$	normal vector
q	quaternion
$\mathbf{R}$	center of mass
r	spherocylinder "radius"
$\mathbf{u}$	axis vector
$\mathbf{v}$	velocity
Δ	buffer
δ_n	overlap
$\boldsymbol{\rho}$	arm
ρ	density
ω	angular velocity

INTRODUCTION

The discrete element method (DEM) is a powerful tool for investigating the behavior of macroscopic particles. In a DEM simulation, each particle is tracked as it moves, rotates and interacts with other particles. In a real granular system, the forces are generated by small deformations of the particles at the contact points. In a soft particle simulation, the particles are allowed to overlap slightly instead of deforming. The overlap distance is assumed to correspond to the sum of the deformations of the contacting particles.

Previously, most soft particle simulations in 3D have used round particles. The round shape is very easy to simulate. Then the moment of inertia tensor is constant also in the laboratory frame, which is a great simplification so the orientations of the particles need not to be calculated. Even more important is the fact that for spherical particles the overlap computation is simple. For a sphere, every point on its surface can be determined by knowing the position of the center and the radius of the particle; in particular, there is no dependence of the orientation of the particle. As a result, the overlap of two particles in contact is just the sum of the radii of the particles subtracted by the distance between the center of the spheres.

For dilute systems where binary encounters dominate, the dynamics of spherical particles can be calculated very efficiently by using event driven simulation and hard sphere collisions or generalizations of Enskog-DSMC to granular particles (Frezzotti 2000). In the hard sphere model the particles are assumed to interact through binary, quasi-instantaneous collisions where contact occurs at a point. The interaction forces are impulsive and therefore all other finite forces are negligible during the collision; hence linear momentum and angular momentum are conserved in the collision. Together with constitutive relations for inelasticity of the particles, the velocities and angular velocities after collisions can be given directly by algebraic relations, without solving differential equations.

However, the use of spherical particles has several limitations. Non-spherical particle shapes can create a big difference in the mechanical behavior of static systems because non-spherical particles do not roll easily. For example, bulk material composed of round particles has angles of repose that are much smaller than natural

materials. For spherical particles, particle rolling can dominate as a deformation mechanism, resulting in very low resistance to applied shear. For less rounded particles, more interlocking occurs, inhibiting the rolling tendency.

The hard sphere collision rules for non-spherical particles are very complex, if at all known, and usually it is not possible to find analytical expressions for the time to collision for a given particle pair. Hence, for non-spherical particles, it is in general not possible to use event driven simulation even for dilute systems and DEM is the only alternative. If a realistic value for the stiffness of the particles is used (i.e. high value for the normal spring), DEM is very inefficient. For a dilute system of non-spherical particles it is probably a good approximation to use artificially soft particles since the contact time is much smaller than the average time between collisions, but for flows with mixed dilute and dense regions, like in a fluidized bed, using artificially soft particles can not be justified.

CLUSTER INTEGRATION METHOD

The basic idea of the cluster integration method (CIM) of Melheim (2005) is to reduce the number of time-consuming force evaluations by using a time-step that is as large as possible. In conventional DEM, all particles are integrated with the same small time-step, which is determined by the fact that somewhere in the system there is almost always 2 particles in contact. In CIM, the particles are divided in clusters that do not interact over a given period of time. A particle can only be part of one cluster. The clusters are integrated independently using an embedded Runge-Kutta method with variable time-step, with the largest time-step possible for the desired accuracy. For a given maximum error, the largest allowed time-step for integration of the particles varies over several orders of magnitude, depending on the forces acting on the particles.

The cluster list is generated from a Verlet neighbor list (Allen 1987), which is typically generated from a cell list. The computational domain is divided into cells of equal size, where the length of the cell-sides must be equal or larger than the largest linear extent of a particle plus a buffer Δ. We assume here that the center of mass of the particle is located midway on the line measuring the largest linear extent of the particle. For each cell, a list of the particles with centroid inside the given cell is constructed. For each particle *i*, we compute the distance to all other particles in the same cell, and to all particles in the 26 neighboring cells. All neighbors in contact or with minimum distance between the surfaces less than Δ are added to the neighbor list of particle *i*, and only those neighbors are checked in force calculations for the next time-steps. The fact that the list contains pairs of particles that are close but not in contact, ensures that we also check pairs that may come in contact. Wall segments are also added to the neighbor lists for each particle, for example with a negative wall index to differentiate between wall and particle neighbors. In this implementation of the neighbor list the pairs are stored twice. When calculating the interparticle forces, we only check for contact for particle indices *i*>*j*, and use Newton's third law. The reason for the double book-keeping is that then a fast and simple recursive algorithm to find the clusters can be used, as described by Melheim (2005). Each cluster contains particles that are more than Δ away from particles in other clusters and hence the particles in a cluster can be integrated independently of the other clusters for as long as the neighbor list is valid.

In DEM it is usual to accumulate the change in position for all the particles and generate a new neighbor list when one of the particles has moved more than Δ/2. This is not possible with CIM, since the clusters are integrated one after the other for a pre-determined period of time, which can be thought of as the lifetime of the cluster list. Hence it is necessary to estimate the lifetime when the clusters are generated, using the velocity, and for non-spherical particles also the angular velocity, of the particles. The procedure for doing this depends on the particle shape.

SPHEROCYLINDERS

The cluster integration method can be used for any non-spherical shape. In this paper we will consider spherocylinders since then the contact detection is fast and simple (Vega 1994). Spherocylinders can vary from spheres to rod-like particles, depending on their aspect ratios, see Fig. 1.

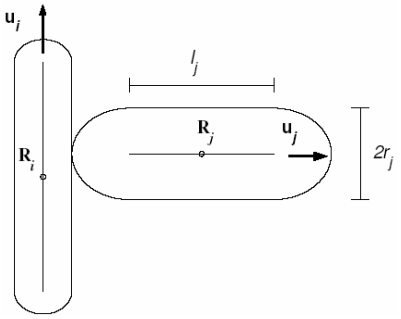

Figure 1: Two spherocylinders with center of mass R_i and R_j, "radius" r_i and r_j, and axis vector $\mathbf{u}_i$ and $\mathbf{u}_j$. The shafts are the straight lines inside the objects, with lengths l_i and l_j.

For spherocylinders, finding the overlap δ_n of two particles is equivalent to finding the closest distance δ_{ij} between the shafts of the spherocylinders, and $\delta_n = r_i + r_j - \delta_{ij}$. The contact detection algorithm for spherocylinders is given in Fig. 2.

Input : $\mathbf{R}_{ij} = \mathbf{R}_j - \mathbf{R}_i$, axis vectors $\mathbf{u}_i$ and $\mathbf{u}_j$, and shaft lengths l_i and l_j.
if $(\mathbf{u}_i \cdot \mathbf{u}_j)^2 = 1$ then
  $\lambda_{ij} = |\mathbf{u}_i \cdot \mathbf{R}_{ij}|$
  $\delta_{ij}^2 = |\mathbf{R}_{ij}|^2 - |\mathbf{u}_i \cdot \mathbf{R}_{ij}|^2 + \{\max[0, \lambda_{ij} - (l_i + l_j)/2]\}^2$
else
  $\lambda_i^{'} = [\mathbf{u}_i \cdot \mathbf{R}_{ij} - (\mathbf{u}_i \cdot \mathbf{u}_j)(\mathbf{u}_j \cdot \mathbf{R}_{ij})] \,/\, \left[1 - (\mathbf{u}_i \cdot \mathbf{u}_j)^2\right]$
  $\lambda_j^{'} = [(\mathbf{u}_i \cdot \mathbf{u}_j)(\mathbf{u}_i \cdot \mathbf{R}_{ij}) - \mathbf{u}_j \cdot \mathbf{R}_{ij}] \,/\, \left[1 - (\mathbf{u}_i \cdot \mathbf{u}_j)^2\right]$
  $\Lambda_i = \left|\lambda_i^{'}\right| - l_i/2, \quad \Lambda_j = \left|\lambda_j^{'}\right| - l_j/2$
  if $\Lambda_i \le 0$ and $\Lambda_j \le 0$ then
    $\lambda_i^* = \lambda_i^{'}, \quad \lambda_j^* = \lambda_j^{'}$
  else
    if $\Lambda_j \ge \Lambda_i$ then
      $\lambda_j^* = \text{sign}\left[l_j/2, \lambda_j^{'}\right]$
      $\lambda_i^* = \max\left[-l_i/2, \min\left[\mathbf{u}_i \cdot \mathbf{R}_{ij} + \lambda_j^* (\mathbf{u}_i \cdot \mathbf{u}_j), l_i/2\right]\right]$
    else
      $\lambda_i^* = \text{sign}\left[l_i/2, \lambda_i^{'}\right]$
      $\lambda_j^* = \max\left[-l_j/2, \min\left[-\mathbf{u}_j \cdot \mathbf{R}_{ij} + \lambda_i^* (\mathbf{u}_i \cdot \mathbf{u}_j), l_j/2\right]\right]$
    endif
  endif
  $\delta_{ij}^2 = |\mathbf{R}_{ij}|^2 + (2\mathbf{u}_j \cdot \mathbf{R}_{ij})\lambda_j^* - (2\mathbf{u}_i \cdot \mathbf{R}_{ij})\lambda_i^* - 2\mathbf{u}_i \cdot \mathbf{u}_j \lambda_i^* \lambda_j^* + (\lambda_j^*)^2 + (\lambda_i^*)^2$
endif

Figure 2: Algorithm for calculating the square of the shortest distance between the shafts of two spherocylinders.

The Vega contact detection algorithm has been used to study packing and segregation of a granular material composed of spherocylinders. Abreu et al. (2003) used Monte Carlo simulation to generate stationary configurations whereas Pournin et al. (2005) used conventional DEM.

If we neglect the deformation at the contact point, the lever arms are

$$\boldsymbol{\rho}_i = \lambda_i^* \mathbf{u}_i + r_i \mathbf{n}$$
$$\boldsymbol{\rho}_j = \lambda_j^* \mathbf{u}_j - r_j \mathbf{n}$$

where

$$\mathbf{n} = \left(\mathbf{R}_{ij} + \lambda_j^* \mathbf{u}_j - \lambda_i^* \mathbf{u}_i \right) / \left| \mathbf{R}_{ij} \right|$$

A plane wall is parameterized by the normal vector $\mathbf{n}_w$ pointing out of the wall and into the domain and a single point $\mathbf{p}$ belonging to the wall. The contact force from a plane wall acts only on one of the hemispheres. The criterion for contact is

$$d = \left(\mathbf{R}_i \pm \frac{l_i}{2} \mathbf{u}_i - \mathbf{p} \right) \cdot \mathbf{n}_w < r_i$$

The resulting overlap is $\delta_n = r_i - d$ and the lever arm is $\boldsymbol{\rho}_i = \pm l_i / 2 \mathbf{u}_i - r_i \mathbf{n}_w$.

The body-fixed system is chosen such that the axis vector $\mathbf{u}$ of the spherocylinder points in the direction of the local $\hat{\mathbf{e}}_3$ unit vector. The principal moment of inertia along the axis for a spherocylinder with constant density ρ is given by $I_3 = \frac{\pi}{2} \rho r^4 l + \frac{8}{15} \pi \rho r^5$. This is just the sum of the moment of inertia for a sphere and the moment of inertia for a cylinder, around the axis of the cylinder. From the definition, we see that

$$I_1 = I_2 = \int_{-r}^{r} \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} \int_{-l/2-\sqrt{r^2-x^2-y^2}}^{l/2+\sqrt{r^2-x^2-y^2}} \rho \left(y^2 + z^2 \right) dz dy dx =$$
$$= \frac{8}{15} \pi \rho r^5 + \frac{3}{4} \pi \rho r^4 l + \frac{1}{3} \pi \rho r^3 l^2 + \frac{1}{12} \pi \rho r^2 l^3$$

Alternatively, we could get the same answer by using the fact that the center of mass for a hemisphere is displaced $3r/8$ along the symmetry axis, from the center of the corresponding sphere, and using the parallel axis theorem (Goldstein 1980) twice.

For spherocylinders, it is easy to find the lifetime of the clusters. The maximum velocity along the shaft of a spherocylinder occurs at the ends, i.e. at the hemisphere centers. The cluster lifetime is then given by $0.5\Delta / \max \left(\mathbf{v}_{cm} \pm l/2 \boldsymbol{\omega} \times \mathbf{u} \right)$.

TEST SIMULATIONS

For dense systems, all particles are in a few clusters and the CPU time of CIM is comparable to conventional DEM with a Verlet neighbor list. The advantage of CIM over DEM is evident for dilute systems, or systems with mixed dilute and dense regions, and particularly for stiff particles. To show this more quantitatively, it is important that we compare CIM and DEM simulations that integrate the particles with the same accuracy. The type particle interaction is immaterial when comparing the speed of DEM and CIM, since only the contact time matters. A convenient way of measuring the accuracy of particle simulations is to turn off all dissipation and check for energy conservation. Alternatively we could have considered the inelastic collision of two particles and compared with an "exact" solution calculated with very small time step, but we use the energy conservation method since we then get a comparison of the relative speed of DEM and CIM simultaneously.

For simplicity, we consider a dilute system of elastic spherocylinders in zero-gravity vacuum, with periodic boundary conditions. In a real-life application, it is the simulation of the particle-particle interactions in the bulk material that consumes most of the simulation time, and with the periodic boundary conditions, our simulation mimics a small homogenous region in the bulk. The only force acting between two particles in contact is a repulsive spring with spring constant k_n. The simulation is conducted in a cubical box with sides 5cm, filled with 746 particles with shaft 4mm and "radius" 1mm (i.e. aspect ratio 3). This gives a volume fraction of 0.1. The density of the particles is 2700kg/m³. The initial configuration is created by "equilibrating" the system; the particles are integrated until equipartition between the rotational and the translational degrees of freedom is achieved, while scaling the kinetic energy to the desired value. We set the initial rotational velocity along the shaft of the particles to zero, and since there is no tangential force that can impart rotation around the shaft axis, this freedom is "frozen", and there are effectively 2 rotational degrees of freedom and 3 translational.

The average kinetic energy of a particle is set to $6.13 \cdot 10^{-6}$ J in all simulations, corresponding to an average particle speed of 0.4m/s, assuming equipartition between the translational and rotational degrees of freedom. The same buffer Δ=0.6mm is used for the neighbor list in the CIM and DEM simulations. For the DEM simulation the classical Runge-Kutta of order 4 (RK4), whereas for the CIM simulation the adaptive time step Runge-Kutta-Fehlberg is used with relative tolerance 10^{-5} and absolute tolerance 10^{-20}. When the forces are large, it turns out that the adaptive integrator gives more restrictive time steps than necessary for keeping the energy constant, and this reduces the computational speed. Hence we specify a minimum time-step $dt_{\min}$ allowed for the RK-Fehlberg method. In the test simulations, for a given k_n, the constant DEM time-step dt_{RK4} and the minimum time-step $dt_{\min}$ for RK-Fehlberg are tuned so that the energy is conserved with less than 0.1% error after 0.1s of physical time. The computations have been carried out on a PC with an AMD Athlon 2.0GHz processor. The speed-up of CIM relative to DEM for the case of no dissipation is shown in Table 1. With dissipation, i.e. a linear dashpot, the speed-up will be a little less, since the dampening leads to somewhat longer contact times.

$k_n\,[N/m]$	$dt_{RK4}[10^{-5}\mathrm{s}]$	RK4[s]	$dt_{\min}[10^{-5}\mathrm{s}]$	RK-F.[s]	speed-up
5000	4.6	21	3.3	17	1.2
10000	3.0	29	2.6	17	1.7
20000	2.3	39	1.6	17	2.3
30000	1.7	53	1.4	20	2.7
40000	1.5	60	1.2	20	3.
60000	1.2	86	1.0	20	4.3
80000	1.1	89	0.93	18	4.9
100000	1.	99	0.77	20	4.95
150000	0.81	128	0.60	20	6.4
200000	0.7	149	0.52	21	7.1
500000	0.45	241	0.34	21	11.5

Table 1: Simulation times for DEM and CIM.

For the dilute system considered, the CPU time for CIM does not increase much when the stiffness of the spherocylinders increase, whereas for DEM the constant time step must be decreased to ensure energy conservation during contacts, increasing the number of force evaluations and hence the CPU time. For very stiff elastic spherocylinders, e.g. steel particles, CIM is an order of magnitude faster than DEM for dilute systems of non-spherical particles. We also see that the minimum time-step allowed for RK-Fehlberg is smaller than the corresponding constant time step for RK4. This makes sense since the adaptive integrator tries to distribute the error in time by taking longer time steps when the forces are small, whereas the constant time step RK4 is very accurate for small forces and hence can use a little longer time step than RK-Fehlberg when the forces are large.

CONCLUSION

In conclusion, the described CIM algorithm makes it possible to simulate the dynamical behavior of non-spherical granular particles in dilute systems with realistic values for the stiffness of the particles, much faster than with conventional DEM. For dense systems CIM and DEM give almost equal simulation time, since then all particles are in a few clusters, and both CIM and DEM essentially use a Verlet neighbor list for optimization. It is expected that CIM will be considerably faster than DEM in mixed dilute/dense systems, like a fluidized bed, since the particles in the dilute regions can be integrated with longer time steps than with conventional DEM.

REFERENCES

ABREAU, C.R.A., TAVARES, F.W. and CASTIER, M., (2003), "Influence of particle shape on the packing and on the segregation of spherocylinders via Monte Carlo simulations", *Powder Tech.* **134,** 167-180.

ALLEN, M.P. and TILDESLY, D.J., (1987), "Computer simulation of liquids", Oxford Science Publications, Oxford.

FREZZOTTI, A, (2000), "Monte Carlo simulation of the uniform shear flow in a dense rough sphere fluid", *Physica A*, **228**, 161-180.

GOLDSTEIN, H., (1980) "Classical mechanics", Addison Wesley, Reading.

MELHEIM, J. A., (2005), "Cluster integration method in Lagrangian particle dynamics", *Computer Physics Communications*, **171**, 155-161.

POURNIN, L. , WEBER, M., TSUKAHARA, M., FERREZ, J.A. , RAMAIOLI, M. and LIEBLING, T.M., (2005), "Three-dimensional distinct element simulation of spherocylinder crystallization", *Granular Matter* **7,** 119-126.

VEGA, C. and LAGO, S, (1994), "A fast algorithm to evaluate the shortest distance between rods", *Computers and Chemistry*, **18**, 55-59.

APPENDIX A

In this work quaternions have been used to represent the orientation of each non-spherical particle, since the Euler angles are awkward for numerical calculations. The quaternions for each particle satisfy the following equations of motion (Allen 1987)

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_1^p \\ \omega_2^p \\ \omega_3^p \end{bmatrix}$$

where the superscript *p* means that the angular momentum components are taken relative to the body-fixed principal axis system.

The angular velocity is given by Euler's equations for the motion of a rigid body (Goldstein 1980)

$$\begin{aligned} N_1^p &= I_1 \dot{\omega}_1^p - \omega_2^p \omega_3^p (I_2 - I_3) \\ N_2^p &= I_2 \dot{\omega}_2^p - \omega_1^p \omega_3^p (I_3 - I_1) \\ N_3^p &= I_3 \dot{\omega}_3^p - \omega_1^p \omega_2^p (I_1 - I_2) \end{aligned}$$

Here I_1, I_2 and I_3 are the principal moments of inertia, i.e. the diagonal components of the inertia tensor in a body-fixed principal axis system. Since the torques are normally calculated in the laboratory system, we need to transform back and forth from the principal axis system. The transformation matrix relating components of the same vector in the principal and laboratory axes are (Allen 1987)

$$A = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

such that $[\mathbf{N}]^p = A[\mathbf{N}]^{lab}$. Here $[\mathbf{N}]^p$ means the coordinate vector of the torque **N** in the principal axes system. To calculate forces and torques we need the angular velocity in the laboratory system. We have

$$[\boldsymbol{\omega}]^{lab} = A^{-1}[\boldsymbol{\omega}]^p = A^t[\boldsymbol{\omega}]^p$$

since the transformation matrix from one orthonormal coordinate system to another is orthogonal, and the inverse of the transformation matrix is just the transpose, $A^{-1} = A^t$.