Mathematical Modelling and Analysis
Volume 6 Number 1, 2001, pages 156–164

DISCRETE ELEMENT MODELS OF PARTICLE FLOWS

R. WAIT

Uppsala University

Department of Scientific Computing

E-mail: richard.wait@tdb.uu.se

Received November 2, 2000

ABSTRACT

Particle flows occur in a very wide range of industrial applications. In a Discrete element model (DEM) the motions of the individual particles in the system are tracked. The model simulates all the interactions between the particles and their environment. The goal is to predict the macro-properties of the flow and the modelling of the separate particles is only as accurate as required for this global aim. It is necessary to simulate particles of many different sizes and densities interacting with complex shaped surfaces.

1. INTRODUCTION

The discrete element method (DEM) has been developed to simulate flows in which the motion has discontinuities that are not modelled well as a continuous fluid. For example because the particles are too big, the particle density is too low or the frictional effects in the particle contacts are too large. Since the DEM models flow at the particle level, a realistic representation requires the use of high performance computers (HPC). The modelling of particle flows has many industrial applications including grinding and crushing [1], silos and bins [8] and conveyors in a variety of areas. Recent applications also include the simulation of crowd evacuation from burning aircraft and buildings[10].

The DEM dates from the 1970's [3] and early models were two-dimensional and assumed that all particles were circular. There is usually a need for a high density of particles in an effective simulation, so that the efficient use of HPC is essential. It is only recently that the availability of sufficient computing power has led to the implementation of 3-dimensional models that incorporate more complex particle shapes. The shape of the model particles does not have to

be a highly accurate representation of the real particle geometry, the only requirement is that it induces the appropriate model dynamics. Although the goal of the DEM model is the simulation of the gross behaviour of the flow, it is the interaction of the individual particles that drives the model. Super-ellipsoids (or superquadratics [13], [5]), given by:

$$\left(\frac{x}{a}\right)^m + \left(\frac{y}{b}\right)^m + \left(\frac{z}{c}\right)^n - 1 = 0 \tag{1.1}$$

can assume a variety of axi-symmetric shapes by varying the principle axes a, b and c and the sharpness of the edges determined by the indices m and n, usually $m = n$. Most of the computational effort is spent in resolving the contact between particles and moving from circular to non-circular particles represents a dramatic increase in the complexity of this aspect of the simulation. In the case $m \neq n$ a more convenient form is

$$\left(\left(\frac{x}{a}\right)^n + \left(\frac{y}{b}\right)^n\right)^{m/n} + \left(\frac{z}{c}\right)^m - 1 = 0 \tag{1.2}$$

as this has a parametric representation with $p = \dfrac{2}{n}$ and $q = \dfrac{2}{m}$ in the form

$$x = a\sin(\phi)^q\cos(\psi)^p \quad y = b\sin(\phi)^q\sin(\psi)^p \quad z = c\cos(\phi)^q \tag{1.3}$$

A number of other alternative shape representations have been suggested with a view to simplifying the contact resolution[4].

2. COLLISION DETECTION

A vital aspect of discrete element modelling is the detection of particle collisions. For large scale industrial modelling such detections must be fast and efficient. In rigid body mechanics, the particles move through space and the collisions are instantaneous and do not occur at regular intervals. A soft particle model allows particles to overlap and uses the size of the overlap (typically <1.0% overlaps are desirable) to determine the forces acting on the colliding particles. These forces are then applied to define the resultant motion of the particles. The model can then be integrated in regular time steps, typically between 20 and 50 time steps are required to accurately integrate each collision. In hard particle models, no overlap is possible, thus when an overlap is detected the simulation must back-track to the instant of first contact (which is different for each pairwise contact), compute the forces on impact and then rerun the simulation for the colliding particles from the moment of impact to the end of the time step (assuming no further contacts).

For particles defined by a boundary surface $f(x) = 0$, such as (1.1) or (1.2) the contact problem reduces to root finding. If the two particles are defined by the surfaces

$$f(x) = 0 \qquad \text{and} \qquad g(x) = 0$$

then for each particle we can define *concentric* surfaces $f(x) = c$ and $g(x) = c$, if $c < 0$ these are internal to the particle and if $c > 0$ they are external. We can identify a point *halfway* between the two surfaces such that

$$f(x) = c = g(x).$$

As the two surfaces are tangential at the common point, the normals are in opposite directions. Thus we solve (using Newton-Raphson) the system

$$\nabla f + \alpha^2 \nabla g = 0, \quad f - g = 0 \tag{2.1}$$

for the coordinates (x, y, z) and the scaling factor α. If the solution is $f = g > 0$ then the particles have not collided yet, if the solution is $f = g < 0$ then there is interpenetration and the solution gives the common normal at a point between the two surfaces (the point of the collision) see figure 1, from which it is possible to determine the depth of the penetration. The figure shows a simple example of the collision of two 3-D particles moving and rotating only in the xy-plane. (The overlap illustrated is much larger than would be acceptable in a practical simulation and gravity has not been included.) In the figures, the feint dashed lines indicate the concentric superellipsoids used to compute the contact normal and the asterisks indicate the "depth" of overlap in the normal direction, or are the the "closest" points for nonoverlapping particles. If it is then assumed that the maximum penetration is in the direction of the common normal, the calculation of the depth of penetration involves two more nonlinear equations to be solved (separately) by Newton-Raphson. The total computational effort is then solve a system of 4 simultaneous nonlinear equations (2.1) and (if there is contact) two single nonlinear equations. An alternative formulation [7] is to compute the solution of

$$\min g(x) \quad \text{such that} \quad f(x) = 0 \tag{2.2}$$

for one contact point, and

$$\min f(x) \quad \text{such that} \quad g(x) = 0 \tag{2.3}$$

as the other. If we use the parameter α^2 as a Lagrange multiplier, then computing each point involves solving a slightly different nonlinear system namely,

$$\nabla f + \alpha^2 \nabla g = 0 \quad g = 0$$

for problem (2.2) and

$$\nabla g + \alpha^2 \nabla f = 0 \quad f = 0$$

for problem (2.3).

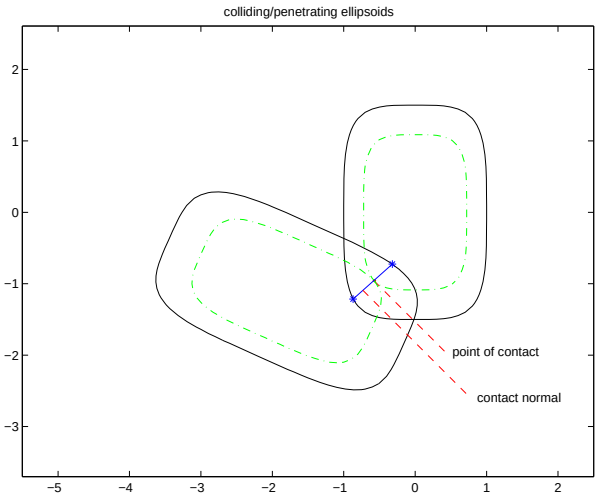

Figure 1. Contact surface

In numerical experiments with this form, as in figure 2, the line joining the contact points is frequently significantly skewed from what would be a reasonable *normal* direction and this has an effect on the subsequent resolution of the forces. The simulation of particle-particle collisions, together with the particle-solid collisions at the boundary, is an obvious candidate for parallel computing as it is an application of the type often characterized as *embarrassingly parallel*. It is however vital to have an efficient collision detection algorithm that scales as $\mathcal{O}(N)$ with N particles. A simple search of all combinations will scale as $\mathcal{O}(N^2)$ and is completely impractical for large scale simulations. A scheme where the particles have address tags works well [8] and is suitable for a parallel implementation. The size of the cells in the partition is comparable to the size of the particles so that good initial approximations can be defined so that the Newton-Raphson iteration will converge rapidly to a contact point.

Partitioning the domain so that it is only necessary to check for contact between near neighbors reduces the number of possible contacts, but the number of actual contacts will only be a fraction of the number possible. Unfortunately, when the particles are highly non-spherical, then a computational problem suffered by both methods of contact detection is the non convergence for near misses when contact does not take place. A more efficient method is to use the parametric definition of the surfaces (1.3) and for each pair of neighbouring particles to define a mesh on one particle. If all the points of

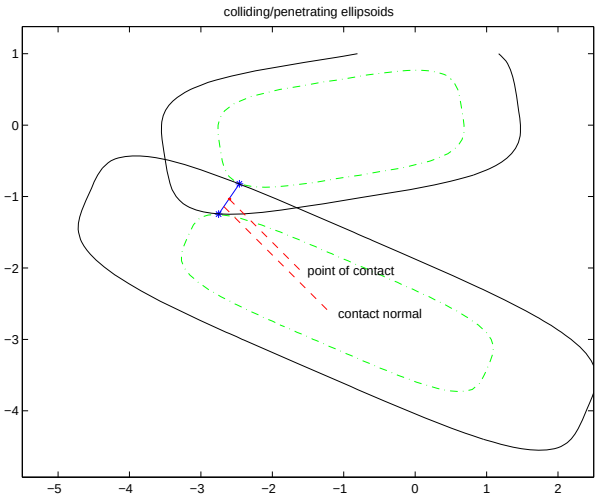

Figure 2. Maximum depth penetration

the mesh lie outside the other particle, checked using (1.2), then no contact is possible, otherwise the first point inside is taken as the initial approximation for the Newton-Raphson iteration. Another alternative that avoids the need for any non linear solver is to use a mesh search to find the deepest penetration and hence solve (2.2) then reverse the roles of the surfaces and perform a second mesh search to solve (2.3).

3. CONTACT FORCES

Once the depth of penetration and the normal direction have been computed the forces can be computed.

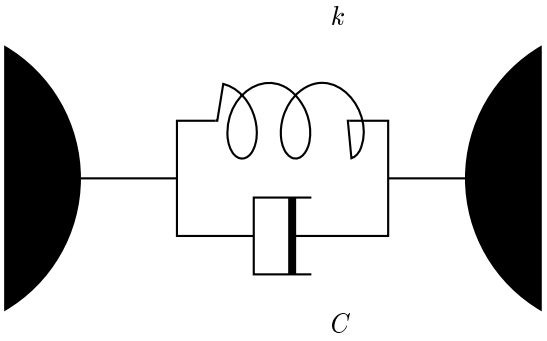

Figure 3. Collision Model

The *soft particle* model uses a simple spring and dash pot (see figure 3) to

estimate the collision forces. The normal force is

$$F_n = -k\Delta x + Cv_n \tag{3.1}$$

and the tangential force is

$$F_t = \min\{\mu F_n, Cv_t\} \tag{3.2}$$

where:

- Δx is the particle overlap,
- k and C are the constants for the spring and dash pot damping respectively,
- v_n and v_t are the normal and tangential velocities,
- μ is the friction coefficient.

4. COLLISION RESOLUTION

Given the forces $F_i = \sum_j F_{ij}$ acting on a particle i colliding with particle j, it is possible to integrate the equations of motion

$$\left.\begin{aligned} m_i\ddot{x}_i &= F_i + f_i \\ I_i\ddot{\theta}_i &= \sum_j a_{ij} \times F_{ij} \end{aligned}\right\} \qquad i = 1, \ldots, N$$

where a_{ij} is the vector from the centroid of particle i to the point of the collision with particle j. The external force on particle i is denoted by f_i, this may be gravitational in a mill simulation, or steam pressure in a wood-pulp refiner or pressure towards an emergency exit in a crowd simulation.

5. APPLICATIONS TO THERMOMECHANICAL PULPING

Many stages of the pulp processing within a modern paper mill have been simulated using mathematical modes, often in terms of finite element models of the fluid flow, e.g. [11]. A discrete element model is appropriate for a simulation of the pulp refiner where the flow, at least at input, cannot realistically be modelled as a fluid. The input to the refiner is in the form of wood chips forced by high pressure steam, the output is wood pulp. Between the input and output, the three dimensional wood chips are effectively reduced to one-dimensional cellulose fibers. Pulp is produced by grinding wood chips in a refiner that has two grinding discs with patterned surfaces. The chips are broken by impact with the bars. The chips are forced between the discs by

the steam pressure. The disks typically run at 2000rpm and the density of wood fiber is very high, i.e. there are a large number of wood chips entering the refiner. There is typically stagnant pulp partially filling the stationary grooves.

A realistic model of the gross behaviour within the refining process should be able to model meaningful parameters such as the energy consumption and the distribution of fiber length in the output pulp (a measure of the quality).

Successful DEM simulations for a typical ball mill (with stone chips) running at 700rpm with 30k particles have been reported [2], where it is quoted that 1 second of simulated time that may take a few hours of CPU time on a workstation. The simulations have been used to estimate the power consumption. Grinding mills typically have poor energy efficiency (1-5%) which is of the same order of efficiency as in thermomechanical pulping.

Existing mechanistic models of pulp refiners, such as [6] that assume axial symmetry of the refiner disc and approximate the pulp densities as continuous functions of r and t. They are able to forecast the motor load and temperature as functions of the varying input. The motor load can be determined from the forces of the collisions on the refiner plates. The efficiency with the chips are broken into fibers is highly correlated to the pattern of bars on the discs but such a simple model is unable to simulate the effects of these non axi-symmetric patterns. The particles to be modelled are of very hot, very wet wood, the spring and damper constants in the DEM model need to be chosen to represent the properties of such wood sufficiently accurately. There is an ongoing research project [12] funded by the Swedish National Graduate School in Scientific Computing (NGSSC) which should provide the data required for accurate simulation. The distribution of fiber length can only be simulated if there is a model of the particle fracture processes. The particle can break in two distinct ways:

- A series of small impacts will cause a particle to split along the fiber, thus preserving the fiber length. One of the axis of the particles is identified as the directions of the fibers and the particle is replaced by several smaller particles by partitioning along the other two axes of the particle.

- A single large impact will break the particle in a direction that is not necessarily aligned with the fibers and hence, in general, reduces the fiber length. The fracturing can be viewed as chopping along the particle axis that is aligned nearest to the direction of the impact force.

So in order to model both forms of fracturing, it is necessary *inter alia*, to estimate the work done by successive small impacts. In any fracturing, there is a loss of soluble (non cellulose) mass of the particle that must also be modelled.

6. CONCLUSIONS

The discrete element method is an effective technique for modelling granular flows when the aim is to model the global properties but the mechanics is driven by the friction effects at the particle level. A realistic model is very compute-intensive as it incorporates several thousand particles and this is now possible given the availability of modern high performance computers.

REFERENCES

[1] P.W. Cleary. Industrial particle flow modelling: Discrete element methods and their applications. In: *D.Davies, editor,* Proc. Chemeca 97, 1–12, 1997.

[2] P.W. Cleary. Predicting charge motion, power draw, segregation and wear in ball mills using discrete element methods. Minerals Engng, **11:1061–1080,** 1998.

[3] P.A. Cundall and O.D.L. Strack. A discrete numerical model for granular assemblies. Geotechnique, **29:47–65** 1979.

[4] J.F. Favier, M.H. Abbaspour-Fard, M. Kremmer and A.O. Raji. Shape representation of axi-symmetrical, non-spherical particles in discrete element simulation using multi-element model particles. Engng. Computations, **16:467–480,** 1999.

[5] C. Hogue. Shape representation an contact detection for discrete element simulations of arbitrary geometries. Engng. Computations, **15:372–390,** 1998.

[6] A. Horch, A.J. Isaksson, B.J. Alison, A Karlström and L. Nilsson. Dynamical simulation of a thermomechanical pulp refiner. Nordic Pulp and Paper Res. Jnl., **12:270–275,** 1997.

[7] X. Lin and T.-T. Ng. Contact detection algorithms for three-dimensional ellipsoids in discrete element modelling. Int. Jnl. Num. and Anal. Methods in Geomechanics, **19:653–659,** 1995.

[8] B. Mathews. An improved discrete element algorithm for particle flow problems. *Master's thesis.* Colorado School of Mines, Geomechanics Research Center, 1993.

[9] R. O'Connor, J.J. Gill, and J.R. Williams. A linear complexity contact detection algorithm for multi-body simulation. In: *J.R Williams and G.W. Mustoe, editors,* Proc. 2nd Int. Conf. on Discrete Element Methods, *IESL Publications, Cambridge, Mass,* 53–63, 1993.

[10] C. Robbins. Computer Simulation of Crowd Behaviour and Evacuation. ECMI Newsletter, **25,** 1999.

[11] P.Tarvainen, R.A.E. Mäkinen and J. Hämäläinen. Shape Optimization for Laminar and Turbulent Flows with Applications to Geometry Design of Paper Machine Headboxes. In: *Hafez and Heinrich, editors,* The 10th Int. Conf. Finite Elements in Fluids, 536–541, 1998.

[12] S. Widehammar and P. Gradin. Mechanical defibration of wood. http://www.tdb.uu.se/∼ngssc/ngssc_home/projects/proj.html 1998

[13] J.R. Williams and A.P. Pentland. Superquadratics and modal dynamics for discrete elements in interactive design. Engng. Computations, **9:115–127,** 1992.

DALELIŲ JUDĖJIMO DISKREČIŲJŲ ELEMENTŲ MODELIS

R. Wait

Daugelyje industrinių taikymų nagrinėjamas dalelių judėjimas. Ši metodika remiasi visų dalelių sąveikos modeliavimu. Darbe keliamas tikslas ištirti ir prognozuoti tekėjimo makrosavybes, remiantis šiuo tikslu parenkamas modelio tikslumas. Kiekvieno eksperimento metu modeliuojamas labai didelio skaičiaus įvairių dydžių dalelių judėjimas, ištirta tokių dalelių sąveika su sudėtigos formos paviršiais. Aptartos lygiagrečiųjų algoritmų panaudojimo galimybės.