Investigation of Efficient Geometric Shape Algorithms for Numerical Simulation of Discrete Particle Systems

by

Scott M. Johnson

B.S. Civil Engineering

Washington University in St. Louis, 2001

Submitted to the Department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Civil and Environment Engineering

at the

Massachusetts Institute of Technology

June 2003

Signature of Author: ______________________ ______________
Department of Civil and Environmental Engineering
May 9, 2003

Certified by: ______________________________
John R. Williams
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by: ______________________________
Oral Buyokozturk
Professor of Civil and Environmental Engineering
Chairman, Committee for Graduate Studies

Investigation of Efficient Geometric Shape Algorithms for Numerical Simulation of Discrete Particle Systems

by

Scott M. Johnson

Submitted to the Department of Civil and Environmental Engineering
on May 9, 2003 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Civil and Environment Engineering

ABSTRACT

The efficiency of a discrete particle simulation implementation relies on several factors, including the geometric representation, contact resolution, and neighbor-sorting algorithm used. The focus here is on the first two of this list: the geometric representation and the corresponding contact resolution. An argument is developed to advocate the use of geometric representations with inconstant radius based on results of a numerical study of angles of repose for deposited grains. The equivalent spheres method is then developed as a potential cure for the problems arising from constant radius geometric representations. A coherent approach to utilizing this geometry in discrete element modeling is developed. This includes the proposal of an accurate, robust contact resolution algorithm and explicit functions to describe moments of inertia. Considerations for numerical modeling are also addressed, including numerical integration, formulation of rotation transformations, and resolution of forces and motions in the context of rigid body motions. The details of a generalized computational implementation of the representation are also given, and empirical convergence properties are compared with a different method for detecting contact between ellipsoidal approximations.

Thesis Supervisor: John R. Williams
Title: Professor of Civil and Environmental Engineering

Table of Contents

List of Figures

Introduction

The capability of discrete element modeling (DEM) to model physics at the grain scale has made it a leading computational method for modeling complex phenomena exhibited by particulate systems in applications as diverse as the manufacture of pharmaceuticals and borehole flow simulation in the petrochemical industry. Three-dimensional DEM simulation at scales on the order of typically performed field tests requires computational resources (in terms of CPU time and memory) that are far in excess of those available to most researchers. These requirements are highly dependent on the geometrical representation, contact resolution, and the contact detection algorithm used in the implementation. This thesis will address the first two topics in this list.

There are advantages and disadvantages to every geometrical representation that makes it amenable to a certain application or not. Several available geometries that have been proposed for modeling of discrete body systems will be compared for their relative merits. The goal of this thesis is to identify a suitable 3-D geometry for use in modeling large numbers of fluid-solid interactions in fluidized sand beds. A recently proposed ellipsoid approximating geometry will be developed and redefined to view the geometry as an assembly of constrained spheres – here called the Equivalent Spheres Method. The equivalent spheres ellipsoid is shown to be suitable, in terms of computational resource requirements and geometric compactness, for use in discrete element modeling. The specific geometric properties of the geometry, including moments of inertia and explicit surface description, for the equivalent spheres ellipsoid is developed, and a formulation of force resolution on such a shape is also given. This also includes proposing a fast, robust contact resolution algorithm that compares favorably with the memory and CPU requirements of other contact resolution algorithms for ellipsoidal representations. Specifically, this proposed contact resolution algorithm is shown to exhibit promising results versus another algorithm in offering faster, more accurate identification of the point of contact between two ellipsoidal geometries. The development of the contact resolution here also decouples the algorithm used for ellipsoid-ellipsoid contact from that used for ellipsoid-sphere and ellipsoid-polyhedron contact, which has the effect of streamlining the algorithm.

Sphere-based DEM implementations remain the most common for 3-D modeling, since they are simple and compact and hence are fast computationally. This is the basis for the popular codes TRUBAL (Cundall, 1987), PFC3D (Cundall, 1996), and DMC (Taylor and Preece, 1992) as well as the application developed by Cleary (1998). Spheres, however, have several drawbacks that hinder their ability to capture key behaviors of real particle systems that contain non-spherical particles. Spheres tend to roll excessively when subjected to small perturbations because the normal forces at contact always pass through the centroid and cannot exert a resistive couple on the body. Some numerical implementations address this shortfall by introducing couples artificially by either perturbing the normal force direction so that it does not pass through the centroid or by adding a "rolling friction" factor (Vu-Quoc, 1999).

Spheres are unable to capture interlocking phenomena exhibited by real granular systems. They also tend to organize into dense, ordered packings as the coefficient of friction decreases (Grest, 2001), which is not representative of most naturally-occurring materials. In highly dynamic systems, such as seismically excited soils and mixed particulate matter found in the pharmaceutical and mining industries, the particles tend to segregate into mono-sized regions due to free surface segregation (in surface cascades), interparticle percolation (in failure zones), and particle migration (high strain rate regions), according to Bridgewater (2002). Regions of mono-sized spheres are especially vulnerable to organization into densely packed regions.

Some of these problems have been overcome by clustering a number spheres together to form a non-spherical shape. Some implementations (Favier 1999, Vu-Quoc 2000, Mustoe 2001) represent ellipsoids in this manner. Typically the discretization is coarse (7 spheres or less) in order to achieve fast execution times. However, the clustering is only C^0 continuous and, therefore, introduces discontinuous surface normals that lead to computational problems especially in contact resolution with polyhedral elements. Favier (2001) also notes that frictional behavior induced by the C^0 continuity of the surface can reach as high as 0.4 between such clustered sphere particles, restricting the types of materials such particles can accurately represent.

Analytical representations of 3-D ellipsoids have been proposed and implemented by Ting (1993), by Ng and Fang (1995), and by Lin and Ng (1997). More generalized representations using superquadrics for DEM are proposed in Williams and Pentland (1992). A compact approach to representing and performing contact detection between discretizations of arbitrarily-shaped particles (discrete function representation) is presented in Williams and O'Connor (1995). The greatest drawback of contact resolution between such representations is that they require significantly more computation time than that for spheres.

The ellipsoidal representation investigated in this thesis takes advantage of the simplicity of spheres while overcoming the problems of clustering. In particular, they possess a smooth representation with C^1 continuity that can accurately represent non-spherical bodies, such as ellipsoids. This improves on the sphere by providing interlocking, resistance to rolling, and realistic statistics of orientation for all friction regimes. The geometry is also convex, eliminating the need for multiple surface contact checking. An improvement is discussed for the four-arc ellipsoid approximation detailed in Wang (1999), called here the Equivalent Contact Sphere Method as used in Johnson et. al. (2003). It is a hybrid representation that provides favorable contact detection properties as well as flexibility in representing non-spherical particle shapes. For completeness, the development of the four-arc approximation in 2-D (Wang, 1997) and its development into 3-D (Wang, 1999) will be followed. Then, the development of a new contact resolution algorithm for the four-arc approximation is illustrated, which exploits the implicit geometric properties to identify the point of contact on the surfaces of the ellipsoids. The contact point is defined as a point coincident to potential surfaces of the contacting ellipsoids where the surface normals for the two ellipsoids are parallel. The geometric properties necessary for implementation of the ellipsoid into a DEM implementation are also developed. Finally, details of the computational implementation are shown.

Results of the implementation of this new algorithm are presented, including convergence properties and increases in efficiency. An overview of a developed computational implementation of the algorithm is given along with results of simulations using the methods described in the first sections.

Pathological Behavior in Constant Radius Shape Representations

Zhou (2002) has shown agreement between empirical and numerical results for piling of spherical glass beads. Several studies have investgated this phenomenon, though, it is debatable whether this is directly applicable to DEM modeling of geotechnically realistic systems, since it does not address the influence of dense packing configurations that emerge in systems of perfectly rounded particles. Behavior at the microscale is important in determining behavior at several length scales above, making it important to accurately capture the dominant forces and behavior at the microscale.

In Johnson and Williams (2002), the phenomenon of spontaneous regular packing configurations is investigated through the performance of a simulated standard angle or repose field test. The simulation discussed involves 6624 circular-shaped polygon particles (in the equilibrium configuration) studied using the MIMES DEM simulation package. All particles are represented in the simulation by 16-sided polygons. The field test is carried out by randomly placing disk source particles above the opening to a vertical container, and disks are allowed to fall and settle under gravity. Once settlement has occurred and the kinetic energy of the system is near zero, the sides of the vessel are removed and the particles are allowed to slump. Particles that fall off of the edge of the settlement plate are eliminated from the simulation. The experimental setup is shown in Figure 1. Material data for the particles is taken from physical data given by Watanabe (1999).

The final configuration of the pile is shown in Figure 2. Light gray lines follow the slope of the dense packed structure, which forms in the center of the system. The full view shows that the piling is roughly conical with concave slopes; whereas, the closer view reveals two distinctly different pile structures, an inner hexagonal packing bordered by a more random packing on the outer slopes.

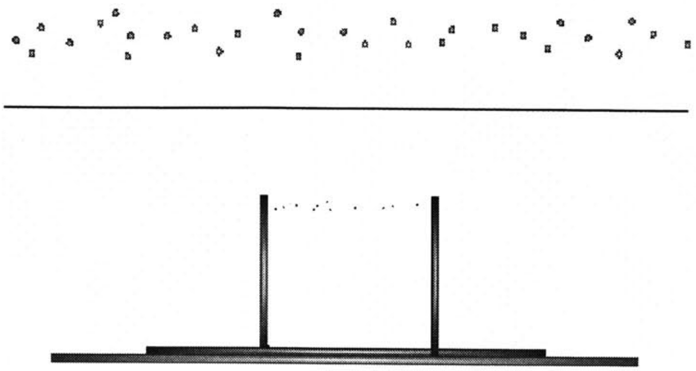


Figure 1: Numerical experiment setup for study of roughly circular polygons; (top) close view of source particle configuration.

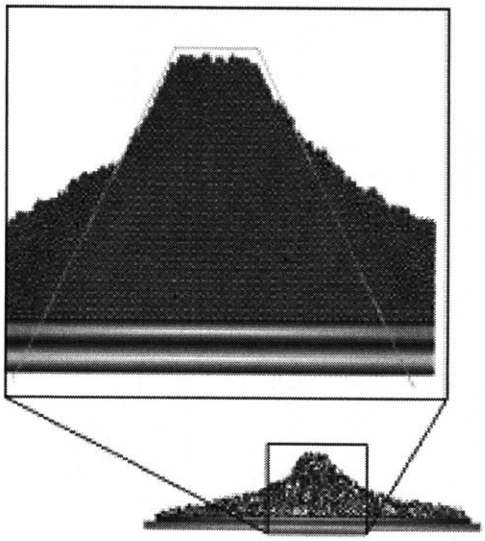


Figure 2. Close view of dense packing in pile of roughly circular polygons with internal friction coefficient of 0.25.

The addition of an aspect ratio of 1.5 yields ellipse-shaped, 16-sided polygonal particles. It is observed that this eliminates the close packing configuration as shown in Figure 3. Trials using the same procedure have been conducted for both the circular and elliptical polygons for a friction coefficients of 0.25 and 0.5. Results obtained for the circular-shaped polygon particles indicate an increase of between 33.2% and 69.1% when the full data set is used as opposed to only that part unsupported by a hexagonal packing.

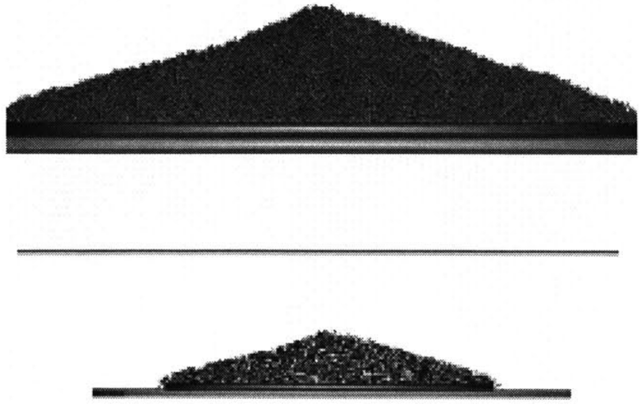


Figure 3. Close and far view of the pile of elliptical, polygonal particles with aspect ratio of 1.5 and internal friction coefficient of 0.25.

Figure 4 shows the angles of repose for the full data sets of the circular and ellipsoidal polygonal particles, as indicated by the legend. Also shown, are the angles of repose for the circular polygon data set truncated to remove the hexagonal packing in the middle of the pile from the calculation of angle of repose. The angle of repose for each trial was acquired by performing a least squares fit on those particles lying on the pile surface to derive an estimate of the slope.

It can immediately be noted that the use of the full data set of surface particles in the circular polygon trials results in angles of repose in excess of those for the ellipsoidal polygon particles under all of the internal friction coefficients investigated. This contradicts empirical research that has generally shown angle of repose to increase with deviation from circular cross-section. However, when the circular data is truncated to remove that part of the pile supported by a hexagonally packed structure, the angle of repose is, as empirically verified, less than those of the ellipsoidal particles.

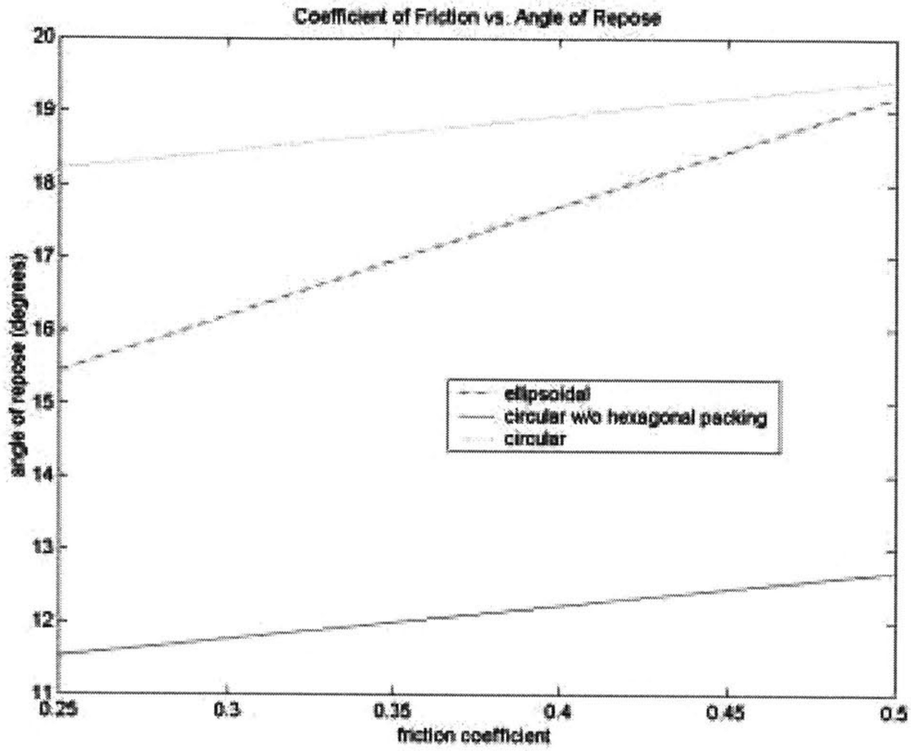


Figure 4. Angles of repose for superquadric ellipsoids and disks for friction coefficients of 0.25 and 0.5.

The study has presently shown that a hexagonal packing structure readily forms in DEM systems using circular polygon elements and that ellipsoidal polygon elements can eliminate this hexagonal packing phenomenon. A relationship between friction coefficient and slope is also illustrated for both the ellipsoidal and circular polygon particle systems.

Existing 3-D Geometric Representations

Several 3-D representations of closed surfaces are discussed in the literature. These can be organized into 3 general categories: discrete, analytical, and hybrid. Discrete representations discretize a shape to a given coarseness. Analytical representations can be described by an explicit or implicit algebraic equation. Hybrid representations usually use analytically-defined surface patches forced to conform to a surface through enforcing compatibility conditions.

The simplest convex analytical representation is the sphere, which uses only four floating point numbers to describe the geometry and requires a trivial contact resolution algorithm (if the difference between the distance between the centroids of the spheres and the sum of the spheres' radii is negative or zero, there is contact). Prolate spheroids have been proposed by Lin (1997) for use in discrete element modeling; the original algorithm for contact detection relies on a geometric potential algorithm to identify a unique point of contact. Ouadfel (1999) presents a more robust contact detection scheme for prolate spheroids that is also formulated to handle memory contact effects and store geometrical data about the contact.

Most other common analytical representations tend to be quadrics, such as tori (Hopkins, 1997 and 1999), ellipsoids (Ting, 1993; Ng and Fang, 1995; and Lin and Ng, 1997), though, more complex shapes such as cardioids (Tijskens, 2002) have also been implemented in 2-D DEM. Development of fast and efficient contact resolution algorithms have met with considerable problems, and most implementations require a costly numerical method to converge to a contact point between two such geometries. The reliance on solving complex expressions in transcendental functions is also a computationally intensive task, though, some of this can be relieved through clever use of memory tables for previous contacts.

An analytical representation that has gained much interest in a wide range of graphics areas, such as computer graphics, camera-aided 3-D modeling, and virtual surgery, is the parent geometry of quadrics, the superquadric. The use of superquadrics in DEM was first proposed in Williams and Pentland (1992). General discussions of the superquadric geometry can be found in Barr (1981), Pentland (1986), and Peter (2000). Superquadrics allow for complex analytical constructions based on an n-power explicit equation of the form: $(\frac{x}{a_1})^{N_1} + (\frac{y}{a_2})^{N_2} + (\frac{z}{a_3})^{N_3} = 1$. Solution of the superquadric, especially at low values of N_i, has been reported to converge quickly (4-6 iterations) due to good conditioning. A ray rejection method is also reported in the source as a method of contact detection. Determining whether a point lies within the 3-D superquadric, on the surface, or outside the surface can be easily determined by solving the equation and evaluating whether the result is less than, greater than, or equal to zero. This is also commonly known as the *inside-out function*. Unfortunately, in the general case, solution of the equation could require evaluation of arbitrary powers of *x*, *y*, and *z*, and contact detection also typically requires evaluation of several points on the surface. This has necessitated other methods of quickly evaluating the geometry, including the use of discrete methods.

The discrete function representation discussed in Williams and O'Connor (1995) divides a convex shape into two shells along a cutting plane, and then discretizes each half shell into a uniform grid projected onto the cutting plane. Associated with each grid point is a height function for the surface. The method is well suited for coarse contact detection through the use of bounding boxes. Efficient contact resolution is achieved through projection of each body's grid onto the grid of the other body and taking the union of the unions of the projections to reduce the possible contact zone to a small enough area that point-wise distance comparisons can be carried out.

Simple polyhedral surfaces are also utilized in DEM. Several researchers have investigated polyhedral surfaces, including Dobkin (1990), and such surfaces are common in computer graphics. Polyhedral elements are easily meshed for use in combined DEM/FEM formulations where the grains are individually treated as elasto-plastic bodies with internal deformation and fracturing. This is discussed in Munjiza (1995, 2002a, 2002b).

Voxels are another type of discretized representations that are commonly used in virtual surgery and organ modeling. These applications are well-suited to voxels, as model information is typically retrieved from scanning instruments (such as CT, CAT, MR, and MRI scans) of a certain resolution and uncertainty that can be captured well by the resolution of the voxel graph. These are often analyzed through some type of binning scheme or binary space partitioning (BSP) trees to decrease contact detection time between a surgical instrument and the tissue. In 3-D, the BSP-tree is called the octtree (i.e., 2 voxels in each of 3 orthogonal directions is $2^3 = 8$). Voxels have also been implemented without BSP trees (Sramek, 2001). Voxels are easily used in a hierarchical representation, since smaller voxels are easily encapsulated by larger (coarser) voxels. Voxelization is also a key concept used in the extended discrete function representation for general (concave or convex) 3-D objects (O'Connor, 1996). Wavelet-based hierarchical representations have also been proposed (Williams and Amaratunga, 1993). In the general case, the hierarchical wavelet approach utilizes a special set of orthogonal basis functions that apply a hierarchy of analytical patches as the surface, making the approach a hybrid approach.

Hybrid approaches are common in the computer gaming and graphics and CAD/CAM industries as well as in the calculation of robot movement and part machining. The most common hybrid representation for closed surfaces is the non-uniform rational b-spline (NURBS), which consists of a non-rational expression for the b-spline, a carefully constructed polynomial typically of degree 3, though, there is no restriction on the degree that may be used. NURBS have become common and are, for instance, included in the OpenGL Utility Library (GLU). Bézier surface patches are also common and are a subset of NURBS defined by the Bernstein basis functions of the form $\sum_{n} u^i (1-u)^{n-i}$ where $\{u \mid 0 \leq u \leq 1\}$. The NURBS set has a convex hull property that allows for rejection of a contact at iteratively more refined levels, using the de Castlejau algorithm for Bézier curves and a similar algorithm in the general case of the non-uniform rational b-spline. Once a certain level of refinement has been reached with the convex hulls of two objects remaining overlapping, the objects are considered in contact.

Jing (2000) describes the application of homeomorphisms to topologically equivalent objects, including conditions for topological equivalence as well as the invariant relationship between such polyhedra under topological transformations (homeomorphisms). These are complex and convex, often requiring expensive transformations and numerical methods to perform contact resolution.

In discrete element modeling, the most popular method for modeling non-spherical or fracturing grains is the multi-sphere cluster. The spheres can be rigidly locked together or bound by certain bonding properties and conditions that allow strain, shear strain, or fracturing within the grain. Sphere-clustering is the basis for the commercial code TRUBAL. The inherent simplicity of the approach, fast contact detection, and compact representation make it preferable for DEM. It is also a straightforward extension of previously developed sphere particle codes. A sphere-clustering approach to approximate ellipsoids is given in Vu-Quoc et. al. (2000), and a general implementation of clustering is given in del Pobil (1996). Kuhn (1993) uses an irregular 3-D polygon chair mesh with vertices as the centroids of spherical elements to represent the space boundary for far-field effects in DEM.

For large-scale simulation of sand transportation in a fluidized bed, a shape representation is desired that will execute quickly as a sphere or sphere cluster approach would provide. However, it should also be characterized by a C^1 continuous surface to ease contact with polyhedral surfaces (as would likely be the model for far field boundaries to be modeled using continuum methods). The C^1 continuity condition would also ease fluid flow calculations, which can be complicated in the region of tangent discontinuities. For this reason a third hybrid representation, the 4-arc approximation, is investigated in this thesis.

Discrete Element Modeling

Before delving into the mechanics of the ellipsoidal representation, a short background on the discrete element method will be presented. A poor understanding of soil behavior compelled Charles Terzaghi to revolutionize soil mechanics in the first part of the 20th Century by developing and applying continuum methods to the study of soil and geotechnical systems. Again these foundations are being driven deeper by not just understanding systems that behave as a continuum but also those that exhibit behavior emergent from the micromechanics of the grains in that system. Discrete element modeling was first formally introduced in Cundall and Strack (1979) as an attempt to address such systems of granular materials where the behavior is driven by grain-scale interactions – behaviors that a continuum method cannot currently replicate without significant effort and empirically derived formulations.

The fundamental difference between continuum and discrete methods is that the continuum method views a system from the whole and attempts to narrow down to micromechanical behaviors. The discrete method looks at the individual particle and tries to pan out to replicate larger systems. If a disturbance of the system at a length scale smaller than that represented by a continuum does not manifest in the length scale of interest, then a continuum approach is valid. However, if the system is characterized by behavior emergent from smaller length scales, a discrete method is more useful. This points to an important subtlety of the discrete element method; it is not always necessary to attempt to model the actual natural system, but rather to use enough particles to capture the behavior of the system at the length scale of interest.

Discrete element methods share the commonality of representing a system as an assembly of discrete bodies interacting through surface contact or indirectly through fluid perturbation. The objects are allowed to rotate and displace, and the method dynamically detects contacts as they occur between bodies. Currently, commercial codes exist for DEM, that represent systems in both 2-D and 3-D. Depending on the application of interest either one of these approaches may be more efficient. For instance, in some fluidized systems there is very little movement of material or change in interaction force in the out-of-plane direction, so it is more efficient to employ a 2-D approach, which can generally simulate more particles and execute algorithms for contact detection more quickly than a 3-D method using similar geometries given that the same computing equipment is used. However, if one was attempting to describe the behavior of a rock slide cascading through a specific value where the surface topology of the slope controls the behavior of the rock slide, it would be inaccurate to employ a 2-D method. The focus of this thesis is on a 3-D approach for modeling particles.

The 4-Arc Approximation

In two dimensions the four-arc approximation for an ellipsoid is defined by the planar region enclosed by four arcs illustrated in Figure 5. The top arc and the side arcs are selected such that there exists a point on the top arc tangent and coincident to a point on each of the side arcs. The bottom arc is the mirror image of the top arc about the horizontal axis. Similarly, the left side arc is the mirror image of the right arc about the vertical axis. The equations for these arcs, which ensure C^0 and C^1 continuity just described, are given by Wang (1999) as:

$$\overline{AC} = \overline{CB} = a \qquad \text{(EQ 1)}$$

$$\overline{DC} = \overline{CE} = b \qquad \text{(EQ 2)}$$

$$\overline{CG} = \frac{(a^2 - b^2)c + (a^2 + b^2)d}{2ac} \qquad \text{(EQ 3)}$$

$$\overline{CF} = \frac{(a^2 - b^2)c + (a^2 + b^2)d}{2bc} \qquad \text{(EQ 4)}$$

where

$$c = \sqrt{a^2 + b^2} \qquad \text{(EQ 5)}$$

$$d = a - b \qquad \text{(EQ 6)}$$

The four-arc approximation for a three-dimensional ellipsoid refers to the volume enclosed by the four arcs shown in Figure 5 rotated about a centerline. In Figure 5, the centerline is the x-axis. The generated surface has the following properties:

- The surface has no singularities or non-differentiable points (i.e., C^0 and C^1 continuity satisfied) to cause contact resolution problems.
- The representation is compact, requiring 10 floating point numbers (2 radii, 1 axial distance of end spheres, 1 transverse distance of arc of rotation vertex, 1 translational position vector, and 1 rotational position vector).
- The curvature, and hence normal, of the surface is known at every point on the three-dimensional surface, which are associated with a set of orthogonal arcs of a circle.

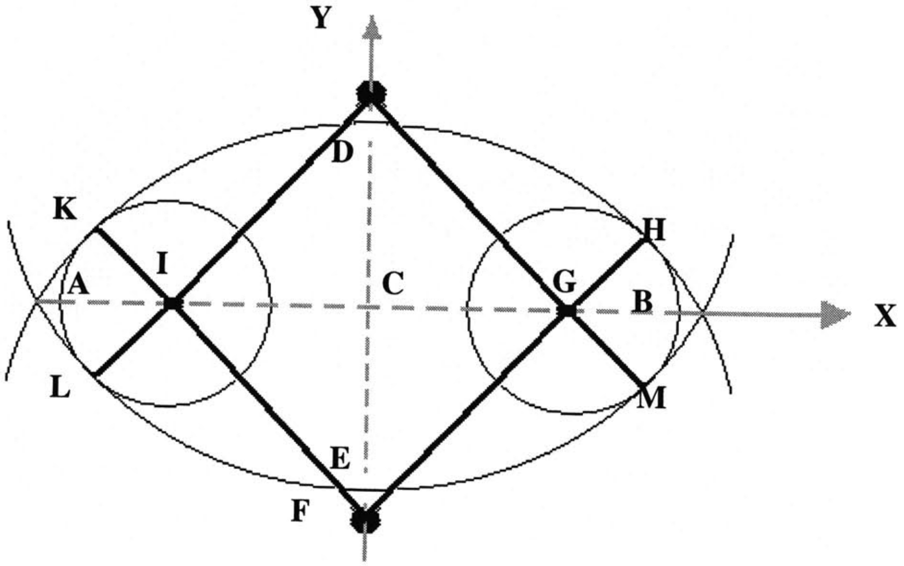


Figure 5: The 4-arc approximation.

Contact Detection

Contact detection typically consists of two distinctly separate operations: neighbor-sorting and detailed contact checking and resolution. Neighbor sorting focuses on reducing the number of candidate pairs of particles to be used Neighbor sorting algorithms are discussed in O'Connor and Williams (1996); Munjiza, Bicanic, and Owen (1993); Wensel and Bicanic (1993); Munjiza and Andrews (1998); and Perkins and Williams (2001). The reader is directed to these references for more information on the neighbor-sorting algorithms available for use in discrete element modeling. This thesis will instead focus on the detailed contact checking and resolution phase of contact detection.

Contact Resolution: Method of Equivalent Spheres

Contact resolution between three-dimensional shapes is rarely intuitive. Decomposing a shape into geometric primitives can significantly decrease the complexity of the algorithm required for contact resolution. The geometry of the four-arc approximation is unique, in that the curvature of the ellipsoid is constant in the radial and longitudinal directions. The case of the ends is simpler. Because the curvature is constant, the ends are spherical caps. The end can therefore be replaced by a complete sphere for contact resolution purposes, since the particle can be viewed as the intersection of the end spheres and the volume enclosed by the rotation of Arc 1 in Figure 5.

The main effort of an algorithm to detect contact between such bodies then must concentrate on the volume enclosed by the rotation of Arc 1 (called the *arc body*). As shown later, contact between the arc body and the end spheres is trivial. Contact between two arc bodies is not trivial.

If the bodies are hard, then the contact is at the intersection of a specific arc from each of the particles. Once these arcs of contact are identified, the point of contact is then easily determined along the line segment connecting the vertices of the arcs of contact (termed here the *vector of contact*). Note that the distance between the arcs of contact is the sum of their radii. In determining contact, the arc of contact is equivalent to a sphere of the arc's of contact radius centered at the arc's of contact vertex in three-space.

If the bodies are soft, and it is assumed that local plastic deformation occurs, then the arcs of contact can be determined as in the hard case. The location of the contact point along vector of contact must be determined through a specific law, such as a simple scaling of the distance between the vertices of the arcs of contact.

In both the hard and soft cases, the arcs of contact can be replaced by equivalent spheres. These equivalent spheres are virtual objects that are a useful tool in evaluating arc body to arc body contact and are unique to each contact pair at each time step. Before giving a formal proof of this method, the determination of regions in contact will first be discussed followed by the development of contact detection involving the end regions and then concluding with a formal proof of the method in the context of arc-body to arc-body contact.

Identifying Region of Contact between Ellipsoids

Figure 6 illustrates the various way in which contact may occur in one of three cases: end sphere to end sphere, end sphere to arc, and arc to arc, as shown in Figure 6. A method of using the angles between the vectors and the positions of the ellipsoids' centroids to identify which of these 3 cases applies is discussed in Wang (1999). However, this operation can be made more computationally efficient by eliminating the evaluation of transcendental functions. This can be achieved by using the inequality in Equation 1.

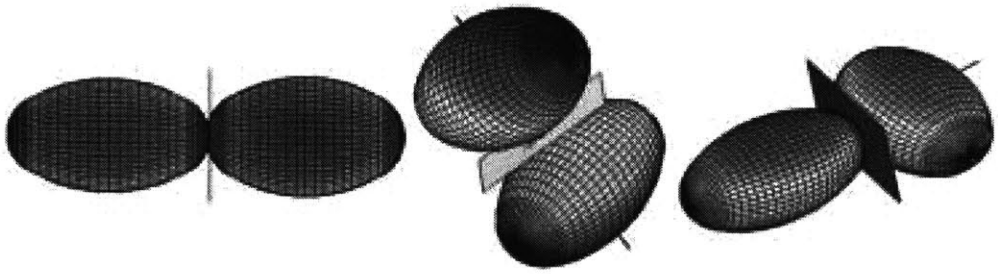


Figure 6: Contact cases: a) end sphere to end sphere, b) arc to arc, c) end sphere to arc

$$\vec{C}_{ij} \bullet (-\vec{P}_i) \geq \frac{\overline{CV}}{r_{arc1} - r_{arc2}} \qquad \text{(EQ 7)}$$

where,

$\vec{C}_{ij}$ ≡ the centroid-to-centroid vector from ellipsoid *i* to ellipsoid *j*.

$-\vec{P}_i$ ≡ candidate normal of contact for ellipsoid *i*, omitting the component of the normal parallel to the axis of rotation for the arcs.

r_{arc1} is the radius of Arc 1 and $r_{arc2} = r_{arc3}$ for ellipsoid *i*.

$\overline{CV}$ is the length of the line segment connecting the centroid of the ellipsoid with the vertex of Arc 1, and it is defined as:

$$\overline{CV} = \frac{(a^2 - b^2)c + (a^2 + b^2)d}{2bc}$$

$$= \frac{(a^2 - b^2) + \sqrt{a^2 + b^2} \cdot (a - b)}{2b}$$

$$= \frac{(a^2 - b^2) + c \cdot d}{2b} \qquad \text{(EQ 8)}$$

where,

a is the major length of the ellipsoid, and *b* is the minor length; $d = a - b$ and $c = \sqrt{a^2 + b^2}$.

By using a dot product in Equation 7 instead of the evaluation of transcendental functions, the cost of left-hand side evaluation of Equation 7 falls to 5 floating point operations. It is important to note that Equation 7 not only eliminates the need for evaluating a transcendental function, but it also requires that the right-hand side quantity in Equation 7 need only be calculated once for each ellipsoid, eliminating the overhead of calculating Equation 8 repeatedly during the simulation.

Connection to the Equivalent Contact Spheres Method

If Equation 7 returns false, then for ellipsoid *i* the equivalent contact sphere will be the end sphere for ellipsoid *i*. The centroid of the equivalent contact sphere associated with ellipsoid *i* is at the centroid of the arc (Arc 2 or Arc 3) for ellipsoid *i*. This simply indicates that if the ellipsoids under consideration are found to be in contact, the contact point on ellipsoid *i* will lie on the surface formed by the rotation of Arc 2 or Arc 3 for ellipsoid *i*. A secondary procedure can then be conducted if an end sphere is identified in order to discover which end sphere (the rotation of Arc 2 or the rotation of Arc 3 for ellipsoid *i*) is the equivalent contact sphere. The radius is derived from the definition of the ellipsoid geometry.

If Equation 1 returns true, then the rotated Arc 1 of ellipsoid *i* will determine the equivalent contact sphere for ellipsoid *i*. This simply indicates that if the ellipsoids under consideration are found to be in contact, the contact point on ellipsoid *i* will lie on the surface formed by the rotation of Arc 1 for ellipsoid *i*. The identification of $\vec{P}_i$ in Equation 7 forms the basis in determining the centroid of the equivalent contact sphere for ellipsoid *i*.

$-\vec{P}_i$ is, by definition, the component of the normal to the surface of ellipsoid *i*, along the local *y* and *z* directions, where the local *x*, *y* and *z* are a Cartesian coordinate system centered about the ellipsoid's centroid and with the local *x* axis along the axis of rotation (also referred to here as the centerline or major axis). Therefore, $\vec{P}_i$ will define the direction of the vertex of Arc 1, when Arc 1 is positioned such that it is coincident with the arc passing through the point of contact. That is, the centroid of the equivalent sphere of contact for ellipsoid *i* is coincident with the vertex of the arc of contact for ellipsoid *i*. This is not intuitively obvious, and this statement is formally proven in the following sections which detail the various types of possible contact.

Framework for Contact Detection and Resolution

This section details a new analytical scheme to detect and localize contact. To achieve this, we need to consider the contact detection between geometric primitives, which results in the following five cases: sphere-sphere, sphere-plane, sphere-arc body, arc body-plane, and arc body-arc body.

Sphere-to-Sphere Contact Detection

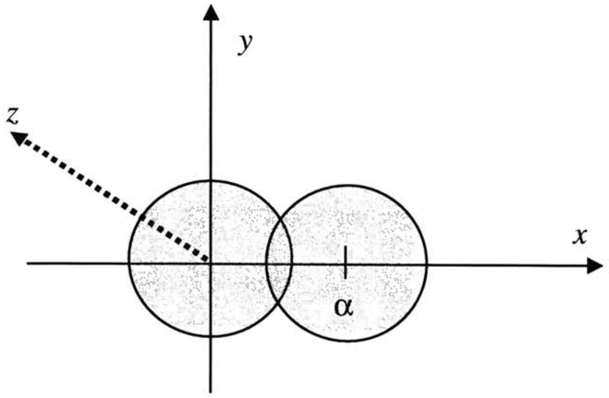


Figure 6: Sphere-Sphere Contact

Sphere to sphere contact detection and resolution is trivial. Contact detection tests whether the sum of the sphere radii exceeds the distance between the sphere centroids, a 39 flop and 1 test operation for the pair (15 flops of which are for the global-local transform on each sphere). The following develops expressions for the surface of contact, the volumetric overlap, and the point of contact.

Let 2 spheres be defined in the orthogonal Cartesian coordinate space *xyz*, such that the following equations define their geometry:

$$\textit{Sphere 1: } x^2 + y^2 + z^2 = {r_1}^2 \qquad \text{(EQ 9)}$$

$$\textit{Sphere 2: } (x+\alpha)^2 + y^2 + z^2 = {r_2}^2 \qquad \text{(EQ 10)}$$

The contact normals to both lie along the x-axis:

$$\textit{Sphere 1: } \langle 1 \quad 0 \quad 0 \rangle$$

$$\textit{Sphere 2: } \langle -1 \quad 0 \quad 0 \rangle$$

Solving for y^2 in Equation 10:

$$y^2 = {r_2}^2 - (x+\alpha)^2 - z^2 \qquad \text{(EQ 11)}$$

Using Equation 9 and Equation 11:

$$x^2 + (r_2^{\ 2} - (x+\alpha)^2 - z^2) + z^2 = r_1^{\ 2}$$

$$r_2^{\ 2} - r_1^{\ 2} = 2\alpha x + \alpha^2$$

$$x = \frac{r_2^{\ 2} - r_1^{\ 2} - \alpha^2}{2\alpha} = \beta \qquad \text{(EQ 12)}$$

From Equation 12, it is noted that the surface of contact lies in a plane parallel to the *y-z* plane, and its equation is given by Equation 13.

$$y^2 + z^2 = r_1^{\ 2} - \beta^2 = R^2 \qquad \text{(EQ 13)}$$

$$R = \sqrt{r_1^{\ 2} - \left(\frac{r_2^{\ 2} - r_1^{\ 2} - \alpha^2}{2\alpha}\right)^2} \qquad \text{(EQ 14)}$$

The volume of this contact is the familiar 3-D lens, whose volume is given by Equation 15.

$$V = \frac{1}{12\alpha}\pi\left((r_2 + r_1 - \alpha)^2\left(\alpha^2 + 2\alpha(r_2 + r_1) - 3\left(r_2^{\ 2} + r_1^{\ 2}\right) + 6r_2r_1\right)\right) \qquad \text{(EQ 15)}$$

Sphere-to-Plane Contact Detection

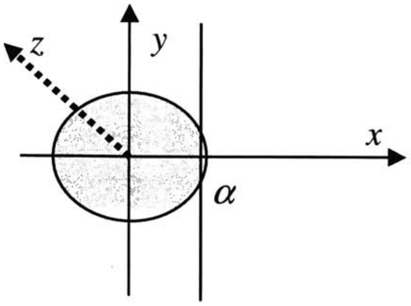


Figure 7: Sphere-Plane Contact

Sphere to plane contact detection and resolution is also trivial. Contact detection tests whether the normal distance between the plane and the centroid is greater than the sphere radius. The following develops expressions for the surface of contact, the volumetric overlap, and the point of contact.

Let a sphere and plane be defined in the orthogonal Cartesian coordinate space *xyz*, such that the following equations define their geometry:

$$\textit{Sphere: } x^2 + y^2 + z^2 = r_1^{\ 2} \qquad \text{(EQ 16)}$$

$$\textit{Plane: } x = \alpha \qquad \text{(EQ 17)}$$

Normals:

Sphere: $\langle 1 \quad 0 \quad 0 \rangle$

Plane: $\langle -1 \quad 0 \quad 0 \rangle$

Replacing x in Equation 16, the surface of contact lies in the plane defined in Equation 18.

$$y^2 + z^2 = r^2 - \alpha^2 \qquad \text{(EQ 18)}$$

The volume of this contact is the familiar 3-D spherical cap, whose volume is given Equation 19.

$$V = \int_{\alpha}^{\alpha+h} \pi \cdot R(x)^2 dx = \pi \int_{\alpha}^{\alpha+h} (R^2 - x^2) dx = \pi \left[R^2 x - \frac{x^3}{3} \right]_{\alpha}^{\alpha+h}$$

$$\frac{\pi}{3}\left(3(\alpha + h)R^2 - \alpha^3 - h^3 - 3\alpha R^2 + \alpha^3\right) =$$

$$\frac{\pi}{3}\left(3hR^2 - h^3\right) = \frac{\pi h}{3}\left(3R^2 - h^2\right) \qquad \text{(EQ 19)}$$

$$h = R - \alpha \qquad \text{(EQ 20)}$$

Sphere-to-Arc Body Contact Detection

Sphere to arc body contact detection and resolution requires minimal development. Contact detection tests whether the sum of the sphere radius and arc radius exceeds the distance between the sphere and arc centroids. It is first necessary to outline why this test is necessary and sufficient to detect contact.

1. Contact is defined as occurring between bodies at a point where the normals of both bodies at that point are opposite and parallel.
2. The rotational symmetry of the pseudo-ellipsoid geometry about the major axis requires that all normals to the body pass through the major axis.
3. The symmetry of the sphere about the centroid requires that all normals to the surface pass through the centroid.
4. The common normal to the pseudo-ellipsoid and sphere, therefore, must lie in the plane defined by the normal $\hat{e}_n$ where $\exists\left(\hat{e}_n \middle| \ \hat{e}_n = \hat{e}_l \times \hat{e}_c\right)$, where $\hat{e}_l$ is the unit vector along the major axis of the

pseudo-ellipsoid and $\hat{e}_c$ is the vector between the pseudo-ellipsoid's centroid and the centroid of the sphere.

Let a sphere and rotated arc be defined in the orthogonal Cartesian coordinate space *xyz*, such that $\hat{e}_n$ lies along the *z*-axis, and the following equations define their geometry:

Arc body: $$y^2 + z^2 = r(y,z)^2 \qquad \text{(EQ 21)}$$

$$(r(y,z)+\alpha)^2 + x^2 = R_A{}^2 \qquad \text{(EQ 22)}$$

$$\left(\sqrt{y^2+z^2}+\alpha\right)^2 + x^2 = R_A{}^2 \qquad \text{(EQ 23)}$$

Sphere: $$(x-a_1)^2 + (y-a_2)^2 + z^2 = R_S{}^2 \qquad \text{(EQ 24)}$$

$$z^2 = R_S{}^2 - (x-a_1)^2 - (y-a_2)^2 \qquad \text{(EQ 25)}$$

Arc body: $$\frac{1}{\sqrt{a_1{}^2 + (a_2+\alpha)^2}}\langle a_1 \quad a_2+\alpha \quad 0\rangle$$

Sphere: $$\frac{1}{\sqrt{a_1{}^2 + (a_2+\alpha)^2}}\langle -a_1 \quad -(a_2+\alpha) \quad 0\rangle$$

Substitute Equation 25 into Equation 23:

$$\left(\sqrt{y^2 + R_S{}^2 - (x-a_1)^2 - (y-a_2)^2} + \alpha\right)^2 + x^2 = R_A{}^2$$

$$y^2 + x^2 + R_S{}^2 - (x-a_1)^2 - (y-a_2)^2 + \alpha^2 + 2\alpha\sqrt{y^2 + R_S{}^2 - (x-a_1)^2 - (y-a_2)^2} = R_A{}^2$$

$$R_S{}^2 + 2a_1x - a_1{}^2 + 2a_2y - a_2{}^2 + \alpha^2 + 2\alpha\sqrt{y^2 + R_S{}^2 - (x-a_1)^2 - (y-a_2)^2} = R_A{}^2$$

Gives the surface of contact, as shown in Equation 26.

$$a_1x + a_2y + \alpha\sqrt{y^2 + R_S{}^2 - (x-a_1)^2 - (y-a_2)^2} = \frac{R_A{}^2 - R_S{}^2 + a_1{}^2 + a_2{}^2 - \alpha^2}{2}$$

(EQ 26)

Arc body-to-Plane Contact Detection

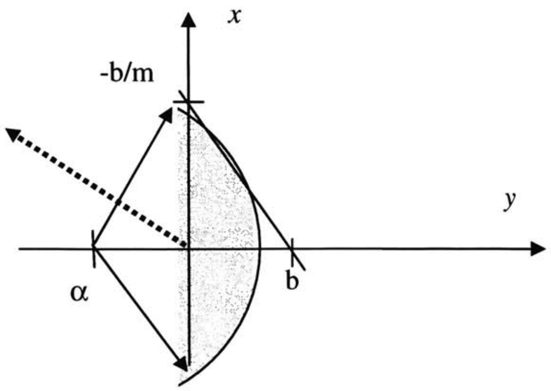


Figure 8: Arc body - Plane

Arc body to plane contact detection and resolution requires minimal development. Contact detection tests whether the boundary of the arc of contact overlaps the plane.

It is first necessary to outline why this test is necessary and sufficient to detect contact.

1. Contact is defined as occurring between bodies at a point where the normals of both bodies at that point are opposite and parallel.
2. The rotational symmetry of the pseudo-ellipsoid geometry about the major axis requires that all normals to the body pass through the major axis.
3. For any line, *L*, and any plane, *P*, there exists exactly 1 plane, *N*, that contains *L* and passes perpendicular to *P*, or:

   Let $\hat{e}_l$ be the unit vector in the direction of *L*, $\hat{e}_p$ be the unit vector perpendicular to *P*, and $\hat{e}_n$ be the unit vector perpendicular to *N*, then:

   $$\exists\left(\hat{e}_n \middle|\ \hat{e}_n = \hat{e}_l \times \hat{e}_p \Rightarrow \hat{e}_l \cdot \hat{e}_n = 0 \wedge \hat{e}_p \cdot \hat{e}_n = 0\right)$$

Let a sphere and rotated arc be defined in the orthogonal Cartesian coordinate space *xyz*, such that $\hat{e}_n$ lies along the *z*-axis, and the following equations define their geometry. The arc body geometry is defined in Equation 21-23, and repeated here.

Arc body:

$$y^2 + z^2 = r(y,z)^2$$

$$\left(r(y,z) + \alpha\right)^2 + x^2 = R_A{}^2$$

$$\left(\sqrt{y^2 + z^2} + \alpha\right)^2 + x^2 = R_A{}^2$$

Plane:

$$y = mx + b \qquad \text{(EQ 27)}$$

Normals:

$$\textit{Arc body: } \frac{1}{\sqrt{1+\left(\frac{1}{m}\right)^2}}\left\langle -1 \quad -\frac{1}{m} \quad 0 \right\rangle$$

$$\textit{Plane: } \frac{1}{\sqrt{1+\left(\frac{1}{m}\right)^2}}\left\langle 1 \quad \frac{1}{m} \quad 0 \right\rangle$$

Substitute Equation 27 into Equation 23:

$$\left(\sqrt{(mx+b)^2+z^2}+\alpha\right)^2+x^2=R_A{}^2$$

$$(mx+b)^2+z^2+2\alpha\sqrt{(mx+b)^2+z^2}+\alpha^2+x^2=R_A{}^2$$

$$m^2x^2+2bmx+b^2+z^2+2\alpha\sqrt{(mx+b)^2+z^2}+\alpha^2+x^2=R_A{}^2$$

Gives the surface of contact as in Equation 28.

$$(1+m^2)x^2+2bmx+z^2+2\alpha\sqrt{(mx+b)^2+z^2}=R_A{}^2-(b^2+\alpha^2) \qquad \text{(EQ 28)}$$

Arc body-to-Arc body Contact Detection

Arc body to arc body contact detection and resolution will now require the majority of the development in this paper. Contact detection is considerably more complex, requiring the solution of a nonlinear set of equations. This nonlinear set of equations, as outlined in Wang (1999), is ill-suited to direct solution, requiring the evaluation of several transcendental functions. Such function calls are computationally expensive. For practicality of the algorithm for implementation, it is necessary to use a less costly numerical method to determine the point of contact.

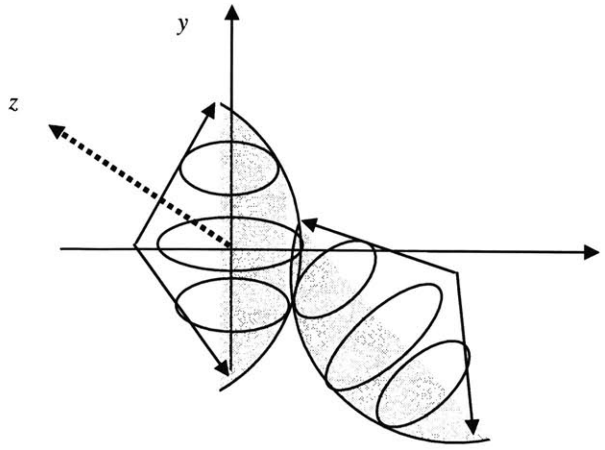


Figure 9: Arc body – Arc body

Initially, the constraints on the domain of possible arcs of contact will be developed. It will be shown that the constraints lead to the definition of a unique pair of arcs of contact. Secondly, the current algorithm to identify the point of contact will be presented and critiqued. Finally, an improved algorithm will be presented that converges to the unique answer, using the geometric properties of the pseudo-ellipsoid to aid in convergence.

The physical constraints of contact:

1. When in contact, the contact points on the two bodies are coincident.
2. Contact forces act normal and tangential to the surface of the bodies at the point of contact.
3. Newton's 3rd Law of Motion stipulates that every force is balanced by an equal and opposite force, so the normal and tangential forces on one body are resisted by equal and opposite normal and tangential forces on the other body.

   $\vec{F}_n^{\,1} + \vec{F}_n^{\,2} = 0 \wedge \vec{F}_t^{\,1} + \vec{F}_t^{\,2} = 0$, where the superscript identifies the body, the subscript *n* denotes normal force and the subscript *t* denotes tangential force.

The geometric constraints for contact:

1. The geometry of the pseudo-ellipsoid can be characterized in the orthogonal cylindrical space (r,x,θ) as $r(x,\theta) = \sqrt{R^2 - x^2} - \alpha$.
2. This yields the normal space along the surface, S, of the pseudo-ellipsoid (transformed to the orthogonal rectangular coordinates *x,y,z*) as:

   $$N(x,y,z) = \left\langle \frac{x}{\alpha} \quad \sin(\theta) \quad \cos(\theta) \right\rangle \cdot f_{normalizing}$$

where $f_{normalizing} = \sqrt{1+\left(\frac{x}{\alpha}\right)^2}$

Let $\tilde{N}(x,y,z)$ be a parallel vector space to $N(x,y,z)$, such that $\tilde{N}(x,y,z) = \left\langle \frac{x}{\alpha} \quad \sin(\theta) \quad \cos(\theta) \right\rangle$

3. This indicates that $\tilde{N}_y$ and $\tilde{N}_z$ are independent of $\tilde{N}_x$.
4. From 4 we decouple the parallel normal space equation:

   $\begin{bmatrix} \tilde{N}_y \\ \tilde{N}_z \end{bmatrix} = \begin{bmatrix} \sin(\theta) \\ \cos(\theta) \end{bmatrix} \Rightarrow$ all vectors $N(x,y,z)$ pass through the x-axis.
5. By defining the geometry of the arc with a vertex at a distance from the centroid rotated about $\hat{e}_l$, where l lies along the major axis (here defined arbitrarily as the x-axis), we find that, by definition, all normals pass through the path formed by the vertex rotated about $\hat{e}_l \Rightarrow r(0,\theta) = \alpha$.
6. From 4, 5, and 2, we have the constraints for the contact vector: there exists a unique line segment, L, with terminal points along $[r(0,\theta) = \alpha]^1_{local}$ and $[r(0,\theta) = \alpha]^2_{local}$ intersecting the lines $[y = 0 \wedge z = 0]^1_{local}$ and $[y = 0 \wedge z = 0]^2_{local}$ and passing through the unknown coincident contact point p. The contact vectors for the ellipsoids lie parallel to L.

Special Cases of Arc-Body to Arc-Body Contact

Arc body to arc body contact detection is a complex problem for most cases. There are, however, two special cases where the problem reduces to a simple evaluation. The first is for the case where the major axes of the pair of bodies are coplanar. The second is where the centroids of the pair of bodies lie along a line that is orthogonal to the major axes of both bodies.

For the case of coplanar major axes, the identification of the line of contact is simple. From the development of the geometrical constraints in the previous section, the line containing the contact point and both vertices of the bodies' arcs of contact must lie in the plane containing both major axes. This line must pass through a point that is along a path perpendicular to the body's major axis. For each body, this leaves two possible points to either side of the major axis in the major axes' common plane. Together, the points offer four possible point pairs. To satisfy that the normals be parallel and opposite, only two of these point pairs are valid. The correct point pair corresponding to the exact vertices of the arcs of contact for the two bodies can then be easily identified by a triple vector product of the directional vector corresponding to the line segment between the centroids of the bodies.

For case of centroids lying on a line perpendicular to the plane defined by the directional vectors of the bodies' respective major axes, the identification of the line of contact is likewise simple. Consider the paths along which the vertices of the respective arcs of contact must lie as discs in space. If there exists a line that passes through the centers of both discs and is also a common line to the planes containing both discs, then the line of contact must be collinear with this common line. This ensures that all constraints are satisfied.

Current Algorithm

The cases of contact outside of generalized arc body to arc body contact are relatively straightforward, as has been shown in the analytical solutions offered in the previous sections. To evaluate the case of general arc body to arc body contact it is best to use an iterative method. One such method is proposed in Wang (1999), which uses the geometry of the circular cross-section to detect whether a pair of ellipsoids is in contact (contact detection) and where on their surfaces they are in contact (contact resolution). It views the pseudo-ellipsoid as being composed of an infinite number of arcs at different rotations about the major axis. The vertices of these arcs are located along a circular path of radius $\overline{CV}$ (see Equation 2) perpendicular to the major axis and centered at the ellipsoid's centroid.

Because of this constraint, Wang (1999) views this as an energy minimization problem: the arc surfaces of the two pseudo-ellipsoids will be most in contact when their respective vertices are farthest apart – their energy is minimized. It is implemented as the iterative method illustrated in Figure 10.

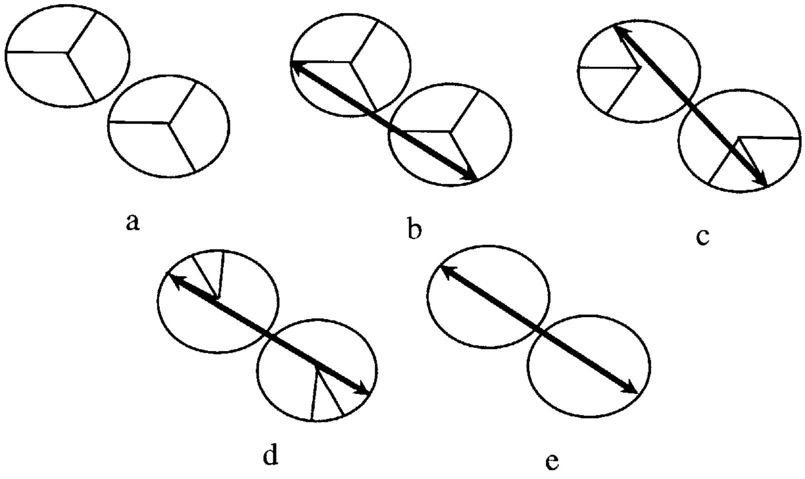


Figure 10: Previous Detailed Contact Detection Algorithm

There are two substantial drawbacks to this approach:

1. The numerical method does not utilize properties of the geometry to aid in convergence, relying instead on a generally convergent algorithm to come to a solution.
2. As implied in the previous section, there is no constraint in the solution to passing through the lines $[y = 0 \wedge z = 0]_{local}^{1}$ and $[y = 0 \wedge z = 0]_{local}^{2}$. This is a necessary condition for identifying the common normal vector. The constraint that the common normal vector passes through the major axes is not necessarily satisfied through this method.

Application of Vector Scaling to Critiquing Current Method

From case 1 and 2 in the previous section, it can be generalized that for the pseudo-ellipsoid pair there exists $\hat{e}_l^{\,1}$ and $\hat{e}_l^{\,2}$ parallel to the major axes of pseudo-ellipsoids 1 and 2, respectively, and a centroid-to-centroid vector $\hat{e}_{c1c2}$, such that if $\hat{e}_l^{\,1}$, $\hat{e}_l^{\,2}$, and $\hat{e}_{c1c2}$ are coplanar, the method described in Wang (1999) will converge on the correct solution. Also, if no dilation is required (that is, if the centroid of the disk lies on the projection of the contact point onto the major axis), then the method will converge to the exact solution. This paraphrases the special cases described in arc body to arc body contact, so the current method only converges to the exact solution for the special cases in arc body to arc body contact.

For most applications, this restriction is unacceptable. The improved algorithm discussed next focuses on the positioning of equivalent contact spheres to ensure an equivalent representation (used only for purposes of contact detection) to the pseudo-ellipsoidal geometry. The method used to position these equivalent spheres improves on the current method by accounting for all constraints on the geometry of the pseudo-ellipsoid. The iterative refinement method proposed later is also shown to converge to the exact solution.

Improved Algorithm

A sufficient algorithm for contact detection would identify a candidate point of contact by identifying the candidate arc of contact on each pseudo-ellipsoid – that is, a pair of candidate arcs of contact. A simple test for contact is then possible by comparing the sum of the radii of the candidate arcs of contact versus the distance between the vertices of those candidate arcs of contact.

Identifying the unique candidate pair of arcs can be separated into 2 parts: identifying the rotation of the contact point about the major axis and translational position along the major axis.

Identifying Rotational Position of Contact Point

If the translational position of the contact point along the major axis is identified (the projection of the contact point onto the major axis), then the rotational position about the major axis can be determined using either the procedure introduced by Wang (1999) modified to be centered on the major axis at the translational position or by the following algorithm.

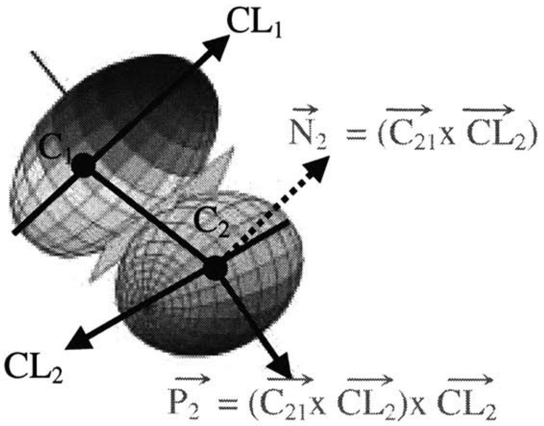

CL_1
$\overrightarrow{N_2} = (\overrightarrow{C_{21}} x \overrightarrow{CL_2})$
C_1
C_2
CL_2
$\overrightarrow{P_2} = (\overrightarrow{C_{21}} x \overrightarrow{CL_2}) x \overrightarrow{CL_2}$

Figure 11: Initial Estimate for Improved Contact Resolution

The algorithm proposed in this section and illustrated in Figure 11 offers an alternative method to the current method. This algorithm utilizes the geometric properties of the pseudo-ellipsoid for an increased rate of convergence and a more efficient (computationally less costly) evaluation per iteration.

The vector, $\vec{C}_{ij}$, between the centroids of the ellipsoids in the ellipsoid pair is found,

$$\hat{C}_{ij} = R_i \cdot (\frac{\vec{C}_{ij}}{\|\vec{C}_{ij}\|}), \qquad \text{(EQ 29)}$$

where R_i is the global-to-local rotation matrix for ellipsoid i, $\vec{C}_{ij}$ is the vector between the centroids of ellipsoids i and j. The usefulness of working in local coordinates becomes apparent, since cross products with the major axis must be computed, and the major axis in local coordinates is represented by the vector (1 0 0).

Next the normal, $\vec{N}_i$, to the centroid-centroid vector and the major axis of ellipsoid i is found. The normal represents the plane formed by the two vectors.

$$\vec{N}_i = \vec{C}_{ij} \times (1 \quad 0 \quad 0) \qquad \text{(EQ 30)}$$

Then the normal, $\vec{P}_i$, to the major axis of the ellipsoid and $\vec{N}_i$ is found. $\vec{P}_i$ is a unit vector in the plane formed by the centroid-centroid vector and the major axis of ellipsoid i. It is also orthogonal to the major axis of ellipsoid i.

$$\vec{P}_i = \vec{N}_i \times (1 \quad 0 \quad 0) \qquad \text{(EQ 31)}$$

The location of the vertex of the arc of contact can now be specified in local coordinates as:

$$\vec{P}_i \cdot \overline{CF} \qquad \text{(EQ 32)}$$

To complete the procedure, the spatial bounds of the arc contact surface must be tested, as detailed in Equation 1.

Approximation for Identifying Translational and Rotational Position of Contact Point

When the location of contact along the major axis is unknown, the relationship of the limiting cases can be exploited to provide an approximation of the position of the contact point. There are two extreme cases in the formulation of a contact detection scheme between these pseudo-ellipsoids: those with aspect ratio near 1 and those with high aspect ratios, as shown in Figure 12.

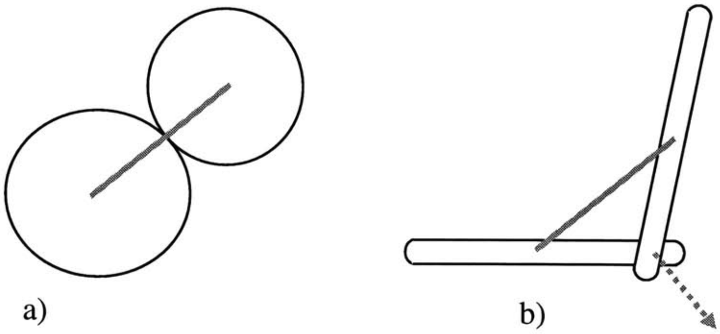


Figure 12: Limiting cases of the initial estimate method

As the aspect ratio diverges from 1, the centroid-to-centroid vector approximation of the contact normal direction begins to degrade in accuracy. A quick linear interpolation based on a metric of the side arc curvature could be used to improve the initial guess. A graphical representation of the procedure is shown in Figure 13.

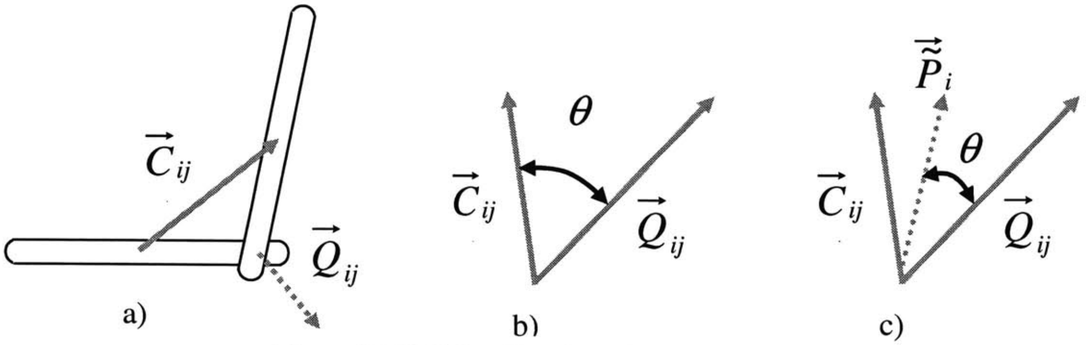


Figure 13: Refining Algorithm for Initial Estimate

The procedure would proceed as follows. Let $\vec{C}_{ij}$ be the unit vector from the centroid of ellipsoid i towards the centroid of ellipsoid j. Let $\overrightarrow{CL_i}$ be the unit vector along the major axis of ellipsoid i. Let $\vec{Q}_{ij}$ be a unit vector normal to the plane formed by $\overrightarrow{CL_i}$ and $\overrightarrow{CL_j}$.

$$\vec{Q}_{ij} = \frac{\overrightarrow{CL_i} \times \overrightarrow{CL_j}}{\left|\overrightarrow{CL_i} \times \overrightarrow{CL_j}\right|}$$

The angle between $\vec{Q}_{ij}$ and $\vec{C}_{ij}$ is then found:

$$\theta = a\cos\left(dot\left(\vec{Q}_{ij}, \vec{C}_{ij}\right)\right)$$

A linear scale factor to quantify curvature is then given by:

$$k = \frac{\overline{FH} - \overline{FG}}{\overline{FH}}$$

A new angle is calculated using the scale factor.

$$\tilde{\theta} = k\theta$$

Let the updated guess, $\vec{\tilde{P}}_i$, be the unit vector in the direction of $\vec{Q}_{ij}$ rotated through $\tilde{\theta}$ towards $\vec{C}_{ij}$ about the normal to the plane formed by $\vec{Q}_{ij}$ and $\vec{C}_{ij}$.

Convergent Method for Identifying Position of Contact Point

Without *a priori* knowledge of the translational position of the contact point along the major axis or the rotational position of the contact point about the major axis, it is necessary to employ an iterative scheme to ensure convergence to the exact solution. If the approximations detailed above do not provide the required precision, the following method will identify the candidate pair of arcs of contact exactly.

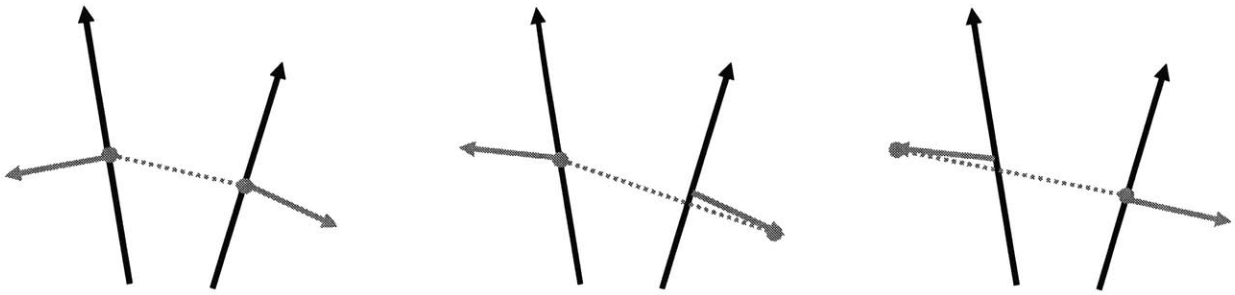
When the axes of the bodies in question are both coplanar with the vector between the centroids of the ellipsoids, it has been shown in previous sections that the direction vector, $\vec{P}_i$, will point towards the vertex of the arc of contact. This is only a specialized case. It works well for the case of a sphere to sphere or sphere to arc contact because the axis of a sphere whose centroid is fixed in space will be arbitrary and, therefore, can be oriented to be parallel with the major axis of the ellipsoid or with an arbitrary major axis of another sphere with known centroid.

When the major axes of the ellipsoids are not oriented parallel, it becomes necessary to approximate, as in the previous section, or iterate to converge to an exact answer. Iterating Equation 33-34, using an initial guess of $\vec{P}_i$ found in Equation 31, the method will converge rapidly. The procedure for 1 iteration is shown graphically in Figure 14.

$$\vec{P}_i = \frac{((\overrightarrow{CG}_{ij} + \vec{P}_j)\times(1\quad 0\quad 0))\times(1\quad 0\quad 0)}{\left|((\overrightarrow{CG}_{ij} + \vec{P}_j)\times(1\quad 0\quad 0))\times(1\quad 0\quad 0)\right|} \qquad \text{(EQ 33)}$$

$$\vec{P}_j = \frac{((\overrightarrow{CG}_{ji} + \vec{P}_i)\times(1\quad 0\quad 0))\times(1\quad 0\quad 0)}{\left|((\overrightarrow{CG}_{ji} + \vec{P}_i)\times(1\quad 0\quad 0))\times(1\quad 0\quad 0)\right|} \qquad \text{(EQ 34)}$$

To complete the procedure, Equation 33 is applied to identify the location of the vertex of the arc of contact for ellipsoid *i*. Then the spatial bounds of the contact surface are test with Equation 1.



a) b) c)

Figure 14: Convergent Iterative Refinement of Initial Estimate

Dynamics in 3-D

Mass of the Geometry

$$dm = \rho \pi r^2 dx$$

$$r(x) = \begin{cases} \sqrt{R^2 - x^2} - a, \{x \mid 0 \le x \le x_0\} \\ \sqrt{r_s^{\,2} - (x - x_{CG})^2}, \{x \mid x_0 \le x \le x_1\} \end{cases}, \qquad \text{(EQ 35)}$$

where r_s = the radius of the end sphere, R = the radius of the body arc. The value of x_{CG} is defined in the geometry and x_0 and x_1 can be determined from x_{CG} and other parameters in the geometry (shown in Figure 1) as $x_0 = x_{CG} \cdot \frac{R + r_s}{R}$, and $x_1 = x_{CG} + r_s$

$$M = \int dm = \rho\pi \int_{-x_1}^{x_1} r^2 dx = 2\rho\pi \int_0^{x_1} r^2 dx = 2\rho\pi\left(\int_0^{x_0} r^2 dx + \int_{x_0}^{x_1} r^2 dx\right)$$

$$= 2\rho\pi\left(\int_0^{x_0}\left(R^2 - x^2 - 2a\sqrt{R^2 - x^2} + a^2\right)dx + \int_{x_0}^{x_1} r_s^{\,2} - (x - x_{CG})^2 dx\right)$$

$$= 2\rho\pi\left(x_0\left(R^2 + a^2\right) - \frac{x_0^{\,3}}{3} - a\left(x_0\sqrt{R^2 - x_0^{\,2}} + R^2 \tan^{-1}\left(\frac{x_0}{\sqrt{R^2 - x_0^{\,2}}}\right)\right) + \left[r_s^{\,2}x - \frac{x^3}{3}\right]_{x_0 - CG}^{x_1 - CG}\right)$$

where $\left[r_s^{\,2}x - \frac{x^3}{3}\right]_{x_0 - CG}^{x_1 - CG} = r_s^{\,2}(x_1 - x_0) - \frac{(x_1 - CG)^3 - (x_0 - CG)^3}{3}$

Moment of Inertia for the Equivalent Spheres

From the development in Beer and Johnston (1996), the moments of inertia for a body of rotation can be derived from the following relationships:

$$dI_x = \frac{r^2}{2} dm \qquad \text{(EQ 36)}$$

$$dI_y = \left(\frac{r^2}{4} + x^2\right) dm \qquad \text{(EQ 37)}$$

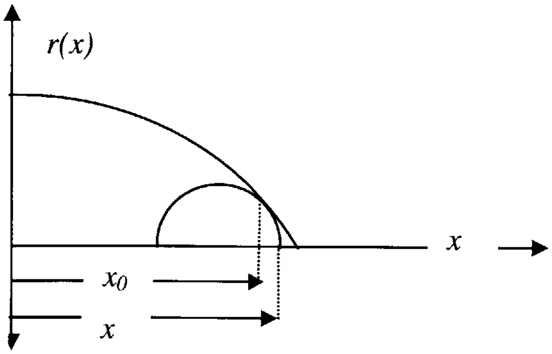


Figure 15: Integration regions on 4-arc approximation.

Moment of Inertia about the Central Axis

Starting with the relationship above and simplifying:

$$dI_x = \frac{r^2}{2}\rho\pi r^2 dx = \frac{\rho\pi r^4}{2}dx$$

$$I_x = \int_{-x_1}^{x_1} \frac{\rho\pi r^4}{2} dx = \rho\pi \int_0^{x_1} r^4 dx = \rho\pi\left(\int_0^{x_0} r^4 dx + \int_{x_0}^{x_1} r^4 dx \right)$$

$$= \rho\pi\left(\int_0^{x_0} \left(\sqrt{R^2 - x^2} - a\right)^4 dx + \int_{x_0}^{x_1} \left(\sqrt{r_s{}^2 - (x - x_{CG})^2} \right)^4 dx \right)$$

$$= \rho\pi\left(\int_0^{x_0} \left(R^2 - x^2 - 2\cdot a\cdot\sqrt{R^2 - x^2} - a^2\right)^2 dx + \int_{x_0}^{x_1} \left(r_s{}^2 - (x - x_{CG})^2\right)^2 dx \right)$$

$$= \rho\pi(A + B)$$

$$A = \int_0^{x_0} \left(R^2 - x^2 - 2\cdot a\cdot\sqrt{R^2 - x^2} - a^2\right)^2 dx$$

$$= \left[\left(a^4 + 6a^2R^2 + R^4\right)x - \frac{2\left(3a^2 + R^2\right)}{3}x^3 + \frac{x^5}{5} + \sqrt{R^2 - x^2}\left(-\frac{a}{2}\left(4a^2 + 5R^2\right)x + ax^3 \right) + \right.$$

$$\frac{aR^2\left(4a^2+3R^2\right)}{2}\tan^{-1}\left(\frac{x}{\sqrt{R^2-x^2}}\right)\Bigg]_0^{x_0}$$

$$=\left(a^4+6a^2R^2+R^4\right)x_0-\frac{2\left(3a^2+R^2\right)}{3}x_0{}^3+\frac{x_0{}^5}{5}+\sqrt{R^2-x_0{}^2}\left(-\frac{a}{2}\left(4a^2+5R^2\right)x_0+ax_0{}^3\right)-$$

$$\frac{aR^2\left(4a^2+3R^2\right)}{2}\tan^{-1}\left(\frac{x_0}{\sqrt{R^2-x_0{}^2}}\right)$$

$$B=\int_{x_0}^{x_1}\left(r_s{}^2-\left(x-x_{CG}\right)^2\right)^2dx$$

Substituting $\tilde{x}=x-x_{CG}$ and $d\tilde{x}=dx$ into the preceding equation yields a more manageable equation:

$$B=\int_{x_0-x_{CG}}^{x_1-x_{CG}}\left(r_s{}^2-\tilde{x}^2\right)^2d\tilde{x}=\int_{x_0-x_{CG}}^{x_1-x_{CG}}\left(r_s{}^4-2\tilde{x}^2r_s{}^2+\tilde{x}^4\right)d\tilde{x}=\left[r_s{}^4x-\frac{2\tilde{x}^3r_s{}^2}{3}+\frac{\tilde{x}^5}{5}\right]_{x_0-x_{CG}}^{x_1-x_{CG}}$$

$$=r_s{}^4\left(x_1-x_0\right)-\frac{2r_s{}^2}{3}\left(\left(x_1-x_{CG}\right)^3-\left(x_0-x_{CG}\right)^3\right)+\frac{\left(x_1-x_{CG}\right)^5-\left(x_0-x_{CG}\right)^5}{5}$$

$$I_x=\rho\pi\left(A+B\right)$$

Inspecting the equation for A, one can see that as $R \to \infty$, the term R^4x_1 increases at the fastest rate of any of the terms in A. Moreover, after examining the equations for B and (as developed next) I_y, this term increases at the fastest rate of any other term in the equations for I_x and I_y. This term acts to limit the ellipsoids that can be represented in a computer using floating point arithmetic. The relationship between the R^4x_1 term and the maximum representable double floating point number can be approximated through:

$$O(R^4x_1)<O(\max(double\_floating\_pt)).$$

This is the most computationally intense step in modeling of the ellipsoidal representation. It need only be performed once for each ellipsoidal particle (when it is initialized) and then the value can be stored in the data structure representing the ellipsoid. Other methods of approximating the moment of inertia are available and could be employed to avoid the constraints of the quadric term.

Moment of Inertia about the Axes Orthogonal to the Central Axis

Proceeding from the definition of the moment of inertia given at the beginning of this section:

$$I_y = \int_{-x_1}^{x_1} \left(\frac{r^2}{4} + x^2 \right) dm = \frac{I_x}{2} + \int_{-x_1}^{x_1} x^2 \rho\pi \cdot r(x)^2 dx = \frac{I_x}{2} + 2\rho\pi \cdot C$$

$$C = \int_0^{x_1} x^2 \cdot r(x)^2 dx = \int_0^{x_0} x^2 \left(\sqrt{R^2 - x^2} - a \right)^2 dx + \int_{x_0}^{x_1} x^2 \left(\sqrt{r_s^{\,2} - (x - x_{CG})^2} \right)^2 dx$$

$$= \int_0^{x_0} \left(R^2 x^2 - x^4 - 2x^2 a \sqrt{R^2 - x^2} + a^2 x^2 \right) dx + \int_{x_0 - CG}^{x_1 - CG} \left(r_s^{\,2} x^2 - x^4 \right) dx = C_1 + C_2$$

$$C_1 = \int_0^{x_0} \left(R^2 x^2 - x^4 - 2x^2 a \sqrt{R^2 - x^2} + a^2 x^2 \right) dx$$

$$= \left[\frac{\left(R^2 + a^2\right) x^3}{3} - \frac{x^5}{5} - 2a \left(\sqrt{R^2 - x^2} \cdot \left(\frac{x^3}{4} - \frac{R^2 x}{8} \right) + \frac{R^4}{8} \tan^{-1} \left(\frac{x}{\sqrt{R^2 - x^2}} \right) \right) \right]_0^{x_0}$$

$$= \frac{\left(R^2 + a^2\right) x_0^{\,3}}{3} - \frac{x_0^{\,5}}{5} - 2a \left(\sqrt{R^2 - x_0^{\,2}} \cdot \left(\frac{x_0^{\,3}}{4} - \frac{R^2 x_0}{8} \right) + \frac{R^4}{8} \tan^{-1} \left(\frac{x_0}{\sqrt{R^2 - x_0^{\,2}}} \right) \right)$$

$$C_2 = \int_{x_0}^{x_1} x^2 \left(r_s^{\,2} - (x - x_{CG})^2 \right) dx = \left[\frac{x^3 r_s^{\,2}}{3} - \left(\frac{x^5}{5} - \frac{x_{CG} \cdot x^4}{2} + \frac{x_{CG}^{\,2} \cdot x^3}{3} \right) \right]_{x_0}^{x_1}$$

$$= \frac{\left(x_1^{\,3} - x_0^{\,3}\right)\left(r_s^{\,2} - x_{CG}^{\,2}\right)}{3} - \left(\frac{\left(x_1^{\,5} - x_0^{\,5}\right)}{5} - \frac{x_{CG} \cdot \left(x_1^{\,4} - x_0^{\,4}\right)}{2} \right)$$

$$I_y = \frac{I_x}{2} + 2\rho\pi \cdot (C_1 + C_2)$$

Force Resolution

Given a rotational velocity vector, ω_1, and a translational velocity vector, v_1, in local coordinates S_1, the velocity at the contact point, c_1, in S_1 will be:

$$v_{c_1} = (c_1 \times \omega_1) + v_1 \qquad \text{(EQ 38)}$$

The velocity of body 1 with respect to 2 at the contact point, c, in the global coordinate system, S, will therefore be:

$$v_{c_{12}} = R_1 \cdot ((c_1 \times \omega_1) + v_1) - R_2 \cdot ((c_2 \times \omega_2) + v_2) \qquad \text{(EQ 39)}$$

Correspondingly, the velocity of body 2 with respect to 1 at the contact point, c, in the global coordinate system, S, is:

$$v_{c_{21}} = -v_{c_{12}}$$

This allows for the application of contact damping (coefficient of resolution) and friction. Using the equations developed for determining the normal at the contact point and the equations for determining the contact point, the normal force due to local plastic deformation can be specified as a penalty force. In this implementation the penalty force is based on Hertzian contact theory:

$F_{stiffness} = \left((\rho_1 + \rho_2) - \left\| R_1 \cdot \overset{r}{P}_1 \cdot \overline{CF}_1 - R_2 \cdot \overset{r}{P}_2 \cdot \overline{CF}_2 \right\|\right) \cdot \left(\frac{k_1 \cdot k_2}{k_1 + k_2}\right) \cdot N$, where ρ_i is the local curvature at the contact point in coordinate space S_i. The other forces can be resolved as:

$$F_{contact\_damping} = c_n \cdot \left(v_{c_{ji}} \cdot N_i\right) \cdot N_i$$

The equivalent normal contact damping in terms of coefficient of restitution can be given by:

$c_n = \dfrac{2\sqrt{k_n m_n} \cdot \ln\left(\frac{1}{e}\right)}{\sqrt{\pi^2 + \left(\ln\left(\frac{1}{e}\right)\right)^2}}$, where e is the coefficient of restitution.

$$F_{Coulomb\_friction}(t_{i+1}) = F_{cf} \cdot \left(\frac{\hat{n} \times (\hat{n} \times v_{c_{ij}})}{\left\|\hat{n} \times (\hat{n} \times v_{c_{ij}})\right\|}\right) \qquad \text{(EQ 40)}$$

where $F_{cf} = \min\begin{cases} \left\| F_{Coulomb\_friction}(t_i) \right\| + k_s \cdot \left\| \left(\hat{n} \times \left(\hat{n} \times v_{c_{ij}} \right) \right) \right\| \Delta t \\ \mu F_{elastic\_\lim it} \end{cases}$ and $\hat{n}$ is the unit normal vector to the surface, μ is the coefficient of friction, and $F_{elastic\_\lim it}$ is the elastic limit normal to the surface. F_{cf} enforces a Coulomb limit.

$k_s = \dfrac{k_{T_1} \cdot k_{T_2}}{k_{T_1} + k_{T_2}}$, where k_{T_i} is the stiffness tangent to the surface of object i.

Condensing the forces as moments and forces about the center of gravity gives:

$$M_{CoG} = \sum \left(c \times F_c \right) \qquad \text{(EQ 41)}$$

$$\theta(t_i) = I^{-1} M_{CoG}, \text{ where } I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \Rightarrow I^{-1} = \begin{bmatrix} I_x^{-1} & 0 & 0 \\ 0 & I_y^{-1} & 0 \\ 0 & 0 & I_z^{-1} \end{bmatrix}$$

$$F_{CoG} = \sum F_c$$

$$u(t_i) = \frac{F_{CoG}}{m}$$

From Rege (1996), a suitable explicit numerical integration method is given by:

$$a = 1 - \frac{\alpha \cdot dt}{2}$$

$$b = \frac{2}{2 + \alpha \cdot dt}$$

$$u(t_{i+1}) = \left(a u(t_i) + u(t_i) \cdot dt \right) b$$

$$\vec{\theta}(t_{i+1}) = \left(a \theta(t_i) + \theta(t_i) \cdot dt \right) b, \text{ where } \vec{\theta}(t_{i+1}) \equiv \omega(t_{i+1})$$

$$\vec{u}(t_{i+1}) = u(t_i) + u(t_{i+1}) \cdot dt$$

$$\theta(t_{i+1}) = \theta(t_i) + \theta(t_{i+1}) \cdot dt$$

Quaternions and Rotation

Quaternions were first discussed in Hamilton (1844) as a successor to the complex number. Quaternions are vectors with a real component and 3 orthogonal complex components. The complex space is defined along the three coordinate axes designated here as $\lfloor \hat{i} \quad \hat{j} \quad \hat{k} \rfloor$. Typically, quaternions are written in one of the following forms:

$$q = w + x\hat{i} + y\hat{j} + z\hat{k} = (s, v)$$

According to Shoemake (1985), a natural extension of quaternions to the rotation of rigid bodies exists and is readily achieved by setting:

$$q = \cos\left(\frac{\theta}{2}\right) + \cos\left(\frac{\theta}{2}\right) \cdot \left(u_x\hat{i} + u_y\hat{j} + u_z\hat{k}\right) = (\tilde{s}, \overset{r}{\tilde{v}})$$

where $\hat{u} = \lfloor u_x \quad u_y \quad u_z \rfloor$ is a unit vector in the direction of the axis of rotation and θ is the angle of rotation about $\hat{u}$. The rotation of a point, P, can then be calculated as:

$$P_{rotated} = q \cdot P \cdot q^T$$

where q is a unit quaternion. Carrying out the multiplication leads to the rotation matrix:

$$R(\tilde{s}, \tilde{v}) = \begin{bmatrix} \tilde{s}^2 + \tilde{v}_x^{\ 2} - \tilde{v}_y^{\ 2} - \tilde{v}_z^{\ 2} & 2 \cdot (\tilde{v}_x\tilde{v}_y + \tilde{s}\tilde{v}_z) & 2 \cdot (\tilde{v}_x\tilde{v}_z - \tilde{s}\tilde{v}_y) & 0 \\ 2 \cdot (\tilde{v}_x\tilde{v}_y - \tilde{s}\tilde{v}_z) & \tilde{s}^2 - \tilde{v}_x^{\ 2} + \tilde{v}_y^{\ 2} - \tilde{v}_z^{\ 2} & 2 \cdot (\tilde{v}_y\tilde{v}_z + \tilde{s}\tilde{v}_x) & 0 \\ 2 \cdot (\tilde{v}_x\tilde{v}_z + \tilde{s}\tilde{v}_y) & 2 \cdot (\tilde{v}_y\tilde{v}_z - \tilde{s}\tilde{v}_x) & \tilde{s}^2 - \tilde{v}_x^{\ 2} - \tilde{v}_y^{\ 2} + \tilde{v}_z^{\ 2} & 0 \\ 0 & 0 & 0 & \tilde{s}^2 + \tilde{v}_x^{\ 2} + \tilde{v}_y^{\ 2} + \tilde{v}_z^{\ 2} \end{bmatrix}$$

Because q is a unit quaternion, $R_{3,3} = 1$. This allows R to be written as:

$$R(\tilde{s}, \tilde{v}) = \begin{bmatrix} 1 - 2 \cdot \left(\tilde{v}_y^{\ 2} + \tilde{v}_z^{\ 2}\right) & 2 \cdot (\tilde{v}_x\tilde{v}_y + \tilde{s}\tilde{v}_z) & 2 \cdot (\tilde{v}_x\tilde{v}_z - \tilde{s}\tilde{v}_y) & 0 \\ 2 \cdot (\tilde{v}_x\tilde{v}_y - \tilde{s}\tilde{v}_z) & 1 - 2 \cdot \left(\tilde{v}_x^{\ 2} + \tilde{v}_z^{\ 2}\right) & 2 \cdot (\tilde{v}_y\tilde{v}_z + \tilde{s}\tilde{v}_x) & 0 \\ 2 \cdot (\tilde{v}_x\tilde{v}_z + \tilde{s}\tilde{v}_y) & 2 \cdot (\tilde{v}_y\tilde{v}_z - \tilde{s}\tilde{v}_x) & 1 - 2 \cdot \left(\tilde{v}_x^{\ 2} + \tilde{v}_y^{\ 2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note: Also, because rotation is a dot product and cross product preserving operation, R can be proven to belong to the special class of orthogonal matrices, an implication of which is $R^{-1} = R^T$.

Application to Rigid Body Motion

In the simulation environment, it is important to resolve rotations accurately. This can be accomplished through the use of the aforementioned quaternion approach or through the use of Euler angles. The difficulty with Euler angles is the order of application of rotations about coordinate axes. Is the ***i, j***, or ***k*** rotation applied first? The error introduced by order of application using Euler angles is small at the scale of rotations under discussion. That is, the value of the timestep, *dt*, is relatively small due to the constraints placed on it for the accuracy of the explicit numerical integration scheme.

It would, however, be best if we could both reduce the number of operations while simultaneously representing rotations of the rigid body in a time step more accurately. In the quaternion approach, only one transcendental function needs to be evaluated, versus two for Euler angles. Also, instead of producing the final rotation matrix through multiplication of three coordinate axis rotation matrices, the quaternion rotation matrix is formed at once without triple multiplication of 3x3 matrices. For this reason, the quaternion approach has been selected. Using the quaternion rotation equation above, the rotations can be applied as follows:

$$q = \cos\left(\frac{\theta}{2}\right) + \cos\left(\frac{\theta}{2}\right) \cdot \left(u_x\hat{i} + u_y\hat{j} + u_z\hat{k}\right) = (\tilde{s}, \vec{\tilde{v}})\text{, where } \theta = \|\omega\| \cdot dt \text{ and } \hat{u} = \frac{\omega}{\|\omega\|}$$

$$R_{GtoL}(t_{i+1}) = R_{GtoL}(t_i) \cdot R(\tilde{s}, \tilde{v}),\ R_{LtoG}(t_{i+1}) = \left[R_{GtoL}(t_{i+1})\right]^T$$

However, the elegance in the quaternion approach derives from the properties of the quaternion laid out in Shoemake (1985). By storing the quaternion instead of the rotation matrix and performing operations on the quaternions with the rotation matrix as a derived quantity generated only for contact resolution steps, the number of operations is reduced, the storage requirements for rotation are reduced, and the numerical accuracy of the resulting rotation after multiple rotations is improved.

Let

$q_1 = (s_1, v_1)$ and $q_2 = (s_2, v_2)$

Then, $q_1 \cdot q_2 = (s_1 \cdot s_2 - v_1 \cdot v_2, s_1 v_2 + s_2 v_1 + v_2 \times v_2)$

$$q(t_{i+1}) = q_{incremental}(t_{i+1}) \cdot q(t_i)$$

$$R_{GtoL}(t_{i+1}) = R(q(t_{i+1}))$$

Implementation

Program execution can be broken into several main functional parts:

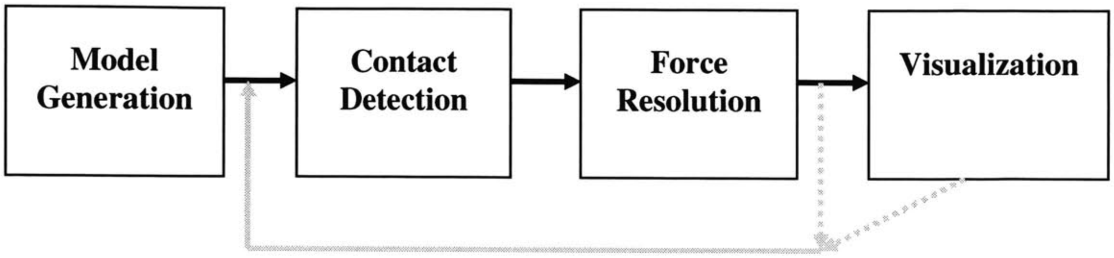


Figure 16: Program flow diagram

Model Generation can be accomplished in a variety of ways. The protype implementation described later uses a batch list approach with a proprietary data format for convenience of programming. However, other approaches include the use of proprietary graphical interfaces (Rege, 1996). In the final version of this design, off-the-shelf 3-D design formats will be used to describe the boundary and interaction surfaces. This will allow for the design of the model space through CAD software such as AutoCad™ and SolidWorks™.

Contact detection is involved in the determination of a contact between particles, including all aspects of the optimization of detecting contacts and restricting the number of detail checks executed during a timestep. Force resolution determines the dynamic behavior of each particle due to the forces applied during contact detection.

Visualization can be separated from the main time loop or it can be incorporated into it. This allows for flexible visualization intervals ranging from never visualize to visualize every timestep. This helps in the identification of trends in the simulation as the simulation progresses and also allows for the researcher to discover problems in the model without the necessity of progressing through an entire simulation.

Internal Data Schema

Several data structures have been determined for use in a simulation package. The main structures are identified here as:

***Root Classes*:**

- Material(double density, double viscous_damping, double contact_damping[3], double stiffness[6]);
- Shape(double volume, double CoG[3], double K_energy, double P_energy, double I[3], double v[3], double w[3])
- Force(string function);
- Contact(Shape* object1, Shape* object2, force, timestamp);
- Continuum(integer* boundary_list) ;

***Derived Classes*:**

- ISParticle: interface instituting SetKineticEnergy, SetMomentOfInertia, SetPotentialEnergy;
- Boundary(VertexList* vertices, EdgeList* edges): Shape, Particle

***Derived Shape Classes*:**

In this context, geometries and particle types can easily be added. The specification for the ellipsoid, sphere, and discretized surface particles can be given as:

- Ellipsoid(Quaternion quat, double length, double width): Shape
- EllipsoidParticle: Ellipsoid, ISParticle
- Sphere(double radius): Shape
- SphereParticle: Sphere, ISParticle
- Polyhedron(Quaternion quat, VertexList* vertices, EdgeList* edges): Shape
- PolyhedronParticle: Polyhedron, ISParticle
- DiscreteFunctionRepresentation(Quaternion quat, HeightFunction[2*nx,ny]): Shape
- DFRParticle: DiscreteFunctionRepresentation, ISParticle

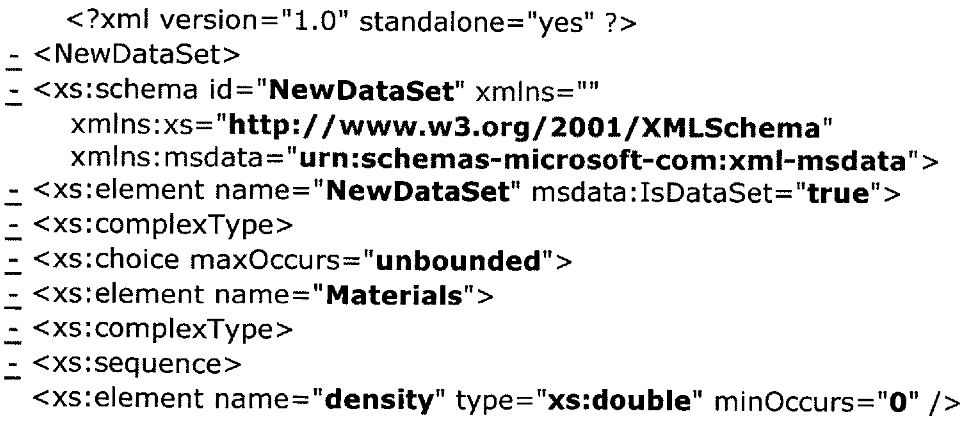
Contact detection is described in earlier sections. The general contact detection element would act to determine contact between two generalized entities. This would require a specific contact detection module for each type of contact, (e.g., a separate contact detection module would be required for sphere-sphere contact and ellipsoid-sphere contact). The necessary contact algorithms for the ellipsoid are described in this thesis in previous sections; however, other contact algorithms are beyond the scope of this thesis.

The persistent contact is necessary for calculation of Coulomb damping and other forces dependent on memory effects. The list records all contact forces applied at a certain time. The persistent contact is a database table with a primary key of: entity1, entity2, and timestamp. For simple Coulomb damping, the database entries with a timestamp of earlier than $t_{current}$-dt could be dropped at the beginning of timestep $t_{current}$. This approach is implemented in the Matlab prototype developed in this thesis. The timestep allows this data structure to be more flexible in retaining entries for 0 or more timesteps.

External Data Representation

Particular attention is given to the way the different entities in the model will be represented. To make this implementation flexible and interoperable with other engineering software packages, there is a need to attempt to standardize as much as possible the data schema used.

The first step in this is to adopt extensible markup language (XML) in describing the data schema. XML is a text-based format for generic data. XML itself does not provide a specific structure for data; rather, it provides a framework that one may use to structure data. XML is well-supported by libraries in Java, C++, and Microsoft's .NET framework, making it especially attractive for cross-application integration, which would allow for the DEM model to be imported to and exported from a variety of FEM, CAD, and CAM packages. XML is also well adapted to web service communication, since its text-based format can pass through the HTTP port (ports 80 and 8080). The text-only format precludes direct execution or insertion of malicious code to the host computer, allowing the administrator an interface to limit/filter the traffic entering the port. XML would be well-suited as a way of representing DEM data structures, offering a way to share DEM models between different DEM implementations.



```
  <?xml version="1.0" standalone="yes" ?>
- <NewDataSet>
- <xs:schema id="NewDataSet" xmlns=""
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
- <xs:element name="NewDataSet" msdata:IsDataSet="true">
- <xs:complexType>
- <xs:choice maxOccurs="unbounded">
- <xs:element name="Materials">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="density" type="xs:double" minOccurs="0" />
```

```
 <xs:element name="viscous_dampingx" type="xs:double" minOccurs="0" />
 <xs:element name="viscous_dampingy" type="xs:double" minOccurs="0" />
 <xs:element name="viscous_dampingz" type="xs:double" minOccurs="0" />
 <xs:element name="contact_dampingx" type="xs:double" minOccurs="0" />
 <xs:element name="contact_dampingy" type="xs:double" minOccurs="0" />
 <xs:element name="contact_dampingz" type="xs:double" minOccurs="0" />
 <xs:element name="tensile_stiffnessx" type="xs:double" minOccurs="0" />
 <xs:element name="tensile_stiffnessy" type="xs:double" minOccurs="0" />
 <xs:element name="tensile_stiffnessz" type="xs:double" minOccurs="0" />
 <xs:element name="compressive_stiffnessx" type="xs:double" minOccurs="0" />
 <xs:element name="compressive_stiffnessy" type="xs:double" minOccurs="0" />
 <xs:element name="compressive_stiffnessz" type="xs:double" minOccurs="0" />
 <xs:element name="friction_coefficientx" type="xs:double" minOccurs="0" />
 <xs:element name="friction_coefficienty" type="xs:double" minOccurs="0" />
 <xs:element name="friction_coefficientz" type="xs:double" minOccurs="0" />
   </xs:sequence>
   </xs:complexType>
   </xs:element>
- <xs:element name="Spheres">
- <xs:complexType>
- <xs:sequence>
 <xs:element name="radius" type="xs:double" minOccurs="0" />
 <xs:element name="CoGx" type="xs:double" minOccurs="0" />
 <xs:element name="CoGy" type="xs:double" minOccurs="0" />
 <xs:element name="CoGz" type="xs:double" minOccurs="0" />
 <xs:element name="velocityx" type="xs:double" minOccurs="0" />
 <xs:element name="velocityy" type="xs:double" minOccurs="0" />
 <xs:element name="velocityz" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityx" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityy" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityz" type="xs:double" minOccurs="0" />
 <xs:element name="forcex" type="xs:double" minOccurs="0" />
 <xs:element name="forcey" type="xs:double" minOccurs="0" />
 <xs:element name="forcez" type="xs:double" minOccurs="0" />
 <xs:element name="momentx" type="xs:double" minOccurs="0" />
 <xs:element name="momenty" type="xs:double" minOccurs="0" />
 <xs:element name="momentz" type="xs:double" minOccurs="0" />
 <xs:element name="matptr" type="xs:int" minOccurs="0" />
 <xs:element name="sphereResolution" type="xs:int" minOccurs="0" />
   </xs:sequence>
   </xs:complexType>
   </xs:element>
- <xs:element name="Ellipsoids">
- <xs:complexType>
- <xs:sequence>
 <xs:element name="a" type="xs:double" minOccurs="0" />
 <xs:element name="b" type="xs:double" minOccurs="0" />
 <xs:element name="CoGx" type="xs:double" minOccurs="0" />
 <xs:element name="CoGy" type="xs:double" minOccurs="0" />
 <xs:element name="CoGz" type="xs:double" minOccurs="0" />
 <xs:element name="rotationw" type="xs:double" minOccurs="0" />
 <xs:element name="rotationx" type="xs:double" minOccurs="0" />
 <xs:element name="rotationy" type="xs:double" minOccurs="0" />
 <xs:element name="rotationz" type="xs:double" minOccurs="0" />
 <xs:element name="velocityx" type="xs:double" minOccurs="0" />
 <xs:element name="velocityy" type="xs:double" minOccurs="0" />
 <xs:element name="velocityz" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityx" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityy" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityz" type="xs:double" minOccurs="0" />
 <xs:element name="forcex" type="xs:double" minOccurs="0" />
 <xs:element name="forcey" type="xs:double" minOccurs="0" />
 <xs:element name="forcez" type="xs:double" minOccurs="0" />
 <xs:element name="momentx" type="xs:double" minOccurs="0" />
 <xs:element name="momenty" type="xs:double" minOccurs="0" />
 <xs:element name="momentz" type="xs:double" minOccurs="0" />
 <xs:element name="matptr" type="xs:int" minOccurs="0" />
 <xs:element name="ellipsoidResolutionx" type="xs:int" minOccurs="0" />
 <xs:element name="ellipsoidResolutiont" type="xs:int" minOccurs="0" />
   </xs:sequence>
   </xs:complexType>
   </xs:element>
- <xs:element name="Planes">
- <xs:complexType>
- <xs:sequence>
 <xs:element name="CoGx" type="xs:double" minOccurs="0" />
 <xs:element name="CoGy" type="xs:double" minOccurs="0" />
 <xs:element name="CoGz" type="xs:double" minOccurs="0" />
 <xs:element name="rotationw" type="xs:double" minOccurs="0" />
 <xs:element name="rotationx" type="xs:double" minOccurs="0" />
 <xs:element name="rotationy" type="xs:double" minOccurs="0" />
 <xs:element name="rotationz" type="xs:double" minOccurs="0" />
 <xs:element name="velocityx" type="xs:double" minOccurs="0" />
 <xs:element name="velocityy" type="xs:double" minOccurs="0" />
 <xs:element name="velocityz" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityx" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityy" type="xs:double" minOccurs="0" />
 <xs:element name="rotationVelocityz" type="xs:double" minOccurs="0" />
 <xs:element name="forcex" type="xs:double" minOccurs="0" />
 <xs:element name="forcey" type="xs:double" minOccurs="0" />
 <xs:element name="forcez" type="xs:double" minOccurs="0" />
 <xs:element name="momentx" type="xs:double" minOccurs="0" />
 <xs:element name="momenty" type="xs:double" minOccurs="0" />
 <xs:element name="momentz" type="xs:double" minOccurs="0" />
 <xs:element name="matptr" type="xs:int" minOccurs="0" />
   </xs:sequence>
   </xs:complexType>
   </xs:element>
```

```
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
```

Due to the transience of schema standards (e.g., the lack of agreement on the X3D standard) and the lack of consistency between competing standards, the schema developed here has been kept purposefully high-level. This allows for ease of context mediation between this schema and other 3-D schemas being forwarded as standards. The richness of a general 3-D schema is unnecessary for the purposes of a particle simulation, and it is beneficial to avoid such unnecessary attributes in our representation. By using a less rich schema, translation to other schemas is made much simpler.

Prototype

A prototype system has been developed using Matlab v6.5 to test the operation of the ellipsoidal representation. The implementation consists of three separate modules: model generation, contact detection, numerical simulation, and visualization.

These 3 modules have been used to test the robustness of the contact detection algorithm and verify the numerical integration techniques by observing momentum transfer between two particles. The results of these simulations are given in the following sections.

Validation of Normal Resolution

The algorithm computes a contact normal for each body. When converged, these normals should be coincident and parallel. The error in the algorithm is defined as the angle between these normals. In empirical tests, the rate of convergence is exponential, such that:

$$\left\{A \in \left(0 \quad 0.4928\right) \middle| \, error = A^{n}\left(initial\_guess\right)\right\} \qquad \text{(EQ 42)}$$

where the normals are defined by Equation 45.

$$error = \cos^{-1}\left(-1 \cdot N_i\big|_{global} \bullet N_j\big|_{global}\right) \qquad \text{(EQ 43)}$$

$$N_i\big|_{local} = \left[\frac{\left(\tilde{x}_i \quad 0 \quad 0\right) - \vec{P}_i \cdot \overline{CF}}{\left|\left(\tilde{x}_i \quad 0 \quad 0\right) - \vec{P}_i \cdot \overline{CF}\right|}\right]_{local} \qquad \text{(EQ 44)}$$

$$N_i\big|_{global} = R_i^{T}\left(N_i\big|_{local}\right) \qquad \text{(EQ 45)}$$

where R_i is the global-to-local rotation matrix for ellipsoid i. Note that R_i is an orthogonal matrix, so the $R_i^{-1} = R_i^{T}$. $\tilde{x}_i$ is the point along the ellipsoid's major axis through which the line of contact passes. By applying the constraint that the normal pass through the major axis and that the line pass through the path traced by $\left[r(0,\theta) = \alpha\right]_{local}^{i}$, which is enforced through Equation 44, a valid evaluation of contact is achieved.

For all trials, convergence is achieved, and rate of convergence within each contact detection step remained constant to within ±0.5%. Assuming worst case convergence of 0.493 (based on empirical studies) and a worst case algorithm starting point (11.5 degrees error in the normal vector skew), then convergence would be bounded by $error_{worst} = 0.493^{n}(11.5)$. Best case algorithm starting point (0.00031 degrees error in the normal vector skew) and best case convergence of 0 (based on empirical studies), then the convergence would be bounded by $error_{best} = 0^{n}(0.00031) = 0$. For example, applying 5 iterations would be bounded by $\left\{error \,\middle|\, error \in \left(0 \quad 0.335\right)\right\}$, where error is in units of degrees and represents the skew of the calculated normals given in Equation 45.

Conclusion

DEM methods provide a micro-mechanical basis for evaluating large, complex systems of particles, capturing the emergent macro behavior of such systems. This is especially of interest to the mining, petroleum, and pharmaceutical industries. The mining industry routinely encounters particulate matter generated in the production of usable ores. In the petroleum industry, local failure of the sand in the near well-bore region can and often does lead to a phenomenon known as "sanding" where the initiated failure of the well-bore spreads outward, causing catastrophic collapse of the hole and possibly destroying the usefulness of the well and damage drilling machinery. In the pharmaceutical industry, a poor understanding of powder mixing has led to costly quality control problems, where incompletely mixed pharmaceuticals have caused individual doses to exceed tolerances set by federal regulation, resulting in costly fines.

These costly and important problems give discrete element modeling a high degree of relevance for research. Unfortunately, DEM methods have been plagued by insufficient computing resources available for the types of problems being investigated. Much research has been invested in streamlining discrete element methods and in offering implementations that are both accurate at the scale of interest and computationally efficient enough to be run on PCs. The ellipsoidal geometry described here attempts to achieve both efficiency and accuracy by capturing basic interlocking and resistance to rotation that deeply affect the micromechanical behavior of real materials.

This thesis also describes a new contact detection algorithm for DEM analysis that approximates ellipsoids as partial spheres that are oriented such that in the plane of contact the spheres are matched to maintain C^1 continuity – this is called here the Method of Equivalent Spheres. The representation maintains the simplicity of clustered spheres without the drawback of discontinuous normals. The algorithm for determining the contact point and the unique normal is described and empirical convergence properties demonstrated. The Equivalent Sphere Ellipsoid is implemented alongside planes (including planar regions such as polyhedral facets) and spheres to allow a deeper richness to a particulate system model than ellipsoids alone can afford.

References

Barr, A. (1981) "Superquadrics and angle-preserving transformation", IEEE Computer Graphics Applications, CGA-1, Vol 1, pp. 11-23.

Beer, F.P, and Johnston, E.R. (1996) "Vector Mechanics for Engineers: Statics," NY: McGraw-Hill, p. 500.

Bridgewater, J. (2002). "Understanding how to process particulate matter" Lectures in Chemical Engineering, Massachusetts Institute of Technology, April 30, 2002.

Cundall, P.A. and Strack, O.D.L. (1979) "A distinct element model for granular assemblies." Geotechnique, Vol. 29, No. 1, pp. 47-65.

Cleary, P.W. (1998) "Discrete element modeling of industrial granular flow applications", TASK. Quarterly – Scientific Bulletin, Vol 2, pp. 385-416.

Cook, B.K. and Jensen, R.P. (2002) Proceedings of the 3rd International Conference on Discrete Element Methods – Numerical Modeling of Discontinua, Sept. 23-25, ASCE Geotechnical Special Publication No. 117

Cundall, P.A. (1987) "Computer simulations of dense sphere assemblies", Proceedings of the U.S./Japan Seminar on the Micromechanics of Granular Materials, Sendai-Zao, Japan, October 26-30, 1987.

Cundall, P. A., Konietzky, H., and Potyondy, D. O. (1996) "PFC Ein Neues Werkzeug für Numerische Modellierungen", Bautechnik, Vol 73, No 8, pp. 492-498.

Dobkin, D., and Kirkpatrick, D., (1990) "Determining the separation of preprocessed polyhedra – a unifed approach." Lecture Notes in Computer Science (ICALP-90), Vol. 443, pp. 400-413.

Favier, J.F., Abbaspour-Fard, M.H., Kremmer, H., and Raji, A.O. (1999) "Shape representation of axi-symmetric, non-spherical particles in discrete element simulation using multi-element model particles", Engineering Computations, Vol. 16, No. 4, pp. 467-480.

Favier, J.F., Abbaspour-Fard, M.H., and Kremmer (2001) "Modeling nonspherical particles using multisphere discrete elements", Journal of Engineering Mechanics, Vol. 127, No. 10, pp. 971-977.

Grest, G., Landry, J., Silbert, L., and Plimpton, S. (2001) "Rheology of granular flow", Lecture at the Kavli Institute for Theoretical Physics, Santa Barbera, CA, 4.June, 2001

Hamilton, W.R. (1844) "On quaternions; or on a new system of imaginaries in algebra," Philosophical Magazine, XXV, pp. 10-13.

Hopkins, M.A. (1997) "Onshore ice pile-up: a comparison between experiments and simulations." Cold Regions Science and Technology, Vol. 26, No. 3, pp. 205-214.

Hopkins, M.A. (1999) "Compression of floating ice fields." Journal of Geophysical Research, Vol 104, No. C7, pp. 15815-15825.

Hopkins, M.A. (2002) "Discrete element modeling based on a mathematical morphology" Third International Conference on Discrete Element Modeling, Santa Fe, NM, September 22-25.

Jing, L. (2000) "Block system construction for three-dimensional discrete element models of fractured rocks." International Journal of Rock Mechanics and Mining Sciences, Vol. 37.

Johnson, S.M. and Williams, J.R. (2002) "Formation of Packing Structures in Discrete Element Modeling with Disks." Third International Conference on Discrete Element Modeling, Santa Fe, NM, September 22-25.

Johnson, S.M., Williams, J.R., and Cook, B.K. (2003) "Contact detection algorithm for an ellipsoid approximation for discrete element modeling." Submitted to Engineering Computations, March.

Kuhn, M.R. (1993) "A flexible boundary for three-dimensional DEM particle assemblies." Second International Conference on Discrete Element Methods. Cambridge, MA.

Lin, X., and Ng, T-T. (1997) "A three-dimensional discrete element model using arrays of ellipsoids", Géotechnique, Vol 47 No 2, pp. 319-329.

Mustoe, G.G.W., Henriksen, M., and Huttelmaier, H.-P. (1989) Proceedings of the 1st International Conference on Discrete Element Methods, Colorado School of Mines, Golden, Colorado, CSM Press.

Mustoe, G.G.W. (2001) "Cluster2D Discrete Element Computer Code – Users Manual and 2-D Discrete Element Workshop Notes", http://egweb.mines.edu/gmustoe/eg535/main_EGES535.htm.

Munjiza, A., Bicanic, N., and Owen, D.R.J. (1993) "BSD contact detection algorithm for discrete elements in 2D", Proceedings of the 2nd International Conference on Discrete Element Methods, Massachusetts Institute of Technology, March 18-19, IESL Publications.

Munjiza, A., Owen, DRJ, and Bicanic, N. (1995) "A combined finite-discrete element method in transient dynamics of fracturing solids." International Journal of Engineering Computation, Vol. 12, pp. 15-174.

Munjiza, A., and Andrews, K.R.F. (1998) "NBS contact detection algorithm for bodies of similar size", International Journal for Numerical Methods in Engineering, Vol 43, pp. 131-149.

Munjiza, A., and John, N. (2002a) "Mesh size sensitivity of the combined FEM/DEM fracture and fragmentation algorithms." Engineering Fracture Mechanics, Vol. 69, pp. 281-295.

Munjiza, A., and Andrews, K. (2002b) "Penalty function method for combined finite-discrete element systems comprising a large number of separate bodies." International Journal for Numerical Methods in Engineering, Vol. 49, pp. 1377-1396.

Ng, T-T. and Fang, H.E. (1995) "Cyclic behavior of arrays of ellipsoids with different particle shapes", In Proceedings of the Joint ASME Applied Mechanics and Materials Summer Conference, Mechanics of Materials with Discontinuities and Heterogeneities Symposium, UCLA, Los Angeles, CA, Vol. 201, pp. 59-70.

O'Connor, R.M., and Williams, J.R. (1996) "A three dimensional geometric representation scheme for contact detection in discrete element simulation," International Journal Computer Aided Engineering & Software Engineering Computation.

O'Connor, R.M. (1996) "A distributed discrete element modeling environment: algorithms, implementation and applications" Thesis (Sc. D.)--Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering.

Ouadfel, H., and Rothenburg, L. (1999) "An algorithm for detecting inter-ellipsoid contacts." Computers and Geotechnics, Vol. 24, pp. 245-263.

Perkins, E. and Williams, J.R. (2001) "A fast contact detection algorithm insensitive to object sizes", International Journal for Computer-Aided Engineering, Vol 18, No 1, pp. 48-62.

Pentland, A. (1987), "Perceptual organization and the representation of natural form", in: Fischler, M.A. & Firschein, O. (eds) Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, Morgan Kaufman, San Francisco.

Peter, J. (2000) "Analytical versus voxelized phantom representation for monte carlo simulation in radiological imaging", IEEE Transactions on Medical Imaging, Vol 19 No 5, pp. 556-564.

Rege, N. V. (1996) "Computational modeling of granular materials." PhD. Thesis: Massachusetts Institute of Technology, Department of Civil and Environmental Engineering.

Shoemake, K. (1985) "Animating rotations with quaternion curves," Computer Graphics, Proceedings of SIGGRAPH, Vol. 19, No. 3, pp. 245-254.

Sramek, M. (2001) "High precision non-binary voxelization of geometric objects." IEEE 0-7695-1215-1/01.

Taylor L.M., Preece D.S., (1992) "Simulation of blasting induced rock motion using spherical element models", Engineering Computations, Vol 9, No 2.

Tijskens, E., Ramon, H., and De Baerdemaeker, J. (2002) "Contact resolution for general level surfaces, using automatic differentiation." Third International Conference on Discrete Element Methods, Santa Fe, NM, USA, September 22-25.

Ting, J.M., Khwaja, M., Meachum, L.R., and Rowell, J.D. (1993) "An ellipse-based discrete element model for granular materials", International Journal for Numerical and Analytical Methods in Geomechanics, Vol 17, pp. 603-623.

Vu-Quoc, L., and Zhang, X. (1999) "An accurate and efficient tangential force-displacement model for elastic frictional contact in particle-flow simulations", Mechanics of Materials, Vol 31, pp. 235-269.

Vu-Quoc, L., Zhang, X., and Walton, O.R. (2000) "A 3-D discrete element method for dry granular flows of ellipsoidal particles", Computational Methods in Applied Mechanical Engineering, Vol 187, pp. 483-528.

Wang, C.-Y., and Liang, V.-C. (1997) "A packing generation scheme for the granular assemblies with planar elliptical particles" International Journal for Numerical and Analytical Methods in Geomechanics, Vol 21, pp. 347-358/

Wang, C-Y, Wang, C-F, and Sheng, J. (1999) "A packing generation scheme for the granular assemblies with 3D ellipsoidal particles", International Journal for Analytical Methods in Geomechanics, Vol 25, pp. 815-828.

Watanabe, H. (1999). "Critical rotation speed for ball-milling." *Powder Technology*, 104, 95-99.

Wensel, O., and Bicanic, N. (1993) "A quad tree based contact detection algorithm", Proceedings of the 2nd International Conference on Discrete Element Methods, Massachusetts Institute of Technology, March 18-19, IESL Publications.

Williams, J.R., and Pentland, A.P. (1992) "Superquadrics and modal dynamics for discrete elements in interactive design." International Journal of Computer-Aided Engineering, Engineering Computations, Vol 9, No 2.

Williams, J.R. and Amaratunga, K. (1993) "Wavelet representation of geometry for analysis" Proceedings of the 2nd International Conference on Discrete Element Methods, Massachusetts Institute of Technology, March 18-19, IESL Publications

Williams, J.R., and O'Connor, R. (1995) "A linear complexity intersection algorithm for discrete element simulation of arbitrary geometries", International Journal of Computer-Aided Engineering, Engineering Computations, Vol 12, No 2.

Zhou, Y. C., Xu, B. H. , Yu A. B., and Zulli, P. (2002). "An experimental and numerical study of the angle of repose of coarse spheres," *Powder Technology*, 125(1), 45-54.

Acknowledgements:

I wish to thank Sandia National Laboratory for their financial support, which has made this work possible. I would like to especially thank Dr. Benjamin Cook at Sandia National Laboratory for all of his advice on both this work and everything else and Professor John Williams for sharing his knowledge on discrete element methods.

This work is the result of collaboration between Sandia National Laboratories, Albuquerque, NM, and the Massachusetts Institute of Technology, Cambridge, MA. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.