Non Spherical Particles

When the particles were assumed spherical with smooth surface, computational costs associated with representing their geometries, tracking them and finding the contact plane/point are low.

However many practical granular flows involve non-spherical particles that may be near round or with irregular shapes.

Using non-spherical in the DEM as compared with spherical particles, requires additional computational costs due to implementing more complex algorithms for contact detection, tracking particles and contact force calculations. This is compulsory for performing accurate DEM simulation, even though the costs are high.

Shape Representation:

In the implementation of DEM for non-spherical particles, four issues should be addressed:

1. The kinematics of the particle
2. The method of representing particle shape to DEM
3. Calculation of the contact force
4. Finding the contact plane between particles.

1. Equation of rotational motion of a non-spherical particle is different from that of a sphere. A method should be therefore considered for defining particle orientation in 3D space and solving Euler's equations.

1. A sphere can be represented by its center point and radius in the computer, while shape representation of a non-spherical particle (exact or approximate) requires additional data.

1. Shape representation is still one of the key challenges in the DEM modeling of granular flows.

1. According to the way that particle shape is represented, the method of finding a contact plane between particles and force calculation should be changed.

Shape Representation:

A variety of methods for representing particle shapes can be found in the literature. They are categorized into two main groups:

1. Single-element particle approach
2. Method of Intersecting Shapes
3. Multi-element particle approach
4. Polyhedral Approach

Single-Element Particle Approach:

1. In the single-element approach, the surface of the particle is defined by a set of continuous analytical equations. In this way, a particle with smooth surface and no bump is obtained.

1. Using a superellipsoid (or in a more general term, superquadrics) is the most important method in this group.

1. A superellipsoid is a smooth surface defined by an implicit equation, in form of $F(x, y, z) = 0$ in 3D space. By changing the parameters in this equation, different shapes with a closed surface can be obtained.

1. Superellipsoids have been used in various DEM simulation of granular flows like hopper discharge [3, 8], flow pattern of spherical and non-spherical particles in screw feeders [9] and gas fluidization of ellipsoidal particles [5, 10].

1. Although this method can provide an accurate estimation of the particle surface, not all shapes can be represented by this method. Especially, a particle with sharp edges cannot be represented by this family of equations.

Single Element Approach:

Perhaps the most challenging issue associated with the use of the family of super-ellipsoids is detection of contact between two particles.

This requires solving a set of non-linear equations numerically by methods like the Newton's method or the secant method.

Usually, initial guess for starting the iterative solution affects number of iterations required to find the contact plane. Sometimes, the iterative solution may even diverge. Therefore, a suitable algorithm for determining the initial guess should be used.

The problem with the initial guess and larger number of iterations become serious when particles first come into contact.

In the next time steps, since particles do not move significantly relative to each other, the initial guess would be the contact plane in the previous time step and a fewer number of iterations would be required.

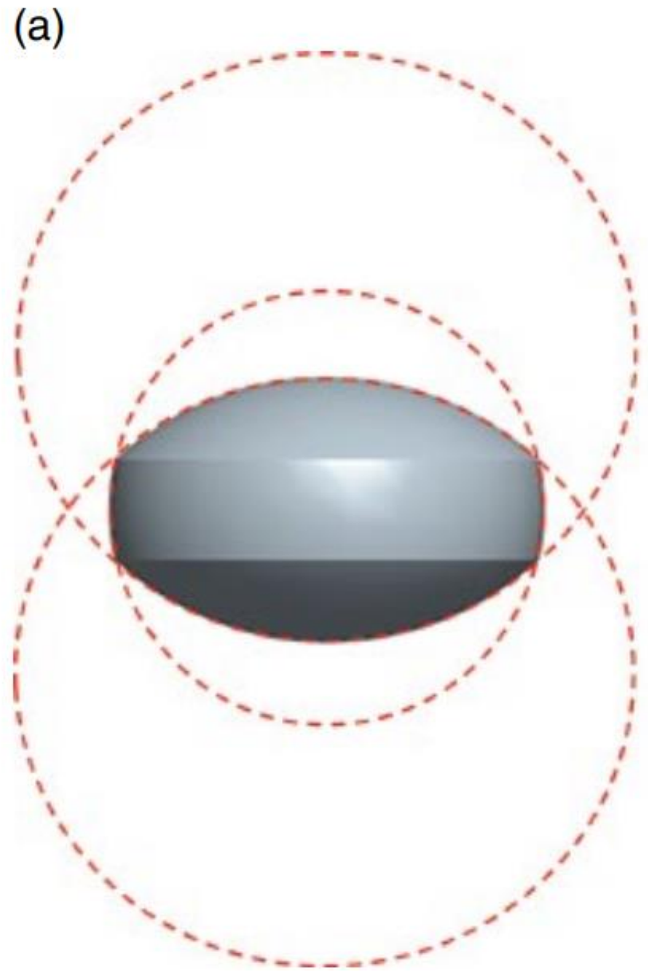
In some regions of surface where the slope is gentle, the iterative method needs extra iterations to converge into the solution and in regions with a steep slope, the solution may diverge and relaxation may be required.

Method of Intersecting Shapes:

Smoothed surface of particles can be resembled by intersecting surfaces of spheres or other simple geometric shapes. For Example, the surface of a standard round tablet with three spherical surfaces.

The advantage of the method of intersecting shapes is obtaining a smooth surface with edges, which accurately represents the real surface.

However, from one shape to another, the method for representing the shape and contact detection algorithm should be changed.



Multi-Element Approach:

In the multi-element particle approach, complex surface of the particle is constructed from smaller sub-elements (with simple shapes) that are connected to each other.

Among different methods in this group, the multi-sphere method, also called glued spheres or clustered spheres, is one of the most common one. The complex surface of a particle can be formed by overlapped spheres.

By increasing the number of spheres used in the construction of the complex particle, the real surface would be represented more accurately.

However, regardless of the number of spheres, the resultant surface would be bumpy and without a sharp edge or vertex

A DEM code for spherical particle can be easily extended to handle multi-sphere method. The contact search and contact force calculations are similar, although calculation of force needs some modifications.

The multi-sphere method is not limited to a particular shape or a family of geometric shapes and theoretically can produce any complex surface, except surfaces with sharp edges.

By increasing the number of spheres in a particle, the particle surface is represented more accurately, while at the same time it increases the computational costs and errors due to multiple contact points.

Large number of spheres

Small number of spheres

Polyhedral Approach:

Particles with irregular shapes and sharp edges can be generated by the polyhedral method. Surface of a polyhedral particle consists of triangular sub-elements. Other polygons also can be used instead of triangle.

These triangles can be rigidly connected to each other to create a rigid body.

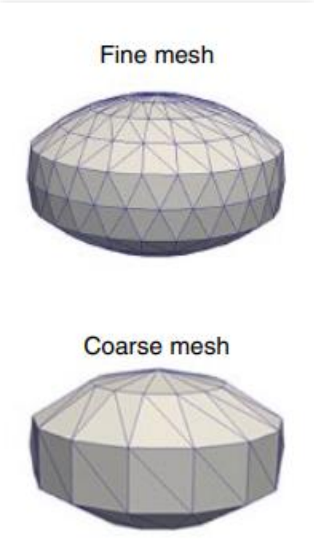
Alternatively, they can be connected to each other by elastic, deformable, dissipative bounds to create deformable shapes.

The shape of a deformable particle can be changed by external forces and bending moments acting on it.

By increasing the number of sub-elements, the surface of a real particle is more accurately approximated, but it needs higher computational resources for storing vertices and face data in the memory and performing contact detection tests between triangles that belong to different particles.

The polyhedral method is very similar to the multi-sphere method. Interaction between two particles is calculated based on the interaction of triangles that constitute these particles while in the multi-sphere method this is done through the constituent spheres.

The differences between polyhedral and multi-sphere contact detec and calculation of force between sub-elements. number of algorithms have been suggested for con and force calculations.



Kinematics and Dynamics of Rigid Bodies:

A non-spherical particle may be a single body whose surface is defined by analytical equations or a body composed of sub-elements that are rigidly connected to each other.

Therefore, the basic principles of the kinematics and dynamics of the non-spherical rigid body apply to both groups.

- The position of a non-spherical rigid body in the space is defined by the location of its center of mass and its orientation.
- In Cartesian coordinates, two different coordinates systems are required. First, an orthogonal body-fixed frame (local coordinates) located on the center of mass and moves and rotates with the rigid body.1 Second, an orthogonal space-fixed frame (global coordinates) that is inertial and it does not move or rotate in the space.
- In a DEM simulation, all variable such as position, orientation, and velocities of the particle, as well as location of walls, are defined in this space-fixed frame.
- The transformation of vector variables between the body-fixed and space-fixed frames are frequently performed to compute contact force and torque as well as to solve equations of motion.
- The location and orientation of the body-fixed frame (hence the body itself) is defined relative to the space-fixed frame. Therefore, a convention should be used to transform variables from the space-fixed frame to the body-fixed frame and vice versa.

Euler Angles and Transformation Matrix:

In 2D space, three scalar variables (three degrees of freedom) are required to define the rigid body motion: two for coordinates of the center of mass and one for the rotation angle θ.

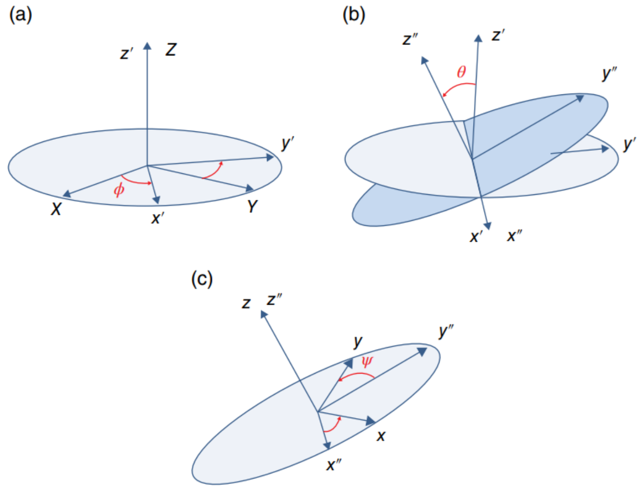
The rotation by the angle θ around the center of mass is simply done using the following rotation matrix:

$$A_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

In 3D space, six scalar variables (six degrees of freedom) are required for defining the rigid body motion: three for the coordinates of the center of mass and three angles for

There is a variety of methods to define the orientation of a body in 3D space. The most well-known is by using Euler angles.

According to the classical Euler angles, the orientation of the body-fixed frame is defined by three Euler angles ϕ, θ, and ψ.



Euler Angles:

Consider the spaced-fixed frame XYZ. It is desirable to perform a set of rotations to transform this frame to the body-fixed frame xyz. For this purpose, the following steps should be carried out in sequence:

1. First, a rotation is done by ϕ around Z-axis that results in a new coordinate system $x'y'z'$.

1. Then, in the $x'y'z'$ coordinates system, the second rotation is done by θ around x'-axis, which gives a new coordinate system $x''y''z''$.

1. Finally, in the $x''y''z''$ coordinates system, the third rotation is done by ψ around z''-axis and gives the body-fixed frame coordinates xyz.

The rotation matrices for these steps are as follows:

$$A_\phi = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad 0 \le \phi < 2\pi$$

$$A_\theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \qquad 0 \le \theta \le \pi$$

$$A_\psi = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad 0 \le \psi < 2\pi$$

In general, one can combine these three rotations into a single rotation by the following matrix:

$$\begin{aligned} A &= A_\psi A_\theta A_\phi \\ &= \begin{bmatrix} \cos\phi\cos\psi - \sin\phi\cos\theta\sin\psi & \sin\phi\cos\psi + \cos\phi\cos\theta\sin\psi & \sin\theta\sin\psi \\ -\cos\phi\sin\psi - \sin\phi\cos\theta\cos\psi & -\sin\phi\sin\psi + \cos\phi\cos\theta\cos\psi & \sin\theta\cos\psi \\ \sin\phi\sin\theta & -\cos\phi\sin\theta & \cos\theta \end{bmatrix} \end{aligned} \qquad (4.3)$$

Euler Angles:

A is the transformation matrix that transforms an arbitrary vector **xs** in the space-fixed frame to **xb** in the body-fixed frame by the following equation:

$$\vec{x}^b = A\vec{x}^s \tag{4.4}$$

The reverse of this rotation can be done to transform an arbitrary vector $\vec{x}^b$ from the body-fixed frame to the space-fixed frame. This is done by multiplying two sides of Equation 4.4 by A^{-1}:

$$\vec{x}^s = A^{-1}\vec{x}^b \tag{4.5}$$

Since the rotation matrix is orthogonal, $A^{-1} = A^T$. Thus, we can use the transpose of A instead of its inverse, which is less computationally intensive.

Each element of the transformation matrix A is a direct cosine of the angle between an axis of the body-fixed frame and the space-fixed frame.

Therefore, the transformation matrix also describes the relative orientation of the two-coordinate systems with nine parameters. For instance, the first column of A, transforms the unit vector i of the space-fixed frame to the body-fixed frame. Similarly, the second column transforms the unit vector j and the third column, the unit vector k

We are also interested in knowing the relation between the angular velocity of body in the body-fixed frame and incremental change (time derivatives) of Euler angles. The absolute angular velocity in the body-fixed frame is the vector sum of time derivatives change of Euler angles:

$$\omega^b = \dot{\phi} + \dot{\theta} + \dot{\psi} \tag{4.6}$$

It is relatively easy to show that [31]:

$$\omega_x^b = \dot{\phi}\sin\theta\sin\psi + \dot{\theta}\cos\psi \tag{4.7a}$$

$$\omega_y^b = \dot{\phi}\sin\theta\cos\psi - \dot{\theta}\sin\psi \tag{4.7b}$$

$$\omega_z^b = \dot{\psi} + \dot{\phi}\cos\theta \tag{4.7c}$$

If we solve these three equations for $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$, we get time derivatives of Euler angles in terms of fixed-body angular velocity:

$$\dot{\phi} = \frac{\omega_x^b \sin\psi + \omega_y^b \cos\psi}{\sin\theta} \tag{4.8a}$$

$$\dot{\theta} = \omega_x^b \cos\psi - \omega_y^b \sin\psi \tag{4.8b}$$

$$\dot{\psi} = \omega_z^b - \dot{\phi}\cos\theta \tag{4.8c}$$

Equations 4.8a–4.8c are important relations, since the angular equation of motion is solved in the orthogonal body-fixed frame and ωb is obtained, while non-orthogonal Euler angles are used to represent the orientation. Thus, these equations are used to obtain new orientation of the body in each time-step of simulation by integrating time derivatives of Euler angles.

Equations of Motion:

The motion of a body in 3D space is defined by translational motion of the center of mass and rotational motion about the center of mass.

The general form of the translational equation of motion of the center mass is:

$$m_i \vec{a}_i = m_i \frac{d\vec{v}_i}{dt} = \vec{f}_i$$

Where 'f' is the sum of forces acting on the center of mass of the body in the space-fixed frame and 'v' is the translational velocity of the center of mass in the space-fixed frame.

We can solve the translational motion of the center of mass of a non-spherical body similar to a spherical particle.

Rotational Motion of a 3D body:

The rotational motion of a rigid body in 3D space is given by:

$$\dot{\vec{L}} = \vec{M} \tag{4.10}$$

where $\dot{\vec{L}}$ is the rate of change of angular momentum and $\vec{M}$ is sum of torques acting on a particle. The reference point to evaluate torques and angular momentum should be similar and it can be any arbitrary point that moves with the center of mass of the body. It is more common to choose the center of mass as the reference point. Then, the angular momentum reads as:

$$\vec{L}^b = I\,\vec{\omega}^b \tag{4.11}$$

where I is the inertia tensor of the body at its center of mass and is defined as:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yx} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{4.12}$$

Diagonal elements of this matrix are called centroidal mass moment of inertia and off- diagonal elements are called centroidal mass product of inertia.

Rotational Motion (Euler's Equations):

The derivative of angular momentum of the rigid body is obtained by:

$$\dot{\vec{L}}^b = I\dot{\vec{\omega}}^b + \vec{\omega}^b \times I\vec{\omega}^b$$

Therefore, the equation of rotational motion around its center of mass in the body-fixed frame becomes:

$$\vec{M}^b = I\dot{\vec{\omega}}^b + \vec{\omega}^b \times I\vec{\omega}^b$$

and in matrix form:

$$\begin{bmatrix} M_x^b \\ M_y^b \\ M_z^b \end{bmatrix} = I \begin{bmatrix} \dot{\omega}_x^b \\ \dot{\omega}_y^b \\ \dot{\omega}_z^b \end{bmatrix} + \begin{bmatrix} 0 & -\omega_z^b & \omega_y^b \\ \omega_z^b & 0 & -\omega_x^b \\ -\omega_y^b & \omega_x^b & 0 \end{bmatrix} I \begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix}$$

This set of equations should be solved to obtain the rotational velocity of body in the fixed- body frame.

If the reference point is kept on the center of mass and the body-fixed coordinates is aligned to coincide with principal axes of the body, then the off-diagonal elements of the inertia tensor vanish and the given matrix equation turns into well known Euler's Equations:

$$M_x^b = \hat{I}_1\dot{\omega}_x^b + \left(\hat{I}_3 - \hat{I}_2\right)\omega_y^b\omega_z^b$$

$$M_y^b = \hat{I}_2\dot{\omega}_y^b + \left(\hat{I}_1 - \hat{I}_3\right)\omega_x^b\omega_z^b$$

$$M_z^b = \hat{I}_3\dot{\omega}_z^b + \left(\hat{I}_2 - \hat{I}_1\right)\omega_x^b\omega_y^b$$

where Î 1 , Î 2 , and Î 3 are principal central moments of inertia about the center mass of body.

Since Euler's equations are simpler and involve less non-linear terms, we choose the orientation of the body-fixed frame in a way that it coincides with the principal axes of the body.

Rotational Motion:

1. Method of Calculating the new orientation of the non spherical particle by calculating the derivatives of Euler angles and then Using Numerical Integration to compute the new orientation cannot be used because of the problem of singularity.

1. We discuss two methods here: using the transformation matrix to define the orientation instead of Euler angles and using quaternions to describe orientation.

1. As we discussed earlier, elements of the transformation matrix are direct cosine of angles between axes of the body-fixed frame and axes of the space-fixed frame. Thus, the transformation matrix describes the orientation of the body-fixed frame relative to the space-fixed frame with nine parameters.

If the time derivative of the transformation matrix (dA/dt) is calculated, a new transformation matrix (new orientation of body) at time $t + \Delta t$ is obtained by numerical integration.

In contrast to the Euler angles method, in which the time derivative of Euler angles is used to obtain new orientation, time derivative of the transformation matrix is used in this method.

This solution remedies the problem of singularity associated with the derivative of Euler angles for $\theta = 0$. The transformation matrix uses nine parameters for describing the orientation of body in 3D space. This means that it requires six constraints (independent equations) to make the system determined.

These constraints come from the orthogonal property of the transformation matrix.

If dA/dt is integrated to obtain the new orientation of the body in each time step, very small numerical errors are introduced in the elements of A. These small numerical errors are accumulated over time and matrix A loses its orthogonality and hence it would be no longer a transformation matrix.

Step-wise procedure for numerical integration is complex and can be implemented rather hard into a regular DEM code. In addition, it involves too many matrix operations, which degrades the computational efficiency.

Rotational Motion:

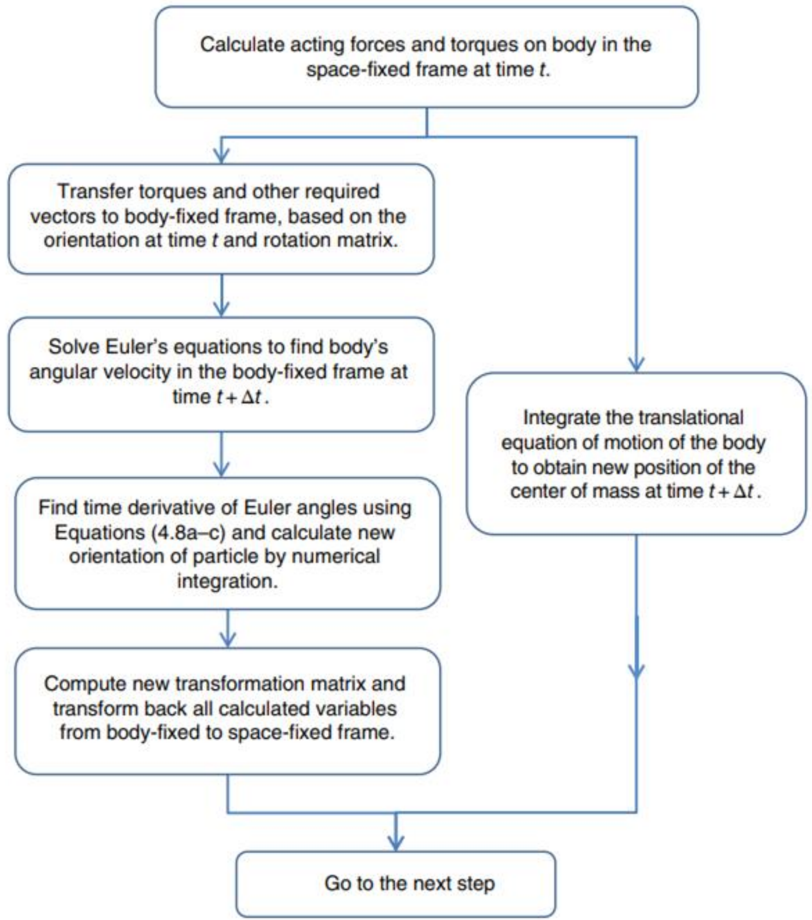


Figure 4.4 Steps to integrate/solve equations of motion of a non-spherical rigid particle in 3D space. This flow chart corresponds to steps 5–7 in Figure 3.1

Rotational Motion: Method of Quaternions

Quaternion is a vector in 4D space that extends a complex number into a higher dimension. A quaternion is defined using a scalar value s and a 3D vector **v** as follows:

$$q=(s,\vec{v}) \quad \text{or} \quad q=s\tilde{e}_0+v_1\tilde{e}_1+v_2\tilde{e}_2+v_3\tilde{e}_3$$

where $(\tilde{e}_0,\tilde{e}_1,\tilde{e}_2,\tilde{e}_3)$ are elements of the quaternion space that should hold the following relation:

$$\tilde{e}_0^2+\tilde{e}_1^2+\tilde{e}_2^2+\tilde{e}_3^2=1 \tag{4.20}$$

The product of two quaternions is defined as follows:

$$q_1.q_2=(s_1,\vec{v}_1).(s_2,\vec{v}_2)=(s_1s_2-\vec{v}_1.\vec{v}_2,\ s_1\vec{v}_2+s_2\vec{v}_1+\vec{v}_1\times\vec{v}_2)$$

Quaternion product is not commutative.

We can also define the quaternion in terms of a rotation angle α around vector u:

$$q_\alpha=\left(\cos\frac{\alpha}{2},\vec{u}\sin\frac{\alpha}{2}\right)$$

The following expression is used to rotate and arbitrary vector $\vec{r}$ by angle α around vector $\vec{u}$

$$\vec{r}'=q_\alpha.(0,\vec{r}).q_\alpha^* \quad (4$$

where q_α^* is the conjugate of q_α and is defined as:

$$q_\alpha^*=\left(\cos\frac{\alpha}{2},-\vec{u}\sin\frac{\alpha}{2}\right)$$

We can also define the components of the unit quaternion in terms of Euler angles:

$$\tilde{e}_0=\cos\frac{\theta}{2}\cos\frac{\phi+\psi}{2}$$

$$\tilde{e}_1=\sin\frac{\theta}{2}\cos\frac{\phi-\psi}{2}$$

$$\tilde{e}_2=\sin\frac{\theta}{2}\sin\frac{\phi-\psi}{2}$$

$$\tilde{e}_3=\cos\frac{\theta}{2}\sin\frac{\phi+\psi}{2}$$

Rotational Motion: Method of Quaternions

In this way, the 3D orientation of the body-fixed frame is mapped into the components of the unit quaternion.

Therefore, quaternions are capable of representing the orientation of a body in 3D space by four parameters (contrary to the transformation matrix with nine parameters).

The rotation matrix A can be expressed in terms of components of the unit quaternion:

$$A = \begin{bmatrix} \tilde{e}_0^2 + \tilde{e}_1^2 - \tilde{e}_2^2 - \tilde{e}_3^2 & 2(\tilde{e}_1\tilde{e}_2 + \tilde{e}_0\tilde{e}_3) & 2(\tilde{e}_1\tilde{e}_3 - \tilde{e}_0\tilde{e}_2) \\ 2(\tilde{e}_1\tilde{e}_2 - \tilde{e}_0\tilde{e}_3) & \tilde{e}_0^2 - \tilde{e}_1^2 + \tilde{e}_2^2 - \tilde{e}_3^2 & 2(\tilde{e}_2\tilde{e}_3 + \tilde{e}_0\tilde{e}_1) \\ 2(\tilde{e}_1\tilde{e}_3 + \tilde{e}_0\tilde{e}_2) & 2(\tilde{e}_2\tilde{e}_3 - \tilde{e}_0\tilde{e}_1) & \tilde{e}_0^2 - \tilde{e}_1^2 - \tilde{e}_2^2 + \tilde{e}_3^2 \end{bmatrix}$$

To transform an arbitrary vector, r, between the space-fixed frame and the body-fixed frame, one can use the transformation matrix or use the quaternion and its conjugate:

$$\vec{r}^{\,b} = A\vec{r}^{\,s}$$

$$\left(0, \vec{r}^{\,b}\right) = q^*.\left(0, \vec{r}^{\,s}\right).q$$

and from the body-fixed frame to the space-fixed frame:

$$\vec{r}^{\,s} = A^T\vec{r}^{\,b}$$

$$\left(0, \vec{r}^{\,s}\right) = q.\left(0, \vec{r}^{\,b}\right).q^*$$

In addition to the transformation relations, we also need a time derivative of the quaternion to update the instantaneous orientation of the body-fixed frame. The time derivative of quaternion is given by:

$$\dot{q} = \frac{1}{2} q.\left(0, \vec{\omega}^{\,b}\right) = \frac{1}{2}\begin{bmatrix} -\tilde{e}_1\omega_x^b - \tilde{e}_2\omega_z^b - \tilde{e}_3\omega_z^b \\ \tilde{e}_0\omega_x^b + \tilde{e}_2\omega_z^b - \tilde{e}_3\omega_y^b \\ \tilde{e}_0\omega_y^b - \tilde{e}_1\omega_z^b + \tilde{e}_3\omega_x^b \\ \tilde{e}_0\omega_z^b + \tilde{e}_1\omega_y^b - \tilde{e}_2\omega_x^b \end{bmatrix}$$

Superellipsoids:

Using analytical equation of superellipsoids is one of the methods for representing particle shape in the DEM model. In this section, we present the general form of mathematical equation of superellipsoids and their geometric properties. We also discuss:

1. The method of evaluating contact forces.
1. The method of evaluating torques between superellipsoidal particles.
1. Method of Detection of Contact Plane between two colliding particles.

The edge of a particle in 2D space and the surface of a particle in 3D space can be defined by a set of analytical equations. In the DEM simulation of granular flows, superellipse (2D) and superellipsoid (3D) are mostly used to represent the particle analytically. A superellipse can be defined by a closed curve with the following equation:

$$\left(\frac{x^b}{a}\right)^m + \left(\frac{y^b}{b}\right)^m = 1$$

where a and b are positive real numbers that show the size of superellipse on major and minor axes, and m is a positive number that shows the shape of curve and is calculated by:

$$m = \frac{p}{q} > 0, \text{ where } \begin{cases} p \text{ is an even positive integer} \\ q \text{ is an odd positive integer} \end{cases}$$

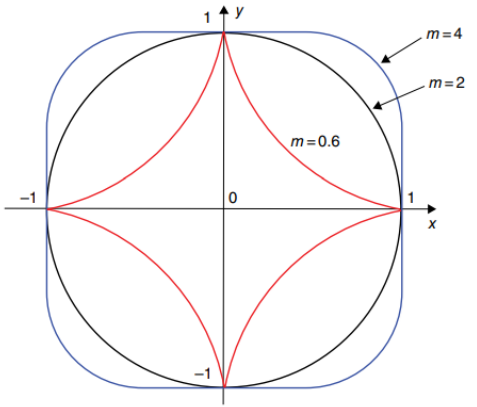


Figure 4.5 A superellipse can change gradually from a star-shape to a circle and to a square in the limit by changing the value of m from near zero to infinity in Equation 4.30 with $a = b = 1$

Superellipsoids:

We define an equation of a superellipsoid in the body-fixed frame that coincides with its principal axes. In this way, all off-diagonal elements of the inertia tensor become zero.

The parametric equation of the surface of a superellipsoid in the body-fixed frame is expressed as:

$$\begin{bmatrix} x^b \\ y^b \\ z^b \end{bmatrix} = \begin{bmatrix} a\cos^{\varepsilon_1}\gamma\ \cos^{\varepsilon_2}\lambda \\ b\cos^{\varepsilon_1}\gamma\ \sin^{\varepsilon_2}\lambda \\ c\sin^{\varepsilon_1}\gamma \end{bmatrix}, \quad \begin{matrix} -\pi/2 \le \gamma \le \pi/2 \\ -\pi \le \lambda < \pi \end{matrix}$$

It should be noted that the terms with exponents ε_1 and ε_2 are singed power functions, meaning that $\sin^{\varepsilon_2}\lambda = \text{sign}(\sin\lambda)|\sin\lambda|^{\varepsilon_2}$. In this equation, parameters a, b, and c are positive real numbers that show the size of body on the coordinate axes and exponents ε_1 and ε_2 are positive real numbers that determine the shape of the superellipsoid cross-section. ε_2 determines the shape of superellipsoid cross section from the top (a plane parallel to xy plane) and ε_1 determines the side-view cross section of a superellipsoid.

There are five parameters for defining the surface of a superellipsoid in the body-fixed frame.

Alternatively, the equation of the superellipsoid can be expressed in implicit form in the

$$F(x^b, y^b, z^b) = \left(\left(\frac{x^b}{a}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y^b}{b}\right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z^b}{c}\right)^{\frac{2}{\varepsilon_1}} - 1$$

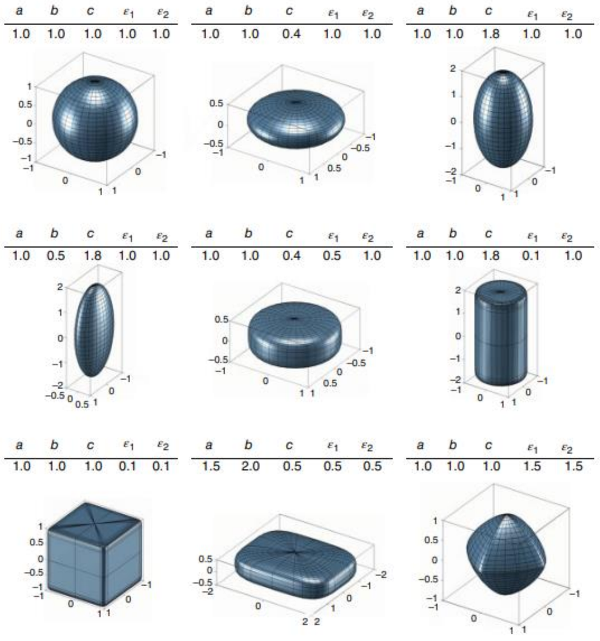


Figure 4.6 Some shapes that can be generated by superellipsoid equation by changing values of size parameters a, b, c, and exponents ε_1 and ε_2

Contact Forces:

1. The contact force at contact point C between two particles is comprised of normal and tangential contact forces. The tangential overlap, normal, and tangential relative velocities are required to calculate contact force and torques.

1. The velocity of particle at the contact point in the space-fixed frame is given by:

$$\vec{v}^s_{c,i} = \vec{v}^s_{CM,i} + \vec{\omega}^s_i \times \vec{R}^s_{C,ij}$$

$$\vec{v}^s_{c,i} = \vec{v}^s_{CM,i} + A^T\left(\vec{\omega}^b_i \times \vec{R}^b_{C,ij}\right)$$

1. The relative velocity of contacting particles is then computed from:

$$\vec{v}^s_{ij} = \vec{v}^s_{c,i} - \vec{v}^s_{c,j}$$

Normal and tangential components of relative velocity at the contact point can be calculated.

Having the tangential velocity at the contact point, the tangential overlap can be evaluated. Thereafter, the contact force between two colliding particles can be calculated.

The net contact force acting on the particle in the space-fixed frame is the sum of the contact forces acting on the particle from all surrounding colliding particles:

$$\vec{f}^s_{cont,i} = \sum_{j \in CL_i} \vec{f}^s_{cont,ij}$$

Linear Force Model is used for Macrosopic Observations while Non Force Linear Model is used for Microscopic Observations.

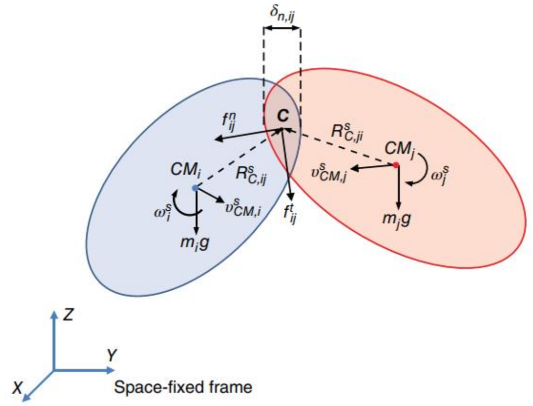


Figure 4.7 Schematic illustration of colliding superellipsoid *i*, *j*, and the forces acting on them in the space-fixed frame

Effective Radius and Radius of Curvatures:

1. Calculating Effective Radius is necessary for computing the Contact Forces.
2. The following equation can be used to calculate the effective radius of two non-spherical particles:

$$R^* = \frac{1}{2\sqrt{A'B'}}$$

where A′ and B′ can be obtained by solving the following two equations simultaneously:

$$2(A'+B') = \frac{1}{R_i'} + \frac{1}{R_i''} + \frac{1}{R_j'} + \frac{1}{R_j''}$$

$$4(A'-B')^2 = \left(\frac{1}{R_i'} - \frac{1}{R_i''}\right)^2 + \left(\frac{1}{R_j'} - \frac{1}{R_j''}\right)^2 + 2\left(\frac{1}{R_i'} - \frac{1}{R_i''}\right)\left(\frac{1}{R_j'} - \frac{1}{R_j''}\right)\cos(2\beta)$$

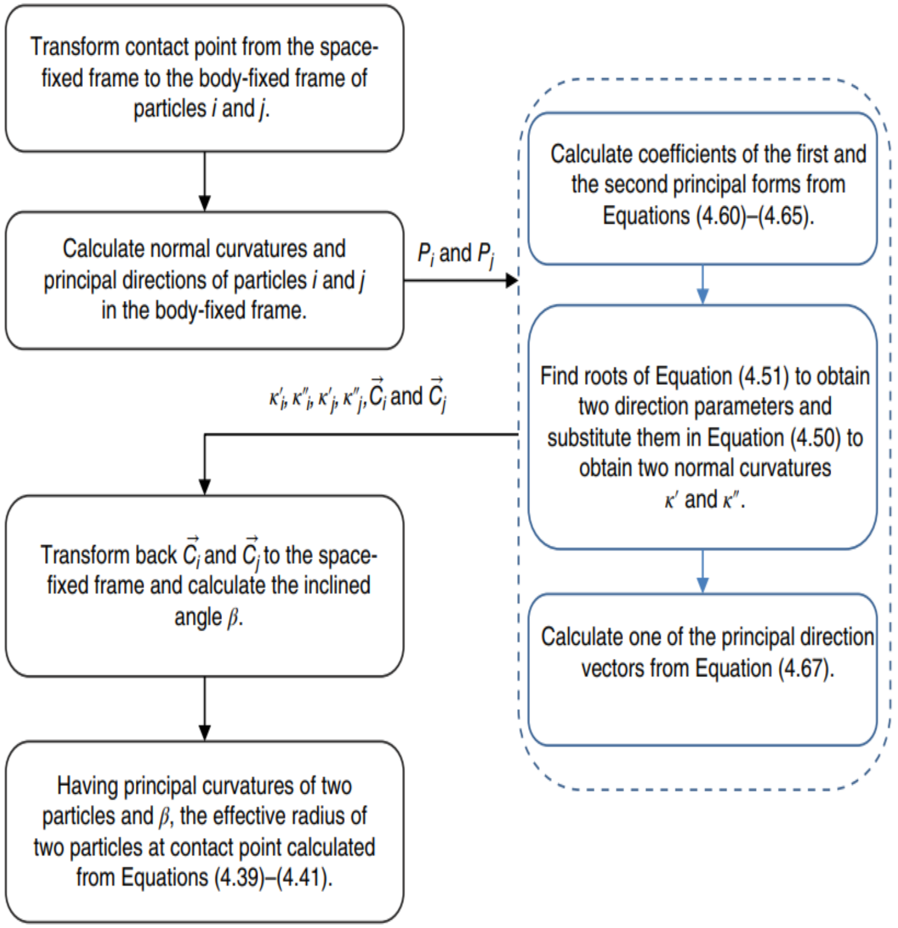


Figure 4.8 The procedure of calculating the curvatures of two ellipsoidal particles and the inclined angle at their contact point

Torque Calculations:

Contact of non-spherical particles with each other and with wall can cause torque on them. Contact forces are calculated in the space-fixed frame.

The resulting contact torque in the space-fixed frame is calculated from:

$$\vec{M}^{s}_{cont,i} = \sum_{j \in CL_i} \left(\vec{R}^{s}_{C,ij} \times \vec{f}^{s}_{cont,ij} \right)$$

It is also possible to include rolling torques as follows:

$$\vec{M}^{r}_{ij} = -\mu_r R_{eff} \left| \vec{f}^{n}_{ij} \right| \hat{\omega}_{ij}$$

$$\vec{M}^{s}_{i} = \vec{M}^{s}_{cont,i} + \vec{M}^{s}_{r,i}$$

To solve the Euler's equations, the total torque on the particle should be transformed to its body-fixed frame using either the rotation matrix or the quaternion products.

It is also needed to calculate mass and principal moments of inertia of superellipsoids. The mass of a superellipsoid with density ρ_i is given by Jaklic *et al.* [35]:

$$m_i = 2\rho_i abc\varepsilon_1\varepsilon_2 B\left(\frac{\varepsilon_1}{2}+1, \varepsilon_1\right) B\left(\frac{\varepsilon_2}{2}, \frac{\varepsilon_2}{2}\right) \tag{4.71}$$

where $B(x, y)$ is the beta function. The principal moments of inertia read as follows:

$$\hat{I}_1 = \frac{1}{2}\rho_i abc\varepsilon_1\varepsilon_2 \left[b^2 B\left(\frac{3}{2}\varepsilon_2, \frac{1}{2}\varepsilon_2\right) B\left(\frac{1}{2}\varepsilon_1, 2\varepsilon_1+1\right) + 4c^2 B\left(\frac{1}{2}\varepsilon_2, \frac{1}{2}\varepsilon_2+1\right) B\left(\frac{3}{2}\varepsilon_1, \varepsilon_1+1\right) \right] \tag{4.72}$$

$$\hat{I}_2 = \frac{1}{2}\rho_i abc\varepsilon_1\varepsilon_2 \left[a^2 B\left(\frac{3}{2}\varepsilon_2, \frac{1}{2}\varepsilon_2\right) B\left(\frac{1}{2}\varepsilon_1, 2\varepsilon_1+1\right) + 4c^2 B\left(\frac{1}{2}\varepsilon_2, \frac{1}{2}\varepsilon_2+1\right) B\left(\frac{3}{2}\varepsilon_1, \varepsilon_1+1\right) \right] \tag{4.73}$$

$$\hat{I}_3 = \frac{1}{2}\rho_i abc\varepsilon_1\varepsilon_2 \left(a^2+b^2\right) B\left(\frac{3}{2}\varepsilon_2, \frac{1}{2}\varepsilon_2\right) B\left(\frac{1}{2}\varepsilon_1, 2\varepsilon_1+1\right) \tag{4.74}$$

Multi Spherical Particles:

1. The multi-sphere method is another method for generating non-spherical particles.

1. In spite of superellipsoids, which are non-spherical axisymmetric particles, the multi-sphere method can generate particles with irregular shapes.

1. Based on this method, spheres with smaller size are glued to each other to approximately fill the volume/surface of the real particle.

1. These spheres are allowed to overlap to any extent. In this way, any number of spheres, with different sizes and overlaps, can be used.

1. After positioning spheres in the non-spherical particle and producing the final shape, the relative position of spheres within the particle will not change.

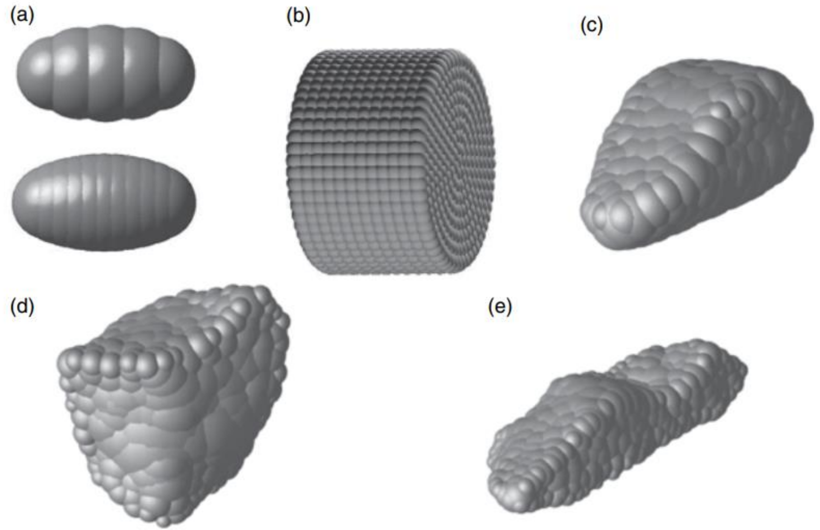
1. Thus, the dynamics of the rigid body can be applied to multi-sphere particles.

The particle shape/surface can be more accurately approximated by increasing the number of spheres and reduce the level of bumpiness of the surface.

On the other hand, the computational cost of simulation also increases with increasing the number of spheres.

Although the accuracy of representation of the surface is improved by using a large number of spheres, the mechanical behavior of the particle is not necessarily improved due to additional dominant errors introduced with increasing the number of spheres in the particle.

We are not still definite about an optimal number of spheres as it changes from one case to another.



Multi Spherical Particles:

$$\vec{r}_{ik}^{\,s} = \vec{r}_{CM,i}^{\,s} + A^T \vec{r}_{ik}^{\,b}$$

$$\vec{v}_{ik}^{\,s} = \vec{v}_{CM,i}^{\,s} + A^T \left(\vec{\omega}_i^{\,b} \times \vec{r}_{ik}^{\,b}\right)$$

The net contact force acting the particle, $\vec{f}^{\,s}_{cont,i}$, is sum of contact forces acting on all spheres in that particle:

$$\vec{f}^{\,s}_{cont,i} = \sum_{jl \in CL_i} \vec{f}^{\,s}_{cont,ik,jl} \tag{4.87}$$

where $\vec{f}^{\,s}_{cont,ik,jl}$ is the contact force between sphere ik in particle i and sphere jl in particle j. Since the contact force acts on the surface and it is considered on the center of mass of the particle, a torque should be considered around this point. The net contact torque on particle in the space-fixed frame is obtained by:

$$\vec{M}^{\,s}_{cont,i} = \sum_{jl \in CL_i} \vec{R}^{\,s}_{C,ik,jl} \times \vec{f}^{\,s}_{cont,ik,jl} \tag{4.88}$$

For solving linear and rotational equations of motion, the total mass of particle and principal moments of inertia should be known. It is common to calculate these values from properties of the spheres inside the particle. The following equations are used to obtain the total mass and principal moments of inertia of a multi-sphere particle:

$$m_i = \sum_{k=1}^{NS} m_{ik} \tag{4.89}$$

$$\hat{I}_i = \begin{pmatrix} \sum I_{ik} + \sum m_{ik}(y_{ik}^2 + z_{ik}^2) & 0 & 0 \\ 0 & \sum I_{ik} + \sum m_{ik}(x_{ik}^2 + z_{ik}^2) & 0 \\ 0 & 0 & \sum I_{ik} + \sum m_{ik}(x_{ik}^2 + y_{ik}^2) \end{pmatrix} \tag{4.90}$$

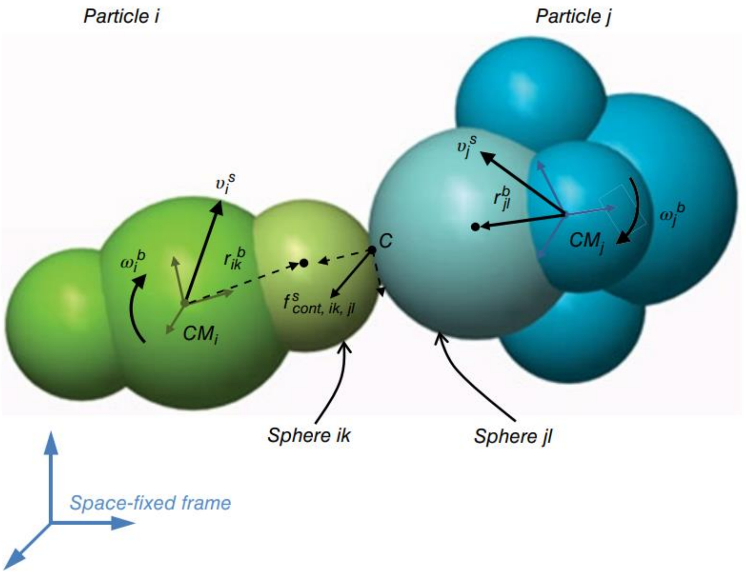


Figure 4.13 Two non-spherical particles i and j are in contact via their constituent spheres ik and jl

Contact Force in Multi-spherical particles:

$$\vec{f}^{\,s}_{cont,ij} = \frac{1}{NCP_{ij}} \sum \vec{f}^{\,s}_{cont,ik,jl}$$

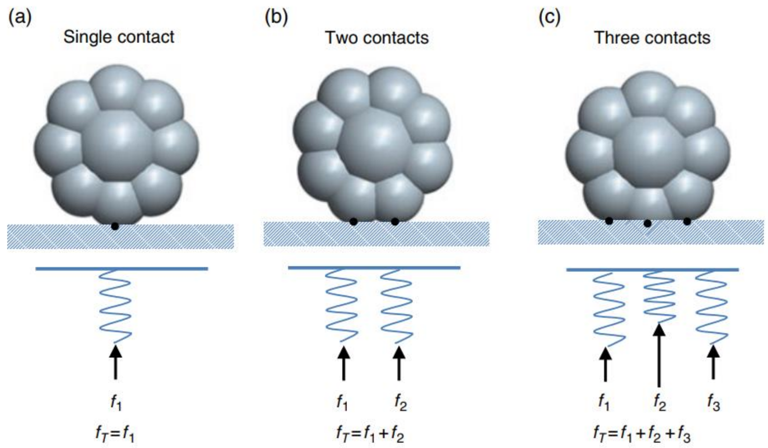


Figure 4.15 The schematic of possible contact conditions between a multi-sphere particle and a flat wall. With more contact points between particle and wall, the equivalent spring would be stiffer

Höhner *et al.* [42] proposed a simple approach for adjusting the total contact force between particle and wall (also applicable to two particles). Their approach is based on two facts: first, the overlaps between spheres and wall in the contact points can be unequal and second, the number of contacts can be changed during the collision. They proposed an incremental approach for calculating contact force at each time step. Having the contact force at the previous time step, $\vec{f}^{\,s,0}_{cont,ij}$, the contact force at the current time step is calculated from:

$$\vec{f}^{\,s}_{cont,ij} = \vec{f}^{\,s,0}_{cont,ij} + \Delta\vec{f}^{\,s}_{cont,ij} \tag{4.92}$$

where $\Delta\vec{f}^{\,s}_{cont,ij}$ is the incremented contact force from the previous time step to the current time step. This method can be used for all linear and non-linear contact force models. As an example, we describe how to calculate the incremental contact force for a linear model. In normal direction, the incremental contact force is obtained from:

$$\Delta\vec{f}^{\,s,n}_{cont,ij} = -\frac{k_{n,ij}}{NCP_{ij}} \sum \left(\delta_{n,ik,jl} - \delta^0_{n,ik,jl}\right)\vec{n}_{ik,jl} - \frac{\eta_{n,ij}}{NCP_{ij}} \sum \left(v_{rn,ik,jl} - v^0_{rn,ik,jl}\right)\vec{n}_{ik,jl} \tag{4.93}$$

where $\delta^0_{n,ik,jl}$ and $\delta_{n,ik,jl}$ are previous and current normal overlaps between sphere *ik* from particle *i* and sphere *jl* from particle *j*, and $v^0_{rn,ik,jl}$ and $v_{rn,ik,jl}$ are previous and current normal relative velocities between spheres *ik* and *jl*. $k_{n,ij}$ and $\eta_{n,ij}$ are normal spring stiffness and normal damping coefficient between particles *i* and *j*, respectively. Note that the summations are done over current contact points between two particles. The incremental contact force for the tangential direction can be obtained using similar methodology. It can be seen that there is no need to adjust the spring stiffness or change the contact force law. Since the incremental approach is used, the number of contacts changes and the overlap of spheres at contact points can be different during a collision. A drawback of this approach is usage of extra memory for storing contact parameters, such as overlap, relative velocity, and so on.