CHAPTER 3

THEORETICAL MODEL:
DISCRETE ELEMENT METHOD

3.1 Introduction

In this chapter we introduce the problem and outline our strategy for solving it. It is clear from the literature review presented in the earlier chapter that the most appropriate method to model granular motion in rotating cylinder is the Discrete Element Method or soft sphere method since it is capable of handling multiple particle contacts, which is particularly important when modelling quasi-static systems like granular flows in rotating cylinder. Hence the soft sphere method is chosen for the present work to study the dynamics of granular material motion in a rotating cylinder. This chapter describes the general principles of Discrete Element Method.

The Discrete Element Method (DEM) is based on the Lagrangian approach. Hence it is possible to simulate the motion of granular material at the microscopic level, which in turn can be used to obtain fundamental informatics that are difficult to obtain experimentally or through macroscopic modelling. In the discrete element method, the particle position, orientation, translational and angular velocity are assumed as independent variables. They are obtained by integrating a system of fully deterministic classical equations based on Newtonian dynamics for each particle. For this purpose, explicit expressions have to be evaluated for all the forces acting on and between the particles.
Some of the forces between the particles have their origin based on the deformation experienced by particles when they are in contact with their neighbouring particles. An overlap area of the particles pressed against each other generally provides a good approximation as shown in Figure 3.1.

This assumption is valid since the deformation is much smaller than the particle size. Thus, the contact forces between them depend on the overlap geometry, the properties of the material and the relative velocity between the particles in the contact area. Hence in the perfect contact model, it is required to describe the effects of elasticity, energy loss through internal friction and surface friction and attraction on the contact surface for describing the contact force calculations, (Kohring [1995]).
3.2 Equations of particle motion

The equations of motion for each particle are derived from Newton's law of classical Newtonian dynamics. These include a system of equations for the translational motion of center of gravity and rotational motion around the center of gravity for each particle in the granular medium.

3.2.1 Translational motion

Translational motion of the center of gravity of a particle $i$ can be fully described by a system of equations (Landau and Lifshitz, [1960])

$$m_i \frac{d^2 x_i}{dt^2} = m_i a_i = F_i$$

(3.1)

$$v_i = \frac{dx_i}{dt}$$

(3.2)

where $v_i$, $a_i$ and $x_i$ are vectors of velocity, acceleration and the position of the center of gravity $m_i$ of the particle $i(i = 1,...N)$ respectively. $N$ is the total number of particles in the granular material. Since the motion of particles is considered to be that of a rigid body, the sum of all forces $F_i$ is assumed to act on the center of gravity of the particle:

$$F_i = F_{i,contact} + F_{i,gravity} + F_{i,external}$$

(3.3)
Theoretical Model: Discrete Element Method

where $F_{i,\text{contact}}$ is the summation of direct contact forces between the particle $i$ and all other particles that are in contact with particle $i$.

$$F_{i,\text{contact}} = \sum_{j=1, j \neq i}^{n_{ij}} F_{ij} \quad (3.4)$$

where $F_{ij}$ is a force acting on the contact area of elastic impacts between the particles $i$ and $j$, $n_{ij}$ denotes the number of particles that are in contact with $i$. It should be noted here that Newton’s third law of motion applies

$$F_{ij} = -F_{ji} \quad (3.5)$$

In general, contacting forces can also include inter-particle forces acting between charged particles. $F_{i,\text{gravity}}$ is the gravitational force acting on the particle:

$$F_{i,\text{gravity}} = m_i g = V_i \rho_i g \quad (3.6)$$

where, $V_i$ is the particle volume, $\rho_i$ is the particle density and $g$ denotes the gravity acceleration vector.

$F_{i,\text{external}}$ corresponds to the total sum of all the external forces acting on the particle such as the fluid drag forces and fluid lift forces, which are prominent in solid-gas phase systems. If a particle is charged, then $F_{i,\text{external}}$ includes the external magnetic field also.
3.2.2 Rotational Motion

Rotational motion of the particle \(i\) around the center of gravity can be fully described by the following systems of equations (Landau and Lifshitz, [1960])

\[
I_i \frac{d^2 \theta_i}{dt^2} = T_i
\]  

(3.7)

\[
\omega_i = \frac{d\theta_i}{dt}
\]  

(3.8)

where the \(\theta_i\) and \(\omega_i\) are the vectors of orientation and angular velocity. \(I_i\) is the inertial tensor of the particles and is expressed as \(I_i = (I_{1i}, I_{2i}, I_{3i})\). The sum of all torques \(T_i\) acting on the particle \(i\) is given by

\[
T_i = T_{i, contact} + T_{i, fluid} + T_{i, external}
\]  

(3.9)

where \(T_{i, fluid}\) denotes the total torque caused by anti-symmetric fluid drag forces, \(T_{i, external}\) denotes the summation of torques caused by other external forces and \(T_{i, contact}\) is the summation of all the torques caused by the contact forces between the particles and is expressed as

\[
T_{i, contact} = \sum_{j=1, j\neq i}^{N} T_{ij} = \sum_{j=1, j\neq i}^{N} d_{cij} \times F_{ij}
\]  

(3.10)

where \(d_{cij}\) is the vector of relative contact positions.
Orientation $\theta_i$, angular velocity $\omega_i$, torque $T_i$ and inertia momentum $I_i$ of two-dimensional particles can be treated as scalars. Therefore, the equations of rotational motion of the particle $i$ around the center of gravity can be simplified as follows:

$$I_i \frac{d^2 \theta_i}{dt^2} = I_i \omega_i = T_i$$

$$\omega_i = \frac{d \theta_i}{dt}$$

For three-dimensional particles, however, the inertial moment must be calculated at every time step, according to the new orientation of the particle in the space. Therefore, it is convenient to use the space-fixed and body-fixed co-ordinate systems. The space-fixed (or laboratory) co-ordinate system is fixed in a laboratory space. The body-fixed (or local) co-ordinate system is a moving Cartesian co-ordinate system, which is fixed with the particle and whose axes are superposed by the principal axes of inertia (Algis Dzingys & Bernhard Peters [2001]).

The inertial tensor always is diagonal $I_i = (I_{li}, I_{2i}, I_{3i})$ in a body-fixed co-ordinate system. For spherical particles, $I_{li} = I_{2i} = I_{3i} = I_i$, and body-fixed co-ordinates can be set in the same direction as space-fixed ones. Therefore, orientation is not used for spheres and the equations of motion can be written as

$$\frac{d \omega_i}{dt} = \frac{T_i}{I_i}.$$
3.3 Boundary conditions

The properties of granular flow are strongly dependent on the boundary conditions at the wall (Thompson and Grest, [1991]). Therefore the boundary conditions are very important for an adequate simulation of the granular material behaviour. Several types of boundary conditions can be employed:

- (a) Walls that may be moving or stationary
- (b) Inflow and outflow
- (c) Periodic

Walls can be constructed using planes, spheres, cylinders or any other shape as big particles or by an array of small particles. In general, boundaries of the system such as walls are required for the motion of granular material or particles within enclosures, where the wall may have an important influence on the motion of a granular material due to wall-particle interaction. Furthermore, walls can move and rotate around a point of rotation. The rotation of wall particles, in particular is unavoidable in the present study, in which the motion of granular material on the rotating cylinder solely depends on the moving wall.

A rotating cylinder can be constructed by a cylinder or sphere with a negative radius. Collisions between particles and walls are defined by the material and geometry of the particles and walls, as in the case of collisions between particles. It is convenient to construct rough walls by an array of particles (Thompson and Grest, [1991]).

For the present formulation the following methodology is adapted; the system under consideration has been modelled by an ensemble of spheres possessing the
same material constants as that of the grains inside the container. The motion of the wall spheres is not affected by the impacts but is strictly governed by the continuous rotation of the cylinder. The calculation of contact forces between the particles and wall are defined in the same way as between particles.

### 3.4 Particle shape

In three dimensional co-ordinate systems, particles can be represented by spheres, ellipsoids, super quadrics, polyhedrons, etc. and for two dimensional co-ordinate system by disks, ellipses, polygons, polar forms etc. and by strings in one dimensional co-ordinate system. Reviews of various possible shapes of particles and some aspects of applications of shapes are presented in Haff [1993] and Ristow [1996]. It is not a difficult problem to construct particles of various shapes. The major problem however is to detect the contact between the neighbouring particles and to calculate the overlap area, intersection and contact points and normal and tangential contact vectors. For some of the analytical shapes, such as ellipsoids or super quadrics, analytical solutions may be found, However, for complicated shapes, considerable computational effort is needed.

The choice of sphere offers considerable simplification since the center of gravity of a sphere coincides with the geometrical center and the particle can be described only by its radius with no need of specifying the orientation (Lubachevsky [1991], Lubachevsky et al. [1996], Kornilovsky et al. [1996], Sadd et al. [1993], Hoomans et al. [1996], Luding et al. [1996], Kumaran [1997]).

This especially applies to molecular dynamics simulations, because it is quiet natural to describe an atom as a sphere (Grest et al. [1989], Gilkman et al. [1996]). The particle is described only with the radius and no orientation is needed. Campbell
and Brennen [1985] used a disk to simulate three dimensional cylinders oriented in the same direction and located in the same plane. Hence for the present study, the particle shape is assumed to be spherical.

3.5 General Scheme for the contact geometry

Let any two particles $i$ and $j$ be in contact with position vectors $x_i$ and $x_j$ with center of gravity lying at $O_i$ and $O_j$ having linear velocities $v_i$ and $v_j$, angular velocities $\omega_i$ and $\omega_j$ respectively as shown in Figure 3.2. (Algis Dzingys and Bernhard Peters [2001]).

Figure 3.2: Contact between two particles $i$ and $j$. 
The contact point $C_{ij}$ is defined to be at the center of the overlap area with the position vector $x_{cij}$. The vector $x_{ij}$ of the relative position point from the center of gravity of particle $i$ to that of particle $j$ is defined as $x_{ij} = x_i - x_j$.

The depth of overlap is $h_{ij}$. Unit vector in the normal direction of the contact surface through the center of the overlap area is denoted by $n_{ij}$. It extends from the contact points to the inside of the particle $i$ as $n_{ij} = -n_{ij}$.

The vectors $d_{cij}$ and $d_{cji}$ are directed towards the contact point from the centers of particle $i$ and particle $j$ respectively and are represented as

$$d_{cij} = x_{cij} - x_i$$
and

$$d_{cji} = x_{cji} - x_j \tag{3.12}$$

Since the particle shape is assumed to be spherical, for spheres of any dimension the contact parameters can be written as follows:

$$h_{ij} = \begin{cases} R_i + R_j - |x_{ij}|, & |x_{ij}| < R_i + R_j \\ 0, & |x_{ij}| \geq R_i + R_j \end{cases} \tag{3.13}$$

$$n_{ij} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad x_i \neq 0 \tag{3.14}$$
Theoretical Model: Discrete Element Method

\[ d_{eq} = \left( R_i - \frac{h}{2} \right) n_i \]  

(3.15)

where \( R_i \) is the radius of the particle.

The relative velocity of the contact point is defined as

\[ v_{ij} = v_{cij} - v_{cji} \]  

(3.16)

where,

\[ v_{cij} = v_i + \omega_i \times d_{cij} \]

\[ v_{cji} = v_j + \omega_j \times d_{cji} \]

are the velocities of particle \( i \) and particle \( j \) respectively.

The normal and tangential components of the relative velocities are defined by

\[ v_{n,ji} = (v_{ij} \cdot n_{ij}) n_{ij} \]  

(3.17)

and

\[ v_{t,ji} = v_{ij} - v_{n,ij} \]  

(3.18)

In case of contact with partial slip, particles may slip relative to the distance \( \delta_{t,ij} \) in tangential direction. \( \delta_{t,ij} \) is the integrated slip in tangential direction after particles \( i \) and \( j \) came into contact and can be defined by the equation, (Algis Dzingys and Bernhard Peters [2001]):

\[ \delta_{t,ij} = \left| \int v_{t,ij}(t) \, dt \right| \]  

(3.19)
Here $\delta_{t,ij}$ is allowed to increase until the tangential force exceeds the limit imposed by static friction. The vector of tangential displacement $\delta_{t,ij}$ is defined to be perpendicular to the normal contact direction and located on the same line as $v_{t,ij}$. If the tangential component of the contact velocity $v_{t,ij}$ is not equal to zero, then the unit vector $t_{ij}$ of the tangential contact direction is directed along $v_{t,ij}$. If $v_{t,ij}$ is equal to zero, $t_{ij}$ has the same direction as that of the slip. Otherwise $t_{ij}$ is equal to zero, if $v_{t,ij}$ and $\delta_{t,ij}$ are equal to zero, then

$$
t_{ij} = \begin{cases} 
\frac{v_{t,ij}}{|v_{t,ij}|}, & v_{t,ij} \neq 0 \\
\frac{\delta_{t,ij}}{|\delta_{t,ij}|}, & v_{t,ij} = 0, \delta_{t,ij} \neq 0 \\
0, & \text{otherwise}
\end{cases} \quad (3.20)
$$

### 3.6 Inter particle contact forces

The contact force $F_{ij}$ of a viscoelastic collision between two particles $i$ and $j$ acts on the contact surface and is convenient to calculate $F_{ij}$ acting on an imaginary contact point $C_{ij}$. According to Kohring [1995] a model of inter-particle viscoelastic contact forces has to describe the following four effects:
Theoretical Model: Discrete Element Method

- Particle elasticity
- Energy loss through internal friction
- Attraction on the contact surface
- Energy loss due to surface friction

\[ F_{ij} = F_{n,ij} + F_{t,ij} \]  

(3.21)

which is in their general form, would be a function of the relative normal \( (n_{ij}) \) and tangential \( (\delta_{ij}) \) displacements of contact as well as the relative normal and tangential velocities, Sadd et al. [1993],

\[ F_{n,ij} = F_{n,ij}(h_{ij}, v_{n,ij}, \delta_{t,ij}, v_{t,ij}) \]  

(3.22)

\[ F_{t,ij} = F_{t,ij}(h_{ij}, v_{n,ij}, \delta_{t,ij}, v_{t,ij}) \]  

(3.23)

Algis Dziugys and Bernhard Peters [2001] have given a detailed survey of the various types of contact forces used for discrete element simulation along with their merits and demerits. For the present theoretical formulation the contact forces between the spherical particles are modelled as springs, dash-pots and a friction slider as originally proposed by Cundall and Strack [1979]. The schematic representation of contact forces adopted for the theoretical formulation using spring, dash-pot and slider is shown in Figure 3.3 for particle-particle contact and particle-wall contact. The spring accounts for elastic repulsion, dash-pots express the damping effect, and friction sliders express the tangential friction force in the
presence of a normal force. The effect of these mechanical elements on particle motion appears through the stiffness $k$, the damping coefficient $\eta$ and the friction coefficient $\mu$.

Figure 3.3: Schematic representation of contact forces
The normal components of contact forces between particles can be expressed as the sum of elastic repulsion, internal friction and the surface attraction forces.

\[
F_{n,ij} = F_{n,ij,\text{elastic}} + F_{n,ij,\text{viscous}}
\]  

(3.24)

**Normal Elastic repulsion force,** \(F_{n,ij,\text{elastic}}\): This force is based on the linear Hooke’s law of a spring with a spring stiffness constant \(k_{n,ij}\) and is given by the expression,

\[
F_{n,ij,\text{elastic}} = k_{n,ij}h_{ij}n_{ij}
\]  

(3.25)

where \(h_{ij}\) is the depth of overlap between the contacting particles, \(n_{ij}\) is the normal component of the displacement between the particles \(i\) and \(j\). The maximum overlap is dependent on the stiffness coefficient. Typically average overlaps of 0.1-1.0% are desirable requiring stiffness of the order of \(10^5 - 10^7\) N/m (Cleary, 2000)

**Normal energy dissipation force,** \(F_{n,ij,\text{viscous}}\): Energy is dissipated during real collisions between particles and, in general, it depends on the history of impact. A very simple and popular model is based on the linear dependency of force on the relative velocity of the particles at the contact point with a constant normal dissipation coefficient \(\gamma_n\) and is expressed as

\[
F_{n,ij,\text{viscous}} = -\gamma_nm_{ij}v_{n,ij}
\]  

(3.26)

where \(m_{ij}\) is the effective mass of the contacting particles \(i\) and \(j\) and is given by

\[
m_{ij} = \frac{m_im_j}{m_i + m_j}.
\]
The tangential component force model is more complicated due to the consideration of static and dynamic frictional forces. Moreover these forces depend on the normal force and normal displacement. Further the model for static friction must include energy dissipation, because perpetual oscillations in tangential direction will be obtained during the time of static friction. In the literature two major approaches can be found to represent tangential contact forces namely, global and complex models. Global models describe all the phenomena of the tangential force through a single expression. Complex models describe static and dynamic friction by separate equations and the Coulomb criteria. Of course, the continuous particle interaction models require special models for tangential forces. The present theoretical formulation is based on the complex model approach where the evolution of tangential force $F_{t,ij}$ being divided into parts of static friction or dynamic friction. When the tangential force $F_{t,ij}$ is larger than the Coulomb-type cut-off limit, dynamic friction predominates. When $F_{t,ij}$ is lower than the limit, the model of static friction force $F_{t,ij,static}$ must be implemented. Such an approach can be modelled by

$$F_{t,ij} = \begin{cases} F_{t,ij,static} & \text{for } |F_{t,ij,static}| < |F_{t,ij,dynamic}| \\ F_{t,ij,dynamic} & \text{for } |F_{t,ij,static}| \geq |F_{t,ij,dynamic}| \end{cases}$$

or, in a more convenient form for programming purposes, by

$$F_{t,ij} = -t_{ij} \min \left( F_{t,ij,static}, |F_{t,ij,dynamic}| \right)$$

where $t_{ij}$ is the unit vector of the tangential direction of the contact point.
Theoretical Model: Discrete Element Method

The dynamic frictional force can be described by the following equation,

\[ F_{i,j,\text{dynamic}} = -\mu \lvert F_{n,ij} \rvert t_{ij} \]  

(3.29)

where \( \mu \) is the dynamic friction coefficient.

The static friction force is composed of contributions from both the tangential spring and energy dissipation terms and they are expressed as

\[ F_{t,ij,\text{static}} = F_{t,ij,\text{spring}} + F_{t,ij,\text{dissipation}} \]  

(3.30)

where the tangential spring force is defined as

\[ F_{t,ij,\text{spring}} = -k_{t,ij} \delta_{t,ij} t_{ij} \]  

(3.31)

Here, \( k_{t,ij} \) is the spring stiffness coefficient and \( \delta_{t,ij} \) is the integrated slip in tangential direction after the particles \( i \) and \( j \) come into contact.

Friction model for energy dissipation in the tangential direction can be used in the energy dissipation in normal direction,

\[ F_{t,ij,\text{dissipation}} = -\gamma_t m_y \nu_{t,ij} \]  

(3.32)

where \( \gamma_t \) is the shear dissipation coefficient and \( m_y \) the effective mass of the contacting particles.
Based on the above description of the general formulation of the discrete element method, the governing equations for the motion of granular material inside a rotating cylinder can be summarized as follows:

\[ m_i \frac{d^2 x_i}{dt^2} = m_i a_i = F_i \]  (3.33)

\[ v_i = \frac{dx_i}{dt} \]  (3.34)

\[ I_i \frac{d^2 \theta_i}{dt^2} = T_i \]  (3.35)

\[ \omega_i = \frac{d\theta_i}{dt} \]  (3.36)

Here

\[ F_i = m_i g + F_{i,\text{contact}} \]

\[ = m_i g + \sum_{j=1, j \neq i}^{N} F_{ij} \]

\[ = m_i g + \sum_{j=1}^{N} (F_{n,ij} + F_{t,ij}) \]

\[ = m_i g + \sum_{j=1}^{N} F_{n,ij} + \sum_{j=1}^{N} F_{t,ij} \]  (3.37)

and

\[ T_i = T_{i,\text{contact}} \]

\[ = \sum_{j=1, j \neq i}^{N} T_{ij} \]
Theoretical Model: Discrete Element Method

\[ \mathbf{u} = \sum_{j=1, j\neq i}^{N} \mathbf{d}_{ij} \times \mathbf{F}_{ij} \]  
\[ (3.38) \]

where \( \mathbf{d}_{ij} \) is the vector pointing from the center of gravity of particle \( i \) to the contact point with particle \( j \).

In the next section the procedure for solving these equations is presented.

3.7 Time integration:

Various time integration schemes can be used to solve the equations. The main requirements for a good scheme are given below:

- It should be stable
- It should satisfy the required accuracy
- It preferably should satisfy energy and momentum conservation
- It should not require excessive memory
- Time consuming calculation of inter-particle forces should be carried to the minimum possible extent—ideally once per time step, \( \Delta t \)

Some of the most popular schemes used in DEM by various authors include; first order Euler’s scheme, Fourth-order Runge Kutta method (Ovensen et al., [1996], Allen and Tildseley, [1987], Shida et al., [1997]), velocity verlet scheme (Aoki and Akiyama [1995], Kopf et al., [1997], Satoh [1995a, 1995b]), second order Adams-Bashforth scheme (Sundaram and Collins [1996]) and predictor-corrector schemes (Newmark and Asce, [1959]), Thompson and Grest [1991], Form et al. [1993], Lee and Hermann [1993]).
Van Gunsteren and Berendsen [1977] compared the Gear predictor-corrector, Runge-Kutta and verlet schemes for macromolecular simulations and concluded that the Gear scheme is the best for small time steps and verlet algorithm for larger time steps. Hence the 5th order Gear predictor-corrector scheme (Allen and Tildseley, [1987]) is used in this work to solve the equations, which is stable for second-order differential equations with global truncation error of $O(\Delta t^{q+1-2}) = O(\Delta t^{q-1})$.

3.7.1 Gear’s Predictor-Corrector Algorithm

Generally predictor-corrector methods are composed of three steps, namely,

(i) Prediction
(ii) Evaluation
(iii) Correction

Using the current particle position $x(t)$ and particle velocity $v(t)$, the new particle position and new particle velocity is updated using the following steps:

a) Predict the particle position $x(t + \Delta t)$ and particle velocity $v(t + \Delta t)$ at the end of each iteration
b) Evaluate the forces at $t + \Delta t$ using the predicted position
c) Correct the predicted values using some combination of the predicted and previous values of the particle position and particle velocity.

In fifth order Gear predictor-corrector algorithm the particle positions $x$, at time $t + \Delta t$ was predicted using a fifth-order Taylor series based on particle positions
and their derivatives at time $t$. The derivatives $\dot{x}_i(t), \ddot{x}_i(t), \dddot{x}_i(t), x_i^{(v)}(t)$ and $x_i^{(v)}(t)$ are also predicted at each time $t + \Delta t$ by applying Taylor expansions at $t$:

$$x_i(t + \Delta t) = x_i(t) + \dot{x}_i(t)\Delta t + \ddot{x}_i(t)\frac{(\Delta t)^2}{2!} + \dddot{x}_i(t)\frac{(\Delta t)^3}{3!} + x_i^{(v)}(t)\frac{(\Delta t)^4}{4!} + x_i^{(v)}(t)\frac{(\Delta t)^5}{5!}$$

$$\dot{x}_i(t + \Delta t) = \dot{x}_i(t) + \ddot{x}_i(t)\Delta t + \dddot{x}_i(t)\frac{(\Delta t)^2}{2!} + x_i^{(v)}(t)\frac{(\Delta t)^3}{3!} + x_i^{(v)}(t)\frac{(\Delta t)^4}{4!}$$

$$\ddot{x}_i(t + \Delta t) = \ddot{x}_i(t) + \dddot{x}_i(t)\Delta t + x_i^{(v)}(t)\frac{(\Delta t)^2}{2!} + x_i^{(v)}(t)\frac{(\Delta t)^3}{3!}$$

$$x_i^{(v)}(t + \Delta t) = x_i^{(v)}(t) + x_i^{(v)}(t)\Delta t$$

Then the inter particle forces acting on each particle at time $t + \Delta t$ is evaluated using the predicted particle positions. Applying the obtained evaluated forces at time $t + \Delta t$ and Newton’s second law, the particle accelerations $\dddot{x}_i(t + \Delta t)$ can be determined. The difference between the predicted accelerations and evaluated accelerations is then computed:

$$\Delta \dddot{x}_i = \dddot{x}_i(t + \Delta t) - \dddot{x}_i^P(t + \Delta t)$$
The predicted particle positions and their derivatives are corrected using the difference $\Delta \ddot{x}$, obtained between the predicted accelerations and that given by the evaluated force.

In the Gear's Predictor Corrector algorithm, this difference term is used to correct all the predicted particle positions and their derivatives. The correction terms are given by:

$$x_i = x_i^P + \alpha_2 \Delta R_2$$

$$\dot{x}_i \Delta t = \dot{x}_i^P \Delta t + \alpha_1 \Delta R_2$$

$$\frac{x_i (\Delta t)^2}{2!} = \frac{\dot{x}_i^P (\Delta t)^2}{2!} + \alpha_2 \Delta R_2$$

$$\frac{x_i (\Delta t)^3}{3!} = \frac{\dddot{x}_i^P (\Delta t)^3}{3!} + \alpha_3 \Delta R_2$$

$$\frac{x_i^{(iv)} (\Delta t)^4}{4!} = \frac{x_i^{(iv)} (\Delta t)^4}{4!} + \alpha_4 \Delta R_2$$

$$\frac{x_i^{(v)} (\Delta t)^5}{5!} = \frac{x_i^{(v)} (\Delta t)^5}{5!} + \alpha_5 \Delta R_2$$

where,
The parameter $\alpha_i$ promotes numerical stability of the algorithm. The solution of equations (3.33) to (3.36) was carried out by a 5th order Gear Predictor-Corrector scheme (Allen and Tildseley, [1987]). The time step $\Delta t$ of the integration was chosen such that the entire contact between the particles was resolved within 10 time steps at least.

Theoretical Model: Discrete Element Method

$$\Delta R_2 = \frac{\Delta \dot{x}_i (\Delta t)^2}{2!}$$

Values of parameters $\alpha_i$ (Allen and Tildseley, [1987]) for second order differential equations of predicting order $q$ are presented in table 3.1.

**Table 3.1:** Values of the parameter $\alpha_i$

<table>
<thead>
<tr>
<th>$\alpha_i$</th>
<th>$q = 3$</th>
<th>$q = 4$</th>
<th>$q = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_0$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{19}{120}$</td>
<td>$\frac{3}{16}$</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>$\frac{5}{6}$</td>
<td>$\frac{3}{4}$</td>
<td>$\frac{251}{360}$</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{11}{18}$</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>-</td>
<td>$\frac{1}{12}$</td>
<td>$\frac{1}{6}$</td>
</tr>
<tr>
<td>$\alpha_4$</td>
<td>-</td>
<td>-</td>
<td>$\frac{1}{60}$</td>
</tr>
</tbody>
</table>
The theoretical model: Discrete Element Method

The time step $\Delta t$, for the time integration of the particle position, velocity, orientation and angular velocity depends on the time of contact $T_c$, which can be expressed as

$$T_c = \frac{m}{\sqrt{k}}$$

which is estimated based on the single degree of freedom system of mass $m$ connected to the ground by a spring of stiffness $k$. Hence the time step must be sufficiently small to ensure a stable numerical scheme of time integration and Cundall and Strack [1979] proposed that the time step must be smaller than the critical time step $\Delta T_c = \frac{m}{\sqrt{k}}$

3.8 Collision detection

One of the most difficult barriers concerning the DEM simulation is the enormous number of particles, which represents the actual system, due to the reason that as the number of particles increases, the computational time also increases. This problem is inherent in contact-dominated flows. For real life systems, the particle number is usually very large compared to any small-scale models. Therefore, in order to make the simulation technique a really practical notion, Tsuji, *et. al.,* [1993] presented a method where a group of particles where replaced with one big ball and calculated the contact forces only for these balls. He also found similarity laws between cases of large number and small number of particles.

Agrawala *et al.,* [1997] implemented a computational scheme termed "boxing" to reduce the number of particles that have to be tested for potential contact with each
Theoretical Model: Discrete Element Method

entity. To do so, the entire space was divided into small boxes, and a set of pointers was used to store all the particles inside each box in a linked list called a contact list. Hence, tests for potential contacts are done only between particles in the same or in neighbouring boxes.

In this work, we have adapted a methodology to reduce the computational time by calculating a neighbour list for each particle \( i \). The particle \( i \) is said to be the neighbour list of particle \( j \), if the distance \( D \) between the centers of particles \( i \) and \( j \), is less than 2.5 times the larger particle diameter. i.e., if \( D \leq 2.5 \text{dp} \).

Two particles \( i \) and \( j \) are said to be in contact if the distance between their centers is less than or equal to the sum of the radii. Hence the particles that are in contact with particle \( i \) is determined by checking this criteria with the particles in the neighbour list of particle \( i \) only. This procedure reduces lot of computational time since simulation of \( n \) interacting particles using DEM involves \( \frac{n(n-1)}{2} \) pair of contacts.

3.9 Input parameters

The input parameters needed to run the simulations can be divided into three groups viz.

- Geometrical data of the system under investigation
- The particle physical parameters
- The outputs required

The geometrical parameters of the system under study for the present simulation are the diameter, length, fill fraction and rotational speed of the cylinder. The
The particle physical properties are specified by their size distribution, density, stiffness parameters and damping parameters.

The outputs required depends on the type of information required such as the positions and velocities of the particle at different time intervals, the trajectories of a set of sample particles, the variation of normal, tangential forces and kinetic energy of the whole system with respect to time.

3.10 Implementation of the code

The major computational tasks of DEM at each time step can be summarized as follows:

- Finding the neighbour list for each particle
- Detection of contacts between a particle and its neighbours
- Computation of contact forces from relative displacement between particles
- Summation of contact forces to determine the total unbalanced force
- Computation of acceleration from force
- Velocity and displacement by integrating the acceleration
- Updating the position of particles

Based on the theoretical formulation presented above, a software code called DEMCYL was written in Visual Fortran (version 6.6), which simulates the dynamics of granular material motion in a rotating cylinder. Visual Fortran is chosen as a programming language for the following reasons.
Theoretical Model: Discrete Element Method

- It is the most powerful structured programming language for scientific simulation.
- Improved flexibility like dynamic array concept, linking routines etc

The program currently runs on a Pentium IV machine with 256 MB RAM and Windows 2000 operating system. The maximum number of particles that can be used for this system is around 4000 since for each particle, the number of variables that has to stored is 12 and hence the array size becomes Number of particles X 12 and this array size has a significant effect on the program run time.

The program starts by reading all the input variables, and depending on the fill fraction and size distribution the number of particles is determined for each size. Since it is very difficult to generate a packed bed with a velocity distribution, the following methodology was adapted to generate the initial packed bed. First the particles are placed in an orthogonal grid and are assigned random initial velocities.

The particles are given different identities to identify their sizes and are shown in different colours during the output plots. Initially the particles do not experience any inter particle contact forces but experiences only the gravity force. But once the particles start colliding with each other the program computes the inter-particle contact forces. The total kinetic energy of the system is calculated and when the kinetic energy distribution dissipates to zero, the particles has to come to rest.

This data is stored as the initial configuration of the packed bed of particles along with the initial velocity distribution of particles. The cylinder wall is then allowed to rotate with a specified angular velocity and the program performs the computation at each time step. The program stores the position, velocity and force
information for all the particles and hence an enormous amount of memory is needed which increases tremendously with the increase in the number of particles.

The corresponding flow charts are given in Figures 3.4 and 3.5. The flow chart in Figure 3.4 shows the algorithm used for generating the initial packed bed and Figure 3.5 represents the flow chart showing the algorithm used for updating the particle position at every time step, after updating the neighbour list. The total force acting on each particle is calculated using the equations 3.37. Then the particle positions are updated using Gear predictor algorithm. The whole procedure is repeated till the required amount of time or till the system reaches steady state.

Once the program receives the stop signal, it stores the position, the velocity, acceleration, the neighbour list, kinetic energy of each particle at that instant of time as a separate file called "redata". A flag has been set in the "restart" option program. Once the program is restarted, depending on the value of the flag, the program either reads from the redata file or start from the initial distribution.

The code also has a provision to be stopped at intermediate values and it can be automatically restarted using the following procedure.
Calculate the neighbour list
Calculate the collision list of contacting particles

Is collision list = 0

Yes
Calculate the contact forces

No
Calculate the external forces
$t = t + \Delta t$
Calculate acceleration and velocity
Calculate the particle positions

Does the particle settle down completely?

No

Yes
Stop

Figure 3.4: Flow chart showing the initial distribution of solid particles as a packed bed
Theoretical Model: Discrete Element Method

Figure 3.5: Flow chart showing the calculation of particle positions at time $t + \Delta t$
3.11 Special Validation tests

There are two fundamental aspects to test the accuracy and predictability of a program model. First, the program must accomplish what is specified in the model and secondly how accurately the prediction of the model is validated with data from experiments. In order to decrease the chances of error, it is necessary to check the program using different types of validation tests [Asmar et al., (2002)] as follows.

(i) Large runs to inspect the overall running of the program  
(ii) Manual calculations to check small test runs  
(iii) Special cases to inspect the implementation of individual program  
(iv) Inspection of graphical output for the above to see if it looks sensible  
(v) Inspection by an independent assessor  
(vi) Use of specialist software for testing code against specifications.

DEMCYL was tested based on some of the above-mentioned strategies. Since the contact forces calculations are crucial for any DEM simulation some special tests are conducted to test the validity of the software code developed for this purpose.

3.11.1 Effect of normal elastic force in the vertical direction (Test 1)

This test simulates the free falling particle under gravity and also tests the particle-wall contact force. For this test, parameters corresponding to the tangential and damping forces are set zero and the effect of normal elastic force alone is studied. This test also confirms the particle-particle interaction since the wall is represented as a set of particles. The diameter of the particle is 6 cm and the dropping height chosen is 5 cm and the density of the particle is 1300 kg/m³. The value of the normal stiffness parameter is $5 \times 10^3$ units. The particle is dropped from the same height
Theoretical Model: Discrete Element Method

hitting the wall/particle, and the particle rebounds to its original height and the normal elastic force reaches a peak during contact as shown in Figures 3.6 and 3.7

Figure 3.6: y-position vs time
3.11.2 Effect of normal elastic force in the horizontal direction (Test2)

This test is identical to test 1 but with one particle moving in the horizontal direction with initial velocity in x-direction. The gravitational, tangential and normal damping forces are set at zero. The parameters chosen are same as in test 1 and the initial velocity of the particle is taken as 1 m/s. Figure 3.8 shows the results for a particle with initial velocity in the x-direction. As can be seen, the particle rebounds horizontally between the walls with no loss of energy. There is no movement in the y-z directions and no rotation.
3.11.3 Effect of normal damping force (Test 3)

This test is also similar to test 1 but with the normal damping force accounted for. The normal damping constant is taken as 100 s\(^{-1}\). Figures 3.9 and 3.10 shows the effect of damping on the motion of the particle. The particle fails to reach the original height and its height decays due to damping effect. Also it can be seen form Figures the decline in the normal elastic force and damping force at consecutive contacts. There is no movement in the x-z directions and there is no rotation.
Further tests were also carried out to check the validity of tangential forces and the results are found to be in agreement with the expected outputs.

3.12 Conclusion

Since soft sphere approach is found to be the most appropriate method for modelling quasi-static systems like granular flows in rotating cylinder, the theoretical model based on Discrete Element Method (soft sphere approach) is presented in this
Chapter for simulating the dynamics of granular bed motion in the transverse plane of the horizontal rotating cylinder.

After introducing the general concepts of DEM, appropriate governing equations for the present system is presented along with the corresponding expressions for the normal and tangential contact forces. A strategy for finding the neighbour list and collision partners is explained. This is followed by the flow chart of the software code DEMCYL. Finally special validation tests have been carried to check the validity of the code developed. Hence the software code is tested for its accuracy and bugs and then results are generated for the dynamics of particle motion in a rotating cylinder. These results are presented in the next chapter.