THE UNIVERSITY OF CALGARY

On Efficiency of Multibody Approach in Granular Dynamics Simulations

by

Yi Sun

A DISSERTATION

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

CALGARY, ALBERTA

AUGUST, 1996

0-612-20898-2

Abstract

An effective computational methodology based on a multibody approach has been proposed for analyzing and simulating granular systems. Each particle in the system is treated as an independent body. A 3-D model of a granular system consisting of interconnected spheres is treated as a multibody system with variable topology and one-sided constraints between the spheres. The motion of this system is governed by a set of nonlinear algebraic and differential equations. Two formulations (Lagrangian and Newton-Euler) and two solvers (Runge-Kutta and Iterative) are discussed. It is shown numerically for a 2-D model that a combination of the Newton-Euler formalism and an iterative method maintains the accuracy of the fourth order Runge-Kutta solver while substantially reducing CPU time.

The accuracy and efficiency are achieved by integrating the error control into the iterative process. Two levels of error control are introduced: one based on satisfying the position and velocity constraints and another on satisfying the energy conservation requirement. An adaptive time step based on the rate of convergence at the previous time step is introduced which also reduces the simulation time. The efficiency and accuracy is investigated for plane problems: a physically unstable vertical stack of disks, multibody pendulums and a falling chain.

Some methodological aspects of the simulations have been investigated and new algorithms dealing with the storage and updating of system topology data, handling of collision events, application of variable time step technique, et al. have been developed. A FORTRAN program implementing these methods and algorithms has been written.

An application to the problem of jamming in a two-phase flow has been investigated. It is shown that there is a critical particle density above which a jam always takes place and a minimal density below which jam occurrence is not likely. The range between these two densities is the transition zone which requires further investigation. The results of this simulation agree qualitatively with the physical experiments.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to my supervisor Dr. *Oleg Vinogradov* for his help, guidance, and encouragement during my studies and work.

I would like to acknowledge my indebtedness to the University of Calgary and National Research Council for providing financial support to me during my postgraduate studies.

Special thanks to my wife *Hu Baohong*, and my parents *Sun Jiaheng* and *Lu Xuenan*, for their support, encouragement and assistance during my graduate studies.

Finally, I wish to express my gratefulness to my dear friends and colleagues, *Wang Changqing*, *Dai Liming*, *Wu Zengqiang*, *Dmitri Gavrilova* and *Marina Gavrilova*, for their assistance and suggestions, and to my Canadian friends, *Bill Soukoreff* and *Eric Clavelle*, for their help in editing the thesis.

Table of Contents

List of Tables

List of Figures

Nomenclature

A	-	Area
$\mathbf{A}$	-	Coefficient Matrix in Lagrangian Equations
$\mathbf{B}$	-	Coefficient Matrix in Lagrangian Equations
C_A	-	Coefficient of Attenuation
$\mathbf{C}$	-	Diagonal Matrix (Cosine Function of θ)
$\mathbf{C}_\phi$	-	Diagonal Matrix (Cosine Function of ϕ)
DEM	-	Discrete Element Method
E	-	Spherical Excess of the Triangle
$\mathbf{E}$	-	Incidence Matrix
e	-	Coefficient of Restitution
F^n	-	External Force per Unit Mass Acting in Normal Direction
F^τ	-	External Force per Unit Mass Acting in Tangential Direction
f_{iv}^n	-	External Force Acting on the *i*th Particle in the *v*th Link Along the Link Direction
f_{jv}^τ	-	External Force Acting on the *j*th Particle in the *v*th Link Perpendicular to the Link Direction
$\mathbf{G}$	-	Constraint Jacobian Matrix
g	-	Gravitational Acceleration
$\mathbf{H}$	-	Coefficient Matrix of Newton-Euler Equations
h	-	Time Step; Width of Shadow

$\boldsymbol{I}$	Unit Matrix
$\boldsymbol{K}$	Coefficient Matrix of Newton-Euler Equations
k	Factor of Shadow
$\boldsymbol{L}$	Diagonal Length Matrix
LHS	Left Hand Side
l	Length
$\boldsymbol{M}$	Diagonal Mass Matrix
$\boldsymbol{M}_{\psi}$	Diagonal External Momentum Matrix
N, N_b	Number of Bodies
N_e	Number of Links
ODE	Ordinary Differential Equation
$\boldsymbol{P}$	Path Matrix
$\boldsymbol{Q}$	Generalized Forces
QRB	Quasi-Rigid-Body
RSH	Right Hand Side
$\boldsymbol{r}$	Position Vector
$\boldsymbol{S}$	Diagonal Matrix (Sine Function of θ)
$\boldsymbol{S}_{\phi}$	Diagonal Matrix (Sine Function of ϕ)
TOL	Tolerance of Error
$\boldsymbol{U}$	Mass Matrix Associated with Path Matrix
u v w	Index of Links
$\boldsymbol{u}$	Velocity Vector of Flow

V	-	Inverse Mass Matrix Associated with Incidence Matrix
v	-	Velocity
$X\ Y\ Z$	-	Relative Position between Two Particles in Rectangular Coordinates
$x\ y\ z$	-	Absolute Position of Particle in Rectangular Coordinates
α	-	Angle
γ	-	Reaction Force
Δ	-	Error in Position
ε	-	Error in Energy
η	-	Error in Velocity
θ	-	Angle (Generalized Coordinate)
λ	-	Lagrangian Multiplier
μ	-	Coefficient of Friction
ρ	-	Density
ϕ	-	Angle (Generalized Coordinate)
ψ	-	Angle of Rotation
ω	-	Angular Velocity

Chapter 1

INTRODUCTION

1.1 A Historical Review of Granular Dynamics

1.1.1 Origin of Granular Dynamics and Continuum Model

The area of granular dynamics has a relatively short history, starting essentially after the Second World War in the area of molecular dynamics and expanding progressively over the past two decades. Granular matter, often referred to as the "fifth state of matter", is used in applications ranging from pharmaceutical to the construction of buildings and for this reason, has been studied extensively in many disciplines.

Modern industrial materials may be classified as powders or granular solids. The category "granular solids" may also include materials that occur naturally in vast quantities. Granular materials are also found in many industrially important operations and processes. Thus in addition to complex flows such as chute flow, hopper discharge, pneumatic conveying and surface avalanches, granular materials are regularly subjected to compaction, segregation, granulation, fines production, fluidization and so on. Some particulate materials, most typically sand, have been the target of both experimental and theoretical investigations since the '50s (Bagnold, 1954, 1966; Brown and Richards, 1970).

Granular models are also used to describe many physical systems, such as soils, sand, grain, rock, pills, broken ice, et al. It is instructive to define granular materials in terms of their physical features. Table (1.1) categorizes granular media based on size.

Table 1.1 Dimensions of various granular media

Granular Media	Particle size (m)	Number of molecules
molecule	10^{-10}	1
soot particle	10^{-8}	10^{6}
dust, powder	10^{-6}	10^{12}
sand	10^{-4}	10^{18}
gravel	10^{-2}	10^{24}

For matter with a small size ($< 10^{-8}$ m), the thermal energy at room temperature is more important than gravitational energy. This type of matter is not within the scope of this research. Common granular materials are those with large sizes ($> 10^{-4}$ m), such as grain, sand, coal, pellets, and solid particles, et al. An important property of granular materials is that they can flow through hoppers or pipes in a gravity field and each particle has its own physical characteristics, such as shape, size, weight, et al.

Research and development in granular materials dynamics, sometimes called particle dynamics, is based on powder mechanics, but it is different from powder mechanics in both methods and theories. Beginning as early as one hundred years ago, powder mechanics had been developed according to three fundamental principles. They are: the principle of dilatancy, mobilization of friction and minimum energy of flowing granules. Most of the mathematical models used in early research were based on a continuum model only. Correspondingly, partial differential equations were used to describe such systems.

Bagnold (1954) investigated a one-dimensional flow of uniformly dispersed granular materials in fluid under shear. Besides considering the effect of particle interaction, he assumed a constant kinetic energy density and no relative velocity between the fluid and the solid particles. He concluded that in such instances the solid-liquid system could be treated as a liquid medium, and continuum theory could be applied.

Two basic approaches to the modelling of the mechanical behaviour of granular materials appeared in the '70s. The first approach was based on microscopic theory, referred to as particulate theory. This approach considers an ensemble of particles of finite size, and attempts to deduce the laws governing the mechanical behaviour of the entire system. Bagnold's model belongs to this approach.

The second approach is referred to as the macroscopic approach of Jenkins and Cowin (1979) and is based on considering the granular material as a continuous medium. This approach can more readily provide quantitative results, but it loses the concept of individual solid particles and cannot incorporate inter-particle interactions. Both approaches can be applied only to systems that are at, or near to, the closely packed state.

Most researchers have combined the two approaches into a so-called mixed approach (Kanatani, 1979; Ogava, et al.; 1980 and Ackermann and Shen, 1982). Among them, Kanatani proposed a micropolar continuum theory for the flow of closely packed granular materials. He set up quantitative equations based on conservation of mass, linear momentum, angular momentum and energy. By analogy to turbulent flow, Ogave et al., (1980) noted the importance of the fluctuation velocities of the particles which inevitably results from collisions among the particles. Ogave et al., (1980) assumed that a fraction of the particles adhered to the spheres with the remainder reflecting off them with a loss of energy during the collisions. They then proceeded to determine the total rate of change of fluctuation energy by averaging over all possible collisions. Ackermann and Shen (1982) devised a similar model including the effect of the interstitial fluid and the mechanical properties of the solid particles on the rate of change of fluctuation energy. They considered that the rate of dissipation of fluctuation energy depends on the frequency of inter-particle collisions.

Later models which appeared in the '80s, were based on the kinetic theory of gasses (Lun, et al., 1984). The kinetic theory for rapid granular flows developed by Jenkins and Richman (1985) provides a system of transport equations for the mean velocity $\boldsymbol{u}$, the mean density ρ and the granular temperature T, which is a measure of the kinetic energy per unit mass associated with velocity fluctuations. The macro physical feature for the whole system can be obtained based on the three balance equations for mass, linear momentum and fluctuation energy. These can be represented as the following

$$\dot{\rho} + \nabla(\rho \boldsymbol{u}) = 0 \tag{1.1}$$

$$\rho \dot{\boldsymbol{u}} = \nabla \cdot \boldsymbol{P} + \rho \boldsymbol{g} \tag{1.2}$$

$$\rho \dot{T} = -\nabla \cdot \boldsymbol{q} + tr(\boldsymbol{P} \cdot \nabla \boldsymbol{u}) - \gamma \tag{1.3}$$

where an overdot indicates the time derivative. In the equations above, ***P*** is the stress tensor; ***g*** is the body force per unit mass; ***q*** is the flux of fluctuation energy and γ is the rate of energy dissipation in inelastic collisions per unit volume.

These balance equations can never be neglected when a continuum model is used though they may take a different form in some papers. It remains to be determined whether a continuum model is justified when one or more spatial dimensions of the problem are measured in terms of a small number of grain diameters. Shear flows (Hanes and Inman, 1985; Savage, 1984) are often only a few to a few tens of grains thick. In a collision model, substantial shear across a relatively small number of grain diameters is a direct consequence of grain inelasticity (Haff, 1983; Hui and Jaff, 1986). Thin shear zones are also predicted by constitutive models incorporating the effect of frictional contacts (Johnson and Jackson, 1987). Indeed, it is in granular fluids that one has perhaps the clearest microscopic view of the origin of one kind of shear band. Until the '90s, some researchers, such as Babic (1993), still used kinetic theory to solve the problem of granular flow.

1.1.2 Discrete Model and Computer Simulation

So far, the models we have discussed are all continuum models. In the research of granular dynamics, continuum models are not suitable for analyzing loosely packed systems or systems with non-homogenous dispersion of solid bodies. Moreover, these models cannot be used to analyze processes in which identification of critical areas of solid body accumulation or predictions of jamming are essential. Considering the disadvantages of continuum models and the difficulties in application, a new approach to the modelling of such systems based on discrete analysis was proposed by Cundall and Strack (1979). Perhaps the first attempts at discrete modelling of granular systems were done by Cundall (1971). In this approach, each solid particle or rigid body in the system is looked at as an individual entity. The interactions among the particles depend on their physical properties rather than on averaging procedures, as is the case in continuum models. Following the discrete models, the Discrete Element Method (DEM) appeared and it has been utilized in granular dynamical systems. Especially with the rapid growth of computer power, this method has widely developed in the area of granular dynamics and the computer simulation of granular systems. The DEMs will be discussed in detail in Chapter 2.

In studies of granular dynamics, the discrete model, including the DEM algorithm, has been utilized and improved for granular systems by many people since it was first developed by Cundall. One of its main advantages is that it has a very simple mathematical model and is computationally efficient. This method, however, does not describe a real system accurately enough. In other words, this model needs a very short time step for a system with a high density of particles.

Other recent approaches for simulating discrete granular systems are based on Cundall's model and the DEM. Campbell and Brennan (1985a,b) and Campbell (1989) have also studied steady state rapid shear flow of rough particles in both two and three dimensions. In two-dimensional chute flow simulations, Campbell and Brennan (1985a) identify a high-temperature, low-density zone next to the base of the chute. They note that two-dimensional simulation results depend strongly on system parameters such as friction coefficients.

Baxter and Behringer (1990) have developed a cellular automaton to simulate the flow of irregular particles from a wedge-shaped hopper. Cellular automata are discrete, lattice-based models with simple evolution rules. Cellular automata can be used to describe complex behaviour in extended systems and they have significant computational advantages for simulating many interacting particles in systems with complex geometry.

Gutt and Haff (1990) and Fitt and Wilmott (1992) have also used cellular automata to model granular flows including steady chute flow, flow down a vertical channel and particle size segregation. It is clear that lattice-based models make powerful representations of flowing granular systems over a range of dynamic regimes and that their development to include three dimensions, isotropy, disorder and particle properties (such as coordination and shape) are valuable areas for future research.

Since the Cundall model was proposed, discrete models have developed into many branches according to various specific applications. One of the applications is ice mechanics (Hopkins, et al., 1985 and Loset, 1994).

Another use of discrete mechanics is to study in detail system behaviour at the level of the individual grain or of a few individual grains, i.e., at scales that are obliterated by the averaging necessary to generate the partial differential equations. Studies of mixing of sediments caused by fluid traction, see (Jiang and Haff, 1993) or of the development of sedimentary stratigraphy (Forrest and Haff, 1992) reveal interesting details of the process and structure relevant to the very small scales, e.g., a few particle diameters are often important for geological interpretation or environmental remedy. The studies of granular microstructure (Campbell and Brennan, 1985b; Hopkins and Louge, 1990 and Walton et al., 1991) involving the existence of clumps and chains of small groups of particles may also begin to inform us of ways in which assumptions of molecular kinetic theory might be relaxed.

Methods based on computer simulation provide alternative approaches to problems in discrete mechanics. One method is to model a collection of particles as a system of hard spheres (Campbell and Brennan, 1985b; Haff and Werner, 1987). In the hard sphere model, when a contact between two particles is detected, the relative velocity of the two colliding particles is transformed so as to conserve the total two-particle momentum. Another method is to model particles as soft spheres. The first discrete element model with the soft-particle approach was developed by Cundall and Strack (1979).

Although there are several different approaches in the application of discrete models and DEMs, the basic modelling principle is still based on Cundall's model. There were no essential distinctions among these approaches until a different approach to discrete analysis was postulated by Vinogradov (1985), and further developed by Springer (1989) and Wierzba (1991).

In the new approach used in (Vinogradov, 1985), rigid bodies are approximated by disks and the whole system is treated as a multibody dynamics system while the mathematical formulation is based on Lagrangian dynamics. In such a system, the equations of motion are derived in terms of generalized coordinates which are the direction angles between two disks. Such approach leads to a set of *Differential Algebraic* Equations (DAEs) based on the system topology. It is important to note that the differential equations in the above system are coupled.

The advantage of such a model is that it describes accurately the mechanics of granular materials including momentum transfer, friction force, et al. The disadvantage of this method, however, is the need to solve a nonlinear system of coupled differential algebraic equations. Besides the numerical difficulties associated with DAEs, there are other computer-intensive calculations, such as handling collisions and constraints, updating the topology and relevant physical parameters.

The multibody dynamics model of a granular system with variable topology is a remarkable innovation in the study of granular dynamics. The challenge of applying a multibody dynamics approach to a granular system comprised of a large number of particles is computational, i.e., the simulations should be done in a reasonable time while maintaining the desired accuracy of results. Although there is extensive experience in this area in machine dynamics, it cannot be applied directly to a granular system because, first of all, machines are systems with fixed topology while granular systems are systems with variable topology. Secondly, granular systems may have internal degrees of freedom due to sliding between the particles and, third, granular system are relatively large. As in machine dynamics, both the formulation of equations and of the equation solver affect the efficiency and accuracy of the simulation in a granular system.

A FORTRAN program developed by Springer (1989) and Wierzba (1991) was used to simulate the movement of ice blocks in an open river channel. This program can also be used for the simulation of an arbitrary planar multibody system in which the bodies are modelled by disks. The model used in this program is simple and suitable for simulation of granular materials. However, the method for the analysis and handling of various events (e.g., collisions and new disk generation) is complicated. In the numerical solution, the traditional Runge-Kutta method and matrix operations, which are computationally expensive and need a large amount of storage space, were employed. Obviously the size of a granular system was limited by the capacity of the computer and CPU time constraints.

1.2 Objectives of Research

Granular materials dynamics is currently one of the most active fields in applied mechanics. As with any growing field, it is expanding into a number of subfields. The most active subfields are: modelling, governing equations formulation, solution methods, computational methods, graphical representations and applications. In a specific application, almost all subfields above will be involved. In order to implement computer simulation of a granular material system, one can use different methods from respective subfields. The combinations of these methods can lead to different ways of achieving our objective, which is finding the most efficient computer simulation of a large-scale granular system.

When a multibody dynamics model is used for granular systems, the accuracy can be greatly improved. However, it requires a more complicated mathematical model and additional requirements for computer time and storage space. Efficiency is a major problem in computer simulations of granular materials as multibody systems and it is the main focus of this thesis.

In considering a larger size system with variable topology, a dynamical data structure was introduced by Sun, et al., (1994). This was the first step for improving computational efficiency. The improvements included: governing equations based on a Newton-Euler approach and formulated in the decoupled form of first and second order derivatives; a mixed iterative scheme to solve the governing differential equations and error control and correction techniques based on physical conditions such as geometric constraints and energy conservation. These methods not only improve the accuracy of solutions but also eliminate violations of both velocities and positions.

The objective of the present research is to find more efficient algorithms and computational methods for simulations of granular materials as multibody systems.

In theory such subjects as topological description and updating of the system, derivation of equations of motion, random generation of particles, events handling and error control are investigated numerically. More specifically, the following topics are investigated:

(1) The efficiency of the iterative scheme in numerical solution of equations of motion for a granular system of large size.

(2) The use of geometrical constraints and the energy conservation for error correction and control.

(3) Dynamic data storage and updating in an array form.

(4) The elimination of matrix expressions and operations in system data storage and generation of equations of motion.

(5) The improvement of methods of events handling.

1.3 Organization of Text

A general review of the subject of granular dynamics and a brief introduction to this thesis is given in Chapter 1.

In Chapter 2, the subject of modelling of granular systems will be discussed in detail, including assumptions and suggested approaches. In addition, the graph analysis and associated matrices are given as preliminaries for later applications.

The governing equations of motion of the system will be given in Chapter 3 for two approaches: Lagrangian and Newton-Euler. However, only the Newton-Euler approach will be discussed in detail because the Lagrangian approach can be found in the literature.

Some special topics, such as various action forces, collision events and random generation of initial conditions will be discussed in Chapter 4.

An iterative method used for solving the equations of motion will be discussed in detail in Chapter 5. This is a customized algorithm which incorporates error correction into the iterative cycles. The flowcharts for these algorithms are also given in this chapter.

In Chapter 6, several simple and specific applications are given as examples to test our new algorithms. The obtained results are compared with those obtained by classical numerical methods with controlled accuracy. A FORTRAN program incorporating developed algorithms is briefly discussed. Known experimental results are used to validate this program.

Finally, the conclusions and recommendations are given in Chapter 7.

Some derivations of major formulations can be found in the appendix. In addition, the FORTRAN program for the simulation of granular materials in a straight pipe with a rough wall is also given in Appendix E.

Chapter 2

MATHEMATICAL MODELLING OF GRANULAR SYSTEMS

2.1 Introduction

A mathematical model represents a complicated physical system and its validity should be checked against physical system behaviours. Generally, two types of models are used in dynamical studies of granular-type materials: one describing the material as a continuum; and another describing it as a discrete system. In this thesis, we will discuss discrete models.

Our purpose is to simplify the discrete model so that it can be used efficiently to describe granular systems. If the currently available models and solvers for multibody systems, such as Nastran and ANSYS, are used for computer simulations of granular materials, the size of the system will be limited by the computer capacity and the computational speed will be limited by the computer's CPU speed. Although large computers are available, it is not an economical way for us to solve a simple granular system.

Besides accuracy, computational efficiency is an important factor which should be considered in computer simulations of granular systems. With reference to the old model and algorithm (Springer, 1989) and (Wierzba, 1991), the related assumptions and approaches have to be revised for improvement of computational efficiency. This will also be discussed in this chapter.

The main features of granular systems are their large scale and the variable topology of the configuration structure. In order to express the topology efficiently, a graph and associated matrices are introduced in the derivation of the equations of motion. The graph and its associated matrices are not new concepts and have been used for multibody systems with a certain topology (Roberson and Schwertassek, 1988).

The advantages of utilizing matrices in topology description are not only because simplicity and clarity, but also easier updating. On the other hand, because matrices take more storage space and computational time, they reduce computational efficiency in simulations of granular systems. In this and the following chapters, matrices are still used to express equations of motions. The algorithms in the new simulation program are not based, however, on matrix operations.

2.2 Discrete Models

2.2.1 Distinct Element Method

Some mechanical systems are composed of physically distinct elements the number of which is relatively small so that their description as a continuum is not warranted. Alternatively, some systems though large in number cannot be linked to macroscopic behaviour through presently known or constitutive laws. If interaction forces between individual elements are known or can be estimated and modelled, then the behaviour of these elements or particles can be studied by solving the Newton equations of motion for each particle in the group simultaneously. This method is called the *distinct element method* (DEM).

In describing the above dynamic behaviour numerically, time steps are taken over which velocities and accelerations are assumed to be constant. The DEM is based on the idea that the time step chosen may be so small that during a single time step disturbances cannot propagate from any disk further than its immediate neighbours. Then the resultant forces on any disk are determined exclusively by its interaction with the disk with which it is in contact. The resulting overlap of the displaced particle with the neighbours is translated into some internal forces, which are treated as external during the next time step for the body which was motionless. The interactions are thus one-to-one (as in molecular dynamics). The DEM approach takes into account the variability of the system density, the internal degrees of freedom, the irregular shapes of the particles and various particle properties. This approach has been modified and used in many applications. The literature is very extensive and the reader is referred to recent reviews in this area for more details (Savage, 1995; Barker, 1994 and Haff, 1994).

The DEM is a simple algorithm for granular dynamics, which can be applied to a large-size system without the need for an excessive memory and complicated calculations. Many improvements of the original DEM algorithm have been made by Walton and Braun (1986), Rothenburg and Bathurst (1992) and Borja and Wren (1995). However, the basic assumptions in all models remain the same.

Since the DEM explicitly integrates Newton's second law for every particle, an idealized sample in quasi-static experiments must be loaded at a slow enough rate for inertial effects to be negligible. In fact, even with an extremely slow rate of loading, DEM still uses inertial effects to predict the future position of each particle since the particle displacements are calculated based on the bounces and collisions that they make with other particles.

Moreover, the DEM (Cundall model) looks like a model of molecular dynamics in which the particles (molecules) are always separated and the interactions are one-to-one. The looser the system is, the better results the DEM should give. In a closely packed system almost all particles may form into clusters and behave as quasi-rigid bodies during some time interval. In this case, the results by the DEM could be erroneous. The reason is that the effect of simplifications of the DEM has never been investigated in the field of granular dynamics. However, these simulations may be important in problems dealing with local phenomena, local instabilities, the timing of the nonsteady processes, problems of particles jamming.

2.2.2 Multibody Dynamics Model

Multibody dynamics, as a branch of dynamics, has grown rapidly since the '60s. The task of multibody dynamics is the analysis and prediction of the dynamic behaviour and performance of physical systems, very often comprising a large number of components, also called bodies or segments, acting together as a single entity. To analyze and study such systems, the various components must be identified and their physical properties determined. Once the characteristics of individual components are known, a mathematical model is constructed which represents an idealization of the actual physical system. For the same physical system it is possible to construct a number of mathematical models. The most desirable is the simplest model that retains the essential features of the actual physical system.

When granular systems are described as multibody systems, several choices have to be made in defining the methodology, such as:

(1) the dynamical formalism,

(2) the set of dependent variables,

(3) the method for describing the system topology and mechanical properties of the system bodies and their interconnections,

(4) the frame of reference with respect to which the system motion is described, and

(5) a description of the system's initial configuration.

Obviously, the multibody dynamics method is a more accurate algorithm than DEM in the description of a granular system because it satisfies the constraint requirements explicitly. It is, however, more complicated and computationally expensive. This is probably why it is not used in granular mechanics.

Apparently, many dynamical systems of granular materials cannot be completely described as a continuum. In addition, it is impossible to investigate the physical behaviour of every particle in a granular system and simulate their dynamics process using macro parameters of the system. Therefore, the discrete model should be adopted as the basic mathematical model of granular systems. Although computers are becoming more and more powerful in both computing speed and memory space, the size of the system is still limited, especially in computer simulations. Hence, the existing discrete models for multibody systems need to be improved so as to be suitable for granular-type materials.

2.3 Assumptions and Approaches

Many granular-type materials can be represented by a two-dimensional model with regularly shaped particles, such as a disk or rectangle. A spherical model can be used for a three-dimension system (Vinogradov, 1992).

In a multibody system, each body has its geometrical parameters (length, width), physical parameters (density, mass, inertia, variable internal and external forces) and state parameters (position, velocity, angular velocity). Some of them are constants and several of them are variables. All of these can be used in granular materials systems. First of all, the shape of bodies can be a simple sphere in a 3-D case or a disk in a 2-D case. Then only one geometrical parameter, radius, for each body is introduced. In addition, the angular inertia of particles is neglected because the angular position of each particle is considered to be not important.

A granular system, depending on how complicated it is, can comprise some bodies, boundaries and clusters of interconnected bodies. Also, the boundaries can be considered as bodies whose velocities are zero and masses are infinitely large.

In taking into account the assumptions above, a point-mass truss model that was used for a multibody system (Wierzba, 1991) can be employed for a granular system.

2.3.1 Point-mass Truss System

The basic assumptions of a point-mass truss system are:

(1) each body or particle in the system consists of one point mass,

(2) connections between two point masses are weightless rigid links,

(3) each centre of the point mass is an ideal joint without friction torque, and

(4) external forces act directly on the point masses.

From assumption (3), it can be deduced that the links are loaded by only axial forces.

2.3.2 Formulation of Constraints

There are various kinds of constraints in multibody systems, as for example, in robots or mechanisms. However the constraints in granular systems are different since they are one-sided constraints between bodies and bodies and boundaries. In a broad sense, the boundary can also be thought of as a group of bodies or particles with a finite or infinite radius and an infinite mass. As mentioned above, the radii of boundaries can be infinite (for a straight line) or negative (for a concavity). So there is only one type of constraint which can be expressed mathematically as

$$|\boldsymbol{r}_i - \boldsymbol{r}_j| = c_{ij} \tag{2.1}$$

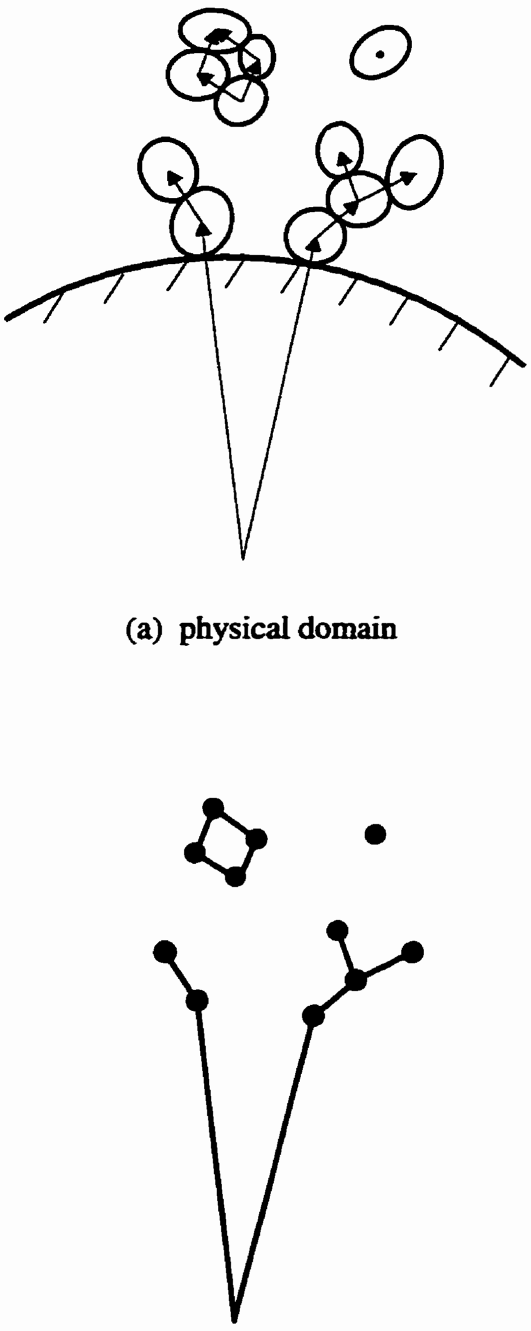
$$v_i^n - v_j^n = 0 \tag{2.2}$$

where $\boldsymbol{r}_i$ is the position vector of *i*th body; c_{ij} is a constant and the v_i^n denotes a velocity component of *i*th body in the normal direction (parallel to the link). Equations (2.1) and (2.2) are, respectively, position and velocity constraint conditions.

When a point-mass truss system is used, the links can take both compressive and tensile forces which are so-called constraint forces, denoted here by γ_{ij} , where *i* and *j* are the indexes of particles. Two forces are equal and apposite in their actions on particles *i* and *j*. They act along the centre line of particles *i* and *j*. If the constraint force is tensile then $\gamma_{ij} > 0$, and if it is compressive then $\gamma_{ij} < 0$. In granular materials the constraint disappears if the reaction force γ_{ij} is positive. Thus the condition that the link exists between the particles *i* and *j* is

$$\gamma_{ij} \leq 0 \tag{2.3}$$

Figure (2.1) illustrates a typical example of a granular system and its corresponding mathematical model.



(a) physical domain

(b) corresponding point-mass truss model

Figure 2.1 A typical example of a granular system using multibody dynamics model

2.3.3 Generalized Constraints and Overconstrained Systems

If a body is in contact with other bodies or boundaries, its motion is restricted in the normal direction. This restriction is called here the *general constraint.*

In a coupled multibody system, if the topology of the system is a tree-like one, then the system can be solved in generalized coordinates (Vinogradov, 1993a). In this case the constraints are imposed without their explicit formulation, and then the number of generalized coordinates is equal to the number of degrees of freedom of the system.

When the topology of the system has a circuit or a closed-loop, then a constraint equation is written in an explicit form (Vinogradov, 1993a). Those extra coordinates, which need to be determined by additional constraint equations, are called here the *generalized constraints* so as to distinguish them from the *general constraints.*

For a system in which all bodies are coupled and the number of bodies is N_b (excluding the number of boundaries N_f), the conditions for a system without generalized constraints are: 1) $N_f \geq 1$ and 2) $N_e = N_b - N_f + 1$. Where N_e is number of links.If $N_e > N_b - N_f + 1$ or $N_f > 1$, then generalized constraints exist, and their number N_c can be determined by

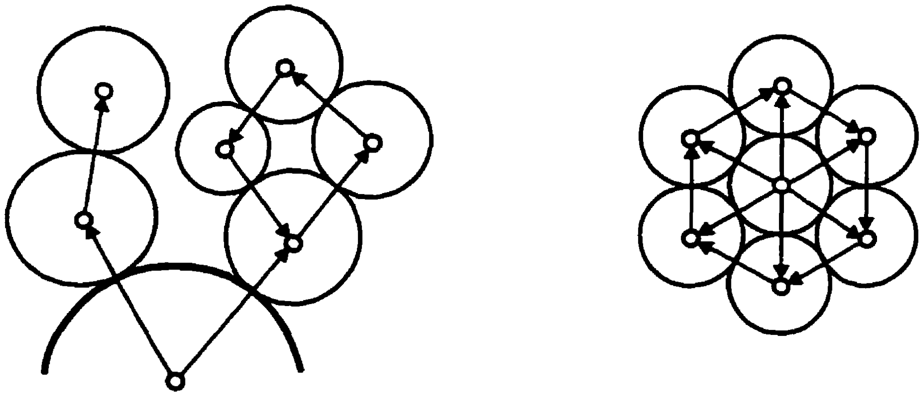
$$N_c = N_e - N_b \tag{2.1}$$

The overconstrained system is one in which all algebraic equations to describe the generalized constraints are linearly dependent. In the constrained system, the overconstraints occur only if

$$N_e > 2N_b - 3 \qquad (2.2)$$

Note, that N_b includes N_f if these bodies are connected with the boundaries.

These different constraints can be illustrated clearly by Figure (2.2) as follows



(a) A system with a generalized constraint (b) An overconstrained system

Figure 2.2 Examples of constraints

In Figure (2.2a), the number of bodies $N_b = 6$ and the number of links $N_e = 7$, so the number of generalized constraints is $N_c = 7 - 1 = 1$. In Figure (2.2b), the $N_b = 7$ and $N_e = 12$, which satisfy the Equation (2.2), therefore it is an overconstrained system.

The overconstrained system does not have internal degrees of freedom. Thus the motion of bodies belonging to it are functionally dependent, which means that such a system behaves as a rigid body. For such systems, a concept of a *Quasi-Rigid-Body* (QRB) was introduced by Vinogradov and Springer (1990).

2.4 Topology and Associated Matrices

A granular system is a system with variable topology. In computer simulations, significant computer time is spent on updating the equations of motion because the equations are topology dependent.

To create a computer-oriented general dynamical formalism for granular systems, one must devise a means for telling the computer how the system is connected. Therefore, a data structure that can be used to keep track of interconnected bodies must be chosen and it must be done in a way that can be implemented easily on a computer.

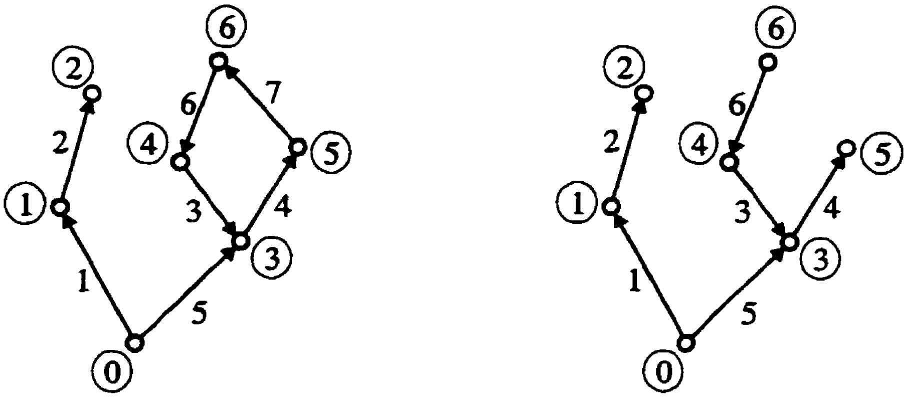
A simple geometrical expression of topology is a graph which shows the interconnection between the bodies and boundaries. Associated with the graph, a matrix or pointer array can be employed as a mathematical form of topology.

2.4.1 Graph Analysis of Topology

A graph that can summarize a lot of information is a very useful tool in many fields of science and technology. Let us consider a simple graph in which there are only two kinds of elements: points and lines. A point, also called vertex, can be an isolated point in space or one on the end of lines. A line is said to connect or join the two vertices. In other words, a line must be associated with two vertices.

When such a graph is used to represent a topology for a granular system, each particle can be described by a point in the graph and the connection between the particles is expressed by lines. In considering regular shapes and no overlap for all particles, the further limitation for the corresponding graph is that all lines on the graph are straight lines with direction.

Figure (2.1) shows an example of a simple graph. The circled numbers from 0,1,2, ... N_b are used to denote the vertices and the numbers without a circle (1,2, ... N_e) are used to denote lines.



(a) A circuit graph

(b) A tree graph

Figure 2.3 Examples of graphs

Such a graph can be used to describe the topology of the granular system no matter what the planar multidisk system or spacial multisphere system is.

According to the constraints requirements, a graph can be classified as follows:

Tree Graph

Definition: *In a graph, if after any one of the lines is removed and the graph falls into two separate pieces, then this graph is called a tree graph, or open loop.*

For N lines there are N+1 vertices in a tree graph. It is convenient to designate one vertex as a reference point in analysis of a tree graph. Any point or vertex can be designate as a reference point, which is also called the root of the tree, and its number can be denoted as 0.

The simplest tree graph has one point (root). A planar system with N bodies and a tree like topology has N degrees of freedom.

Circuit Graph

Definition: *If in a tree graph one or more lines are added while the number of vertices is unchanged, then the tree graph is transformed into a circuit graph, also called a closed loop.*

The simplest circuit graph consists of 3 points and at least three lines. The number of degrees of freedom of a planar system with N bodies and a closed loop-like topology is less than N. Figure 2.2. shows a circuit graph and a tree graph.

An arbitrary graph comprises tree branches and loops. Such a graph can represent a topological structure of a granular system.

2.4.2 Matrices Associated with a Graph

A graph is described by a matrix which is defined in terms of vertices and their connections. There are two matrices associated with a graph which are interdependent.

(1) Path matrix

For a tree like graph, assume the 0*th* vertex is the root (reference point), a *path matrix* can be represented by

$$\boldsymbol{P} = [P_{ij}] \tag{2.3}$$

in which each component P_{ij} has the following meaning:

(a) $P_{ij} = 1$ if the direction from i to j is the same as from the root to the vertex j;

(b) $P_{ij} = -1$ if the direction from i to j is opposite to that from the root to the vertex j;

(c) $P_{ij} = 0$ if i and j are not on the same path from the root.

When a path matrix is used to present a tree-like graph, a reference point, i.e., the "root" needs to be identified. Since any point can be used as a reference point, the path matrix has many different forms for a tree like graph. Moreover, a path matrix can not completely represent a circuit graph.

(2) Incidence matrix

For an arbitrary graph whether it is a tree or a circuit, or whether it is composed of multi trees or multi circuits, a matrix associated with this graph can be represented by

$$\boldsymbol{E} = [E_{ij}] \tag{2.4}$$

where each component E_{ij} has the following meaning:

(a) $E_{ij} = -1$ if the direction of *v*th line points towards the *i*th vertex;

(b) $E_{ij} = 1$ if the direction of *v*th line is in the direction opposite to the *i*th vertex;

(c) $E_{ij} = 0$ if i and j are not connected.

The matrix $\boldsymbol{E}$ with N_b rows and N_e columns is called the *incidence matrix* (Roberson and Schwertassek, 1988).

The columns of the incidence matrix correspond to a line in a graph and the two non-zero elements identify the vertices on this line. An arbitrary graph has a unique incidence matrix.

For the Figure (2.2)a the incidence matrix is

$$\boldsymbol{E} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{2.5}$$

and the path matrix for Figure (2.2)b is

$$\boldsymbol{P} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{2.6}$$

(3) Relationship between the two matrices

The relationship between the path matrix and the incidence matrix can be, according to (Roberson and Schwertassek, 1988), expressed as

$$\boldsymbol{E}\boldsymbol{P} = -\boldsymbol{I} \tag{2.7}$$

where $\boldsymbol{E}$ is N_b by N_e matrix representing a tree-like topology and $\boldsymbol{I}$ is a unit matrix. Note, that $\boldsymbol{E}$ can represent a closed-loop topology, while $\boldsymbol{P}$ cannot. Also, the sign in Equation (2.7) may be different in some books and papers, such as (Ju, 1989), depending on the convention adopted.

2.5 Summary

A discrete mathematical model has been discussed in this chapter. In considering the features of a granular system and through corresponding simplifications, the point-mass system can be used as a basic model of a granular system.

The constraint requirements are very important in the mathematical modelling of granular systems. Here the simplest constraints requirements are formulated. Some concepts dealing with constraints will be discussed in later chapters.

Due to topological variability of granular systems, graph theory is used to describe it. Two associated matrices, incidence and path, have been discussed here. They will be used in the next chapter to describe different formulations of the dynamics equations.

Chapter 3

EQUATIONS OF MOTION

3.1 Introduction

In this chapter two formulations of the equations of motion are used: Lagrangian and Newton-Euler formulations. There are essential distinctions in the form of these equations. The former have been used widely in multibody dynamics systems, the latter were seldom used because there are additional unknowns in the equations. Since the Newton-Euler equations can give us more advantages in numerical calculation, they will be used in the following chapters and their derivation will be given in this chapter in detail.

Because the derivation of the equations of motion is based on the topology of system while the topology is described by the associate matrix ***E*** or ***P***, the expression of the equations of motion will be in matrix form in this chapter. In addition, some diverse and complicated processes for the derivation of equations are omitted in some sections and put in the appendix instead.

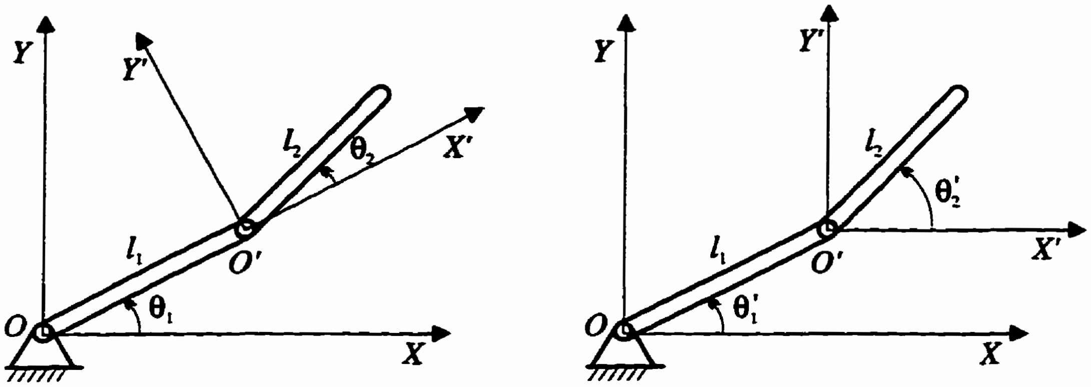
3.2 Coordinate System

A mathematical model is usually dependant on a set of specific coordinates. Proper selection of the coordinates allows one to obtain simpler expressions for positions, velocities and accelerations and faster algorithms in the derivation of the governing equations. Apparently, the use of a set of rectangular Cartesian coordinates is a simple and direct way to simulate granular systems.

In order to get a simple form of the equations of motion, most people often use generalized coordinates in analysis of dynamics.

In a planar granular system, direction angles of centre lines of connected disks are considered as generalized coordinates. The position of each body in space can also be determined by angles in a spherical coordinate system. In order to describe a physical system conveniently, relative coordinates are often used. However, a promising area for improving computational efficiency is to use absolute coordinates. This has been discussed by Huston, et al., (1994).

Absolute coordinates measure the position and orientation of the bodies of the system in space as opposed to measuring the position and orientation relative to adjoining bodies. Two kinds of coordinates are shown in Figure (3.1)



(a) Relative angles

(b) Absolute angles

Figure 3.1 Relative and absolute coordinates

According to the results of Huston, et al, (1994), the principal advantage of using absolute coordinates is simplicity of the resulting equations leading to more efficient and more accurate numerical solution and the advantage of relative coordinates is only their intuitive description of the physical system. Hence, absolute coordinates are chosen here in the derivation and solutions of the governing equations. In addition, rectangular coordinates are also used to derive another form of the equation of motion, which can give simple and explicit integral expressions.

3.3 Lagrange Equations

3.3.1 Lagrangian Approach

The equations of motion can be derived by the standard Lagrange function which is defined as

$$L = T - V \tag{3.1}$$

where T is the total kinetic energy of the system and V is its corresponding potential energy. The equations of motion are then obtained by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0 \tag{3.2}$$

If the system is not conservative, (for example, there are friction forces and some time-varying forces in the system), the right side will be the corresponding forces rather than zero. In addition, There is another form of Lagrange equations which is presented as

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} = Q_i \tag{3.3}$$

where Q_i is generalized force acting on the *i*th body. The equations of motion derived by the equation above are called Lagrange equations here. The detailed procedure of derivation for planar and spacial systems can be found in (Springer, 1989) and (Vinogradov, 1992) respectively.

Assuming there is a multibody system with N bodies and the topology of this system is represented by a path matrix $\boldsymbol{P}$, ($\boldsymbol{P} = [p_{ij}]$), then the equations of motion could be, with reference to (Springer, 1989) and (Vinogradov,1993a), written as

$$\sum_{j}(\sum_{k} p_{ik}p_{jk}m_k)\cos(\theta_i-\theta_j)l_j\ddot{\theta}_j + \sum_{j}(\sum_{k} p_{ik}p_{jk}m_k)\sin(\theta_i-\theta_j)l_j\dot{\theta}_j^2 = Q_i \qquad (3.4)$$

If the components of external forces acting on the kth body are f_{xk} and f_{yk} , then Q_i can be presented as

$$Q_i = -\sin\theta_i(\sum_{k} p_{ik}f_{xk}) + \cos\theta_i(\sum_{k} p_{ik}f_{yk}) \qquad (3.5)$$

where $i = 1,2...N_e$ and m_k is the mass of the kth body and l_j is the length of the jth link and θ_j is the generalized coordinate. $\dot{\theta}_j$ and $\ddot{\theta}_j$ are respectively the first and second derivative of θ_j. If the topology of the system is tree-like or called an open loop, and the number of independent variables is N_e, then the Equation (3.4) is a set of pure ordinary differential equations. Otherwise, the equations become a set of algebraic differential equations, which will be discussed in the next section.

Equations of motion, i.e., Equations (3.4) and (3.5) can also be written in a matrix form as following

$$\boldsymbol{AL}\{\ddot{\theta}_i\} + \boldsymbol{BL}\{\dot{\theta}_i^2\} = \{Q_i\} \qquad (3.6)$$

where

$$\boldsymbol{A} = \boldsymbol{C}\boldsymbol{P}\boldsymbol{M}\boldsymbol{P}^T\boldsymbol{C} + \boldsymbol{S}\boldsymbol{P}\boldsymbol{M}\boldsymbol{P}^T\boldsymbol{S} \tag{3.7}$$

$$\boldsymbol{B} = \boldsymbol{S}\boldsymbol{P}\boldsymbol{M}\boldsymbol{P}^T\boldsymbol{C} - \boldsymbol{C}\boldsymbol{P}\boldsymbol{M}\boldsymbol{P}^T\boldsymbol{S} \tag{3.8}$$

and

$$\{Q_i\} = -\boldsymbol{S}\boldsymbol{P}\boldsymbol{M}\{f_{xj}\} + \boldsymbol{C}\boldsymbol{P}\boldsymbol{M}\{f_{yj}\} \tag{3.9}$$

In the equations above, the notations in braces are a set of arrays. The bold letters identify matrices. ***M***, ***L***, ***S*** and ***C*** are all diagonal matrices of order $N_e \times N_e$ (note that the N_e is equal to N_b for a system with a tree-like topology) and are defined as, respectively

$$\boldsymbol{M} = \begin{bmatrix} m_1 & 0 & . & 0 \\ 0 & m_2 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & m_{N_b} \end{bmatrix}_{N_b \times N_b} \tag{3.10}$$

$$\boldsymbol{L} = \begin{bmatrix} l_1 & 0 & . & 0 \\ 0 & l_2 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & l_{N_e} \end{bmatrix}_{N_e \times N_e} \tag{3.11}$$

and

$$C = \begin{bmatrix} \cos\theta_1 & 0 & . & 0 \\ 0 & \cos\theta_2 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & \cos\theta_{N_e} \end{bmatrix}_{N_e \times N_e} \tag{3.12}$$

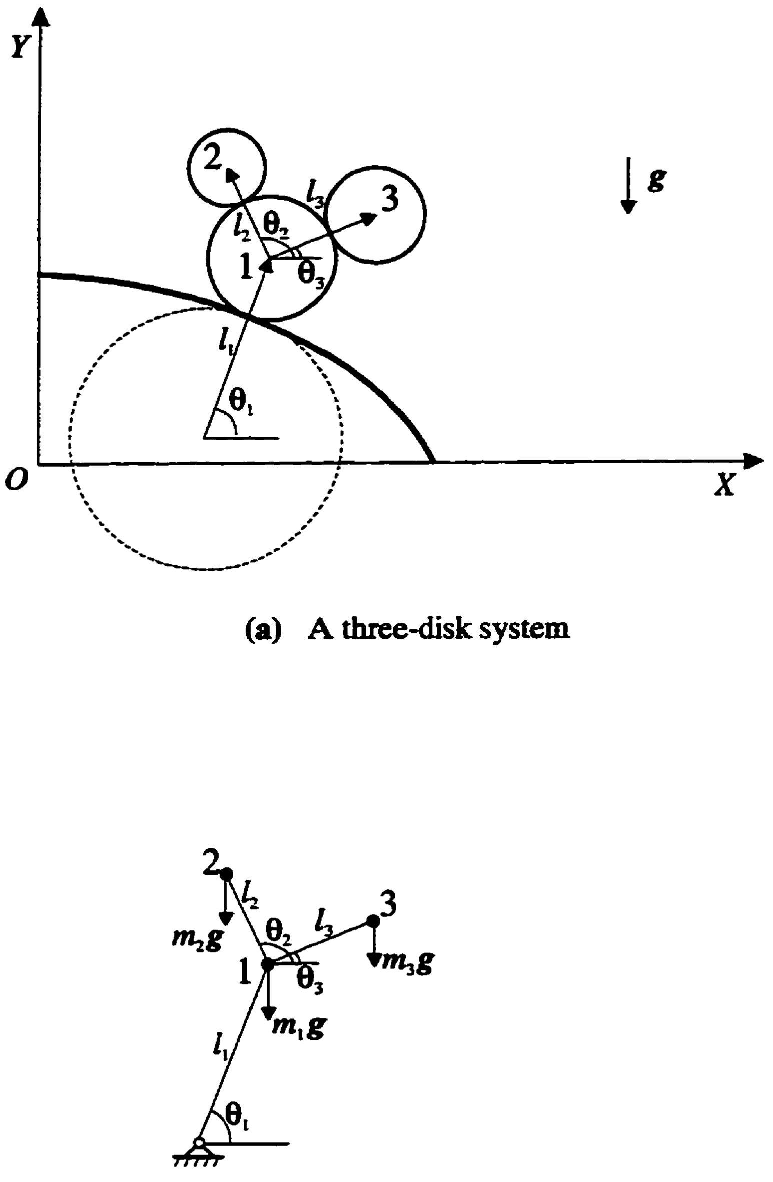
$$S = \begin{bmatrix} \sin\theta_1 & 0 & . & 0 \\ 0 & \sin\theta_2 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & \sin\theta_{N_e} \end{bmatrix}_{N_e \times N_e} \tag{3.13}$$

Equations (3.10)~(3.13) can also be written in the following simple forms

$$\begin{aligned} &\boldsymbol{M} = dial(m_j) && j = 1,2,...N_b \\ &\boldsymbol{L} = dial(l_i) && \\ &\boldsymbol{S} = dial(\sin\theta_i) && \Big\} \; i = 1,2,...N_e \\ &\boldsymbol{C} = dial(\cos\theta_i) && \end{aligned} \tag{3.14}$$

3.3.2 An Example

A specific application of these equations of motion in matrix form can be illustrated via the following example which Assumes a simple planar system with a fixed boundary and three disks [shown in Figure (3.2)a]. and its corresponding point-mass truss system [shown in Figure (3.2)b].



(a) A three-disk system

(b) Corresponding point-mass truss system

Figure 3.2 A three-disk system

In this example the path matrix $\boldsymbol{P}$ is

$$\boldsymbol{P} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

and

$$\boldsymbol{M} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \qquad \boldsymbol{L} = \begin{bmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{bmatrix}$$

$$\boldsymbol{S} = \begin{bmatrix} \sin\theta_1 & 0 & 0 \\ 0 & \sin\theta_2 & 0 \\ 0 & 0 & \sin\theta_3 \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} \cos\theta_1 & 0 & 0 \\ 0 & \cos\theta_2 & 0 \\ 0 & 0 & \cos\theta_3 \end{bmatrix} \tag{3.16}$$

The external forces are

$$\begin{aligned} \{f_{xj}\} &= (0, 0, 0) \\ \{f_{yj}\} &= (-m_1 g, -m_2 g, -m_3 g) \end{aligned} \tag{3.17}$$

Substituting Equations (3.15)-(3.17) into Equations (3.6)-(3.9) gives

$$\begin{aligned} &(m_1+m_2+m_3)l_1\ddot{\theta}_1 + m_2 l_2\cos(\theta_1-\theta_2)\ddot{\theta}_2 + m_3 l_3\cos(\theta_1-\theta_3)\ddot{\theta}_3 + \\ &m_2 l_2\sin(\theta_1-\theta_2)\dot{\theta}_2^2 + m_3 l_3\sin(\theta_1-\theta_3)\dot{\theta}_3^2 = -(m_1+m_2+m_3)g\cos\theta_3 \\ &m_2 l_1\cos(\theta_2-\theta_1)\ddot{\theta}_1 + m_2 l_2\ddot{\theta}_2 + m_2 l_2\sin(\theta_2-\theta_1)\dot{\theta}_1^2 = -m_2 g\cos\theta_2 \\ &m_3 l_1\cos(\theta_3-\theta_1)\ddot{\theta}_1 + m_3 l_3\ddot{\theta}_3 + m_3 l_3\sin(\theta_3-\theta_1)\dot{\theta}_1^2 = -m_3 g\cos\theta_3 \end{aligned} \tag{3.18}$$

3.3.3 Handling of Constraints

The equations of motion for a system with an open-loop topology is a set of second order differential equations. If the topology has a closed loop, the path matrix will not be able to describe it. For each loop a cut is necessary and a constraint equation is formulated. It introduces a new unknown, an internal force, and results in a modified Lagrangian equations

$$\boldsymbol{AL}\{\ddot{\theta}_i\} + \boldsymbol{BL}\{\dot{\theta}_i^2\} = \{Q_i\} - \boldsymbol{G}^T\{\lambda_j\} \tag{3.19}$$

where λ is Lagrangian multipliers, $\boldsymbol{G}$ is usually called the *constraint Jacobian matrix* and its components are

$$G_{\alpha i} = \frac{\partial \phi_\alpha}{\partial \theta_i} \tag{3.20}$$

and ϕ_α is the αth constraint equation with variables θ_i ($i = 1, 2, \ldots N_c$), i.e.

$$\phi_\alpha(\theta_1, \theta_2, \ldots \theta_{N_c}) = 0 \qquad \alpha = 1, 2, \ldots N_c \tag{3.21}$$

The governing equations of motion are given by Equations (3.19) and (3.21), which are called *Differential Algebraic Equations* (DAEs).

3.4 Newton-Euler Equations

3.4.1 Newton-Euler Approach and Equation Formulation

The equations of motion can also be obtained in terms of the so-called Newton-Euler approach, which has been developed by J. Wittenburg. There is a detailed discussion in (Wittenburg, 1977).

In our model, all point-masses act upon each other by massless rods with only axial forces. These axial forces are represented as a set of components of the vector γ as the following

$$\{\gamma_i\} = (\gamma_1, \gamma_2, \gamma_3 \cdots \gamma_{N_r})^T \tag{3.22}$$

Let us consider a rod with an index v and with two point-masses m_i and m_j, shown as Figure (3.3).

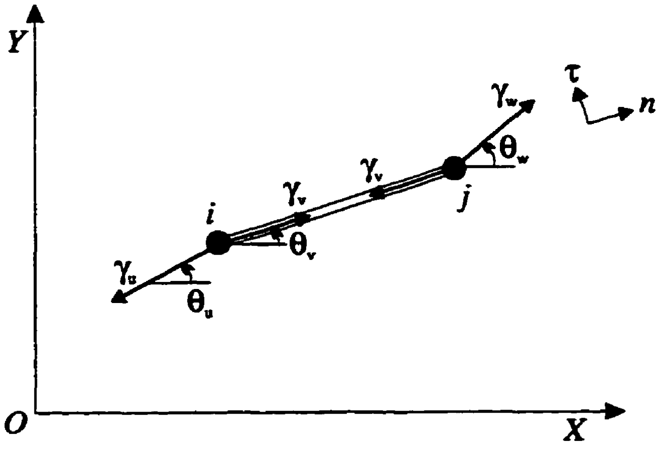


Figure 3.3 A single link with point-masses i and j

The magnitude of the force acting on the rod v is represented by γ_v. If this force is tensile then $\gamma > 0$; otherwise $\gamma < 0$. Applying Newtonian laws of motion in local generalized coordinates, we have the following

$$m_i a_{iv}^{n} = \gamma_u \cos(\theta_v - \theta_u) + \gamma_v + f_{iv}^{n} \tag{3.23}$$

$$m_i a_{iv}^{\tau} = \gamma_u \sin(\theta_v - \theta_u) + f_{iv}^{\tau} \tag{3.24}$$

and

$$m_j a_{jv}^{n} = -\gamma_v + \gamma_w \cos(\theta_v - \theta_w) + f_{jv}^{n} \tag{3.25}$$

$$m_j a_{jv}^{\tau} = -\gamma_w \sin(\theta_v - \theta_w) + f_{jv}^{\tau} \tag{3.26}$$

where

a_{iv}^{n}: acceleration of point-mass m_i in the n direction of the rod v;

a_{iv}^{τ}: acceleration of point-mass m_i in the τ direction of the rod v;

f_{iv}^{n}: component of external force on point-mass m_i in the n direction of the rod v;

f_{iv}^{τ}: component of external force on point-mass m_i in the τ direction of the rod v;

where n indicates the direction along the rod from i to j, and τ is perpendicular to it, as shown in the Figure (3.3).

In the inertial reference kinematic system, the accelerations of two point-masses should satisfy the following equations

$$m_j(a_{jv}^n - a_{iv}^n) = -m_j l_v \dot{\theta}_v^2 \tag{3.27}$$

$$m_j(a_{jv}^\tau - a_{iv}^\tau) = m_j l_v \ddot{\theta}_v \tag{3.28}$$

Eliminating a_{iv}^n, a_{iv}^τ, a_{jv}^n and a_{jv}^τ in Equations (3.23)~(3.28) gives

$$l_v \dot{\theta}_v^2 = -\frac{\gamma_u}{m_i}\cos(\theta_v - \theta_u) + \left(\frac{1}{m_i} + \frac{1}{m_j}\right)\gamma_v - \frac{\gamma_w}{m_j}\cos(\theta_v - \theta_w) + \left(\frac{f_{iv}^n}{m_i} - \frac{f_{jv}^n}{m_j}\right) \tag{3.29}$$

$$l_v \ddot{\theta}_v = -\frac{\gamma_u}{m_i}\sin(\theta_v - \theta_u) - \frac{\gamma_w}{m_j}\sin(\theta_v - \theta_w) - \left(\frac{f_{iv}^\tau}{m_i} - \frac{f_{jv}^\tau}{m_j}\right) \tag{3.30}$$

For each link in the system, we can obtain a pair of equations as above. If the number of links is N_e, then the total number of equations will be N_e. The first derivative and the second derivative of the generalized coordinate θ_i will be decoupled for every equation through the internal force parameter γ_i .

All of these equations in a component form, using the inductive method and the concept of the incidence matrix $\boldsymbol{E}$ can be written as follows

$$\boldsymbol{L}\{\dot{\theta}_i^2\} = \boldsymbol{H}\{\gamma_i\} + \{F_i^n\} \tag{3.31}$$

$$\boldsymbol{L}\{\ddot{\theta}_i\} = \boldsymbol{K}\{\gamma_i\} - \{F_i^\tau\} \tag{3.32}$$

$$\{F_i^n\} = \boldsymbol{C}\boldsymbol{E}^T\boldsymbol{M}^{-1}\{f_{xj}\} + \boldsymbol{S}\boldsymbol{E}^T\boldsymbol{M}^{-1}\{f_{yj}\} \tag{3.33}$$

$$\{F_i^\tau\} = -\boldsymbol{S}\boldsymbol{E}^T\boldsymbol{M}^{-1}\{f_{xj}\} + \boldsymbol{C}\boldsymbol{E}^T\boldsymbol{M}^{-1}\{f_{yj}\} \tag{3.34}$$

where it is assumed that the number of bodies is N_b (index j) and the number of links is N_e (index i).

Note: in Equations (3.33) and (3.34), f_{xi} and f_{yi} are respectively x and y components of the external forces. $\boldsymbol{E}$ is the incidence matrix of the order $N_b \times N_e$, $\boldsymbol{M}$ is a diagonal matrix of the order $N_b \times N_b$ and $\boldsymbol{L}$, $\boldsymbol{C}$ and $\boldsymbol{S}$ are all diagonal matrices of the order $N_e \times N_e$.

If we define

$$\boldsymbol{V} = \boldsymbol{E}^T \boldsymbol{M}^{-1} \boldsymbol{E} \tag{3.35}$$

then the components of matrix $\boldsymbol{H}$ and $\boldsymbol{K}$ will be

$$H_{ij} = V_{ij} \cos(\theta_i - \theta_j) \tag{3.36}$$

$$K_{ij} = V_{ij} \sin(\theta_i - \theta_j) \tag{3.37}$$

Since $\boldsymbol{V}$ is symmetric, it can be shown that $\boldsymbol{H}$ is also symmetric and $\boldsymbol{K}$ is anti-symmetric. In addition, using the trigonometric identities

$$\begin{aligned} \cos(\alpha - \beta) &= \cos\alpha \cos\beta + \sin\alpha \sin\beta \\ \sin(\alpha - \beta) &= \sin\alpha \cos\beta - \cos\alpha \sin\beta \end{aligned} \tag{3.38}$$

the matrices $\boldsymbol{H}$ and $\boldsymbol{K}$ can also be represented in another matrix form

$$\boldsymbol{H} = \boldsymbol{C}\boldsymbol{E}^T \boldsymbol{M}^{-1} \boldsymbol{E}\boldsymbol{C} + \boldsymbol{S}\boldsymbol{E}^T \boldsymbol{M}^{-1} \boldsymbol{E}\boldsymbol{S} \tag{3.39}$$

$$\boldsymbol{K} = \boldsymbol{S}\boldsymbol{E}^T \boldsymbol{M}^{-1} \boldsymbol{E}\boldsymbol{C} - \boldsymbol{C}\boldsymbol{E}^T \boldsymbol{M}^{-1} \boldsymbol{E}\boldsymbol{S} \tag{3.40}$$

The Equations (3.31) and (3.32) describe the system motion and they are in a decoupled form. If the system has a tree-like topology, then using the relationship the between incidence and path matrices $\boldsymbol{EP} = -\boldsymbol{I}$, [see Equation (2.7)], Equations (3.31) and

(3.32) can be converted into a coupled Lagrange form, which is exactly the form obtained by using a Lagrangian approach (Vinogradov, 1993a). The proof can be found in Appendix A.

Now, let us consider the example shown in Figure (3.2). The corresponding incidence matrix for the system topology is

$$E = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{3.41}$$

and the equations of motion are

$$\begin{aligned} l_1\dot{\theta}_1^2 &= \frac{\gamma_1}{m_1} - \frac{\gamma_2}{m_1}\cos(\theta_1 - \theta_2) - \frac{\gamma_3}{m_1}\cos(\theta_1 - \theta_3) + g\sin\theta_1 \\ l_2\dot{\theta}_2^2 &= -\frac{\gamma_1}{m_1}\cos(\theta_2 - \theta_1) + \frac{m_1 + m_2}{m_1 m_2}\gamma_2 - \frac{\gamma_3}{m_1}\cos(\theta_2 - \theta_3) \\ l_3\dot{\theta}_3^2 &= -\frac{\gamma_1}{m_1}\cos(\theta_3 - \theta_1) + \frac{\gamma_2}{m_1}\cos(\theta_3 - \theta_2) + \frac{m_1 + m_3}{m_1 m_3}\gamma_3 \end{aligned} \tag{3.42}$$

and

$$\begin{aligned} l_1\ddot{\theta}_1 &= -\frac{\gamma_2}{m_1}\sin(\theta_1 - \theta_2) - \frac{\gamma_3}{m_1}\sin(\theta_1 - \theta_3) - g\cos\theta_1 \\ l_2\ddot{\theta}_2 &= -\frac{\gamma_1}{m_1}\sin(\theta_2 - \theta_1) + \frac{\gamma_3}{m_1}\sin(\theta_2 - \theta_3) \\ l_3\ddot{\theta}_3 &= -\frac{\gamma_1}{m_1}\sin(\theta_3 - \theta_1) + \frac{\gamma_2}{m_1}\sin(\theta_3 - \theta_2) \end{aligned} \tag{3.43}$$

These equations are the Newton-Euler form of equations of motion, also simply called Newton-Euler equations or Newton-Euler formulations. Those equations from the Lagrange approach are called Lagrangian equations or Lagrangian formulations in this thesis.

3.4.2 Newton-Euler Equations in Rectangular Coordinates

For each component in matrices $\boldsymbol{C}$ and $\boldsymbol{S}$, we have

$$\begin{aligned} \cos\theta_v &= \frac{X_v}{l_v} = -\frac{x_i - x_j}{l_v} \\ \sin\theta_v &= \frac{Y_v}{l_v} = -\frac{y_i - y_j}{l_v} \end{aligned} \tag{3.44}$$

so the $\boldsymbol{C}$ and $\boldsymbol{S}$ matrices can be expressed as

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{L}^{-1}\,\mathrm{diag}(X_i) \\ \boldsymbol{S} &= \boldsymbol{L}^{-1}\,\mathrm{diag}(Y_i) \end{aligned} \tag{3.45}$$

or

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{L}^{-1}\mathrm{diag}(\boldsymbol{E}^T\{x_j\}) \\ \boldsymbol{S} &= \boldsymbol{L}^{-1}\mathrm{diag}(\boldsymbol{E}^T\{y_j\}) \end{aligned} \tag{3.46}$$

The left hand side of Equation (3.31) can also be expressed in the rectangular coordinates system

$$\boldsymbol{L}\{\dot\theta^2\} = \boldsymbol{L}^{-1}\boldsymbol{E}^T(\{\dot x_j^2\} + \{\dot y_j^2\}) \tag{3.47}$$

where $\boldsymbol{X}$, $\boldsymbol{Y}$ are relative coordinates and $\boldsymbol{x}$, $\boldsymbol{y}$ are absolute coordinates. Their relationship can be represented by

$$\begin{aligned}\{X_i\} &= -\boldsymbol{E}^T\{x_j\}\\ \{Y_i\} &= -\boldsymbol{E}^T\{y_j\}\end{aligned} \tag{3.48}$$

and so are the velocities

$$\begin{aligned}\{\dot{X}_i\} &= -\boldsymbol{E}^T\{\dot{x}_j\}\\ \{\dot{Y}_i\} &= -\boldsymbol{E}^T\{\dot{y}_j\}\end{aligned} \tag{3.49}$$

In order to obtain the reaction forces γ, the equations of motion of the system can be written directly in rectangular coordinates as follows

$$\boldsymbol{M}\{\ddot{x}_j\} = \boldsymbol{EC}\{\gamma_i\} + \{f_{xj}\} \tag{3.50}$$

$$\boldsymbol{M}\{\ddot{y}_j\} = \boldsymbol{ES}\{\gamma_i\} + \{f_{yj}\} \tag{3.51}$$

Correspondingly, the equation (3.31) in rectangular coordinates can be written as

$$\boldsymbol{H}\{\gamma_i\} = \boldsymbol{L}^{-1}\boldsymbol{E}^T(\{\dot{x}_j^2\} + \{\dot{y}_j^2\}) - \{F_i^n\} \tag{3.52}$$

It is necessary to point out that Equation (3.52) can be obtained directly, rather than from Equation (3.31). Let us consider the constraint relations

$$\{X_i^2\} + \{Y_i^2\} = \{L_i^2\} \tag{3.53}$$

after differentiating twice we have

$$\{X_i\ddot{X}_i\} + \{Y_i\ddot{Y}_i\} + \{\dot{X}_i^2\} + \{\dot{Y}_i^2\} = 0 \tag{3.54}$$

multiplying by $\boldsymbol{L}^{-1}$ gives

$$\boldsymbol{C}\{\ddot{X}_i\} + \boldsymbol{S}\{\ddot{Y}_i\} + \boldsymbol{L}^{-1}(\{\dot{X}_i^2\} + \{\dot{Y}_i^2\}) = 0 \tag{3.55}$$

Utilizing the available relations, i.e., Equations (3.46), (3.48)~(3.51), we can easily obtain Equation (3.52), which is a proof of validity of the constraint equations.

So far we have two different forms of Newton-Euler Equations. One in generalized coordinate angles and another in rectangular coordinates. They are, respectively, Equations (3.31)~(3.34) and Equations (3.50)~(3.52).

Introducing the internal reaction forces as parameters can cause the first and second order derivatives to be decoupled. Although the number of variables and equations increases for the Equations (3.50)~(3.52), the equations become simpler. Therefore, the amount of computing work is unchanged. One problem that needs to be pointed out is that if the number of variables in the equations of motion is in excess of the number of degree of freedom, violations of results can occur. How to handle this problem will be discussed in the later chapter.

3.4.3 Newton-Euler Equations with Constraints

The Newton-Euler equations are based on the incidence matrix $\boldsymbol{E}$. The incidence matrix can be used to describe either an open-loop topology or a closed-loop topology, so topology does not affect the form of the equations, (i.e., the differential equations are of the second order for an arbitrary topology).

It must be pointed out that if the system is over-constrained, then the matrix $\boldsymbol{H}$ will become singular, which means that the solutions of equations of motion will not exist.

The over-constrained system represents a cluster behaviour as a QRB (*Quasi-Rigid-Body*). If this is the case, then the system should be treated along the lines discussed in (Vinogradov and Springer, 1990) and (Wierzba and Vinogradov, 1991).

In the present thesis the concept of a QRB is not considered in numerical applications due to the limited objectives of the investigation. The complete governing equations, however, are given in the following for the sake of completeness.

3.5 Several Extended Forms of Newton-Euler Equations

3.5.1 Dynamics Equations of a Multi-Rigid-Body System

The point-mass system is not suitable for a rigid body system if the effect of body size cannot be neglected. It occurs when some particles form a cluster behaving as a QRB. Clusters such as this are over-constrained systems.

Now let us consider a system with three QRBs of the type described in (Vinogradov 1993b), shown in Figure (3.4).

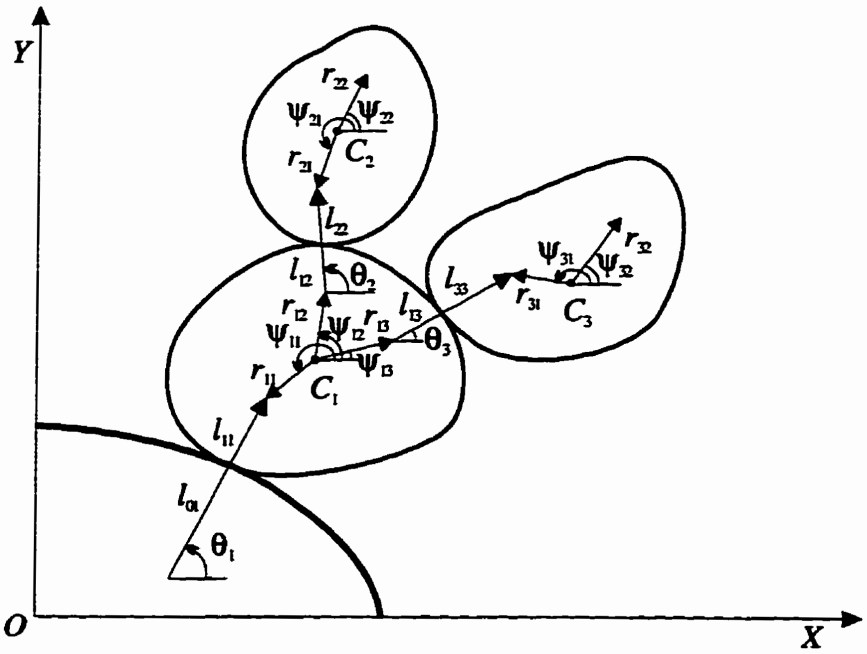


Figure 3.4 A multi-rigid-body model

Several matrices, L_r, C_ψ and S_ψ, are introduced to describe the system geometry and positions. They are

$$\boldsymbol{C}_r = [C_{rij}] = [E_{ij}\, r_{ik} \cos\psi_{ik}] \tag{3.56}$$

$$\boldsymbol{S}_r = [S_{rij}] = [E_{ij}\, r_{ik} \sin\psi_{ik}] \tag{3.57}$$

$$\boldsymbol{L}_r = [L_{ij}] = [\,|E_{ij}|\, l_{ij}] \tag{3.58}$$

where superscript k (1,2 ...) is from 1 to the maximum number of contact bodies. Those point to which the r_{ij} pointed are the centres of disks (in QRB) that are connecting with the external disks. The r and l in the equations above are scalars and they can be determined from the geometric configuration of the system. See (Vinogradov and Springer, 1990) and (Wierzba and Vinogradov, 1991).

In order to demonstrate clearly the meaning of the components of the matrices above, an example will be given based on the system shown in Figure (3.4). In this example, the new auxiliary matrices are

$$\begin{gathered}
\boldsymbol{C}_r = \begin{bmatrix} -r_{11}\cos\psi_{11} & r_{12}\cos\psi_{12} & r_{13}\cos\psi_{13} \\ 0 & -r_{21}\cos\psi_{21} & 0 \\ 0 & 0 & -r_{31}\cos\psi_{31} \end{bmatrix} \\
\boldsymbol{S}_r = \begin{bmatrix} -r_{11}\sin\psi_{11} & r_{12}\sin\psi_{12} & r_{13}\sin\psi_{13} \\ 0 & -r_{21}\sin\psi_{21} & 0 \\ 0 & 0 & -r_{31}\sin\psi_{31} \end{bmatrix} \\
\boldsymbol{L}_r = \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ 0 & l_{22} & 0 \\ 0 & 0 & l_{33} \end{bmatrix}
\end{gathered} \tag{3.59}$$

And the matrix $\boldsymbol{L}$ is correspondingly

$$\boldsymbol{L} = \begin{bmatrix} l_{01}+l_{11} & 0 & 0 \\ 0 & l_{12}+l_{22} & 0 \\ 0 & 0 & l_{13}+l_{33} \end{bmatrix} \tag{3.60}$$

Where the number of variables ψ_i is the same as the number of bodies (QRBs). Because there is only a constant difference between ψ_i and ψ_{ij}, the ψ_{ij} should be correspondingly upgraded as soon as the ψ_i is determined.

The Newton-Euler equations for a system of bodies with arbitrary shapes are as follows

$$\boldsymbol{L}\{\dot{\theta}_i^2\} = \boldsymbol{H}\{\gamma_i^n\} + \boldsymbol{K}\{\gamma_i^\tau\} - \boldsymbol{C}_\psi^T\{\dot{\psi}_j^2\} - \boldsymbol{S}_\psi^T\{\ddot{\psi}_j\} + \{F_i^n\} \tag{3.61}$$

$$\boldsymbol{L}\{\ddot{\theta}_i\} = \boldsymbol{K}\{\gamma_i^n\} - \boldsymbol{H}\{\gamma_i^\tau\} + \boldsymbol{S}_\psi^T\{\dot{\psi}_j^2\} - \boldsymbol{C}_\psi^T\{\ddot{\psi}_j\} - \{F_i^\tau\} \tag{3.62}$$

$$\boldsymbol{I}_\psi\{\ddot{\psi}_j\} = -\boldsymbol{S}_\psi\{\gamma_i^n\} + (\boldsymbol{L}_r + \boldsymbol{C}_\psi)\{\gamma_i^\tau\} + \{M_{\psi j}\} \tag{3.63}$$

The derivations of these equations can be found in Appendix B.

Note that Equations (3.31) and (3.32) are a particular case of the system represented by Equations (3.61)~(3.63), if in the latter ψ disappears.

In the case of absence of any slip between the bodies, the kinematic relations can be expressed in the form

$$\boldsymbol{L}\{\dot{\theta}_i\} = \boldsymbol{L}_r\{\dot{\psi}_j\} \tag{3.64}$$

In the Equations (3.61)~(3.64), the ψ is the self rotation angle of the body, $\boldsymbol{I}_\psi$ is the inertia matrix, $\boldsymbol{M}_\psi$ is the external moment vector acting on each body, and $\boldsymbol{C}_\psi$ and $\boldsymbol{S}_\psi$

are as follows

$$C_\psi = C_r C + S_r S \quad \text{or} \quad C_\psi^T = C C_r^T + S S_r^T \tag{3.65}$$

$$S_\psi = S_r C - C_r S \quad \text{or} \quad S_\psi^T = C S_r^T - S C_r^T \tag{3.66}$$

The reaction force γ is split into two parts. One is the normal reaction force along the l direction; another is the tangential force perpendicular to the l direction, where l_i is the link vector.

The final form of the equations of motion for a general system can be obtained from Equations (3.61)~(3.64), in which the second order derivatives of θ and ψ will be decoupled. Note:

(1) if slips occur, the kinematics relations, i.e. Equation (3.64), are invalid. However, the reaction forces in this case are known and therefore the system of equations is closed;

(2) the Equations (3.61)~(3.63) are not the decoupled form of equations of motion, but they can be obtained by substituting Equation (3.63) into Equations (3.61) and (3.62).

Moreover, the Lagrangian form of equations of motion for an open loop system can be found in (Vinogradov, 1993b).

3.5.2 Equations of Motion for a Multi-Rigid-Body System with Link Connectors

Assume an ideal interface between the bodies, and the reaction forces act only in the normal direction. In this case, the interface can be modelled by a link connector. See Figure (3.5).

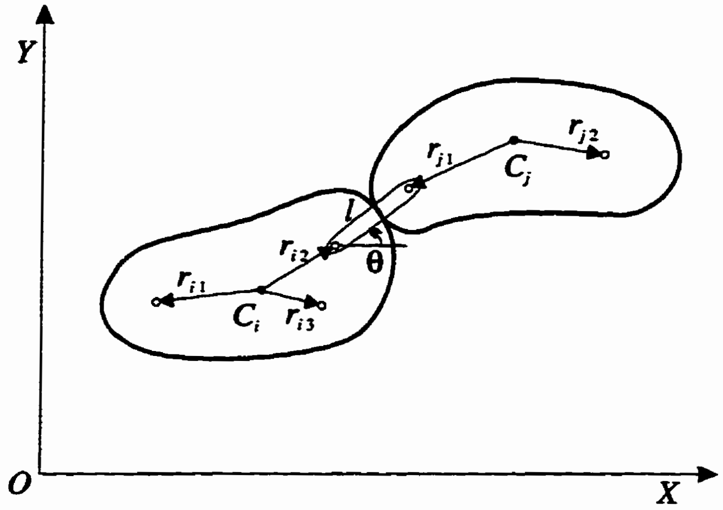


Figure 3.5 Two rigid bodies with a link connector

where C is the centre of mass and $\boldsymbol{r}_{ik}$ is the vector with directions from C_i to local link point. The normal contact force acts along the link. In actual situations, the link connectors can take only compressive forces.

In dynamic equations, i.e., Equations (3.61)~(3.64), let $\{\gamma_i^n\} = \{\gamma_i\}$ and $\{\gamma_i^\tau\} = \{0\}$, then these equations become

$$\boldsymbol{L}\{\dot{\theta}_i^2\} = \boldsymbol{H}\{\gamma_i\} - \boldsymbol{C}_\psi^T\{\dot{\psi}_j^2\} - \boldsymbol{S}_\psi^T\{\ddot{\psi}_j\} + \{F_i^n\} \tag{3.67}$$

$$\boldsymbol{L}\{\ddot{\theta}_i\} = \boldsymbol{K}\{\gamma_i\} + \boldsymbol{S}_\psi^T\{\dot{\psi}_j^2\} - \boldsymbol{C}_\psi^T\{\ddot{\psi}_j\} - \{F_i^\tau\} \tag{3.68}$$

$$\boldsymbol{I}_\psi\{\ddot{\psi}_j\} = -\boldsymbol{S}_\psi\{\gamma_i\} + \{M_{\psi j}\} \tag{3.69}$$

Utilizing Equation (3.69), we can rewrite Equations (3.67) and (3.68) as

$$\boldsymbol{L}\{\dot{\theta}_i^2\} = (\boldsymbol{H}+\boldsymbol{H}_\psi)\{\gamma_i\} + \{F_i^n\} - \boldsymbol{C}_\psi^T\{\dot{\psi}_j^2\} - \boldsymbol{S}_\psi^T\boldsymbol{I}_\psi^{-1}\{M_{\psi j}\} \tag{3.70}$$

$$\boldsymbol{L}\{\ddot{\theta}_i\} = (\boldsymbol{K}+\boldsymbol{K}_\psi)\{\gamma_i\} - \{F_i^\tau\} + \boldsymbol{S}_\psi^T\{\dot{\psi}_j^2\} - \boldsymbol{C}_\psi^T\boldsymbol{I}_\psi^{-1}\{M_{\psi j}\} \tag{3.71}$$

where

$$\boldsymbol{H}_\psi = \boldsymbol{S}_\psi^T\boldsymbol{I}_\psi^{-1}\boldsymbol{S}_\psi \tag{3.72}$$

$$\boldsymbol{K}_\psi = \boldsymbol{C}_\psi^T\boldsymbol{I}_\psi^{-1}\boldsymbol{C}_\psi \tag{3.73}$$

Equations (3.70), (3,71), and (3.69) are the equations of motion with the decoupled form on the second derivative.

3.5.3 Equations of Motion for a Multi-Rigid-Body System with Revolute Joints

Most multi-rigid-body systems have revolute joints. For example, in a manipulator the connections between the bodies are by revolute joints. See Figure (3.6).

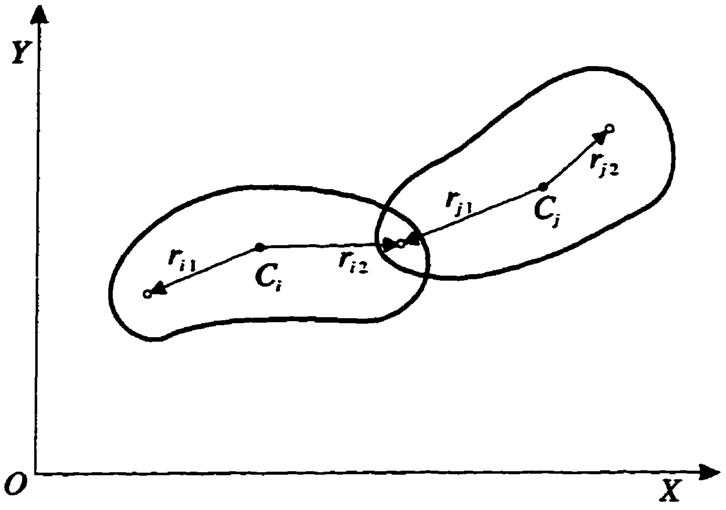


Figure 3.6 Two rigid bodies with a revolute joint

This kind of system can be considered as a special case of the system mentioned above. If the length of the link becomes zero in the above system, the link connector is transformed into a revolute joint. Therefore, the equations of motion can be directly derived from Equations (3.61), (3.62) and (3.63).

Clearly, when the length of each rod l becomes zero, the corresponding angle θ disappears. However, the reaction forces between the bodies still exist. Assuming the length of rod l is very close to zero, then the θ still has meaning. Let the reaction force γ be divided into two parts, γ_x and γ_y, which can be determined by

$$\{\gamma_{xi}\} = C\{\gamma_i\} \tag{3.74}$$

$$\{\gamma_{yi}\} = S\{\gamma_i\} \tag{3.75}$$

Let l be zero in the Equations (3.70) and (3.71), which gives

$$(V+H_{\psi})\{\gamma_{xi}\}-D_{1}\{\gamma_{yi}\} = E^{T}M^{-1}\{f_{xj}\}-S_{r}^{T}I_{\psi}^{-1}\{M_{\psi j}\}-C_{r}^{T}\{\dot{\psi}_{j}^{2}\} \tag{3.76}$$

$$-D_{2}\{\gamma_{xi}\}+(V+K_{\psi})\{\gamma_{yi}\} = E^{T}M^{-1}\{f_{yj}\}+C_{r}^{T}I_{\psi}^{-1}\{M_{\psi j}\}-S_{r}^{T}\{\dot{\psi}_{j}^{2}\} \tag{3.77}$$

where

$$D_{1} = S_{\psi}^{T}I_{\psi}^{-1}C_{\psi} \tag{3.78}$$

$$D_{2} = C_{\psi}^{T}I_{\psi}^{-1}S_{\psi} \tag{3.79}$$

Equations (3.76) and (3.77) can also be written in a matrix form

$$\begin{bmatrix} (V+H_{\psi}) & -D_{1} \\ -D_{2} & (V+K_{\psi}) \end{bmatrix}\begin{bmatrix} \{\gamma_{xi}\} \\ \{\gamma_{yi}\} \end{bmatrix} = \begin{bmatrix} -S_{r}^{T}I^{-1} & -C_{r}^{T} \\ C_{r}^{T}I^{-1} & -S_{r}^{T} \end{bmatrix}\begin{bmatrix} \{M_{\psi j}\} \\ \{\dot{\psi}_{j}^{2}\} \end{bmatrix} + \begin{bmatrix} \{B_{xi}\} \\ \{B_{yi}\} \end{bmatrix} \tag{3.80}$$

where

$$\begin{bmatrix} \{B_{xi}\} \\ \{B_{yi}\} \end{bmatrix} = \begin{bmatrix} E^{T}M^{-1} & 0 \\ 0 & E^{T}M^{-1} \end{bmatrix}\begin{bmatrix} \{f_{xj}\} \\ \{f_{yj}\} \end{bmatrix} \tag{3.81}$$

Using Equations (3.74) and (3.75), the Equation (3.69) can be written as

$$I_{\psi}\{\ddot{\psi}_{j}\} = -S_{\psi}\{\gamma_{xi}\}+C_{\psi}\{\gamma_{yi}\}+\{M_{\psi j}\} \tag{3.82}$$

Equations (3.80), (3.81) and (3.82) are the Newton-Euler Equations for a multi-rigid-body system with hinge joints. Note that these equations can also be directly derived from the dynamics equations, i.e., Equations (3.61)~(3.64), in terms of the following relations

$$\{\gamma_i^n\} = C\{\gamma_{xi}\} + S\{\gamma_{yi}\} \tag{3.83}$$

$$\{\gamma_i^\tau\} = -S\{\gamma_{xi}\} + C\{\gamma_{yi}\} \tag{3.84}$$

3.5.4 Equations of Motion in Spatial Coordinates

A granular system in 3-D is often treated as multisphere system. An example of a multisphere system is shown in Figure (3.7).

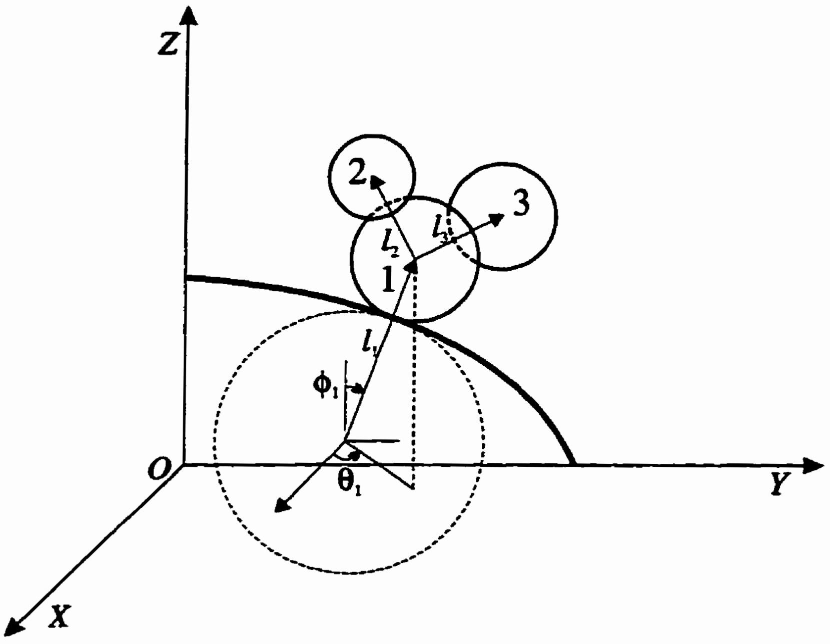


Figure 3.7 A simple multisphere system comprising 3 balls

In the paper by Vinogradov (1992), the equations of motion were derived based on Lagrangian equations and the path matrix is used to represent the topology of the system. Here, we will give the Newton-Euler Equations for a 3-D multisphere system.

The details of derivation are in Appendix C.

If θ and ϕ are the generalized coordinates, the equations of motion can be given in the form

$$L(\{\dot{\phi}_i^2\}+S_\phi^2\{\dot{\theta}_i^2\}) = H_\phi\{\gamma_i\}+\{F_i^n\} \tag{3.85}$$

$$L(\{\ddot{\phi}_i\}-S_\phi C_\phi\{\dot{\theta}_i^2\}) = K_\phi\{\gamma_i\}-\{F_i^\phi\} \tag{3.86}$$

$$LS_\phi\{\ddot{\theta}_i\}+2LC_\phi\{\dot{\phi}_i\dot{\theta}_i\} = KS_\phi\{\gamma_i\}-\{F_i^\theta\} \tag{3.87}$$

where

$$H_\phi = C_\phi V C_\phi + S_\phi H S_\phi \tag{3.88}$$

$$K_\phi = S_\phi V C_\phi - C_\phi H S_\phi \tag{3.89}$$

and

$$\{F_i^n\} = S_\phi C E^T M^{-1}\{f_{xj}\}+S_\phi S E^T M^{-1}\{f_{yj}\}+C_\phi E^T M^{-1}\{f_{zj}\} \tag{3.90}$$

$$\{F_i^\phi\} = C_\phi S E^T M^{-1}\{f_{xj}\}+C_\phi C E^T M^{-1}\{f_{yj}\}-S_\phi E^T M^{-1}\{f_{zj}\} \tag{3.91}$$

$$\{F_i^\theta\} = -S E^T M^{-1}\{f_{xj}\}+C E^T M^{-1}\{f_{yj}\} \tag{3.92}$$

Equations (3.86)~(3.88) can also be represented in matrix form as

$$\begin{bmatrix}\{F_i^n\}\\ \{F_i^\phi\}\\ \{F_i^\theta\}\end{bmatrix} = \begin{bmatrix}S_\phi C & S_\phi S & C_\phi\\ C_\phi S & C_\phi C & -S_\phi\\ -S & C & 0\end{bmatrix}\begin{bmatrix}E^T M^{-1} & 0 & 0\\ 0 & E^T M^{-1} & 0\\ 0 & 0 & E^T M^{-1}\end{bmatrix}\begin{bmatrix}\{f_{xj}\}\\ \{f_{yj}\}\\ \{f_{zj}\}\end{bmatrix} \tag{3.93}$$

Similarly, for 3-D rectangular coordinates, the Newton-Euler equations are

$$\boldsymbol{H}_{\phi}\{\gamma_i\} = \boldsymbol{L}^{-1}\boldsymbol{E}^{T}(\{\dot{x}_i^2\} + \{\dot{y}_i^2\} + \{\dot{z}_i^2\}) - \{F_i^n\} \tag{3.94}$$

$$\boldsymbol{M}\{\ddot{x}_i\} = \boldsymbol{E}\boldsymbol{S}_{\phi}\boldsymbol{C}\{\gamma_i\} + \{f_{xi}\} \tag{3.95}$$

$$\boldsymbol{M}\{\ddot{y}_i\} = \boldsymbol{E}\boldsymbol{S}_{\phi}\boldsymbol{S}\{\gamma_i\} + \{f_{yi}\} \tag{3.96}$$

$$\boldsymbol{M}\{\ddot{z}_i\} = \boldsymbol{E}\boldsymbol{C}_{\phi}\{\gamma_i\} + \{f_{zi}\} \tag{3.97}$$

In addition, Equation (3.93) can also be directly obtained through the constraint equations if a derived differentiated form are used. The specific procedure is similar as that mentioned in Section 3.4.2.

The equations in 2-D can be obtained in a particular case if we let z = Const. and $\phi = \pi/2$, (i.e., $\boldsymbol{S}_{\phi} = \boldsymbol{I}$ and $\boldsymbol{C}_{\phi} = \boldsymbol{0}$). In addition, when the system has a tree-like topology, the Lagrangian equation for a multisphere system can be obtained directly from Equations (3.85)~(3.88) by eliminating reaction forces γ. The component form of the equations is the same as in the Lagrangian approach (Vinogradov, 1992).

3.6 Summary

In this chapter the equations of motion for a granular system were given for two formulations: Lagrangian and Newton-Euler. As well, the Newton-Euler equations were discussed in detail with various forms of the equations discussed.

For purpose of comparison, the Lagrangian equations used for multibody systems are given here in a similar matrix form. An obvious distinction is that the reaction forces are presented in the Newton-Euler equations explicitly.

In computer simulations, the reaction forces must be known during the integration so that the variability of the system topology can be monitored. That is why we prefer to use the Newton-Euler form of equations of motion. In addition, they decouple second derivatives and thus, as it will be shown, make simulations more efficient.

In summary, the Lagrangian formulation gives the minimum number of equations. This form of equations can easily be obtained through mapping on a data structure. The characteristic feature of the Lagrangian formalism that negatively affects its numerical efficiency is that the second derivatives in the differential equations are coupled. On the other hand, the Newton-Euler formalism leads to a larger number of equations because the internal forces are additional unknowns. However, the second derivatives in this case are uncoupled. Thus it is not straightforward which formulation is more computationally advantageous.

Chapter 4

SOME ASPECTS OF METHODOLOGY OF COMPUTER SIMULATION

4.1 Introduction

In a computer simulation program for granular systems, we must meet two objectives. One is to use a mathematical model as close to the actual system as possible. The other objective is to make it computationally efficient. In this chapter, we will address both of these objectives.

Since a granular dynamics system is a system with variable topology and since equations of motion depend on the topological structure of the system, the storage and updating of the topological data will affect the computational efficiency directly. Therefore, a new handling of topology will be introduced in Section 4.2.

The handling of driving forces and their effective description is very important in computer simulations and this will be discussed in Section 4.3. These forces are gravitational forces, drag forces, friction forces, and impact forces.

In a dense granular system, collisions occur very frequently. The handling of these collision events takes significant CPU time. Hence, in Section 4.4, we will introduce a new method to handle collision events so as to improve the efficiency of computer simulation.

The generation of disks with random size, position and time interval must yield some specific distribution density. The method, presented in (Wierzba, 1991), is called sampling and it has two disadvantages. One is its complexity. Another is its restriction of particles density. In Section 4.5, a simple generation technique will be introduced which can give a high particles density distribution of granular particles in a known generation area.

4.2 Topological Data Updating

Granular dynamical systems have a topological structure which is variable in time. Usually matrices are used to describe the system topology. Consequently, the equations of motion are also expressed in a matrix form. Certainly, a matrix expression is simple and clear. However, matrix calculations require additional computer time due formation of matrices and operation on them.

Matrices describing system topology are usually sparse. Obviously, it is not economical to store the zero components and to use them in computations. The way to improve the computational efficiency is to apply a one- dimensional array to replace a diagonal matrix and to use a multi-dimensional array to replace a block-diagonal matrix in the generation of equations of motion and topological data updating.

In the incidence matrix $\boldsymbol{E}$, each column has no more than two elements. They are used to represent the beginning and the end of a link. When a body is interfaced with an obstruction or a boundary, the corresponding column has only one element. So an array with two rows can be used to represent an incidence matrix completely. For instance, if the incidence matrix $\boldsymbol{E}$ is of order $N_b \times N_e$, the corresponding array $\boldsymbol{A_E}$ is of order $2 \times N_e$. Let us consider the example in Section 3.3.2. The incidence matrix in this case is

$$\boldsymbol{E} = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4.1}$$

and the corresponding array is

$$\boldsymbol{A}_{\boldsymbol{E}} = \begin{bmatrix} -1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \tag{4.2}$$

In the array, each column denotes a link. The first row represents the numbers of bodies located at the beginning of the link and the second row represents the numbers of bodies located at the end of the link. The negative number denotes the obstructions or boundaries.

An array representation, instead of a matrix one, can reduce not only computer storage space but also computational time. However, this expression can be inefficient if a path matrix $\boldsymbol{P}$ is used, because the latter is not a banded matrix in general. That is one of the reasons why we use the incidence matrix and the Newton-Euler equations in computer simulations.

The equations of motion can also be stored in an array so as to avoid unnecessary multiplication of zero components. In the array, the number of columns is equal to that of $\boldsymbol{A}_{\boldsymbol{E}}$, while the number of rows depends on the maximum number of neighbouring bodies. In other words, the number of rows can be determined by the number of bodies interfacing a given body.

In a planar multidisk system, if all of the disks are of the same size or close to the same size, this number should be 6. In a spatial multisphere system, if all of the spheres are of the same size or close to the same size, the number should be 12. If the sizes of bodies in a system vary, the maximum number of rows can be obtained in terms of the maximum and minimum sizes of bodies in the system. Let N_m be the maximum possible number of contacts in the planar multidisk system, then it can be found from

$$N_m = \mathrm{INT}\left[\frac{\pi}{\alpha}\right] \tag{4.3}$$

where INT [ ] is the conversion integer function and α is defined by

$$\sin\alpha = \frac{r_{\min}}{r_{\min} + r_{\max}} \tag{4.4}$$

In Equation (4.4) $r_{\min}$ and $r_{\max}$ are the minimum and maximum radii of disks in the system, respectively. If $r_{\min} = r_{\max}$, then $\alpha = \pi/6$ and we have that $N_m = 6$. The proof of Equation (4.3) can easily be obtained from Figure (4.1).

$$R = OO' = r_{\min} + r_{\max}$$

$$\overset{\frown}{A'O'B'} = 2\alpha R$$

$$N_m = \frac{2\pi R}{2\alpha R} = \frac{\pi}{\alpha}$$

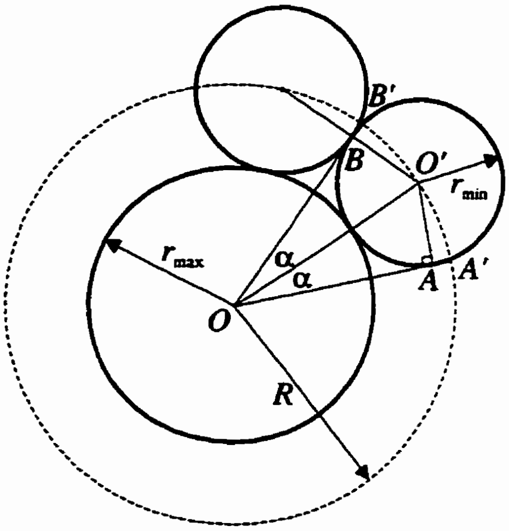


Figure 4.1 Illustration of the maximum coordination number in a multidisk system

In a spatial multisphere system N_m can be determined approximately by

$$N_m = \mathrm{INT}\left[\frac{2(\pi + E)}{E}\right] \tag{4.5}$$

where E is the spherical excess of the triangle i.e. $E = A + B + C - \pi$, where A, B, and C are the angles of the triangle on a spherical surface. The definition of the spherical excess of the triangle can also be found in the spherical geometry section of any mathematics handbook, such as (Zwillinger, 1996). In our problem, E can be determined from

$$\tan\frac{1}{4}E = \sqrt{\tan\frac{3\alpha}{2}\tan^3\frac{\alpha}{2}} \tag{4.6}$$

where α can still be obtained from Equation (4.4). Only $r_{\min}$ and $r_{\max}$ are respectively the minimum and maximum radii of spheres in the system.

To prove Equation (4.5) requires reference to spherical geometry and can be done in a similar manner to the proof of Equation (4.3). Please refer to Appendix D.

Note that E is measured in radians in Equation (4.5), and also because the interstices between the disks are irregular, the obtained N_m may be greater than the actual value.

Let us assume that $r_{\min} = r_{\max}$, then $\alpha = \pi/6$ and in this case $N_m = 13$. In fact, the maximum number for this situation is 12. Although there is still space left when 12 similar size spheres are used, it is not possible to have any more spheres. In any case, using the maximum row number which is determined by Equation (4.5) provides a degree of safety.

Hence, the size of the array describing the topology is variable. It depends on the dimensions of the system and on the maximum and minimum sizes of the bodies in the system. Obviously, the use of an array can greatly save the storage space and avoid unnecessary operations, thereby greatly improving the efficiency of computer simulation.

4.3 Variable External Forces

In a granular system there are acting external forces and induced internal forces. The external forces, in general, are gravitational forces when a system is in a gravity field and drag forces when a system is in a fluid field. Gravitational forces are potential forces and are time independent. The friction and impact forces are internal forces. These forces are non-conservative and time dependent.

4.3.1 Gravitational and Drag Forces

Every body in a granular system will be acted upon by the gravitational force as long as the system is located in a gravitational field. The gravitational force acting on the *i*th body is given by

$$\boldsymbol{F}_i = m_i \boldsymbol{g} \tag{4.7}$$

where m_i is the mass of the *i*th body and $\boldsymbol{g}$ is the gravitational acceleration.

The gravitational force must be considered if it is the dominant force for the motion of a multibody system. When a multibody system is carried by a fluid, the drag force must be determined since it is a driving force. In general, the latter is proportional to the square of the relative velocity of the body with respect to the fluid. If $\boldsymbol{u}_B$ denotes the velocity vector of the body mass centre and $\boldsymbol{u}_F$ the velocity vector of the fluid at that point, then the drag force is

$$\boldsymbol{F}_D = -\frac{1}{2}\rho\, A_p C_D C_A (\boldsymbol{u}_B - \boldsymbol{u}_F)\,|\boldsymbol{u}_B - \boldsymbol{u}_F| \tag{4.8}$$

where ρ is the fluid density, A_p is the projected area of the body and C_D is the non-dimensional drag coefficient. It is necessary to point out that C_D is not a constant but a function of the Reynolds number *Re*. *Re* is defined by

$$Re = \frac{u_0 d}{\nu} \tag{4.9}$$

where u_0 is the velocity of flow, ν is the kinematic viscosity of fluid and d is the size of the body. Actually, C_D depends on the shape of the body, roughness of the solid surface and the viscosity of the fluid, and is found experimentally. C_A in Figure (4.8) is another parameter which is called here the attenuation coefficient. In (Wierzba, 1991), the drag forces acting on each disk are not affected by the neighbouring disks. However, their influence causes errors, especially in the system with a dense distribution of particles. Here we will make corrections to eliminate this kind of error.

Before introducing the attenuation coefficient, C_A, of drag force, let us introduce the shadow factor k which is defined as the ratio

$$k_{ij} = \frac{h_{ij}}{2r_j} \tag{4.10}$$

where h_{ij} is the shadowed part of disk j caused by disk i. For a single disk or the disk not in a shadow, $k = 1$. For a disk located in the flow shadow of neighbouring disk(s), which can be illustrated in Figure (4.2), $k < 1$. From Figure (4.2) we can find

$$h_{ij} = (r_i + r_j)(1 - \sin\theta_{ij}) \tag{4.11}$$

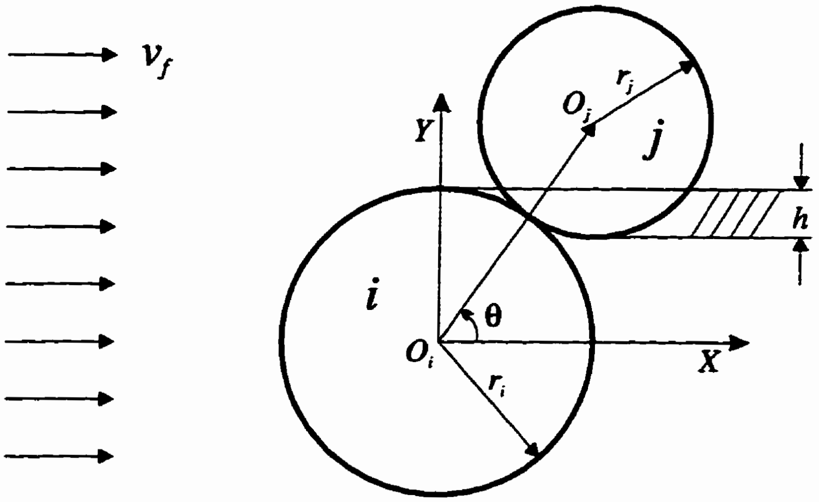


Figure 4.2 Illustration of the concept of reduction of drag force for two disks in a flow

If more than one disk is in contact with disk *i* then

$$k_j = \frac{1}{2r_j}\sum_k h_{jk} \tag{4.12}$$

and the coefficient C_A for the disk *j* is defined by

$$C_{Aj} = \begin{cases} 1 - k_j & k_j < 1 \\ 0 & k_j \geq 1 \end{cases} \tag{4.13}$$

Note that C_A is always greater than zero for the existence of shadow. If $k_j \geq 1$, which means that the disk *j* is fully shadowed, C_A is equal to zero, as in Equation (4.13).

In addition, k should be determined in each direction. Here C_A is given only in the x direction. The component of C_A for y direction can easy be obtained in a similar way.

4.3.2 Friction Forces

A friction force between two rigid bodies can be produced only if they have a relative motion in the tangential direction. Because of the assumption that the rotation of each body is not considered, the relative velocity between the two bodies can be substituted by the relative velocity of the mass centre of the two bodies. See Figure (4.3).

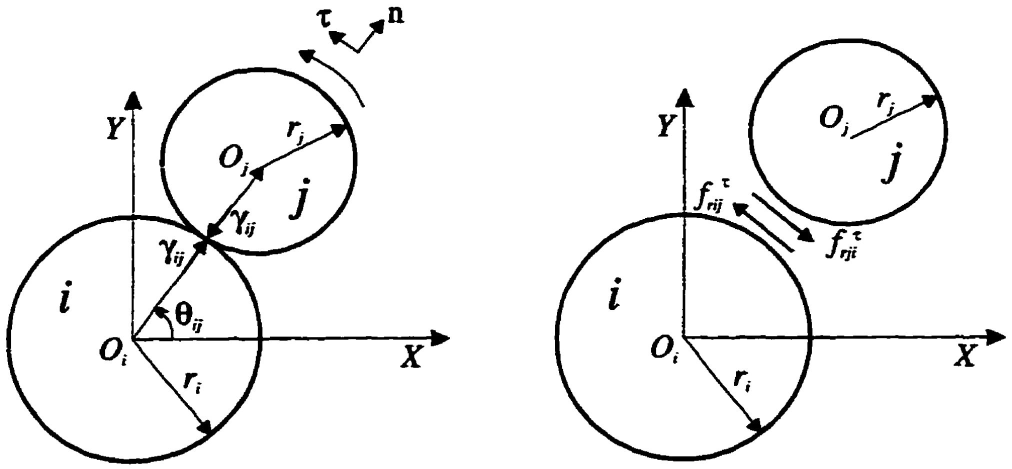


Figure 4.3 Friction force between two disks

Generally, the friction force is proportional to the normal force γ_{ij} between the two bodies and its direction is always perpendicular to the latter. Its specific direction can be determined by the relative velocity between the two disks. The friction force between disks i and j can be given in the form

$$f^{\tau}_{rij} = \mathrm{SGN}(v_i^{\tau} - v_j^{\tau})\mu\gamma_{ij} \qquad \text{for} \quad \gamma_{ij} < 0 \tag{4.14}$$

On disk *i* the friction forces in *X* and *Y* direction are respectively

$$\begin{aligned} f_{rxi} &= \sum_j \mathrm{SGN}(\dot{x}_i - \dot{x}_j)\mu\gamma_{ij}\sin\theta_{ij} \\ f_{ryi} &= \sum_j \mathrm{SGN}(\dot{y}_i - \dot{y}_j)\mu\gamma_{ij}\cos\theta_{ij} \end{aligned} \tag{4.15}$$

where μ is the coefficient of friction and SGN ( ) used here is a special sign function that is defined as

$$\mathrm{SGN}(x) = \begin{cases} 1 & x > \varepsilon \\ \dfrac{x}{\varepsilon} & |x| \le \varepsilon \\ -1 & x < -\varepsilon \end{cases} \tag{4.16}$$

where ε is the minimum allowance value, or the 1/ε is the slope. See the solid line in Figure (4.4).

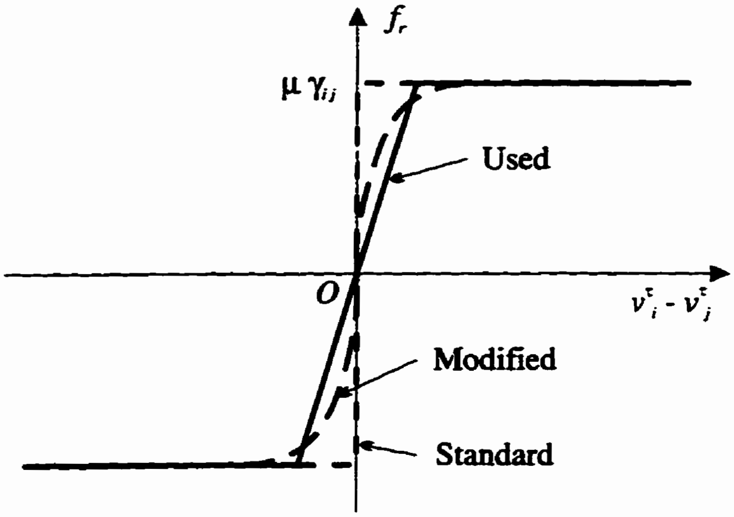


Figure 4.4 Several different friction models

In the Figure (4.4), the dotted line represents the ideal dry friction model, mathematically defined as

$$\mathrm{SGN}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \tag{4.17}$$

In order to avoid the discontinuity at zero, some authors, such as Threlfall (1978) and Rooney and Deravi (1982), used a modified model. See the broken line in Figure (4.4).

The friction force is a passive force which is proportional to the reaction force γ_{ij} according to Equation (4.14). If the reaction force γ_{ij} on a disk is very large, then the friction force is also large and can be larger than the resultant of external forces acting on the disk. The direction of velocity of this disk may be changed during a time step, which results in a friction force with an opposite direction in the next time step. And so forth, it is just like a vibration. In order to avoid this case, a restriction to the friction force acting on each disk should given in the form

$$\begin{aligned} f_{mrxi} &= \frac{m\dot{x}_t}{h} + \sum f_{xi} \\ f_{mryi} &= \frac{m\dot{y}_t}{h} + \sum f_{yi} \end{aligned} \tag{4.18}$$

where Σf_{xi} and Σf_{yi} are the sum of total forces, except the friction force, acting on the disk i in x and y direction respectively, and h is the time step.

If the friction force is less than this limitation value, it will obey Equation (4.15). Otherwise, its quantity will be determined by Equation (4.18). The relationship between the magnitudes of a friction force and reaction force can be seen in Figure (4.5). Note that f_m is not a constant. It has a different values for different disks and at different times.

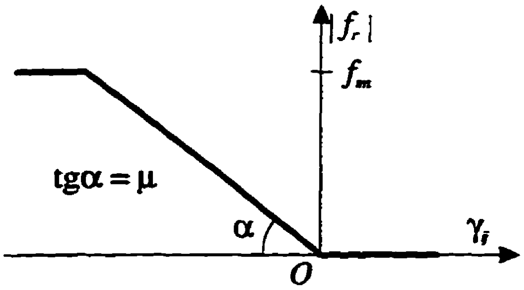


Figure 4.5 Relationship between friction force and reaction force

The limit on friction forces can result in a stable process when a granular system is close to a packed state. If this is not the case, this limitation can be omitted in simulations of a granular system.

Since the friction force is a function of the reaction force, its computation is a part of the iterative cycle in the simulation program, which means that the friction forces are updated in every iteration cycle. The detailed iterative procedure will be discussed in the next chapter.

4.4 Impact

In a moving multibody granular system collisions between the bodies and between the bodies and boundaries occur frequently. During a collision, there is always a pair of equivalent impact forces which act respectively on the two bodies that participate in a collision. A collision is an instantaneous event compared to the time-scale of motion. However, impact forces in a granular system can make changes in the state of motion and the topological structure of the system. It is these changes that we are concerned with. Our focus is to handle the collision events correctly rather than how to find those impact forces.

Let us consider a simple collision between two single disks and assume that the velocities before and after the collision are identified by indexes 1 and 2. The velocities of disks i and j which participate in a collision should satisfy Newton's collision rule, which is

$$v_{i2}^{n} - v_{j2}^{n} = -e(v_{i1}^{n} - v_{j1}^{n}) \tag{4.19}$$

where $0 < e < 1$ is called the kinetic coefficient of restitution. The superscript n denotes the normal direction of velocities and subscripts i, j and 1, 2 denote, respectively, the disk's numbers and the time index before and after the collision.

If the collision occurs between two sub-systems, according to (Brach, 1991), a set of collision equations needs to be set up in the form

where

T the kinetic energy of the system;

$$\left(\frac{\partial T}{\partial \dot{q}_i}\right)_2 - \left(\frac{\partial T}{\partial \dot{q}_i}\right)_1 = P_i \qquad i=1,2,...N \tag{4.20}$$

q_i the velocities associated with the generalized coordinates;

P_i the generalized forces;

N the number of independent coordinates.

Obviously, at least N equations must be formulated and solved, which consumes CPU time and data storage space. Actually, the handling of collision events during the computer simulation takes even more time than solving the equations of motion of the system. In order to improve the efficiency of the computer simulation, a completely new method of handling collision events is used in this study.

The handling of collision events includes three parts: (1) identifying the events; (2) updating the system topology and (3) correcting the velocities. The system topology is checked once at the end of the time step. All new overlaps (not including the position errors for each link) will be identified. If the velocities of disks with overlap are larger than the allowance for velocity violation, then the overlaps are handled as collisions. Otherwise, they are corrected as a violation of the position constraint.

In the computer simulation program, the handling of collision events is done together with the velocity error correction once for each simulation time step. The velocity for each body is found independently. However, they have to obey the velocity constraint conditions for two bodies in contact, namely

$$v_i^n - v_j^n = 0 \tag{4.21}$$

If this condition is not satisfied, i.e., the right side of Equation (4.21) is not equal to zero, corrections have to be done. The method of error correction is to introduce a virtual reaction force between disks *i* and *j* that brings the velocity error to zero.

When velocity errors exist, the Equation (4.21) is

$$v_i^n - v_j^n = \eta_{ij} \tag{4.22}$$

We can set a limiting parameter η_m such that

$$|\eta_{ij}| \leq \eta_m \tag{4.23}$$

where η_m is the maximum allowable velocity violation error which is used as a threshold determining the occurrence of collision. Usually the absolute value of velocity errors is much less than η_m.

If a collision occurs between disks *i* and *j*, η_{ij} must be positive as an indication that two particles approach each other before the collision. If $\eta_{ij} < \eta_m$, then it is handled as a velocity violation error, i.e., it is regarded as a new link just formed. If $\eta_{ij} > \eta_m$, then it is handled as a general collision event and for a given coefficient of restitution *e* we have

$$\eta'_{ij} = (1+e)\eta_{ij} \tag{4.24}$$

After correcting velocity according to η'_{ij}, the disks *i* and *j* will have a relative velocity $-e\eta_{ij}$ that is it will obey Newton's collision law. From Equation (4.24) it can also deduced that if $e = 0$, i.e., a perfectly plastic impact, then a new link is formed and its velocity violation error will be corrected.

When most of the bodies in the system are interconnected, the correction of velocity violation errors and velocity corrections due to collisions in the system can be implemented by solving a set of linear equations. In a planar multidisk system the linear equations are as follows

$$\boldsymbol{H}\{\delta v_i\} = \{\eta_i\} \tag{4.25}$$

and the corresponding corrections of the velocities in the x and y directions are

$$\begin{aligned} \{\delta \dot{x}_j\} &= \boldsymbol{M}^{-1}\boldsymbol{E}\,\boldsymbol{C}\{\delta v_i\} \\ \{\delta \dot{y}_j\} &= \boldsymbol{M}^{-1}\boldsymbol{E}\,\boldsymbol{S}\{\delta v_i\} \end{aligned} \tag{4.26}$$

The detailed derivation of Equations (4.25) and (4.26) will be given in the next chapter.

4.5 Random Generation of Initial Conditions

It is difficult and unnecessary to give artificially the specific sizes and positions of all particles as initial conditions for a granular system. The generation of initial conditions, also called the sampling technique, has been mentioned in (Wierzba, 1991) However, due to the complexity of the sampling technique and the difficulties of high density generation one can not use directly this sampling method in a granular system.

In the model by Wierzba (1991), the distribution errors of the generation of disks in either size or position, have to satisfy the error function with a Gaussian distribution. Thus the generation problem becomes more complicated.

The generation have been simplified here. All random distributions are arbitrary according to the random generator which can give unlimited random numbers between 0 and 1. The generated physical quantities x are in terms of

$$x(i) = x_{\min} + (x_{\max} - x_{\min})\mathrm{Ran}(i) \tag{4.27}$$

provided the minimum and maximum of x are known. Here x could be radii, positions or velocities of the particles and Ran(i) is a random number generation function.

High density generation with random arrangement is one of the new sampling techniques that has been used in the computer simulation. The random sizes and positions generation is simple and easy if the densities are low. When the density becomes high, the generation method used in (Wierzba, 1991) will take much more computational time.

The new method here adopts a planar generation area, instead of line generation for a planar system. If the required generation density is not very high, the distribution of positions for each particle is random. If overlaps happen, they will be handled as position errors and corrected in the next time step. Obviously the higher the generation density is, the more regular is their arrangement. When the required generation density tends to its limit, each particle has a fixed reference base point. If all of the particles are the same size, these base points will be arranged regularly and their interval is according to the size of particles. See the cross signs in the following figure.

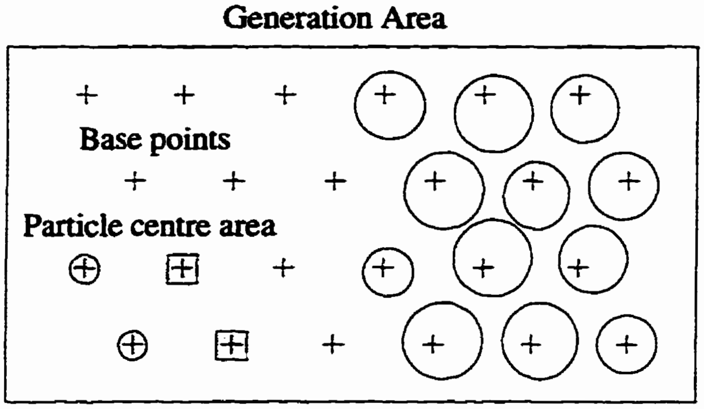


Figure 4.6 High density particle generation

In Figure (4.6) the cross signs indicate the base points, while the particle centres are located in the shadow area based on those base points and a function of a random number. If x and y are random, the particle centre will be in a rectangular area; if r and θ are random then it will be in a circle. The size of the particle can also be determined by a random generation function. The results of the generation are shown in the right hand side of Figure (4.6).

4.6 Summary

The contents in this chapter briefly deal with some technical problems in simulations, (except those with respect to the equations of motion and numerical methods) and provide corresponding solutions. All of these have been implemented in the computer simulation program and have been verified using specific examples.

The method of storing and updating topological data is briefly addressed in Section 4.2. In order to decrease storage space, two formulas for estimating the minimum requirement for row number for an array is given for both the 2-D and 3-D cases.

In this chapter we also discussed how to identify effectively the drag forces and the friction forces on disks. The introduction of the coefficient C_A allows us to make the computational model more realistic, which is very important for a granular system located in a flow field.

The method of handling impacts among the disks is presented in Section 4.4. This method allows us to handle all impacts only once in a time step, thus the required computational time is decreased.

The random generation of disks according to size, position, and time can be done in a different way. The method proposed here is simple and suitable for the generation of a high density disk distribution.

Among the methods and techniques discussed in this chapter, two are important for the efficiency of simulations: (1) the use of arrays; and (2) handling of collision events. The former can save computer storage space and while the latter makes collision problems simple and can save computational time. It is due to the use of these new methods and techniques in granular system simulations that the presented approach is not only original but also effective.

Chapter 5

NUMERICAL METHODS

5.1 Introduction

The equations of motion for a multibody system, whether based on Newton-Euler or Lagrangian Equations, are a set of second order nonlinear *Ordinary Differential Equations* (ODEs). The problem is an initial value problem for the ODEs, for which there is no alternative to a numerical solution. The accuracy of the solution and convergence of equations will directly influence the entire simulation result. Besides, the computational time of the simulation of a granular system should be taken into account because the computer's CPU time will be greatly increased if the number of equations is large.

In order to simulate a granular system efficiently, an iterative method based on the Newton-Euler equations is developed. The main iteration scheme adopts classical Gauss-Seidel iteration. In addition, numerical integral and error correction are performed in each iterative cycle.

Since this type of numerical method is for our specific equations and has not been used before, it will be discussed in some detail. The specific iterative scheme is given in the flowchart. The correction equations for position, velocity and energy are also given. Numerical tests can be found in the next chapter.

5.2. Selection and Comparison of Numerical Methods

5.2.1 Single-step and Multistep Method

First of all, we tested various numerical methods to solve a set of ODEs. The model used in the test is a multibody pendulum which is shown in Figure (5.1). The numerical methods such as Runge-Kutta, Adams', Gear's, and Bulirsch-Stoer extrapolation method were tested. All of these methods have library subroutines (IMSL) available. The results show that the Adams' method is the fastest and the Runge-Kutta method is the most accurate. And these two methods are compared in Table (5.1), in which errors are based on the violation of the energy conservation principle. It can be seen by Table (5.1) that the differences in efficiency are not significant between these methods.

Since the library routines include a variable time step option, the programs for the single step (Runge-Kutta) method and multistep (Adams') method with a fixed time step were also used for test purpose and results showed very little difference. It can thus be concluded that both the single step and the multistep method have almost the same efficiency in the solution of ODEs.

When considering a system with a variable topology, a single step integration method should be chosen so as to keep the same accuracy and computational efficiency for each time step. If the system has a fixed topology, use of the multistep method can save computational time. Comparison of the two methods can be seen in Table (5.1).

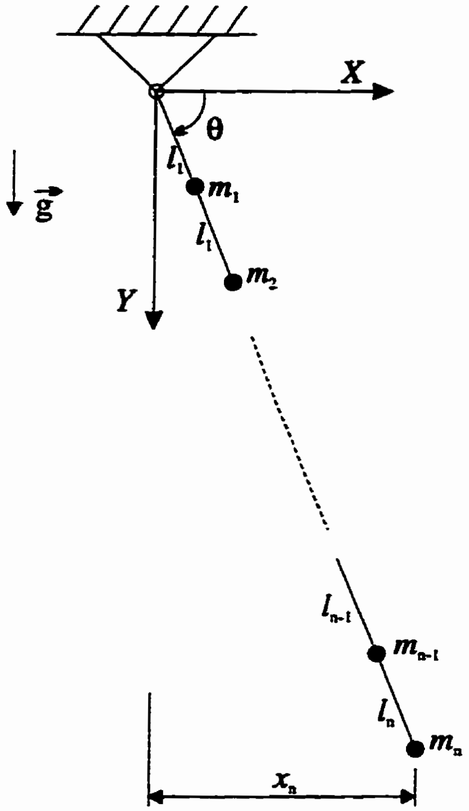


Figure 5.1 A multibody pendulum with *N* bodies

5.2.2 Lagrange Equations and Newton-Euler Equations

The second portion of our test is to see if there are different efficiencies for the different equations of motion. The results of Lagrange equations and Newton-Euler equations can be found in Table (5.1).

Table 5.1 Comparison of CPU times and errors for different dynamic formulations and numerical methods

	Lagrange Equations				Newton-Euler Equations			
Number of Bodies	Adams' Method		R-K Method		Adams' Method		R-K Method	
	Times (sec)	Relative Errors	Times (sec)	Relative Errors	Times (sec)	Relative Errors	Times (sec)	Relative Errors
N=100	613	1.7×10^{-4}	1,010	4.4×10^{-6}	475	2.2×10^{-4}	761	2.2×10^{-6}
N=200	4,703	1.4×10^{-4}	10,388	5.8×10^{-5}	3,832	1.3×10^{-3}	8,322	1.7×10^{-5}
N=400	65,878	2.5×10^{-4}	138,892	4.3×10^{-5}	54,759	8.0×10^{-4}	117,604	1.6×10^{-5}

Newton-Euler equations can give us shorter CPU times than Lagrangian equations. However, the amount saved will be decreased from 25% at N=100 to 15% at N=400. This result is still not ideal.

Regardless of which type of equations of motion are employed, the procedure of solution can always be divided into two steps: the first is to solve a set of linear algebraic equations so as to get the equations with the explicit second order derivatives; the second is to solve the ODEs. It should be noted that when the classical fourth-order Runge-Kutta method is used, the solution of linear equations has to be computed at least 4 times in a single time step. On the other hand, the multistep method (Adams' method) only requires to calculate once in a single time step. It can also utilize the results which have been obtained in the preceding calculations to improve the results. Hence this method can obviously reduce calculation time. However, the multistep method requires more storage space and is not suitable for a system with variable topology. It is necessary for us to search for a more efficient numerical method.

5.2.3 Iterative Methods

Under some specific conditions an iterative method can give solutions to linear equations effectively. The advantage in computational speed will be very significant especially for a large number of equations and a coefficient matrix with a large number of zero elements. It has been found through many numerical tests that only the Newton-Euler equations can be efficiently solved by an iterative method. The reason is very simple: the matrices $\boldsymbol{A}$ in Equation (3.6) and $\boldsymbol{H}$ in Equation (3.31) have different features. $\boldsymbol{A}$ is almost a full matrix and $\boldsymbol{H}$ is a sparse matrix.

5.3 Mixed Iteration

5.3.1 Gauss-Seidel Iteration

Computational efficiency is greatly improved by using the iterative method for solving a large set of linear equations. Here the Gauss-Seidel method is adopted as the basic iterative process for accelerating convergence. If we have a set of linear equations

$$\boldsymbol{H}\{\gamma_i\} = \{b_i\} \tag{5.1}$$

then its Gauss-Seidel iterative form is

$$\gamma_i^k = \frac{1}{H_{ii}}(b_i - \sum_{j=1}^{i-1} H_{ij}\gamma_j^k - \sum_{j=i+1}^{n} H_{ij}\gamma_j^{k-1}) \tag{5.2}$$

where superscript k is the iteration index and n is the number of equations, $\boldsymbol{H}$ is a matrix and $\{b\}$ is a vector. The elements in both of them are constant.

The difference in our problem is that the elements in both $\boldsymbol{H}$ and $\{b\}$ are functions of time, rather than constants. They need to be identified in time. Hence an integral operation is put into the iteration.

5.3.2 Mixed Iteration Scheme

In the equations of motion, the coefficients are functions of variables that are a functions of time. In considering this factor, the integral calculation should be put into the iterations so that the variables can be obtained and be used for updating the coefficients.

After m iterations the convergence of γ is checked. The value of m from numerical tests could be chosen between 10 and 20. The less it is, the worse the computational accuracy will be and the larger it is, the longer the computational time requirement will be.

Since both linear equations and differential equations are solved in an iterative cycle, it is called here a mixed iterative scheme. A flowchart describing this iterative scheme is shown below [see Figure (5.2) on the next page].

When this iterative scheme is used to solve equations of motion, the advantage in terms of computational time is very remarkable but the accuracy is decreased correspondingly. In order to improve accuracy, extra computations, such as using multistep formula or the high order Runge-Kutta formulas need to be made. These formulas can easily be introduced and implemented in the mixed iteration, but these numerical tests have not given us significant results yet. As usual, the increase of accuracy must be paid the price for increased computational time.

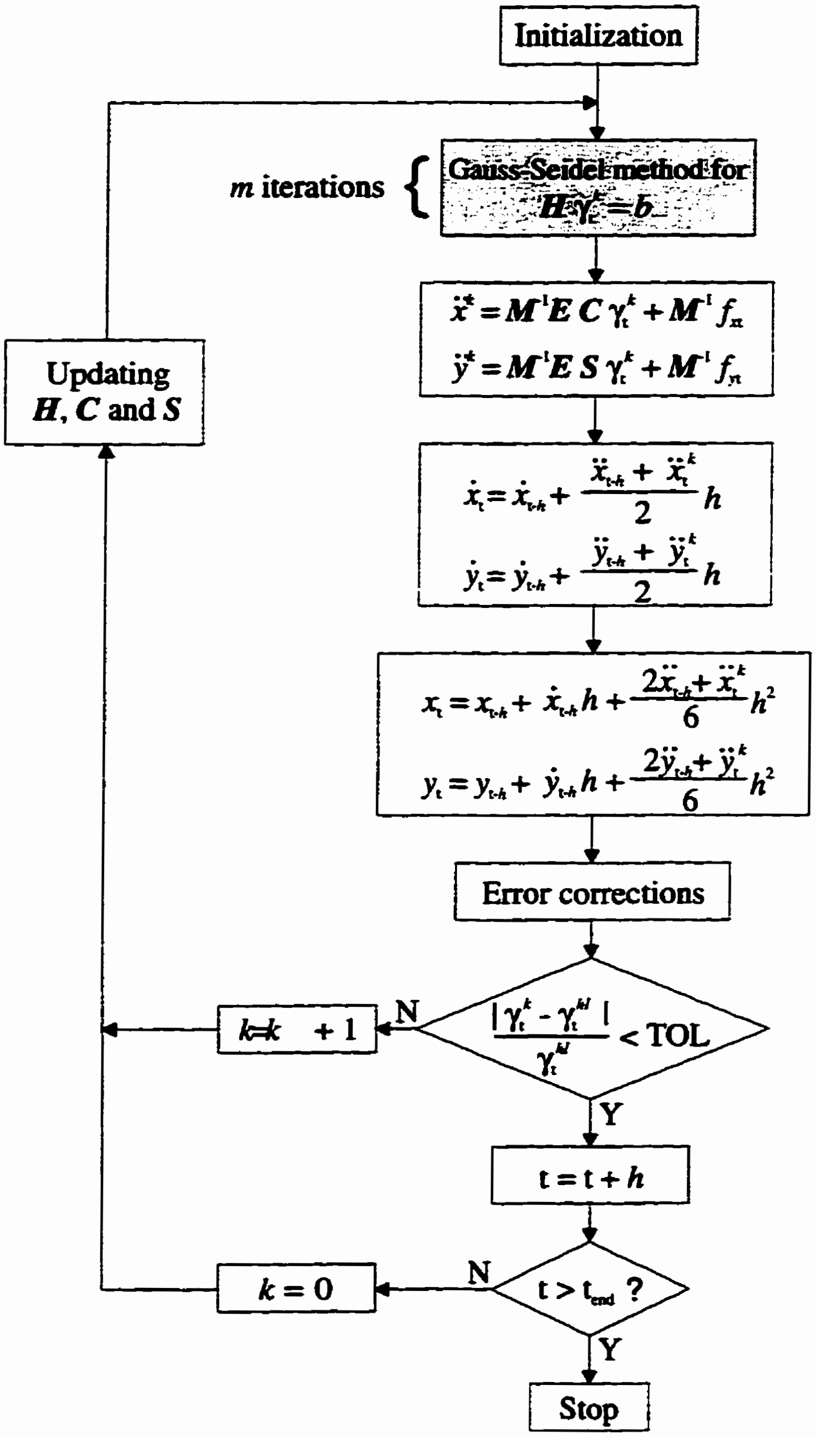

Initialization
m iterations
Gauss-Seidel method for $H\gamma_t^k = b$
$\ddot{x}^k = M^{-1}EC\gamma_t^k + M^{-1}f_x$
$\ddot{y}^k = M^{-1}ES\gamma_t^k + M^{-1}f_y$
$\dot{x}_t = \dot{x}_{t-h} + \frac{\ddot{x}_{t-h} + \ddot{x}_t^k}{2}h$
$\dot{y}_t = \dot{y}_{t-h} + \frac{\ddot{y}_{t-h} + \ddot{y}_t^k}{2}h$
$x_t = x_{t-h} + \dot{x}_{t-h}h + \frac{2\ddot{x}_{t-h} + \ddot{x}_t^k}{6}h^2$
$y_t = y_{t-h} + \dot{y}_{t-h}h + \frac{2\ddot{y}_{t-h} + \ddot{y}_t^k}{6}h^2$
Error corrections
$\frac{|\gamma_t^k - \gamma_t^{k-1}|}{\gamma_t^{k-1}} < \mathrm{TOL}$
N
$k=k+1$
Y
$t = t + h$
$t > t_{end}$?
N
$k = 0$
Y
Stop
Updating H, C and S

Figure 5.2 A flowchart of the mixed iterative scheme

5.3.3 Convergence Conditions

The convergence of the solution is dependent upon many factors. In our mixed iteration the conditions of convergence can be considered from two aspects. One is the convergent condition for the linear equations Equation (5.1). Another is the convergence condition for the integral formula.

According to relevant theory of numerical solution, if there is a linear system such as

$$\boldsymbol{A}\,\{x\} = \{b\} \tag{5.3}$$

and the corresponding iterative formula is

$$\{x\}^{i+1} = \boldsymbol{B}\,\{x\}^{i} + \{c\} \tag{5.4}$$

the sufficient condition for convergence will be $\| \boldsymbol{B} \| < 1$, i.e.,

$$||\boldsymbol{B}|| = \max_{1\le j\le m}\,(\sum_{i=1}^{m} |B_{ij}|) < 1 \tag{5.5}$$

For Equation (5.1) the convergence condition can be written as

$$|\boldsymbol{H}_{ii}| > \sum_{i=1\,(j\neq i)}^{m} |\boldsymbol{H}_{ij}| \tag{5.6}$$

This is only a sufficient condition, not a necessary condition. Moreover, in mixed iterations the integral calculations are also made, and the coefficients of matrix $\boldsymbol{H}$ are updated by new iterative results during each iterative cycle, even though these changes are very small.

There are two main reason why numerical solutions are affected when this iterative scheme is used. One is local truncation and round off errors in integral calculation; the other is global errors from the accumulation of local errors. In order to decrease global errors, we have to control local errors. Otherwise, they can accumulate rapidly bringing about a divergent numerical process.

Theoretically, the smaller the time step used in the mixed iterations is, the more accurate the numerical results are. This is may not be true because of computer round off errors. Furthermore, a smaller time step will lead to a longer calculation time which is inefficient. How to choose a proper time step is a complicated question because of the complexity of our dynamics system. This problem involves the size of the system, the accuracy requirement, the number of particles in contact and their configuration. Therefore, the time step should remain fixed during the whole calculation. This problem will be discussed in the following section.

5.3.4 Variable Time Step Algorithm

The technique of using a variable time step in numerical integration is very important. It has been used in many standard subroutines. Usually, the control of the time step is based on relative errors. Extra calculations are necessary for determining these errors. In the mixed iteration, the time step can be controlled by the number of iterations. In this way, without extra computations, the computational efficiency could probably be improved. The corresponding flowchart is shown on the next page.

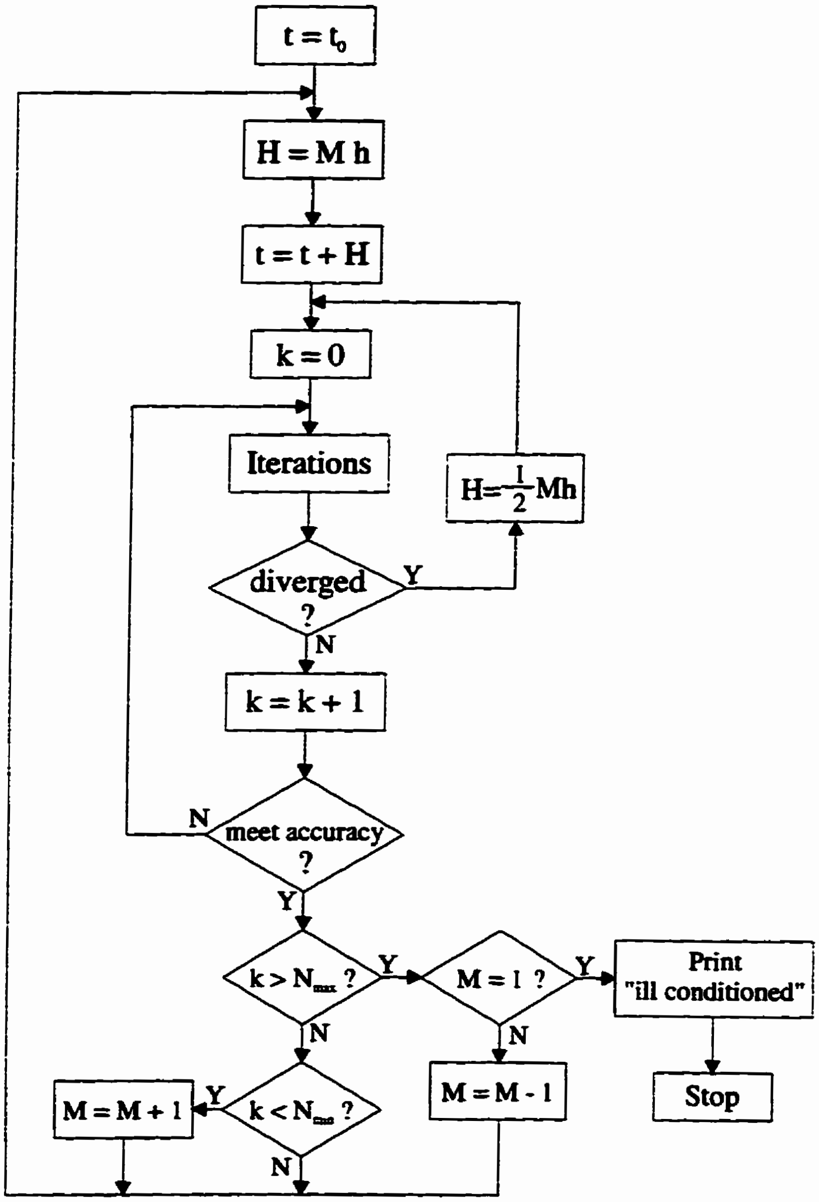


Figure 5.3 A variable time step algorithm

5.4 Violation Corrections

The mixed iterative scheme can attain third order accuracy, i.e., $O(h^3)$ in velocity and $O(h^4)$ in position, which is only less one order than the classical Runge-Kutta method. Violations exist only in a system with constraints. So-called violations are a kind of error which can be measured. In our problem, the computational errors are relative and the constraint violations are absolute.

The violations of results can be seen in the graphics. For instance, two connected disks could overlap or be separated in numerical results. Although velocity violations cannot be discovered in graphics, their existence will directly increase the position violations. The existence and correction of two kinds of errors is illustrated in Figure (5.4).

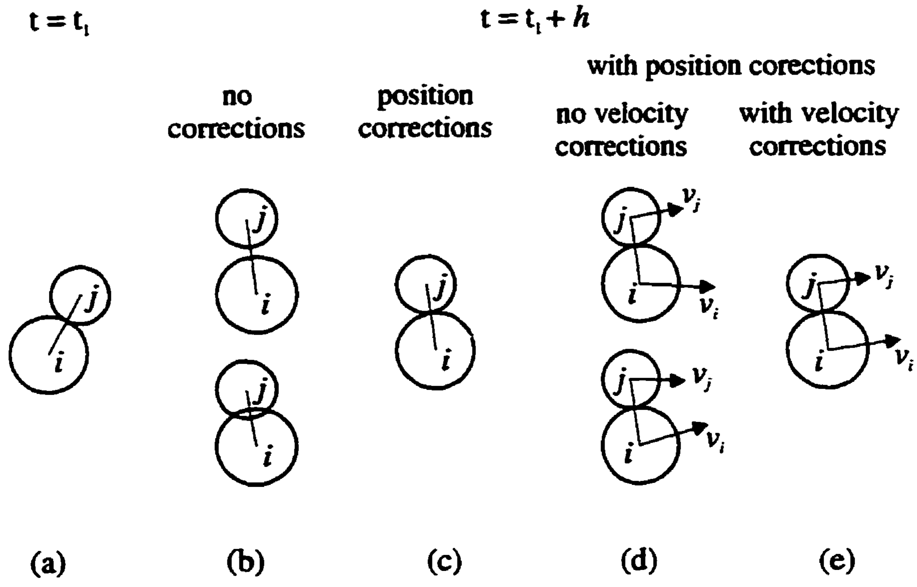


Figure 5.4 Illustration of position and velocity errors

If these violations are gradually accumulated without any corrections, they will make the iterative procedure longer and possibly even make the iterative results divergent until there is a complete crash. On the contrary, the correction of violation errors can improve the calculation accuracy and accelerate the numerical convergence. For this reason we put this capability into the iterative cycle. Some numerical results can be seen in the next chapter.

Violations are usually used to evaluate the accuracy of the numerical method, as in (Borri, et al., 1992). If the numerical method is accurate and the simulation time is not long, the violations needn't be corrected at all. Otherwise, the violations in both position and velocity will become large with time until they exceed the allowance. In this cases, they must be corrected.

In the simulation of a granular system, overlap between particles is not allowed. To overcome this difficulties one method is to introduce a "spring with damping" between two overlapping particles, such as in (Loset, 1994) and (Langston et al. 1994). The introduced spring force was acted between the two particles and its amount was proportional to the overlap. This method is applicable if the granular system is sparse, resulting a molecular dynamics model. By this method, the results of simulation depend on the computational accuracy. In addition, the equilibrium state of the system would be impossible or only maintainable over a short time if this method were used.

In a constrained system, errors in numerical results lead to violations of constraints. There are various ways of correcting these errors. For example, one based on the least-squares method, can be employed for the error corrections. The method which is used here is based on finding corrective forces which restore positions and velocities satisfying the corresponding constraint equations. The specific procedure is as follows:

Firstly, calculate the position and velocity of each particle whether it is connected with other particles or isolated. Secondly, check the violations of constraints for each pair of disks. Finally, find a corrective force for each pair of disks which eliminates (with a prescribed tolerance) the constraint violations.

5.4.1 Position Correction

The relative coordinates of the particles should, provided the link exits, satisfy the following relation

$$\{X_{ci}^2\} + \{Y_{ci}^2\} = \{L_i^{\,2}\} \tag{5.7}$$

where X_c and Y_c are ideal values. However, if Equation (5.7) is violated, then the error Δ is equal to

$$\{\Delta_i\} = \{(X_i^2 + Y_i^2)^{1/2}\} - \{L_i\} \tag{5.8}$$

Let the X and Y components of the position error vector be δX and δY, and they satisfy

$$\begin{aligned} \{X_{ci}\} &= \{X_i\} + \{\delta X_i\} \\ \{Y_{ci}\} &= \{Y_i\} + \{\delta Y_i\} \end{aligned} \tag{5.9}$$

By substituting Equation (5.9) into Equation (5.8) and leaving out the higher order terms, we obtain

$$-\boldsymbol{L}^{-1}(X_{ci}\{\delta X_i\} + Y_{ci}\{\delta Y_i\}) = \{\Delta_i\} \tag{5.10}$$

The error δX and δY are due to errors in individual disks positions, and the relationship between the two follows from Equation (3.48). Taking into account that X_i and Y_i are very closed to X_{ci} and Y_{ci} in Equation (3.45) for any pair of disks, Equation (5.10) can be written in the following form

$$\boldsymbol{CE}^T\{\delta x_j\} + \boldsymbol{SE}^T\{\delta y_j\} = \{\Delta_i\} \tag{5.11}$$

where $\{\Delta_i\}$ is position error vector. The first four terms of the Taylor's expansion in time for a disk position can be expressed as

$$\boldsymbol{r} = \boldsymbol{r}_0 + \dot{\boldsymbol{r}}_0 h + \frac{2\ddot{\boldsymbol{r}}_0 + \ddot{\boldsymbol{r}}}{6} h^2 \tag{5.12}$$

where $\boldsymbol{r} = (x, y)^T$ and the third order derivative of position in the fourth term is an average value. In considering of the position and its derivatives are constant during the iterations, we can obtain

$$\delta \boldsymbol{r} = \frac{1}{6} h^2 \delta \ddot{\boldsymbol{r}} \tag{5.13}$$

The x and y components of increments in the RHS of Equation (5.13) are found from Equation (3.50) and (3.51) taking into account that external forces are assumed to be constant during the time step. Then Equation (5.13) are transformed into the following

$$\begin{aligned} \{\delta x_j\} &= \frac{1}{6} h^2 \boldsymbol{M}^{-1} \boldsymbol{E}^T \boldsymbol{C} \{\delta \gamma_i\} \\ \{\delta y_j\} &= \frac{1}{6} h^2 \boldsymbol{M}^{-1} \boldsymbol{E}^T \boldsymbol{S} \{\delta \gamma_i\} \end{aligned} \tag{5.14}$$

By substituting Equation (5.14) into Equation (5.11), which gives

$$(\boldsymbol{CE}^T\boldsymbol{M}^{-1}\boldsymbol{EC}+\boldsymbol{SE}^T\boldsymbol{M}^{-1}\boldsymbol{ES})\{\frac{1}{6}h^2\delta\gamma_i\} = \{\Delta_i\} \tag{5.15}$$

If we denote

$$\{\frac{1}{6}h^2\delta\gamma_i\} = \{\delta l_i\} \tag{5.16}$$

then we can obtain the correction equations for the positions as follows

$$\boldsymbol{H}\{\delta l_i\} = \{\Delta_i\} \tag{5.17}$$

Equation (5.17) is solved for $\{\delta l\}$, and then position corrections are found from Equations (5.14) taking into account Equation (5.16), i.e.

$$\begin{aligned}\{\delta x_j\} &= \boldsymbol{M}^{-1}\boldsymbol{EC}\{\delta l_i\}\\ \{\delta y_j\} &= \boldsymbol{M}^{-1}\boldsymbol{ES}\{\delta l_i\}\end{aligned} \tag{5.18}$$

5.4.2 Velocity Correction

Similarly, the directions of relative velocities should be perpendicular to the centre line. The requirement which follows from the constraint equations, i.e., Equation (3.53). The latter equation after differentiation can be written in the form

$$\boldsymbol{C}\{\dot{X}_{ci}\} + \boldsymbol{S}\{\dot{Y}_{ci}\} = 0 \tag{5.19}$$

where X_c and Y_c are ideal values. If Equation (5.19) is not satisfied, then

$$\boldsymbol{C}\{\dot{X}_i\} + \boldsymbol{S}\{\dot{Y}_i\} = \{\eta_i\} \tag{5.20}$$

where $\{\eta_i\}$ is a velocity error vector. We can represent the X and Y components of the velocity error vector as the following form

$$\begin{aligned} \{\dot{X}_{ci}\} &= \{\dot{X}_i\} + \{\delta\dot{X}_i\} \\ \{\dot{Y}_{ci}\} &= \{\dot{Y}_i\} + \{\delta\dot{Y}_i\} \end{aligned} \tag{5.21}$$

and then substitute Equation (5.21) into Equation (5.20) and apply Equation (5.19) The result is

$$-\boldsymbol{C}\{\delta\dot{X}_i\} - \boldsymbol{S}\{\delta\dot{Y}_i\} = \{\eta_i\} \tag{5.22}$$

From Equation (3.49) the relationship between the relative and absolute velocity increments can be found and used in Equation (5.22) to obtain the following result

$$\boldsymbol{C}\boldsymbol{E}^T\{\delta\dot{x}_j\} + \boldsymbol{S}\boldsymbol{E}^T\{\delta\dot{y}_j\} = \{\eta_i\} \tag{5.23}$$

Considering again the first three terms of the Taylor's expansion for velocities

$$\dot{\boldsymbol{r}} = \dot{\boldsymbol{r}}_0 + \frac{\ddot{\boldsymbol{r}}_0 + \ddot{\boldsymbol{r}}}{2} h \tag{5.24}$$

The velocity increments are

$$\delta\dot{\boldsymbol{r}} = \frac{1}{2} h \,\delta\ddot{\boldsymbol{r}} \tag{5.25}$$

The x and y components of increments in the RHS of Equation (5.25) can be found as before, from Equations (3.50) and (3.51), and after their substitution into Equation (5.25), the latter become

$$\{\delta \dot{x}_j\} = \frac{1}{2} h \boldsymbol{M}^{-1} \boldsymbol{E}\boldsymbol{C}\{\delta\gamma_i\}$$
$$\{\delta \dot{y}_j\} = \frac{1}{2} h \boldsymbol{M}^{-1} \boldsymbol{E}\boldsymbol{S}\{\delta\gamma_i\} \tag{5.26}$$

By substituting Equation (5.26) into Equation (5.23) we obtain

$$(\boldsymbol{C}\boldsymbol{E}^T\boldsymbol{M}^{-1}\boldsymbol{E}\boldsymbol{C} + \boldsymbol{S}\boldsymbol{E}^T\boldsymbol{M}^{-1}\boldsymbol{E}\boldsymbol{S})\{\frac{1}{2}h\,\delta\gamma_i\} = \{\eta_i\} \tag{5.27}$$

If we denote

$$\{\frac{1}{2}h\,\delta\gamma_i\} = \{\delta v_i\} \tag{5.28}$$

then we can obtain the correction equations of velocities in the form

$$\boldsymbol{H}\{\delta v_i\} = \{\eta_i\} \tag{5.29}$$

From the latter $\{\delta v_i\}$ is found, and then the velocity corrections are found from Equation (5.26) taking into account notations in Equation (5.28)

$$\{\delta \dot{x}_j\} = \boldsymbol{M}^{-1}\boldsymbol{E}\boldsymbol{C}\{\delta v_i\}$$
$$\{\delta \dot{y}_j\} = \boldsymbol{M}^{-1}\boldsymbol{E}\boldsymbol{S}\{\delta v_i\} \tag{5.30}$$

5.4.3 Velocity Correction by Energy Conservation Law

The above correction of velocities makes changes only in the normal directions of relative velocities. When connected bodies meet velocity constraint conditions, all of the bodies in the system should satisfy an energy conservation condition. In other words, the changes of system kinetic energy should be equal to the work done by total external forces according to the energy conservation and conversion law. Because positions usually have a higher order accuracy than velocities, the work done in a small time step can be used for correction of velocities. If energy is conserved, we have

$$\frac{1}{2}\sum m_i(\dot{x}_i^2+\dot{y}_i^2)_{t+h}-\frac{1}{2}\sum m_i(\dot{x}_i^2+\dot{y}_i^2)_t-\sum(w_i)_h=0 \qquad (5.31)$$

The first and second terms in Equation (5.31) are respectively the kinetic energy of the system at time *t+h* and *t*, and the third term is the sum of work done by the external forces. If the right hand side in the equation above is not equal to zero (assumed to be ε), because of computational errors, i.e.,

$$\frac{1}{2}\sum m_i(\dot{x}_i^2+\dot{y}_i^2)_{t+h}-\frac{1}{2}\sum m_i(\dot{x}_i^2+\dot{y}_i^2)_t-\sum(w_i)_h=\varepsilon \qquad (5.32)$$

For *t+h*, let

$$\begin{aligned}\{\dot{x}_{cj}\} &= \{\dot{x}_j\}+\{\delta\dot{x}_j\}\\ \{\dot{y}_{cj}\} &= \{\dot{y}_j\}+\{\delta\dot{y}_j\}\end{aligned} \qquad (5.33)$$

Substituting Equation (5.33) into Equation (5.32) and eliminating high order terms, we have

$$-\sum m_i(\dot{x}_{ci}\delta\dot{x}+\dot{y}_{ci}\delta\dot{y})_{t+h}=\varepsilon \qquad (5.34)$$

We assume that the velocity error is proportional to the velocity magnitude and the proportionality is the same for all disks, i.e.

$$\begin{aligned}\{\delta\dot{x}_j\} &= k\{\dot{x}_j\}\\ \{\delta\dot{y}_j\} &= k\{\dot{y}_j\}\end{aligned} \qquad (5.35)$$

where k is a constant of proportion. Substituting Equation (5.35) into Equation (5.34), and considering that $(x_c,y_c)_{t+h}$ and $(x,\ y)_{t+h}$ are negligibly different we can obtain

$$k = -\frac{\varepsilon}{\sum m_i(\dot{x}_i^2+\dot{y}_i^2)} \tag{5.36}$$

Consequently the correction equations of velocity based on the energy conservation are

$$\begin{aligned} \{\delta\dot{x}_j\} &= -\frac{\varepsilon}{\sum m_i(\dot{x}_i^2+\dot{y}_i^2)}\{\dot{x}_j\} \\ \{\delta\dot{y}_j\} &= -\frac{\varepsilon}{\sum m_i(\dot{x}_i^2+\dot{y}_i^2)}\{\dot{y}_j\} \end{aligned} \tag{5.37}$$

Note:

(1) The velocity correction changes the direction of the velocity vector, while the correction on based on energy conservation changes the magnitude of velocity. Therefore, the two corrections are independent of each other.

(2) The corrections of positions after collisions is done at the end of the iterations for the time interval h.

(3) Corrections based on energy conservation in nonconservative systems can be implemented if a dissipative work can be accurately determined during each time step.

5.5 Summary

In this chapter we have discussed the numerical methods which were used for solving the equations of motion and we have given a comparison of results by those methods. In order to improve computational efficiency, a mixed iterative scheme which is aimed at our specific equations of motion was introduced and developed. With respect to accuracy, correction of the position and the velocity violation as well as correction of energy conservation violation were used for reducing the system global error during each time step.

The solution in traditional numerical methods is a pure mathematical problem, but here physical conditions were involved and used as grounds for improving accuracy. The new results after correction were used to feed back into the iterative cycle to accelerate convergence, or were used directly as new initial values for the next time step. Although this correction or control technique requires more calculations during the solution of the equations of motion, its effect is not only to improve computational accuracy but also to save computational time, which will be seen in the next chapter.

Chapter 6

APPLICATIONS AND ANALYSIS OF EFFICIENCY

6.1 Introduction

Because there are no analytical solutions to most dynamics problems, a direct and simple way to verify the efficiency of a numerical method is to apply it to various dynamics problems and compare the numerical results and the required CPU times. A combination of the Lagrangian formulism and classical Runge-Kutta methods will still be used here as a benchmark for comparisons.

The efficiency of computer simulation is affected by many factors such as system size, time step of integration, error tolerance and error control. The main factors we will take into account here are the system size, error tolerance and error control. The computational efficiency and a parametric study are performed for three simple and typical examples.

To implement the developed numerical methods and techniques presented in early chapters, a computer simulation program has been written. The program allows a user, through a choice of system parameters, to model virtually any two-dimensional granular system. Here this computer simulation program is used for three granular system examples and the results of simulation are compared with those of relevant physical experiments.

6.2 Multibody Pendulum Problems

As the first example a multibody pendulum problem described in Chapter 5 is considered. It is found that the computational efficiency is highest when the Newton-Euler equations are solved by the mixed iterative method. The efficiency increases with the system size [see the Figure (6.1)]. The total length of the pendulum was 3m, the initial angles from the vertical was 30°.

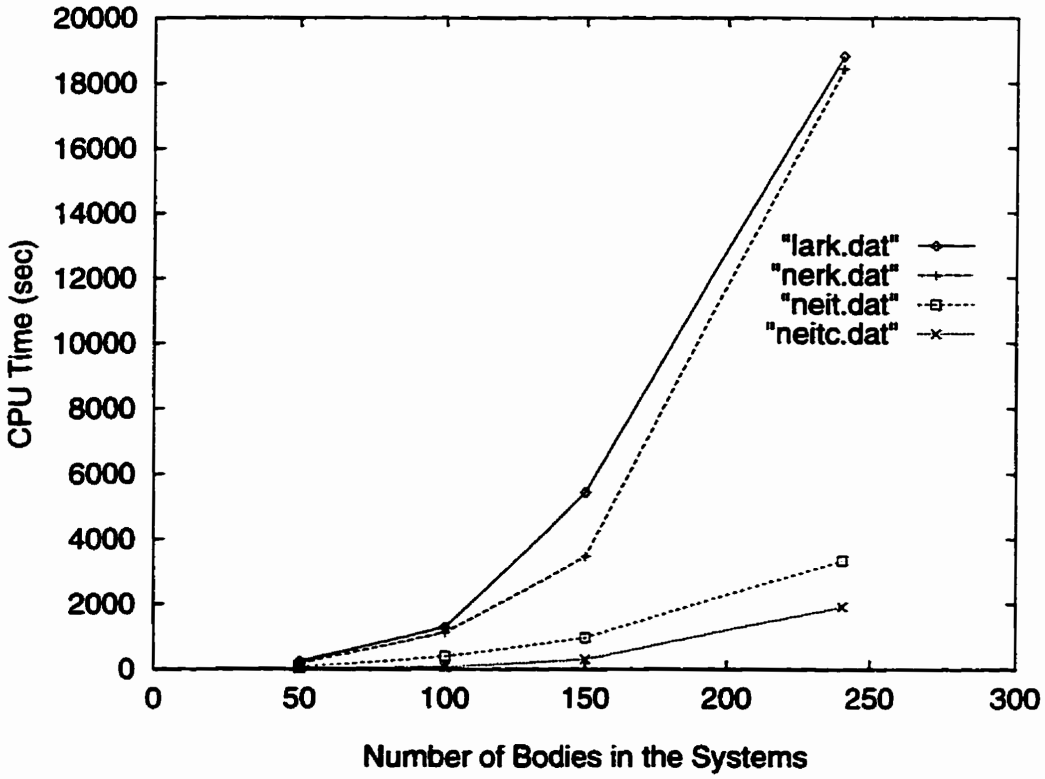


Figure 6.1 CPU time for different system sizes and numerical methods

In the figure above, some notations have the following meaning:

"lark.dat": Lagrangian formalism and Runge-Kutta solver,

"nerk.dat": Newton-Euler formalism and Runge-Kutta solver,

"neit.dat": Newton-Euler formalism and iterative solver with error correction,

"neitc.dat": Newton-Euler formalism and iterative solver with integrated error control.

Violations of both position and velocity need to be corrected at each time step. Usually, an increase in accuracy requires more computational time. But, as it will be shown, by using an iterative method, we can not only increase the accuracy but also decrease the number of iterations and thus save computational time. The following figure provides results which present the relationship between CPU time and error tolerance for the iterative solver with or without corrections.

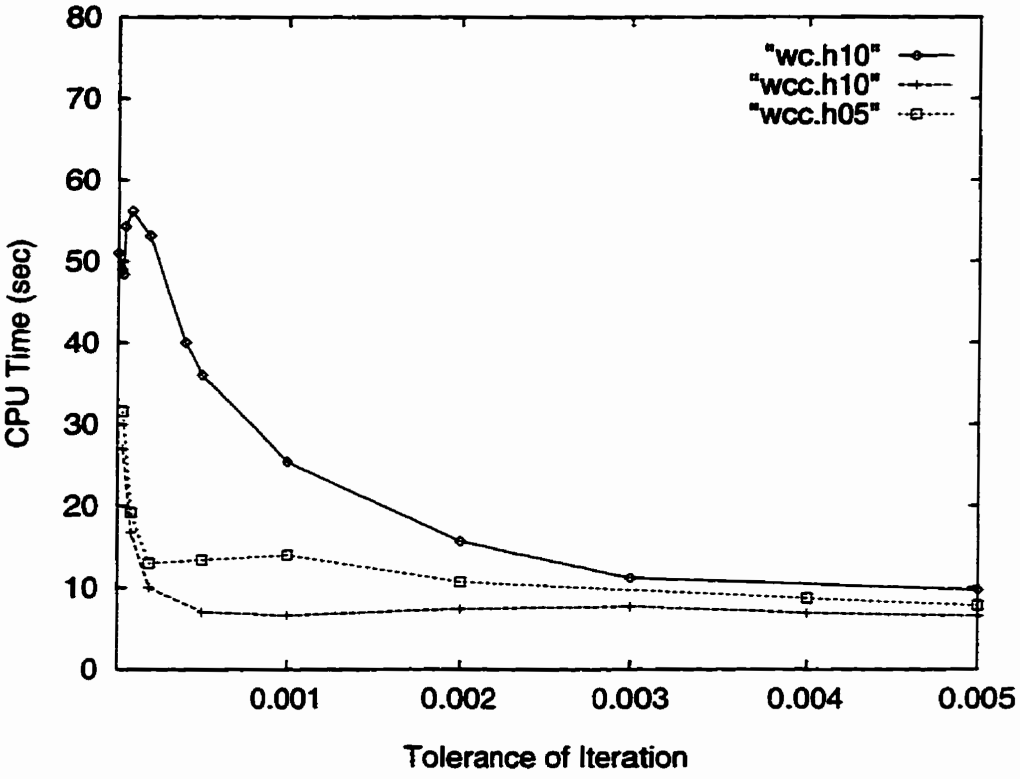


Figure 6.2 CPU time vs. error tolerance for N = 50

Where the meaning of data files is as follows:

"wc.h10": with error corrections at the end of iterations and h = 0.001 sec,

"wcc.h10": with integrated error control and h = 0.001 sec,

"wcc.h05": with integrated error control and h = 0.0005 sec.

The accuracies of computations can be seen in Figures (6.3) and (6.4).

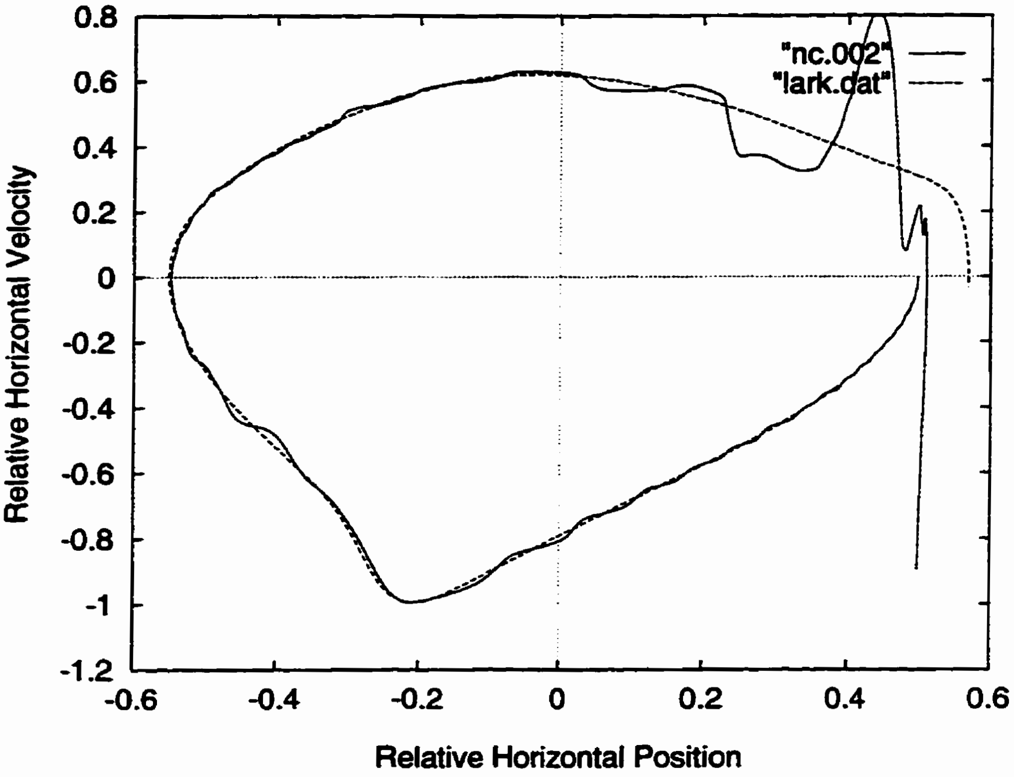


Figure 6.3 Phase diagram without error correction and control for $N = 50$

where the "lark.dat", in this and the following figures, identifies the results obtained by Lagrangian formalism and Runge-Kutta solver (the benchmark), the "nc.002" identifies the results obtained by Newton-Euler formalism and mixed iterative method without any correction for TOL=0.002 and h=0.0005 sec.

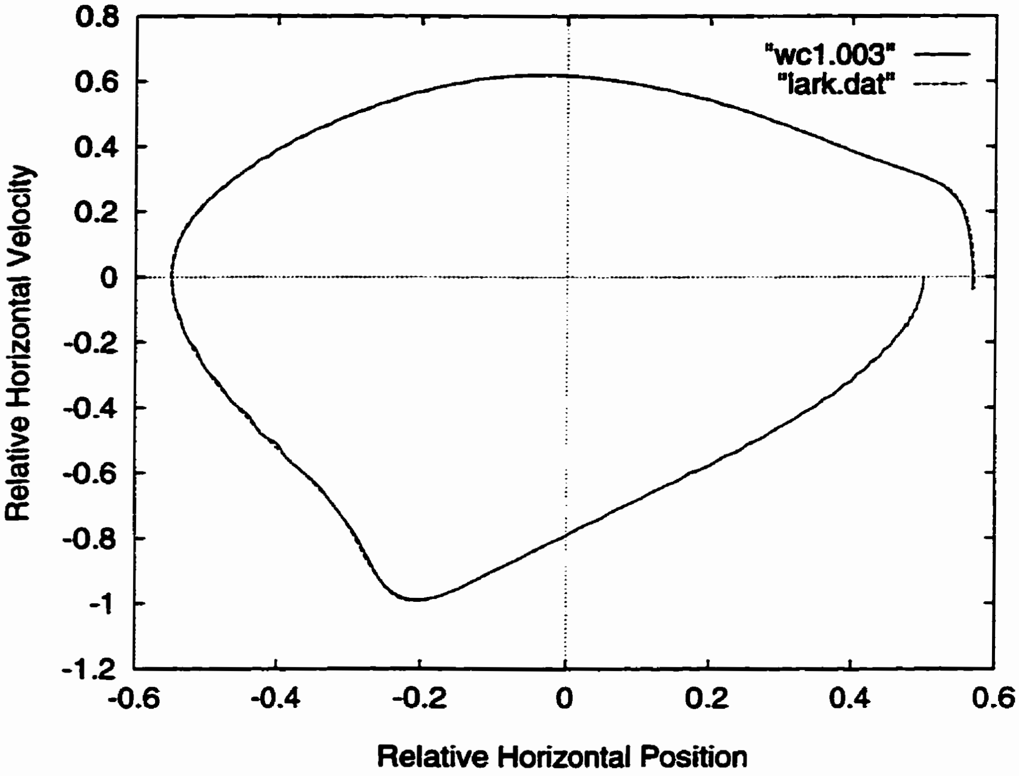


Figure 6.4 Phase diagram with error correction for $N = 50$

where "wc1.003" identifies the results obtained by Newton-Euler formalism and mixed iterative method with correction for TOL=0.003 and h=0.001 sec.

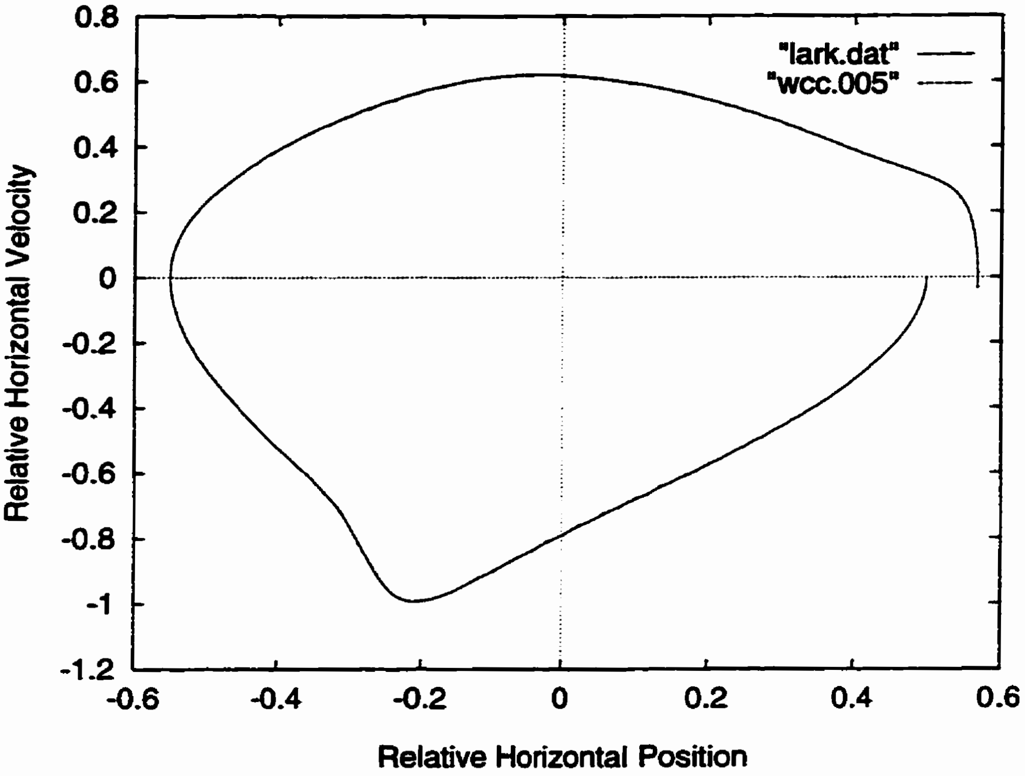


Figure 6.5 Phase diagram with error correction and control for $N = 50$

where "wc1.003" identifies the results obtained by Newton-Euler formalism and mixed iterative method with correction for TOL=0.005 and h=0.001 sec.

When error correction is integrated into the iterations, the accuracy is further improved. At the same time, the CPU time becomes less, which means that the integrated error control accelerates the convergence of the iterative process.

From Figures (6.3), (6.4), and (6.5) we can see that the integrated error control in the iterative scheme allows us to increase accuracy for larger tolerance TOL and time step *h*. This is important since choosing the TOL and *h* becomes less critical to the problem of convergence.

The effect of the TOL on the CPU time and convergence is investigated in detail in the following two figures for different numbers of bodies.

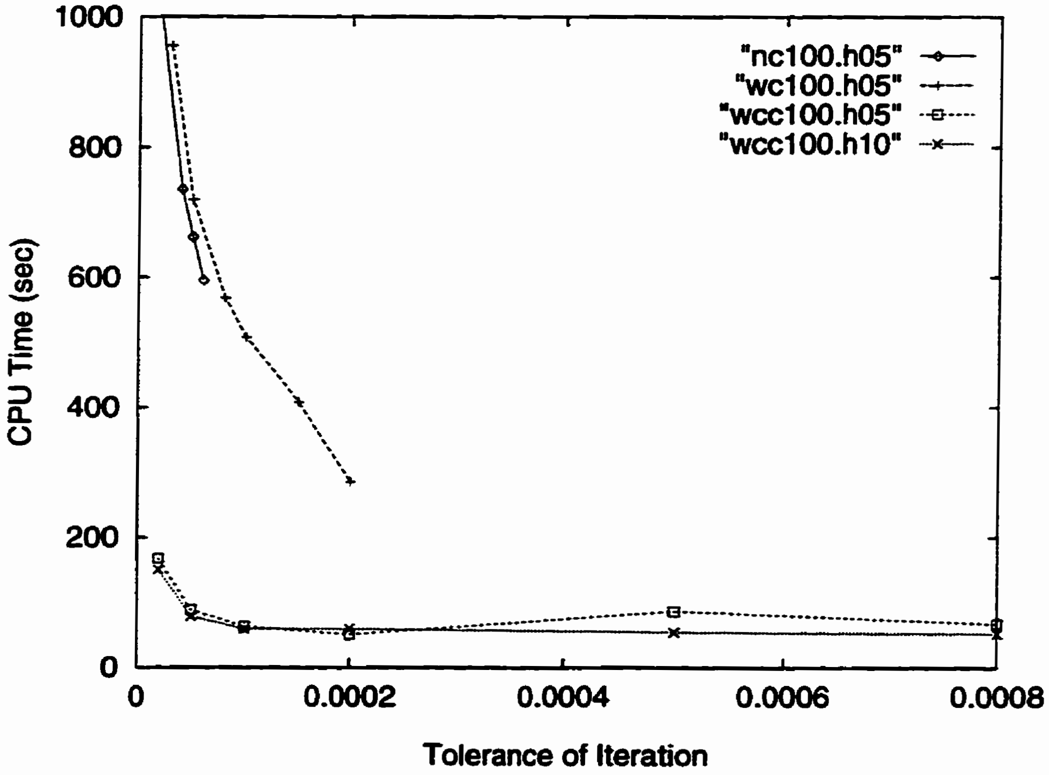


Figure 6.6 CPU time vs. error tolerance for $N = 100$

where:

"nc100.h05": no error correction and control (N=100, h=0.0005 sec),

"wc100.h05": with corrections only (N=100, h=0.0005 sec),

"wcc100.h05": with corrections and control (N=100, h=0.0005 sec),

"wcc100.h10": with corrections and control (N=100, h=0.001 sec).

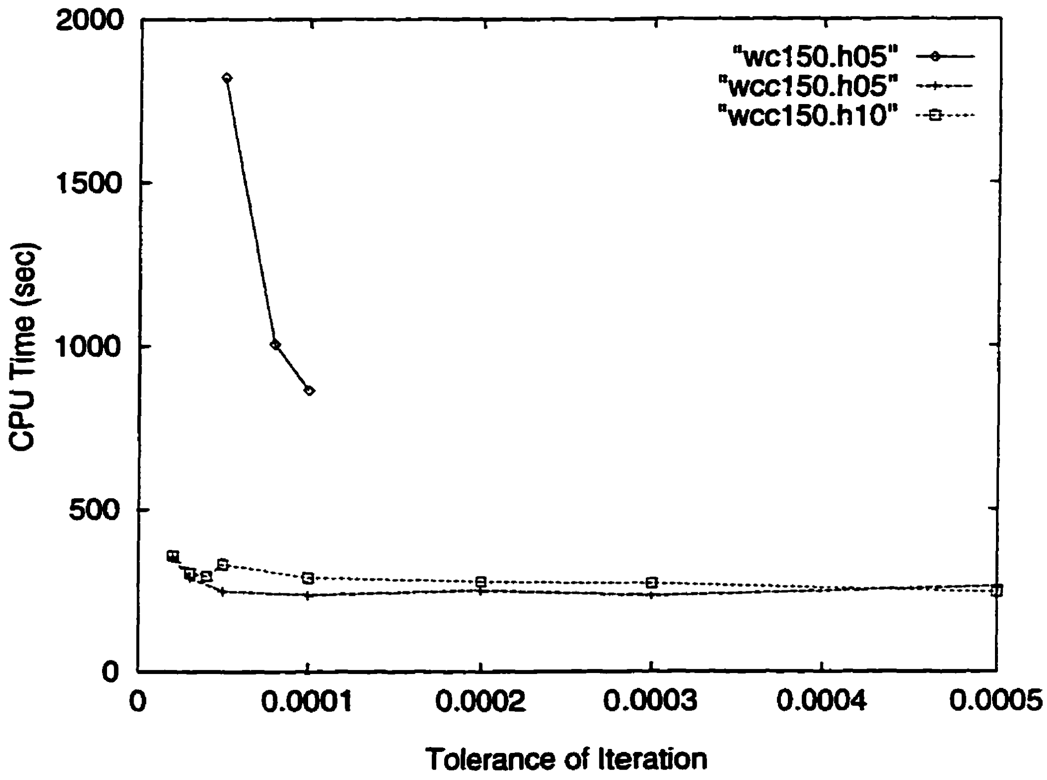


Figure 6.7 CPU time vs. error tolerance for N = 150

where:

"wc150.h05": with corrections only (N=150, h=0.0005 sec),

"wcc150.h05":	with corrections and control (N=150, h=0.0005 sec),
"wcc150.h10":	with corrections and control (N=150, h=0.001 sec).

Note that if there are no error corrections, a correct solution can be obtained for a large size system (N=150) only for very small error tolerances.

Therefore, error corrections are very important in numerical simulations in general. In particular, when they are integrated into the iterative cycle, they allow us to accelerate the convergence of iterations and at the same time improve the accuracy of results.

6.3 Falling Chain Problems

This is another problem test which has been solved by different numerical methods (Kamman and Huston, 1984; Mello, 1989 and Borri, et al., 1992). Two kinds of models are used here for the solution of this problem: One is a multi-rigid-body model in which the initial momentum of each body needs to be taken into account. The other is the point-mass model in which the masses are concentrated at the joints and the inertia of links is ignored.

For the first model, the chain is composed of fifteen identical rigid bars connected by spherical joints and acted upon by gravity. The two ends of the chain are grounded by means of two spherical joints. The configuration of this system and its initial position are shown in Figure (6.8).

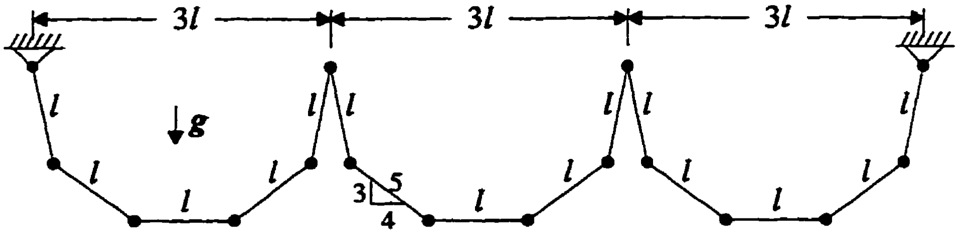


Figure 6.8 A falling chain with 15 links

Besides Lagrangian formulism, the Newton-Euler formulism for a multi-rigid-body system is used for the governing equations. The form of equations has been given in Section 3.5.2 of Chapter 3.

In a constrained system, the equations of motion using Lagrangian formulism are a modification to the Lagrangian equations [refer to Equations (3.19)]. This system has 13 degrees of freedom and two Lagrangian multipliers are needed. So two algebraic equations based on constraint relations need to be set up.

When using the Newton-Euler formula for a multi-rigid-body system, we do not need to consider the algebraic constraint equations because the constraint relationships (derivative form) are integrated into the equations. The constraint problem does not change the form of equations of motion. Here we still use generalized coordinates, so the number of internal force parameters is 2x14=28, and the number of integration variables is 15. Both equations of motion use the Runge-Kutta solver without the violation constraints.

The two different formulations of the governing equations give us almost the same results [see Figure(6.9)]. The first reflection point occurs at time $t = 0.47$ sec and the second point at $t = 0.68$ sec. These results are in complete agreement with those presented by Kamman & Huston (1984); Mello (1989) and Borri et al., (1992). Note that each rod has the same mass and a length of one foot.

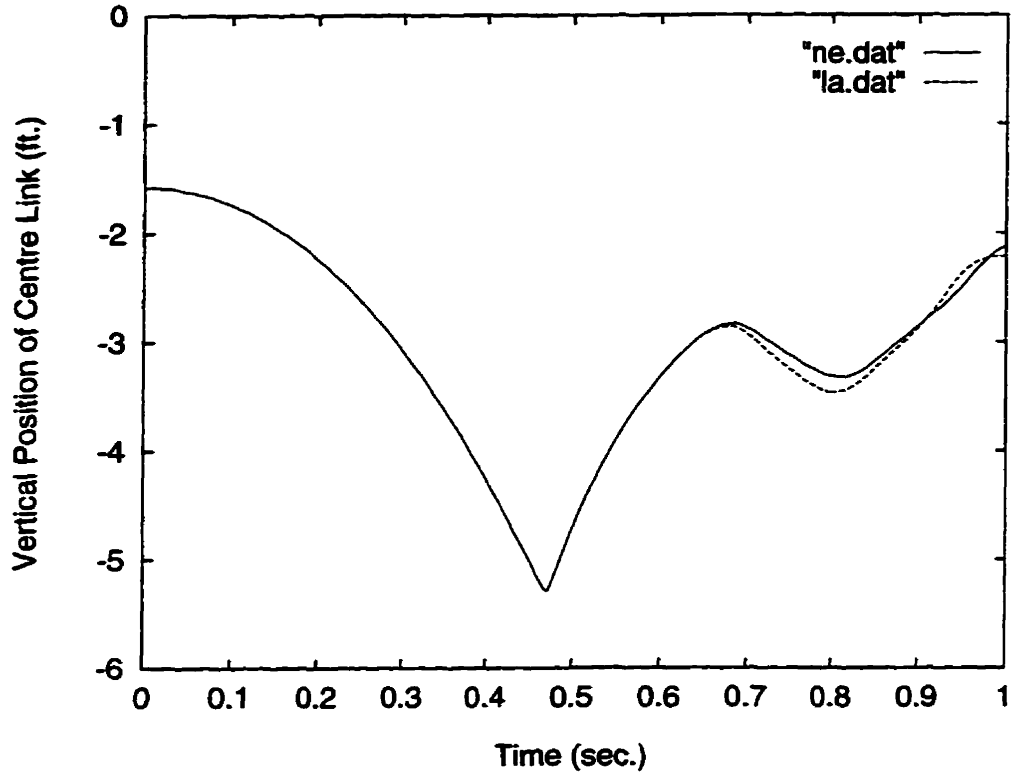


Figure 6.9 Vertical position time history of centre link (rigid body model)

In Figure (6.9), the "ne.dat" is the result based on the Newton-Euler formula and the "la.dat" is the result based on the Lagrangian formula.

A point-mass model is adopted here, i.e., a chain which is composed of fifteen massless rigid bars connected by fourteen spherical joints. All of the joints are mass points with the same mass and the whole system is acted upon by gravity. The configuration of the system and its initial position are the same as in a multibody model shown in the Figure (6.8). The number of degrees of freedom for this system is still 13 although the system model has changed. There is no difference between the results when the Lagrangian or the Newton-Euler formulation is used. However, we used rectangular coordinates for the Newton-Euler formulism, so the number of internal force parameters is 15, and the number of integration variables is 2×14=28. Meanwhile, for an iterative method, an error control was used to obtain numerical results. All these results for two different formulas and solvers are shown in Figure (6.10)

In this system, the distance between two fixed suspended points is 3 metres, i.e $3l$ = 1 m rather than l = 1 ft taken in (Kamman and Huston, 1984) and (Mello, 1989). Moreover, relative displacement is used to represent vertical displacement, which is equal to the vertical displacement divided by the distance between two suspended points.

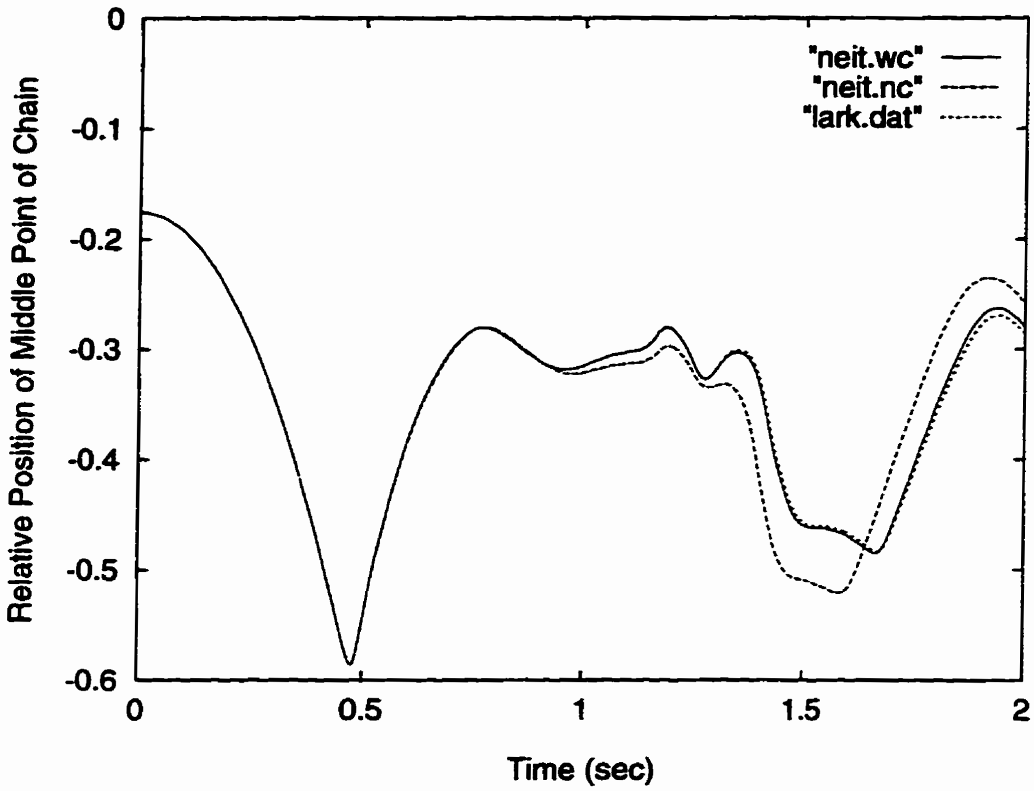


Figure 6.10 Relative vertical position time history of centre link (point-mass model)

In Figure (6.10) the "neit.wc" indicates the results based on the Newton-Euler formulation and mixed iterative method with full error control (in position, velocity and energy); the "neit" indicates the results based on Newton-Euler equations and mixed iterative method. The suffix "nc" means without any corrections and controls and the suffix "wc" means with all error controls; the "lark.dat" indicates the results based on the Lagrangian formulation and Runge-Kutta solver, which is used as a benchmark for comparisons.

From Figure (6.10) it can be seen that the iterative method with all error corrections has a higher accuracy. The question people might ask is which kind of correction is more important. The answer can be found in Figure (6.11).

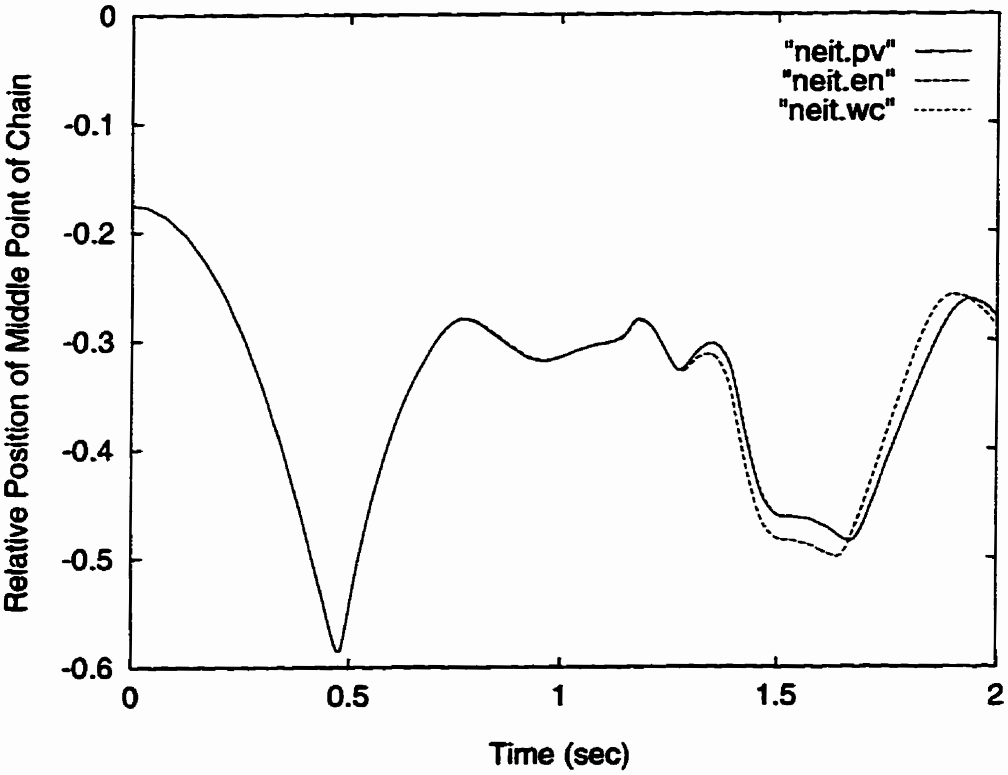


Figure 6.11 Relative vertical position time history of centre link (point-mass model) using different error controls

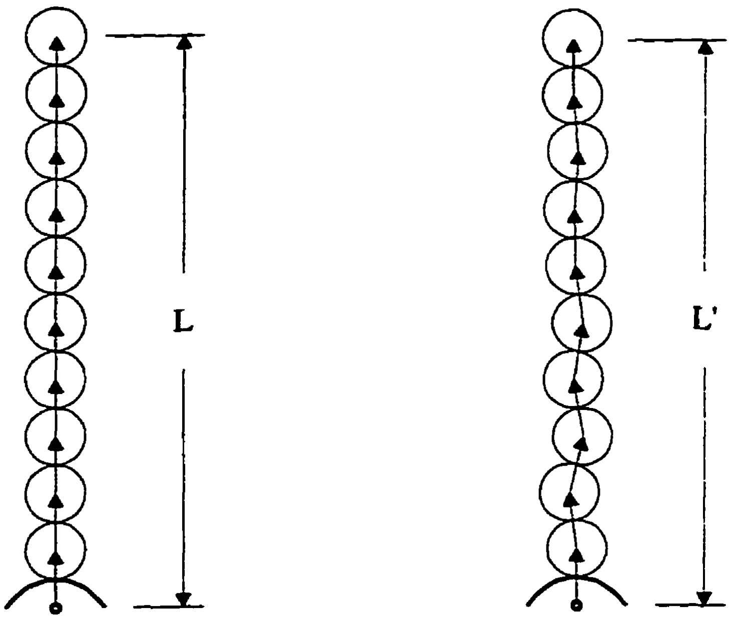
In Figure (6.11) the suffix "pv" means the results obtained by position and velocity error control. The suffix "en" means the results obtained by energy error control and the suffix "wc" means the results obtained by all of those error controls.

Obviously, position and velocity corrections play an important role in all error corrections. Is this true for all dynamics problems? To answer, let us consider the next example.

6.4 Unstable Problem for Vertical Stack of Disks

This is a simple and typical unstable dynamics problem which can be used for verification of accumulation of computational errors and evaluation of numerical methods, such as in (Wierzba, 1991).

The test model is still the point-mass truss system [see Figure (6.12)].



(a) Initial Position (b) Losing Stability

Figure 6.12 Instability test for N disks

Here is an assumption that the instability occurs when $\Delta \geq 0.1\%$, where Δ is determined by

$$\Delta = \frac{L - L'}{L} \tag{6.1}$$

In this example the number of disk N was varied from 10 to 60, while $L = 3$m. The results indicating the times to keep the system stable vs. different number of disks are shown in Figure (6.13). Also , the effect of different error control mechanisms is shown in the Figure.

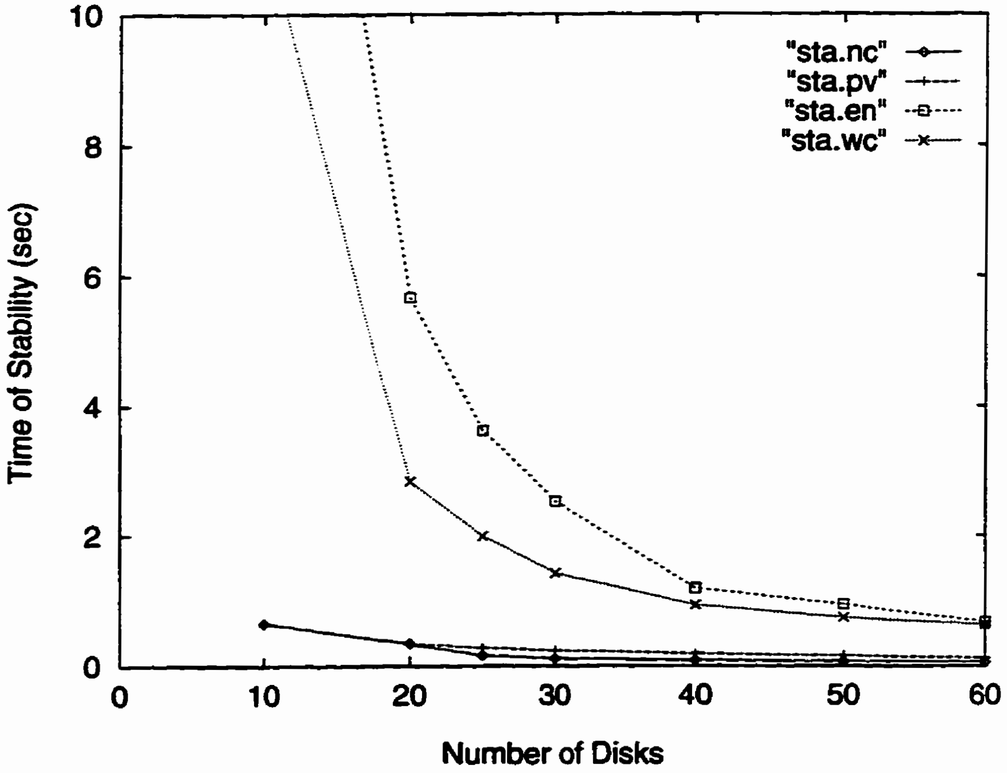


Figure 6.13 Time to instability vs. number of disks

In Figure (6.13) the meaning of suffixes is the same as that mentioned above.

In this physically unstable problem, energy error correction and control is very important especially when the system is not large. For this example the position and velocity violation corrections appear to be useless and they may even have a negative effect. This result requires additional investigations.

After comparisons of many computational results, it can be found that if a system is a moving one, then correction in velocity is most important in three types of error corrections (position, velocity and energy); if a system is close to a static one, then energy correction becomes most efficient. Although position correction is not significant, it cannot be eliminated because of the geometric and topological structure of the system. Hence, the choice of corrections is based on the state of the system. For a complicated granular system whose state is not known in advance, all of the error corrections should be utilized.

6.5 Jamming problem of solid particles in a straight pipe

This is a real granular materials dynamics system which is completely different from the examples above. Not only does it have a large size but also a variable topology. In computer simulations of this system, almost all aspects discussed above are important. It is an overall verification of the general simulation program.

6.5.1 Physical Model of the System

Particles carried by a fluid in a channel interact with each other and with the boundaries. This interaction results in a loss of kinetic energy and thus reductions in particle velocities. If the rate of particle supply remains the same, then the fraction of particles in a given volume, where speed reduction takes place, will increase. If the inflow exceeds outflow then the net balance will grow until the particle density reaches its maximum, which will correspond to a complete stagnation or a flow jam.

It is a discrete system which comprises isolated particles and clusters. All of the particles were modelled by rigid disks with different radii. At the beginning, the disks were uniformly distributed in the area which is called the *Generation Area* [see Figure (6.14)]. Disks with velocities the same as the flow gradually move into a *Buffer Area* where they acquire the velocity of the corresponding potential fluid. The buffer area is needed to allow sufficient number of disks to maintain a potential flow distribution of velocities before mixing due to interactions with the boundaries starts. From the buffer area the disks enter the real pipe with corrugated boundaries, i.e., *Control Area*. New disks are generated in the process of simulation, while the disks leaving the control area are discarded.

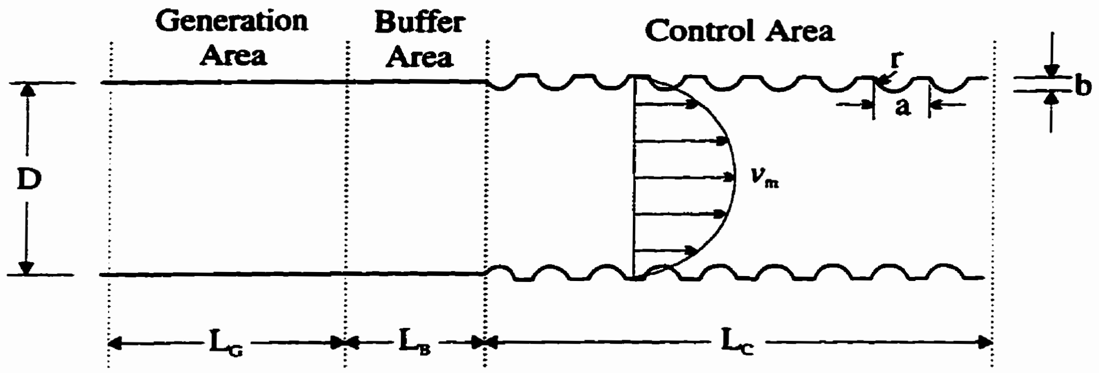


Figure 6.14 A schematic diagram of simulation area

6.5.2 Numerical Results

Some numerical results have been published in (Sun and Vinogradov 1996a, 1996b, and 1996c) and (Vinogradov and Sun, 1996), which will not be given in detail. However, the following results obtained here are based on the different size of particles.

Numerical simulations were carried out for the following channel and disk geometries: D = 0.4 m, d = (0.040 and 0.068) m, L_G = 1.25D, L_B = 0.5D, L_C = 3.75D, a = 0.125D, b = 0.05D, r = 0.1D. The coefficient of friction was μ = 0.3, and the coefficient of restitution e = 0.7. The maximum velocity at the centre of the channel was $v_m = 1.5v$, where v is the mean velocity corresponding to a given discharge and it varied in numerical experiments from v = 0.025 m/sec to v = 0.10 m/sec.

The simulations were carried out on the IBM-RS6000 computer. There was no limit on the number of disks. Those out of the control area were discarded and new ones were generated as long as the simulation was running. The simulation run was stopped when the buffer area was full, i.e., when the discharge was zero or close to zero, which was the sign of a jam in the pipe. For a typical run with a number of disks over 400 (average radius is 0.02 m, v= 0.025 m/sec), the CPU time was about 20 minutes while the real time was about 45 minutes. If average radius of disks is 0.034 m and v=0.05 m/sec, then total number of the disks during the simulation is not over 200 and the CPU time was about 1 minute, while the real time was about 4 minutes. The numerical experiment results for the two cases are plotted in Figures (6.15) in relative surface density - Froude number. The Froude number is defined by

$$F_r = \frac{v}{\sqrt{gD}} \tag{6.2}$$

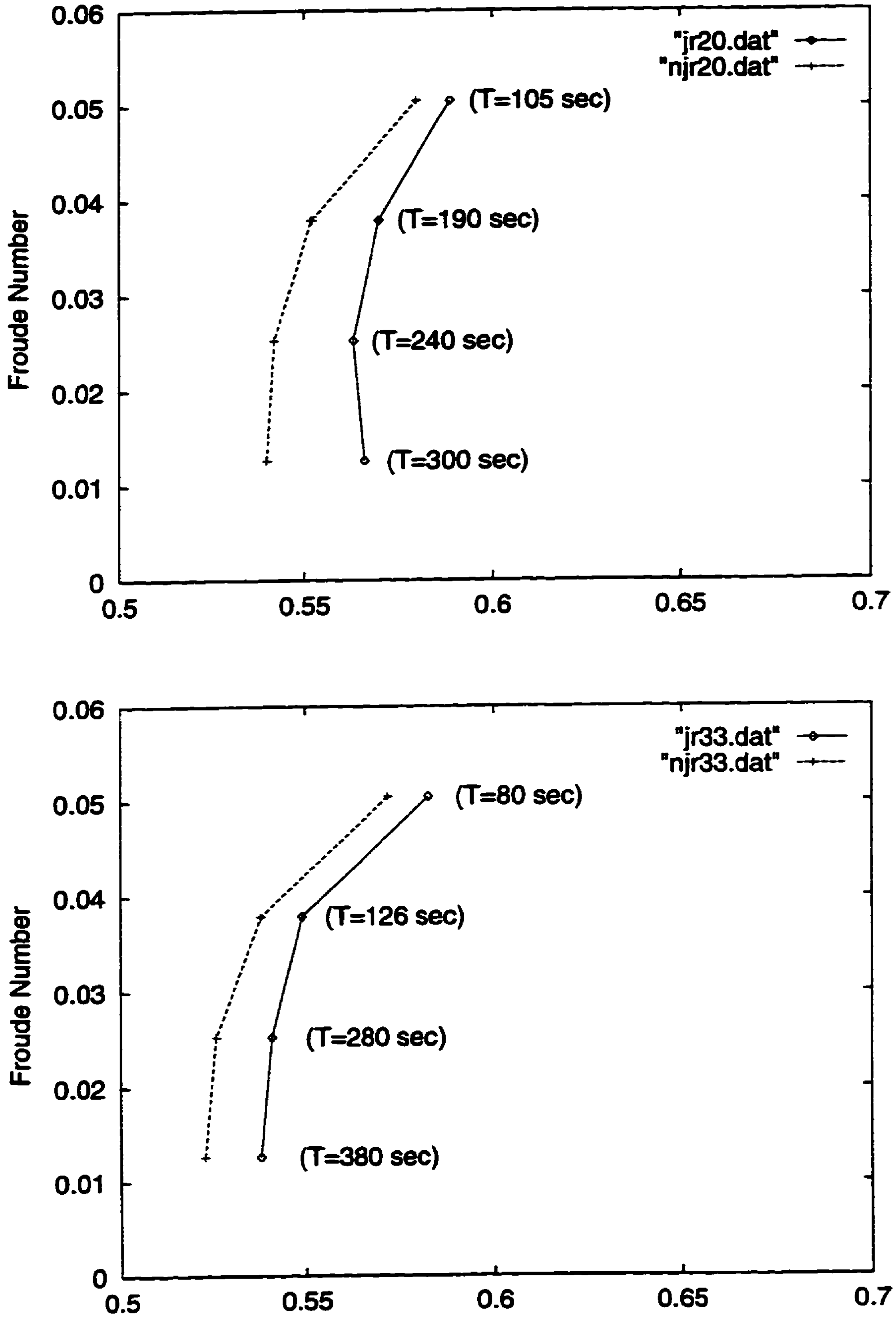


Figure 6.15 Relative densities of particles resulting in a flow jam

In the top of Figure, the dotted line represents no jam; the rigid line represents jam and the numbers in brackets indicate the time to jam initiation. The results for the average radius r_A= 0.020 m are shown in the upper part; the results for the average radius r_A= 0.034 m are shown in the lower part of the Figure (6.15). Between the two lines, the initiation of a jam is uncertain and the faster the flow velocity is the wider the area of uncertainty.

It is seen that where the relative size of the disks is d/D = 0.1 and 0.17 and the relative size of the wall roughness b/D = 0.05, the boundary between the no-jam and certain-jam areas is almost a vertical line (or transition zone). Although these results could not be checked quantitatively against the experimental data, since in (Ettema, 1989) the initiation of a jam is associated with the volume discharge of the spheres, the experimental data indicates that for a relatively large particle size (in experiments it was d/D = 0.02) the no-jam-certain-jam boundary is almost a vertical line. Similar results are obtained for relatively large blocks in a curved channel (Urroz and Ettema, 1994). It is also of interest to note that the time to the beginning of the onset of jamming depends on the initial flow velocity, namely, it is almost in inverse proportion to the latter.

The results of simulations for the systems with different size are shown in Figures (6.16a), (6.16b) and Figures (6.17a), (6.17b).

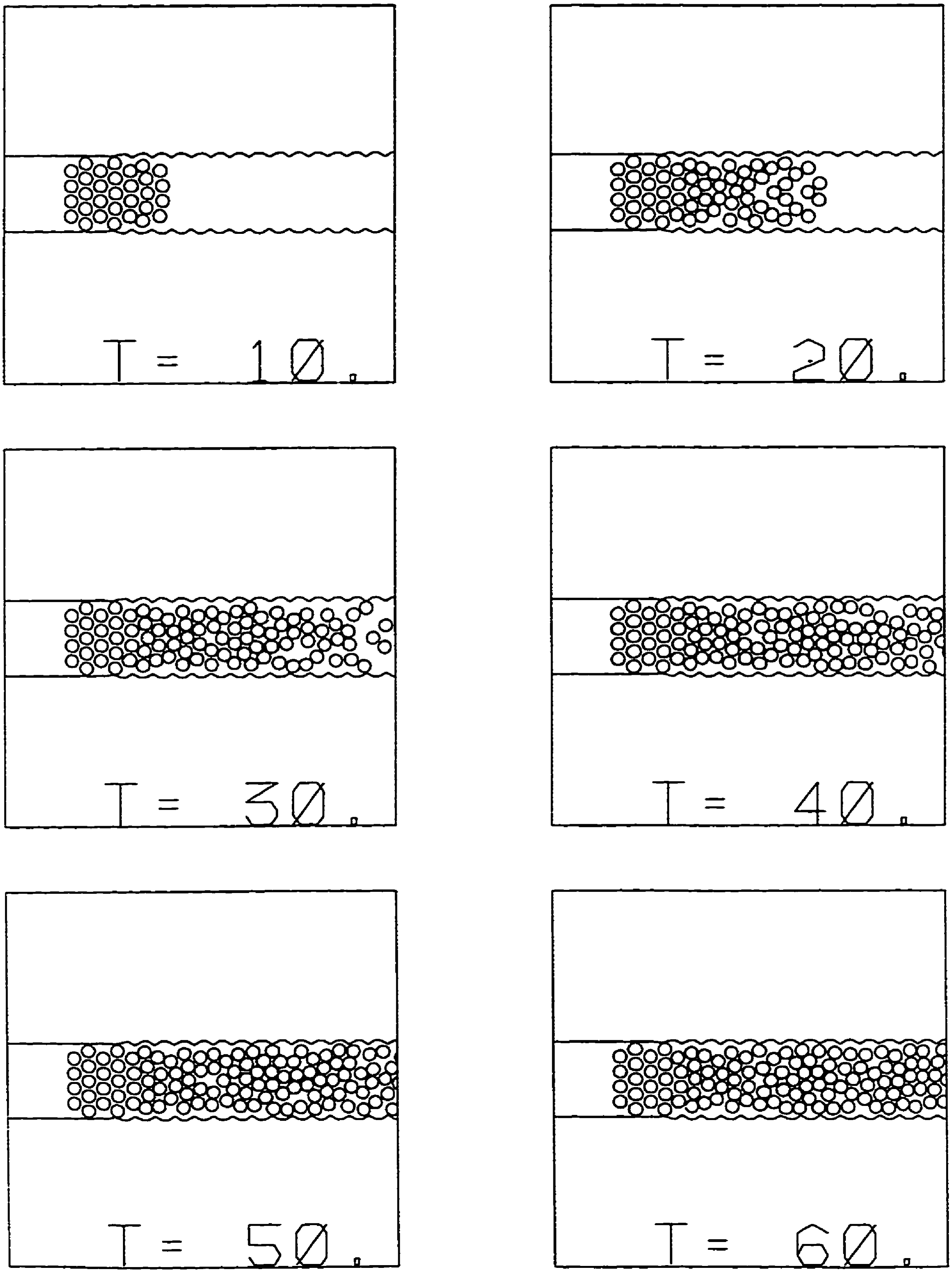


Figure 6.16a Simulations of a two-phase granular flow (large particles) in a straight pipe (from the beginning)

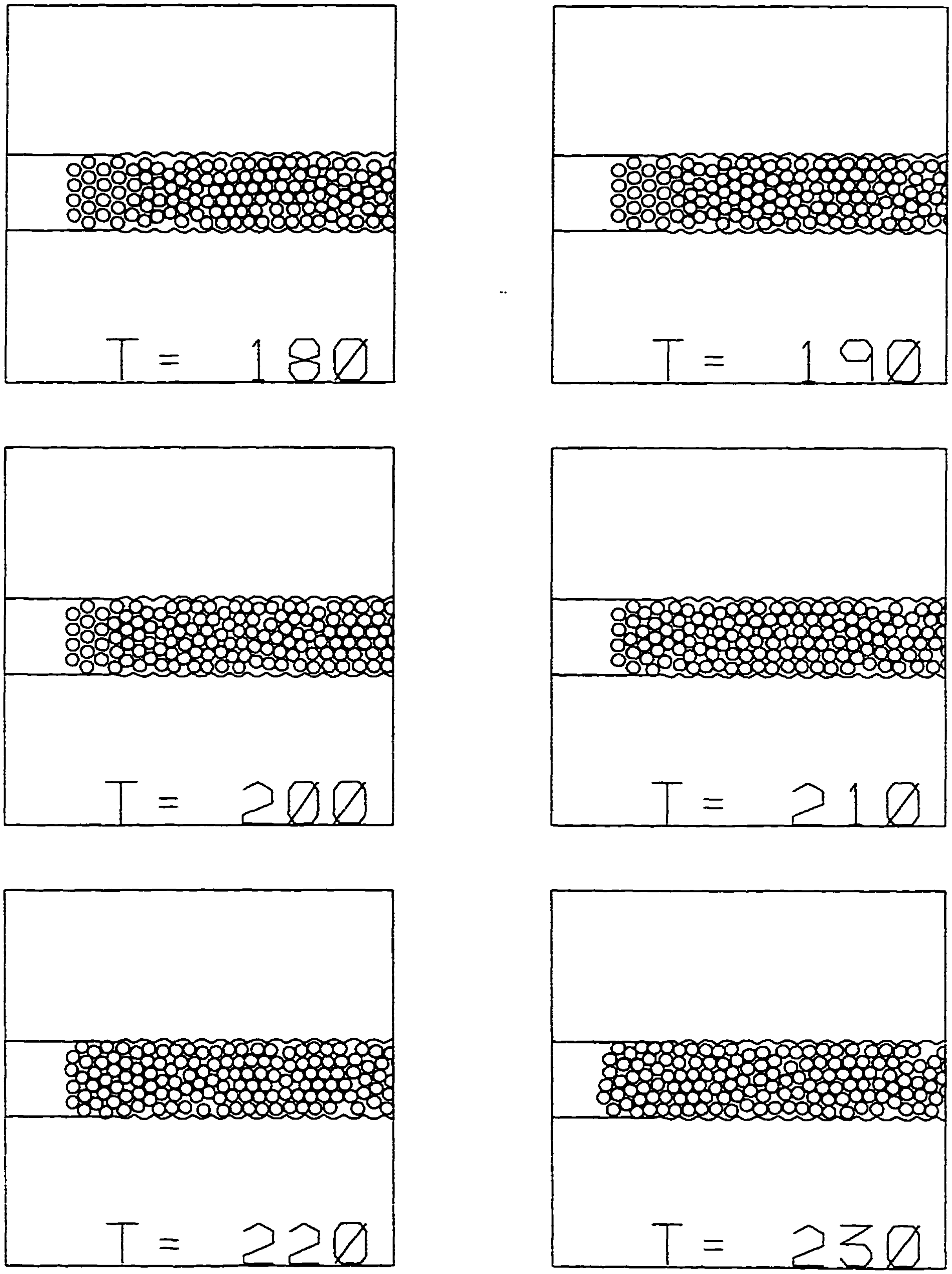


Figure 6.16b Simulations of a two-phase granular flow (large particles in a straight pipe (before jamming)

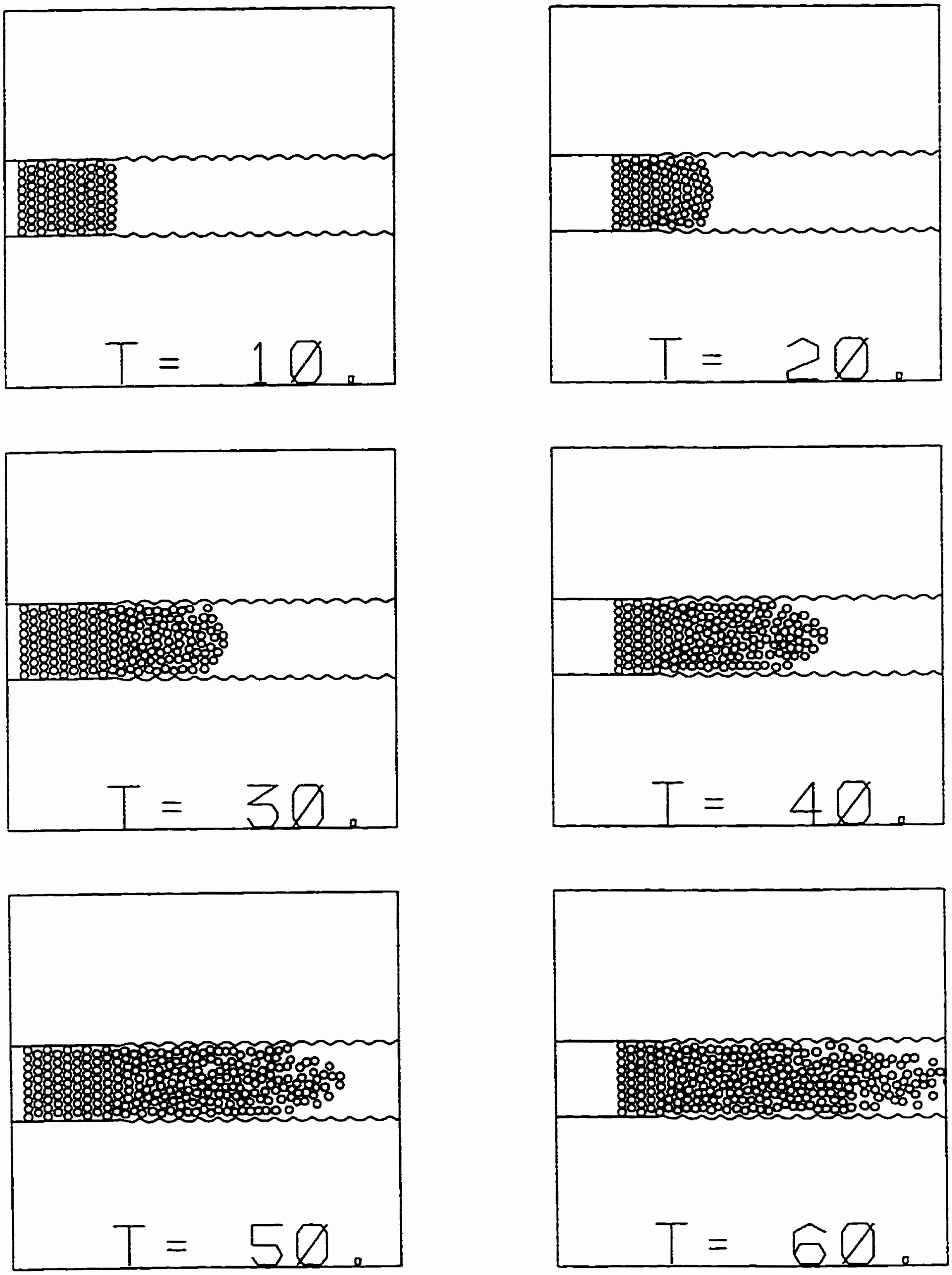


Figure 6.17a Simulations of a two-phase granular flow (small particles) in a straight pipe (from the beginning)

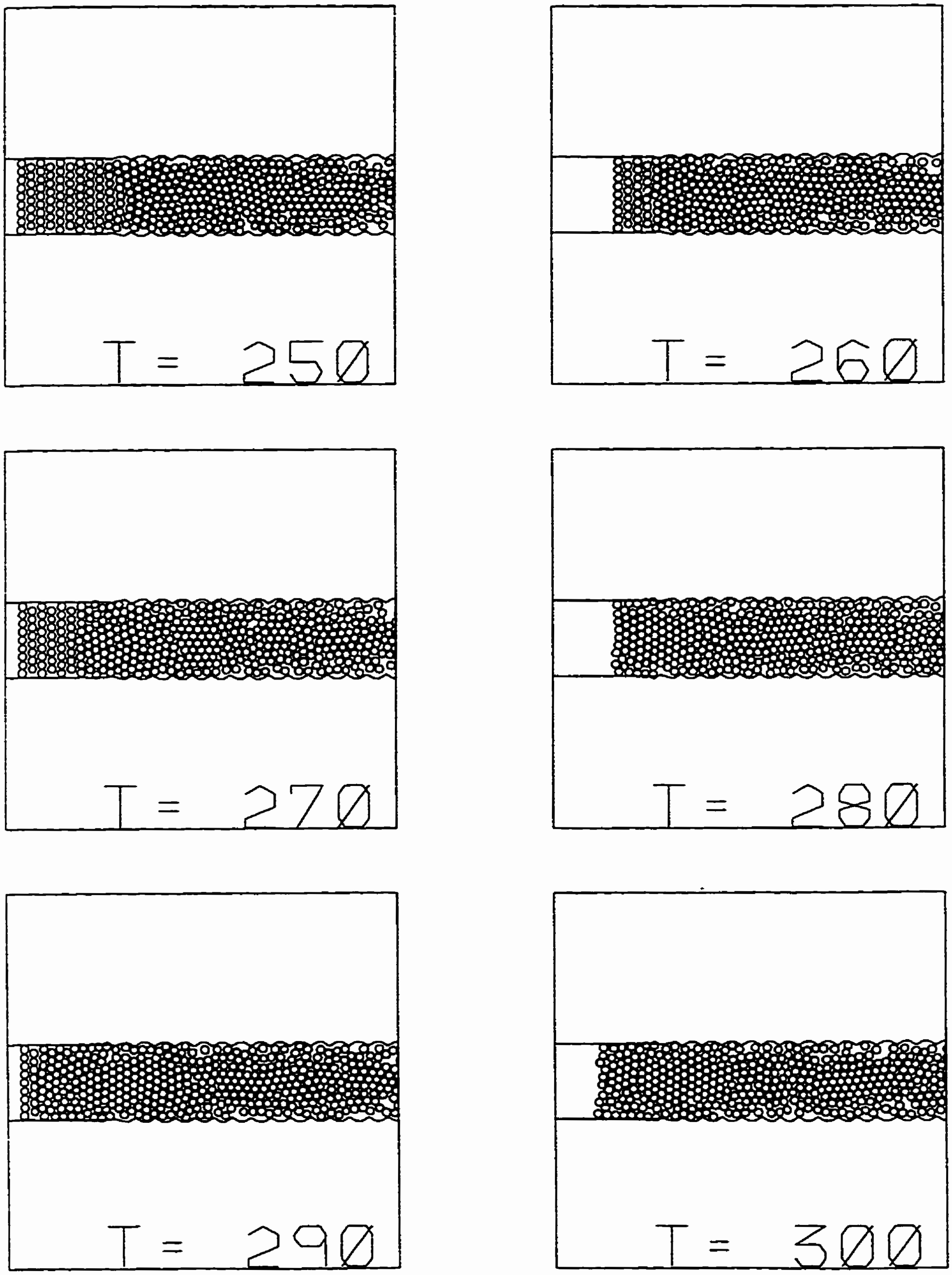


Figure 6.17b Simulations of a two-phase granular flow (small particles) in a straight pipe (before jamming)

6.6 Summary

Three typical examples and a specific granular system are discussed in this chapter. The examples are used to test different numerical methods and some parameters in the corresponding methods. The advantages of the iterative method are very significant in terms of the computational speed. The improvement of accuracy is achieved through error corrections and controls in each iterative cycle. This control technique allowed us to further improve computational efficiency.

In this chapter, we also investigated the influences and effects of error corrections in different dynamics system.

The application of the computer simulation program is an actual verification of methods proposed in the earlier chapters. In this program, none of the matrices was used. Both computational time and storage space were greatly saved.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

7.1. Conclusions

A general methodology for simulating the dynamics of systems has been developed. A simplified multibody dynamics model has been introduced based on the Newton-Euler approach. A novel integrative scheme for solving the governing equations has been developed, which allowed significant improvement in the computational efficiency of simulations. An error correction algorithm integrated into the iterative cycle has been developed. This algorithm not only improved the accuracy of results but also accelerated the convergence of iterations. In short, the methodology makes it possible to analyze the dynamics of granular systems using a multibody model without need for a supercomputer.

The main accomplishments of the present investigation can be summarized as follows:

(1) A physically correct discrete model in 2D and 3D granular dynamics simulations has been introduced.

(2) The topology of the system is described by the incidence matrix which allows updating of the topological data and equations of motion in a scalar rather than a matrix form. The latter resulted in savings of both

computational time and storage space.

(3) A customized algorithm based on splitting the iterative process in two coupled branches: one for unknown internal forces, and another for unknown positions, has been developed. This algorithm has proved to be very efficient for large size systems.

(4) An algorithm for error corrections in position, velocity and energy has been developed and made a part of the iterative scheme, which accelerated the convergence and improved the accuracy of results.

(5) A methodology of handling events (such as collisions, break up or joining of particles, generation of new particles) has been presented. This new methodology allows improved efficiency, in terms of CPU time, of simulations of large systems.

(6) A FORTRAN program implementing the methodology and algorithms has been developed, tested and successfully implemented on a digital computer.

7.2. Recommendations

The following tasks and investigations may be carried out as an extension of the present work:

(1) A more rigorous treatment of quasi-rigid-bodies, which may lead to both better accuracy and more efficient simulations, should be investigated.

(2) Slip between particles in contact should be taken into account and the rotational energy of an individual disk or sphere should be considered.

(3) The effect of the fluid velocity field can be included in the calculation of the drag forces on particles.

(4) The problems of stability and convergence for the iterative scheme requires a more rigorous mathematical investigation.

(5) A dynamic storage space in the current computer simulation program can be implemented by using a FORTRAN 90 compiler or the entire program can be converted into another language, such as Pascal or C++, to utilize more efficiently dynamic data management capabilities of these languages. In addition, a graphical interface can be used to output the data directly onto the screen. Then it will be possible to realize a real time simulation for a small size granular system.

References

Ackermann, N.L., Shen, H.T., 1982, Stresses in Rapidly Sheared Fluid-solid Mixtures, *Journal of the Engineering Mechanics Division*, ASCE., 108, 95-113.

Babic, M., 1990, Discrete Element Modelling of Granular Flows, *Ph.D. Thesis*, Clarkson University, Potsdam, New York.

Babic, M., 1993, Gravity-Driven Flows of Smooth Inelastic Disks Between Parallel Bumpy Boundaries, *Journal of Applied Mechanics*, Vol. 60, 59-64.

Bagnold, R.A., 1954, Experiments on Gravity-free Dispersion of Large Solid Sphere in a Newtonian Fluid under Shear, *Proceedings of the Royal Society, London*, Ser. A, 225, 49-63.

Bagnold, R.A., 1966, The Shearing and Dilatation of Dry Sand and the 'Singing' Mechanism, *Proceedings of the Royal Society, London*, Ser. A, 295, 219-232.

Barker, G.C., 1994, Computer Simulations of Granular Materials, *Granular Matter - An Interdisciplinary Approach*, Ed. A. Mehta, Springer-Verlag, 35-83.

Baxter, G.W. and Behringer, R.P., 1990, Cellular Automata Models of Granular Flow, *Physics Review*, A, 42, 1017-1020.

Borja, R.I. and Wren, J.R., 1995, Micromechanics of granular Media Part I: Generation of Overall Constitutuive Equaiton for Assemblies of Circular Disks, *Computer Methods in Applied Mechanics and Engineering*, 127, 13-36.

Borri, M. et al., 1992, A Modified Phase Space Formulation for Constrained Mechanical System - Differential Approach, *European Journal Mechanics, A/Solids*, 11 No. 5, 701-727.

Brach, Raymond M., 1991, Mechanical Impact Dynamics - *Rigid Body Collisions*, John Wieley & Sons, 164-165.

Brown, R.L. and Richards, J.C., 1970, Principles of Powder Mechanics, Pergamon Press, Oxford.

Campbell, C.S. and Brennan, C.E., 1985a, Chute Flows of Granular Material: Some Computer Simulations, *Journal of Applied Mechanics*, 52, 172-178.

Campbell, C.S. and Brennan, C.E., 1985b, Computer Simulation of Granular Shear Flows, *Journal of Fluid Mechanics*, 151, 167-188.

Campbell, C.S., 1989, The Stress Tensor for Simple Shear Flows of a Granular Material, *Journal of fluid Mechanics*, 203, 449-473.

Cundall, P.A., 1971, A Computer Model for Simulating Progressive, Large-scale Movements in Block Rock Systems, *Proc. Symp. Int. Soc. Rock Mech.*, Nancy 2, No.8.

Cundall, P.A. and Strack, D.L. (1979) A Discrete Numerical Model for Granular Assemblies, *Geotechnique* Vol. 29, No. 1 47-65.

Ettema, R., 1989, Jam Initiation in Unobstructed Channels, *Proceedings of 23d Congress of IAHR*, Ottawa, 19-29.

Fitt, A.D. and Wilmott, P., 1992, Cellular-automation Model for Segregation of a Two-species Granular Flow, *Physics Review*, A 45, 2383-2388.

Forrest, S.B. and Haff, P.K., 1992, Mechanics of Wind Ripple Stratigraphy, *Science*, 255, 1240-1243.

Gutt, G.M. and Haff, P.K., 1990, An Automation Model of Granular Materials, in *Proceeding 5th Distributed Memory Computing Conference*, Vol. 1, IEEE. Computer Society Press, Charleston, SC.

Haff, P.K., 1983, "Grain Flow as a Fluid Mechanical Phenomena", *Journal of Fluid Mechanics*", 134, 401-430.

Haff, P.K. and Werner, B.T., 1987, Collisional Interaction of a Small Number of Confined Inelastic Grains, *Colloidal and Interfacial Phenomena*, Vol. 3, *Particulate and Multiphase Processes*, Ariman, T. and Vesiro Veziroglu, T.N., Eds., Hemisphere, Washington, D.C., 483-501.

Haff, P.K., 1994, Discrete Mechanics, *Granular Matter - An Interdisciplinary Approach*, Ed. A. Mehta, Springer-Verlag, 141-160.

Hanes, D.M. and Inman, D.L., 1985, Observations of Rapidly Flowing Granular-fluid Materials, *Journal of Fluid Mechanics*, 150, 357-380.

Hopkins, M.A. and Louge, M.Y., 1990, Inelastic Microstructure in Rapid Granular Flow of Smooth Disks, *Physics Fluids*, A, 3, 47-57.

Hui, K. and Haff, P.K., 1986, Kinetic Grain Flow in a Vertical Channel, *International Journal of Multiphase Flow*, 12, 289-298.

Huston, R.L. et al., 1994, Use of Absolute Coordinates in Computational Multibody Dynamics, *Computers & Structures*, Vol. 52, No.1, 17-25.

Jenkins, J.T. and Cowin, S.C., Theories for Flow of Granular Materials", *Symp. Mechanics Applied to the Transport of Bulk Materials, American Society of Mechanical Engineers*, Buffallo, N.Y., 79-89.

Jenkins, J.T. and Savage, S.B., 1983, A Theory for the Rapid Flow of Identical, Smooth, Nearly Elastic Particles, *Journal of Fluid Mechanics*, 130, 187-202.

Jiang, Z. and Haff, P.K., 1993, Multiparticle Simulation Methods Applied to the Micromechanics of Bed Load Transport, *Water Resources Research*, 29, 399-412.

Johnson, P.C. and Jackson, R., 1987, Frictional-collisional Constitutive Relation for Granular Materials, with Application to Plane Shearing, *Journal of Fluid Mechanics*, 176, 67-93.

Ju, M.S. and Mansour, J.M., 1989, Comparison of Methods for Developing the Dynamics of Rigid-body System, *The International Journal of Robotics Research*, Vol. 8, No. 6, 19-27.

Kamman, J.W. and Huston, R.L., 1984, Dynamics of Constrained Multibody Systems, *Journal of Applied Mechanics*, 51, 899-903.

Kanatani, K.I., 1979, A Micropolar Continuum Theory for the Flow of Granular Materials, *International Journal of Engineering Science*, 17, 419-432.

Langston, P.A., Tuzun, U., and Heyes, D.M., 1994, Continuous Potential Discrete Particle Simulations of Stress and Velocity Fields in Hoppers: Transition from Fluid to Granular Flow, *Chemical Engineering Science*, 49, 1259-1275,

Loset, Sveinung, 1994, Discrete Element Modelling of a Broken Ice Field - Part I: Model Development, *Cold Regions Science and Technology*, 22, 339-347.

Lun, C.K.K. Savage, S.B., Jeffrey, D.J. and Chepurniy, N., 1984, Kinetic Theories for Granular Flow: Inelastic Particles in Couette Flow and Slightly Inelastic Particles in a General Flow Field, *Journal of Fluid Mechanics*, 140, 223-256.

Mello, F., 1989, Weak Formulations in Analytical Dynamics, with Applications to Multi-Rigid-Body Systems, Using Time Finite Elements, *Ph.D. thesis*, Georgia Institute of Technology, Atlanta, Georgia.

Ogava, S., Umemura, A., Oshima, M., 1980, On Equations of Fully Fluidized Granular Materials, *Journal of Applied Mathematics and Physics*, ZAMP., 31, 483-493.

Richman, M.W. and Chou, C.S., 1988, Boundary Effects on Granular Shear Flows of Smooth Disks, *Z. Angew. Mathematical Physics (Journal of Applied Physics Mathematics)*, 39, 885-901.

Roberson, R.E. and Schwertassek, R., 1988, Dynamics of Multibody Systems, Springer-Verlag, 184-200.

Rooney, G.T. and Deravi, P., 1982, Coulomb Friction in Mechanism Sliding Joints, *Mechanism and Machine Theory*, 17, 207-211.

Rothenburg, L. and Bathurst, R.J., 1992, Micromechanics Features of Granular Assemblies with Planar Elliptical Particles, *Geotechnique*, 42, 79-95.

Savage, S.B., 1984, The Mechanics of Rapid Granular Flow, *Advanced Applied Mechanics*, 24, 289-366.

Savage, S.B., 1995, Flows of Granular Materials, *Proceedings 15th Canadian Congress of Applied Mechanics*, Edrs. B. Tabarrok and S. Dost, Victoria, B.C., 62-73.

Springer, W.A.J., 1989, A Simulation of the Motion of Clusters of Rigid Bodies around Obstruction, *M.Sc. Thesis*, Department of Mechanical Engineering, University of Calgary.

Sun, Y., Vinogradov, O., Gavrilova, M. and Rokne, J., 1994, An Algorithm of Updating System State in Simulation of Dynamics of Granular-type Materials, *Proceedings of the 1994 Summer Computer Simulation Conference*, SCS., San Diego, CA., 45-50.

Sun, Y. and Vinogradov, O., 1996a, On Efficiency and Accuracy in Simulations of Granular-type Systems, *Proceedings of the 11th Engineering Mechanics Conference*, ASCE., Fort Lauderdale, FL., Vol. 1, 96-99.

Sun, Y. and Vinogradov, O., 1996b, Jamming of the Flow of Granular Materials, *Proceedings of the 11th Engineering Mechanics Conference*, ASCE., Fort Lauderdale, FL., Vol. 1, 265-268.

Sun, Y. and Vinogradov, O., 1996c, Numerical Investigation of Jaming of Solid Particles in a Straight Pipe, *Proceedings of the 1st International Pipeline Conference*, ASME., Calgary, AB., Vol. 2, 593-601.

Threlfall, D.C., 1978, The Inclusion of Coulomb Friction in Mechanisms Programs with Particular Reference to DRAM, *Mechanism and Machine Theory*, 13, 475-483.

Urroz, G.E. and Ettema, R., 1994, Small-Scale Experiments on Ice-Jam Initiation in Curved Channel, *Canadian Journal of Civil Engineering*, 21, 719-727.

Vinogradov, O.G., 1985, Simulation Methodology for a Flow of Interactive Ice Floes around an Obstacle, *International Journal of Modelling & Simulation.* Vol. 7, No.1, 28-31.

Vinogradov, O.G. and Springer, A., 1990, Simulation of Motion of Multi-Body System with Interactions, *Proceedings of the 1990 Summer Computer Simulation Conference*, SCS., Calgary, AB., 51-55.

Vinogradov, O.G., 1992, Explicit Equations of Motion of Interacting spherical Particles, *Recent Advances in Structural Mechanics*, PVP-Vol. 248 (Editors: Kwon, Y.W. & Chung, H.H.), 111-115.

Vinogradov, O.G., 1993a, Explicit Equation of Motion of Discrete System of Disks in Two Dimensions, *Journal of Engineering Mechanics*, Vol. 118, No. 9, 1850-1858.

Vinogradov, O.G., 1993b, Dynamics Equations for System of Irregularly Shaped Plane Bodies, *Journal of Engineering Mechanics*, Vol. 119, No. 11, 2226-2237.

Vinogradov, Oleg and Sun, Yi, 1996, A Multibody Approach in Granular Dynamics Simulations, *International Journal of Computational Mechanics*, Accepted.

Walton, O.R. and Braun, R.L., 1986, Viscosity, Granular-Temperature, and Stress Calculations for Shearing Assemblies of Inelastic, Frictional Disks, *Journal of Rheology*, 30, 949-980.

Walton, O.R., Kim, H., and Rosato, A.D., 1991, Microstructure and Stress Differences in Shearing Flows, Mechanics Computing in 1990s and Beyond. Proc. Engineering Mechanics Div., ASCE., Columbus, Ohio, 20-22.

Wierzba, P., 1991, Dynamics of Multi-body Systems, *Ph.D. Thesis*, Department of Mechanical Engineering, University of Calgary.

Wierzba, P., and Vinogradov, O.G., 1991, Simulation of Topologically Variable Multibody System in Plane Motion, *Proceedings of 1991 European Simulation Multiconference Modelling & Simulation*, Society for Computer Simulation, San Diego, CA, 935-940.

Wittenburg, J., 1977, *Dynamics of System of Rigid Bodies*, Stuttgart: B.G.Teubner, 80-107.

Zwillinger, D., 1996, *Standard Mathematical Tables and Formulae*, (CRC Press) 30th Edition, 468-469.

Appendices

Appendix A

On the Identity of Lagrange and Newton-Euler Equations for the System with a Tree-like Topology

From the Equations (3.7) and (3.8), we have

$$A_{ij} = U_{ij}\cos(\theta_i - \theta_j) \tag{A.1}$$

$$B_{ij} = U_{ij}\sin(\theta_i - \theta_j) \tag{A.2}$$

where

$$\boldsymbol{U} = \boldsymbol{P}\boldsymbol{M}\boldsymbol{P}^T \tag{A.3}$$

these equations are similar to the Equations (3.35)~(3.37). Now let Equation (3.31) be multiplied by matrix ***B*** and Equation (3.32) be multiplied by matrix ***A*** which gives

$$\boldsymbol{B}\boldsymbol{L}\{\dot{\theta}_i^2\} = \boldsymbol{B}\boldsymbol{H}\{\gamma_i\} + \boldsymbol{B}\{F_i^n\} \tag{A.4}$$

$$\boldsymbol{A}\boldsymbol{L}\{\ddot{\theta}_i\} = \boldsymbol{A}\boldsymbol{K}\{\gamma_i\} - \boldsymbol{A}\{F_i^\tau\} \tag{A.5}$$

Add Equation (A.4) and Equation (A.5), then LHS of the result is equal to the LHS of Equations (3.6). In the RHS, the γ disappears because of $\boldsymbol{BH} + \boldsymbol{AK} = \boldsymbol{0}$. This can be proved by the component form as follows:

First of all, let us use the simple notations

$$\begin{aligned} c_i &= \cos\theta_i \\ s_i &= \sin\theta_i \end{aligned} \qquad \text{(A.6)}$$

so we have

$$\begin{aligned} \cos(\theta_i - \theta_j) &= (c_i c_j + s_i s_j) \\ \sin(\theta_i - \theta_j) &= (s_i c_j - c_i s_j) \end{aligned} \qquad \text{(A.7)}$$

Substitute Equations (A.7) into Equations (A.1), (A.2) and (3.36),(3.37), gives

$$[BH]_{ij} = \sum_{k=1}^{n} B_{ik} H_{kj} = \sum_{k=1}^{n} (c_i c_k s_k c_j + s_i s_k s_k c_j - c_i c_k c_k s_j - s_i s_k c_k s_j) U_{ik} V_{kj} \qquad \text{(A.8)}$$

$$[AK]_{ij} = \sum_{k=1}^{n} A_{ik} K_{kj} = \sum_{k=1}^{n} (s_i c_k s_k c_j - c_i s_k c_k c_j + s_i c_k s_k s_j - c_i s_k s_k s_j) U_{ik} V_{kj} \qquad \text{(A.9)}$$

Equations (A.8) plus Equations(A.9) and note that $c_k^2 + s_k^2 = 1$, which gives

$$[BH]_{ij} + [AK]_{ij} = \sum_{k=1}^{n} (s_i c_j + c_i s_j) U_{ik} V_{kj} \qquad \text{(A.10)}$$

or

$$[BH]_{ij} + [AK]_{ij} = \sin(\theta_i - \theta_j)\sum_{k=1}^{n} U_{ik} V_{kj} \tag{A.11}$$

and note that ***UV*** = ***I*** from the Equation (2.7), i.e.,

$$\sum_{k=1}^{n} U_{ik} V_{kj} = \begin{cases} 1 & i=j \\ 0 & i \neq k \end{cases} \tag{A.12}$$

and

$$\sin(\theta_i - \theta_j)\begin{cases} =0 & i=j \\ \neq 0 & i \neq k \end{cases} \tag{A.13}$$

From the Equations (A.12) and (A.13), we can obtain

$$[BH]_{ij} + [AK]_{ij} = \sin(\theta_i - \theta_j)\sum_{k=1}^{n} U_{ik} V_{kj} = 0 \tag{A.14}$$

that is

$$\boldsymbol{BH} + \boldsymbol{AK} = \boldsymbol{0} \tag{A.15}$$

Moreover, according to the Equations (3.33), (3.34) and (3.7), (3.8), through the simple operations of matrices and note that $\boldsymbol{C}^2 + \boldsymbol{S}^2 = \boldsymbol{I}$, which gives

$$\boldsymbol{B}\{F_i^n\} - \boldsymbol{A}\{F_i^\tau\} = -\boldsymbol{S}\boldsymbol{P}\{f_{xi}\} + \boldsymbol{C}\boldsymbol{P}\{f_{yi}\} = \{Q_i\} \tag{A.16}$$

So far, we have proved that the Newton-Euler Equations can be converted into the Lagrange Equations when ***EP*** = -***I***. Similarly, the Lagrange Equations can also be converted into the Newton-Euler Equations if parameters γ are introduced. Therefore, the two equations are identical for a system with a tree-like topology.

Appendix B

Derivation of the Newton-Euler Equations for the Irregular Body System

Let us consider a sub-system of two bodies connected with a link, which is shown in Figure (B.1).

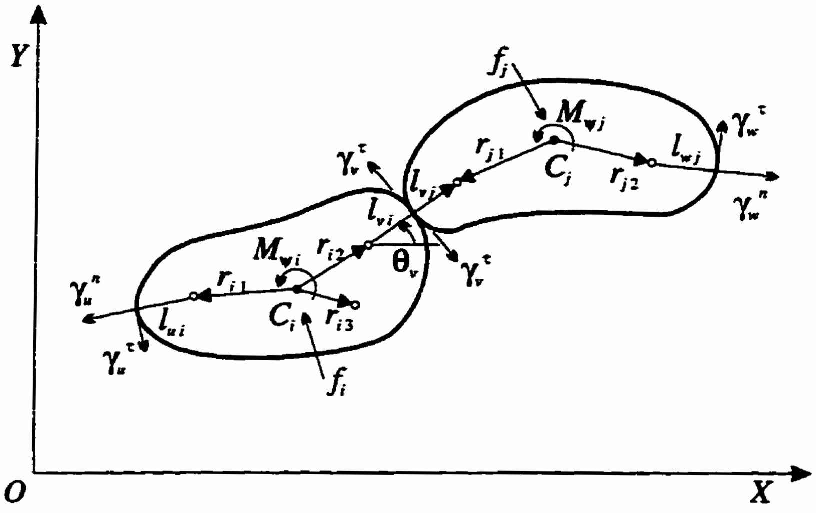


Figure B.1. A planar system with two linked rigid bodies

Note that the direction angle of vector r is ψ, which is not shown in the Figure. Its signs is positive if it goes anti-clockwise from the x to r direction; negative if it goes clockwise from x to r. There is constant difference for all ψ in the same body.

In this case, the internal force γ is divided into two parts, the length of link is also divided into two parts, as shown in the Figure. Moreover, f is the external force and M_ψ is the external moment.

Apply the rigid body dynamics equations to body i and j, we have

$$\begin{aligned} I_i\ddot{\psi}_i &= \gamma_u^n r_{i1}\sin(\psi_{i1}-\theta_u)-\gamma_v^n r_{i2}\sin(\psi_{i2}-\theta_v)+\gamma_u^\tau[l_{ui}-r_{i1}\cos(\psi_{i1}-\theta_u)] + \\ &+\gamma_v^n[l_{vi}-r_{i2}\cos(\psi_{i2}-\theta_v)]+M_{\psi i} \end{aligned} \tag{B.1}$$

$$\begin{aligned} I_j\ddot{\psi}_j &= \gamma_v^n r_{j1}\sin(\psi_{j1}-\theta_w)-\gamma_w^n r_{j2}\sin(\psi_{j2}-\theta_w)+\gamma_v^\tau[l_{vj}-r_{j1}\cos(\psi_{j1}-\theta_v)] + \\ &+\gamma_w^n[l_{wj}-r_{j2}\cos(\psi_{j2}-\theta_w)]+M_{\psi j} \end{aligned} \tag{B.2}$$

In addition, according to Newton's law we have the dynamics equations in the normal direction

$$m_i a_{iv}^n = -\gamma_u^n\cos(\theta_v-\theta_u) + \gamma_v^n - \gamma_u^\tau\sin(\theta_v-\theta_u) + f_{iv}^n \tag{B.3}$$

$$m_j a_{jv}^n = -\gamma_v^n + \gamma_w^n\cos(\theta_v-\theta_w) + \gamma_w^\tau\sin(\theta_v-\theta_w) + f_{jv}^n \tag{B.4}$$

in the tangential direction

$$m_i a_{iv}^\tau = \gamma_u^\tau\sin(\theta_v-\theta_u) - \gamma_u^\tau\cos(\theta_v-\theta_u) + \gamma_v^\tau + f_{iv}^\tau \tag{B.5}$$

$$m_j a_{jv}^\tau = -\gamma_w^\tau\sin(\theta_v-\theta_w) - \gamma_w^\tau\cos(\theta_v-\theta_w) - \gamma_v^\tau + f_{jv}^\tau \tag{B.6}$$

From the Equations (B.3)~(B.6), we have

$$\begin{aligned} a_{jv}^{n} - a_{iv}^{n} = {} & \frac{\gamma_u^n}{m_i}\cos(\theta_v - \theta_u) - (\frac{1}{m_i} + \frac{1}{m_j})\gamma_v^n + \frac{\gamma_w^n}{m_j}\cos(\theta_v - \theta_w) + \\ & + \frac{\gamma_u^\tau}{m_i}\sin(\theta_v - \theta_u) + \frac{\gamma_w^\tau}{m_j}\sin(\theta_v - \theta_w) + (\frac{f_{jv}^n}{m_j} - \frac{f_{iv}^n}{m_i}) \end{aligned} \tag{B.7}$$

$$\begin{aligned} a_{jv}^{\tau} - a_{iv}^{\tau} = {} & \frac{\gamma_u^\tau}{m_i}\cos(\theta_v - \theta_u) - (\frac{1}{m_i} + \frac{1}{m_j})\gamma_v^\tau + \frac{\gamma_w^\tau}{m_j}\cos(\theta_v - \theta_w) - \\ & - \frac{\gamma_u^n}{m_i}\sin(\theta_v - \theta_u) - \frac{\gamma_w^n}{m_j}\sin(\theta_v - \theta_w) + (\frac{f_{jv}^\tau}{m_j} - \frac{f_{iv}^\tau}{m_i}) \end{aligned} \tag{B.8}$$

$$\begin{aligned} a_{jv}^{n} - a_{iv}^{n} = {} & -l_v\dot{\theta}_v^2 + \ddot{\psi}_{i2} r_{i2}\sin(\theta_v - \psi_{i2}) - \dot{\psi}_{i2}^2 r_{i2}\cos(\theta_v - \psi_{i2}) - \\ & - \ddot{\psi}_{j1} r_{j1}\sin(\theta_v - \psi_{j1}) + \dot{\psi}_{j1}^2 r_{j1}\cos(\theta_v - \psi_{j1}) \end{aligned} \tag{B.9}$$

$$\begin{aligned} a_{jv}^{\tau} - a_{iv}^{\tau} = {} & l_v\ddot{\theta}_v + \ddot{\psi}_{i2} r_{i2}\cos(\theta_v - \psi_{i2}) + \dot{\psi}_{i2}^2 r_{i2}\sin(\theta_v - \psi_{i2}) - \\ & - \ddot{\psi}_{j1} r_{j1}\cos(\theta_v - \psi_{j1}) - \dot{\psi}_{j1}^2 r_{j1}\sin(\theta_v - \psi_{j1}) \end{aligned} \tag{B.10}$$

So the RHS of Equation (B.7) is equal to the RHS of Equation (B.9), Equation (B.8) is equal to the RHS of Equation (B.10) and by applying the inductive method we can obtain the matrix form of these equations

$$\begin{aligned} & -(\boldsymbol{CVC} + \boldsymbol{SVS})\{\gamma_i^n\} - (\boldsymbol{SVC} - \boldsymbol{CVS})\{\gamma_i^\tau\} + \{F_i^{\,n}\} = \\ & = -\boldsymbol{L}\{\dot{\theta}_i^2\} - (\boldsymbol{SC}_r^{\,T} - \boldsymbol{CS}_r^{\,T})\{\ddot{\psi}_j\} - (\boldsymbol{CC}_r^{\,T} + \boldsymbol{SS}_r^{\,T})\{\dot{\psi}_j^2\} \end{aligned} \tag{B.11}$$

$$\begin{aligned} & (\boldsymbol{SVC} - \boldsymbol{CVS})\{\gamma_i^n\} - (\boldsymbol{CVC} + \boldsymbol{SVS})\{\gamma^\tau\} - \{F_i^\tau\} = \\ & = \boldsymbol{L}\{\ddot{\theta}_i\} + (\boldsymbol{CC}_r^{\,T} + \boldsymbol{SS}_r^{\,T})\{\ddot{\psi}_j\} - (\boldsymbol{SC}_r^{\,T} - \boldsymbol{CS}_r^{\,T})\{\dot{\psi}_j^2\} \end{aligned} \tag{B.12}$$

From the Equation (B.1) or the Equation (B.2), we have

$$\boldsymbol{I}_{\psi}\{\ddot{\psi}_j\} = -(\boldsymbol{S}_r\boldsymbol{C}-\boldsymbol{C}_r\boldsymbol{S})\{\gamma_i^n\}+[\boldsymbol{L}_r+(\boldsymbol{C}_r\boldsymbol{C}+\boldsymbol{S}_r\boldsymbol{S})]\{\gamma_i^t\}+\{\boldsymbol{M}_{\psi j}\} \qquad \text{(B.13)}$$

Making the further simplification to the Equations (B.11)~(B.13), Equations (3.61)~(3.63) can be obtained.

Appendix C

Derivation of the Newton-Euler Equations in a 3-D System

Let us consider a sub-system of two bodies (mass-points) connected with a link, which is shown in the Figure (C.1).

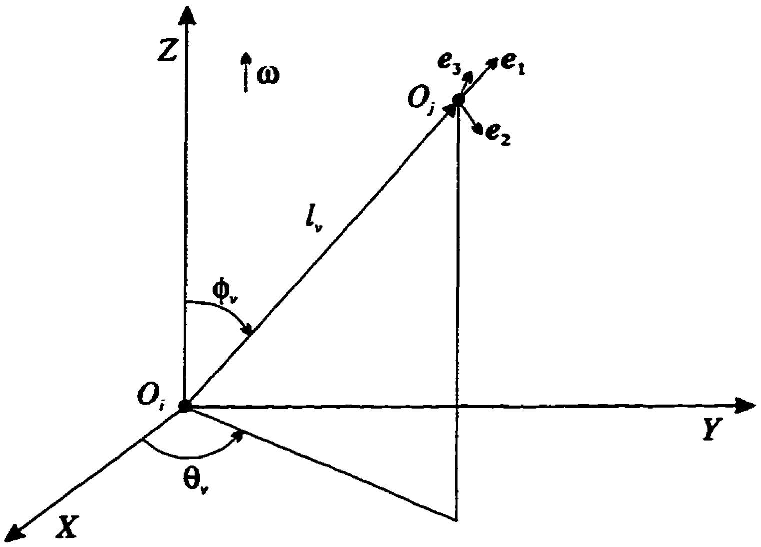


Figure C.1 A spatial system with two connected particles

Where a fixed reference frame (X, Y, Z) and a moving frame (e_1, e_2, e_3) are employed. If the acceleration of point-mass i is known, the acceleration of the point-mass j can be determined by

$$\boldsymbol{a}_j = \boldsymbol{a}_i + \dot{\omega}\times\boldsymbol{r} + 2\omega\times\boldsymbol{v} + \omega\times\omega\times\boldsymbol{r} \tag{C.1}$$

where

$$\omega = \dot{\theta}\cos\phi\ \boldsymbol{e}_1 - \dot{\theta}\sin\phi\ \boldsymbol{e}_2 + \dot{\phi}\ \boldsymbol{e}_3 \tag{C.2}$$

$$\dot{\omega} = (\ddot{\theta}\cos\phi - \dot{\theta}\dot{\phi}\sin\phi)\boldsymbol{e}_1 - (\ddot{\theta}\sin\phi + \dot{\theta}\dot{\phi}\cos\phi)\boldsymbol{e}_2 + \ddot{\phi}\ \boldsymbol{e}_3 \tag{C.3}$$

$$\boldsymbol{v} = \boldsymbol{0} \tag{C.4}$$

$$\boldsymbol{r} = l\boldsymbol{e}_1 \tag{C.5}$$

Therefore, we have

$$\begin{aligned} \boldsymbol{a}_j - \boldsymbol{a}_i &= \dot{\omega}\times\boldsymbol{r} + 2\omega\times\boldsymbol{v} + \omega\times\omega\times\boldsymbol{r} = \\ &= -(l\dot{\phi}^2 + l\dot{\theta}^2\sin^2\phi)\boldsymbol{e}_1 + (l\ddot{\phi} - l\dot{\theta}^2\sin\phi\cos\phi)\boldsymbol{e}_2 + \\ &\quad + (l\ddot{\phi}\sin\phi + 2l\dot{\phi}\dot{\theta}\cos\phi)\boldsymbol{e}_3 \end{aligned} \tag{C.6}$$

From Equation (C.6), we can obtain the relative acceleration between point-mass i and j on the 3 directions ($\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$) as following

$$a_j^n - a_i^n = -(l\dot{\phi}^2 + l\dot{\theta}^2\sin^2\phi) \tag{C.7}$$

$$a_j^\phi - a_i^\phi = l\ddot{\phi} - l\dot{\theta}^2\sin\phi\cos\phi \tag{C.8}$$

$$a_j^\theta - a_i^\theta = l\ddot{\phi}\sin\phi + 2l\dot{\phi}\dot{\theta}\cos\phi \tag{C.9}$$

In addition, we can obtain the following equations by Newton's Second Law in the 3 directions:

(1) In the direction of $\boldsymbol{e}_1$, the movement of the point-mass i and j obey

$$m_i a_{iv}^n = -\gamma_u[\cos\phi_u \cos\phi_v + \sin\phi_v \cos(\theta_v - \theta_u)\sin\phi_u] + \gamma_v + f_{iv}^n \qquad \text{(C.10)}$$

$$m_j a_{jv}^n = -\gamma_v + \gamma_w[\cos\phi_v \cos\phi_w + \sin\phi_v \cos(\theta_v - \theta_w)\sin\phi_w] + f_{jv}^n \qquad \text{(C.11)}$$

(2) In the direction of $\boldsymbol{e}_2$, the movement of the point-mass i and j obey

$$m_i a_{iv}^\phi = \gamma_u[\cos\phi_u \sin\phi_v - \cos\phi_v \cos(\theta_v - \theta_u)\sin\phi_u] + f_{iv}^\phi \qquad \text{(C.12)}$$

$$m_j a_{jv}^\phi = \gamma_w[-\cos\phi_w \sin\phi_v + \cos\phi_v \cos(\theta_v - \theta_w)\sin\phi_w] + f_{vj}^\phi \qquad \text{(C.13)}$$

(3) In the direction of $\boldsymbol{e}_3$, the movement of the point-mass i and j obey

$$m_i a_{iv}^\theta = \gamma_u \sin(\theta_v - \theta_u)\sin\phi_u + f_{iv}^\theta \qquad \text{(C.14)}$$

$$m_j a_{jv}^\theta = -\gamma_w \sin(\theta_v - \theta_w)\sin\phi_w + f_{jv}^\theta \qquad \text{(C.15)}$$

where f_{iv}^n is the component of external force acting on the point-mass i in the direction of $\boldsymbol{e}_1$, f_{iv}^ϕ is the component of external force in the direction of $\boldsymbol{e}_2$ and f_{iv}^θ is the component of external force in the direction $\boldsymbol{e}_3$.

From the Equation (C.10) and (C.11), we can get the expression of a_{jv}^n - a_{iv}^n, then substitute it into the Equation (C.7), which gives

$$l\dot{\phi}^2 + l\dot{\theta}^2\sin^2\phi = -\frac{\gamma_u}{m_i}\cos\phi_v\cos\phi_u - (\frac{1}{m_i}+\frac{1}{m_j})\gamma_v - \frac{\gamma_w^n}{m_j}\cos\phi_v\cos\phi_w - \tag{C.16}$$
$$-\frac{\gamma_u}{m_i}\sin\phi_v\cos(\theta_v-\theta_u)\sin\phi_u - \frac{\gamma_w}{m_j}\sin\phi_v\cos(\theta_v-\theta_w)\sin\phi_w + (\frac{f_{iv}^n}{m_i} - \frac{f_{jv}^n}{m_j})$$

Applying the inductive method we can easily get its matrix form

$$\boldsymbol{L}(\{\dot{\phi}_i^2\} + \boldsymbol{S}_\phi^2\{\dot{\theta}_i^2\}) = [\boldsymbol{C}_\phi \boldsymbol{V} \boldsymbol{C}_\phi + \boldsymbol{S}_\phi(\boldsymbol{CVC}+\boldsymbol{SVS})\boldsymbol{S}_\phi]\{\gamma_i\} + \{F_i^n\} \tag{C.17}$$

Here Equation (C.17) is Equation (3.85). Similarly, from Equations (C.12), (C.13) and (C.8), we can get Equation (3.86); From Equations (C.14), (C.15) and (C.9), we can get Equation (3.87).

Appendix D

Estimation of the Maximum Coordination Number in a Multisphere System

Let us consider a sub-system of two bodies (mass-points) connected with a link, which is shown in the Figure (D.1).

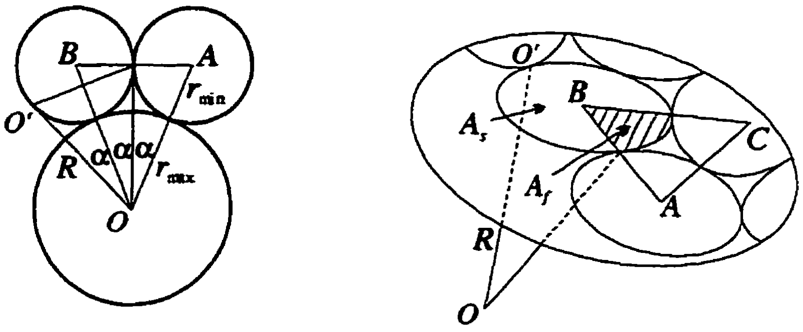


Figure D.1 Illustration of a multisphere system

When a sphere with a radius $r_{\max}$ is surrounded by spheres with radii $r_{\min}$ (see the left hand side of the figure), a spherical surface with radius R will form (see the right hand side of the figure), where

$$R = OO' = (r_{\max} + r_{\min})\cos\alpha \tag{D.1}$$

and α can be determined by

$$\sin\alpha = \frac{r_{\min}}{r_{\max} + r_{\min}} \tag{D.2}$$

On this spherical surface, according to the relevant spherical geometry principle, each circle area A_s is

$$A_s = 2\pi R^2(1 - \cos\alpha) \tag{D.3}$$

and the area of the spherical triangle A_t is

$$A_t = ER^2 \tag{D.4}$$

where E is the spherical excess of the triangle i.e.

$$E = \angle A + \angle B + \angle C - \pi \tag{D.5}$$

Assuming there are N_t triangles in the whole spherical surface ($4\pi R^2$), i.e.

$$N_t A_t = 4\pi R^2 \tag{D.6}$$

or

$$N_t = \frac{4\pi}{E} \tag{D.7}$$

The relationship of A_f and A_s is

$$A_f = \frac{\angle B}{2\pi} A_s = \frac{\frac{\pi + E}{3}}{2\pi} A_s \tag{D.8}$$

where the sum of 3 angles is $\pi + E$, and since these angles are same, each of them is $(\pi + E) / 3$. Now let us calculate the percentage of the all spherical area A_f in the area of triangle A_t

$$\frac{3A_f}{A_t} = \frac{N_s}{N_t} \tag{D.9}$$

where N_s is the number of spheres with radii $r_{\min}$. From the Equations (D.8) and (D.9) we can obtain

$$N_s = \frac{\pi + E}{2\pi}\frac{4\pi}{E} = \frac{2(\pi + E)}{E} \tag{D.10}$$

In the *Standard Mathematical Tables and Formulae* (Zwillinger, 1996), E has been given in the form

$$\tan\frac{1}{4}E = \sqrt{\tan\frac{s}{2}\tan\frac{1}{2}(s-a)\tan\frac{1}{2}(s-b)\tan\frac{1}{2}(s-c)} \tag{D.11}$$

where a, b, c represent the sides of the triangle (in radian measure), and

$$s = \frac{a+b+c}{2} \tag{D.12}$$

In our problem, $a = b = c = 2\alpha$, hence $s = 3\alpha$, hence the Equation (D.11) is simplified as

$$\tan\frac{1}{4}E = \sqrt{\tan\frac{3\alpha}{2}\tan^3\frac{\alpha}{2}} \tag{D.13}$$

So the Equations (4.5) and (4.6) have been proved.

Note that the estimation of N_s can be obtained by different ways. This is only a simple form.

Appendix E

A FORTRAN Program for Simulations of a Granular System

```
C*********************************************************************
C     MAIN PROGRAM
C--------------------------------------------------------------------
C     Using the mixed iterative method to solve a constrained multibody
C     dynamics system (jamming problem of granular two phase flow in a
C     straight pipe)
C--------------------------------------------------------------------
      PARAMETER (N=5000,M=3000,K=10,KC=80,DEN=1.1)
      CHARACTER FILE*8,INPF*8,OUTF*8,WD*4
      DIMENSION XRB(M),XB(M),YB(M),XDB(M),YDB(M)
      DIMENSION DF(N),DL(N),DX(M),DY(M),DV(N),DFV(N)
      DIMENSION XDD(M),YDD(M),FX(M),FY(M),F(N),FT(N),B(N),RX(M),RY(M)
      DIMENSION X1(M),Y1(M),XD1(M),YD1(M),XDD1(M),YDD1(M),FX1(M),FY1(M)
      COMMON /TOPO/ IB(N),IE(N),IBB(N,K),IEE(N,K),KB(N),KE(N)
      COMMON /PARA1/ XCR(KC),XC(KC),YC(KC),XR(M),X(M),Y(M),XD(M),YD(M)
      COMMON /PARA2/ C(N),S(N),XM(M),XL(N)
C--------------------------------------------------------------------
C     Open a data file (.inp) and read data from the file
C--------------------------------------------------------------------
      GOTO 6
 5    PRINT*,"File not found"
 6    PRINT*,"Enter name of data file (no extension) :"
      READ(*,"(A8)") FILE
      DO 8 I=1,8
        IF (FILE(I:I).EQ." ") THEN
          NCH=I-1
          GOTO 9
        ENDIF
 8    CONTINUE
 9    INPF=FILE(1:NCH)//".inp"
      OUTF=FILE(1:NCH)//".plt"
      OPEN (1,FILE=INPF,STATUS="OLD",ERR=5)
      OPEN (2,FILE=OUTF)
C
 10   READ(1,"(A4)") WD
      IF (WD.EQ."REM") GOTO 10
      IF (WD.EQ."PARA") THEN
 11     READ(1,"(A4)") WD
        IF (WD.EQ."END") GOTO 10
        BACKSPACE 1
        READ (1,*) X0,XEND,XIN,DH,VX,VY,TH,FMAX,E,U,TOL
        GOTO 11
      ELSE IF (WD.EQ."RAND") THEN
 12     READ(1,"(A4)") WD
        IF (WD.EQ."END") GOTO 10
        BACKSPACE 1
        READ (1,*) IG,ISEED,RI,XI,YI
        GOTO 12
      ELSE IF (WD.EQ."DISK") THEN
        NBB=0
 13     READ (1,"(A4)") WD
        IF (WD.EQ."END") GOTO 10
        BACKSPACE 1
        READ (1,*) I,XRB(I),XB(I),YB(I),XDB(I),YDB(I)
        NBB=NBB+1
        GOTO 13
      ELSE IF (WD.EQ."ROOT") THEN
        NC=0
 14     READ (1,"(A4)") WD
        IF (WD.EQ."END") GOTO 15
        BACKSPACE 1
        READ (1,*) I,XCR(I),XC(I),YC(I)
        NC=NC+1
        GOTO 14
      ENDIF
 15   CONTINUE
C-----------------------------------------------------------------------
C     Initialize parameters
C-----------------------------------------------------------------------
      PI=4.0*ATAN(1.0)
      NB=0
      KG=1
      L=0
      XT=X0
      H=50.0*DH
C---------------------------------------------------------------------
C     Conditional Randomization of Disks
C---------------------------------------------------------------------
 16   IF (IG.EQ.1) THEN
      DO 17 I=1,NBB
        XB(I)=XB(I)-0.5*XI+XI*RAN(ISEED)
        YB(I)=YB(I)-0.5*YI+YI*RAN(ISEED)
        XRB(I)=XRB(I)-0.5*RI+RI*RAN(ISEED)
 17   CONTINUE
      ENDIF
      P0=0.0
      DO 18 I=1,NBB
        P0=P0+PI*XRB(I)*XRB(I)
 18   CONTINUE
      P0=P0/2000.
      PRINT*, P0
C---------------------------------------------------------------------
      DO 19 I=1,NBB
        XR(NB+I)=XRB(I)
        X(NB+I)=XB(I)
        Y(NB+I)=YB(I)
        XD(NB+I)=XDB(I)
        YD(NB+I)=YDB(I)
        XM(NB+I)=1.333333*DEN*PI*XR(NB+I)**3
 19     CONTINUE
      NB=NB+NBB
C
      DO 20 I=1,NB
          X1(I)=X(I)
          Y1(I)=Y(I)
          XD1(I)=XD(I)
          YD1(I)=YD(I)
 20   CONTINUE
C---------------------------------------------------------------------
C     Checking disks overlap
C---------------------------------------------------------------------
      CALL PEVENT(NC,NB,NL,IFLAG,IB,IE,XL,DL)
c      IF (IFLAG.EQ.1) THEN
c      PRINT*, "           Disks overlap !"
```

```
c      PRINT*, "    Please check initial position !"
c      PRINT*, "Overlap happened between disk ",IC,"and ",J
c      PRINT*, SS
c      ENDIF
C-------------------------------------------------------------
C    Initialize topology data
C-------------------------------------------------------------
      CALL TOPOLOGY(NL)
      CALL DRAG(NB,NL,VX,VY,FX,FY)
C
C-------------------------------------------------------------
C    Checking initial velocity violation if links exist
C-------------------------------------------------------------
      IF (IG.EQ.1) GOTO 22
      DO 21 I=1,NL
        IF (IB(I).LT.0) THEN
          DV(I)=XD(IE(I))*C(I)+YD(IE(I))*S(I)
        ELSE
          DV(I)=(XD(IE(I))-XD(IB(I)))*C(I)+(YD(IE(I))-YD(IB(I)))*S(I)
        ENDIF
        IF (ABS(DV(I)).GT.0.1) THEN
          PRINT*, "      Initial velocity violation !"
          PRINT*, "     Please check initial velocity !"
          PRINT*, "Velocity violation between disks ",IB(I),"and ",IE(I)
          PRINT*, "Overlap = ",DV(I)
        ENDIF
 21   CONTINUE
C
C-------------------------------------------------------------
C    Considering external force on each body and composing vector B
C-------------------------------------------------------------
C
 22   DO 23 I=1,NL
          IF (IB(I).LT.0) THEN
            B(I)=(XD(IE(I))**2+YD(IE(I))**2)/XL(I)+
     &                C(I)*FX(IE(I))/XM(IE(I))+S(I)*FY(IE(I))/XM(IE(I))
          ELSE
      B(I)=((XD(IE(I))-XD(IB(I)))**2+(YD(IE(I))-YD(IB(I)))**2)/XL(I)+
     &      (FX(IE(I))/XM(IE(I))-FX(IB(I))/XM(IB(I)))*C(I)+
     &      (FY(IE(I))/XM(IE(I))-FY(IB(I))/XM(IB(I)))*S(I)
          ENDIF
```

```
  23   CONTINUE
C--------------------------------------------------------------------
C     Solving initial reaction force in each link by iterative method
C--------------------------------------------------------------------
  25   NT=0
       CALL ITERAT(NL,B,F)
       W=1.5
       DO 26 I=1,NL
          F(I)=W*F(I)+(1.0-W)*FT(I)
          IF (ABS(F(I)).LT.TOL) GOTO 26
          ERR=ABS((F(I)-FT(I))/F(I))
          IF (ERR.GT.TOL) NT=1
          FT(I)=F(I)
  26   CONTINUE
       IF (NT.EQ.1) GOTO 25
C
C--------------------------------------------------------------------
C     Computing initial accelerations for each body
C--------------------------------------------------------------------
       CALL MULT(NB,NL,F,XDD1,YDD1)
C
  30   L=0
  31   CONTINUE
       CALL DRAG(NB,NL,VX,VY,FX1,FY1)
       NT=0
C
       L=L+1
       DO 32 I=1,NL
          IF (IB(I).LT.0) THEN
              B(I)=(XD(IE(I))**2+YD(IE(I))**2)/XL(I)+
     &        C(I)*FX1(IE(I))/XM(IE(I))+S(I)*FY1(IE(I))/XM(IE(I))
          ELSE
          B(I)=((XD(IE(I))-XD(IB(I)))**2+(YD(IE(I))-YD(IB(I)))**2)/XL(I)
     &          +(FX1(IE(I))/XM(IE(I))-FX1(IB(I))/XM(IB(I)))*C(I)
     &          +(FY1(IE(I))/XM(IE(I))-FY1(IB(I))/XM(IB(I)))*S(I)
          ENDIF
  32   CONTINUE
C--------------------------------------------------------------------
C     Solving reaction force in each link, new position and velocity
C     of each body by iterative method with a variable time step
C--------------------------------------------------------------------
```

```
      CALL ITERAT(NL,B,F)
C
      DO 34 I=1,NL
         AF=ABS(F(I))
         IF ((AF.GT.1.0E22).OR.(L.GT.500)) THEN
         F(I)=0.0
         DO 33 J=1,NB
         X(J)=X1(J)
         Y(J)=Y1(J)
         XD(J)=XD1(J)
         YD(J)=YD1(J)
 33   CONTINUE
         H=H-DH
         IF (H.GT.DH) GOTO 30
         PRINT*, "Please reduce the minimum time step DH"
         PRINT*, L, H, F(I),I
         STOP
      ENDIF
      IF (AF.LT.TOL) GOTO 34
      ERR=ABS((F(I)-FT(I))/F(I))
      IF (ERR.GT.TOL) NT=1
 34   CONTINUE
C
      DO 35 I=1,NL
         IF (IB(I).LT.0) THEN
           DV(I)=XD(IE(I))*C(I)+YD(IE(I))*S(I)
         ELSE
           DV(I)=(XD(IE(I))-XD(IB(I)))*C(I)+(YD(IE(I))-YD(IB(I)))*S(I)
         ENDIF
 35   CONTINUE
C
      DO 37 I=1,NL
         IF ((F(I).GT.FMAX).OR.(ABS(DV(I)).GT.0.1)) THEN
            DO 36 J=I,NL-1
               F(J)=F(J+1)
               FT(J)=FT(J+1)
               XL(J)=XL(J+1)
               IB(J)=IB(J+1)
               IE(J)=IE(J+1)
 36         CONTINUE
            IB(NL)=0
```

```
              IE(NL)=0
              NL=NL-1
              CALL TOPOLOGY(NL)
           ENDIF
 37     CONTINUE
C
        CALL MULT(NB,NL,F,XDD,YDD)
        DO 38 I=1,NB
           XDD(I)=XDD(I)+FX1(I)/XM(I)
           YDD(I)=YDD(I)+FY1(I)/XM(I)
           RX(I)=0.0
           RY(I)=0.0
 38     CONTINUE
C----------------------------------------------------------------------
C     Determining friction forces
C----------------------------------------------------------------------
        DO 39 I=1,NL
          IF (F(I).LT.0.0) THEN
             RX(IE(I))=RX(IE(I))+ABS(F(I)*S(I))*U/XM(IE(I))
             RY(IE(I))=RY(IE(I))+ABS(F(I)*C(I))*U/XM(IE(I))
             IF (IB(I).LT.0.0) THEN
               RX(IE(I))=RX(IE(I))+ABS(F(I)*S(I))*2.0*U/XM(IE(I))
               RY(IE(I))=RY(IE(I))+ABS(F(I)*C(I))*2.0*U/XM(IE(I))
             ELSE
             RX(IB(I))=RX(IB(I))+ABS(F(I)*S(I))*U/XM(IB(I))
             RY(IB(I))=RY(IB(I))+ABS(F(I)*C(I))*U/XM(IB(I))
             ENDIF
          ENDIF
 39     CONTINUE
C
        DO 53 I=1,NL
           IF (IB(I).LT.0) THEN
              IF (XD(IE(I)))41,43,42
 41           XDD(IE(I))=XDD(IE(I))+RX(IE(I))
              GOTO 43
 42           XDD(IE(I))=XDD(IE(I))-RX(IE(I))
 43           CONTINUE
              IF (YD(IE(I))) 44,46,45
 44           YDD(IE(I))=YDD(IE(I))+RY(IE(I))
              GOTO 46
 45           YDD(IE(I))=YDD(IE(I))-RY(IE(I))
 46          CONTINUE
          ELSE
             IF(XD(IE(I))-XD(IB(I))) 47,49,48
 47          XDD(IE(I))=XDD(IE(I))+RX(IE(I))
             XDD(IB(I))=XDD(IB(I))-RX(IB(I))
             GOTO 49
 48          XDD(IE(I))=XDD(IE(I))-RX(IE(I))
             XDD(IB(I))=XDD(IB(I))+RX(IB(I))
 49          CONTINUE
             IF(YD(IE(I))-YD(IB(I))) 50,52,51
 50          YDD(IE(I))=YDD(IE(I))+RY(IE(I))
             YDD(IB(I))=YDD(IB(I))-RY(IB(I))
             GOTO 52
 51          YDD(IE(I))=YDD(IE(I))-RY(IE(I))
             YDD(IB(I))=YDD(IB(I))+RY(IB(I))
 52          CONTINUE
          ENDIF
 53   CONTINUE
C
      DO 54 I=1,NB
        X(I)=X1(I)+H*XD1(I)+H*H*(2.0*XDD1(I)+XDD(I))/6.0
        Y(I)=Y1(I)+H*YD1(I)+H*H*(2.0*YDD1(I)+YDD(I))/6.0
        XD(I)=XD1(I)+0.5*H*(XDD1(I)+XDD(I))
        YD(I)=YD1(I)+0.5*H*(YDD1(I)+YDD(I))
 54   CONTINUE
C---------------------------------------------------------------------
C     Position correction
C---------------------------------------------------------------------
      DO 60 I=1,NL
        IF (IB(I).LT.0) THEN
        DL(I)=SQRT((X(IE(I))-XC(-IB(I)))**2+(Y(IE(I))-YC(-IB(I)))**2)
     &          -XL(I)
        ELSE
        DL(I)=SQRT((X(IE(I))-X(IB(I)))**2+(Y(IE(I))-Y(IB(I)))**2)-XL(I)
        ENDIF
 60   CONTINUE
C
      DO 61 J=1,5
      CALL ITERAT(NL,DL,DF)
 61   CONTINUE
C
```

```
       CALL MULT(NB,NL,DF,DX,DY)
       DO 62 I=1,NB
          X(I)=X(I)+DX(I)
          Y(I)=Y(I)+DY(I)
 62    CONTINUE
C----------------------------------------------------------------------
C     Velocity correction
C----------------------------------------------------------------------
       DO 63 I=1,NL
         IF (IB(I).LT.0) THEN
           DV(I)=XD(IE(I))*C(I)+YD(IE(I))*S(I)
         ELSE
           DV(I)=(XD(IE(I))-XD(IB(I)))*C(I)+(YD(IE(I))-YD(IB(I)))*S(I)
         ENDIF
 63    CONTINUE
C
       DO 64 J=1,5
          CALL ITERAT(NL,DV,DFV)
 64    CONTINUE
C
       CALL MULT(NB,NL,DFV,DX,DY)
C
       DO 65 I=1,NB
          XD(I)=XD(I)+DX(I)
          YD(I)=YD(I)+DY(I)
 65    CONTINUE
C----------------------------------------------------------------------
C     Energy correction
C----------------------------------------------------------------------
       EK=0.0
       EK1=0.0
       DO 66 I=1,NB
          EK=EK+0.5*XM(I)*(XD(I)**2+YD(I)**2)
          EK1=EK1+0.5*XM(I)*(XD1(I)**2+YD1(I)**2)
 66    CONTINUE
C
       CALL DRAG(NB,NL,VX,VY,FX,FY)
C
       DW=0.0
       DO 67 I=1,NB
          DW=DW+0.5*(FX1(I)+FX(I))*(X(I)-X1(I))+
     &          0.5*(FY1(I)+FY(I))*(Y(I)-Y1(I))
 67   CONTINUE
C
      DE=EK-EK1-DW
      SUM=0.0
      DO 68 I=1,NB
         SUM=SUM+XM(I)*(XD(I)**2+YD(I)**2)
 68   CONTINUE
      DO 69 I=1,NB
         XD(I)=XD(I)-XD(I)*DE/SUM
         YD(I)=YD(I)-YD(I)*DE/SUM
 69   CONTINUE
C----------------------------------------------------------------------
      IF (NT.EQ.1.AND.L.LT.200) THEN
        DO 70 I=1,NL
          FT(I)=F(I)
 70     CONTINUE
        GOTO 31
      ENDIF
C----------------------------------------------------------------------
C     Checking and handling of collisions
C----------------------------------------------------------------------
      CALL PEVENT(NC,NB,NL,IFLAG,IB,IE,XL,DL)
      CALL TOPOLOGY(NL)
      DO 71 I=1,NL
        IF (IB(I).LT.0) THEN
          C(I)=(X(IE(I))-XC(-IB(I)))/XL(I)
          S(I)=(Y(IE(I))-YC(-IB(I)))/XL(I)
        ELSE
          C(I)=(X(IE(I))-X(IB(I)))/XL(I)
          S(I)=(Y(IE(I))-Y(IB(I)))/XL(I)
        ENDIF
 71   CONTINUE
C----------------------------------------------------------------------
C     Position Correction
C----------------------------------------------------------------------
C
      DO 72 J=1,5
        CALL ITERAT(NL,DL,DF)
 72   CONTINUE
C
```

```
       CALL MULT(NB,NL,DF,DX,DY)
       DO 73 I=1,NB
         X(I)=X(I)+DX(I)
         Y(I)=Y(I)+DY(I)
 73    CONTINUE
C-----------------------------------------------------------------------
C     Velocity correction and collision handling
C-----------------------------------------------------------------------
       IFV=0
       DO 74 I=1,NL
         IF (IB(I).LT.0) THEN
           DV(I)=XD(IE(I))*C(I)+YD(IE(I))*S(I)
         ELSE
           DV(I)=(XD(IE(I))-XD(IB(I)))*C(I)+(YD(IE(I))-YD(IB(I)))*S(I)
         ENDIF
 74    CONTINUE
C
       DO 75 I=1,NL
          IF (ABS(DV(I)).GT.0.1) THEN
             IF (DV(I).LT.0.0) DV(I)=(1.+E)*DV(I)
             IF (DV(I).GE.0.0) DV(I)=0.0
          ELSE
             DV(I)=0.0
          ENDIF
 75    CONTINUE
C
       DO 76 J=1,5
          CALL ITERAT(NL,DV,DFV)
 76    CONTINUE
C
       CALL MULT(NB,NL,DFV,DX,DY)
C
       DO 77 I=1,NB
          XD(I)=XD(I)+DX(I)
          YD(I)=YD(I)+DY(I)
 77    CONTINUE
C-----------------------------------------------------------------------
       XT=XT+H
C-----------------------------------------------------------------------
C     Determining the next time step
C-----------------------------------------------------------------------
```

```
C
      IF (L.LT.20) H=H+DH
      IF ((L.GT.90).AND.(H.GT.DH)) H=H-DH
C
C----------------------------------------------------------------------
C     Output to screen and a plot file
C----------------------------------------------------------------------
C
      IF (XT.LT.(KT*XIN)) GOTO 91
C
      WRITE (6,90) XT, X(1),Y(1),XD(1),YD(1)
 90   FORMAT(1X,F6.2,2X,4F7.3)
C
      WRITE (2,*) XT,NB
      WRITE (2,*) (X(I),I=1,NB)
      WRITE (2,*) (Y(I),I=1,NB)
      WRITE (2,*) (XR(I),I=1,NB)
      KT=KT+1
C
 91   IF (XT.GT.XEND) GOTO 99
      IF (XT.LT.(50./VX)*KG) GOTO 96
      KG=KG+1
      DO 94 I=1,NB
        IF (X(I).GT.150.) THEN
 92       DO 93 J=I,NB-1
            XM(J)=XM(J+1)
            XR(J)=XR(J+1)
            X(J)=X(J+1)
            Y(J)=Y(J+1)
            XD(J)=XD(J+1)
            YD(J)=YD(J+1)
 93       CONTINUE
          NB=NB-1
          IF (X(I).GT.150.) GOTO 92
        ENDIF
 94   CONTINUE
C----------------------------------------------------------------------
C     Checking Jamming
C----------------------------------------------------------------------
      JAM=0
      DO 95 I=1,NB
```

```
          IF (X(I).LT.-21.) THEN
             PRINT*,P0," ---- JAMMING !!! ----"
             STOP
          ENDIF
 95    CONTINUE
       GOTO 16
C
 96    DO 97 I=1,NB
          X1(I)=X(I)
          Y1(I)=Y(I)
          XD1(I)=XD(I)
          YD1(I)=YD(I)
          XDD1(I)=XDD(I)
          YDD1(I)=YDD(I)
 97    CONTINUE
       GOTO 30
 99    END
C************************************************************************
C     SUBROUTINES
C------------------------------------------------------------------------
C     Subroutine for position events
C------------------------------------------------------------------------
       SUBROUTINE PEVENT(NC,NB,NL,IFLAG,IB,IE,XL,DL)
       PARAMETER (N=5000, M=3000, K=10, KC=80)
       DIMENSION IB(N),IE(N),XL(N),DL(N)
       COMMON /PARA1/ XCR(KC),XC(KC),YC(KC),XR(M),X(M),Y(M),XD(M),YD(M)
       KL=1
       KN=NL
       IFLAG=0
C
       DO 131 I=1,NC
          DO 130 J=1,NB
             SL=XCR(I)+XR(J)
             SS=SQRT((X(J)-XC(I))**2+(Y(J)-YC(I))**2)-SL
             IF (SS.LT.0.0001) THEN
               IB(KL)=-I
               IE(KL)=J
               XL(KL)=SL
               DL(KL)=SS
               KL=KL+1
               IF (SS.LT.-0.001) IFLAG=1
```

```
            ENDIF
 130      CONTINUE
 131    CONTINUE
C
        DO 133 I=1,NB-1
          DO 132 J=I+1,NB
            SL=XR(I)+XR(J)
            SS=SQRT((X(J)-X(I))**2+(Y(J)-Y(I))**2)-SL
            IF (SS.LT.0.0001) THEN
              IB(KL)=I
              IE(KL)=J
              XL(KL)=SL
              DL(KL)=SS
              KL=KL+1
              IF (SS.LT.-0.001) IFLAG=1
            ENDIF
 132      CONTINUE
 133    CONTINUE
        NL=KL-1
        IF (NL.NE.KN) IFLAG=1
 140    RETURN
        END
C-----------------------------------------------------------------------
C     Subroutine for setting up topology data structure
C-----------------------------------------------------------------------
        SUBROUTINE TOPOLOGY(NL)
        PARAMETER (N=5000,K=10)
        COMMON /TOPO/ IB(N),IE(N),IBB(N,K),IEE(N,K),KB(N),KE(N)
C
        IF (NL.EQ.0) RETURN
        DO 220 I=1,NL
          KB(I)=0
          KE(I)=0
          DO 210 J=1,NL
            IF(IB(I).EQ.IB(J)) THEN
              IF(I.NE.J) THEN
                KB(I)=KB(I)+1
                IBB(I,KB(I))=-J
              ENDIF
            ENDIF
            IF(IB(I).EQ.IE(J)) THEN
```

```
              IF(I.NE.J) THEN
                 KB(I)=KB(I)+1
                 IBB(I,KB(I))=J
              ENDIF
            ENDIF
C
            IF(IE(I).EQ.IE(J)) THEN
              IF(I.NE.J) THEN
                 KE(I)=KE(I)+1
                 IEE(I,KE(I))=-J
              ENDIF
            ENDIF
            IF(IE(I).EQ.IB(J)) THEN
              IF(I.NE.J) THEN
                KE(I)=KE(I)+1
                IEE(I,KE(I))=J
              ENDIF
            ENDIF
 210     CONTINUE
 220  CONTINUE
      END
C---------------------------------------------------------------------
C    Subroutine for iterative scheme based on topology data structure
C---------------------------------------------------------------------
      SUBROUTINE ITERAT(NL,B,Z)
      PARAMETER (N=5000,M=3000,K=10)
      DIMENSION B(N),Z(N)
      COMMON /TOPO/ IB(N),IE(N),IBB(N,K),IEE(N,K),KB(N),KE(N)
      COMMON /PARA2/ C(N),S(N),XM(M),XL(N)
      DO 340 KK=1,10
        DO 330 I=1,NL
          TMP=B(I)
          IF (IB(I).LT.0) GOTO 310
          DO 300 J=1,KB(I)
            IF (IBB(I,J).LT.0) THEN
              IJ=-IBB(I,J)
              TMP=TMP-Z(IJ)*(C(I)*C(IJ)+S(I)*S(IJ))/XM(IB(I))
            ELSE
              IJ=IBB(I,J)
              TMP=TMP+Z(IJ)*(C(I)*C(IJ)+S(I)*S(IJ))/XM(IB(I))
            ENDIF
 300       CONTINUE
 310       DO 320 J=1,KE(I)
              IF (IEE(I,J).LT.0) THEN
                IJ=-IEE(I,J)
                TMP=TMP-Z(IJ)*(C(I)*C(IJ)+S(I)*S(IJ))/XM(IE(I))
              ELSE
                IJ=IEE(I,J)
                TMP=TMP+Z(IJ)*(C(I)*C(IJ)+S(I)*S(IJ))/XM(IE(I))
              ENDIF
 320       CONTINUE
C
           IF (IB(I).LT.0) THEN
             Z(I)=XM(IE(I))*TMP
           ELSE
             Z(I)=TMP*XM(IB(I))*XM(IE(I))/(XM(IB(I))+XM(IE(I)))
           ENDIF
 330     CONTINUE
 340   CONTINUE
       RETURN
       END
C----------------------------------------------------------------------
C     Subroutine for calculating X=M^(-1)*E*C*F and Y=M^(-1)*E*S*F
C----------------------------------------------------------------------
       SUBROUTINE MULT(NB,NL,F,X,Y)
       PARAMETER (N=5000,M=3000,K=10)
       DIMENSION F(N),X(M),Y(M)
       COMMON /TOPO/ IB(N),IE(N),IBB(N,K),IEE(N,K),KB(N),KE(N)
       COMMON /PARA2/ C(N),S(N),XM(M),XL(N)
       DO 400 I=1,NB
          X(I)=0.0
          Y(I)=0.0
 400   CONTINUE
       DO 420 I=1,NL
          IF (IB(I).LT.0) GOTO 410
          X(IB(I))=X(IB(I))+F(I)*C(I)/XM(IB(I))
          Y(IB(I))=Y(IB(I))+F(I)*S(I)/XM(IB(I))
 410      CONTINUE
          X(IE(I))=X(IE(I))-F(I)*C(I)/XM(IE(I))
          Y(IE(I))=Y(IE(I))-F(I)*S(I)/XM(IE(I))
 420   CONTINUE
       RETURN
```

```
      END
C-----------------------------------------------------------------------
C     Subroutine for calculating drag force on each body
C-----------------------------------------------------------------------
      SUBROUTINE DRAG(NB,NL,VX,VY,FX,FY)
      PARAMETER (N=5000,M=3000,K=10,KC=80,CD=0.05,RHO=1.0)
      DIMENSION FX(M),FY(M),DKX(M),DKY(M)
      COMMON /TOPO/ IB(N),IE(N),IBB(N,K),IEE(N,K),KB(N),KE(N)
      COMMON /PARA1/ XCR(KC),XC(KC),YC(KC),XR(M),X(M),Y(M),XD(M),YD(M)
      COMMON /PARA2/ C(N),S(N),XM(M),XL(N)
C
      PI=4.0*ATAN(1.0)
      DO 500 I=1,NL
        IF (IB(I).LT.0) THEN
          C(I)=(X(IE(I))-XC(-IB(I)))/XL(I)
          S(I)=(Y(IE(I))-YC(-IB(I)))/XL(I)
        ELSE
          C(I)=(X(IE(I))-X(IB(I)))/XL(I)
          S(I)=(Y(IE(I))-Y(IB(I)))/XL(I)
        ENDIF
 500  CONTINUE
C
      DO 510 I=1,NB
         DKX(I)=1.0
         DKY(I)=1.0
 510  CONTINUE
C
      DO 520 I=1,NL
         IF (IB(I).LT.0.0) THEN
            DKX(IE(I))=ABS(S(I))
            DKY(IE(I))=ABS(C(I))
            GOTO 520
         ENDIF
C
         IF (((VX-XD(IB(I)))*C(I)).LT.0.0) THEN
             DKX(IB(I))=0.5*(1.0+ABS(S(I)))
     &         -0.5*(1.0-ABS(S(I)))*XR(IE(I))/XR(IB(I))
         ENDIF
         IF (((VX-XD(IE(I)))*C(I)).GT.0.0) THEN
            DKX(IE(I))=0.5*(1.0+ABS(S(I)))
     &         -0.5*(1.0-ABS(S(I)))*XR(IB(I))/XR(IE(I))
```

```
          ENDIF
          IF (((VY-YD(IB(I)))*S(I)).LT.0.0) THEN
             DKY(IB(I))=0.5*(1.0+ABS(C(I)))
     &         -0.5*(1.0-ABS(C(I)))*XR(IE(I))/XR(IB(I))
          ENDIF
          IF (((VY-YD(IE(I)))*S(I)).GT.0.0) THEN
             DKY(IE(I))=0.5*(1.0+ABS(C(I)))
     &         -0.5*(1.0-ABS(C(I)))*XR(IB(I))/XR(IE(I))
          ENDIF
 520   CONTINUE
       DO 530 I=1,NB
          IF(DKX(I).LT.0.1) DKX(I)=0.1
          IF(DKY(I).LT.0.1) DKY(I)=0.1
          VR=VX
          IF (X(I).GT.-10.0) THEN
             VR=VX*(1.5-0.00375*Y(I)**2)
          ENDIF
          FX(I)=-CD*PI*XR(I)*XR(I)*(XD(I)-VR)*ABS(XD(I)-VR)*DKX(I)
          FY(I)=-CD*PI*XR(I)*XR(I)*(YD(I)-VY)*ABS(YD(I)-VY)*DKY(I)
 530   CONTINUE
       RETURN
       END
C---------------------------------------------------------------------
C     Function of random number generation
C---------------------------------------------------------------------
       FUNCTION RAN(IDUM)
       PARAMETER (M=714026,IA=1366,IC=150899,RM=1.0/M)
       DIMENSION IR(97)
       DATA IFF /0/
       IF ((IDUM.LT.0).OR.(IFF.EQ.0)) THEN
         IFF=1
         IDUM=MOD(IC-IDUM,M)
       DO 600 J=1,97
         IDUM=MOD(IA*IDUM+IC,M)
         IR(J)=IDUM
 600   CONTINUE
       IDUM=MOD(IA*IDUM+IC,M)
       IY=IDUM
       ENDIF
       J=1+(97*IY)/M
C       IF ((J.GT.97).OR.(J.LT.1)) PAUSE
```

```
       IY=IR(J)
       RAN=IY*RM
       IDUM=MOD(IA*IDUM+IC,M)
       IR(J)=IDUM
       RETURN
       END
C----------------------------------------------------------------------
C     End of Program  (July, 1996)
C**********************************************************************
```

An Example of Input Files

```
REM
REM Start, End, Min.step, Vx, Vy, Thick, Fmax, E, U, TOL
PARA
0.0  500.0  1.0  0.00005  5.0  0.0  2.5  0.0  0.7  0.3  0.00001
END PARA
REM  IG,  ISEED,   RI,    XI,    YI
RAND
    1      2     0.0     0.1     0.02
END RAND
DISK
1        3.37      -21.0       -12.0        5.0        0.0
2        3.32      -21.0        -4.0        5.0        0.0
3        3.35      -21.0         4.0        5.0        0.0
4        3.40      -21.0        12.0        5.0        0.0
5        3.31      -28.5        16.0        5.0        0.0
6        3.42      -28.5         8.0        5.0        0.0
7        3.34      -28.5         0.0        5.0        0.0
8        3.38      -28.5        -8.0        5.0        0.0
9        3.32      -28.5       -16.0        5.0        0.0
10       3.41      -36.0       -12.0        5.0        0.0
11       3.35      -36.0        -4.0        5.0        0.0
12       3.40      -36.0         4.0        5.0        0.0
13       3.31      -36.0        12.0        5.0        0.0
14       3.37      -43.5        16.0        5.0        0.0
15       3.40      -43.5         8.0        5.0        0.0
16       3.38      -43.5         0.0        5.0        0.0
17       3.36      -43.5        -8.0        5.0        0.0
18       3.35      -43.5       -16.0        5.0        0.0
19       3.37      -51.0       -12.0        5.0        0.0
20       3.39      -51.0        -4.0        5.0        0.0
21       3.34      -51.0         4.0        5.0        0.0
22       3.36      -51.0        12.0        5.0        0.0
23       3.33      -58.5        16.0        5.0        0.0
24       3.40      -58.5         8.0        5.0        0.0
25       3.28      -58.5         0.0        5.0        0.0
26       3.40      -58.5        -8.0        5.0        0.0
27       3.35      -58.5       -16.0        5.0        0.0
28       3.32      -66.0       -12.0        5.0        0.0
29       3.38      -66.0        -4.0        5.0        0.0
30       3.35      -66.0         4.0        5.0        0.0
31       3.40      -66.0        12.0        5.0        0.0
END DISK
REM
ROOT
1      180.0     -20.0     -200.0
2      180.0     -20.0      200.0
3        3.5       5.0      -23.0
4        3.5       5.0       23.0
5       14.5      10.0      -35.0
6       14.5      10.0       35.0
7        3.8      15.0      -23.0
8        3.8      15.0       23.0
9       14.2      20.0      -35.0
10      14.2      20.0       35.0
11       4.0      25.0      -23.0
12       4.0      25.0       23.0
```

```
13        14.0      30.0      -35.0
14        14.0      30.0       35.0
15         4.0      35.0      -23.0
16         4.0      35.0       23.0
17        14.0      40.0      -35.0
18        14.0      40.0       35.0
19         4.0      45.0      -23.0
20         4.0      45.0       23.0
21        14.0      50.0      -35.0
22        14.0      50.0       35.0
23         4.0      55.0      -23.0
24         4.0      55.0       23.0
25        14.0      60.0      -35.0
26        14.0      60.0       35.0
27         4.0      65.0      -23.0
28         4.0      65.0       23.0
29        14.0      70.0      -35.0
30        14.0      70.0       35.0
31         4.0      75.0      -23.0
32         4.0      75.0       23.0
33        14.0      80.0      -35.0
34        14.0      80.0       35.0
35         4.0      85.0      -23.0
36         4.0      85.0       23.0
37        14.0      90.0      -35.0
38        14.0      90.0       35.0
39         4.0      95.0      -23.0
40         4.0      95.0       23.0
41        14.0     100.0      -35.0
42        14.0     100.0       35.0
43         4.0     105.0      -23.0
44         4.0     105.0       23.0
45        14.0     110.0      -35.0
46        14.0     110.0       35.0
47         4.0     115.0      -23.0
48         4.0     115.0       23.0
49        14.0     120.0      -35.0
50        14.0     120.0       35.0
51         4.0     125.0      -23.0
52         4.0     125.0       23.0
53        14.0     130.0      -35.0
54        14.0     130.0       35.0
55         4.0     135.0      -23.0
56         4.0     135.0       23.0
57        14.0     140.0      -35.0
58        14.0     140.0       35.0
59         4.0     145.0      -23.0
60         4.0     145.0       23.0
61        14.0     150.0      -35.0
62        14.0     150.0       35.0
63         4.0     155.0      -23.0
64         4.0     155.0       23.0
65        14.0     160.0      -35.0
66        14.0     160.0       35.0
67         4.0     165.0      -23.0
68         4.0     165.0       23.0
69        14.0     170.0      -35.0
70        14.0     170.0       35.0
71         4.0     175.0      -23.0
72         4.0     175.0       23.0
END ROOT
```