Creation, optimization and verification of a three dimensional numerical model to simulate a dragline bucket during the digging cycle using modern *DEM* software

by

Graeme Francois Dryden Dymond

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Mechanical Engineering at the University of Stellenbosch

Department of Mechanical and Mechatronic Engineering
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Supervisor: D.N.J Els and C.J Coetzee

December 2007

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

G.F.D Dymond
Date : December 2007

Abstract

Creation, optimization and verification of a three dimensional numerical model to simulate a dragline bucket during the digging cycle using modern *DEM* software

G.F.D Dymond

Department of Mechanical and Mechatronic Engineering
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Thesis: MscEng (Mechanical)

December 2007

Dragline bucket designers are required to evaluate new bucket designs by building and testing scale buckets. Concerns about the reliability and accuracy of scale testing have been raised in recent years, but there was no alternative. However, recent advances in computing power and granular flow modeling are changing this and, we are entering an era where it is possible to numerically simulate dragline bucket filling. However, verification of the numerical simulation is necessary before useable data can be obtained.

This thesis explains the algorithm used by modern discrete element codes to simulate granular materials. The process used to create the numerical model and calibrate the material will be discussed. An experimental test bench was also built to record experimental data for the verification the numerical model.

As the project progressed it became clear that the time needed to run a single simulation dramatically limits the number of simulations that could be run. Consequently, different approaches that could reduce simulation time were also investigated.

Unlike the other material parameters, there is no test that can be used to directly calibrate the damping. An array of numerical simulations were therefore conducted testing different damping schemes. The comparison performed between the numerical and experimental data showed that the numerical models could not accurately simulate the experimental measurements of the scale model dragline bucket. The numerical model did, however, predict many of the trends identified in the experimental simulation. With more realistic contact models and better computer facilities, nonetheless, it is highly probable that numerical models will be capable of simulating dragline bucket filling accurately. Further study is, therefore, justified.

Uittreksel

Skepping, optimisering en verifiëring van 'n driedimensionele numeriese model vir die simulering van 'n sleepgraafbak gedurende die graafsiklus met behulp van moderne DEM-programmatuur

("Creation, optimization and verification of a three dimensional numerical model to simulate a dragline bucket during the digging cycle using modern DEM *software")*

G.F.D Dymond

Departement Meganiese en Megatroniese Ingenieurswese
Universiteit van Stellenbosch
Privaatsak X1, 7602 Matieland, Suid Afrika

Tesis: MscIng (Meganies)
Desember 2007

Die ontwerpers van sleepgraafbakke moet nuwe bakontwerpe evalueer deur bakke op skaal te bou en te toets. In die jongste tyd word kommer oor die betroubaarheid en akkuraatheid van skaaltoetse geopper, maar daar is tans geen alternatief nie. Onlangse verbeterings in rekenaarvermoë en die modellering van granulêre vloei verander tans hierdie situasie, en ons betree nou 'n tydperk waarin dit moontlik is om die vul van 'n sleepgraafbak numeries te simuleer. Die numeriese simulasie moet egter geverifieer word voordat bruikbare gegewens verkry kan word. Hierdie tesis verduidelik die algoritme wat deur moderne diskrete-elementkodes gebruik word om granulêre materiale te simuleer. Die proses wat gebruik is om die numeriese model te skep en die materiaal te kalibreer word bespreek. 'n Eksperimentele toetsbank is ook gebou om eksperimentele gegewens vir die verifiëring van die numeriese model op te neem.

Namate die projek gevorder het, het dit geblyk dat die tyd wat vir 'n enkele simulasie benodig word die aantal simulasies wat uitgevoer kon word, dramaties verminder het. Verskillende benaderings waardeur die simuleringstyd verkort kon word is dus ook ondersoek.

Anders as met die ander matariaalparameters is daar geen toets waarmee die demping direk gekalibreer kan word nie. 'n Stel numeriese simulerings is dus uitgevoer om verskillende dempskemas te toets. Die vergelyking van die numeriese met die eksperimentele gegewens het getoon dat die numeriese model nie die eksperimentele metings van die skaalmodel van die sleepgraafbak akkuraat kon simuleer nie. Met realistieser kontakmodelle en beter rekenaargeriewe is dit nietemin hoogs waarskynlik dat numeriese modelle in staat sal wees om die vul van sleepgraafbakke akkuraat te simuleer. Verdere studie is dus geregverdig.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations ...

D.N.J Els
C.J Coetzee
Vr-Steel

Dedications

This thesis is dedicated to my family for their support, both emotional and financial.

Contents

List of Figures

List of Tables

Nomenclature

Variables

A	Cross sectional area
CS	Cable speed
c_n	Normal velocity damping coefficient
c_s	Shear velocity damping coefficient
dl'	Non dimensional distance
E	Modulus of elasticity
E_{eff}	Effective modulus of elasticity (Hertz - Mindlin contact model)
e	Coefficient of restitution
F'	Non dimensional force
Fd	Drag force
Ft	Tooth force
I	Moment of inertia
J	Polar moment of inertia
k_{bn}	Normal bond stiffness
k_{bs}	Shear bond stiffness
k_c	Cable stiffness
m	Mass
k_n	Normal stiffness
k_{n_1}	Normal loading stiffness (Walton - Braun contact model)
k_{n_2}	Normal unloading stiffness (Walton - Braun contact model)
$\tilde{k}_n$	Normal stiffness (Hertz - Mindlin contact model)
k_s	Shear stiffness
R	Particle radius
Np	Number of particles in a clump
Nw	Number of walls of the bucket
R_{eff}	Effective particle radius (Hertz - Mindlin contact model)
α_d	Damping coefficient

ΔCS	Accumulated cable length change
Δcs	Cable length change
Δcsp	Cable length change (projected)
Δt_c	Critical time step
Δt	Time step
ρ	Density
μ	Friction coefficient

Vectors and Tensors

$\vec{HD}$	Hitch point to drag origin unit vector
$\vec{n}$	Normal unit vector
$\vec{s}$	Shear unit vector
$\overline{\varrho}$	Rotational quaternion

$\overline{F}$	Force
$\overline{FB}$	Bond force
$\overline{Fb}$	Body force
$\overline{Fc}$	Contact force
$\overline{FC}$	Cable force
$\overline{Fd}$	Damping force
$\overline{Fr}$	Resultant force
$\overline{M}$	Moment
$\overline{MB}$	Bond moment
$\overline{Mb}$	Body moment
$\overline{Mc}$	Contact moment
$\overline{Md}$	Damping moment
$\overline{Mr}$	Resultant moment
$\overline{x}$	Displacement
$\overline{xc}$	Displacement form particle to soft contact centers
$\overline{w}$	Rotation

$\overline{\Delta n}$	Normal overlap displacement
$\overline{\Delta s}$	Shear overlap displacement
$\overline{\Delta b}$	Bond overlap displacement

$\dot{\overline{x}}$	Translational velocity
$\dot{\overline{w}}$	Rotational velocity

$\dot{\overline{\Delta n}}$	Normal overlap velocity
$\dot{\overline{\Delta s}}$	Shear overlap velocity
$\ddot{x}$	Translational acceleration
$\ddot{w}$	Rotational acceleration

Matrixes

$\overline{\underline{E}}_S^R$	Static to rotary transform
$\overline{\underline{E}}_R^S$	Rotary to static transform
$\overline{\underline{I}}$	Moment of inertia
$\overline{\underline{Id}}$	Inverted moment of inertia (principal axes)
$\overline{\underline{m}}$	Mass

Subscripts

A	Particle A
B	Particle B
C	Clump
G	Center of gravity
n	Normal direction
P	Point
R	Rotary coordinate system
S	Static coordinate system
s	Shear direction
t	Current time step
$t+1$	Next time step
$t-1$	Previous time step

Chapter 1

Introduction

1.1 Background

1.1.1 Simulation of granular materials

Granular flow theory deals with large shear deformations and displacements. This has, typically, been used in the fields of earth moving and particle flow.

Granular materials have always been of interest to engineers. Civil engineers use granular theories to determine whether bridges, tunnels and other structures will be stable. Mechanical engineers use granular theories in mining, agriculture, transportation and vibrations. In recent years, chemical engineers have also used granular theories to predict particulate formation and behaviour.

There are many theories or models available for specific applications. Although many of the available theories have been determined by experimental means and still await theoretical validation. Only in recent years, with the advent of faster computers, has it become possible to numerically simulate the behavior of these materials. Initially, these numerical models were used to verify existing experimental models, but they have since evolved to the point where they can be used to solve real engineering problems, making many of the older theories redundant.

There are two different schools of thought concerning the simulation of granular materials; namely, continuum mechanics and discrete element methods (*DEM*). Continuum mechanics treats the material as a deformable continuum and is, therefore, better suited to fine materials where the variation in particle size is small. The major advantage continuum methods have over discrete methods is faster processing speeds.

Discrete element methods can use a collection of particles to emulate a granular material. The movement of each particle is solved individually, allowing for a greater size distribution, but making the simulation slower.

There are limitations to both continuum and discrete methods. It is frequently not possible to generate a model that works on the same particle scale as the problem since the computational time becomes far too large. Consequently, continuum and discrete methods create models that simplistically mimic the behavior of a real system and, as a result, experimental validation is needed before these can be used in a specific engineering application. This experimental validation is achieved by determining the level of accuracy with which the numerical simulation can predict the experimental results.

Regarding the topic of this thesis, it should be noted that using *DEM* to model the flow into dragline buckets has been previously conducted by Cleary (1997, 1998); Coetzee (2000).

1.1.2 Open cast mining

In open cast mining the ground above or ’overburden’ needs to be removed in order to mine the ore below. This overburden can vary from topsoil to hard rock. The bulk of the costs involved in open cast mining can be attributed to overburden removal.

There are various ways of removing this overburden and this section briefly lists and discusses some of the common methods.

Bucket wheel excavators are crawlers with rotating buckets mounted on their sides, similar to an old fashioned waterwheel. Bucket wheel excavators have a very high breaking force and can, therefore, remove overburden without blasting. The greatest drawback these excavators have is a lack for redundancy. When one of the buckets breaks, the machine can no longer operate.

Scrapers use a blade to scrape away overburden. Scrapers have a very low breaking force and are primarily used to remove topsoil.

Combinations of Shovels, trucks and Front-End Loaders can be used to break, load and remove overburden. The major disadvantage of this system is the associated high labour and fuel costs.

Draglines are the most popular of all overburden removal devices. They are usually used in combination with scrapers to reduce the amount of rehandle. Draglines are crane-like structures that pull a bucket through the overburden. Once the bucket has been filled, it is hoisted up. The base of the dragline swivels and the overburden is dumped where the ore has already been excavated. The average size of the dragline buckets are between sixty and eighty cubic metres, which generates a very high overburden removal rate. The overburden usually needs to be blasted before the dragline can be used.

Draglines are preferred because they are very economical and are responsible for moving about thirty percent of the world’s overburden. Dragline

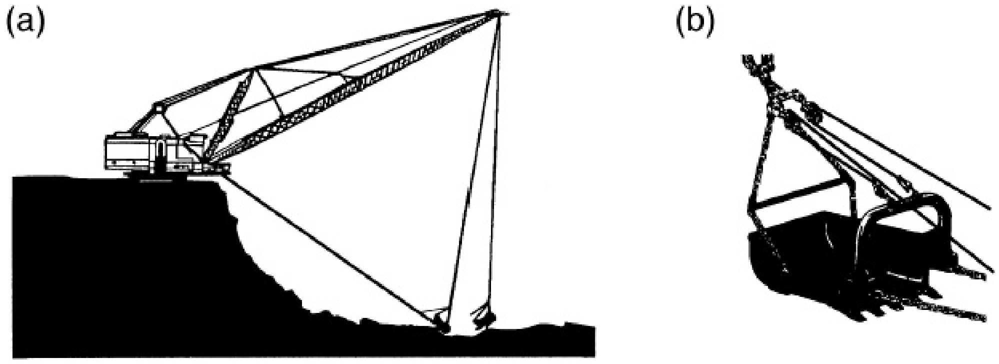


Figure 1.1: Dragline and dragline bucket

productivity is influenced by many different factors. Among the most important are digging conditions, the dragline bucket itself, the dragline setup and the dragline operator.

1.2 Thesis description

1.2.1 Objective

The objective of this thesis is to determine whether a modern *DEM* code, *PFC*, can be used to accurately model the flow of overburden into a dragline bucket. To ascertain this, a PFC simulation of a dragline bucket during filling needed to be designed . *PFC* is a command-based solver with no preprocessor so the simulation needs to be programmed and inputted into the *PFC* solver. Experimental tests that could be used to verify the numerical model need also to be performed and a record of the bucket's dynamics during filling made.

1.2.2 Motivation

Draglines are designed to operate twenty four hours a day for three hundred and sixty four days a year. The cost of the loss of production due to a dragline standing has been estimated at $8000 Australian dollars an hour, (P Dayawansa and Price, 2004). Many dragline breakdowns can be attributed to the design of the dragline bucket. The buckets either fail or overload the machine and cause failures in the dragline boom and main structure. Dragline bucket designs are currently tested by building and testing scale models. This process is slow and does not always provide accurate results. The numerical simulation tools developed over the last decade could aid in dragline bucket design and, one day, replace the need for scale testing.

Chapter 2

Theory and background

2.1 A brief history of discrete element modeling *DEM*

Discrete element modeling has been used to predict the behaviour of granular materials for the last two decades. The theory, nonetheless, dates further back. One of the first applications of discrete element modeling was made by Alder and Wainwright (1967) to try and track the movement of individual atoms and molecules.

Computers were used for the first time in the late nineteen seventies. A discrete element modeling code was developed by Cundall and Strack (1979) to simulate rock fracture mechanics.

Today, there are commercial discrete element codes available for both two and three dimensions. With the development of faster computers, the number of particles that can be simulated has increased, allowing for more accurate material representation. This software can be used to analyse flow patterns, forces, velocity vectors and even deformations of particles in the newer codes.

Most of the research currently associated with discrete element codes deals with the development of more realistic contact models and particle shapes. Research is also being conducted into combining *DEM* code with *FEM* code, which would allow the forces within particles themselves to be analysed.

2.2 *DEM* code

2.2.1 Basic working

The *DEM* algorithm simulates a material by calculating the motion of a collection of particles. The interaction between the particles determines the macroscopic behaviour of the material and allows it to flow, deform or fracture. The motion of the particles is solved by using *Newton's laws of motion* to continually update the particles' dynamics. The forces and moments on the particles are obtained from the contact between particles, see figure 2.1.

The *DEM* code used in this thesis is *PFC v3.0*. It was developed by *Itasca Codes* and is based on the Cundall and Strack (1979) code.

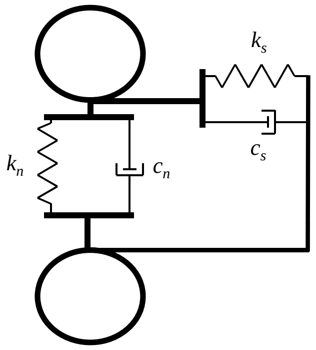


Figure 2.1: General schematic of a contact model

2.2.2 Numerical integration

PFC, like most *DEM* codes, makes use of a central difference explicit time integration scheme for the velocity and displacement of the particles. In order to reduce the inherited inaccuracy of explicit integration, the time step needs to be very small.

The particles in *PFC* are treated like springs that have a finite mass and size. The critical time step needed can, therefore, be calculated using a linear spring system, Cundall and Strack (1979);

$$\Delta t_c = \min \begin{cases} \sqrt{m/k_n} & \text{translation} \\ \sqrt{I/k_s} & \text{rotational} \end{cases} \tag{2.2.1}$$

where m, I, k_n, k_s are the mass, moment of inertia, normal and shear stiffness, respectively. The time step calculation can changed depending on the contact or damping model being used.

Two main processes occur in each time step or cycle, namely, the *law of motion* and the *force displacement law*, see figure 2.2.

2.2.3 Force displacement law

This part of the cycle uses the particle's updated positions to calculate the forces between interacting particles. The magnitude of these forces is determined by means of a contact model, see section 2.3. For now, we will assume

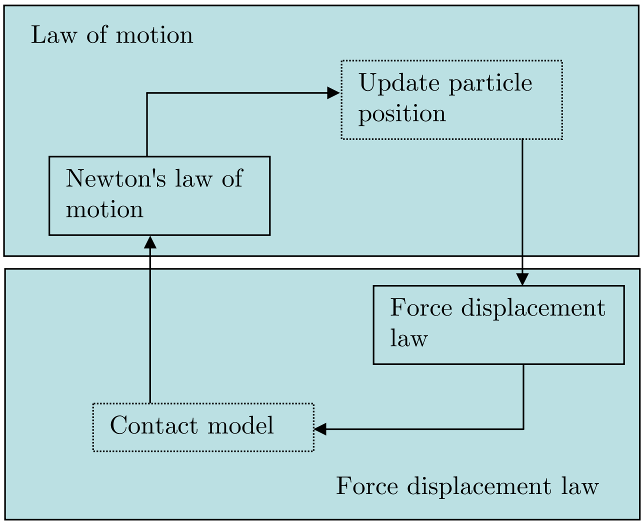


Figure 2.2: Basic flowchart of the *DEM* algorithm

the generated force and moment between two particles in contact can be written as a function of the overlap, namely;

$$\begin{aligned}\overline{Fc} &= f(\overline{\Delta n})\\ \overline{Mc} &= f(\overline{\Delta s})\end{aligned} \tag{2.2.2}$$

where $\overline{\Delta n}$, and $\overline{\Delta s}$ are the overlap in the normal and tangential directions. To simulate non-elastic collisions a damping force and moment is also generated. The damping force and moment can be a function of the overlap and/or overlap velocity. Damping models will be discussed in section 2.4.

$$\begin{aligned}\overline{Fd} &= f(\overline{\Delta n}, \dot{\overline{\Delta n}})\\ \overline{Md} &= f(\overline{\Delta s}, \dot{\overline{\Delta s}})\end{aligned} \tag{2.2.3}$$

2.2.4 Law of motion

At this point in the cycle, see figure 2.2, the particle position is updated using the particle's initial values and the forces calculated by the force displacement law.

At the beginning of each time step each particle has an initial displacement ($\overline{x}_t$), velocity ($\dot{\overline{x}}_t$), acceleration ($\ddot{\overline{x}}_t$) , rotation ($\overline{w}_t$), rotational velocity ($\dot{\overline{w}}_t$) and rotational acceleration ($\dddot{\overline{w}}_t$).

Using *Newton's second law of motion of bodies with constant mass*, namely

$$\begin{aligned}\sum \overline{F} &= \underline{\overline{m}} \cdot \ddot{\overline{x}} \\ \sum \overline{M} &= \underline{\overline{I}} \cdot \ddot{\overline{w}}\end{aligned} \tag{2.2.4}$$

where $\underline{\overline{m}}$ and $\underline{\overline{I}}$ are the mass and moment of inertia matrix. Rearranging equation 2.2.4 and using a forward difference explicit integration approximation, the particles updated velocity can be written as;

$$\begin{aligned}\dot{\overline{x}}_{t+1} &= \dot{\overline{x}}_t + (\underline{\overline{m}})^{-1}\overline{F}_t \cdot \Delta t \\ \dot{\overline{w}}_{t+1} &= \dot{\overline{w}}_t + (\underline{\overline{I}})^{-1}\overline{M}_t \cdot \Delta t\end{aligned} \tag{2.2.5}$$

where $\overline{F}_t$ and $\overline{M}_t$ can be defined as;

$$\begin{aligned}\overline{F}_t &= \sum \overline{Fc}_t + \sum \overline{Fd}_t + \sum \overline{Fb}_t \\ \overline{M}_t &= \sum \overline{Mc}_t + \sum \overline{Md}_t + \sum \overline{Mb}_t\end{aligned} \tag{2.2.6}$$

where $\overline{Fb}$ and $\overline{Mb}$ represent body forces and moments, such as gravity or applied loads. The particles new position can be calculated using another *Euler* explicit approximation and can be written as;

$$\begin{aligned}\overline{x}_{t+1} &= \overline{x}_t + \dot{\overline{x}}_t \cdot \Delta t \\ \overline{w}_{t+1} &= \overline{w}_t + \dot{\overline{w}}_t \cdot \Delta t\end{aligned} \tag{2.2.7}$$

The above equations are very simplified, but illustrate the basic algorithmic activity that takes place each time step. The more important aspects of the algorithm are discussed in detail below.

2.3 Contact models

A force is produced when two particles collide that forces them apart. There are two different methods to simulate this process, namely, rigid and soft body contacts. Rigid body contacts use the particles' initial velocity, with the coefficient of restitution, to determine the velocity of the particle after impact. Only soft body contact will be discussed in this thesis.

Soft body contacts allow the particles to occupy the same space (overlap). The contact force is calculated as a function of the magnitude of the particle overlap, see figure 2.5. The contact force can be divided into a normal ($\overline{Fc}_n$) and a tangential component ($\overline{Fc}_s$). There are many different types of contact models available, including various linear and non-linear models.

2.3.1 Linear contact models

The linear contact model used by *PFC* was first proposed by Cundall and Strack (1979). When two particles, A and B collide, a contact is generated. The contact contains a linear spring in the normal and shear directions.

The line of contact is defined as the line connecting the two particles' centres. The normal direction is defined along the line of contact with unit vector $\vec{n}$. The tangential direction is perpendicular to the line of contract with unit vector $\vec{s}$. At the point where particle A and *B's* circumference intersects, the line of contact is defined as $\overline{xc}_A$ and $\overline{xc}_B$.

Particles A and B have translational velocities of $\dot{\overline{x}}_A$ and $\dot{\overline{x}}_B$, and rotational velocities of $\dot{\overline{w}}_A$ and $\dot{\overline{w}}_B$. The velocity of points $\overline{xc}_A$ and $\overline{xc}_B$ is;

$$\begin{aligned} \dot{\overline{xc}}_{A_t} &= \dot{\overline{x}}_{A_t} + R_A \dot{\overline{w}}_{A_t} \\ \dot{\overline{xc}}_{B_t} &= \dot{\overline{x}}_{B_t} + R_B \dot{\overline{w}}_{B_t} \end{aligned} \tag{2.3.1}$$

where R_A and R_B are radii of particle A and B, respectively. The overlap velocity can now be written as;

$$\begin{aligned} \dot{\overline{\Delta n}}_t &= (\dot{\overline{xc}}_{A_t} - \dot{\overline{xc}}_{B_t}) \cdot \vec{n} \\ \dot{\overline{\Delta s}}_t &= (\dot{\overline{xc}}_{A_t} - \dot{\overline{xc}}_{B_t}) \cdot \vec{s} \end{aligned} \tag{2.3.2}$$

The overlap velocity is integrated to obtain the change in overlap for the given time step (t). The contact forces for the next time step ($t+1$) are therefore;

$$\begin{aligned} \overline{Fc}_{n_{t+1}} &= \overline{Fc}_{n_t} + k_n(\dot{\overline{\Delta n}}_t \cdot \Delta t) \\ \overline{Fc}_{s_{t+1}} &= \begin{cases} \overline{Fc}_{s_t} + k_n(\dot{\overline{\Delta s}}_t \cdot \Delta t) & \text{if } ||\overline{Fc}_{s_{t+1}}|| \leq \mu ||\overline{Fc}_{n_{t+1}}|| \\ \mu ||\overline{Fc}_{n_{t+1}}|| \cdot \vec{s} & \text{if } ||\overline{Fc}_{s_{t+1}}|| > \mu ||\overline{Fc}_{n_{t+1}}|| \end{cases} \end{aligned} \tag{2.3.3}$$

where k_n and k_s are the equivalent stiffness of particles A and B combined.

$$k_n = \frac{k_{n_A} k_{n_B}}{k_{n_A} + k_{n_B}} \qquad \text{and} \qquad k_s = \frac{k_{s_A} k_{s_B}}{k_{s_A} + k_{s_B}} \tag{2.3.4}$$

The magnitude of the shear force is limited by *Coulombs Law*, see figure 2.4. Most modern linear contact models are derivations of this contact model and, therefore, will not be discussed.

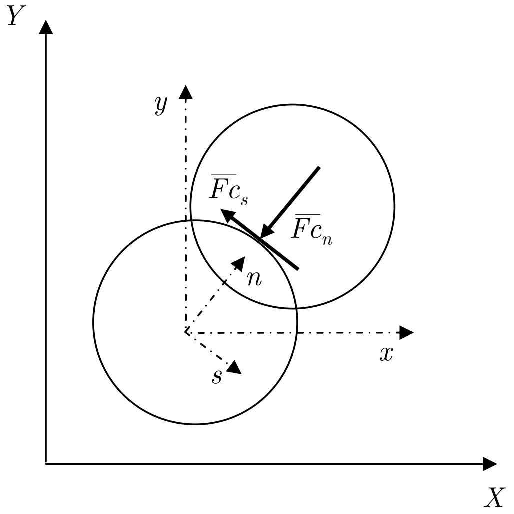


Figure 2.3: Coordinate frame

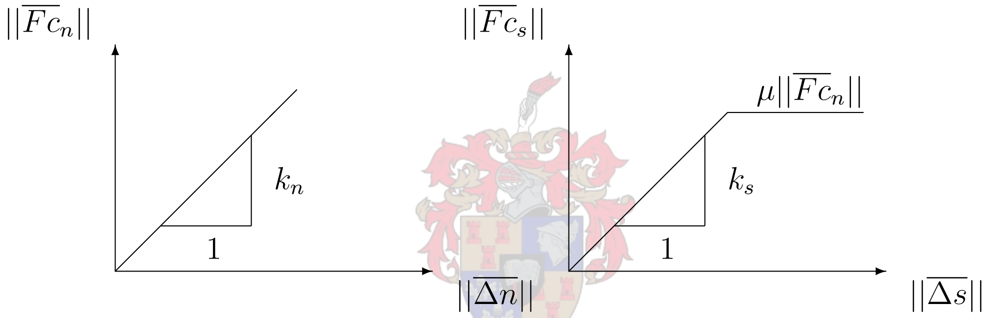


Figure 2.4: Coulombs friction law

2.3.2 Non-linear contact models

The linear contact model is commonly used because of its numerical simplicity. In more recent years, increased computational power has lead to more complex models being developed. These models provide a more realistic representation of contact forces. The basic *DEM* algorithm for linear and non-linear contact models is the same except in the determination of the normal and shear stress.

2.3.2.1 Hertz-Mindlin contact model

Named after Hertz and Mindlin, this was one of the first non-linear contact models. The Hertz contact model Hertz (1882) was based on the theory of elastic contact between spheres.

Using the Hertz contact model, the normal force is calculated as;

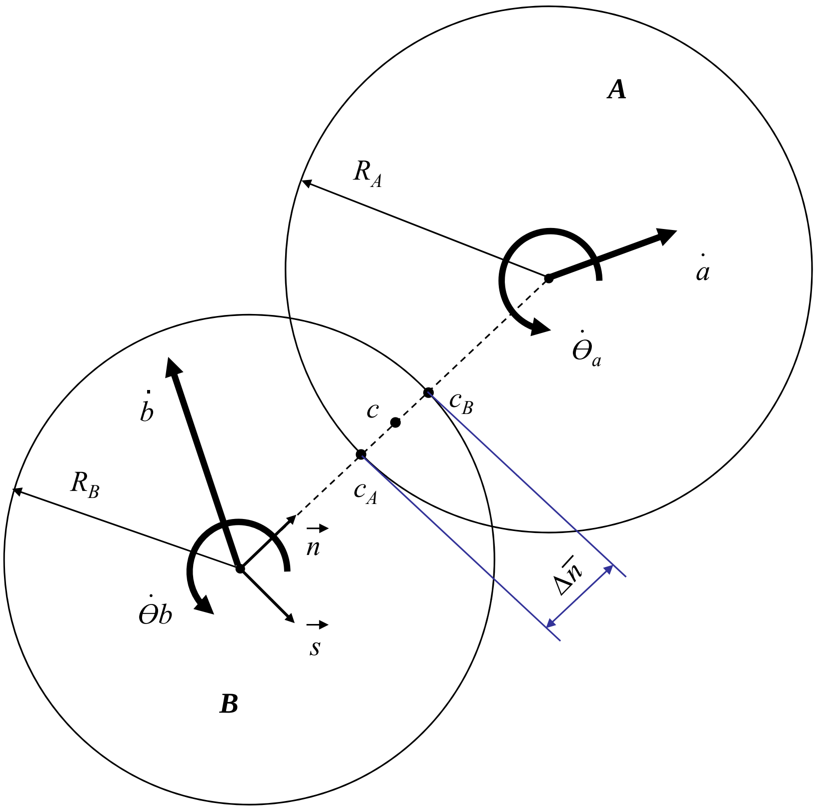


Figure 2.5: Contact overlap model

$$\overline{Fc_n} = -\tilde{k}_n \overline{\Delta n}^{3/2} \tag{2.3.5}$$

The Hertz contact model does not require a predefined stiffness (k_n) like the linear contact model. The Hertz contact model automatically calculates a normal stiffness using the modulus of elasticity (E) and the radius(R) of the particle.

$$\tilde{k}_n = 3/4\sqrt{R_{eff}E_{eff}} \tag{2.3.6}$$

where

$$\begin{aligned} R_{eff} &= \begin{cases} 2R_AR_B/R_A + R_B & \text{particle - particle} \\ R_A & \text{particle - boundary} \end{cases} \\ E_{eff} &= \begin{cases} 1/2(E_A + E_B) & \text{particle - particle} \\ E_A & \text{particle - boundary} \end{cases} \end{aligned} \tag{2.3.7}$$

The shear force is calculated using the theory of Mindlin and Deresiewicz (1953) on the tangential force between elastic spheres.

2.3.2.2 Walton-Braun contact model

This contact model was developed from the Hertz-Mindlin contact model Walton and Braun (1986). They measured the forces between colliding particles and determined that the normal force followed a hysteresis loading/unloading cycle.

Experimental data showed that the Hertz contact model could accurately predict the normal loading force. When adding plastic-elastic material properties, the non-linear normal Hertzian contact model could be approximated by a linear model.

$$\overline{Fc_n} = \begin{cases} k_{n_1}\overline{\Delta n} & \dot{\overline{\Delta n}} \geq 0 \quad \text{(loading)} \\ k_{n_2}(\overline{\Delta n} - \overline{\Delta n_0}) & \dot{\overline{\Delta n}} < 0 \quad \text{(unloading)} \end{cases} \tag{2.3.8}$$

Where $\overline{\Delta n_0}$ is the value where the unloading normal force becomes zero, see figure 2.6. Figure 2.6 shows that when loading occurs ($\dot{\overline{\Delta n}} \geq 0$) the force follows $\bar{ab}$. When unloading ($\dot{\overline{\Delta n}} < 0$) occurs it follows $\bar{bc}$. If loading occurs after unloading has occurred, the force retraces the unloading curve until it intersects the loading curve and the follows normal loading.

$\bar{ab}$(loading) → $\bar{bc}$ (unloading) → $\bar{cb}$(loading) → $\bar{bd}$(loading) → $\bar{de}$(unloading)

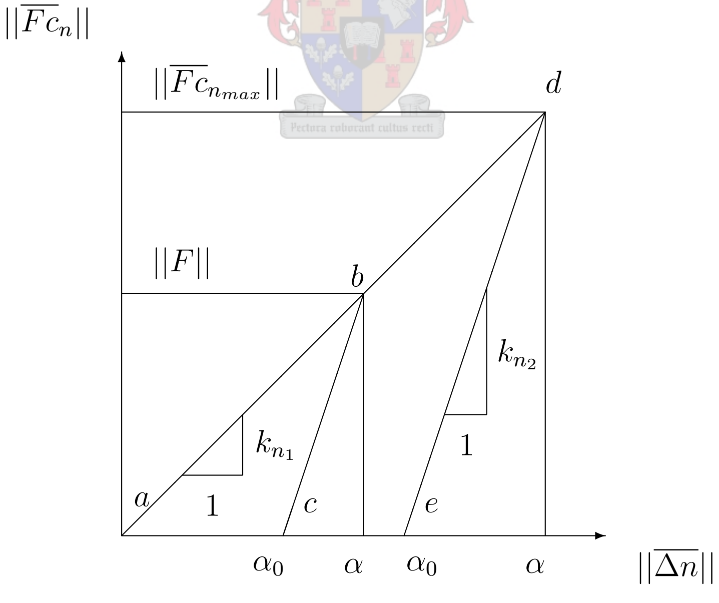


Figure 2.6: Walton-Braun normal contact force

The Walton-Braun contact model makes the normal force dependant on the particles position and hysteresis. Walton-Braun determined two different modes that the normal contact force model could take, namely, constant and variable coefficient of restitution (e). For both modes the loading stiffness (k_{n_1}) remains constant.

Constant coefficient of restitution assumes that the unloading stiffness (k_{n_2}) remains constant. The coefficient of restitution is, therefore, independent of the initial impact velocity. The coefficient of restitution is given by;

$$e_{mode1} = \sqrt{k_{n_1}/k_{n_2}} \tag{2.3.9}$$

In the second mode or variable coefficient of restitution, the unloading force is a function of the maximum force experienced during loading.

$$k_{n_2} = k_{n_1} + S\overline{Fc}_{n_{max}} \tag{2.3.10}$$

Walton-Braun experimental data showed that the coefficient of restitution was dependant on the initial impact velocity. The coefficient of restitution is given by;

$$e_{mode2} = \sqrt{w_0/(Sv_0 + w_0)} \tag{2.3.11}$$

where

$$w_0 = \sqrt{2k_{n_1}/m} \tag{2.3.12}$$

In the Hertz-Mindlin model, the shear force equation makes the assumption that the normal stress distribution is unaffected by an increase in tangential velocity. In the Walton-Braun contact model, the tangential stiffness (k_s) decreases with an increase in tangential velocity until it reaches zero, where slipping occurs. Walton-Braun tangential stiffness was defined as follows;

$$k_s = \begin{cases} k_{s_0}\Big(1 - (F_s - F_s')/(\mu F_n - F_s')\Big)^{\gamma} & F_s \text{ increasing} \\ k_{s_0}\Big(1 - (F_s - F_s')/(\mu F_n + F_s')\Big)^{\gamma} & F_s \text{ decreasing} \end{cases} \tag{2.3.13}$$

where

k_{s_0} = Initial tangential stiffness.
F_s = tangential or shear force
F_s' = initial set to zero and thereafter the total tangential or shear force
γ = constant usually 1/3
μ = friction coefficient
F_n = normal force

The tangential force can now be calculated using;

$$\overline{Fc}_{s_{t+1}} = \overline{Fc}_{s_t} + k_s\dot{\overline{\Delta s}}_t \tag{2.3.14}$$

Where k_s is given by equation 2.3.13.

Linear contact models were used primarily in this thesis since they were provided as part of the base code. The Hertz-Mindlin model would only be used with non-bonded or clumped particles and was, therefore, not employed. Towards the end of this thesis, a trail version of *PFC v3.1* was borrowed from *Itasca* to run simulations using a hysteresis contact model.

2.4 Damping models

When particles collide it is practically never a perfectly elastic collision. To simulate a non-elastic collision in *DEM*, energy needs to be dissipated when particles collide. There are various different damping schemes that can be used to damp the energy in the system. *PFC* has two built-in damping schemes, namely, global and viscous damping.

2.4.1 Global damping

Global damping is the simplest form of damping that can be used but also the most unrealistic. Global damping was the first implemented by Cundall and Strack (1979) in their discrete element code *BALL*. Global damping is best suited for static systems as it rapidly forces the system into an equilibrium state.

Global damping damps the absolute velocity of the particle by using the resultant force acting on the particle;

$$\begin{aligned}
\overline{Fd} &= -\alpha_d \frac{\dot{\overline{x}}}{||\dot{\overline{x}}||}||\overline{F_t}|| \\
\overline{Md} &= -\alpha_d \frac{\dot{\overline{w}}}{||\dot{\overline{w}}||}||\overline{M_t}||
\end{aligned} \tag{2.4.1}$$

where α_d, $\overline{Ft}$ and $\overline{Mt}$ represent the damping coefficient, the total force and moment, respectively. It is important to note that the damping force uses the absolute magnitude of the total force, but the negative direction of the particles' velocity.

2.4.2 Viscous damping

Viscous damping can best be visualised as dashpot acting in the normal ($\vec{n}$) and shear ($\vec{s}$) directions during contact. Viscous damping is, therefore, better suited to dynamic systems since it damps proportional to velocity instead of force. See figure 2.1.

$$\begin{aligned}\overline{Fd} &= -c_n\dot{\overline{\Delta n}} \\ \overline{Md} &= -c_s\dot{\overline{\Delta s}} \times \overline{xc}\end{aligned} \tag{2.4.2}$$

where c_n and c_s are the damping coefficient in the normal and shear directions and $\overline{xc}$ is the distance from the particle's center of mass to the contact point.

2.5 Clusters and bonded particles

Most granular materials are not spherical in nature. *PFC* has allowed particles to be linked or bonded to create more complex particles shapes. *PFC* has two main methods of creating linked particles, namely, clumping and bonding.

2.5.1 Clumps

PFC allows a collection of balls to be clumped together to create a clump. Clumps remain ridged under all conditions. Clumps logic is the fastest of the linked ball algorithms in *PFC*.

The clump's mass and inertia are a function of the individual particles comprising the clump. The clump's mass (m) and center of mass ($\overline{x}_G$) can be calculated as follows;

$$\begin{aligned}m_C &= \sum_{i=1}^{Np} m_i \\ \overline{x}_{G_C} &= \frac{1}{m_C}\sum_{i=1}^{Np} m_p\overline{x}_{G_i}\end{aligned} \tag{2.5.1}$$

where Np is the number of particles comprising the clump and $\overline{x_G}$ is the location of the particles' center of mass. The clumps inertial matrix ($\underline{\overline{I}}_C$) is obtained by;

$$\underline{\overline{I}}_C = \sum_{i=1}^{Np}\left(\underline{\overline{I}}_i + m_i \cdot \underline{\overline{D}}_i\right) \tag{2.5.2}$$

where $\underline{\overline{I}}_i$ is the inertial matrix of particle i about its center of mass. $(\overline{x}_{C_G})$.

$\underline{\overline{D}}_p$ is defined as;

$$\begin{aligned}\underline{D}_i &= \begin{bmatrix} d_y^2 + d_z^2 & -d_x d_y & -d_x d_z \\ -d_x d_y & d_x^2 + d_z^2 & -d_y d_z \\ -d_x d_z & -d_y d_z & d_x^2 + d_y^2 \end{bmatrix} \\ \begin{bmatrix} d_x & d_y & d_z \end{bmatrix}^T &= \overline{x}_{G_i} - \overline{x}_{GC} \end{aligned} \tag{2.5.3}$$

The particles in the clump are treated as normal particles during the *force displacement* part of the *DEM* algorithm. The *law of motion* part of the algorithm can now be used, with the clump's mass and inertial matrix, to calculate the clump's new position and orientation. To increase processing speed, *PFC* has a parameter that specifies how often the clump's inertial matrix is updated due to possible rotations of the clump.

2.5.2 Bonded particles

There are two types of bonds in *PFC*, namely, contact and parallel bonds. Bonds can only be created between particles that are in contact or overlapping.

When a contact bond is created between two particles, it acts like a point of glue. The bond's normal and shear stiffness are the same as the particles that it bonds. The shear force is no longer limited by the slip model, but, rather, by the shear bond strength.

$$\begin{aligned}\overline{FB}_n &= k_n(\overline{\Delta b}_n - \overline{\Delta b}_{n0}) \\ \overline{FB}_s &= k_s(\overline{\Delta b}_s - \overline{\Delta b}_{s0})\end{aligned} \tag{2.5.4}$$

where $\overline{\Delta b}_{n0}$ and $\overline{\Delta b}_{s0}$ are defined when the bond is created. When the bond is created a normal and shear bond strength is specified. If either the normal or shear force exceeds the bond strength, the bond breaks. Contact bonds are not capable of resisting a bending moment, in other words, they can roll on each other.

When two particles are parallel bonded, a *cement disk* is placed between the particles. The *cement disk*, see figure 2.8, is capable of resisting normal and shear forces like contact bonds, as well as bending moments.

Parallel bonds have the following parameters: the bond normal stiffness (k_{bn}), shear stiffness (k_{bs}), normal and shear breaking forces and radius multiplier (λ). The normal and shear stiffness is specified in pressure per length.

The normal and shear force is defined as;

$$\begin{aligned}\overline{FB}_n &= k_n(\overline{\Delta b}_n - \overline{\Delta b}_{n0})A \\ \overline{FB}_s &= k_s(\overline{\Delta b}_s - \overline{\Delta b}_{s0})A\end{aligned} \tag{2.5.5}$$

where A is the area of the *cement disk*. The moments generated are defined as;

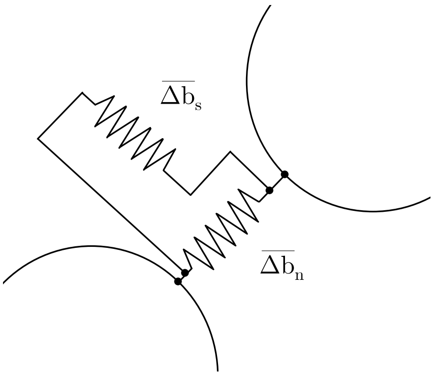


Figure 2.7: Contact bond

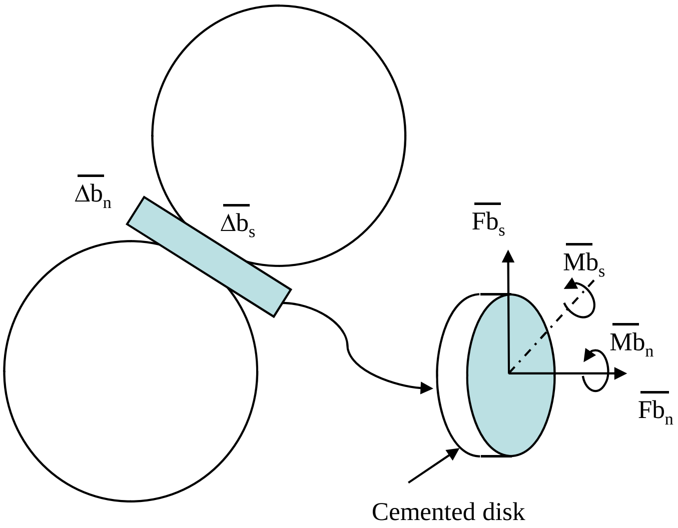


Figure 2.8: Parallel bond

$$
\begin{aligned}
\overline{MB}_n &= (-k_s J(\overline{\Delta\theta_n} - \overline{\Delta\theta_{n0}})) \\
\overline{MB}_s &= (-k_n I(\overline{\Delta\theta_s} - \overline{\Delta\theta_{s0}})) + \overline{Fb}_s \times \overline{p}_c
\end{aligned}
\tag{2.5.6}
$$

where I and J are the polar moment of inertia around the shear and normal directions, θ_n and θ_s are the rotational overlap and $\Delta\theta_{n0}$ and $\Delta\theta_{s0}$ are the overlap on creation.

2.5.3 Non spherical particles

In the last decade, discrete element code has been generated that uses non-spherical, usually ellipsoidal, particles, Vu-Quoc *et al.* (2000); Cleary *et al.*

(1997). Although simulations using these codes have fewer particles, they are very slow due to the complexity of the contact detection.

2.6 Contact detection

Simulations can have hundreds of thousands of particles. The *DEM* algorithm needs to know which particles are in contact with one another, to determine inter-particle forces. Many *DEM* codes use advance contact detection algorithms to reduce the search time. A contact detection algorithm manages a list of particles in contact. This list can then be passed to the contact force models where the magnitude and direction of the forces resulting from the contact can be calculated.

The time needed for the contact detection algorithm to run is dependant on the number of particles in the simulation. In three dimensions, the computation for contact detection can become one of the leading time factors. When non-spherical particles or clusters are present, the computational time dramatically increases.

PFC uses a contact detection algorithm that divides the simulation into a three dimensional matrix. Every particle and wall is mapped into one or more of these cells. *PFC* then checks for contacts between the particles within each cell. If one or more of the elements moves out of its cell, all the cells need to be remapped. To further increase speed, each particle has a contact list of neighboring particle contacts or near contacts. This removes the need to search for contacts every cycle. This list is updated every few cycles or when the system remaps.

2.7 Multiple processors

With large simulations, (over 100,000 balls), the time needed to run on a single computer can become prohibitively long. *PFC* has a parallel interface that allows one simulation to be solved using multiple computers or processors.

The processor assembly consists of one master processor and many slave processors. The simulation is then divided into sub-assemblies and each sub-assembly is then assigned to a slave processor. This is very similar to the process described in section 2.6.

Each of the slave assemblies solves the *force displacement law* of its sub-assembly. Information about boundary particles is then sent to the master processor. The master processor adds each of the slave processor's data pertaining to a boundary particle and solves the particles dynamics before distributing the data back to each of the slaves. The *law of motion* is then applied, see figure 2.2. This process continues until the simulation is completed.

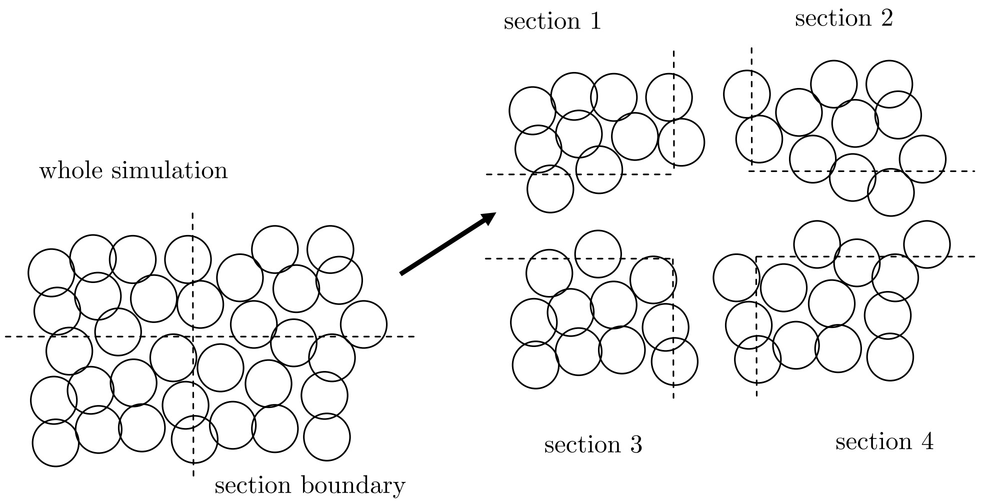


Figure 2.9: Virtual sections or zones

The master processor keeps track of which particles are present in each sub-assembly. Particles are free to move between sub-assemblies. The master process is also responsible for ensuring that the time step in each simulation is the same.

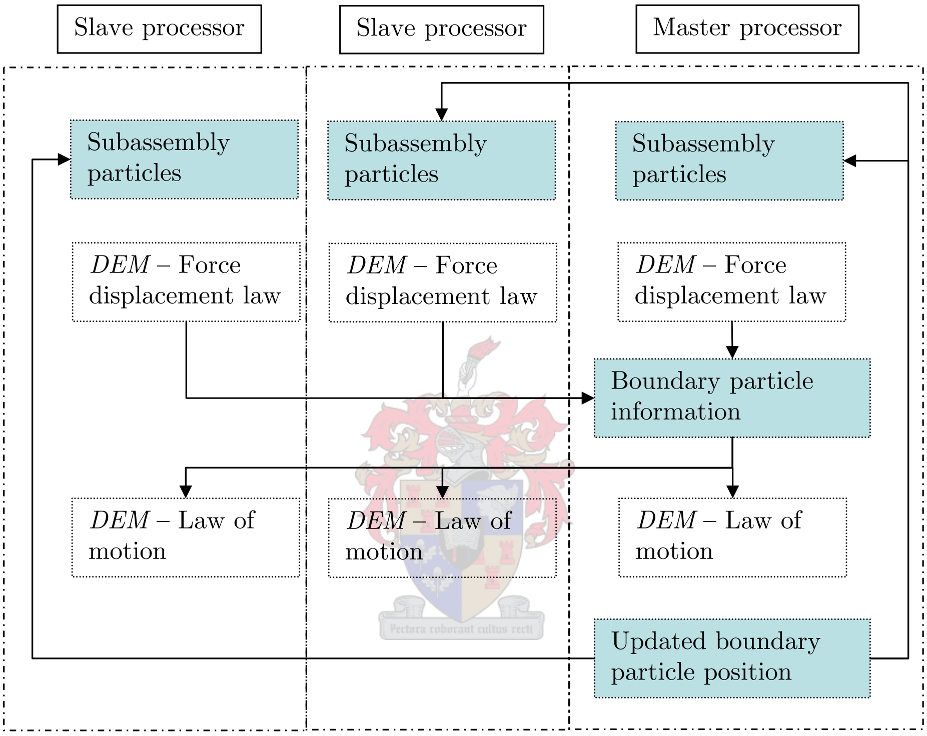


Figure 2.10: Multiple processors flowchart

Chapter 3

Numerical Model

The numerical simulation was performed using *PFC version 3.0* developed by *Itasca*. *PFC* has two main element types, namely, walls and balls. The balls are used to simulate the granular materials and the walls are used to simulated rigid bodies. *PFC* is a command-based solver that relies on the user to define the simulation setup and parameters. Built into the solver is a programming language called *FISH*, which can be used to control and/or modify the simulation in real-time.

The simulation consisted of a drag bed created using walls. The drag bed was then filled with balls to simulate the granular material or overburden, see figure 4.1. Once the balls had settled, a bucket, similar to figure 3.1 and also constructed using walls, was added and, again, the system was allowed to settle. The bucket was then dragged though the balls. The bucket's trajectory and the forces experienced by the drag chains were recorded.

3.1 General wall dynamics

PFC allows balls to have both fixed and forced dynamics. Walls, however, can only have fixed dynamics. Since the dragline bucket is constructed from walls, a FISH program that would continually update the wall dynamics needed to be written. A *FISH* program was written that would execute every time step and manually update the walls' velocities, based on the forces acting upon them. A brief working of the *FISH* program is given below.

PFC only works in a global or static coordinate system ($\underline{\overline{E}}_S$). In order to simplify the mathematics, a second coordinate system was attached to the center of gravity of the bucket and aligned with the principal axes ($\underline{\overline{E}}_R$). The rotation matrix from base $\underline{\overline{E}}_S$ to $\underline{\overline{E}}_R$ is defined as $\underline{\overline{E}}_S^R$ and vice versa.

FISH has built-in functions that can be used to determine the out of balance force and moment acting on a wall (i), namely, $\overline{Fr_{iS}}$ and $\overline{Mr_{iS}}$. The dynamics program uses these forces and moments, together with the physical properties of the bucket, to update the bucket's velocity in base $\underline{\overline{E}}_S$.

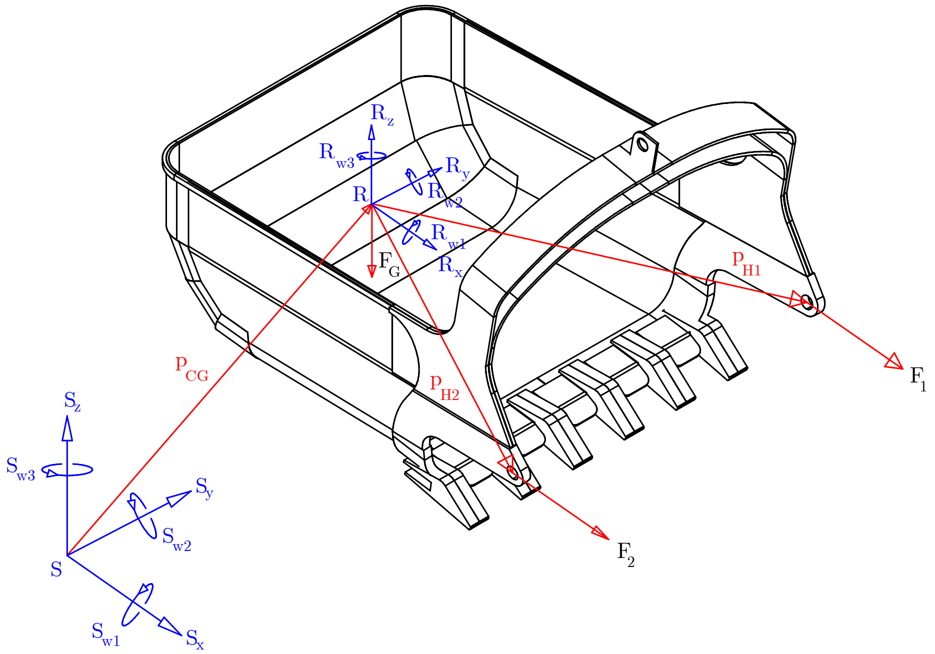


Figure 3.1: Bucket dynamics

Using the principal of super-positioning, in the static coordinate system, a point (p) on the bucket has a velocity of;

$$\dot{\overline{x}}_{pS} = \dot{\overline{x}}_{GS} + \overline{\underline{E}}_R^S\left(\dot{\overline{w}}_{GR} \times (\overline{x}_{pR} - \overline{x}_{GR})\right) \tag{3.1.1}$$

where $\dot{\overline{x}}_{GS}$ and $\dot{\overline{w}}_{GR}$ are the translation in the static axes system, $\overline{\underline{E}}_S$ and rotational velocities around the bucket's center of gravity in the rotational axes system, $\overline{\underline{E}}_R$. $\dot{\overline{x}}_{GS}$ is independent of the bucket's orientation and can be calculated using *Newtons second law.* For instance, the acceleration of the bucket's centre of mass $\ddot{\overline{x}}_{GS}$ at time t is;

$$\ddot{\overline{x}}_{GS_t} = \sum_{i=1}^{Nw}\left(\frac{\overline{Fr}_{iS_t}}{m_i}\right) \tag{3.1.2}$$

where m is the mass of the wall and Nw is the number of walls comprising the bucket. Time $t + \Delta t$, $\dot{\overline{x}}_{GS_{t+\Delta t}}$ is obtained by using an *Euler* explicit approximation;

$$\dot{\overline{x}}_{GS_{t+\Delta t}} = \dot{\overline{x}}_{GS_t} + \int_t^{t+\Delta t} \ddot{\overline{x}}_{GS_t} dt = \dot{\overline{x}}_{GS_t} + \ddot{\overline{x}}_{GS_t}\Delta t \tag{3.1.3}$$

The rotational velocity is calculated using $\overline{\underline{E}}_R$ as it simplifies the mathematics. The rotational acceleration ($\ddot{\overline{w}}_{GR}$) can be written as;

$$\ddot{\overline{w}}_{GR_t} = \underline{\overline{Id}} \cdot \overline{M}_R + \underline{\overline{Id}} \cdot \begin{bmatrix} (I_{[2,2]} - I_{[3,3]})\dot{w}_{[2]R_t}\dot{w}_{[3]R_t} \\ (I_{[3,3]} - I_{[1,1]})\dot{w}_{[1]R_t}\dot{w}_{[3]R_t} \\ (I_{[1,1]} - I_{[2,2]})\dot{w}_{[1]R_t}\dot{w}_{[2]R_t} \end{bmatrix} \tag{3.1.4}$$

where

$$\begin{aligned} \overline{M}_R &= \underline{\overline{E}}^R_{S_t} \sum_{i=1}^{N_w} \overline{Mr}_{iS_t} \\ \underline{\overline{Id}} &= \begin{bmatrix} 1/I_{[1,1]} & 0 & 0 \\ 0 & 1/I_{[2,2]} & 0 \\ 0 & 0 & 1/I_{[3,3]} \end{bmatrix} \end{aligned} \tag{3.1.5}$$

where $I_{[1,1]}, I_{[2,2]}$ and $I_{[3,3]}$ are the diagonal entries of the inertial matrix. The rotational velocity $(\dot{\overline{w}}_{GR_{t+\Delta t}})$ is obtained by using an *Euler* explicit approximation;

$$\dot{\overline{w}}_{GR_{t+\Delta t}} = \dot{\overline{w}}_{GR_t} + \int_t^{t+\Delta t} \ddot{\overline{w}}_{GR_t} dt = \overline{w}_{GR_t} + \ddot{\overline{w}}_{GR_t}\Delta t \tag{3.1.6}$$

The rotational matrix at time t, $\underline{\overline{E}}^R_{S_t}$ can be written in quaternion ($\overline{\varrho}$) form as;

$$\underline{\overline{E}}^R_{S_t}(\overline{\varrho}) = \begin{bmatrix} 2\varrho^2_{[0]} + 2\varrho^2_{[1]} - 1 & 2(\varrho_{[1]}\varrho_{[2]} - \varrho_{[3]}\varrho_{[0]}) & 2(\varrho_{[1]}\varrho_{[3]} + \varrho_{[2]}\varrho_{[0]}) \\ 2(\varrho_{[1]}\varrho_{[2]} + \varrho_{[3]}\varrho_{[0]}) & 2\varrho^2_{[0]} + 2\varrho^2_{[1]} - 1 & 2(\varrho_{[2]}\varrho_{[3]} + \varrho_{[1]}\varrho_{[0]}) \\ 2(\varrho_{[1]}\varrho_{[3]} + \varrho_{[2]}\varrho_{[0]}) & 2(\varrho_{[2]}\varrho_{[3]} + \varrho_{[1]}\varrho_{[0]}) & 2\varrho^2_{[0]} + 2\varrho^2_{[3]} - 1 \end{bmatrix} \tag{3.1.7}$$

with

$$\overline{\varrho} = \begin{bmatrix} \varrho_0 \\ \varrho_1 \\ \varrho_2 \\ \varrho_3 \end{bmatrix} = \begin{bmatrix} \sin\frac{\phi}{2} \\ \cos\frac{\phi}{2}\overline{a} \end{bmatrix} \tag{3.1.8}$$

and ϕ and $\overline{a}$ being the rotation angle and axis of the bucket.

Using the second order integration scheme put forward by (Els, 2003), $\overline{\varrho}_{t+\Delta t}$ can be approximated by;

by

$$\overline{\varrho}_{t+\Delta t} = \left[\cos\left(\frac{\Delta t}{2}\dot{w}\right)\underline{\overline{I}} + \frac{\Delta t}{2}\text{sinc}\left(\frac{\Delta t}{2}\dot{w}\right)\underline{\overline{\Omega}}_R\right] \cdot \overline{\varrho}_t \tag{3.1.9}$$

where

$$
\begin{aligned}
\mathrm{sinc} &= \sin\theta/\theta \\
\dot{w} &= \|\dot{\overline{w}}_{GR_t}\| \\
\underline{\overline{\Omega}}_R &= \begin{bmatrix} 0 & -\dot{\underline{w}}_{GR_t} \\ \dot{\overline{w}}_{GR_t} & -\dot{\underline{\overline{\tilde{w}}}}_{GR_t} \end{bmatrix} \\
\dot{\underline{\overline{\tilde{w}}}}_{GR_t} &= \begin{bmatrix} 0 & -\dot{w}[3]_{GR_t} & \dot{w}[2]_{GR_t} \\ \dot{w}[3]_{GR_t} & 0 & -\dot{w}[1]_{GR_t} \\ -\dot{w}[2]_{GR_t} & \dot{w}[1]_{GR_t} & 0 \end{bmatrix}
\end{aligned}
\tag{3.1.10}
$$

Using equation 3.1.7 with the updated quaternion matrix, $\underline{\overline{E}}^{R}_{S_{t+\varrho t+\Delta t}}$ can be calculated and $\underline{\overline{E}}^{S}_{R}$is simply the transposition of $\underline{\overline{E}}^{R}_{S}$. Equations 3.1.1 to 3.1.10 form the basics of the dynamics program. This algorithm runs every time step and updates the bucket's velocities.

3.2 Drag forces and moments

The program above allows walls to have free, as well as fixed, dynamics. The only dynamic effect on the bucket unaccounted for is the drag force. The drag force is applied to the bucket by means of two cables attached to the bucket at hitch points 1 and 2 ($H1$ and $H2$), see figure 3.1 and denoted by $FC1$ and $FC2$, respectively.

3.2.1 Drag Force

Various attempts were made to model the drag chains using particles in *PFC*. The particles where bonded together using parallel bonds. The bond stiffness was calculated using steel stiffness. The time required was impractical, however, due to the reduced time step caused by the increasing stiffness of the particles in the chain, see equation 2.2.1.

A new approach was needed. The program used to calculate the walls' dynamics could be modified to incorporate the drag force. A simple virtual mathematical spring model was used to determine the magnitude of the applied drag force, which allowed chains to have the correct stiffness without affecting the critical time step. The largest drawback of this spring model was that the chain could no longer interact with granular material. The virtual mathematical spring model is described below.

Using the cable speed (CS) and the length the cable shortens in a time step, the drag force can be calculated as follows;

$$\Delta cs_{i_t} = (CS\Delta t) \cdot \vec{HD}_i \tag{3.2.1}$$

where $\vec{HD}_i$ is the unit vector from hitch point (i) to the origin of the drag force (i).

From equation 3.1.1 the velocity of hitch point (i) can be written as;

$$\dot{\overline{x}}_{H(i)S} = \dot{\overline{x}}_{GS} + \underline{\overline{E}}_R^S\Big(\dot{\overline{w}}_{GR} \times (\overline{x}_{H(i)R} - \overline{x}_{GR})\Big) \tag{3.2.2}$$

Projecting and integrating $\dot{\overline{x}}_{H(i)}$ onto $\vec{HD}_i$ yields the displacement of hitch point (i) per time step along $\vec{HD}_i$;

$$\Delta csp_i = \Delta t\big(\dot{\overline{x}}_{H(i)S} \cdot \vec{HD}_i\big) \tag{3.2.3}$$

The accumulated length change of the drag cable (i) can now be calculated;

$$\Delta CS_{i_{t+\Delta t}} = \Delta CS_{i_t} + (\Delta cs_{i_t} - \Delta csp_{i_t}) \tag{3.2.4}$$

Using the cable stiffness (k_c) the cable force can be calculated

$$\overline{FC_{(i)}} = \begin{cases} k_c \Delta CS_{i_{t+\Delta t}} \cdot \vec{HD}_i & \text{if } \Delta CS_{i_{t+\Delta t}} > 0 \\ 0 & \text{if } \Delta CS_{i_{t+\Delta t}} \leq 0 \end{cases} \tag{3.2.5}$$

The total drag force $(\overline{FC_T})$ is, therefore;

$$\overline{FC_T} = \overline{FC_{(1)}} + \overline{FC_{(2)}} \tag{3.2.6}$$

With the drag force calculated, equation 3.1.2 is modified by adding the drag force. Equation 3.1.2 becomes;

$$\ddot{\overline{x}}_{GS_t} = \frac{(\sum_{i=1}^{N_w} \overline{Fr}_{iS_t} + \overline{FC}_{1S_t} + \overline{FC}_{2S_t} + \overline{Fb}_t)}{(\sum_{i=1}^{N_w} m_i)} \tag{3.2.7}$$

where $\overline{FC}_1$ and $\overline{FC}_2$ represent the drag forces from cables 1 and 2, respectively.

The newly added forces also applied moments to the bucket. These moments need to be added manually. Equation 3.1.5 thus becomes;

$$\overline{M}_{R_t} = \underline{\overline{E}}_{S_t}^R\Big(\sum_{i=1}^{Nw} \overline{Mr}_{iS_t} + \overline{FC}_{1S_t} \times \overline{x}_{H1S_t} + \overline{FC}_{1S_t} \times \overline{x}_{H2S_t} + \overline{Fb}_{S_t} \times \overline{x}_{b_t}\Big) \tag{3.2.8}$$

where $\overline{x}_{H1S}$, $\overline{x}_{H2S}$ and $\overline{x}_{b_t}$ are vectors from the bucket's centre of gravity $(\overline{x}_G)$ to hitch points 1, 2 and the body force, respectively, see figure 3.1.

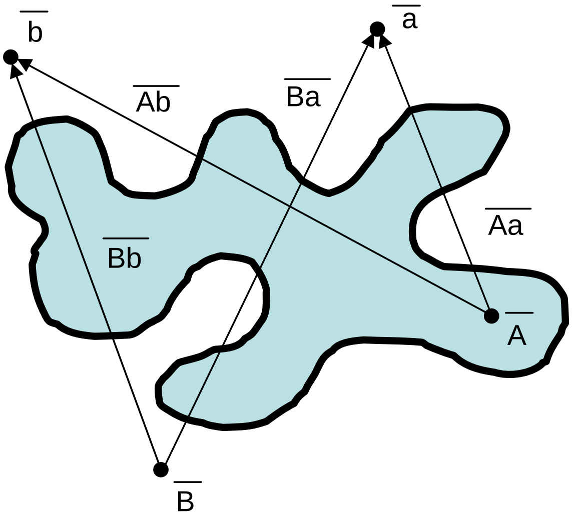


Figure 3.2: Mass flow algorithm - vector test

3.3 Mass flow

A major advantage of a numerical simulation over an experimental one is the ability to calculate the mass of material within the bucket. In experimental simulations it is very difficult, and beyond the scope of this thesis, to measure the mass in the bucket at any given time.

In this instance, a *FISH* algorithm was written to calculate the mass within the modelled bucket at any given time. The algorithm used a *CAD* model in *STL* format of the volume of the bucket. The *STL* was then converted into a virtual collection of walls that where attached to the bucket. These walls, like the chains above, are considered 'virtual' because they do not really exist, except as required mathematical concepts.

The algorithm checks every ball to see whether it is inside the defined volume. Using the ball's centre of gravity, a vector can be generated in a predefined direction. If the vector intersects the volume an even number of times, the point is considered to be outside the volume. If the vector intersects the volume an odd number of times, the point is considered to be inside the volume.

Figure 3.2 demonstrates this principal. Consider point $\overline{A}$ and two random vectors $\overline{Aa}$ and $\overline{Ab}$ from $\overline{A}$ to $\overline{a}$ and $\overline{b}$, respectively. Vector $\overline{Aa}$ intersects the volume three times and vector $\overline{Ab}$ intersects the volume once. According to the algorithm, point $\overline{A}$ must, therefore, lie inside the volume. The test can be corroborated using point $\overline{B}$, where the number of intersections will always be even and, therefore, point $\overline{B}$ lines outside the volume.

The number of times that a given vector will intersect the volume is calculated as follows. The volume is generated using a *STL*, which is made up

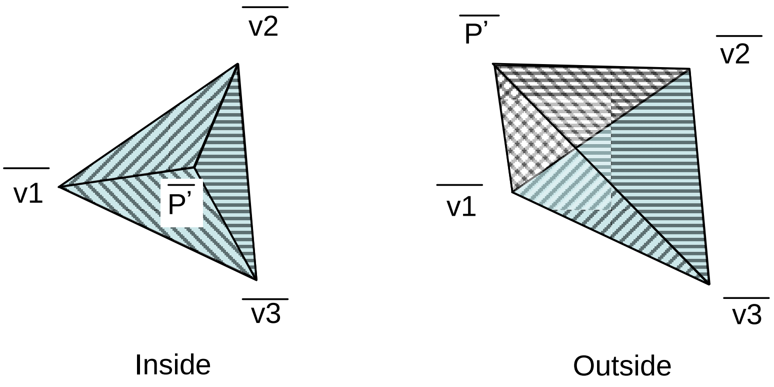


Figure 3.3: Mass flow algorithm - intersection test

of a collection of triangles. A random vector is generated from point $\overline{P}$. The number of times that the vector intersects the volume can be considered equal to the number of triangles making up the volume that the vector intersects. The point where the vector intersects the plane defined by the three vertexes ($\overline{v1}, \overline{v2}, \overline{v3}$) of the triangle is defined as $\overline{P'}$. Four different triangles can be generated using the three vertexes ($\overline{v1}, \overline{v2}, \overline{v3}$) and point $\overline{P'}$, namely, triangles v1v2v3, v1v2P', v1v3P' and v2v3P', see figure 3.3. If point $\overline{P'}$ intersects the triangle, the sum of the areas of triangles v1v2P', v1v3P' and v2v3P' will be equal to the area of the original triangle v1v2v3.

Chapter 4

Numerical simulation optimisation

When this thesis was proposed, it was believed that the majority of available time would be spent tweaking the simulated material properties to achieve better results. As the project progressed, however, it became clear that the time needed to run a single simulation would dramatically limit the number of simulations that could be run. Consequently, different approaches that could reduce simulation time where investigated, together with the effects of the assumptions made by each of these approaches.

The total simulation time can be directly reduced by either increasing the time step or decreasing the cycle time.

4.1 Decreasing the cycle time

The cycle time is defined as the time taken to run one complete cycle of the *DEM* algorithm. The cycle time is primarily dependant on the simulation setup and parameters.

In every cycle, the following needs to happen:

- The forces acting on all the particles need to be updated;
    - Each particle must loop through its contact list and, using the defined contact model, determine the magnitude of the resultant force and moment acting upon itself;
- Using the forces specified above, each particle's acceleration and velocity must be calculated;
- Once the particle's velocity and acceleration have been ascertained, its new position can be calculated.
    - The contact list for each particle is then updated.
- Finally, using the new updated wall and ball positions, each ball and wall is assigned to a cell (the system is remapped), see section 4.1.1.7.

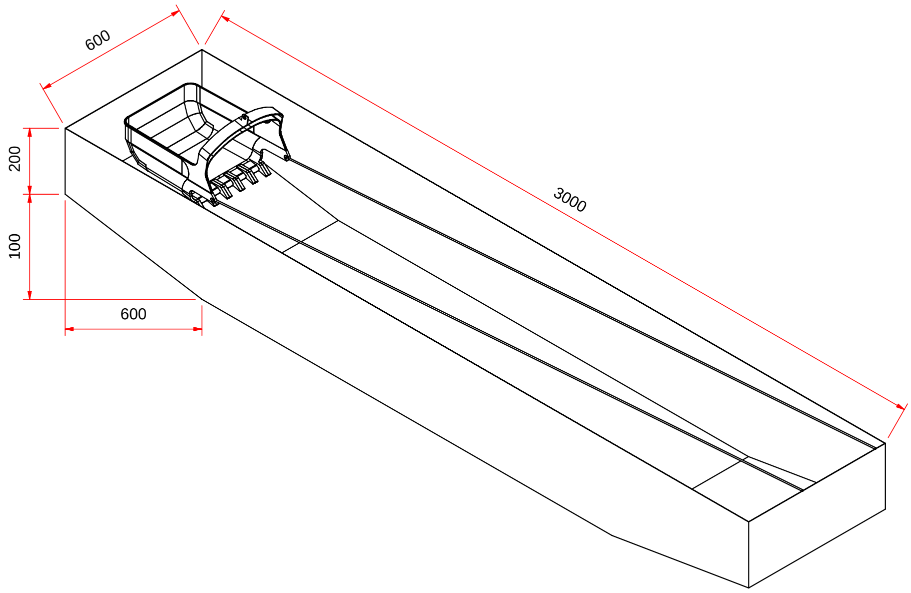


Figure 4.1: Parabolic drag bed

Therefore, in order to reduce the cycle time, either the number of entities (balls and walls) needs to be decreased or the time taken to update the particles needs to be reduced.

4.1.1 Reducing the total number of particles

Reducing the number of particles in the simulation is the most obvious and most effective solution. However, caution must be taken when reducing the number of particles because this can lead to the creation of unwanted boundary effects. Reducing the particles and walls reduces the number of calculations performed per cycle.

4.1.1.1 Drag bed design

A drag bed is considered optimised when its volume is as small as possible, without allowing boundary effects to unduly influence the simulation. Experimental data revealed that the bucket follows a parabolic trajectory while filling. The drag bed was modeled using this information and simplified to reduce the number of balls. The drag bed can be seen in see figure 4.1. This shape requires 25 percent less particles than a standard rectangular drag bed, assuming that the particle geometry remains constant.

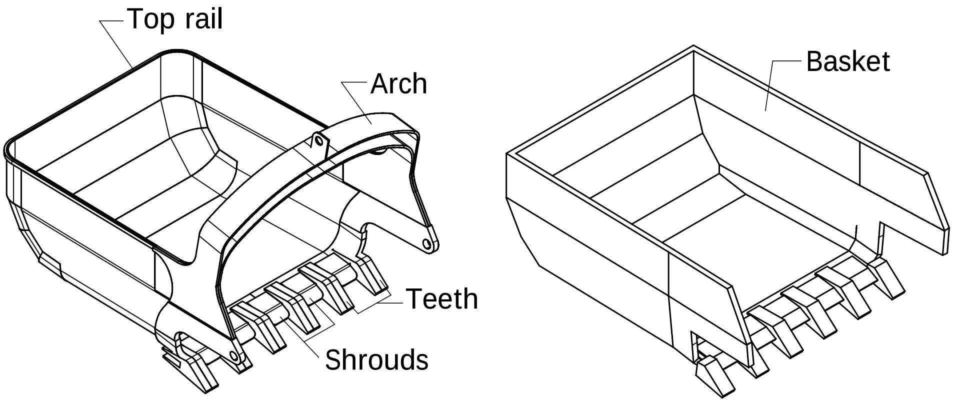


Figure 4.2: Bucket simplification

4.1.1.2 Bucket simplification

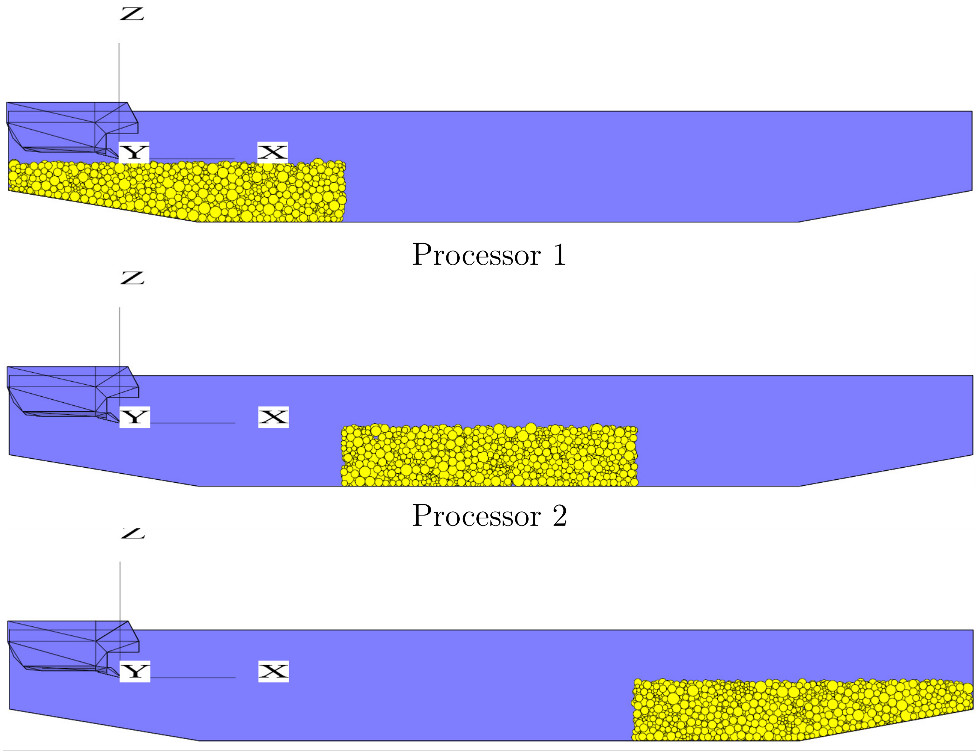
PFC was not written or optimised for multiple or moving walls. Since the simulation is dependant on moving walls, the only optimisation that can be performed is a reduction in the number of walls. Consequently, an interface was written to convert *STL* files into *PFC* code. The bucket was simplified by:

- Replacing rounds or fillets with chamfers or removing them completely;
- Giving the bucket basket a uniform thickness;
- Removing the arch and top rail;
- Closing the gap that normal exists between the teeth and shrouds, see figure 4.2.

The simplifications were restricted to areas that would have little or no effect on the flow of material in or around the bucket. The simplified bucket shape can be seen in figure 4.2. The new bucket allowed the number of walls to be reduced from 5000 to 520.

4.1.1.3 Multiple processors

PFC is capable of using multiple processors to solve a simulation, see section 2.7. The simulation is divided into partitions along the x-axis, see figure 4.3. Each processor only solves the partition assigned to it, which greatly reduces the cycle time. Particles that are on partition boundaries are solved by the master processor. However, as the number of slave processors increases, the amount of time spent solving boundary particles also increases, resulting in an



Processor 1

Processor 2

Processor 3

Figure 4.3: Multiple process - simulation partition

increase in cycle time as well. Therefore, the number of particles and the size of the partition boundaries are the major factors that determine the optimal number of processors to use.

Itasca recommends using a new processor for every hundred thousand elements (balls, walls and bonds). Multiple processor simulations do not accommodate clump logic and, therefore, particles need to be made up of bonded particles. The complexity of the particles determines the number of bonds required. Bonds add additional calculations to each cycle. The bond stiffness influences the time step calculation and can reduce the time step if the stiffness is higher than that of the particles. The current simulation had around one hundred thousand balls, with around eighty thousand bonds and, therefore, should have, technically, been split between two or three processes. Clumping the particles would remove the need for the eighty thousand bonds, nonetheless, reducing the number of elements and allowing the simulation to run on a single processor.

Simulations were run on multiple processors, using parallel bonds, and on a single processor, using clump logic, to measure the amount of time needed to complete every 0.2 seconds of simulation time. The master processor used in the parallel simulation was the same processor used to run the single sim-

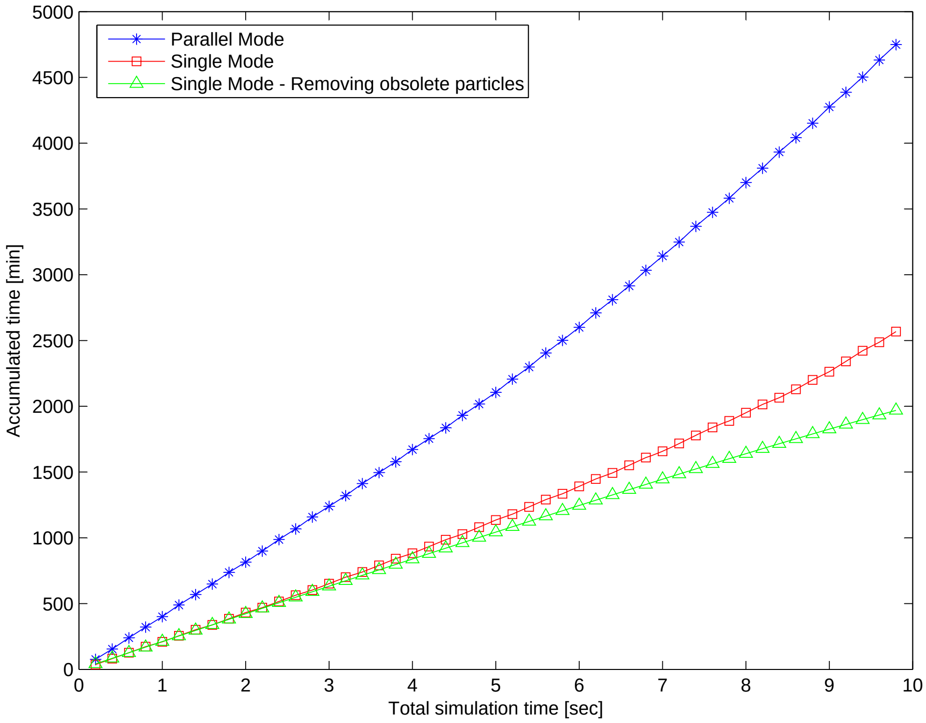


Figure 4.4: Clump logic vs. parallel bonds

ulation. The parallel simulation used three processors and was divided along the drag (x) axis. The parallel bond stiffness had to be at least two orders higher than that of the particles, in order to keep the bonded particles ridged. The time taken to complete a simulation can be see in figure 4.4.

The results reveal that, despite the increase in processing power available to the parallel bonds method, clumps logic was faster because of the greater cumulative negative effect of the bonds and network communication used in the parallel bonds method. Nonetheless, the interval at which clump logic updates the inertial properties of the clumps can lead to inaccuracies. To investigate these effects the bucket's xz trajectory and pitch for each simulation was recorded, see figure 4.5.

Analysing the results of figure 4.5, a slight discrepancy between clump logic and parallel bonds is visible. The effect, however, is small enough to be considered as noise arising from slight changes in the configuration of the drag bed, time step or rigidity of the clumps.

4.1.1.4 Removing obsolete particles

The particles behind the bucket no longer have any real effect on the simulation. These particles can be removed to decrease the total number of particles and, in turn, decrease the cycle time. This has little or no effect in parallel

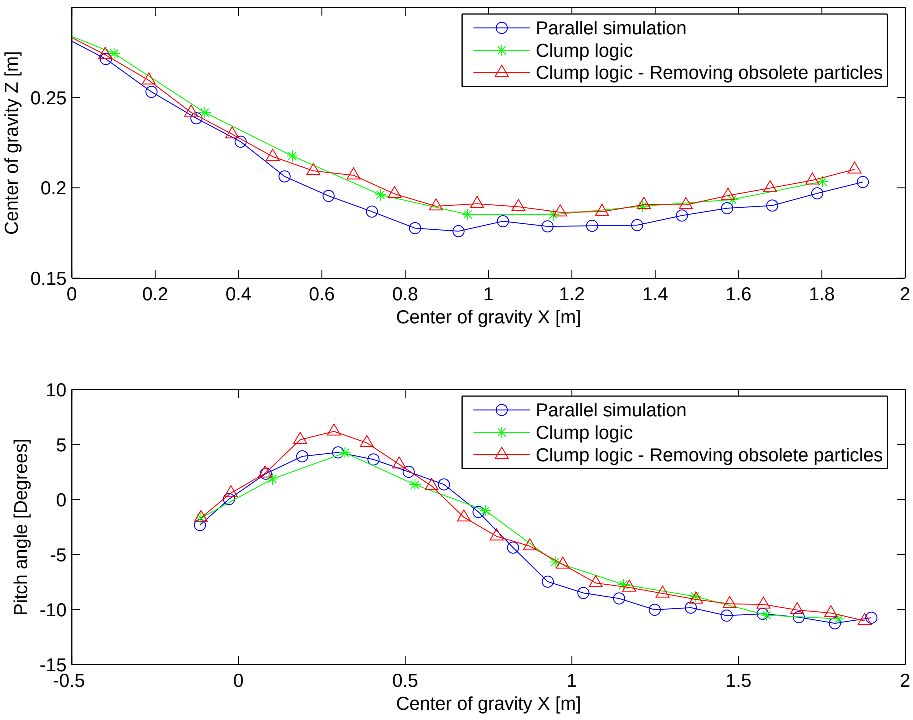


Figure 4.5: Simulation accuracy

simulations because the cycle time is equal to the cycle time of the slowest processor and the last processes, furthest along the x-axis in the drag direction, will always contain particles. The boundaries can be redefined to divide the simulation along the y axis (normal to the soil surface), but this would also greatly increase the size of the boundaries and, consequently, the cycle time.

Removing particles in single processor simulations decrease the cycle time, but when a particle is removed from the simulation it forces a cell remapping (see section 4.1.1.7) to occur, which increases the time again. Therefore, particles were only removed at fix intervals to reduce the resultant remapping time.

The gradient of the results, seen in figure 4.4, represents the change in time taken to 0.2 seconds of simulation time. Basic *DEM* theory predicts that the time taken to run 0.2 seconds anywhere in the simulation will be the same, but the gradients of the parallel and clump simulations increased as the simulation progressed. This can be contributed to an increase in cell remapping since the number of moving particles also increases as the simulation progresses.

As the parallel simulation progresses, the balls both inside and being pushed by the bucket move between processors. As a result, the number of balls in the processors increases along the x-axis. The increased number of balls in a single processor leads to an increased cycle time and explains why the gradient

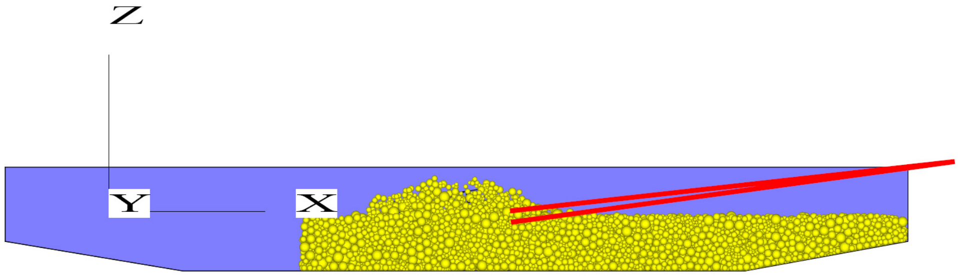


Figure 4.6: Removing obsolete particles

of the parallel increases the steepest.

The effects of removing the obsolete particles can clearly be seen in figure 4.4. As the simulation progresses, the same increasing gradient can be observed, since the number of moving particles remains the same and so does the frequency of cell remappings. However, as the simulation continues to progress further, the number of balls decreases, reducing the cycle time and the time taken for the system to remap. The combined effect of this reducing the total time required to run a simulation.

4.1.1.5 Soil cell method

The effects of removing the particles behind the bucket became more evident as the simulation progressed and the number of particles continued to diminish. Using this theory, if the number of particles in the drag bed can be kept to a minimum at all times, the simulation could, theoretical, reach the optimum minimum number of particles. The soil cell method was created to take advantage of this possibility. A soil cell can be thought of as a slice of the drag bed with a finite length. Soil cells can be generated in front of the bucket and removed behind the bucket as it moves through the drag bed. The soil cells needed to be large enough to be representative samples of the drag bed, but small enough to keep the number of particles in the simulation to a minimum, see figure 4.7.

The soil cells deleted behind the bucket have little to no effect as they no longer contribute to the simulation. The soil cells generated in front on the bucket have a much larger effect. Simulations were run to determine the distance needed in front of the bucket to allow the new particles to settle and to prevent undesired boundary effects. To determine the minimum distance, the magnitude of the forces acting on the boundary walls was recorded. If the force acting on the boundary wall increased after a new soil cell was generated, the distance between the bucket and the boundary wall was insufficient. It was determined that a minimum of two bucket lengths was required.

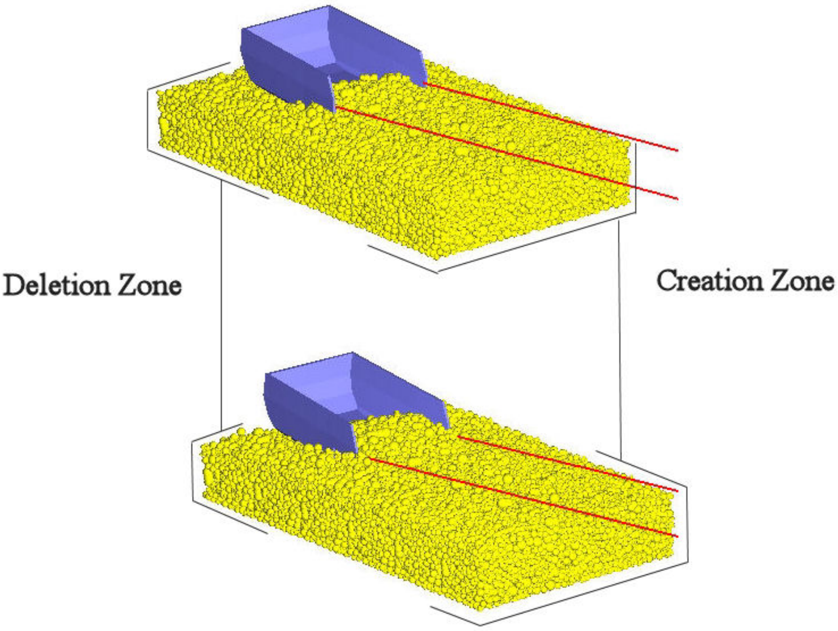


Figure 4.7: Soil cell method

The optimum soil cell had a length of approximate four bucket lengths, namely;

- one bucket length behind the bucket;
- one bucket length for the bucket itself;
- two bucket lengths in front of the bucket.

4.1.1.6 Symmetry

Experimental data confirmed that the bucket trajectory could be approximated in two dimensions very accurately. Reducing the degrees of freedom of the bucket allowed a symmetry plane to be constructed, dividing the simulation in half along the drag bed. The symmetry plane was given the same stiffness as the particles, but zero friction, to minimise undesired effects.

4.1.1.7 Reducing time taken for contact detection and updating

The methods above all focus on reducing the number of particles or walls, thereby reducing the number of calculations per cycle. Another way to reduce the cycle time is to reduce the number of calculations that need to be performed between cycles.

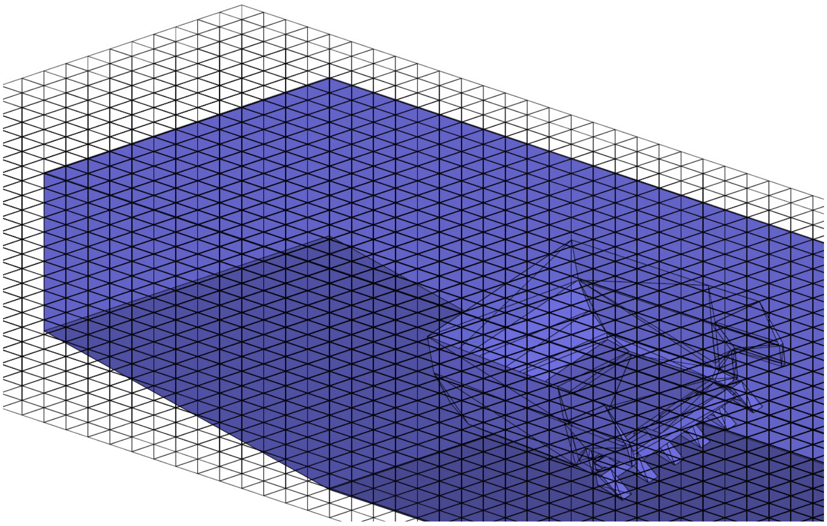


Figure 4.8: *PFC* cell spacing

Although these do not strictly form part of the *DEM* cycle, they happen on a cyclic basis and are virtually independent of particle geometry and can, therefore, be considered part of the cycle time. The user has little to no control over most of these processes, but some have parameters that can be set, namely, the cell spacing.

PFC divides the simulation into a three dimensional matrix of cells, see figure 4.8. Any entity (particle or wall) that lies inside the cell is assigned to that cell. Entities can belong to more than one cell. To determine which particles are in contact, the contact detection algorithm needs only to search within the cells that the particles occupy. This greatly reduces the time needed to update the entities contact lists.

The number of cells generated is a function of the maximum number of balls in the simulation and remains constant throughout the simulation. However, the number of cells can be specified by overwriting the defaults. Smaller cell size reduces the number of particles in each cell and, consequently, the time taken to check for contacts. Conversely, bigger cell size reduces the number of cells, but increases the number of particles in each cell, thus, increasing the time taken to check for contacts.

When a particle moves a predefined distance, it forces *cell remapping* to occur. This predefined distance is a function of the cell and particle size. When *cell remapping* occurs, the cells are updated and *PFC* needs to reassign all the entities to the new cells. This process takes large amounts of time.

If it occurs too often, *Cell remapping* can become a leading factor in the total simulation time. Therefore, to optimise the simulation a balance is needed, where the cells are small enough to decrease the cycle time, but large enough to reduce the number of *cell remappings* required.

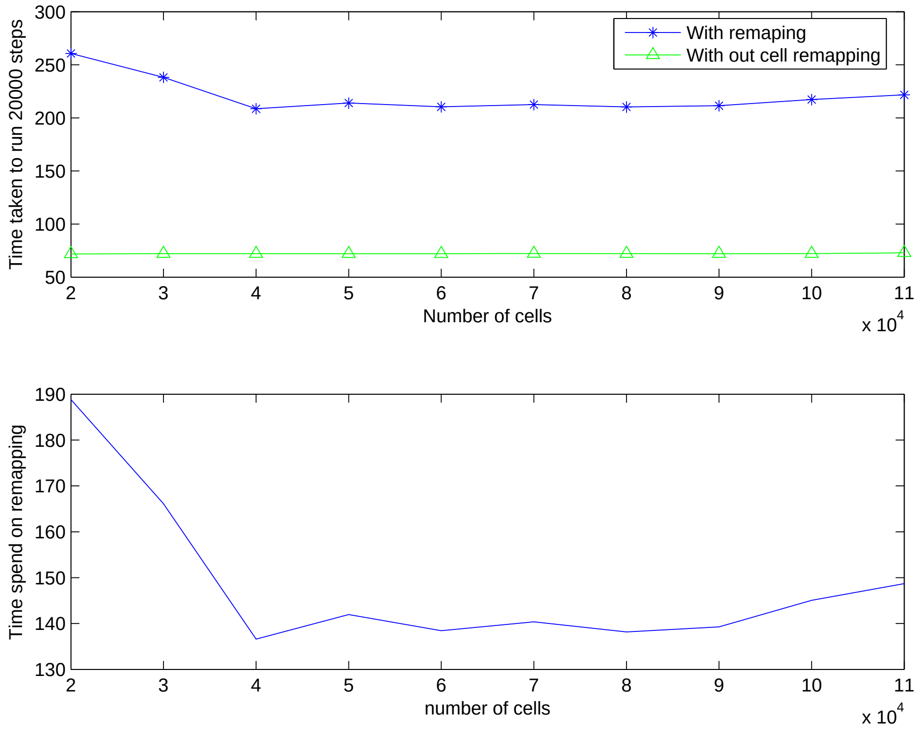


Figure 4.9: Effect of cell spacing on simulation time

The number of *cell remapping* that occur within the simulation is also dependant on the velocity of the particles. Before any attempt to optimise the cell spacing could be made, a way of measuring the effect of the number of cells on the simulation had to be ascertained. To achieve this, simulations were run with different numbers of cells, once with the bucket at rest and once with the bucket in motion. The time taken to run 20000 cycles was recorded for each simulation. The results can be see in figure 4.9. The time spent on system *remapping* can be estimated by subtracting the total simulation time of the system in motion from that of the stationary simulation, second figure of 4.9.

These results are simulation dependent and can, therefore, only be applied to similar simulations. The results show that, when the particles remained stationary, the time taken to complete the 20000 cycles remained constant and once the contact list where established little to no *remapping* occurred. However, when the system was in motion, the amount of *cell remapping* that occurred was dependent on the number of cells and the negative effect of having too many cells became clearly evident.

4.2 Increasing the time step

Different methods for increasing the cycle time were discussed above. These methods are dependant on the setup and geometry of the simulation and, therefore, many different possibilities exist. An alternative method for decreasing the simulation time is to increase the magnitude of the time step. This can be done by adjusting safety factors or by changing the particle parameters.

4.2.1 Safety factors

Every predefined number of cycles, *PFC* calculates a new critical time step. This time step is then multiplied by a safety factor before being used. This safety factor can be manually adjusted. If the safety factor becomes too large, the numerical error of the *Euler* explicit approximation can become large enough to adversely affect the simulation. *Euler* explicit approximation is only conditionally stable.

A simulation was performed where a ball was dropped from 1 metre, with zero damping, and the ball's height and energy were recorded. If the *Euler* explicit approximation had no error and the time step was sufficiently small, the energy in the simulation would remain constant and the ball would bounce to the exactly the same height. It was decided that a two percent error band would provide sufficient accuracy. Simulation were run at different safety factors, for five and half seconds and the change in energy recorded. The safety factor was gradually reduced until the energy fluctuations where within the predefined error band. From figure 4.10 the safety factor of 0,6 was considered to be acceptable.

4.2.2 Particle stiffness

The time step is dependent on the contact model used. The time step for each model is, moreover, dependant on the particle stiffness. For simplicity, only the linear contact model is analysed here, but the same principles apply to the other contact models as well. From equation 2.2.1 the critical time step is;

$$\Delta t_c = \min \left\{ \begin{array}{ll} \sqrt{m/k_n} & \text{translation} \\ \sqrt{I/k_s} & \text{rotational} \end{array} \right. \tag{4.2.1}$$

Based on the equation above, the two parameters that are responsible for the magnitude of the time step are the particles mass and normal stiffness. The time step can, therefore, be altered by adjusting the particle stiffness, density or size. Using the linear contact model, the force generated between two particles in contact is;

$$\overline{Fc}_n = k_n \overline{\Delta n} \tag{4.2.2}$$

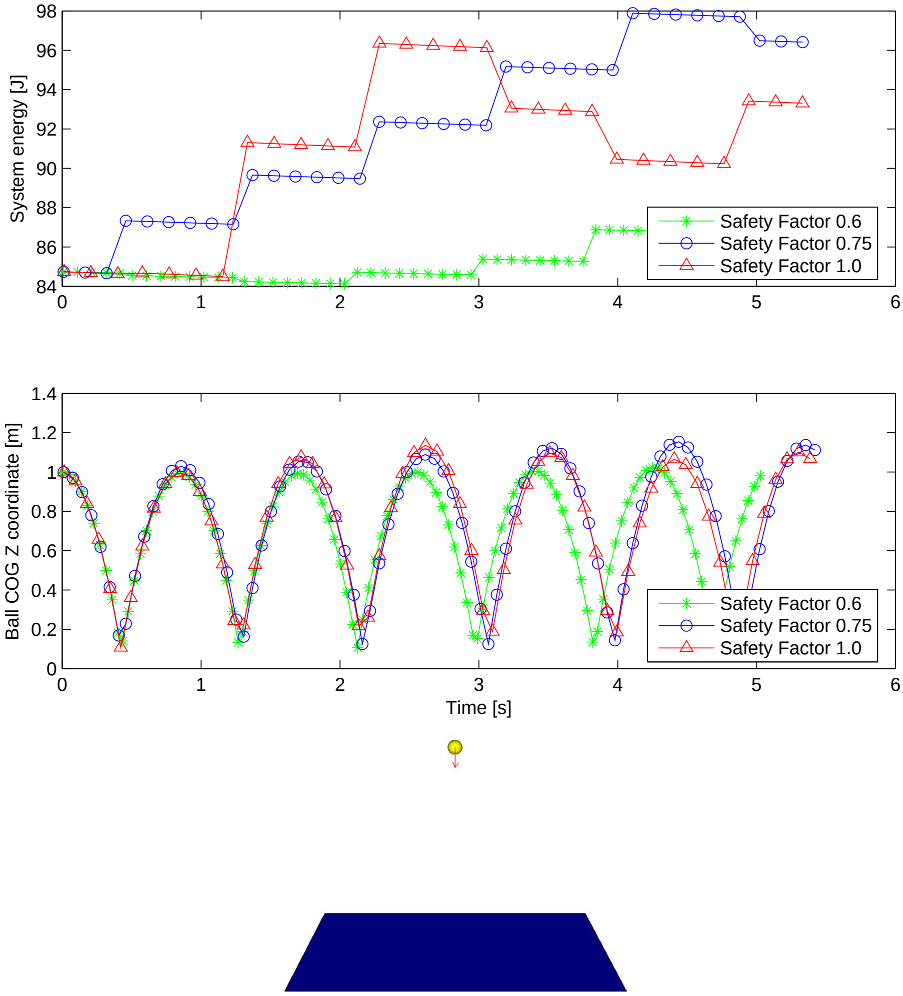


Figure 4.10: Effect of safety factor on simulation accuracy

Since the above equation is linear, any changes in the magnitude of the stiffness will directly affect the magnitude of the force. The force applied by the system on the particles remains constant and the overlap must, therefore, increase. Table 4.2.2 was constructed to investigate the effects of reducing the particle stiffness on the linear contact model.

The peak drag force measured within the experimental tests was around $300N$. Using a calibrated particles stiffness of 1.750×10^7, see chapter 5 for further detail, the overlap is;

$$\Delta n_0 = \frac{||\overline{Fc}_n||}{k_{n0}} = \frac{300N}{1.75 \times 10^7} = 1.7143 \times 10^{-5} m \tag{4.2.3}$$

where k_{n0} and Δn_0 represent the calibrated or datum stiffness and overlap, respectively.

k_n	$1-\frac{k_n}{k_{n0}}[\%]$	$\Delta t[s]$	$\frac{\Delta t_0}{\Delta t}[\%]$	$\Delta n[m]$	$\frac{\Delta n-\Delta n_0}{2R_{av}}[\%]$
5×10^5	97.14	1.41×10^{-3}	16.90	6×10^{-4}	2.33
1×10^6	94.29	1.0×10^{-3}	23.90	3×10^{-4}	1.13
5×10^6	71.43	4.47×10^{-4}	53.45	6×10^{-5}	0.17
8×10^6	54.29	3.54×10^{-4}	67.61	3.75×10^{-5}	0.08
1×10^7	42.86	3.16×10^{-4}	75.59	3×10^{-5}	0.05
1.25×10^7	28.57	2.83×10^{-4}	84.52	2.4×10^{-5}	0.03
1.5×10^7	14.29	2.58×10^{-4}	92.58	2×10^{-5}	0.01
1.75×10^7	0	2.39×10^{-4}	100	1.71×10^{-5}	0

Table 4.1: Particle stiffness effect on simulation time

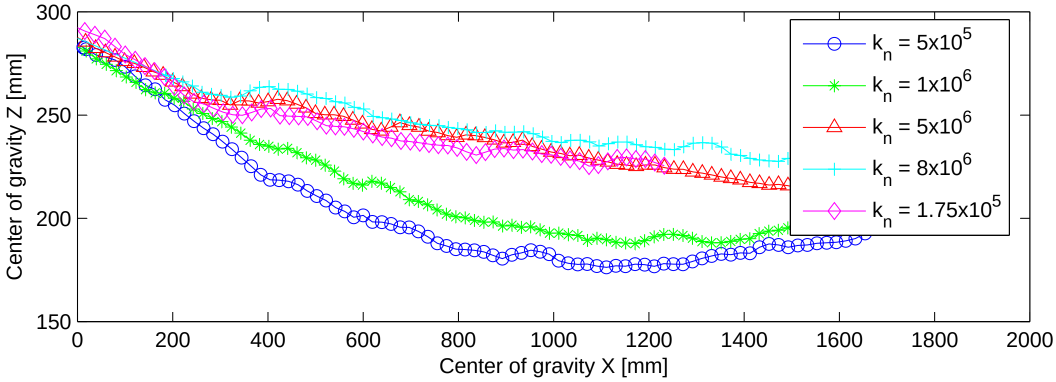


Figure 4.11: Particle stiffness effect on bucket trajectory

Using equation 4.2.1 and 4.2.3, with a force of $300N$ and unit mass, table 4.2.2 can be generated. R_{av} is the average particle diameter measured during the material calibration.

The last column of table 4.2.2 shows the increase in the overlap relative to average particle diameter ($2R$). This column gives an indication of the effect of changing the particle stiffness and the fourth column shows the fractional decrease in cycle time

Table 4.2.2 is generated using the interaction between two identical particles. In a simulation, the stiffness and mass are functions of all the balls and walls and are, therefore, more difficult to calculate.

To compare the effect of the particle stiffness, simulations were run using the particle shape and size distribution measured in section 5. The experimental setup was identical, with only the stiffness changing. The xz trajectory of the bucket's center of gravity was plotted for each simulation.

Before each simulation was initiated, the particle stiffness was changed and the system allowed to return to static equilibrium. This slightly changed the bed setup and led to slight deviations in the results, but was unavoidable.

Figure 4.11 illustrates the resulting convergence when particle stiffness is equal or greater than 5×10^6. According to table 4.2.2, this allows the time step to be doubled. The biggest drawback to this approach is the amount of data needed to predict the effects of changing the time step. The results above are only valid for the linear contact model.

4.2.3 Contact Models

The formulae used to calculate the critical time step is dependant upon the contact model used. This not an ideal optimisation method because certain contact models predict the behavior of soil better than others. The contact model used should be selected on this basis rather than simulation time concerns. The linear contact model is not the most accurate model, but it takes half the amount of time of other models, like the viscous damping model. Thus, it makes sense to use the linear contact model to run and test preliminary models, substituting it for a more advanced model at a later stage.

4.3 Conclusion

This chapter evaluates the different approaches that where used in an attempt to decrease the time taken to run a simulation. It also demonstrates some of the potential errors that can be made when employing the different approaches.

The soil cell method, in conjunction with a symmetry plane, was used for the purposes of this thesis since it more than quartered the time taken to run a simulation, while keeping an acceptable level of accuracy. Section 4.2.2 showed that the particle stiffness could be reduced without creating a noticeably adverse effect on the simulation. However, it was decided that the reduced simulation time due to the soil cell method and symmetry plane was already sufficient and so it would be preferable not to change any of the particle parameters if possible.

Chapter 5

Granular material parameters

Having completed the simulation geometry, the drag bed needed to be filled with balls. The balls, in turn, needing to be calibrated such that they could physically model the overburden.

It has always been difficult to determine granular material properties. Unlike steels and composites, the exact composition and particle orientation of a granular material can be very difficult to measure. The fluctuations in the material's properties are also much larger, making it difficult to determine the properties by empirical means.

For these and other numerical reasons, granular material properties needed to be calibrated for each different material and application. *DEM* software packages, like *PFC*, model a granular system by using an assembly of particles that mimic the behavior of the granular materials. This means that the microscopic properties of the granular material and that of the *DEM* model do not always coincide. Consequently, the *DEM* models are calibrated using macroscopic properties, such as angle of repose and bulk stiffness.

This chapter will discuss the methods used to calibrate the granular material. The value of these parameters will change according to the type of contact model used, but the calibration method remains the same. In order to fully describe the granular medium, in *PFC*, the following material properties are needed;

- the shape of the particles present in the granular material;
- distribution of particle shapes within the granular material;
- particle size distribution within each particle shape;
- ρ density (kg/m^3);
- k_n, k_s normal and shear stiffness (N/m);
- μ friction coefficient;
- damping.

5.1 Shape

The material used to simulate the overburden in the experimental setup was normal crushed rock from a roller mill with a 25 mm aperture between rollers. A size and shape distribution was needed to perform a *DEM* simulation. A random sample of 300 rocks was taken and classified into four distinct particle shapes, see figure 5.1. These shapes were kept as simple as possible, while ensuring that every rock in the sample belonged to one of the particle shapes. The particle shapes where:

- Particle 1: long rectangular type particle with a relationship between the sides greater than 2;
- Particle 2: pyramid shape particle;
- Particle 3: flat particle where the relationship between the length and width is less than 2 and the relationship between the height and length is less than 0.5;
- Particle 4: particles that can be more accurately described as round than as one of the other particle types.

5.2 Distribution

A rock sample was taken and classified according to the above shapes. The number of particles belonging to each shape were recorded and a particle shape distribution was obtained, see figure 5.2. This data was then used to ensure that the same relationship between particle types was maintained when generating particles for the *PFC* model.

5.3 Size

The size distribution of each particle shape was obtained with a sorting machine. The sorting machine consisted of a series of sieves mounted on a vibrating table, see figure 5.3. The results can be see in figure 5.4.

A mathematical distribution function was created to represent each particle size distribution. Due to the limited number of mesh sizes available, linear interpolation was used between adjacent mesh sizes. The linear interpolation was based on the assumption that the particle number at each mesh size was know and that no assumptions could be made about the change in particle size between the two consecutive mesh sizes. Combing the particle type and size distributions made all the data needed to generate numerical particles available.

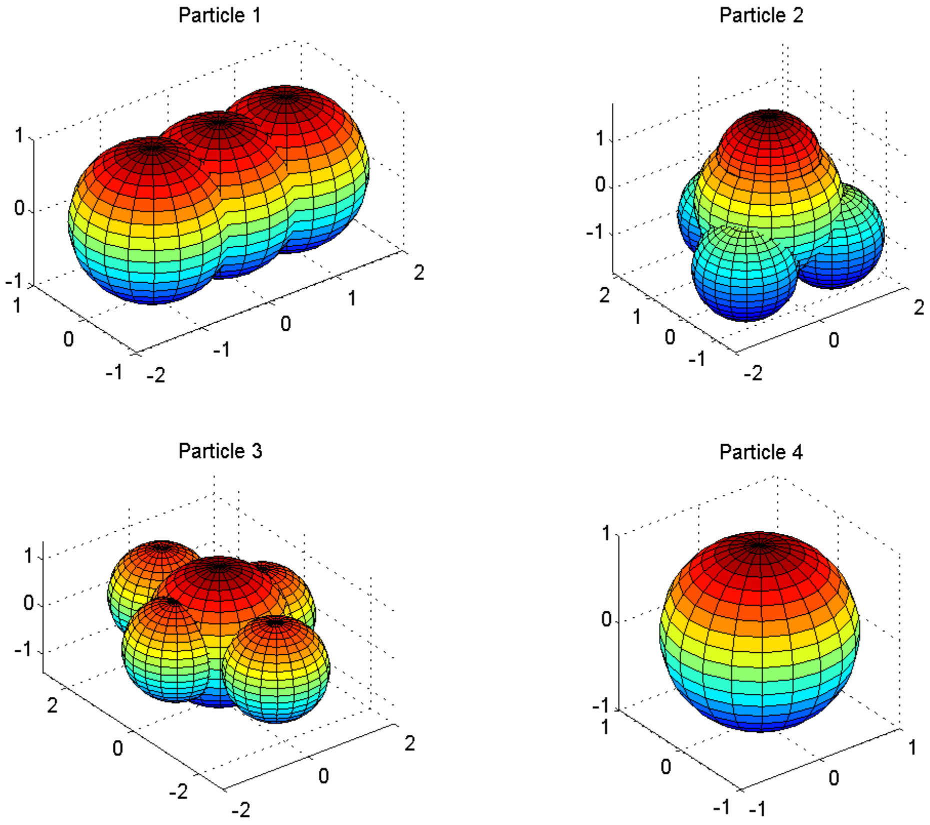


Figure 5.1: Particle shapes

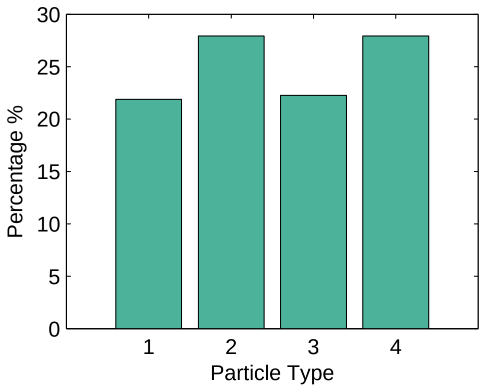


Figure 5.2: Particle distribution

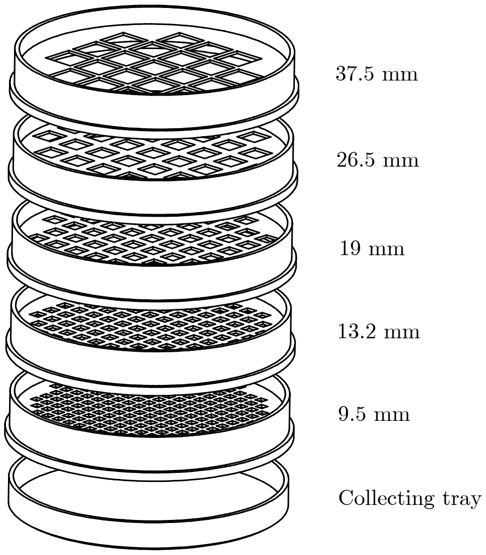


Figure 5.3: Sorting machine

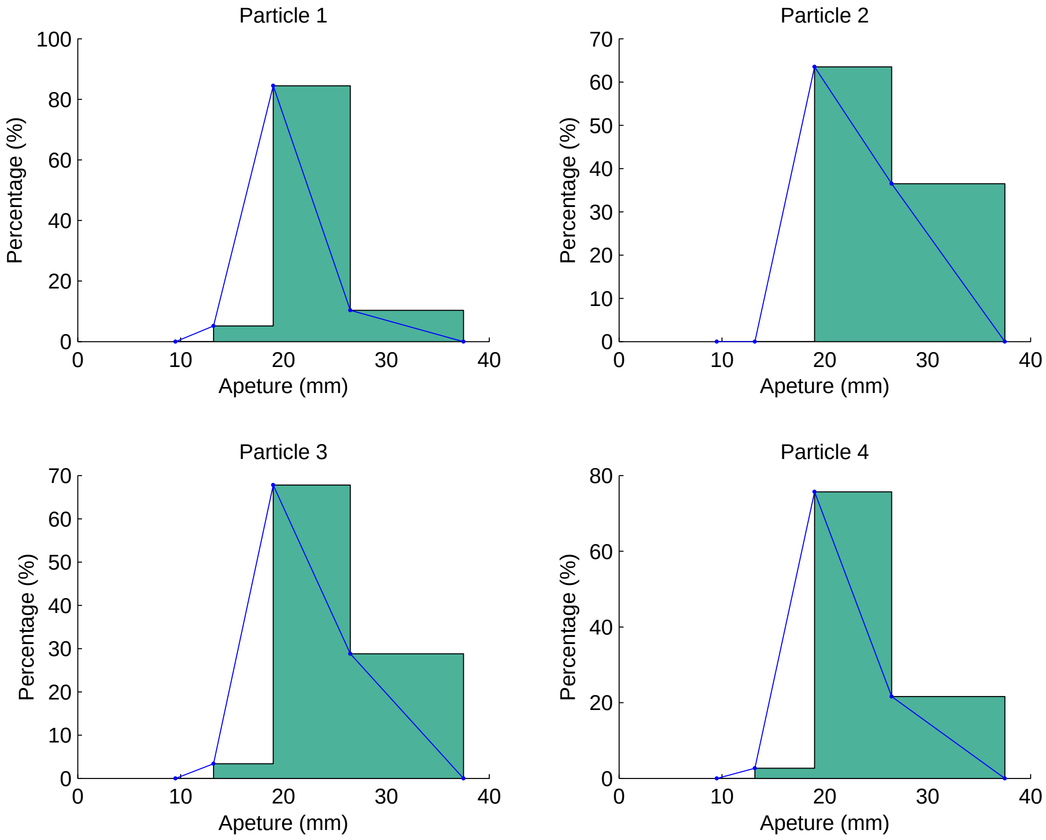


Figure 5.4: Size distribution (25mm rock)

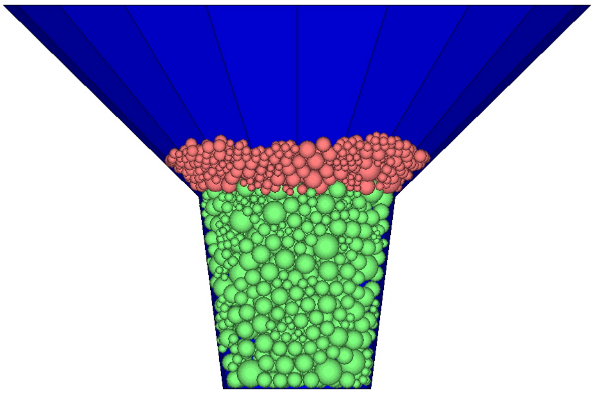


Figure 5.5: *PFC* density calibration model

5.4 Density

The remaining properties are dependent on the particle shape and distribution and, thus, can not be measured directly from the granular material. In order to obtain these properties, the calibration has to be performed on a macroscopic scale.

The density is determined before the other remaining properties since it is virtually unaffected by changes in k_n, k_s and μ.

5.4.1 Experimental method

A container with a known volume and mass was filled with the sample rocks. The container was then weighed, the weight of the bucket subtracted and divided by the volume of the bucket to obtain the bulk density.

5.4.2 Numerical method

An identical container was generated numerically in *PFC*. Particles with the previously determined shape and size distribution were then generated and allowed to fill the container. Once the particles had settled under gravitation, the mass of the particles occupying the container was calculated, see figure 5.5.

The density of the particles was then adjusted and the system was allowed to resettle. This process was repeated until the calculated mass was equal to experimental mass and, thus, the bulk densities were also equal.

5.5 Normal and shear stiffness

The normal bulk stiffness (K) was measured using two different tests, namely, a triaxial and confined compression test. For the linear contact model, the individual rock particle used can be considered as isotropic and an assumption is made that the normal and shear stiffnesses are equal, (Itasca). A sample of the granular material was compressed, within a confined container, with a predetermined force and then relaxed. A graph was generated of applied force versus displacement. When the granular material was initially compressed, the rocks moved and rearranged themselves to fill the voids. The compression and relaxation cycle was then repeated until the loading curves are equal for consecutive cycles. The bulk stiffness or (K) could then be calculated using the gradient of any of the consecutive equal loading curves.

$$K = \frac{\Delta F'}{\Delta dl'} \tag{5.5.1}$$

where $\Delta F'$ and $\Delta dl'$ are the non -dimensional applied force and displacement.

5.5.1 Triaxial test

A triaxial machine, see figure 5.6, works by applying a known loading condition in a given direction and then measuring the displacement in three orthogonal directions. Using this data, the bulk and shear modulus, as well as Possion's ratio, can be determined. The triaxial machine allows an external pressure to be applied to the sample, which allows different failure curves to be generated. These can then be used to compute the internal friction angle of the sample. The triaxial test was used as described in section 5.5 to calibrate the normal and shear stiffness.

5.5.1.1 Experimental method

A cylindrical sample was constructed using a split mold and two steel end-plates. A membrane was then placed around the sample and sealed. The sample was then placed inside a pressure cylinder and an external pressure was applied. The results for an external pressure of 50kPa and 100kPa can be seen in figure 5.7.

5.5.1.2 Numerical method

The same test was performed numerically in *PFC*. The procedure used to generate a numerical test is described below, (see figure 5.8).

- Using the mass of the experimental test, particles were generated and allowed to free fall, under gravity, into a cylindrical container of the same dimensions as the test sample.

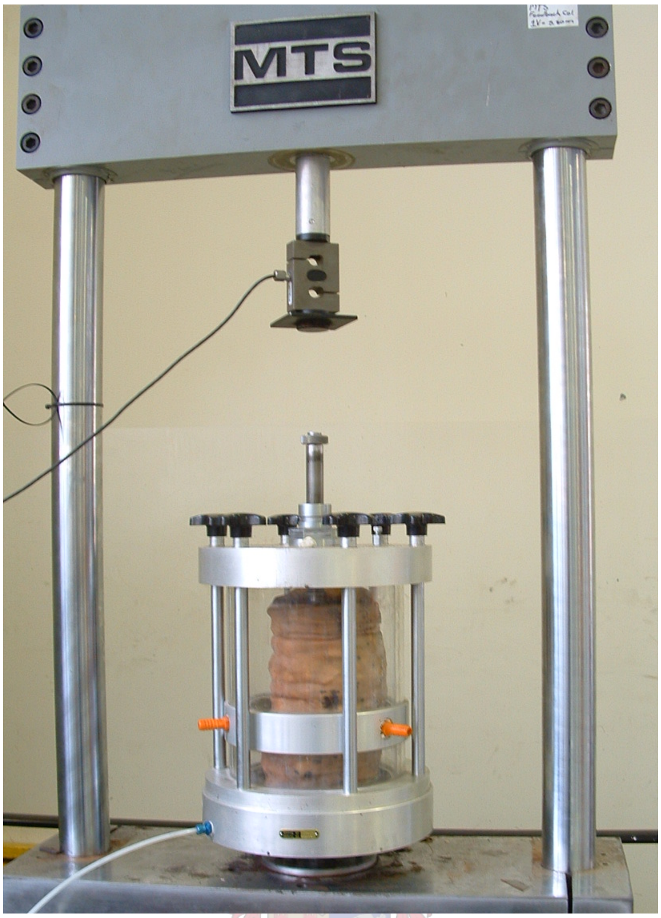


Figure 5.6: *MTS* triaxial tester

- A membrane was then generated from small particles, ten times smaller than the smallest soil particle, and bonded with contact bonds.
- The constraining cylindrical walls were deleted and the particles and membrane allowed to settle.
- An external pressure was applied to the membrane. The pressure was simulated by a normal force applied to each membrane particle.
- The assembly was allowed to reach static equilibrium.

The sample was then compressed and relaxed in an identical fashion to the experimental simulation, while the force and displacement of the cylinder was recorded.

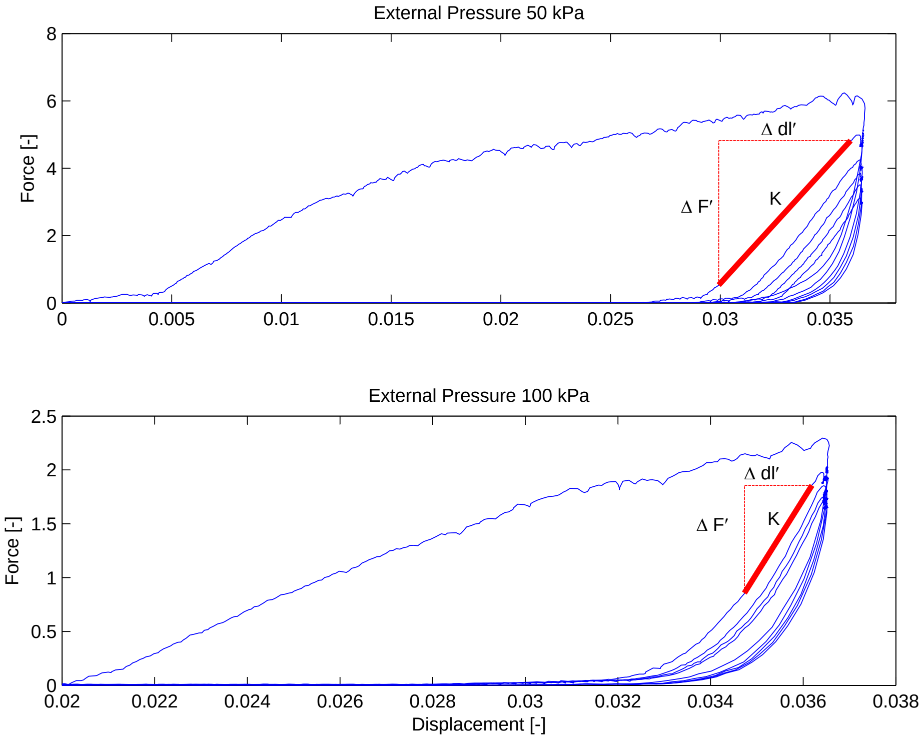


Figure 5.7: Triaxial experimental K calibration

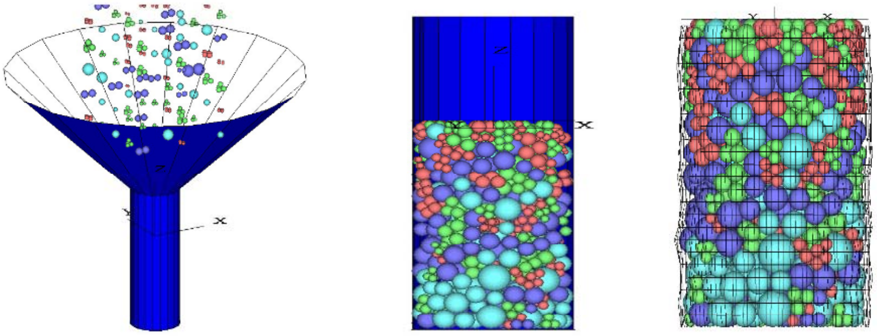


Figure 5.8: *PFC* triaxial test sample generation

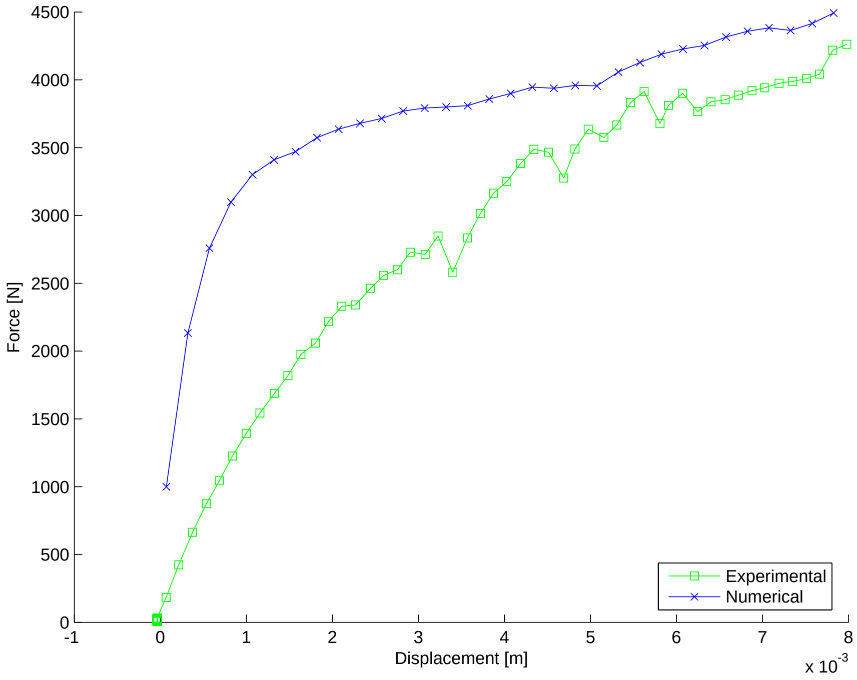


Figure 5.9: Triaxial results (Initial loading cycle) - Experimental vs. Numerical comparison

5.5.1.3 Comparison

Numerical tests with varied levels of friction were performed, but the correlation between the numerical and experimental results was poor at best. The poor correlation in results can be accredited to:

- The size of the sample being too small. The size of the test sample was limited due to restrictions on the triaxial machine and, therefore, the samples contained too few particles, typically, being six to eight particles across in diameter.

- The membrane could not be modeled easily. The forces inside the membrane contribute to the initial offset in the numerical results, see figure 5.9.

- The latex membranes could only be used for a single test and had slight variations in thickness due to the manufacture process. Although these variations were small, they proved signififcant when combined with the small sample size. The effects were evident when tests were duplicated to test for repeatability.

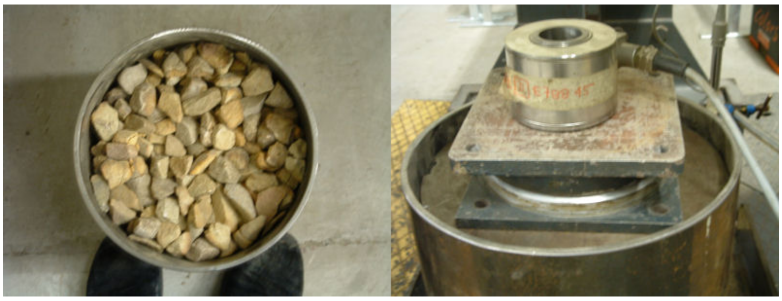


Figure 5.10: Confined compression experimental setup

The membrane properties, size and bonds where varied, but each new membrane configuration led to new and different results. It was determined that the membrane parameters introduced too many unknowns and, therefore, a new calibration method was needed.

5.5.2 Confined compression

The confined compression test works on the same principals as the triaxial test. Instead of a membrane, however, a ridged steel cylinder is used. This method doesn't produce as accurate a calibration because there is no shearing in the material, but it can be easily reduplicated in *PFC* and the results used to calibrate the samples stiffness. The size of the sample was, again, limited by the size of the machine, but a sample roughly twice the size of that used in the triaxial test could be used. The experimental setup for the confined compression test could be more arcuately controlled and reduplicated numerically than the triaxial test. The results therefore did not need to be non-dimensional.

5.5.2.1 Experimental method

The experimental setup can be seen in figure 5.10. The load cell was mounted using a portion of a ball and socket joint, allowing the load cell to only measure the vertical force, see figure 5.10. The experimental and numerical results can be seen in figure 5.11 and 5.12, respectively.

5.5.2.2 Numerical method

The *PFC* simulation was created using the same method as used to create the test sample for the triaxial test, with the exception that the membrane was never added.

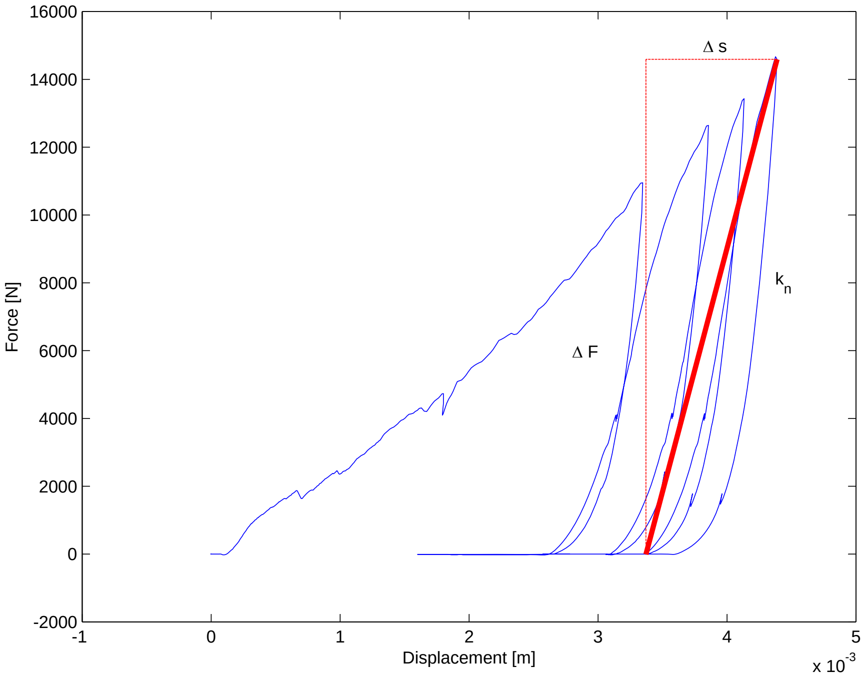


Figure 5.11: Confined compression experimental results

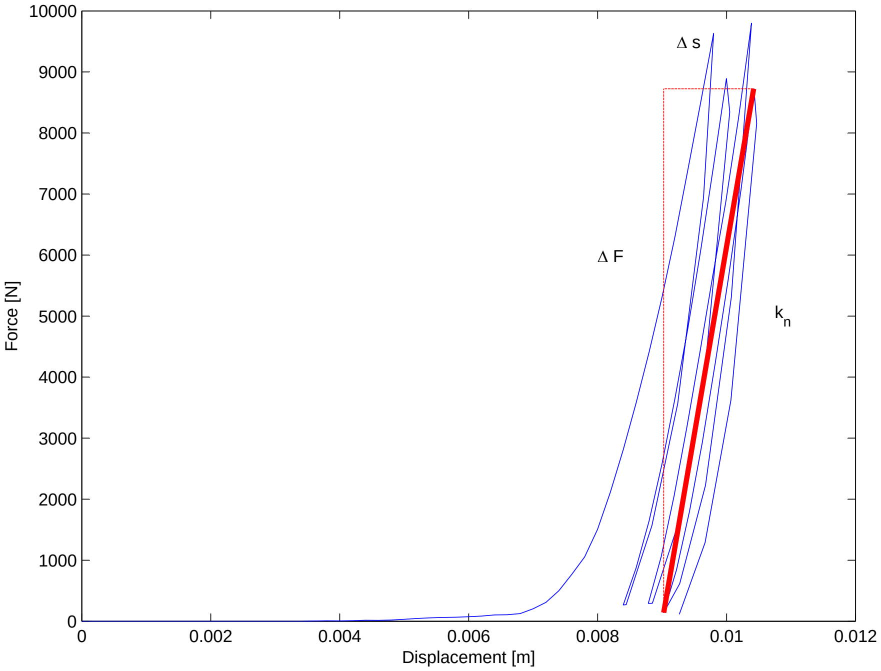


Figure 5.12: Confined compression numerical results

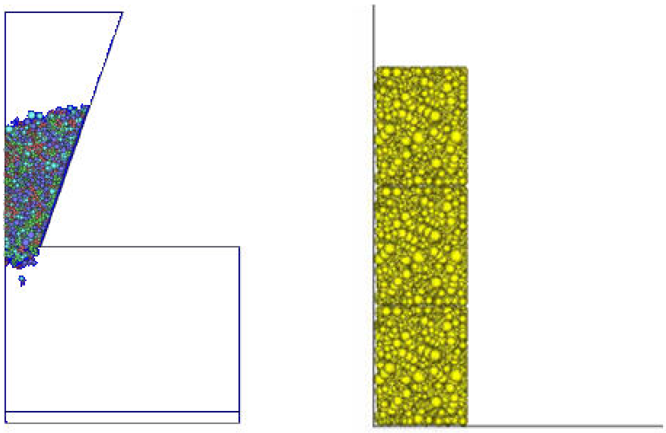


Figure 5.13: Types of angle of repose simulations

5.6 Friction

The friction coefficient or internal friction angle is comprised of two components, namely, the particle interlocking and the friction between particles. The particle interlocking is affected by the particle shapes and sizes.

The friction coefficient can be calibrated using an angle of repose simulation. This test is based on the principle that when a non-cohesive material fails a shear plane is formed and the angle of this shear plane is referred to as the internal friction angle. The material below the plane is not able to support any of the material above the plane and an angle is formed. There are two different approaches to setting up an angle of repose test, namely, the retaining wall and the hopper discharge. In theory, the angle generated by either of the two simulations is equal since it is only dependent on the internal friction angle of the material.

5.6.1 Experimental method

A sample of the material was taken and dropped from a given height through a funnel with a known diameter. This process was continued until a given mass of material had been allowed to flow though the funnel. The angle of repose was then measured. The results can be seen in figure 5.14.

5.6.2 Numerical method

The dimension of the experimental setup was then used to construct an identical numerical model. The same mass of material was allowed to flow though the funnel and the angle generated was measured. This process was continued, varying the inter-particle friction coefficient, until the same angle was obtained.

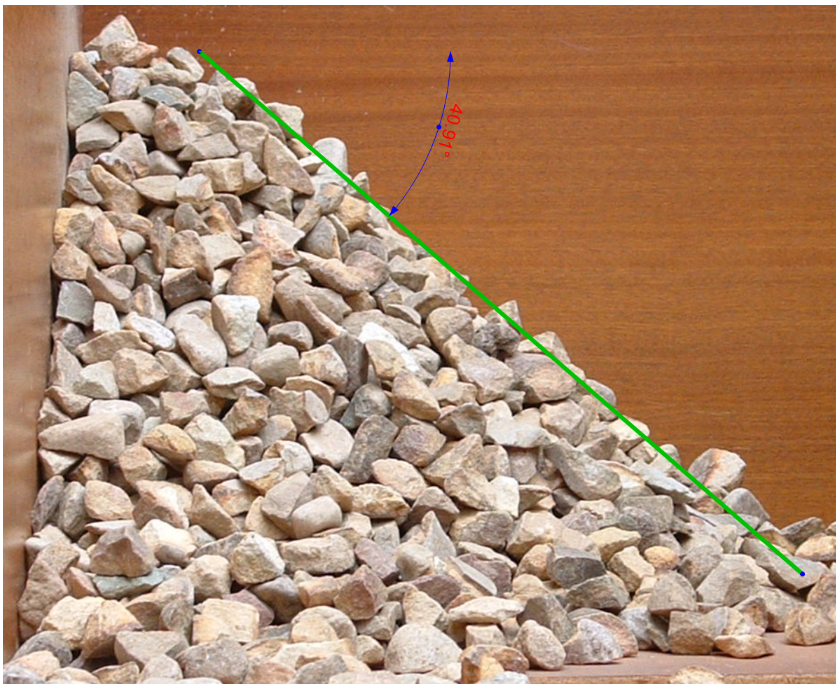


Figure 5.14: Experimental angle of repose

Parameter	Global damping	Viscous damping
Density (ρ)	1523kg/m^3	1518kg/m^3
Normal Stiffness k_n	1x10^{7}N/m	1·75x10^{7}N/m
Shear Stiffness k_n	1x10^{7}N/m	1·75x10^{7}N/m
Friction coefficient μ	0.48	0.53

Table 5.1: Material calibrated parameters

5.7 Calibrate values

Every time the contact model parameters or certain of the simulation parameters were changed, the calibration process had to be repeated. Nonetheless, the particle shapes, distribution and sizes remained constant throughout this thesis.

The calibrated material parameters given below, in table 5.7, are the base parameters of the two different types of damping models used.

Chapter 6

Experimental setup

A scale dragline model was build to obtain data which could be used to validate the numerical results. The only way to determine if the numerical model was accurate was to record the experimental drag forces and bucket trajectory during a filling cycle. The difference between the recorded and measured results would show the level of accuracy of the numerical model.

6.1 Bucket design

The bucket was modeled on the *VR-Steel's* 61m^3 bucket. The bucket had to be slightly modified during scaling. The modified bucket was designed to have properties as close as possible to that of the original bucket, see appendix A.

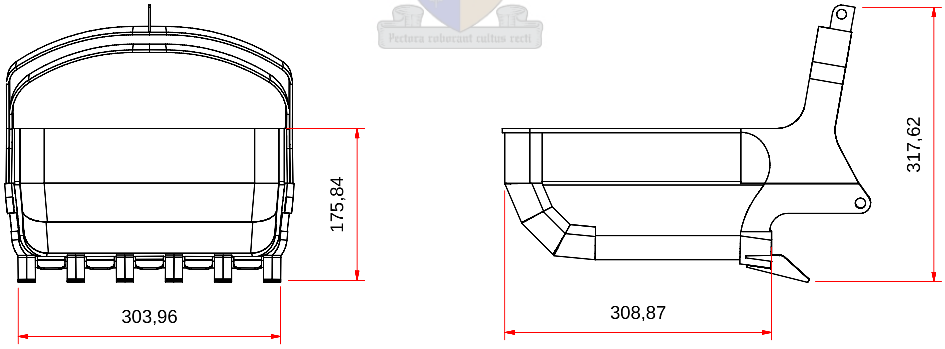


Figure 6.1: Scale bucket basic dimensions

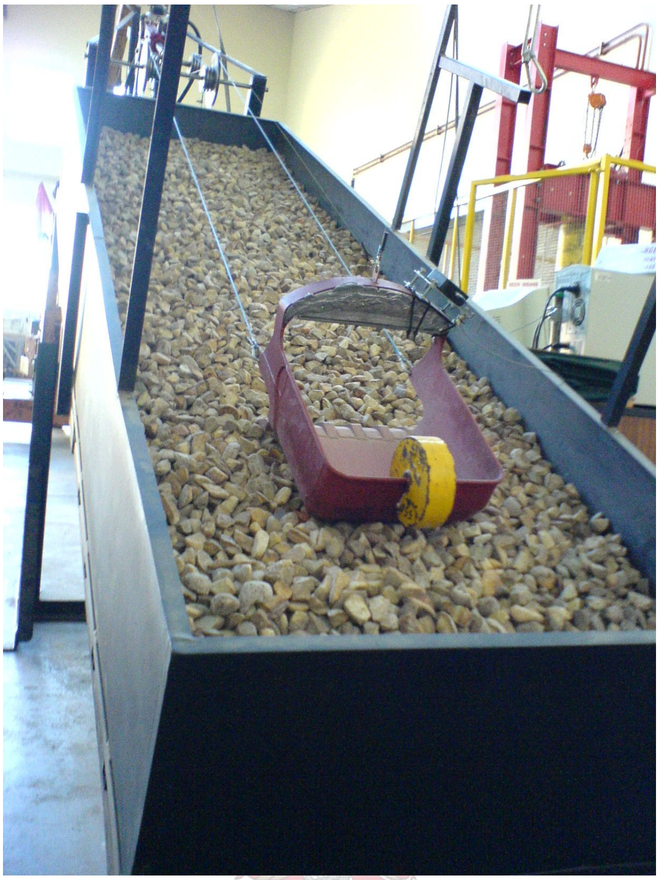


Figure 6.2: Drag bed

6.2 Drag bed

The size of the drag bed was determined using the dimensions of the scale buckets. The width and height where chosen so that there was at least one buckets width clearance, from the buckets edge to the closest wall, to prevent boundary effects. The length of the test bench made provision for the bucket to fill with over eight bucket lengths, however from experimental data the scale bucket filled in three to four bucket lengths.

The drag bed formed the framework for the entire test bench and made provision for sensor attachment. The drag bed was designed to pivot which allowed different drag angles to be tested.

6.3 Drag force and speed control

In practice draglines are speed controlled rather than force controlled. To simulate the constant speed, a hydraulic cylinder was used in conjunction with a servo valve. The servo valve was controlled by a voltage current converter.

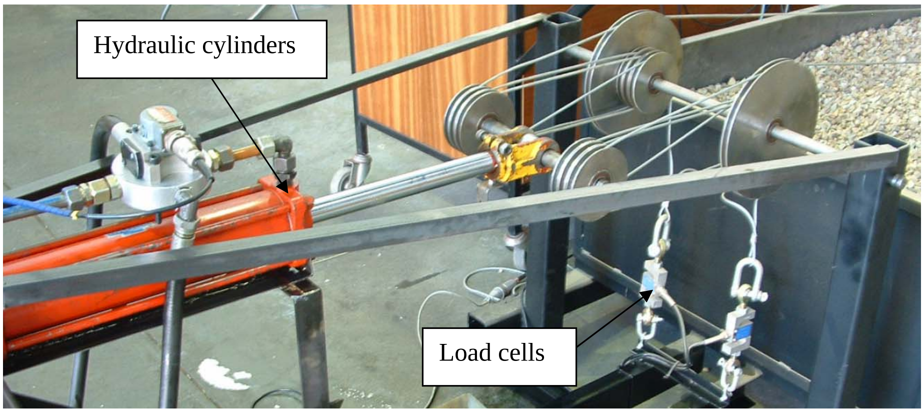


Figure 6.3: Piston and servo valve

Numerous tests were done to determine the required input voltage for an output speed. After these test it was determined that a control system would not be needed as the relationship between the input voltage and output speed could accurately be approximated with a linear curve. See appendix B.1.3.

6.4 Sensors

6.4.1 Piston Speed

The piston speed was measured by means of a linear variable differential transducer (*LVDT*).

6.4.2 Load cells

The force in each of the drag cables was measured by means of two 100kg load cells. The load cell calibration can be seen in B.1.2. The load cell position can be seen in figure 6.3.

6.4.3 Position sensors

The arch anchor was triangulated using three *ASM W12* position sensors with a measuring range of 2m. The three sensors were placed in a plane parallel to the drag bed and positioned to maximize the angle between sensors, see C.3.

6.4.4 Orientation sensor

The buckets orientation was measured using a *Micro-Strain 3DM-G*. The sensor would be used as a three axis inclimoneter. The *Micro-Strain 3DM-G* was attached to the buckets arch, see figure 6.5

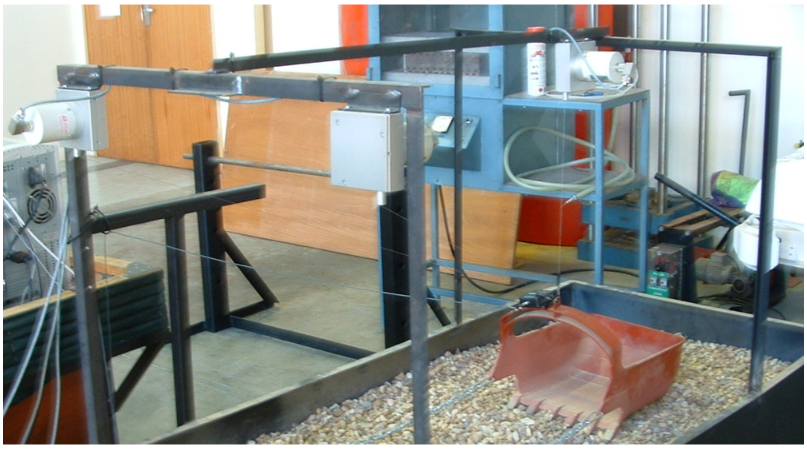


Figure 6.4: Position sensors location

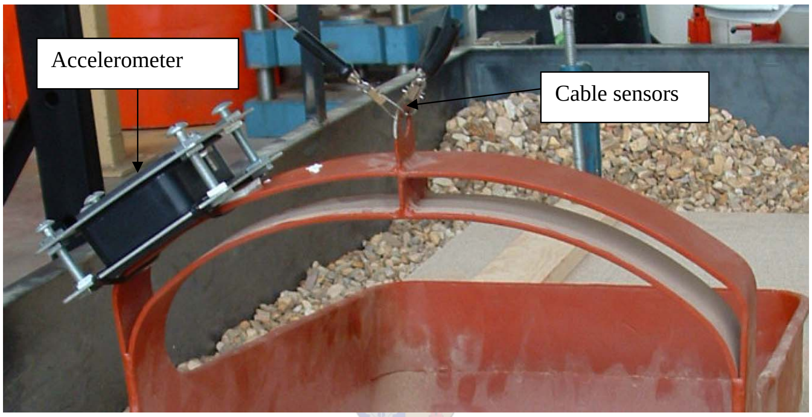


Figure 6.5: *Micro-Strain 3DM-G* location

6.5 Data collection

The sensors were attached to two different data loggers, namely the *Micro-Strain 3DM-Gs* onboard data logger and a *HBM spider 8-30*.

Micro-Strain 3DM-G	*HBM spider*
Pitch	Load cell 1 and 2
Yaw	Position sensor 1,2 and 3
Roll	*LVDT*

The data sets were recorded into the computer where the two data sets were merged. When the test begins and the buckets starts moving a visible spike can be seen in both data sets. This spike is used to synchronize the two different data sets. Both data loggers were set to sample at the same frequency, namely 25Hz.

Chapter 7

Experimental Results

Experimental simulations were performed within the normal working range of a dragline bucket. These results could then be used to verify the accuracy of the numerical model. The test bench was setup and all the input parameters fixed, allowing only the inclination angle and the drag speed to vary. Tests were performed at inclination angles of 1,44°, 15° and 25°, with drag speeds of 0·1m/s, 0·16m/s and 0·2m/s. Each test was repeated three times to ensure the results were reliable and repeatable. The bucket position, orientation and drag force were recorded for each test.

7.1 Constant drag speed and variable inclination angle

To determine the effect of inclination angle on the dragline bucket's dynamics, tests with a constant drag speed were performed at various inclination angles.

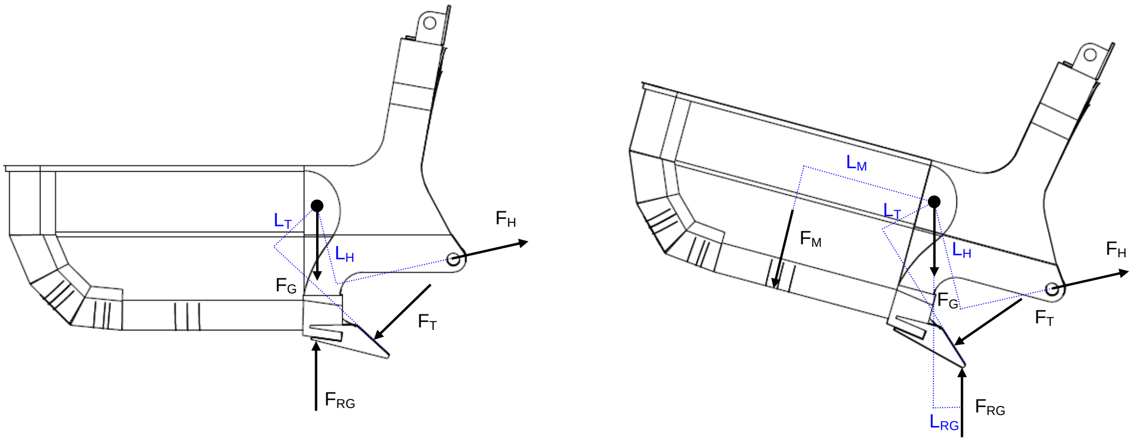


Figure 7.1: Drag and tooth moment

It was observed that the bucket fully engaged at 1,44°, partially engaged at 15° and did not engage at 25°.

Studying the recorded data and using the knowledge of when the bucket engaged, the dragline bucket's digging behavior was reconstructed. A force balance for the bucket can be seen in figure 7.1. When the bucket is initially at rest, the only forces acting on the bucket are gravity (F_G) and the reaction force (F_{RG}) caused by the bucket's weight pushing on the soil. As the drag force (F_H) is applied, the bucket is pulled forward. The applied drag force also results in a moment, forcing the bucket rear or 'basket' down and decreasing the bucket pitch angle. The forward motion forces the bucket teeth into the soil, generating an opposing reaction force of the soil on the bucket teeth (F_T). This reaction force also results in a moment, lifting the bucket basket and increasing the bucket pitch angle. If the moment generated by F_H is greater than the moment generated by F_T, the bucket will not lift and the soil will flow over the teeth into the bucket. In this instance, the bucket is said to have engaged. If the moment generated by F_H is smaller than the moment generated by F_T, the bucket will lift while pivoting around the bucket teeth.

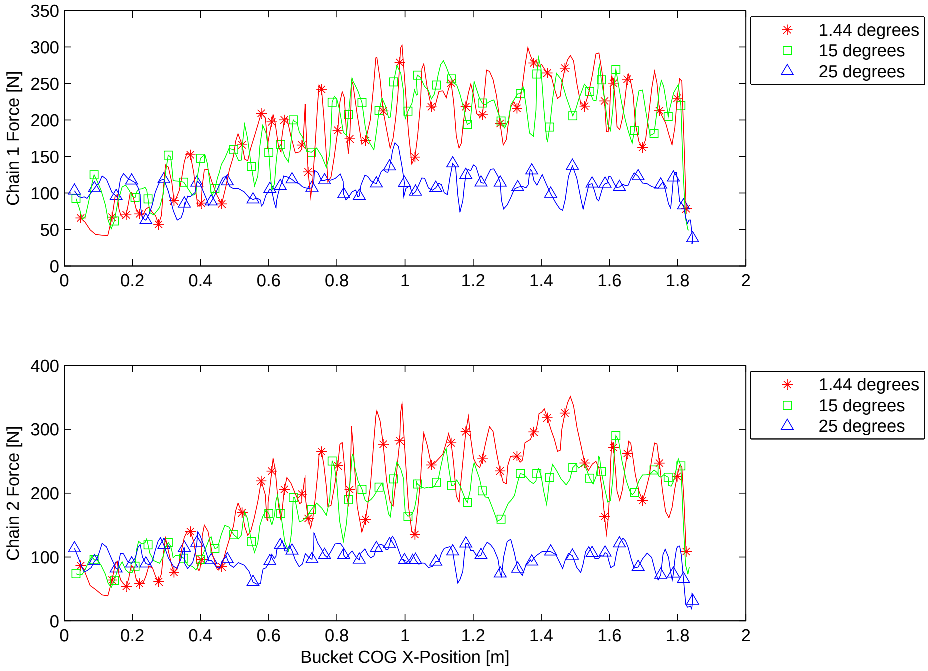


Figure 7.2: Variable inclination angle drag speed 0.2m/s - Drag forces vs COG x-component

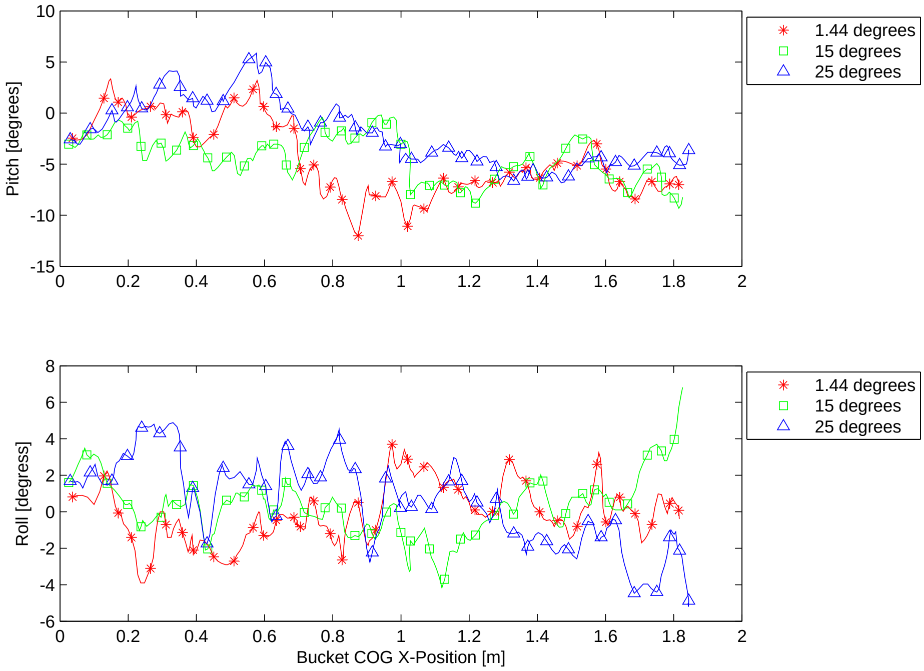


Figure 7.3: Variable inclination angle drag speed 0.2m/s - Bucket orientation vs COG x-component

When the bucket lifts, two things happen: the moment arm of the drag force (L_H) becomes greater and the moment arm of the reaction force of the soil on the teeth (L_T) becomes smaller. The reaction force of the bucket's weight on the soil (F_{RG}) moves to the only remaining point of contact between the bucket and the soil, namely, the bucket teeth. Since the F_{RG} no longer acts through the bucket's center of gravity, a new moment is generated. This moment forces the bucket basket down. As the drag speed remains constant and the bucket is no longer moving, the drag force begins to increase. The increasing drag force results in a greater reaction force of the soil on the teeth. However, the magnitude of the force the soil can generate is limited and, eventually, the soil will fail or shear. When this happens, the bucket's pitch decreases, lifting the material up and into the bucket.

The cycle continues as the bucket fills. In a partially filled bucket, the soil exerts a force (F_M) on the bucket. This force results in a moment, forcing the bucket's basket down. This added force means that the bucket does not have to pitch as high to achieve the required force to cause the soil to fail. Therefore, as the mass of soil in the bucket increases, the bucket pitches less.

The magnitude of the force pushing the teeth into the soil increases with each cycle, as a result of the increasing mass of material in the bucket. The

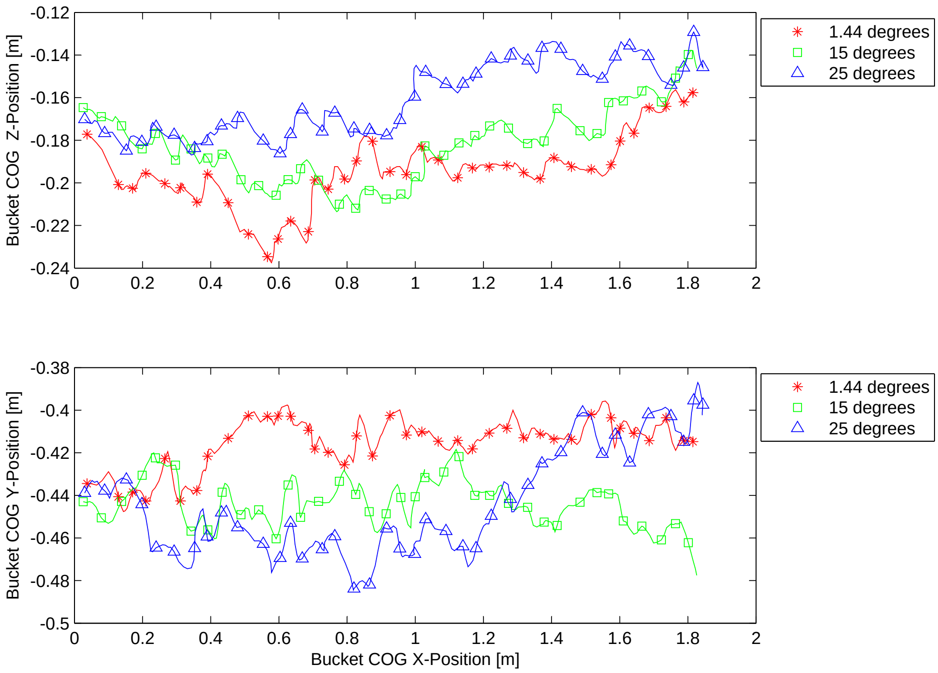


Figure 7.4: Variable inclination angle drag speed 0.2m/s - Bucket COG z-component vs COG x-component

increased force causes the teeth to penetrate further into the soil. The increased penetration depth of the teeth results in more material needing to fail, thus, resulting in a larger shear of failure zone and increasing the drag force required. If a bucket displays the above behavior it is said to have partially engaged.

If the bucket teeth do not penetrate deep enough or the angle of the bucket teeth in insufficient, the teeth will slide over the surface of the material, resulting in little or no flow of material into the bucket. In this case the bucket is said to have not engaged.

Figure 7.2, shows the drag force recorded during the simulations. Both the 1,44° and 15° inclination angle simulations show an increase in the force in each chain as the simulation progresses, implying full or partial engagement. It is, however, very difficult to determine whether the bucket is engaged or partial engaged. The simulation performed at 25° inclination shows little or no increase in drag force, which coincides with the observed data that the bucket did not engage.

Figure 7.3 shows the recorded bucket pitch and roll. After careful analysis, there appears to be no identifiable pattern to the bucket roll . Consequently, it is believed to be a function of the material bed, such as rock orientation, and considered to be noise. Figure 7.4 shows the recorded bucket center of gravity, see appendix C for a sample calculation.

By combing the data from figures 7.2, 7.3 and 7.4 and the dragline bucket dynamics described above, the observed results can be validated.

At the beginning, the 1,44° inclination angle simulation shows a drop in the drag force as the bucket pitch angle increases. As the simulation progresses, the bucket pitch remains high, but the center of gravity decreases and the drag force increases. This implies that the bucket is digging into the material without actively lifting and falling. Slight fluctuations can be seen in the drag force and bucket pitch, but the bucket pitch remains high and center of gravity continually decreases. After about 0·8s, the bucket pitch decreases and the bucket center of gravity begins to plateau and, eventually, increase. This turning point represents the point at which there is not enough force pushing the bucket teeth into the soil for the bucket to continue digging. The point where the drag chains are pulled to remains constant. Therefore, as the simulation progresses, the angle at which the drag force is applied increases. The component pulling the bucket up and out of the soil, consequently, also increases. The ever increasing mass in the bucket also leads to an increasing moment, pushing the basket down and reducing the force pushing the teeth down into the soil. As a result, the bucket center of gravity increases and the bucket moves up and out of the material. The drag force also begins to stabilise.

The simulation performed at 15° also shows an increase in the drag force over time. However, the bucket pitch shows large fluctuations in the begging around the starting point. As the bucket begins to pitch there is insufficient force pushing the bucket teeth into the soil to keep the bucket digging. The bucket pitch then decreases, lifting a small amount of material into the bucket. This process continues until around 0·7s, when there is sufficient soil in the bucket for the bucket to start digging. From the simulation performed at 1,44°, the vertical component of the drag force started lifting the bucket out of the soil around 0·8s. In the 15° test, the reduced weight of soil in the bucket and, therefore, reduced moment forcing the bucket basket down, the bucket only starts rising around 1s.

The recorded drag force of the simulation performed at 25° already showed that the bucket did not engage. The bucket pitch and center of gravity confirm this. The bucket pitches, but the pitch remains high and the center of gravity increases rather than decreases. This implies that the bucket is being pulled over the material and does not have sufficient force to penetrate the material. The increasing center of gravity can be contributed to the bucket moving up the material it pushes in front of it as it is dragged along the surface.

Integrating the area under figure 7.2, combined with the cylinder position, the total work done by the cylinder pulling the bucket through the soil can be calculated. Figure 7.5 shows clearly that the energy required to pull the bucket for both the 1,44° and 25° inclination angle simulations is considerably higher than that of the 25° inclination angle simulation. It is important to remember

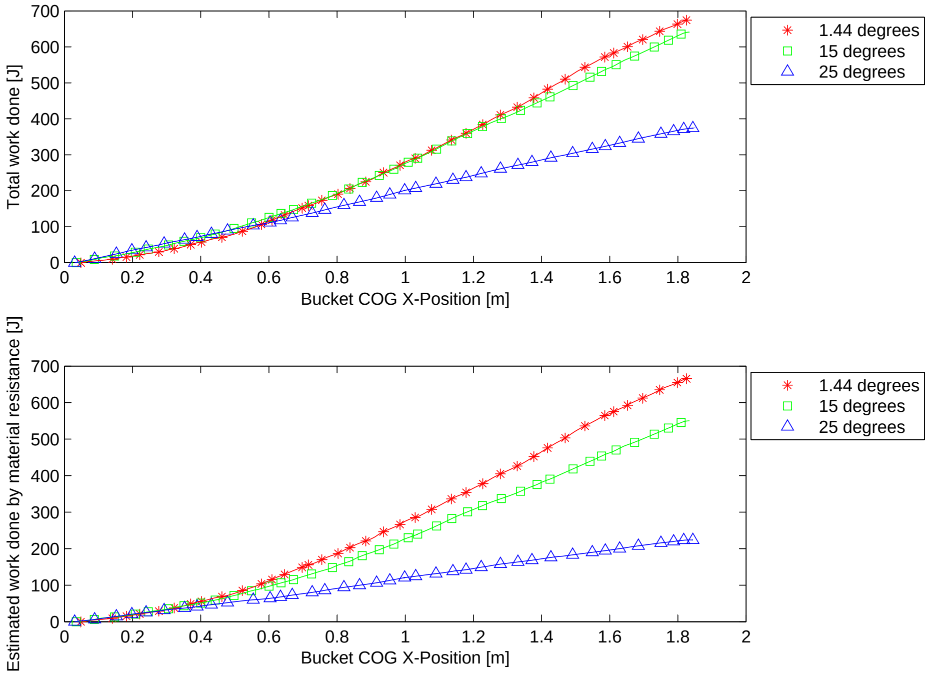


Figure 7.5: Variable inclination angle drag speed 0.2m/s - Work done vs COG x-component

that the work done is calculated using the force required by the cylinder to pull the bucket. As mentioned previously, as the inclination angle increases, the component pushing the bucket into the soil decreases and, therefore, the component working parallel to the material surface and in the opposite direction to the drag force increases. The magnitude of this force can be calculated using the inclination angle and initial weight of the bucket. This is a rather crude approximation, but it is sufficient to illustrate the differences between the simulations performed at 1,44° and 15° inclination angles. The results can be seen in the second figure of 7.5.

7.2 Variable drag speed and constant inclination angle

The second set of tests where performed to determine the effects of the drag speed. The inclination angle was fixed at 1,44° and drag speeds of 0·1m/s, 0·16m/s and 0·2m/s where tested. The bucket was observed to engage in all the simulations.

Figure 7.6 shows the recorded drag force. The simulation performed at

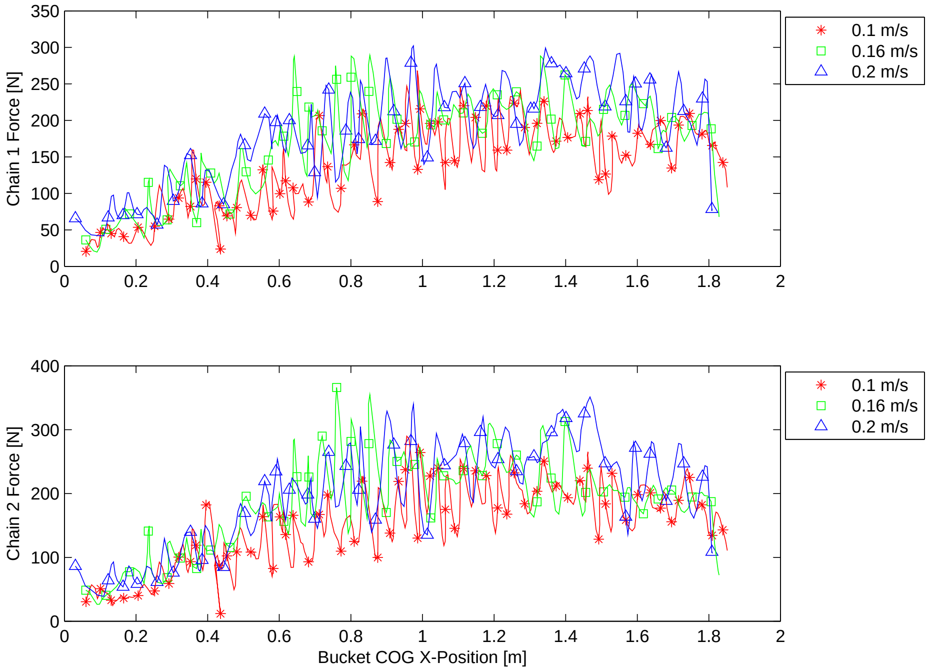


Figure 7.6: 1.44 degree inclination angle variable drag speed - Drag forces vs COG x-component

0·1m/s shows the smallest fluctuations in the magnitude of the drag force. Figure 7.7 shows that at a drag speed of 0·1m/s the bucket pitch angle was the highest and remain higher than achieved by the simulations performed at 0·16m/s and 0·2m/s. The fluctuations in the bucket pitch for the simulation performed at 0.1 metres per second were also the smallest. By combing all of the above and the total work done, figure 7.9, it can be concluded that the bucket jerked less as it was pulled though the soil and, as a result, dug deeper.

The simulation performed with a drag speed of 0·2m/s shows a more jerky drag force and bucket pitch angle, ultimately, resulting in a higher drag force and, thus, higher total work done.

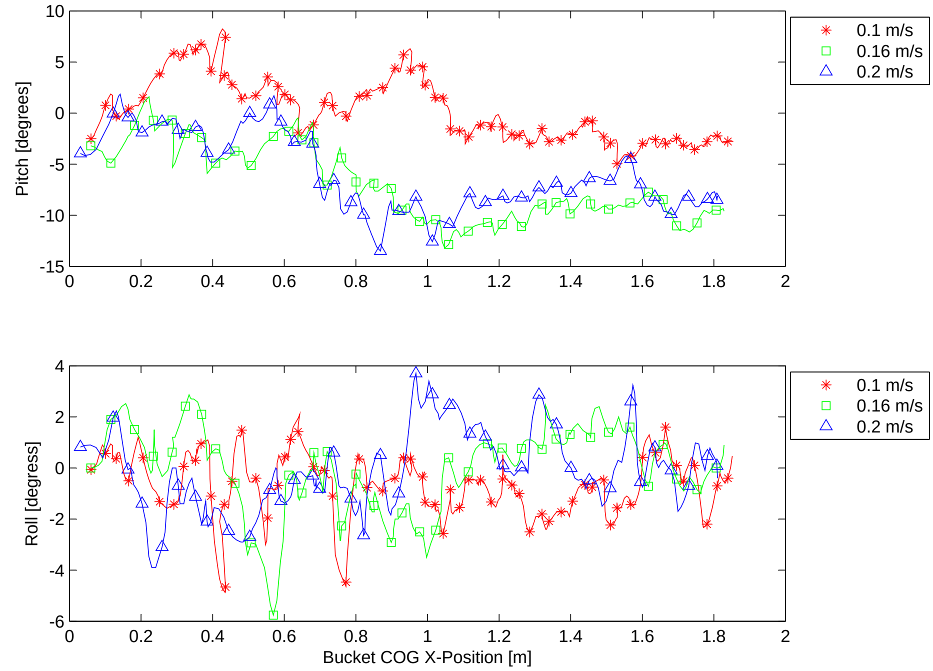


Figure 7.7: 1.44 degree inclination angle variable drag speed - Bucket orientation vs COG x-component

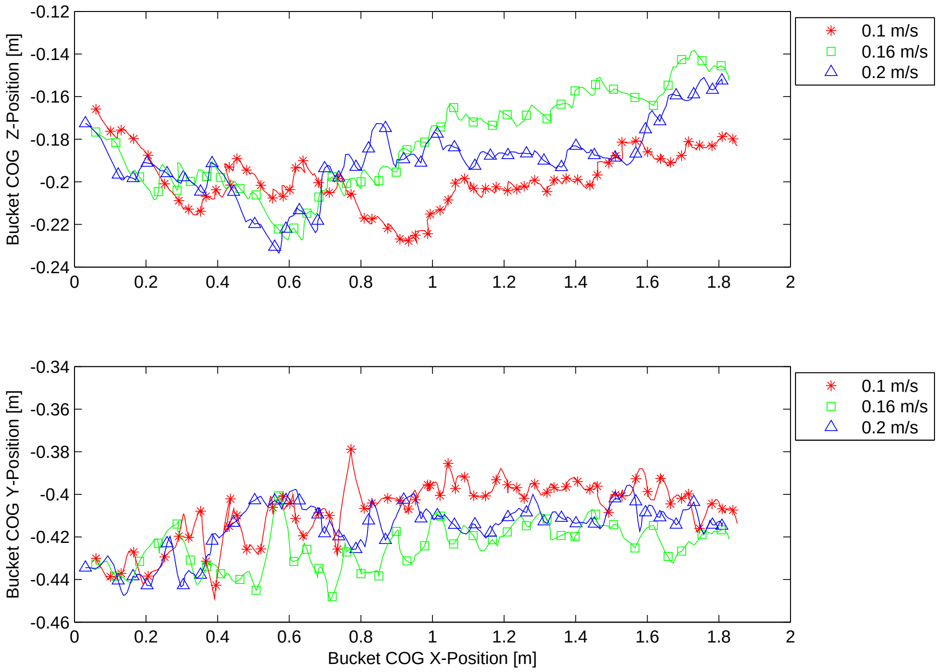


Figure 7.8: 1.44 degree inclination angle variable drag speed - Bucket COG z-component vs COG x-component

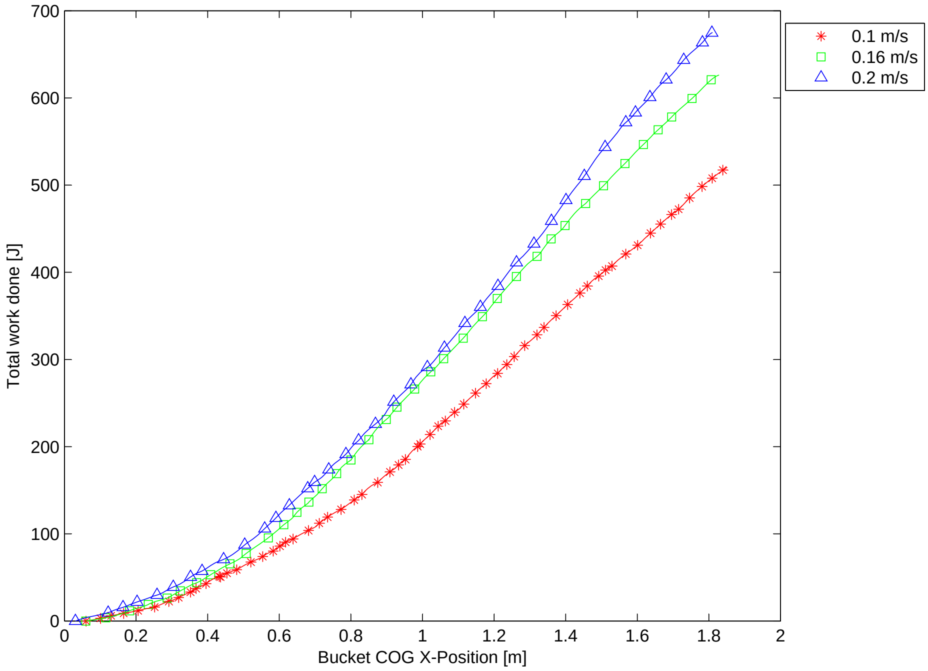


Figure 7.9: 1.44 degree inclination angle variable drag speed - Work done vs COG x-component

Chapter 8

Numerical Results

8.1 Background

Using the data acquired from the experimental simulations, as well as the trends identified, the numerical model could be verified. The amount of time and computer facilities available greatly reduced the number of simulations that could be run. The calibration process provided all the information needed to simulate the soil used in the experimental simulations. The only parameter that still needed to be calibrated was the material damping. Unlike the other parameters, there is no test that can be used to directly calibrate the damping and, therefore, different damping schemes were tested and will be discussed later in this chapter. Every time the damping scheme was changed, the remaining material properties also needed to be recalibrated, limiting the number of damping schemes that could be tested.

The two damping models built into *PFC*, namely, global and viscous damping were tested. Two different global damping values were tested. Due to the nature of global damping, see section 2.4.1, it was decided to test using a very small damping coefficient (0.05), thereby minimising the negative effects of global damping while still providing a basic damping model. The second global damping test used a higher damping coefficient (0.15) to provide a more active form of damping. The second damping model that was tested was viscous damping. The nature of viscous damping, see section 2.4.2, allowed a higher damping coefficient (0.8) to be used without adversely effecting the system dynamics.

8.2 Inclination angle 1,44°

Simulations were performed using each of the damping schemes and the results can be seen below. Figure 8.1 shows that, for the first 0·4s, the drag force for all the damping schemes appears to track the experimental data. After 0·4s, however, the drag force of all the numerical simulations begins to exceed the

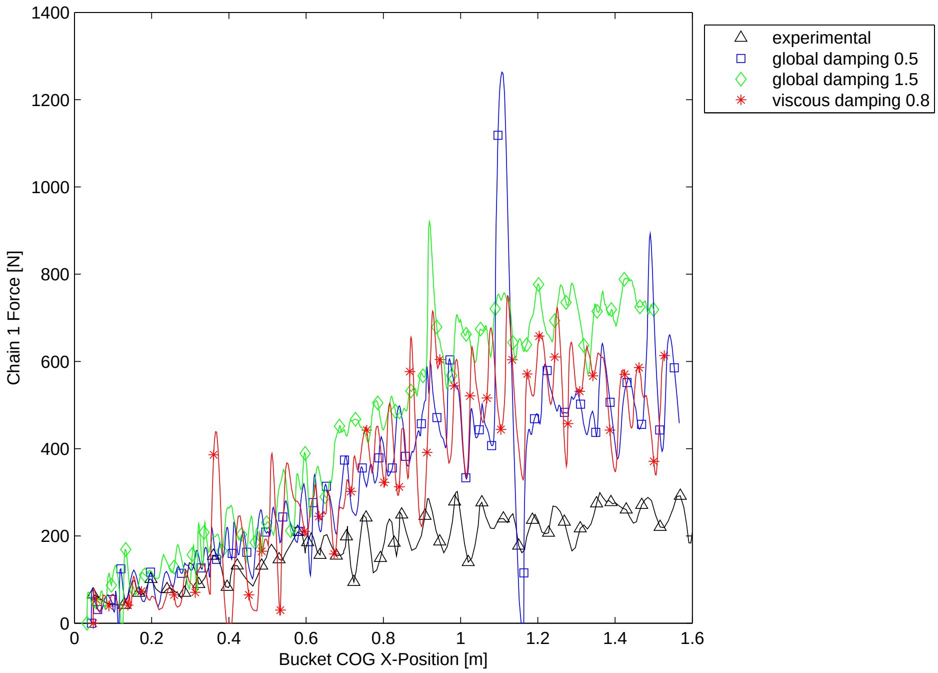


Figure 8.1: Inclination angle 1.44 degrees - Simulated force vs COG x-component

experimental recorded drag force. The reason for the increased drag force can be ascertained by analysing figures 8.2 and 8.3. In all the numerical simulations, the bucket pitches higher than in the experimental simulations. From the dragline bucket digging mechanics described in section 7.1 the bucket is expected to pitch. As the bucket pitches, the entire mass of the bucket is supported by the bucket teeth and, as a result, the teeth are forced into the soil. In the numerical model the bucket pitches as is expected and the teeth are forced into the soil as expected. At this point in the experimental simulation there is not enough force pushing the bucket teeth into the soil and the soil above the teeth is forced to shear, causing the bucket to fall and lifting the sheared material into the bucket. In the numerical simulations, however, the bucket keeps on digging, see figure 8.3. The increased penetration depth of the teeth requires a larger force to shear the material above the teeth and, as a result, it requires less energy for the bucket to continue digging. Possible causes for this behavior could be: insufficient inter-particle friction; the numerical chain model, see section 3.2.1 or incorrect and inaccurate contact and damping models.

The inter-particle friction determines how difficult it is for the particles to move over one another. If the inter-particle friction is to low, it would result in the particles moving too easily over one another and would allow the bucket

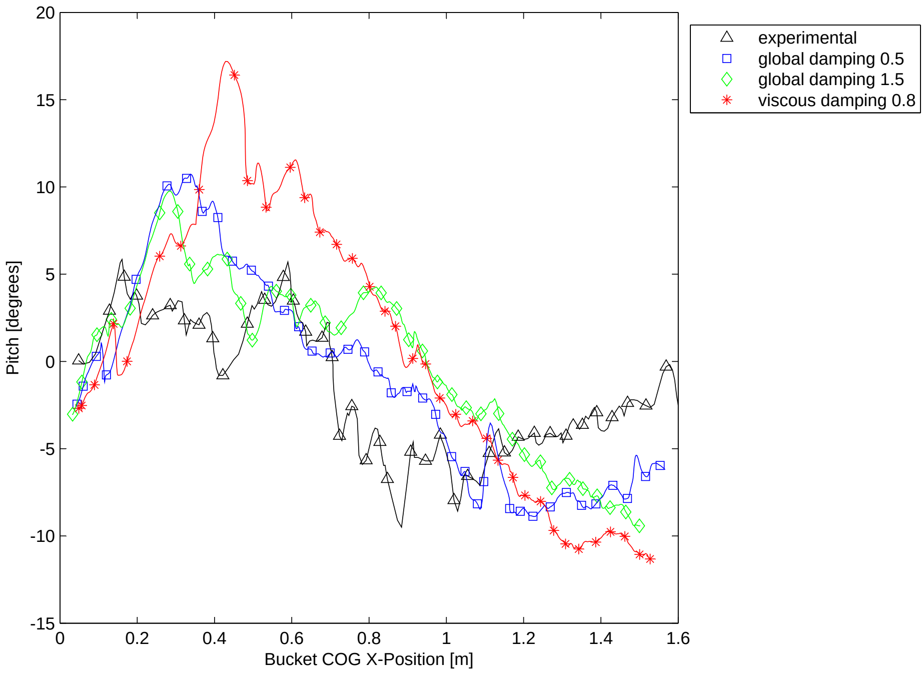


Figure 8.2: Inclination angle 1.44 degrees - Simulated bucket pitch vs COG x-component

teeth to penetrate deeper into the soil. The two factors directly responsible for the inter-particle friction are the particle friction coefficient and particle interlocking, which is a function of the particle shape.

The numerical chain was simulated using a virtual spring. The numerical chain, therefore, had no damping and always acted on a straight line between the bucket hitch points and the origin of the drag force. This inflexibility leading to the spikes observed in the chain force.

The final factors affecting the behavior of the soil are the contact and damping models used. *PFC* provided only two contact and damping models. The contact models were the *Hertz*, section 2.3.2.1, and linear contact models, section 2.3.1. The *Hertz* contact model did not allow particles to be clumped and, therefore, could not be used for simulations, leaving only the linear contact model. As discussed above, both of the damping models were used.

An additional shortcoming of *PFC* is that it was initially written to solve static problems and the linear contact and global damping models were not designed for dynamic systems. The procedures and methods used to calibrate the material were also designed using static methods, but employed because no practical dynamic methods could be found. The only parameter that could, therefore, be adjusted was the damping model used.

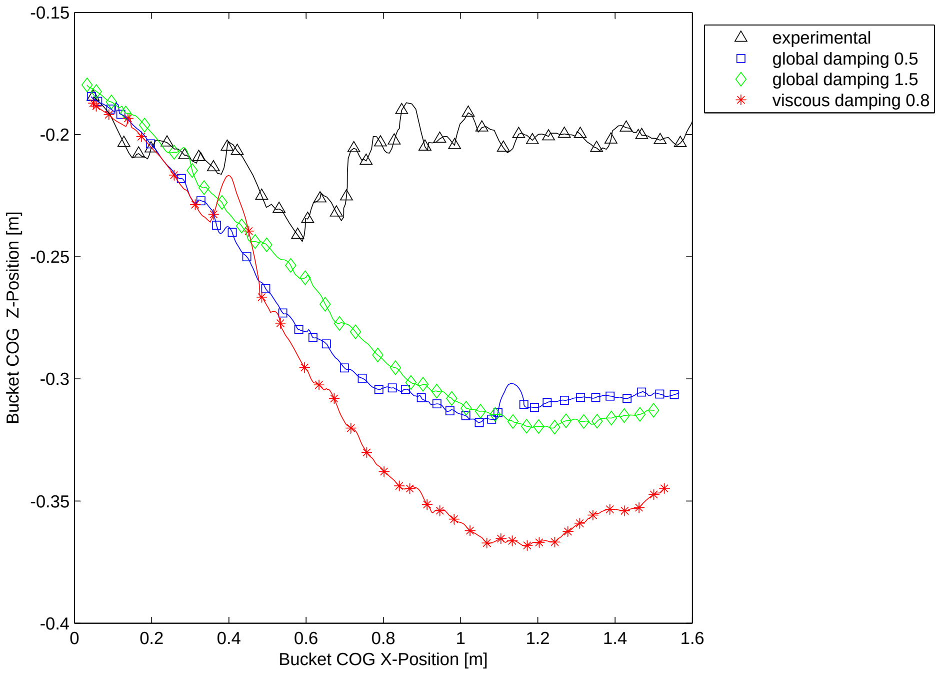


Figure 8.3: Inclination angle 1.44 degrees - Simulated bucket COG vs COG x-component

8.3 Damping models

Three different damping models were used, as discussed in 8.1. The first was global damping, with a very low damping coefficient of 0.05. Global damping damps all the resultant force in the system, reducing the particle and bucket movement while increasing the forces required to pull the bucket through the soil at a constant speed. Comparing the simulations conducted with global damping coefficients of 0.05 and 0.15, the following can be observed.

Figure 8.4 depicts the simulation conducted with a global damping of 0.15. It required considerably more energy to pull the bucket through the soil. Analysing the trajectory of the bucket's center of gravity in figure 8.3, it can be observed that the simulation conducted with a global damping coefficient of 0.15 reacted slower with less fluctuations. Since the fluctuations form a vital part of the bucket digging dynamics, even though the bucket trajectory and pitch are better for the simulation performed with a higher global damping, global damping cannot be used. Global damping makes the movement of all the entities in the simulation slower, which works well for static simulations that tend towards equilibrium, but is not desirable in dynamic simulations.

The third simulation was conducted using a viscous damping model. Figure

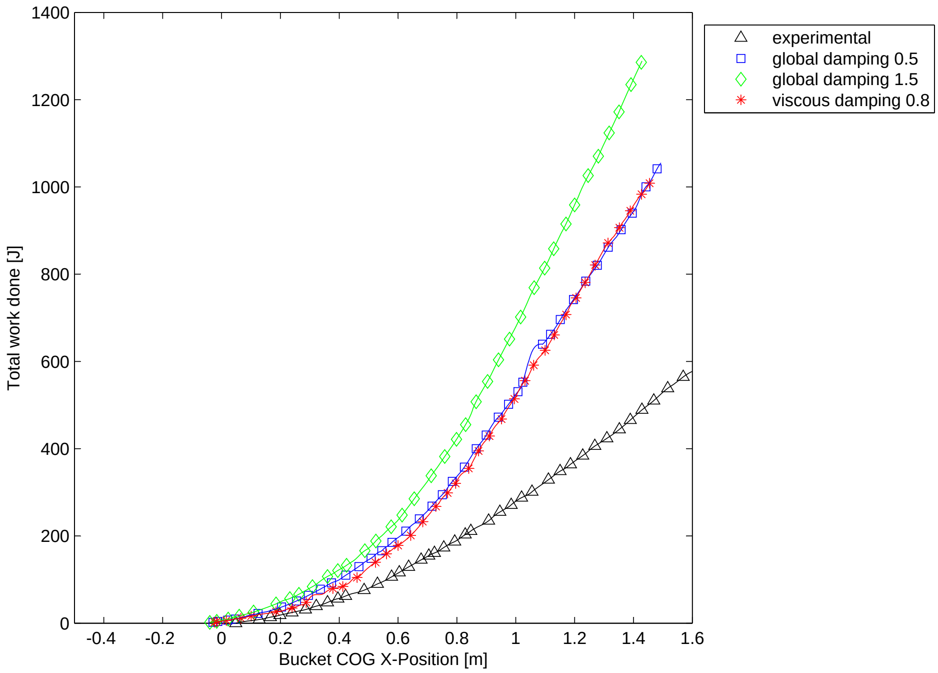


Figure 8.4: Inclination angle 1.44 degrees - Work done vs COG x-component

8.4 shows that the energy used to pull the bucket through the soil is considerably more than required by the simulation conducted with a global damping coefficient of 0.15. Even with a very high viscous damping coefficient of 0.8, the energy used was very close to the simulation performed with a global damping coefficient of 0.05. This demonstrates that the viscous damping damps the system without unnecessarily increasing the forces in the simulation. However, the problem with the simulation conducted using viscous damping is that the bucket dug deeper into the soil, see figure 8.3.

8.4 Viscous damping - variable inclination angle

Even though the viscous damping model did not perform as well as the global damping models, the model was theoretically better and it was decided to run simulations at inclination angles of 1,44°, 15° and 25°. These simulations were then used to determine whether the same trends identified in the experimental simulations could be captured by the numerical model.

Figure 8.5 shows that, as the inclination angle is increased, the bucket's trajectory becomes shallower. The same trend could be observed in the ex-

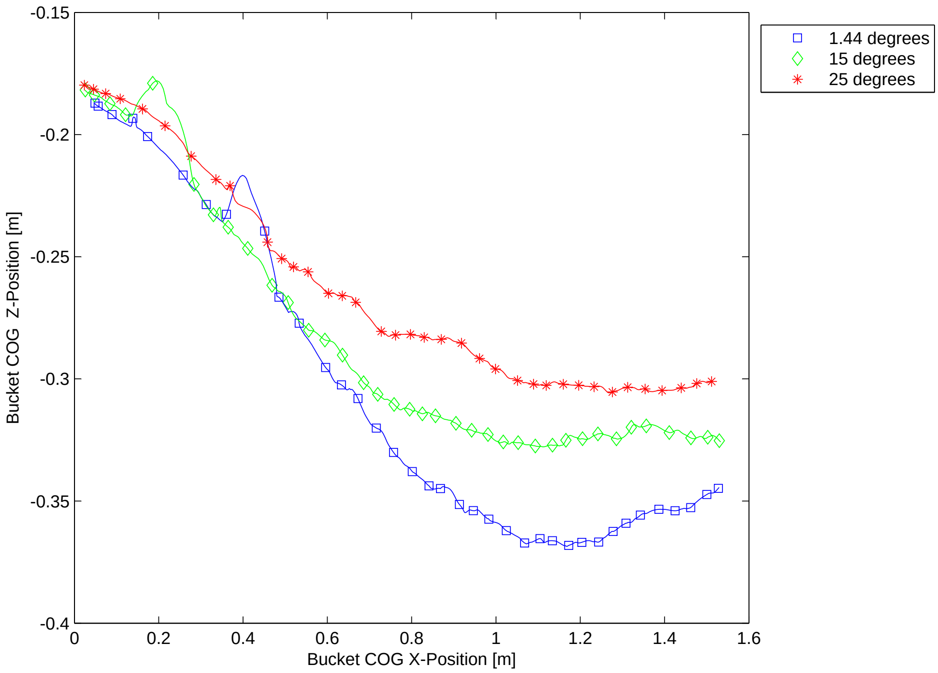


Figure 8.5: Variable inclination angle (Viscous damping) - Simulated bucket COG vs COG x-component

perimental simulations, with the exception that the bucket did not engage at an inclination angle of 25°, see section 7.1. In the numerical simulations, the bucket engaged at all the inclination models, showing, once again, that it dug too easily.

One of the major advantages of the numerical model is the ability to calculate the mass of soil in the bucket. From figure 8.7 it can clearly be seen that, as the inclination angle increased, the mass of the soil in the bucket decreased. Figure 8.6 shows the work done to pull the bucket though the soil. As the inclination angle was increased, the work done by the system decreased, matching the results of the experimental simulations.

8.5 Conclusion

None of the numerical models tested could accurately reproduce the results measured in the experimental simulations. The methods used to calibrate the soil and the assumptions made about the soil parameters were widely accepted and, therefore, assumed to be correct.

In optimising the simulation it was assumed that the bucket trajectory could be approximated by a two dimensional representation and, therefore, a

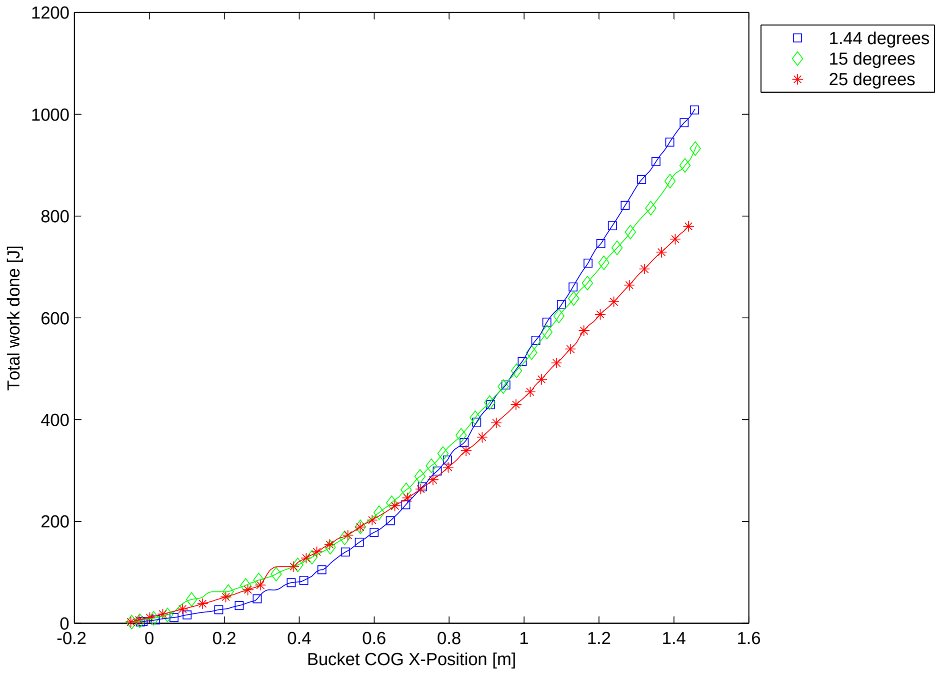


Figure 8.6: Variable inclination angle (Viscous damping) - Work done vs COG x-component

symmetry plane was used. The symmetry plane greatly reduced the number of particles and, thus, the time required to run a simulation. The symmetry plane also removed one translation and two rotational degrees from the system, further reducing the time needed to run a simulation.

The experimental results showed that, even though the assumptions made about the bucket's trajectory were valid and the out of plane rotations and translation of the bucket appeared to be noise, they did have any effect on the simulation. It is possible that the extra translations and rotation made it easier for the bucket to shear the material above the bucket teeth and, therefore, made it more difficult for the bucket to dig. In the numerical simulations these degrees of freedom were removed and, thus, the bucket trajectory was forced, which could have lead to the increased drag force and the bucket digging deeper than expected.

The final factors that could have contributed to a less accurate numerical model were the contact and or damping models. As mentioned previously, the linear contact model works well with global damping for static simulations, but is not suited for dynamic simulations. When a viscous damping model was employed, the numerical model was able to accurately predict the trends observed in the experimental simulations. It should be noted, nonetheless,

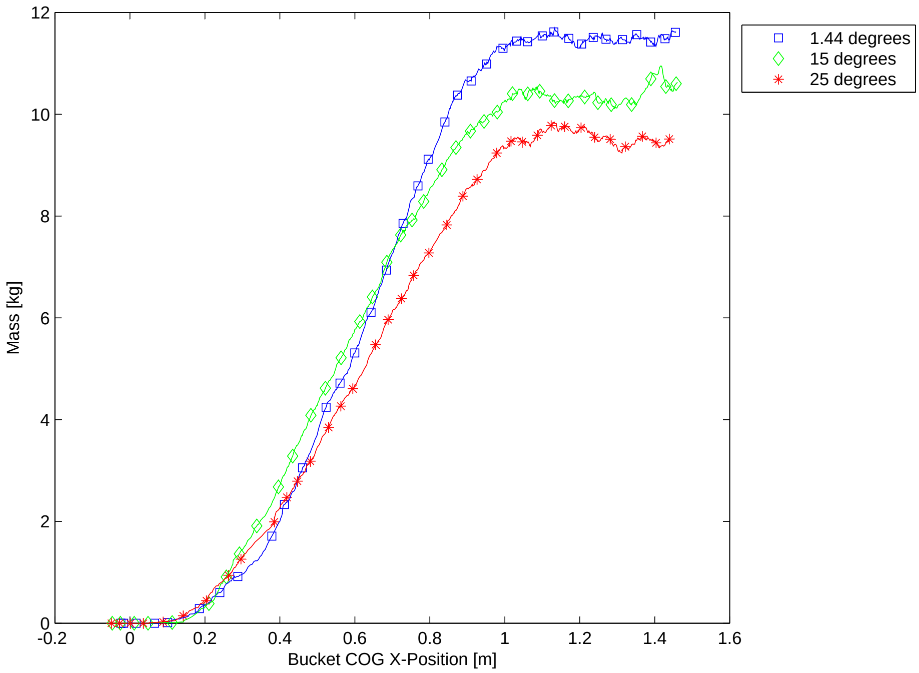


Figure 8.7: Variable inclination angle (Viscous damping) - Mass vs COG x-component

that there are other non linear contact models available in the *PFC 3.1*, one of which is based on the *Walton-Braun* contact model, see section 2.3.2.2. A demo version of *PFC 3.1* was lent to the university and the contact model showed real potential, but is beyond the scope of this thesis.

Chapter 9

Conclusion

In order to increase the productivity of open cast mining, mining companies are forced to continually push the limits. Most draglines in operation today work well above the recommended working loads to reach production targets and this can result in significant damage to the machines. Extensive work has been done on improving and optimising the existing draglines to handle the increased loading, however, little is known about the exact nature and magnitudes of the loads cases that occur. Currently, all the data availably is determined by expensive experimental measurements.

Dragline bucket designers are required to prove their bucket designs by building and testing scale models. In recent years doubt has been cast on the validity of the results obtained by scale testing. The need to simulate the dynamic response of a granular material has, therefore, become a necessity in the modern dragline industry.

Accurate numerical models would allow the design of a dragline bucket to be optimised to specific digging conditions. The numerical models could also be used to determine the exact nature of the forces experienced by the bucket on the dragline, resulting in increased productivity and reduced downtime.

This thesis used an existing *DEM* software code to create and simulate the filling of a dragline bucket in a variety of different scenarios. The results obtained were then compared against experimentally measured results. It was shown that the numerical model could not accurately reproduce the experimental simulations. Possible reasons for the less accurate results of the numerical model are the assumptions made in the creation of the numerical model, as well as the contact and damping models used.

Due to the limitations of current computer technology and facilities, many assumptions had to be made to reduce the number of particles used in a simulation. Assumptions such as the symmetry plane and virtual chain model could have caused or contributed to the inaccuracy of the numerical model. The contact and damping models used by *PFC* were initially designed to solve static systems. Research conducted in recent years is leading to the creation of non linear models designed for dynamic simulations, but, unfortunately, these models are not part of the standard *PFC* code and could, therefore, not be used. The numerical model did, nonetheless, predict many of the trends seen in the experimental simulations.

The advantages of the numerical model, such as mass flow rate, energy usage and shear zone mechanics, could be used to revolutionise the dragline industry by providing the design engineers with invaluable and previous unobtainable data. It is, therefore, believed that further study is justified.

Appendices

Appendix A

Scale bucket

The scale bucket was based on *VR-Steels* $61m^3$ bucket. The model was scaled until the distances between the hitch points where approximately $0.31m$. The scale factor was therefore

$$s_f = l_H/l_l = 1/18 \tag{A.0.1}$$

where the l_H and l_h is the distance between the hitch points in the real and model buckets.

The real bucket has a mass of (m_B) is $44850kg$ and an effective volume (V_B) of $59.853m^3$. Using these, the BER ratio can be calculated.

$$BUR = m_B/V_B \tag{A.0.2}$$

The most important property of the model bucket, other than its shape, is its mass. Using the BUR, bucket unit ratio, and the scale factor, the mass of the scale bucket (m_b) can be determined.

$$m_b = BUR \times s_f^3 \times V_B = 7.69kg \tag{A.0.3}$$

It is important to note that we are scaling a volume and since our scaling factor was determined using a length, the scaling factor needs to be cubed before being used to scale a volume.

To ensure similar filling behavior, the bucket's moment of inertia should be similar to that of the real bucket.

The real bucket's inertial matrix is

$$\underline{\overline{I_B}} = \begin{bmatrix} 2.983 \times 10^5 & 496.195 & 2.622 \times 10^3 \\ 496.195 & 4.322 \times 10^5 & -4.095 \times 10^4 \\ 2.622 \times 10^3 & -4.095 \times 10^4 & 3.077 \times 10^5 \end{bmatrix} kg \times m^2 \tag{A.0.4}$$

Using the scale factor and $\underline{\overline{I_B}}$ the inertial matrix of the scale bucket should be

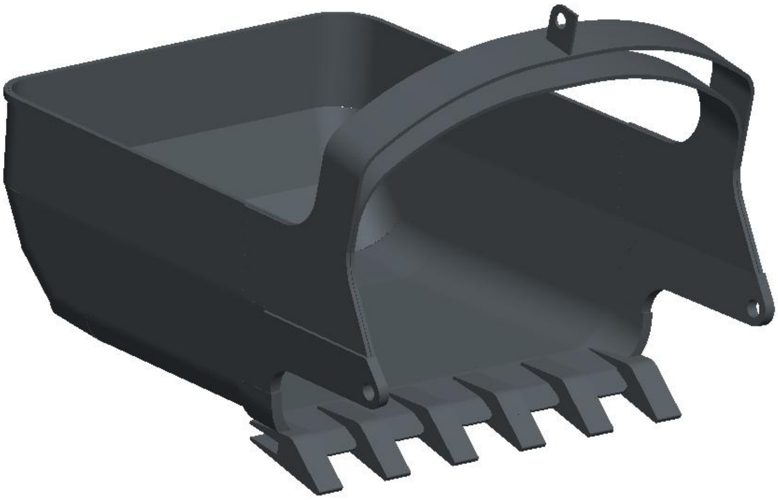


Figure A.1: Scale bucket

$$\underline{\overline{I_b}} = s_f^2 \times \underline{\overline{I_B}} = \begin{bmatrix} 0.1585 & 2.626 \times 10^{-4} & 1.388 \times 10^{-3} \\ 2.626 \times 10^{-4} & 0.229 & -0.022 \\ 1.388 \times 10^{-3} & -0.022 & 0.163 \end{bmatrix} kg \times m^2 \tag{A.0.5}$$

The final parameter of the scale bucket is the location of its center of gravity. The center of gravity of the real bucket is, measured relative to center of the lip

$$\overline{CG_B} = \begin{bmatrix} 0.029 \\ 1.161 \\ -0.754 \end{bmatrix} m \tag{A.0.6}$$

Using the scale factor again the scale bucket's center of gravity should be

$$\overline{CG_b} = s_f \times \overline{CG_B} = \begin{bmatrix} 1.588 \times 10^{-3} \\ 0.065 \\ -0.042 \end{bmatrix} m \tag{A.0.7}$$

The *CAD* model was designed using the parameters above. The final cad model had the following properties

$$
\begin{aligned}
m_b &= 7.891kg \\
\overline{\underline{I_b}} &= \begin{bmatrix} 0.158 & 0 & 0 \\ 0 & 0.23 & -0.014 \\ 0 & -0.014 & 0.161 \end{bmatrix} kg \times m^2 \\
\overline{CG_b} &= \begin{bmatrix} 0 \\ 0.062 \\ -0.042 \end{bmatrix} m
\end{aligned}
\tag{A.0.8}
$$

The deviation (E) of the scale bucket can now be calculated relative to the desired scale bucket

$$
\begin{aligned}
E_m &= 2.611\% \\
\overline{\underline{E_b}} &= \begin{bmatrix} 0.063\% & 100\% & 100\% \\ 100\% & 0.355\% & 34.4\% \\ 100\% & 34.4\% & 1.06\% \end{bmatrix} kg \times m^2 \\
\overline{CG_b} &= \begin{bmatrix} 100\% \\ 3.87\% \\ 0.32\% \end{bmatrix} m
\end{aligned}
\tag{A.0.9}
$$

At this point it should be noted that the reason for the large error in some of the parameters is that the real bucket is not symmetrical, whereas the *CAD* scale bucket is.

Appendix B

Test bench calibration

The following chapter discusses the processes used, as well as the results obtained, while calibrating the various sensors and aspects of the test bench.

B.1 Sensor calibration

B.1.1 Inclinometer

The inclinometer angles are gyro-stabilised and have a predefined orientation. Once the inclinometer was attached to the bucket, the bucket was orientated to coincide with the test bench inertial frame. Readings could then be taken to determine the initial orientation offset, namely, pitch θ_{ini}, yaw ψ_{ini} and roll ϕ_{ini}, from the inertial frame. These offsets were then adjusted so that the new reference coordinate system would coincide with the bucket's principal axes.

B.1.2 Load cells

There are two load cells in the test bench, one in each cable. To ensure that the cable lengths are equal and that there is no pre-tensioning, which would result in one cable exerting a greater force than the other, the following calibration was performed.

A bar was attached to the test bench parallel to that of the pulley shafts, see figure B.4. The cable was then pulled and the force in each of the cables measured. If one cable was longer than the other, the force in that cable would be greater.

From figure B.1 it can be seen that *cable 1* exerts a slightly greater force than that of *cable 2*. A variable link was inserted into each cable to allow fine adjustments to be made to the cable length. The above result was the best that could be achieved and the extra force can be contributed to material, manufacturing and assembling defects.

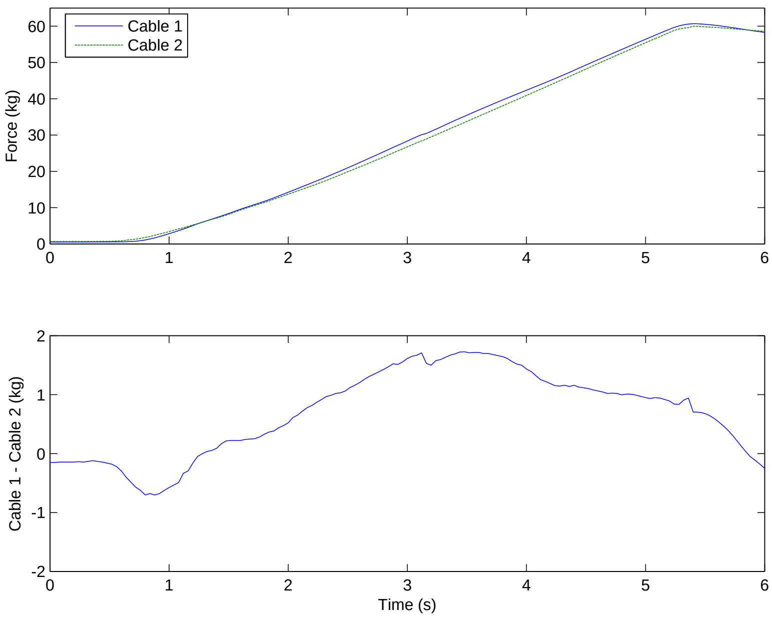


Figure B.1: Load cell calibration

B.1.3 Drag speed

The drag speed was controlled by means of a servo valve mounted on a hydraulic cylinder. The servo valve position is determined by an input current, which is generated by means of a voltage current converter. The assumption was made that the drag force was negligible compared to the force the hydraulic cylinder could provide.

To verify this assumption, tests were done to measure cylinder displacement. This was done by means of a linear variable differential transformer ($LVDT$). The position of the piston (dp) and the time (t) was recorded. The velocity of the piston (v_p) was determined as follows;

$$v_{p_t} = c_p(dp_t - dp_{t-1})/(t_t - t_{t-1}) \tag{B.1.1}$$

where c_p can be defined as the cylinder stroke divided by the $LVDT$ measuring range.

From figure B.2, a typical piston speed graph, it could be seen that the speed of the piston was unaffected by the drag force generated by the bucket. The $LVDT$ is connected in parallel to the piston cylinder. During drag, the force in the drag chain caused small rotations of the piston head. Once a rotation had occurred, the force in the apposing chain increased and produced a corrective moment. This effect can clearly be seen in figure B.2. There is

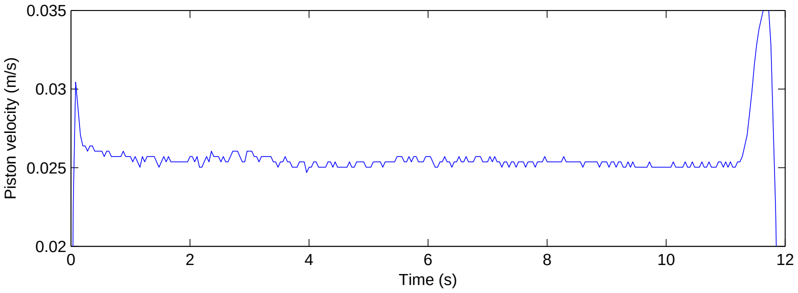


Figure B.2: Piston Speed vs Time

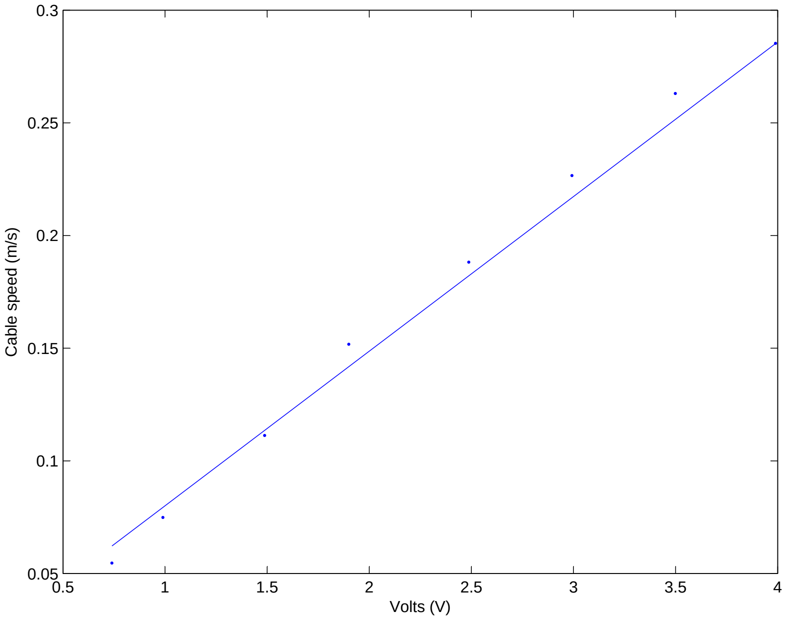


Figure B.3: Cable speed calibration graph

a finite non-zero distance between the axis of the cylinder and the axis of the *LVDT*, so the *LVDT* is susceptible to the rotation of the cylinder head.

Tests were conducted using different voltage inputs to determine the resultant piston velocities. The results can be see in figure B.3. A linear trend line was placed through the data points.

$$DragSpeed = c_v(0.0686V + 0.0115) \tag{B.1.2}$$

where V is the input voltage and c_v is the speed ratio between the piston and the drag cable, which can be obtained from equation B.1.6

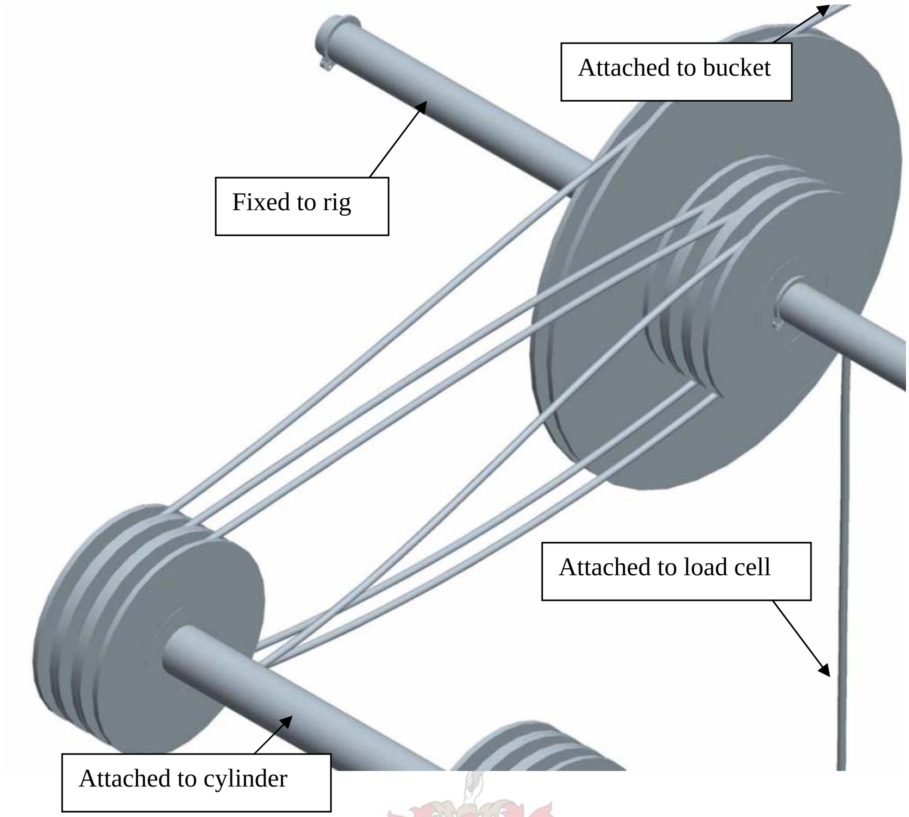


Figure B.4: Pulley assembly

For practical reasons, a cylinder with a stoke of $400mm$ was used. The complete drag length of a simulation was $2m$. An assembly of pulleys was used to multiply the effective stoke of the piston, see figure B.4. All the drag force and speed calculations were made with this pulley assembly already attached.

From figure B.4 it can be seen that the speed is going to be affected by the distance between the two shafts (d_{12}) since the pulleys are different sizes. To investigate this affect, a function was written to determine the speed variation.

The cable speed was determined using the increase in cable length per time step. It was, therefore, only necessary to calculate the parts of the cable length that change per time step.

The length of the cable (L_c) can, therefore, be written as

$$L_c = \sum \Delta L + \sum L_{const} \tag{B.1.3}$$

where ΔL and L_{const} represent the portions of the cable that change with respect to d_{12} and those that remain constant, respectively. Where ΔL is, see figure B.5

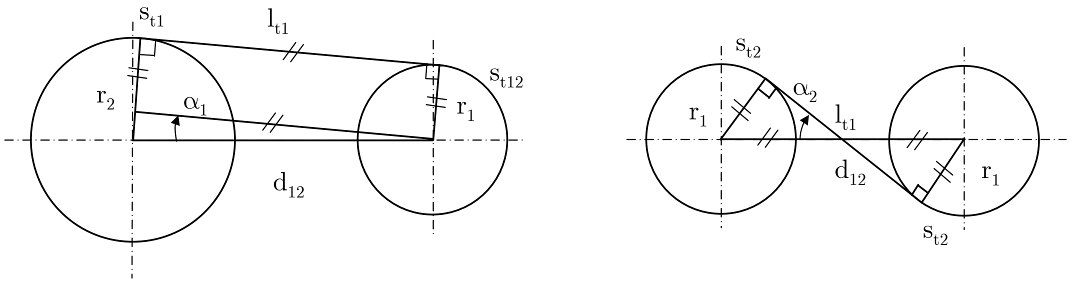


Figure B.5: ΔL cable calculations

$$\Delta L(\alpha_1, \alpha_2)_t = 2d_{12} + 2D_{12} + l_{t1} + l_{t2} + s_{t1} + st12 + 2s_{t2} \tag{B.1.4}$$

where

$$\begin{aligned} D_{12} &= \sqrt{d_{12}^2 + pulley_{width}^2} \\ l_{t1} &= \sqrt{((d_{12}cos(\alpha_1))^2 + pulley_{width}^2)} \\ l_{t2} &= d_{12}cos(\alpha_2) \\ s_{t1} &= \alpha_1 r_2 \\ s_{t12} &= (\pi - \alpha_1)r_1 \\ s_{t2} &= \alpha_2 r_1 \end{aligned} \tag{B.1.5}$$

The cable speed can now be determined as a multiple (c_v) of the piston speed, namely,

$$c_v = v_{c_{av}}/v_p \tag{B.1.6}$$

Where $v_{c_{av}}$ and v_p are the average speed and piston speed, respectively. For unity, a piston speed graph B.6 can be constructed. The maximum relative error in c_v is approximately 0.3 %.

B.1.4 Triangulation sensors

The linear displacement sensors were used to triangulate a point on the bucket. Once the sensors where installed on the test bench, the same triangulation section algorithm used to triangulate the bucket was used to triangulate the positions of the sensors using fixed points on the inertial frame.

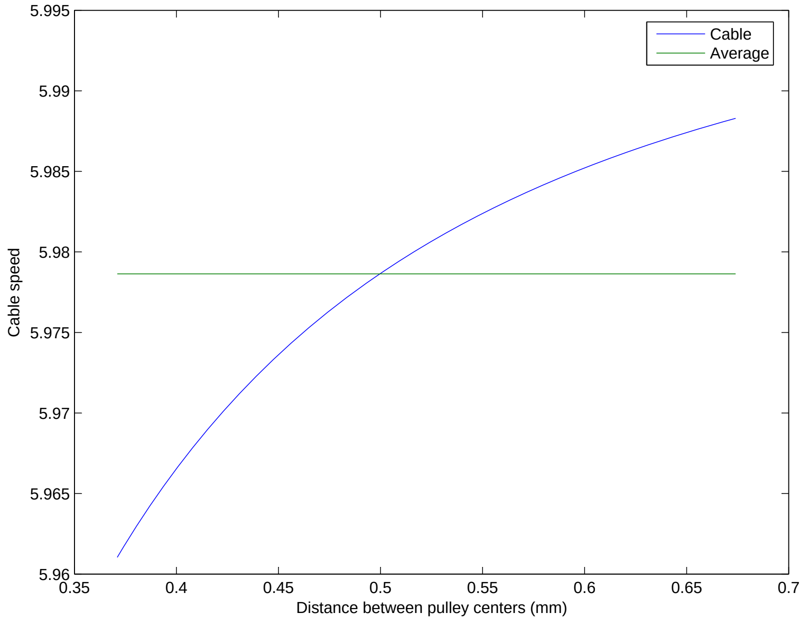


Figure B.6: Variation in c_v

Appendix C

Sample calculation

The sensors are attached to two different data loggers, namely a *Micro-Strain 3DM-G* onboard data logger build into the inclinometer, and a *HBM spider 8-30*. The two different data loggers sample at the same frequency. At every time step the following data is captured

t time in (s)

f_1, f_2 load cell 1 and 2 (mV/V)

dp displacement of cylinder (mV/V)

rv_1, rv_2, rv_3 displacement of triangulation sensors (V)

θ_i, ψ_i, ϕ_i pitch, yaw and roll in degrees

C.1 Drag force calculations

Two $100kg$ load cells where used to measure the drag force, one located at the base of each of the drag cables. The drag force measured therefore will be influenced by the pulley assembly. To minimize this effect the pulleys were mounted on bearings to allow them to turn independently of each other, they however do not affect the friction generated between adjacent pulleys. The weight of the load cells also add a contribution to the drag force.

These affects however are marginal compared to that of the drag force generated by the bucket and therefore have been neglected. The load cells were placed at the base of the cable and not directly on the active part of the the cable to minimize the effect on the dynamic of filling cycle.

Each load cell consisted of a full bridge transducer rated at $2mV$. Test where done to verify this. The drag force could then be determined simply as

$$F_i = (f_i/2mV) \times 100kg \tag{C.1.1}$$

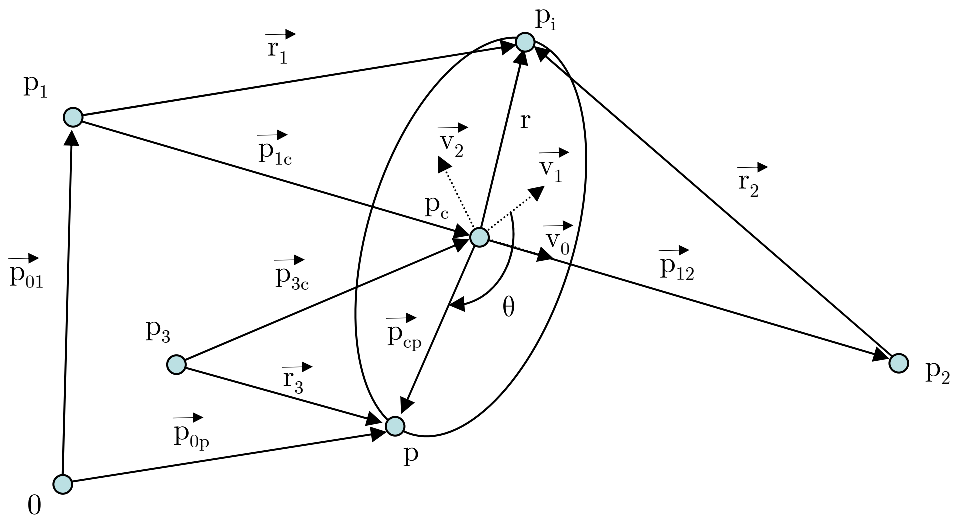


Figure C.1: Spacial representation of triangulation algorithm

C.2 Piston speed

A linear variable differential transformer ($LVDT$) was attached to the piston. The piston speed was determined by

$$v_{p_t} = c_p(dp_t - dp_{t-1})/(t_t - t_{t-1}) \qquad \text{(C.2.1)}$$

where c_p can be defined as the cylinder stroke divide by the $LVDT$ measuring range.

C.3 Arch anchor location

The arch anchor position ($\overline{p_a}_s$) was triangulated using three linear cable position sensors. The sensors had a $2m$ measuring length and were calibrated to $10V$. The sensor cable was spring loaded to ensure that it was always taught.

The distance (r_i) from it's sensor ($\overline{p_i}_s$) is

$$r_i = (rv_i/10V) \times 2m \qquad \text{(C.3.1)}$$

Spheres can be generated about each point ($\overline{p_i}^s$) with it's given radius (r_i). The point where the three spheres intersect will be the location of the arch anchor $\overline{p_a}_s$.

The intersection of the spheres generated from $\overline{p_1}^s$ and $\overline{p_2}^s$ with radii r_1 and r_2 respectively can be viewed as as circle in space. Let this circle's center be located at $\overline{p_c}_s$ and have a radius r. Define $\vec{v_0}^r$ as the unit vector from $\overline{p_1}^s$ to $\overline{p_2}^s$. Point $\overline{p_c}^s$ can now be defined as

$$\overline{p_{cr}} = \overline{p_1}^r + x \cdot \vec{v_0}^r \tag{C.3.2}$$

where x is an unknown scalar. Define $\overline{p_i}^r$ as a random point on the intersecting circle. From pythagoras the following two equations can be obtained

$$\begin{aligned} r_1^2 - x^2 &= r^2 \\ r_2^2 - (\|\overline{p_2}^s - \overline{p_1}^s\| - x)^2 &= r^2 \end{aligned} \tag{C.3.3}$$

Solving the equation above yields

$$\begin{aligned} x &= (r_1^2 + (\|\overline{p_2}^s - \overline{p_1}^s\|)^2 - r_2^2)/(2\|\overline{p_2}^s - \overline{p_1}^s\|) \\ r &= \sqrt{r_1^2 - x^2} \end{aligned} \tag{C.3.4}$$

An equation can now be constructed to define any point p on the circle. The circle however is not necessary on a plane that coincides with our inertial coordinate system. Define $\vec{v_1}$ and $\vec{v_2}$ as

$$\begin{aligned} \vec{v_1} &= \vec{v_0} \times \vec{v_t} \\ \vec{v_2} &= \vec{v_0} \times \vec{v_1} \end{aligned} \tag{C.3.5}$$

where $\vec{v_t}$ is a random vector. $\vec{v_0}$,$\vec{v_1}$ and $\vec{v_2}$ now form and orthogonal set. The circle lies on a plane defined by $\vec{v_1}$ and $\vec{v_2}$. Any point on the circle can now be $\overline{p_i}^r$ can now be expressed as a function of the the angle θ

$$\overline{p_i(\theta)}^r = x\vec{v_0} + rcos(\theta)\vec{v_1} + rsin(\theta)\vec{v_2} \tag{C.3.6}$$

converting $\overline{p_c}^s$ to the rotational frame yields

$$\overline{p_3}^r = c\vec{v_0} + a\vec{v_1} + b\vec{v_2} \tag{C.3.7}$$

where

$$\begin{aligned} a &= \overline{p_c}^s \cdot \vec{v_1} \\ b &= \overline{p_c}^s \cdot \vec{v_2} \\ c &= \overline{p_c}^s \cdot \vec{v_0} \end{aligned} \tag{C.3.8}$$

There are now 2 points on the intersecting circle which will satisfy the following pythagorean equation

$$r_3^2 = (rcos(\theta) - a)^2 + (rsin(\theta) - b)^2 + (x - c)^2 \tag{C.3.9}$$

Because points $\overline{p_1}^s$, $\overline{p_2}^s$ and $\overline{p_3}^s$ lie on a plane, there are two solutions to equation C.3.9, one above and one below the plane. Points $\overline{p_1}^s$, $\overline{p_2}^s$ and $\overline{p_3}^s$

must therefore be chosen such that $\overline{p_a}^s$ always remain on one side of the plane to avoid ambiguity.

Equation C.3.9 can be simplified to

$$sin(u+\theta) = C/\sqrt{a^2+b^2} \tag{C.3.10}$$

where

$$\begin{aligned} sin(u) &= a/\sqrt{a^2+b^2} \\ C &= (r_3^2 - r^2 - a^2 - b^2 + x^2 - c^2)/2r \end{aligned} \tag{C.3.11}$$

Once θ is known, equation C.3.7 will yield two answers for $\overline{p_a}^r$. Choosing one and converting $\overline{p_a}^r$ back to the internal frame ($\overline{p_a}^s$), the arch anchor's position can be calculated

$$\overline{p_a}^s = \overline{p_3}^s + \overline{p_a}^r \tag{C.3.12}$$

C.4 Bucket orientation

The inclinometer had its own onboard data converter. The sampling rate was set to coinside with that of the *HBM spider 8-30*. The system was allowed to settle into a state of equilibrium. As soon as bucket began to move, a clear spike could be seen in both data sets. The two data sets could then be synchronized, using this spike as a reference.

The inclinometer has a fixed zero position, so all the readings obtained had an initial offset, see section B.1.1. The buckets orientation is therefore

$$\begin{aligned} \theta &= \theta_i - \theta_{ini} \\ \psi &= \psi_i - \psi_{ini} \\ \phi &= \phi_i - \phi ini \end{aligned} \tag{C.4.1}$$

C.5 Center of gravity

Using the arch anchor position ($\overline{p_a}^s$) and the buckets orientation (θ, ψ, ϕ) the center of gravity calculated. Using the vector ($\overline{p_{ac}}^r$) from the arch anchor to the center of gravity ($\overline{p_{cg}}^s$) in the buckets principal coordinate frame, the center of gravity can be calculated

$$\overline{p_{cg}}^s = \overline{p_a}^s + \underline{\overline{E}}_r^s \cdot \overline{p_{ac}}^r \tag{C.5.1}$$

where

$$\overline{\underline{E}}_r^s = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & -\sin(\theta) \\ -\sin(\psi)\cos(\phi)+\cos(\psi)\sin(\phi) & \cos(\psi)\cos(\phi)+\sin(\psi)\sin(\phi) & \cos(\theta)\sin(\phi) \\ \sin(\psi)\sin(\phi)+\cos(\psi)\cos(\phi) & -\cos(\psi)\sin(\phi)+\sin(\psi)\cos(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}^T \tag{C.5.2}$$

List of References

Alder, B.J. and Wainwright, T.E. (1967 Jun). Velocity autocorrelations for hard spheres. *Phys. Rev. Lett.*, vol. 18, pp. 988–990.

Cleary, P.W. (1997 Sep). Modelling industrial granular flows for draglines and centrifugal mills. *Australian Journal of Mining*, p. 67.

Cleary, P.W. (1998). The filling of dragline buckets. *Mathematical Engineering in Industry*, vol. 7, no. 1, pp. 1–24.

Cleary, P.W., Stokes, N. and Hurley, J. (1997). Efficient collision detection for three dimensional super-ellipsoidal particles. In: *Computational Techniques and Applications: CTAC97*. World Scientific.

Coetzee, C.J. (2000). *Forced Granular Flow*. Msc Thesis Mechanical Engineering, University of Stellenbosch, Stellenbosch, South Africa.

Cundall, P.A. and Strack, O.D.L. (1979). A discrete numerical model for granular assemblies. *Géotechnique*, vol. 29, no. 1, pp. 47–65.

Els, D.N.J. (2003). Approximate numerical integrator for rotation quaternion kinematic equations. Personal communique.

Hertz, H. (1882). J.fur die reine u. angew. *Math 92, 136.*

Itasca (). *PFC3D - Particle Flow Code in 3 Dimensions*, version 3.0 edn.

Mindlin, R.D. and Deresiewicz, H. (1953). Elastic spheres in contact under varying oblique forces. *Trans. ASME, Series E, Journal of Applied Mechanics*, vol. 20, p. 327.

P Dayawansa, G Chitty, B.K.H.B. and Price, J.W.H. (2004). Fracture mechanics of mining dragline booms.

Vu-Quoc, L., Zhang, X. and Walton, O.R. (2000). A 3-d discrete-element method for dry granular flows of ellipsoidal particles. *Comput. Methods Appl. Mech. Engrg.*, vol. 187, pp. 483–528.

Walton, O.R. and Braun, R.L. (1986). Viscosity, granular-temperature, and stress calculations for shearing assemblies of inelastic, frictions discs. *Journal of Reology*, vol. 30, no. 5, pp. 949–980.