Construction of Force Networks of Static Granular Systems under Constraints

A Thesis

Submitted For the Degree of

MASTER OF SCIENCE (ENGINEERING)

in the Faculty of Sciences

by

Dheeraj Kumar

THEORETICAL SCIENCES UNIT
JAWAHARLAL NEHRU CENTRE FOR ADVANCED SCIENTIFIC RESEARCH
Bengaluru

NOVEMBER 2019

To Yagyik, Deepak and Biswadeep

DECLARATION

I hereby declare that the matter in the thesis entitled "**Construction of Force Networks of Static Granular Systems under Constraints**" is the result of investigations carried out by me at the Theoretical Sciences Unit, Jawaharlal Nehru Centre for Advanced Scientific Research, Bengaluru, India under the supervision of Prof. Srikanth Sastry and that it has not been submitted elsewhere for the award of any degree or diploma. In keeping with the general practice in reporting scientific observations, due acknowledgement has been made whenever the work described is based on findings of other investigators.

Dheeraj Kumar

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

Prof. Srikanth Sastry

Acknowledgements

I thank my supervisor Prof. Srikanth Sastry for his guidance. I also thank my labmates especially Anshul, Vinutha and Yagyik for their help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to granular systems

A granular system is defined as a large collection of macroscopic grains (linear dimensions greater than 1 μm) of solid matter, with thermal fluctuations being irrelevant. Some examples of granular matter are sand, rice, coffee, salt, etc. This kind of matter is very important in industry. The second most manufactured class of materials are in the form of granular matter [2].

Granular materials can flow like a liquid (like sand in an hourglass), resist deformation like a solid (like the sand under feet at the beach), or quickly transition between these states (like pebbles in a rockslide). Granular materials have properties that have no equivalent in regular materials like wood, metal, or rubber. In solids like these a force applied to the surface propagates through the material smoothly and predictably. If a uniform force is applied to the surface of a material, every equally sized cross-section of that material bears the same amount of load. In granular materials, however, the situation is very different: in a sand pile under stress (that is, when a force is applied to its surface), the force is distributed unevenly – some individual sand grains bear far more load than others. Surprisingly, this remains true even when the sand grains themselves are identical. Whats more, the load-bearing grains connect to one another to make a fractal, lightning-like pattern inside the

material, like that shown in Figure 1.1 [1]. These string-like arrangements of load-bearing grains are called force chains.

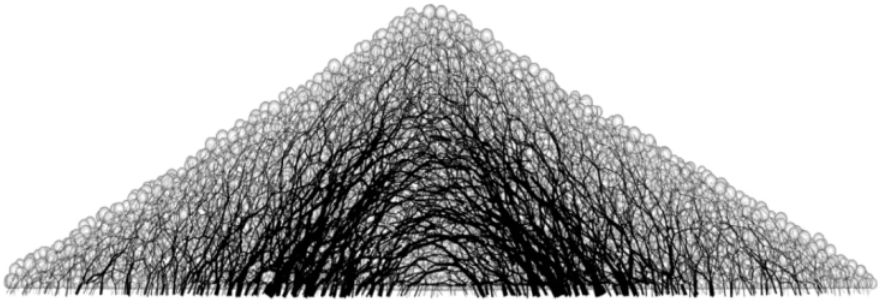

Figure 1.1: Force chains in a computer simulation of a sand pile. Taken from reference [1]. The thickness of a black line indicates the magnitude of the force at that point inside the sand pile. The network of force chains (black lines) form what we call a *force network* in the above pile of grains.

1.2 Force networks in static granular systems

The fascinating properties of static granular matter are closely related to the organization of the interparticle contact forces into highly heterogeneous force networks [2]. Force networks form the skeleton of static granular matter [3] [4]. They are the key factor that determines mechanical properties such as stability, elasticity and sound transmission, which are important for civil engineering and industrial processing [5].

In this thesis, we study cohesionless systems of hard disks which are in mechanical equilibrium. For a given arrangement of disks and boundary conditions (the boundary conditions in the systems we consider are the external forces acting at the boundary of the system), the unknowns are the contact forces. The usual situation is that the total number of unknown forces is significantly greater than the total number of balance equations requiring mechanical equilibrium. The additional conditions are inequalities (for example: requirement of no cohesion at each grain contact) that partly reduce the *space* of admissible *solutions*[1], but the multiplicity of the solutions that

[1]By a solution we mean a force network, the contact forces of which satisfy the balance equations requiring mechanical equilibrium subject to additional constraints (*if any*).

is left is still very large. As a simple illustration, it is obvious that since the number of balance equations is fixed by number of particles, an increasing number of contact per grain will lead to a larger number of undetermined contact forces. In summary, the list of the position of all the grains and contacts is in general not sufficient to determine the precise *state*[2] of a static packing of grains submitted to some given external load. That is, there are more unknown forces to be determined than balance equations that require mechanical equilibrium. This is also called the '*hyperstatic* case'. As we shall see in Chapter 2, the hyperstatic case implies that there are infinite number of force networks consistent with the force balance condition each of which can hold the granular system in mechanical equilibrium. The space of such solutions reduce when extra conditions, such as, zero cohesion at each contact is imposed. This thesis aims at finding a minimum set of states or force networks of a hyperstatic granular packing (consisting of cohesionless grains) of given particle size, arrangement and boundary conditions using which any other state or force network can be constructed. In addition to seeking a minimum number of force networks that obey these constraints we also characterise their space by changing parameters such as system size, coordination number and geometry by taking some simple arrangement of disks in 2-dimensions as described in the next section.

1.3 Polytopes in static granular systems

In order to understand the dependence of the minimum number of force networks necessary to build any other force network consistent with the force balance condition and zero-cohesion we study some simple toy probelms of different geometry and system sizes. For this purpose we consider three simple kinds of hyperstatic system. These systems consist of identical frictionless hard disks in 2 dimensions which have repulsive forces at each contact. The boundaries of the system are also assumed to be frictionless. The three kinds

[2]A state and a solution mean the same thing here. Figure 1.1 shows one force network and thus corresponds to one state for the given pile of sand.

of systems are:

1. Square lattice: Identical 2 dimensional disks packed in a 2 dimensional square box.

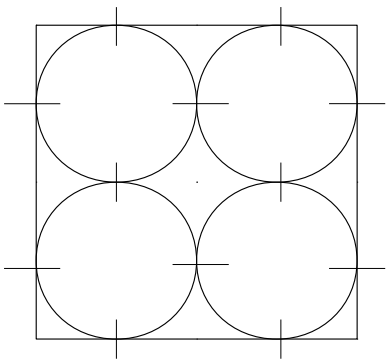

Figure 1.2: The smallest square lattice. Contacts are also marked in the figure. Each particle has four neighbours in contact.

2. Rhombic lattice: Identical 2 dimensional disks packed in a 2 dimensional rhombic box.

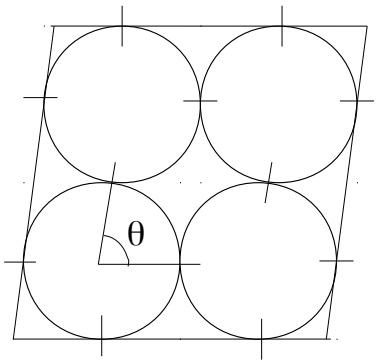

Figure 1.3: The smallest rhombic lattice. Also marked is the interior acute angle of the rhombic lattice and contacts. Each particle has four neighbours in contact.

3. Equilateral-triangular lattice: Identical 2 dimensional disks packed in an equilateral trianglular box.

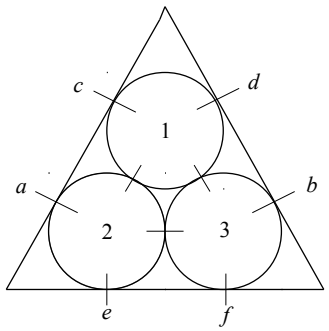

Figure 1.4: The smallest equilateral triangle lattice, with contacts marked in the figure. The numbers and the letters will be later used to identify the forces at contacts in section 2.1. The boundary forms an equilateral triangle and disks in the interior for large lattice of this kind will have six neighbours in contact.

In section 2.1 of Chapter 2 we show how one can associate, abstractly, a *convex polytope*[3] with each kind of these systems. For a given number of disks, their arrangement and boundary conditions, the geometry of the polytope and its orientation in space is fixed. Each vertex of this polytope then corresponds to a force network, the forces of which obey the conditions of mechanical equilibrium and zero cohesion. A complete list of these vertices of the polytope is sufficient to construct any other force network that obeys the constraints of mechanical equilibrium and zero cohesion. We elaborate on these statements in section 2.1 of Chapter 2 by solving an example problem.

[3]A polytope is a geometric object with "flat" sides. It is a generalization in any number of dimensions of the three-dimensional polyhedron. A convex polytope is a special case of a polytope, having the additional property that it is also a convex set of points (see section 2.1.4) in the n-dimensional space.

1.4 Brief summary

We study the effect of coordination number and arrangement of particles on the number of vertices of the polytope as a function of system size. This is essential to determine the feasibility of finding the complete set of vertices of polytopes corresponding to static granular systems that have very large number of grains or particles. As stated before, the knowledge of complete set of vertices, as we shall see in Chapter 2, will help us construct any force network compatible with balance equations and additional constraints (for example: requirement of zero cohesion at each contact) for a given pile of grains. Based on the results of the three kinds of systems that we study we show the following:

1. The number of vertices of the polytope corresponding to the equi-triangular lattice increase exponentially with system size (system size equals number of particles in the system).

2. The polytopes corresponding to square lattices are *simplices.*[4] The number of vertices of polytopes corresponding to square lattices increase as a square root of the system size.

3. The coordination number – number of neighbours in contact with a particle – plays a big role in determining the number of vertices of the polytope for the same system size. For example, the growth of the number of vertices of polytope corresponding to a rhombic lattice, with θ(in figure 1.3) very close to but more than 60-degrees, could be bounded polynomially with respect to system size, while at 60-degrees this growth becomes exponential[5].

4. For the same number of contacts and system size the arrangement of particles also influence the total number of vertices. Rhombic lattices

[4] A simplex (plural: simplices) is a generalization of a tetrahedron to arbitrary dimensions. The number of vertices of a simplex is exactly one more than the dimension in which they lie.

[5] In figure 1.3, for large system size, it can be seen that when θ changes from an angle greater than 60-degrees to 60-degrees the coordination number of particles in the interior of rhombic lattice change from four to six.

of a given system size with θ, as in figure 1.3, tending towards 60-degrees have more vertices than ones tending towards 90-degrees. As this angle varies from 90-degrees to 60-degrees the number of vertices increase monotonically.

5. We also show, based on the results on square and rhombic lattices, that for a lower coordination number of four the number of vertices with respect to system size can be bounded polynomially. This implies that it may be feasibile to find all the vertices of polytopes corresponding to larger system sizes with lower coordination number.

6. We have also proposed and demonstrated methods to do the following:

   (a) Bounding randomly oriented open polyhedra which have the property that no two points in or on it subtend angle greater than or equal to $90-$degrees at origin.

   (b) Approximating volume of randomly oriented polytopes.

   The utility of these methods in regard to solving the main problem of this thesis will become clear in the methods section.

Chapter 2

Methods

As stated before, the aim of this thesis is to find a minimum set of states of a hyperstatic granular packing (with cohesionless grains), for a given particle size, arrangement of particles and boundary conditions, using which any other state can be constructed. A state corresponds to one network of interparticle contact forces which can keep the system in mechanical equilibrium. To demonstrate how we achieve our objective we take a system of 3 frictionless disks in 2 dimensions packed in an equilateral triangular box.

2.1 A solved example: The snooker-triangle problem

The smallest system consists of three disks as shown in the figure 2.1. The disks are identical and frictionless in nature. The boundary is also considered frictionless. Thus only normal force exist at each contact. The unknowns are the contact forces (normal forces) which exist at the disk-disk and the disk-boundary interface. These are also marked in the figure 2.1.

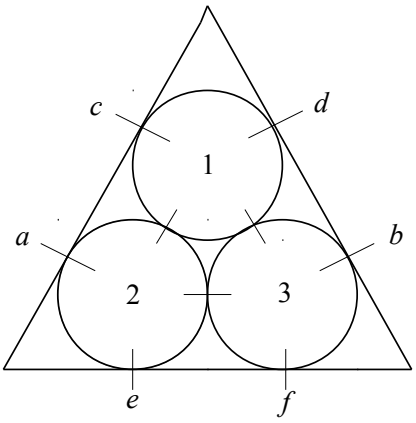

Figure 2.1: Smallest System Size

2.1.1 Contact Matrix A

The force balance condition for the i_{th} disk can be written as:

$$\vec{f}_i = \sum_j \vec{f}_{ij} = 0, \tag{2.1}$$

where, $\vec{f}_{ij}$ is the force acting on the i_{th} particle due to j_{th} contact. The j_{th} contact can be a neighbouring disk or the boundary.

We use a cartesian coordinate system to write down the force balance equation for each disk shown in figure 2.1. The origin "O" is at the centre of disk 2 and the positive x and y directions are assumed to be as shown in figure 2.2. In two dimensional frictionless case each i, that is, each disk contributes to two equations: one coming from balance of forces along x-direction and the other from the balance of forces along y-direction.

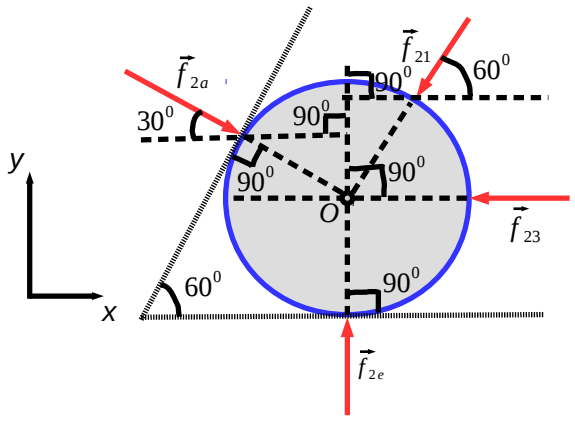

Figure 2.2: Free body diagram of disk 2 in figure 2.1

The red arrows in figure 2.2 represent the assumed direction of unknown forces acting on disk 2 of figure 2.1 due to contact with disk 1 and 3 and the boundary. These forces fall along the line joining the centre of the circle to the circumference because the disks are frictionless in nature. These are represented as:

- $\vec{f}_{23}$: The force acting on disk 2 due to contact with disk 3.
- $\vec{f}_{21}$: The force acting on disk 2 due to contact with disk 1.
- $\vec{f}_{2a}$: The force acting on disk 2 due to contact with boundary at a location a marked in figure 2.1.
- $\vec{f}_{2e}$: The force acting on disk 2 due to contact with boundary at a location e marked in figure 2.1.

The particular angles (in degrees) at which these forces act as shown in figure 2.2 is due to the fact that the boundary forms an equilateral triangle and the disks are identical in nature. Equation 2.1 is a vector equation which for disk 2 in the cartesian coordinate system give the following two equations:

$$-f_{23} - f_{21}\cos 60^0 + f_{2a}\cos 30^0 = 0, \text{ for } x\text{-direction},$$

$$-f_{21}\sin 60^0 - f_{2a}\sin 30^0 + f_{2e} = 0, \text{ for } y\text{-direction}.$$

Where, f_{ij} represents the magnitude of the force acting on i_{th} disk due to j_{th} contact. Collecting similar such equations for disk 1 and 3 we obtain the following sytem of force balance equations:

$$\begin{bmatrix} -1 & -0.5 & 0 & 0.866 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.866 & 0 & -0.5 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0.5 & 0 & -0.866 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.866 & 0 & -0.5 & 0 & 0 & 0 & 1 \\ 0 & 0.5 & -0.5 & 0 & 0 & 0.866 & -0.866 & 0 & 0 \\ 0 & 0.866 & 0.866 & 0 & 0 & -0.5 & -0.5 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} f_{23} \\ f_{21} \\ f_{31} \\ f_{2a} \\ f_{3b} \\ f_{1c} \\ f_{1d} \\ f_{2e} \\ f_{3f} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.2}$$

Equation 2.2 is a system of equations which we refer to as $\mathbf{Ax}=\mathbf{0}$[1]. Here, $\mathbf{A}$ is a matrix, we call it the contact matrix, of size 6×9 and $\mathbf{x}$ is an unknown vector whose components consist of the magnitude of $\vec{f}_{ij}$. While writing equation 2.2 the force acting on the i_{th} particle due to j_{th} contact is assumed to be acting towards the centre of the i_{th} particle. We have also made use of the Newton's third law, that is, $\vec{f}_{ij} = -\vec{f}_{ji}$ via the free body diagrams of disks and noting in equation 2.2 that the magnitude of force acting on i_{th} particle due to j_{th} contact is same as the magnitude of force acting on j_{th} particle due to i_{th} contact, that is, $f_{ij} = f_{ji}$. For example, in addition to assuming that the force $\vec{f}_{32}$ points towards the center of disk 3, a replacement of the kind $f_{32} = f_{23}$ has been made while writing equation 2.2. Together they ensure that the Newton's third law is obeyed for each particle at each contact.

[1]In this thesis we will stick with the notation $\mathbf{A}$ to mean the contact matrix obtained using equation 2.1 and $\mathbf{x}$ as a vector of unknowns consisting of magnitude of forces at each contact.

We note that the number of unknown forces is more than the number of force balance equations. And therefore there will be infinite solutions to **Ax=0**. This is the hyperstatic case.

2.1.2 Constraint on Ax=0

Since we are interested in cohesionless disks only we impose the following constraint on the unknown vector **x**: $x_i \geq 0 \ \forall \ i$. This ensures that the normal force exerted on a given particle is compressive in nature.

We aim to find a minimum number of solutions to **Ax=0** (subject to $x_i \geq 0 \ \forall \ i$) using which any other solution to **Ax=0** (such that $x_i \geq 0 \ \forall \ i$) can be found.

2.1.3 The inequality representation of the problem: Zc $\geq$ 0

We now convert the problem into a more convenient form. Let the rank of the contact matrix **A** be r. If **A** is of the size $m \times n$ and $m < n$ (which is true in our problem because the number of unknown contacts are more than the number of force balance equations), then the space of solutions of **Ax=0** is $n - r$ dimensional. We call this $n - r$ dimensional space the null-space of matrix **A**. Let the vectors $\boldsymbol{z_1}, .., \boldsymbol{z_{n-r}}$ be an orthogonal set of basis vectors of the null-space of matrix **A**. Then, any solution **x** of **Ax=0** can be expanded in the basis of null-space of **A** in the following way:

$$\mathbf{x} = c_1\boldsymbol{z_1} + c_2\boldsymbol{z_2} + .. + c_{n-r}\boldsymbol{z_{n-r}} \tag{2.3}$$

where, $c_1, c_2, .., c_{n-r}$ are the scalar expansion coefficients and $\boldsymbol{z_1}, \boldsymbol{z_2}, .., \boldsymbol{z_{n-r}}$ are the basis vectors of the null space of matrix **A**[2].

For the 3-disk problem, the rank of the matrix **A** in equation 2.2 is 6 and its size is 6 $\times$ 9. Hence the nullspace of **A** (that is, the space of solutions

[2]Note that since **A** is of the size $m \times n$ and **Ax=0**, therefore, **x** is of the size $n \times 1$. Hence, the solutions **x** of **Ax=0** are vectors with n components embedded in a space which is $n - r$ dimensional. This $n - r$ dimensional space is the null-space of matrix **A**.

to **Ax=0**) is 3-dimensional. Expanding the vector **x** in an orthogonal basis (accurate upto first decimal place) of nullspace of **A** we get the following:

$$\begin{bmatrix} f_{23} \\ f_{21} \\ f_{31} \\ f_{2a} \\ f_{3b} \\ f_{1c} \\ f_{1d} \\ f_{2e} \\ f_{3f} \end{bmatrix} = \begin{bmatrix} -35 & 31 & -22 \\ 45 & 18 & -18 \\ 0 & 15 & 50 \\ -14 & 46 & -36 \\ -40 & 45 & 3 \\ 26 & 27 & 47 \\ 52 & 29 & 8 \\ 32 & 39 & -34 \\ -20 & 35 & 45 \end{bmatrix} \times \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \tag{2.4}$$

We refer to the above system of equations in equation 2.4 as **x=Zc**[3]. Where **x** is the unknown vector in **Ax=0** and **Z** consists of column vectors which form an orthogonal basis of nullspace of matrix **A**. These basis vectors have the same length in the space they lie. The **c** vector consists of the coefficients that express the vector **x** in the basis of nullspace of matrix **A**. Imposing the **componentwise non-negativity constraint on vector x** we get the following system of inequalities:

[3]In this thesis we will stick with the notation **Z** to mean a matrix consisting of column vectors which are basis vectors of nullspace of **A** matrix and **c** as a vector consisting of components that are coeffiecients of expansion of **x** in the basis of nullspace of **A**. Also, note that the column vectors of the **Z** matrix have been obtained by using the "null" command available in MATLAB which gives an orthogonal basis of null-space of any matrix **A**.

$$\begin{bmatrix} -35 & 31 & -22 \\ 45 & 18 & -18 \\ 0 & 15 & 50 \\ -14 & 46 & -36 \\ -40 & 45 & 3 \\ 26 & 27 & 47 \\ 52 & 29 & 8 \\ 32 & 39 & -34 \\ -20 & 35 & 45 \end{bmatrix} \times \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \geq 0 \tag{2.5}$$

Equation 2.5 consists of nine inequalities in three variables c_1, c_2 and c_3 with a zero vector on the right hand side. We refer to the system of inequalities in equation 2.5 as $\mathbf{Zc} \geq 0$. Now, instead of solving for unknown vector $\mathbf{x}$ such that $\mathbf{Ax=0}$ with component-wise non-negativity criteria on $\mathbf{x}$, that is, $x_i \geq 0 \; \forall \; i$, we instead solve for unknown vector $\mathbf{c}$ such that $\mathbf{Zc} \geq 0$ and then use the relation $\mathbf{x=Zc}$ to obtain the actual force balanced solution[4]. It should be noted that $\mathbf{Z}$ consists of column vectors which are independent, and therefore by the relationship $\mathbf{x=Zc}$, corresponding to a $\mathbf{c}$ vector there exists a unique $\mathbf{x}$ vector.

2.1.4 Geometry and Boundedness of $\mathbf{Zc} \geq \mathbf{0}$

A convex region is a region such that, for every pair of points within the region, every point on the straight line segment that joins the pair of points is also within the region. Formally it is defined as follows:

If S is a convex region [10] in n-dimensional space, then for any collection of r $(r > 1)$ n-dimensional vectors $\boldsymbol{u_1}, ..., \boldsymbol{u_r}$ in S, and for any nonnegative numbers $\lambda_1, ..., \lambda_r$ such that $\lambda_1 + ... + \lambda_r = 1$, one has:

[4]We emphasise that an $\mathbf{x}$ vector consist of components which are magnitude of unknown forces, see equation 2.2 for example. On the other hand a $\mathbf{c}$ vector consists of components which are scalar expansion coefficients which express an $\mathbf{x}$ vector in the basis of nullspace of matrix $\mathbf{A}$, see for example equation 2.3 or 2.4.

$$\sum_{k=1}^{r} \boldsymbol{\lambda_k u_k} \in \boldsymbol{S}$$

A vector of this type is known as a convex combination of $\boldsymbol{u_1}, ..., \boldsymbol{u_r}$. Since an intersection of convex regions is convex [10] and a linear inequality geometrically is a *halfspace*[5] which is convex, the region defined by $\mathbf{Zc} \geq 0$, which is an intersection of halfspaces, is also convex therefore.

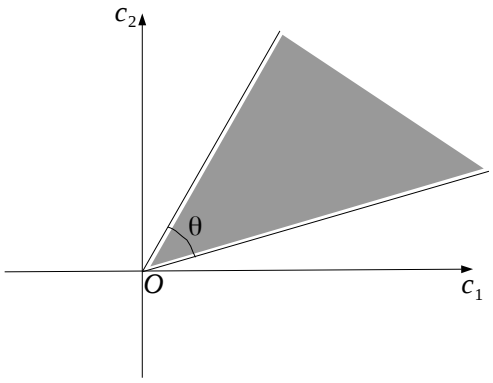

Figure 2.3: Convexity of $\mathbf{Zc} \geq 0$, for example, in two dimensional case implies that $\theta \leq$ 180-degrees

Next we use a cartoon figure of $\mathbf{Zc} \geq 0$ to highlight some of its basic geometric properties. In figure 2.3, for the sake of simplicity, we have assumed that the space of $\mathbf{c}$ vectors is 2-dimensional. The region in which $\mathbf{Zc} \geq 0$ will be satisfied in this 2-dimensional space is shaded in grey. The boundaries of this region are straight lines because $\mathbf{Zc} \geq 0$ consists of linear inequalities only. In higher dimension these boundaries would become straight edges and flat planes. In figure 2.3 one must also note that a line segment joining any two points in the shaded region can not subtend an angle greater than $180 - degrees$ at the origin. This is true because the the region defined by

[5] A half-space is either of the two parts into which a plane divides an n-dimensional space.

$\mathbf{Zc} \geq 0$ is convex. This also implies that $\theta \leq 180$-degrees in figure 2.3. The convex polyhedra defined by $\mathbf{Zc} \geq 0$ is also unbounded because the boundary of a half-space corresponding to any inequality in $\mathbf{Zc} \geq 0$ passes through the origin. The unboundedness of $\mathbf{Zc} \geq 0$ implies that we can find $\mathbf{c}$ vectors in the shaded region which lie at infinite distance from origin.[6]

2.1.5 Bounding the polyhedra $\mathbf{Zc} \geq 0$

Our goal is to pick a point from every edge of the open polyhedra $\mathbf{Zc} \geq 0$. For example, in figure 2.3 there would be two such points corresponding to the two edges shown as solid black lines emanating from the origin. Using these edge points we can obtain any other point $\mathbf{c}$ satisfying $\mathbf{Zc} \geq 0$ and hence by the relationship $\mathbf{x}$=$\mathbf{Zc}$ any force network that the three-disk system might admit. We will demonstrate this through a cartoon figure as below:

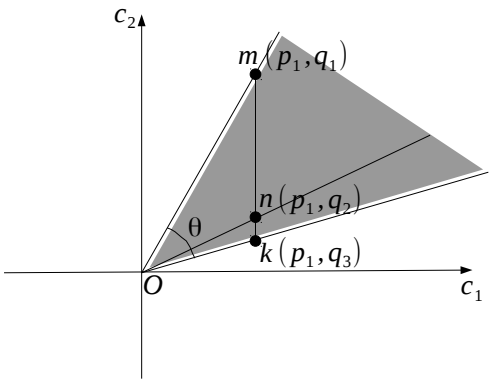

Figure 2.4: Spanning $\mathbf{Zc} \geq 0$ using vertices on the edges of $\mathbf{Zc} \geq 0$

We are in the space of $\mathbf{c}$ vectors[7]. In this space of vectors the shaded

[6]Note that in figure 2.3 a $\mathbf{c}$ vector has two components c_1 and c_2.

[7]If $\mathbf{c}$ is an n-dimensional vector then the space of $\mathbf{c}$ vectors would consist of all the n-dimensional vectors that might exist. This space is a vector space by definition which we also call the $\mathbf{c}$-space later on in the thesis. The set of $\mathbf{c}$ vectors which satisfy $\mathbf{Zc} \geq 0$ is a subset of the $\mathbf{c}$-space.

region in figure 2.4 satisfy the set of linear inequalities $\mathbf{Zc} \geq 0$. We want to show through figure 2.4 that knowing a point from every edge of the open polyhedra $\mathbf{Zc} \geq 0$ can help us construct any other point $\mathbf{c}$ consistent with $\mathbf{Zc} \geq 0$. To see this let us assume that we have found a point from each edge of the open polyhedra $\mathbf{Zc} \geq 0$. These points are marked as k and m with coordinates (p_1, q_3) and (p_1, q_1) respectively in figure 2.4. Now, a convex combination of these two points gives a third point with coordinates:

$$(p_1, q_2) = \lambda_1(p_1, q_1) + \lambda_2(p_1, q_3) = (p_1(\lambda_1 + \lambda_2), (\lambda_1 q_1 + \lambda_2 q_3)),$$

$$where, \lambda_1 + \lambda_2 = 1, \ and \ \lambda_1, \ \lambda_2 \geq 0$$

It is easy to see through the above relationship that the third point with coordinates (p_1, q_2) will lie on the line joining the points k and m. Let this point with coordinates (p_1, q_2) be called as n as marked in the figure 2.4. Now, if the coordinates of n is scaled by a non-negative number α, then the resulting coordinate, that is, $(\alpha p_1, \alpha q_2)$, will trace a straight trajectory beginning at origin, passing through the point n, and travelling to infinity as α is varied from 0 to ∞. We thus can span the entire shaded region in figure 2.4, which is the solution to $\mathbf{Zc} \geq 0$, by varying λ_1, λ_2 and α, just by knowing the coordinates of the points on the edges of $\mathbf{Zc} \geq 0$. It's important to note that there are infinite number of points on a given edge. However, we just need one, any one, for our purpose. Now, given that we can construct any vector $\mathbf{c}$ consistent with $\mathbf{Zc} \geq 0$, by the relationship $\mathbf{x}$=$\mathbf{Zc}$, means that we can construct any force network that the three disk system might admit. So finding points on the edges of $\mathbf{Zc} \geq 0$ suffices our purpose.

Now, the method that we use to pick points on the edges[8] of $\mathbf{Zc} \geq 0$ is based on a method called the *Simplex Method* [11] which can only identify vertices of a convex region. The region defined by $\mathbf{Zc} \geq 0$ has a trivial vertex at origin, that is $\mathbf{c{=}0}$, which, by the relationship $\mathbf{x{=}Zc}$, corresponds to a force network with all contact forces equal to zero. In order to pick points from the edges of $\mathbf{Zc} \geq 0$ we must must first create them as vertices so that the Simplex Method could identify them. This makes bounding $\mathbf{Zc} \geq 0$ important. The bounding of $\mathbf{Zc} \geq 0$ is achieved by removing the part of $\mathbf{Zc} \geq 0$ that extends to infinity[9] that we summarise through the following steps and then take a 2-dimensional geometric example to explain the idea.

Consider the $\mathbf{c}$-space looked at from the point of view of a cartesian coordinate system. Let a vector in $\mathbf{c}$-space be represented by n-components $c_1, .., c_n$. Also in the same vector space, let a unit vector $\hat{e}_i$ represent the i_{th} coordinate direction. This implies that all the components of $\hat{e}_i$ are zero except the i_{th} component which is equal to 1. Then following steps bound the open polyhedra $\mathbf{Zc} \geq 0$:

2.1.5.1 Approach I of bounding $\mathbf{Zc} \geq 0$

(i) Construct a linear function of the form ${\hat{e}_i}^T\mathbf{c}$. We will call this function an objective function. n distinct objective functions of the form ${\hat{e}_i}^T\mathbf{c}$ exist. Subject to the constraint $\mathbf{Zc} \geq 0$ perform minimisation and maximisation of ${\hat{e}_i}^T\mathbf{c}$[10] for different coordinate directions $\hat{e}_i$ up until a

[8] If vector $\mathbf{c}$ has n-components, say $c_1, .., c_n$, then an edge would be defined as an intersection of the bounding hyperplanes of any of the $n-1$ inequalities in $\mathbf{Zc} \geq 0$ (A bounding hyperplane is the boundary of a half-space. A half-space is either of the two parts into which a plane divides an n-dimensional space. The region in which an inequality $a_1c_1 + a_2c_2 + .. + a_nc_n \geq 0$ (where, $a_1, .., a_n$ are known scalar coefficients coming from a row in $\mathbf{Z}$ matrix) holds true is a half-space. Its bounding hyperplane will be $a_1c_1 + a_2c_2 + .. + a_nc_n = 0$.). Of course such an edge might not satisfy all the inequalities in $\mathbf{Zc} \geq 0$. But we are looking for points on those edges only which satisfy all the inequalities in $\mathbf{Zc} \geq 0$. Note again that such edges will always emanate from the origin, see for example figure 2.3, where we would have picked points from the two edges which come out of the origin and act as the boundaries of the shaded region in which $\mathbf{Zc} \geq 0$ holds true.

[9] This is what we mean by bounding the unbounded polyhedra $\mathbf{Zc} \geq 0$.

[10] A minimisation or a maximisation of a linear constraint ${\hat{e}_i}^T\mathbf{c}$ subject to a set of linear inequalities $\mathbf{Zc} \geq 0$ was done using the "LINPROG" command in MATLAB.

point at which we find an $\hat{e_i}$ for which the minimisation or maximisation of the objective function $\hat{e_i}^T\mathbf{c}$ occurs at $\mathbf{c}$=$\mathbf{0}$.

(ii) Let's say for some i_{th} coordinate direction the objective function $\hat{e_i}^T\mathbf{c}$, constrained by $\mathbf{Zc} \geq 0$, is extremised at origin $\mathbf{c}$=$\mathbf{0}$. Let us call such an $\hat{e_i}$ an $\hat{e_*}$[11]. Use this $\hat{e_*}$ unit vector to construct an inequality of the form $\hat{e_*}^T\mathbf{c} \leq d$ such that it satisfies $\mathbf{c}$=$\mathbf{0}$ and its bounding plane, that is, $\hat{e}_*^T\mathbf{c} = d$ has non-zero intersection with $\mathbf{Zc} \geq 0$[12].

(iii) The inequality $\hat{e_*}^T\mathbf{c} \leq d$ constraint when imposed on $\mathbf{Zc} \geq 0$ will then eliminate the part of the region of $\mathbf{Zc} \geq 0$ that extends to infinity and hence bound the polyhedra.

2.1.5.2 Approach II of bounding $\mathbf{Zc} \geq 0$

Approach I may fail due to step (ii) of Approach I not succeeding in finding any $\hat{e_i}$ which can become an $\hat{e_*}$. Conceptually, this may happen only if there exist pairs of vectors, satisfying $\mathbf{Zc} \geq 0$, having angles equal to or greater than 90-degrees between them. While we prove that the angle between any pair of vectors, satisfying $\mathbf{Zc} \geq 0$, cannot be greater than 90-degrees (see section 2.1.5.6), they can be orthogonal to each other. However, in our numerical calculations, we have not encountered any bounded polyhedra the vertices of which are orthogonal to each other and we suspect that the occurence of failure of step (ii) of Approach I is due to numerical inaccuracy that arise when angle between vectors which satisfy $\mathbf{Zc} \geq 0$ is close to 90-degrees.
As a remedy, for bounding $\mathbf{Zc} \geq 0$, we resort to Approach II and continuing with the assumption that a vector $\mathbf{c}$ in $\mathbf{c}$-space has n-components, $c_1, .., c_n$, we take the following steps to bound the polyhedra $\mathbf{Zc} \geq 0$:

(i) Construct inequality constraints of the form $-\beta \leq c_i \leq \beta$ for all i,

[11] In worst case, finding such an $\hat{e_*}$ will take n minimisation and n maximisation steps, where, n is the dimensionality of the $\mathbf{c}$-space.

[12] Whether the inequality $\hat{e_*}^T\mathbf{c} \leq d$ satisfies $\mathbf{c}$=$\mathbf{0}$ and its bounding plane $\hat{e_*}^T\mathbf{c} = d$ has a non-zero intersection with $\mathbf{Zc} \geq 0$ will depend upon the sign of the constant d. We demonstrate steps to find the correct sign of such a constant d as we proceed ahead in this section and explain by solving a 2-dimensional problem.

where $\beta > 0$. Let these system of $2n$ inequalities be represented by $-\beta \leq \mathbf{c} \leq \beta$.

(ii) Choose *any* particular $\hat{e_i}$. Minimise ${\hat{e_i}}^T\mathbf{c}$ subject to $\mathbf{Zc} \geq 0$ and $-\beta \leq \mathbf{c} \leq \beta$. If the vector $\mathbf{c}$ that minimises ${\hat{e_i}}^T\mathbf{c}$, constrained by $\mathbf{Zc} \geq 0$ and $-\beta \leq \mathbf{c} \leq \beta$, is a non-zero vector then such a vector $\mathbf{c}$ will be called as $\mathbf{c}_*$. If the vector $\mathbf{c}$ that minimises ${\hat{e_i}}^T\mathbf{c}$, constrained by $\mathbf{Zc} \geq 0$ and $-\beta \leq \mathbf{c} \leq \beta$, is a zero vector then the $\hat{e_i}$ will be called a $\mathbf{c}_*$.

(iii) Use the $\mathbf{c}_*$ to construct an inequality of the form $\mathbf{c}_*^T\mathbf{c} \leq d$ such that it satisfies $\mathbf{c}$=$\mathbf{0}$ and its bounding plane, that is, $\mathbf{c}_*^T\mathbf{c} = d$ has non-zero intersection with $\mathbf{Zc} \geq 0$.

(iv) The inequality $\mathbf{c}_*^T\mathbf{c} \leq d$ constraint when imposed on $\mathbf{Zc} \geq 0$ will then eliminate the part of the region of $\mathbf{Zc} \geq 0$ that extends to infinity and hence bound the polyhedra.

2.1.5.3 Illustrating Approach I

Let us now take a 2-dimensional example to see what the steps of Approach I do to bound the $\mathbf{Zc} \geq 0$.

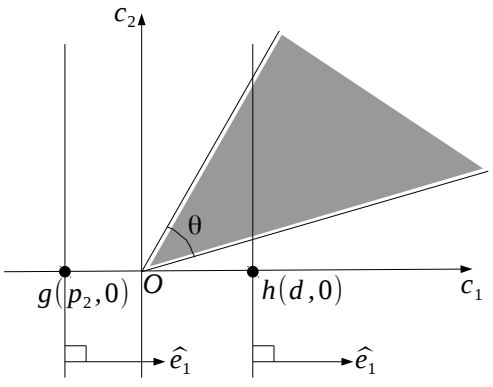

Figure 2.5: Bounding $\mathbf{Zc} \geq 0$

Let us assume that the shaded region in figure 2.5 consists of all the **c** vectors which satisfy $\mathbf{Zc} \geq 0$. We have again assumed for simplicity that the **c**-space is 2-dimensional. Now, in figure 2.5, $\hat{e_1} = (1, 0)$ is a unit vector pointing along the c_1 coordinate axis. Therefore, in figure 2.5, equation of a straight line, with its normal pointing along c_1 coordinate axis, say passing through a point g with coordinates $(p_2, 0)$, is $\hat{e_1}^T\mathbf{c}$=p_2 or simply $c_1 = p_2$ (a **c** vector has two components and therefore **c**= (c_1, c_2)).

Note that $\hat{e_1}$ is a direction in which the value p_2 of the function $\hat{e_1}^T\mathbf{c}$ increases. Now, consider the minimization of $\hat{e_1}^T\mathbf{c}$ constrained by $\mathbf{Zc} \geq 0$. This minimization will occur at a point **c** in the region $\mathbf{Zc} \geq 0$ where the value p_2 of the function $\hat{e_1}^T\mathbf{c}$ is a minimum. By the geometry of $\mathbf{Zc} \geq 0$ in figure 2.5 it is clear that the minimization will happen at **c=0** (note: **c=0** satisfies $\mathbf{Zc} \geq 0$) because beyond that the value of the function $\hat{e_1}^T\mathbf{c}$ will start increasing[13]. Such a coordinate direction $\hat{e_i}$ for which $\hat{e_i}^T\mathbf{c}$, when subjected to $\mathbf{Zc} \geq 0$, gets minimsed at **c=0** will be called an $\hat{e_*}$[14] (Note for figure 2.5 this $\hat{e_*}$ is $\hat{e_1}$.). Now, consider the equation $\hat{e_*}^T\mathbf{c}$=0, that is, c_1=0. This equation intersects $\mathbf{Zc} \geq 0$ in one and one point only which is **c=0**. This is very useful information because we can always generate, using an $\hat{e_*}$, a line (or an $n-1$ dimensional plane when the **c**-space is n-dimensional) of the form $\hat{e_*}^T\mathbf{c}$=d, shown as a line with normal vector $\hat{e_1}$ (note $\hat{e_1} = \hat{e_*}$ in figure 2.5) passing through a point h with coordinates $(d, 0)$ in figure 2.5, that is $c_1 = d$, such that the line intersects *all* the *edges* of $\mathbf{Zc} \geq 0$ if the sign of the constant d is chosen correctly. To find such a line we will have to find the correct sign of d in $\hat{e_*}^T\mathbf{c}$=d (we shall see later that such a line helps us bound the polyhedra $\mathbf{Zc} \geq 0$). The absolute value of d is allowed to be anything because that will only determine where on the edges of $\mathbf{Zc} \geq 0$ the line intersects. And since we are interested in picking any point on an edge of $\mathbf{Zc} \geq 0$ the absolute value of d doesn't matter as long as it is non-zero.

[13]Note that the orientation of the polyhedra $\mathbf{Zc} \geq 0$ is not known a priori and finding the fact that $\hat{e_1}$ can serve as a $\hat{e_*}$ in worst case could have taken 2 minimisations and 2 maximisations in figure 2.5.

[14]Such an $\hat{e_*}$ need not be unique and in fact for the particular geometry of $\mathbf{Zc} \geq 0$ considered in figure 2.5 both $\hat{e_1}$ and $\hat{e_2}$ could have served as an $\hat{e_*}$. For our purpose we just need one, anyone. We will make further comments if such an $\hat{e_*}$ can always be found and if its existence depends upon the geometry of $\mathbf{Zc} \geq 0$ in **c**-space.

Before we proceed to find the correct sign of such a constant d we make an important point. For any given d, $\hat{e_*}^T\mathbf{c}{=}d$ will either intersect all the edges of $\mathbf{Zc} \geq 0$ or will have no intersection with $\mathbf{Zc} \geq 0$ at all. This is true because of two reasons:

(a) The region satisfying $\mathbf{Zc} \geq 0$ lies entirely in one half or the other that a hyperplane $\hat{e_*}^T\mathbf{c}{=}0$ divides a $\mathbf{c}$-space into. To see this, note that $\hat{e_*}^T\mathbf{c}{=}0$ and $\mathbf{Zc} \geq 0$ intersect at one point, that is, $\mathbf{c{=}0}$, only.

(b) $\hat{e_*}^T\mathbf{c}{=}0$ and $\hat{e_*}^T\mathbf{c}{=}d$ are parallel to each other and a *non-zero* distance d apart in the $\mathbf{c}$-space.

As an example, consider figure 2.6, where the line $\hat{e_*}^T\mathbf{c}{=}0$ divides the 2-dimensional $\mathbf{c}$-space into two halves, one shaded in green and the other shaded in yellow, and intersects $\mathbf{Zc} \geq 0$ ($\mathbf{Zc} \geq 0$ shaded in grey) at $\mathbf{c{=}0}$. The entire $\mathbf{Zc} \geq 0$ region then as a result is bound to be contained either in the green or the yellow region and for the specific geometry of $\mathbf{Zc} \geq 0$ that we have considered it will lie in the yellow region.

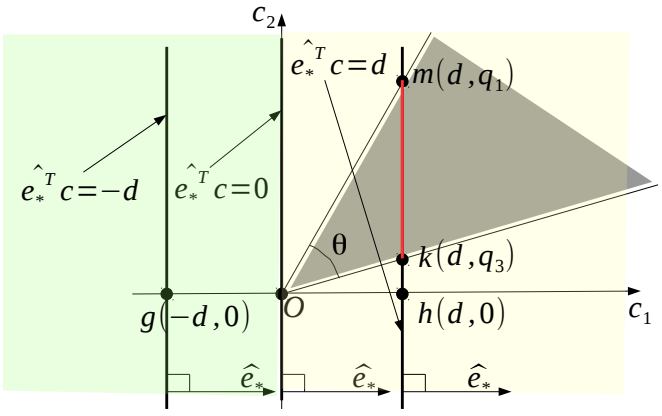

Figure 2.6: Bounding $\mathbf{Zc} \geq 0$ and intersection of $\hat{e_*}^T\mathbf{c}{=}d$ and $\mathbf{Zc} \geq 0$

Now if the line $\hat{e_*}^T\mathbf{c}{=}0$ is pushed into the yellow region by changing zero on the right hand side of the equation to a positive number d but with the

same unit normal vector $\hat{e_*}$ then it is visually suggestive that it is bound to intersect all the edges of $\mathbf{Zc} \geq 0$. This line is shown as $\hat{e_*}^T\mathbf{c}{=}d$ passing through a point h with coordinates $(d, 0)$ in the figure 2.6. Also shown is a line in red joining the points k and m, with coordinates (d, q_3) and (d, q_1) respectively, which is the intersection of $\hat{e_*}^T\mathbf{c}{=}d$ and $\mathbf{Zc} \geq 0$ which will be our next topic of discussion.

Since the geometry of $\mathbf{Zc} \geq 0$ is not known a priori we now therefore proceed to devise a method to find what sign on d ensures intersection of $\hat{e_*}^T\mathbf{c}{=}d$ with $\mathbf{Zc} \geq 0$. Choose any finite number d. Minimise the function $\hat{e_*}^T\mathbf{c}$ subject to $\mathbf{Zc} \geq 0$ and $\hat{e_*}^T\mathbf{c}{=}d$. Then following steps will guide to ensure whether $\hat{e_*}^T\mathbf{c}{=}d$ intersects $\mathbf{Zc} \geq 0$ or $\hat{e_*}^T\mathbf{c}{=}{-}d$ intersects $\mathbf{Zc} \geq 0$:

1. If a minimum value of $\hat{e_*}^T\mathbf{c}$ doesn't exist then this would mean that the regions $\mathbf{Zc} \geq 0$ and $\hat{e_*}^T\mathbf{c}{=}d$ have no point in common and hence $\hat{e_*}^T\mathbf{c}{=}d$ doesn't intersect $\mathbf{Zc} \geq 0$. In which case the equation $\hat{e_*}^T\mathbf{c}{=}{-}d$ will intersect $\mathbf{Zc} \geq 0$.

2. If a minimum value of $\hat{e_*}^T\mathbf{c}$ exists then $\hat{e_*}^T\mathbf{c}{=}d$ intersects $\mathbf{Zc} \geq 0$.

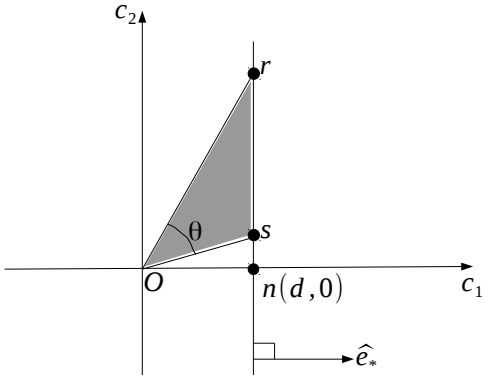

Figure 2.7: Bounded $\mathbf{Zc} \geq 0$ by imposing $\hat{e_*}^T\mathbf{c} \leq d$ on $\mathbf{Zc} \geq 0$

The reason why this method will successfully determine if $\hat{e_*}^T\mathbf{c}{=}d$ intersects $\mathbf{Zc} \geq 0$ is because an intersection of $\mathbf{Zc} \geq 0$ and $\hat{e_*}^T\mathbf{c}{=}d$ is a constraint

on minimisation of $\hat{e_*}^T\mathbf{c}$. And therefore if such an intersection doesn't exist then the minimisation will fail because the constraints have no point in common. Let's say $\hat{e_*}^T\mathbf{c}{=}d$ intersects $\mathbf{Zc} \geq 0$. The final operation is then to convert the equality in $\hat{e_*}^T\mathbf{c}{=}d$ to an inequality of the form $\hat{e_*}^T\mathbf{c} \leq d$ or $\hat{e_*}^T\mathbf{c} \geq d$ whichever satisfies $\mathbf{c{=}0}$. Let's say $\hat{e_*}^T\mathbf{c} \leq d$ satisfies $\mathbf{c{=}0}$. Then the constraint $\hat{e_*}^T\mathbf{c} \leq d$ when imposed on $\mathbf{Zc} \geq 0$ bounds the polyhedra. The bounded polytope is shown in figure 2.7.

2.1.5.4 Illustrating Approach II

Approach II generates a point $\mathbf{c}_*$, satisfying $\mathbf{Zc} \geq 0$, lying either on the boundary or the interior of $\mathbf{Zc} \geq 0$. This point is then used to construct a linear function of the form $\mathbf{c}_*^T\mathbf{c} = 0$ intersecting $\mathbf{Zc} \geq 0$ at one point , that is, $\mathbf{c{=}0}$ only.

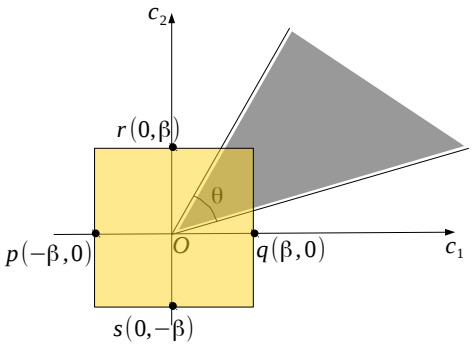

Figure 2.8: Intersection of $-\beta \leq \mathbf{c} \leq \beta$ and $\mathbf{Zc} \geq 0$

Such kind of intersection has been observed to be true for all the systems that we have considered in this thesis. This property then becomes a sufficient condition for an inequality of the form $\mathbf{c}_*^T\mathbf{c} \leq d$ to bound the polyhedra $\mathbf{Zc} \geq 0$. As before, the sign of d is chosen such that $\mathbf{c}_*^T\mathbf{c} \leq d$ satisfies $\mathbf{c{=}0}$ and the bounding plane of $\mathbf{c}_*^T\mathbf{c} \leq d$, that is, $\mathbf{c}_*^T\mathbf{c} = d$ has non-zero intersection with $\mathbf{Zc} \geq 0$.

We will now illustrate how such a $\mathbf{c}_*$ is generated. In figure 2.8 the region shaded in yellow corresponds to $-\beta \leq \mathbf{c} \leq \beta$ and the region shaded in grey corresponds to $\mathbf{Zc} \geq 0$. If a vector $\mathbf{c}$ has to satisfy both $-\beta \leq \mathbf{c} \leq \beta$ and $\mathbf{Zc} \geq 0$ then it must lie in the intersection of the grey and the yellow region. We show intersection $-\beta \leq \mathbf{c} \leq \beta$ and $\mathbf{Zc} \geq 0$ in figure 2.9:

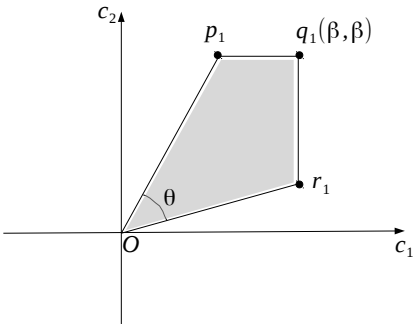

Figure 2.9: Intersection of $-\beta \leq \mathbf{c} \leq \beta$ and $\mathbf{Zc} \geq 0$

The points p_1 and r_1 lie on the boundary of $\mathbf{Zc} \geq 0$ and q_1 lies in the interior of $\mathbf{Zc} \geq 0$. A minimisation of $\hat{e_i}^T\mathbf{c}$, for any particular $\hat{e_i}$, subject to $\mathbf{Zc} \geq 0$ and $-\beta \leq \mathbf{c} \leq \beta$ will generate one of the four points: $\mathbf{c{=}0}$, p_1, q_1 or r_1 depending upon which $\hat{e_i}$ is chosen and the geometry of $\mathbf{Zc} \geq 0$. For example, if $\hat{e_i} = (1, 0)$ is chosen then $\hat{e_i}^T\mathbf{c}$ equals c_1, and then for the particular geometry in figure 2.9, a minimsation of c_1, subject to $-\beta \leq \mathbf{c} \leq \beta$ and $\mathbf{Zc} \geq 0$ will yield the point $\mathbf{c{=}0}$ (that is $c_1 = 0$ and $c_2 = 0$). This then according to step (ii) of Approach II implies that $\mathbf{c}_* = \hat{e_i}$. The inequality $\mathbf{c}_*^T\mathbf{c} \leq d$, when imposed on $\mathbf{Zc} \geq 0$ bounds $\mathbf{Zc} \geq 0$ as shown in figure 2.10. As discussed while illustrating Approach I, the sign of d is chosen such that $\mathbf{c}_*^T\mathbf{c} \leq d$ satisfies $\mathbf{c{=}0}$ and the bounding plane of $\mathbf{c}_*^T\mathbf{c} \leq d$, that is, $\mathbf{c}_*^T\mathbf{c} = d$ has non-zero intersection with $\mathbf{Zc} \geq 0$. In figure 2.10, the bounded polyhedra which is the intersection of $\mathbf{c}_*^T\mathbf{c} \leq d$ and $\mathbf{Zc} \geq 0$ is shown as shaded region in grey. The bounding plane of $\mathbf{c}_*^T\mathbf{c} \leq d$, that is, $\mathbf{c}_*^T\mathbf{c} = d$ intersects the edges of $\mathbf{Zc} \geq 0$ to give the points p_2 and q_2 also marked in the figure 2.10.

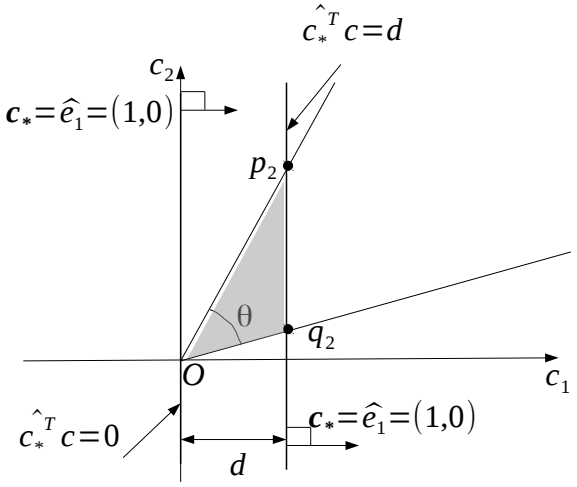

Figure 2.10: Bounded $\mathbf{Zc} \geq 0$ shaded in grey

Before we close our discussion on bounding of $\mathbf{Zc} \geq 0$ we would like to make a final remark that other alternatives of bounding $\mathbf{Zc} \geq 0$ can be invented by choosing an arbitrary point within $\mathbf{Zc} \geq 0$, differently from Approach II, to find the hyperplane with the surface normal vector pointing along the position vector of the chosen point. This hyperplane will bound $\mathbf{Zc} \geq 0$ if no pair of vectors satisfying $\mathbf{Zc} \geq 0$ have an angle equal to or greater than 90-degrees between them. This completes our discussion on bounding the polyhedra $\mathbf{Zc} \geq 0$. The goal was to create a point on each edge of $\mathbf{Zc} \geq 0$. A knowledge of the coordinates of all such points helps us span the entire set of $\mathbf{c}$ vectors which satisfy $\mathbf{Zc} \geq 0$.

2.1.5.5 Comments and comparison of Approach I and Approach II of bounding $\mathbf{Zc} \geq 0$

- Approach I will succeed in bounding $\mathbf{Zc} \geq 0$ if no pair of vectors $\mathbf{c}_1$ and $\mathbf{c}_2$, satisfying $\mathbf{Zc} \geq 0$, have angle greater than or equal to 90-degrees between them. As has been seen, Approach I fails in bounding $\mathbf{Zc} \geq 0$ for some systems considered in this thesis. The reason is not yet

clear to us and would require further investigations and forms a part of future directions of this project.

- Similar to Approach I, Approach II, will succeed in bounding $\mathbf{Zc} \geq 0$ if no pair of vectors $\mathbf{c}_1$ and $\mathbf{c}_2$, satisfying $\mathbf{Zc} \geq 0$, have angle greater than or equal to 90-degrees between them. But on the contrary, Approach II has been observed to bound $\mathbf{Zc} \geq 0$ for all the systems that we have considered in the thesis.

- The problem of minimising or maximising a linear function subject to linear constraints is also called linear programming problem (LPP). Approach I requires solving $2n$ LPPs inorder to bound $\mathbf{Zc} \geq 0$ but needs $2n$ less inequality constraints to solve each LPP.

- Approach II solves **one** LPP to bound $\mathbf{Zc} \geq 0$ but requires $2n$ additional inequality constraints.

- The general strategy would be therefore to use Approach II directly to bound $\mathbf{Zc} \geq 0$.

2.1.5.6 Some questions

Here we raise some questions to frame our study and provide answers.

Question 1: What motivates finding the extremal vertices of the bounded polyhedra and how is it related to finding the force network of an under determined static granular system?

Answer: Let's say that the contact geometry of a system of granular particles is given, then, an under-determined system of granular particles has infinite number of force network solutions each satisfying the constraint of mechanical equilibrium. The space of such solutions reduce when extra constraints such as zero-cohesion is imposed at each contact between particles. The reduced space of solutions still contain an infinite number of force network solutions. In this thesis this reduced space is denoted by $\mathbf{Zc} \geq 0$, where, through the relationship $\mathbf{x}=\mathbf{Zc}$, different $\mathbf{c}$ vectors yield different force network solutions. The geometry of $\mathbf{Zc} \geq 0$ is an unbounded polyhedra extending to infinite distance from $\mathbf{c}=\mathbf{0}$. We bound this polyhedra by imposing

an inequality of the form $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$). This results into a set of vertices and we identify all of them by finding their co-ordinates. As we have shown before, knowing these extremal vertices is sufficient to construct any $\mathbf{c}$ vector consistent with $\mathbf{Zc} \geq 0$.

We now point out that the value of d in $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) is arbitrary and, since $\mathbf{x}$=$\mathbf{Zc}$, determines the maximum magnitude of contact forces existing in the granular system. Hence, d sets the force scale in the granular system. Therefore, in order to determine the force network of an under-determined system one must generate points, say $\mathbf{c_1}, \mathbf{c_2}, .., \mathbf{c_p}$, which are spread uniformly in the space of intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) and $\mathbf{Zc} \geq 0$ and then take an average of these points[15], which say is denoted by $\mathbf{c}_{avg}$, to generate, by the relationship $\mathbf{x}$=$\mathbf{Zc}_{avg}$, the force network for the under-determined system. This of course has not been demonstrated in this thesis and forms an important part of future directions of this project. The role of the extremal vertices of the intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) and $\mathbf{Zc} \geq 0$ is to ensure that the uniformly generated points $\mathbf{c_1}, \mathbf{c_2}, .., \mathbf{c_p}$ span over all the feasible space of solutions which is the intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) and $\mathbf{Zc} \geq 0$.

Question 2: The success of bounding $\mathbf{Zc} \geq 0$ lies in the success of finding an $\hat{e_*}$ (or $\mathbf{c}_*$). Can an $\hat{e_*}$ (or $\mathbf{c}_*$) always be found?

Answer: A necessary requirement for finding an $\hat{e_*}$ (or $\mathbf{c}_*$) is that for any pair of vectors $\mathbf{c}_1$ and $\mathbf{c}_2$, which satisfy $\mathbf{Zc} \geq 0$, the angle between them must be less than 90-degrees.

We explain the necessity for such a requirement, firstly for Approach I, using a 2-dimensional example. In figure 2.11, the regions shaded in red and grey represent the two different ways in which the region satisfying $\mathbf{Zc} \geq 0$ could be oriented such that the angles θ_1 and θ_2 are less than 90-degrees due to the requirement. Consider the case when $\mathbf{Zc} \geq 0$ is the grey region. Since, the directions in which $\hat{e_1}^T\mathbf{c}$ and $\hat{e_2}^T\mathbf{c}$ increase are $\hat{e_1}$ and $\hat{e_2}$ respectively, therefore, a minimisation of $\hat{e_1}^T\mathbf{c}$ or $\hat{e_2}^T\mathbf{c}$ against $\mathbf{Zc} \geq 0$, would both result

[15]This of course assumes that the corresponding force networks $\mathbf{x_1}, \mathbf{x_2}, .., \mathbf{x_p}$, which are solutions to $\mathbf{Ax}$=$\mathbf{0}$, obtained using the relationship $\mathbf{x}$=$\mathbf{Zc}$, are *equally likely* to exist in the under-determined granular system. The i_{th} component of $\mathbf{c}_{avg}$ will be equal to the average value of the i_{th} component of the vectors $\mathbf{c_1}, \mathbf{c_2}, .., \mathbf{c_p}$.

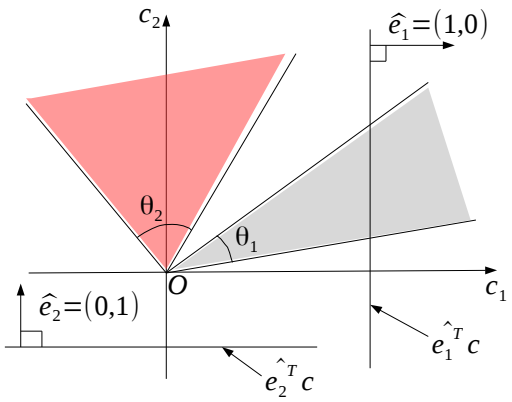

Figure 2.11: Approach I : Constraint on $\mathbf{Zc} \geq 0$. $\theta_1, \theta_2 < 90$-degrees

in the point **c=0**. Therefore, both $\hat{e_1}$ and $\hat{e_2}$ could become an $\hat{e_*}$. Now, consider the case when $\mathbf{Zc} \geq 0$ is the red region. In this case, only the minimisation of $\hat{e_2}^T\mathbf{c}$ against $\mathbf{Zc} \geq 0$ would result in the point **c=0**. A maximisation of $\hat{e_2}^T\mathbf{c}$, or, a minimisation or a maximisation of $\hat{e_1}^T\mathbf{c}$, against $\mathbf{Zc} \geq 0$, would all be unbounded and would not give **c=0**. Hence, only $\hat{e_2}$ could become an $\hat{e_*}$. Any other possible orientation of $\mathbf{Zc} \geq 0$ in 2-dimension, with $\theta < 90$-degrees, would fall in one of the two categories shown in figure 2.11 and hence by making similar arguments as above one can say that an $\hat{e_*}$ can always be found[16]. We will now explain such a requirement in Approach II using a 2-dimensional example. Approach II does one minimisation of any particular $\hat{e_i}^T\mathbf{c}$ (chose any one) against $\mathbf{Zc} \geq 0$ and $-\beta \leq \mathbf{c} \leq \beta$. Two possibilities exist:

(i) The minimisation happens at **c=0**.

[16]One may say that such a requirement need not hold for $\mathbf{Zc} \geq 0$ but we will prove that angle between any pair of vectors $\mathbf{c}_1$ and $\mathbf{c}_2$, which satisfy $\mathbf{Zc} \geq 0$, will be less than or equal to 90-degrees. While we prove that the angle between such pairs of vectors cannot be greater than 90-degrees they can be orthogonal to each other. However, in our numerical calculations, we have not encountered any bounded polyhedra the vertices of which are orthogonal to each other and we suspect that the occurence of failure of being able to find an $\hat{e_*}$ is due to numerical inaccuracy that arise when angle between vectors which satisfy $\mathbf{Zc} \geq 0$ is close to 90-degrees.

(ii) The minimisation happens at one of the points on the boundary or the interior of $\mathbf{Zc} \geq 0$.

If case (i) is true then $\hat{e_i}^T\mathbf{c} = 0$ will intersect $\mathbf{Zc} \geq 0$ in one and one point only, that is, $\mathbf{c{=}0}$. This then implies that $\hat{e_i}{=}\mathbf{c}_*$. This case puts no restriction on the angles between pairs of vectors satisfying $\mathbf{Zc} \geq 0$. The convexity of $\mathbf{Zc} \geq 0$ will ensure that an inequality of the form $\mathbf{c}_*^T\mathbf{c} \leq d$ (for a correct sign of d) bounds $\mathbf{Zc} \geq 0$.

If case (ii) is true then the minimisation will happen at p_1, q_1 or r_1 shown as points with co-ordinates (m_1, β), (β, β) and (β, m_2) respectively.

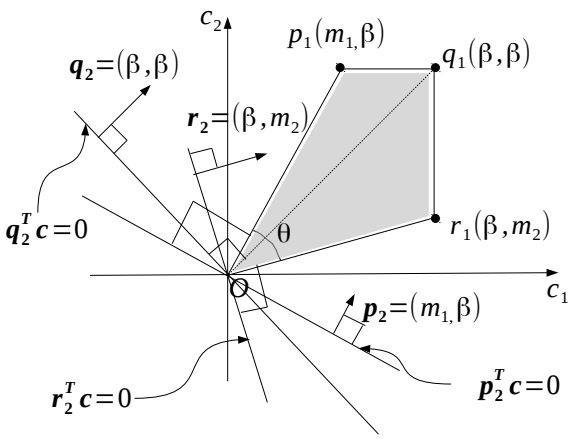

Figure 2.12: Approach II : Constraint on $\mathbf{Zc} \geq 0$, $\theta < 90$-degrees

Say the minimisation in case (ii) happened at point p_1. The co-ordinates of the point p_1 then serve as a vector $\mathbf{p}_2$ in determining the orientation of a line $\mathbf{p}_2^T\mathbf{c} = 0$ such that the line is guaranteed, since $\theta < 90$-degrees, to intersect $\mathbf{Zc} \geq 0$ only at $\mathbf{c{=}0}$. This then gives us $\mathbf{c}_* = \mathbf{p}_2$ which helps bound the $\mathbf{Zc} \geq 0$. Note that if co-ordinates of any other point (q_1 or r_1) were chosen in determining the orientation of such a line then the line were guaranteed, since $\theta < 90$-degrees, to have intersected $\mathbf{Zc} \geq 0$ only at $\mathbf{c{=}0}$.

We state again that Approach II was able to bound every $\mathbf{Zc} \geq 0$ considered in this thesis corresponding to different systems (equi-triangular, square

and rhombic lattice) and sizes. Approach I, however failed in bounding $\mathbf{Zc} \geq 0$ for some systems.

We will now prove that for frictionless granular systems, which obey the constraint of zero cohesion at each contact, the angle between any two non-zero vectors $\mathbf{c}_1$ and $\mathbf{c}_2$, each of which satisfy $\mathbf{Zc} \geq 0$, will be less than or equal to 90-degrees. We don't have a proof for such an angle to be not equal to 90-degrees.

Consider any two vector solutions $\mathbf{x}_1$ and $\mathbf{x}_2$ of the force balance equation $\mathbf{Ax=0}$ (see equation 2.2 for example) and the constraint of zero cohesion. The constraint of zero cohesion at each contact implies that all the components of the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are non-negative (see section 2.1.2 for example). Since the components of $\mathbf{x}_1$ and $\mathbf{x}_2$ are non-negative, the dot product of the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ will always be greater than or equal to zero. This implies that the angle between the two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ is less than or equal to 90-degrees. Let us now for such an $\mathbf{x}_1$ and $\mathbf{x}_2$, which obey $\mathbf{Ax=0}$ and componentwise non-negativity constraint on $\mathbf{x}$, use the relationship $\mathbf{x}_1=\mathbf{Zc}_1$ and $\mathbf{x}_2=\mathbf{Zc}_2$ to obtain $\mathbf{c}_1$ and $\mathbf{c}_2$ (note that such $\mathbf{c}_1$ and $\mathbf{c}_2$ will satisfy $\mathbf{Zc} \geq 0$). We prove in the next section that the angle between $\mathbf{x}_1$ and $\mathbf{x}_2$ is same as the angle between $\mathbf{c}_1$ and $\mathbf{c}_2$. This then implies that the angle between $\mathbf{c}_1$ and $\mathbf{c}_2$ will be less than or equal to 90-degrees.

Question 3: Can an $\hat{e_*}$ (or $\mathbf{c}_*$) always be found for frictional systems?

Answer: It has been shown empirically for 3-dimensional frictional monodisperse system of spheres that the angle between any two $\mathbf{x}$ vectors[17], when constrained by zero cohesion at each contact is less than $90 - degrees$ [9]. Hence the method must work for such systems as well.

Question 4: What is the computational cost of finding such an $\hat{e_*}$ (or $\mathbf{c}_*$)?

Answer: If the $\mathbf{c}$-space is n dimensional then in the worst case finding an $\hat{e_*}$ would require n minimisations and n maximisations of linear functions subject to linear constraint $\mathbf{Zc} \geq 0$. On the other hand, finding a $\mathbf{c}_*$ requires *one* minimisation only. The problem of minimising or maximising

[17] An $\mathbf{x}$ vector, which satisfies the force and torque balance equation $\mathbf{Ax=0}$, would contain normal and radial components of contact forces for a frictional granular system and the constraint of zero cohesion would put the non-negativity constraint on the normal components of contact forces contained in the vector $\mathbf{x}$.

a linear function subject to linear constraints is also called linear programming problem(LPP). We solve such LPPs via the Simplex Method using the "LINPROG" command in MATLAB. The Simplex Method in general tends to run in time linear to the number of constraints of the problem but in certain worst cases it tends to run in polynomial time algorithm. Examples have also been constructed for which the worst case performance, measured through the number of iterations required, was known to be exponential [17]. Another popular method which is used to solve LPPs is the Interior Point Method but they haven't been used or explored in this thesis.

2.1.5.7 A proof

We will now prove that the angle between $\mathbf{x}_1$ and $\mathbf{x}_2$ is same as the angle between $\mathbf{c}_1$ and $\mathbf{c}_2$, that is:

$$\frac{\mathbf{x_1}^T\mathbf{x_2}}{|\mathbf{x_1}|\,|\mathbf{x_2}|} = \frac{\mathbf{c_1}^T\mathbf{c_2}}{|\mathbf{c_1}|\,|\mathbf{c_2}|} \tag{2.6}$$

where, $\mathbf{x_1}$, $\mathbf{x_2}$ are any two solutions to the force balance equation $\mathbf{Ax}=\mathbf{0}$ and $\mathbf{x_1} = \mathbf{Zc_1}$ and $\mathbf{x_2} = \mathbf{Zc_2}$.
Proof:

$$\frac{\mathbf{x_1}^T\mathbf{x_2}}{|\mathbf{x_1}|\,|\mathbf{x_2}|} = \frac{(\mathbf{Zc_1})^T\mathbf{Zc_2}}{|\mathbf{x_1}|\,|\mathbf{x_2}|} \tag{2.7}$$

$$= \frac{\mathbf{c_1}^T(\mathbf{Z}^T\mathbf{Z})\mathbf{c_2}}{|\mathbf{x_1}|\,|\mathbf{x_2}|} \tag{2.8}$$

$$= k\frac{\mathbf{c_1}^T\mathbf{c_2}}{|\mathbf{x_1}|\,|\mathbf{x_2}|} \tag{2.9}$$

where we have used the following:

$$\mathbf{x} = \mathbf{Zc} \tag{2.10}$$

$$\mathbf{Z}^T\mathbf{Z} = k\mathbf{I} \text{ (\textit{since } } \mathbf{Z} \text{ \textit{is an orthogonal matrix}}) \tag{2.11}$$

where, we chose to keep k as some non-zero positive scalar. Equation 2.11 just

means that the column vectors in $\boldsymbol{Z}$ are obtained by scaling an *orthonormal basis*[18] by a positive factor k (can be 1).

Since,

$$\mathbf{x_1}^T\mathbf{x_1} = (\mathbf{Zc_1})^T\mathbf{Zc_1} = \mathbf{c_1}^T(\mathbf{Z}^T\mathbf{Z})\mathbf{c_1}$$

$$Therefore,\ |\mathbf{x_1}|^2 = k|\mathbf{c_1}|^2 \tag{2.12}$$

$$Similarly,\ |\mathbf{x_2}|^2 = k|\mathbf{c_2}|^2 \tag{2.13}$$

Using equation 2.12 and 2.13 (relates the length of vector $\boldsymbol{x}$ and vector $\boldsymbol{c}$) in equation 2.9 we get:

$$\frac{\mathbf{x_1}^T\mathbf{x_2}}{|\mathbf{x_1}|\,|\mathbf{x_2}|} = \frac{\mathbf{c_1}^T\mathbf{c_2}}{|\mathbf{c_1}|\,|\mathbf{c_2}|}$$

Thus proving equation 2.6.

We end the three disk problem by finding the vertices of the bounded polyhedra[19]. The method of finding the vertices of a bounded polyhedra will be discussed in section 2.4.

We list the coordinates of the vertices of the bounded polyhedra, excluding the trivial vertex at $\mathbf{c}=\mathbf{0}$, as row vectors of the $\mathbf{V}$ matrix, for a choice of $d = 100$:

$$\mathbf{V} = \begin{bmatrix} 60.5475 & 56.3607 & -16.9082 \\ -243.3962 & 490.5660 & -147.1698 \\ 4.6595 & 42.6523 & 52.6882 \\ 4.2039 & 42.7746 & 53.0215 \\ 4.0486 & 42.9150 & 53.0364 \\ -224.2424 & 442.4242 & -118.1818 \\ -249.8911 & 497.3856 & -147.4946 \end{bmatrix}$$

[18] An orthonormal basis consists of vectors which are orthogonal to each other and are of unit length.

[19] We use a freely available code developed and benchmarked by authors mentioned in reference [13] in **c**-space. This code finds all the vertices of a convex region in an n-dimensional space.

The row vectors in **V** represent the coordinates of the extremal vertices of the bounded polyhedra in **c**-space. Using these vectors one can span all the **c** vectors that satisfy $\mathbf{Zc} \geq 0$ and by using **x=Zc** construct all the force networks that the three-disk system might admit (the method for doing so has already been discussed before in this chapter). Although we didn't have a mathematical proof but here it should be noted importantly that the maximum of the angles between a pair of vertices listed in **V** is close to 71-degrees which serves as an evidence that the angle between no pair of solutions to $\mathbf{Zc} \geq 0$ is greater than or equal to 90-degrees.

Using **x=Zc** (A row vector of matrix **V** is nothing but a **c** vector satisfying $\mathbf{Zc} \geq 0$), we obtain the following force balanced solution for the three-disk system in figure 2.1:

$$\mathbf{F} = \begin{bmatrix} 0 & 26964 & 0 & 12 & 22 & 24164 & 27410 \\ 4043 & 526 & 29 & 5 & 0 & 0 & 363 \\ 0 & 0 & 3274 & 3293 & 3296 & 727 & 86 \\ 2354 & 31272 & 0 & 0 & 8 & 27745 & 31688 \\ 64 & 31370 & 1891 & 1916 & 1928 & 28524 & 31936 \\ 2301 & 0 & 3749 & 3756 & 3757 & 561 & 0 \\ 4648 & 392 & 1901 & 1883 & 1879 & 224 & 250 \\ 4710 & 16347 & 21 & 0 & 0 & 14097 & 16416 \\ 1 & 15415 & 3771 & 3799 & 3808 & 14652 & 15769 \end{bmatrix}$$

The column vectors of the matrix **F** are the solutions to **Ax=0** which also satisfy the non-negativity constraint on put on the components of vector **x**.

2.1.6 Summary

We now list and summarise the steps that we followed to solve the 2-dimensional 3-disk frictionless problem:

(i) Construct the contact matrix **A** by writing down the force balance equation for each particle in the system. The resulting system of linear equations will be of the form **Ax=0**. Let **A** be of the size $m \times n$ and rank equal to r. The number of rows, m, in matrix **A** equals 2N, where

N is the number of particles in the system. The number of columns, n, in matrix $\mathbf{A}$ equals L, where L is the total number of contacts in the system. The force at each contact must satisfy the non-negativity criteria as we are looking at systems with zero cohesion, also, such forces point along the radial direction of a particle due to absence of friction. The problem is to solve $\mathbf{Ax=0}$ such that all the components of $\mathbf{x}$ are positive[20].

(ii) This problem is converted equivalently to another form by expressing the $\mathbf{x}$ vector as a linear combination of the basis vectors of the nullspace of matrix $\mathbf{Ax=0}$ and then imposing the non negativity constraint on the components of the vector $\mathbf{x}$. This results into a system of linear inequalities of the form $\mathbf{Zc} \geq 0$ where $\mathbf{Z}$ is a matrix of size n by $n - r$ and $\mathbf{c}$ is an unknown vector whose components are the coefficients used to expand the vector $\mathbf{x}$ in the basis of nullspace of $\mathbf{A}$.

(iii) The inequality $\mathbf{Zc} \geq 0$ describes an open convex polyhedra emanating from the origin in an $n-r$ dimensional space that we call a $\mathbf{c}$-space. The problem is now to pick a point from every edge of this open polyhedra as a representation of all the solutions $\mathbf{c}$ which satisfy $\mathbf{Zc} \geq 0$. We solve this problem by bounding the polyhedra $\mathbf{Zc} \geq 0$ and then identifiying its extremal vertices. Rescaling the convex combinations of the extremal vertices of the bounded $\mathbf{Zc} \geq 0$ by non-zero scalars can generate any solution to $\mathbf{Zc} \geq 0$ and hence by the relationship $\mathbf{x} = \mathbf{Zc}$ any possible force network.

2.2 The Simplex Method

In this section we describe and illustrate the Simplex Method using a text-book reference [6]. In this thesis this method has been used to determine the plane, identified by its surface normal vector $\hat{e_*}$, that bounds the polyhedra

[20]We write the force balance equation assuming that the contact force acting on a particle is directed towards itself. Hence, imposing the non-negativity criteria on components of $\mathbf{x}$ ensures that the force at each contact is non-cohesive in nature.

$\mathbf{Zc} \geq 0$. This method is also central to finding vertices of the polyhedra $\mathbf{Zc} \geq 0$ after it has been bounded. The computational efficiency of Simplex Method has been mentioned in section 2.1.5. We will now begin by taking a simple example problem and solve it step by step. We then summarise the steps and explain how it is used to find vertices of bounded polyhedra. We have used the code which finds all the vertices of a bounded polyhedra which is freely available due to [13]. We emphasise that in this section the notations and their meaning have no connection with the same notation if/when used elsewhere in this thesis.

2.2.1 Inroduction

An inequality divides an n-dimensional space into two **halfspaces**, one where the inequality is satisfied and the other where it is not. A typical example is $x + 2y \geq 4$. The boundary between the two halfspaces is $x + 2y = 4$ (a line) where also the inequality is satisfied. The picture would be almost the same in three dimensions except that the boundary becomes a plane. In n-dimensions we still call the $n-1$ dimensional boundary a 'plane'. In addition to the inequalities of this kind there is another constraint fundamental to the problems (these problems are called **Linear Programming Problems**) that Simplex Method solves: x and y may be required to be non-negative. This requirement itself is a pair of inequalities $x \geq 0$, $y \geq 0$. The important step is to impose all the three inequalities $x + 2y \geq 4$, $x \geq 0$, $y \geq 0$ at once. They combine to give the colored region in the figure 2.13.

The colored region is an intersection of the three halfspaces. It is no longer a halfspace. It is called a **feasible set**. A feasible set is composed of the solutions to a family of linear inequalities. A system of m-inequalities like $\mathbf{Ax} \geq \mathbf{b}$ describes the intersection of m halfspaces. If we also require that every component of $\mathbf{x}$ is non-negative (written as $\mathbf{x} \geq 0$), this adds n more halfspaces. The more constraints we impose the smaller the feasible set. The feasible set may be bounded, unbounded or empty. In linear programming, we are interested in a particular point in the feasible set that **maximizes** or **minimizes** a certain "**cost function**". To the example $x + 2y \geq 4$, $x \geq 0$,

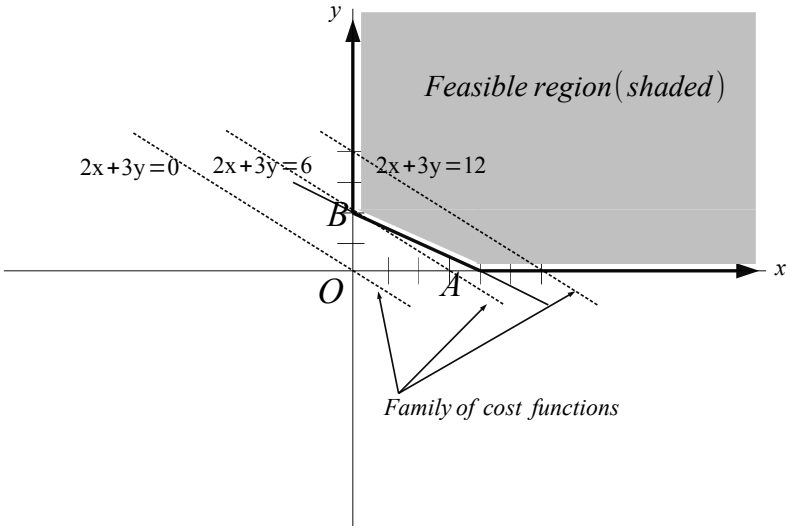

Figure 2.13: Minimization of linear cost function against linear inequalities

$y \geq 0$, we add the cost function (or the objective function) $2x + 3y$ (linear cost function). Then the problem in linear programming is to find the point x, y that **lies in the feasible set and minimizes the cost**. The problem is illustrated by the geometry of the figure 2.13. The family of costs $2x + 3y$ gives a family of parallel lines and we have to find the minimum cost i.e. the first line to intersect the feasible set. In the example figure 2.13 that intersection occurs at point B where $x^* = 0$ and $y^* = 2$. The minimum cost is $2x^* + 3y^* = 6$. The vector $(0, 2)$ is called *feasible* because it lies in the feasible set, it is *optimal* because it minimizes the cost function and the minimum cost 6 is the value of the program. We denote optimal vectors by an asterisk. We see that the **optimal vector occurs at the corner of the feasible set**. This is guaranteed by the geometry, because the lines that give the cost function (or the planes when we get to more unknowns) are moved steadily up until they intersect the feasible set. The first contact must occur along its boundary. The Simplex Method will stay on the boundary, going

from one corner of the feasible set to the next until it finds the corner with lowest cost.

Note: With a different cost function the intersection might not just be a single point. If the cost happened to be $x+2y$ the whole edge between B and A would intersect at the same time and there would be an infinity of optimal vectors along that edge. The optimal value is still unique ($x^* + 2y^* = 4$ for all these optimal vectors) and therefore the minimum problem still has a definite answer. On the other hand, the maximum problem with the cost function $x+2y$ would have no solution. On our feasible set this cost function can go arbitrarily high and our max cost is infinite.

Every linear programming problem falls into one of two possible categories:

- The feasible set is empty.
- The cost function has a minimum (or maximum) on the feasible set which could be a finite or an infinite value.

2.2.2 The Simplex method

The Simplex method is a standard method of extremising linear functions which are subject to linear constraints and for a detailed discussion one can look at [6]. We will choose to solve a problem in the following form to explain briefly the steps involved:

$$Minimize\ \boldsymbol{dy},\ subject\ to\ \boldsymbol{y} \geq \boldsymbol{0},\ \boldsymbol{Gy} \geq \boldsymbol{b}$$

where, $\boldsymbol{G}$ is an $m \times n$ matrix, $\boldsymbol{b}$ is a column vector with m components and $\boldsymbol{d}$ is a *row vector* with n components. The"optimal" vector is the feasible vector of least cost and the cost is $\boldsymbol{dy} = d_1y_1 + .. + d_ny_n$

In this section we will see a mathematical description of making a choice to move along the edge of the feasible set from a given vertex. We start with rewriting $\boldsymbol{Gy} \geq \boldsymbol{b}$ in a different form by introducing *slack variables*. It is

used so that only simple inequalities remain. We write slack variables as:

$$\boldsymbol{w} = \boldsymbol{G}\boldsymbol{y} \; - \; \boldsymbol{b}$$

.

which goes in the matrix form as:

$$\begin{bmatrix} \boldsymbol{G} & -\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{w} \end{bmatrix} = \boldsymbol{b}$$

Where, $\boldsymbol{I}$ is an identity matrix of the size $m \times m$ and $\boldsymbol{w}$ is $m \times 1$. The feasible set is governed by these m equations and the $n+m$ simple inequalities $\boldsymbol{y} \geq \boldsymbol{0}, \; \boldsymbol{w} \geq \boldsymbol{0}$. We rename the larger matrix $\boldsymbol{A}$ and the longer vector $\boldsymbol{x}$. The original cost vector $\boldsymbol{d}$ extended by adding m more components of all zeros is renamed as $\boldsymbol{c}$. The problem has now become:

$$\textit{Minimize } \boldsymbol{c}\boldsymbol{x} \textit{ subject to } \boldsymbol{x} \geq \boldsymbol{0}, \; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

A corner is a point where n components of the new vector $\boldsymbol{x}$ are zero. In $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ these n components are the $co-basic$ variables and the remaining m components are the *basic* variables. Then setting n co-basic variables to zero the m equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ determine the m basic variables. This solution $\boldsymbol{x}$ is called basic. It will be a feasible corner if its m non-zero components are positive.

In order to move from a vertex to another we exchange one basic variable with a cobasic variable. This cobasic variable is called the *entering variable*(entering the basis) while the basic variable being removed is called the *leaving variable* (leaving the basis). The cost function decides which co-basic variable to bring into the basis. The co-basic variable in the cost function whose coefficient has the most negative value is the entering variable (since this reduces the objective function the most). As the entering component of the vector $\boldsymbol{x}$ increases from zero, other components of $\boldsymbol{x}$ may decrease(in order to maintain $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$). The first component that decreases to zero becomes the leaving variable.

When we start at a corner the basic variables in the vector $\boldsymbol{x}$ in $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$

may be mixed. We renumber them (at the current corner) so that the first m components of $\boldsymbol{x}$ is basic(this amounts to rearranging the columns of $\boldsymbol{A}$). The last n components therefore is co-basic(zero value). The first m columns of $\boldsymbol{A}$ now form a square matrix B (the *basic matrix* for that corner) and the last n columns give an mxn matrix N. Similarly, the cost vector is split into $[c_B\ c_N]$ and the unknown $\boldsymbol{x}$ into $[x_B\ x_N]^T$. At the corner $x_N = 0$ and $\boldsymbol{Ax = b}$ becomes $Bx_B = b$ and determines the basic variables x_B. The cost at this corner is $\boldsymbol{cx} = c_B x_B$. We now construct a large matrix or a tableau:

$$T = \begin{array}{c|c} A & b \\ \hline c & 0 \end{array}$$

The tableau is $m + 1$ by $m + n + 1$. To operate with it we separate it into basic columns first:

$$T = \begin{array}{c|c|c} B & N & b \\ \hline c_B & c_N & 0 \end{array}$$

We do row operations to reach the following:

$$T' = \begin{array}{c|c|c} I & B^{-1}N & B^{-1}b \\ \hline c_B & c_N & 0 \end{array}$$

To finish c_B times the top part is subtracted from the bottom which gives:

$$T'' = \begin{array}{c|c|c} I & B^{-1}N & B^{-1}b \\ \hline 0 & c_N - c_B B^{-1}N & -c_B B^{-1}b \end{array}$$

We now interpret the table properly. To do this we rewrite $\boldsymbol{Ax = b}$ as $Bx_B + Nx_N = \boldsymbol{b}$. This can be written as:

$$x_B + B^{-1}Nx_N = B^{-1}\boldsymbol{b}$$

and the cost $\boldsymbol{cx} = c_B x_B + c_N x_N$ has been turned into:

$$\boldsymbol{cx} = (c_N - c_B B^{-1}N)x_N + c_B B^{-1}\boldsymbol{b}$$

The main point is that every important quantity appears in the tableau. On the far right in T'' are the basic variables $x_B = B^{-1}\boldsymbol{b}$ (the co-basic variables are just $x_N = 0$). The current cost is $\boldsymbol{cx} = c_B B^{-1}\boldsymbol{b}$ which is in the bottom corner with a minus sign. Most important, we can decide if the current corner is optimal by looking at $r = c_N - c_B B^{-1} N$ (r is called the *reduced cost*). If any entry in r is negative the cost can still be reduced. The optimal corner is found when all entries in r are non-negative. For the current corner we go from T to T'' and check the optimality condition put on r. If r has negative components then the current corner is not optimal. We then choose entering and leaving variables (going to an adjacent vertex) to reconstruct T. We repeat this process until for a given corner all entries in r become positive in the tableau T''.

We finally summarize the simplex method in steps:

1. Compute the row vector $\lambda = c_B B^{-1}$ and the reduced costs $r = c_N - \lambda N$
2. If $r \geq 0$ stop: current solution is optimal. Otherwise if r_i is the most negative component, choose the ith column of N to enter the basis. Denote it by u.
3. Compute $v = B^{-1}u$.
4. Calculate the ratios of $B^{-1}\boldsymbol{b}$ to $B^{-1}u$, admitting only the positive components of $B^{-1}u$. If there are no positive components, the minimal cost is $-\infty$. If the smallest ratio occurs at component k, then the kth column of the current B will leave. This is also called the **Minimum Ratio Test**.
5. Update B^{-1} and the solution $x_B = B^{-1}\boldsymbol{b}$. Return to 1.

2.2.3 A solved example

A corner is a vector $\boldsymbol{x} \geq \boldsymbol{0}$ that satisfies the m equations $\boldsymbol{Ax} = \boldsymbol{b}$ with at most m positive components. The other n components are zero. (Those are the free variables. Back substitution gives the m basic variables. All variables must be nonnegative or $\boldsymbol{x}$ is a false corner i.e. not in the feasible

set.) For a neighboring corner, one zero component of $\boldsymbol{x}$ becomes positive and one positive component becomes zero.

The simplex method must decide which component "enters" by becoming positive, and which component "leaves" by becoming zero. That exchange is chosen so as to lower the total cost. This is one step of the simplex method, moving toward $\boldsymbol{x}^*$.

Here is the overall plan. Look at each zero component at the current corner. If it changes from 0 to 1, the other nonzeros have to adjust to keep $\boldsymbol{Ax} = \boldsymbol{b}$. Find the new $\boldsymbol{x}$ by back substitution and compute the change in the total cost $\boldsymbol{cx}$. This change is the "reduced cost" r of the new component. The entering variable is the one that gives the most negative r. This is the greatest cost reduction for a single unit of a new variable.

The more of the entering variable we include, the lower the cost. This has to stop when one of the positive components (which are adjusting to keep $\boldsymbol{Ax} = \boldsymbol{b}$) hits zero. The leaving variable is the first positive x_i (calling it as the i_{th} component of the $\boldsymbol{x}$ vector) to reach zero. When that happens, a neighboring corner has been found. Then start again (from the new corner) to find the next variables to enter and leave. When all reduced costs are positive, the current corner is the optimal $\boldsymbol{x}^*$. No zero component of $\boldsymbol{x}^*$ can become positive without increasing $\boldsymbol{cx}^*$. No new variable should enter. The problem is solved.

Example:

Minimize the cost $\boldsymbol{cx} = 3x_1 + x_2 + 9x_3 + x_4$. The constraints are $\boldsymbol{x} \geq \boldsymbol{0}$ and two equations $\boldsymbol{Ax} = \boldsymbol{b}$:

$$x_1 + 2x_3 + x_4 = 4,\ m = 2\ \textit{equations}$$
$$x_2 + x_3 - x_4 = 2,\ m + n = 4\ \textit{unknowns}$$

A starting corner[21] is $\boldsymbol{x} = (4, 2, 0, 0)$ which costs $\boldsymbol{cx} = 14$. It has $m = 2$ nonzeros and $n = 2$ zeros. The zeros are x_3 and x_4. The question is whether

[21] In order to understand how such a starting corner is found refer [18].

x_3 or x_4 should enter (become nonzero). Try one unit of each of them:

$$If\ x_3 = 1\ and\ x_4 = 0,\ then\ \boldsymbol{x} =\ (2,1,1,0)\ costs\ 16$$
$$If\ x_4 = 1\ and\ x_3 = 0,\ then\ \boldsymbol{x} =\ (3,3,0,1)\ costs\ 13$$

Compare those costs with 14. The reduced cost of x_3 is $r = 2$, positive and useless. The reduced cost of x_4 is $r = -1$, negative and helpful. The entering variable is x_4. How much of x_4 can enter? One unit of x_4 made x_1 drop from 4 to 3. Four units will make x_1 drop from 4 to zero (while x_2 increases all the way to 6). The leaving variable is x_1. The new corner is $\boldsymbol{x} = (0,6,0,4)$, which costs only $\boldsymbol{cx} = 10$. This is the optimal $\boldsymbol{x}^*$, but to know that we have to try another simplex step from $(0,6,0,4)$. Suppose x_1 or x_3 tries to enter: **Start from the corner (0,6,0,4)**

$$If\ x_1 = 1\ and\ x_3 = 0,\ then\ \boldsymbol{x} =\ (1,5,0,3)\ costs\ 11$$
$$If\ x_3 = 1\ and\ x_1 = 0,\ then\ \boldsymbol{x} =\ (0,3,1,2)\ costs\ 14$$

Those costs are higher than 10. Both r's are positive-it does not pay to move. The current corner (0,6,0, 4) is the solution $\boldsymbol{x}^*$.

2.3 Finding redundant inequalities

If a vertex in an n-dimensional space, satisfies more than n inequalties as equations then such a vertex is called *degenerate*. A vertex in an n-dimensional space is non-degenerate if it satisfies exactly n inequalties as equations. As we had discussed before, after bounding the polyhedra $\mathbf{Zc} \geq 0$ we use the vertex enumeration method to find all its extremal vertices the relevance of which has already been discussed in section 2.1.5. Such a bounded polyhedra contains origin, that is, $\mathbf{c=0}$ which is trivially satisfied as an extremal vertex. The number of inequalities in $\mathbf{Zc} \geq 0$ is equal to number of rows in $\mathbf{Z}$ matrix and the number of variables that appear in each such inequality is equal to the number of columns in $\mathbf{Z}$. Since the system of force balance equation $\mathbf{Ax=0}$ is *under-determined* (number of rows in matrix $\mathbf{A}$

less than the number of columns in **A**) therefore the number of rows in **Z** will always be greater than the number of columns in **Z**. And since each inequality in $\mathbf{Zc} \geq 0$ satisfies **c=0** therefore this implies that the origin **c=0** is degenerate. Such degenerate vertices reduce the efficiency of the vertex enumeration method [19]. We would therefore like to remove those inequalities in $\mathbf{Zc} \geq 0$ which are *redundant* (precisely defined below) which will reduce the degeneracy of **c=0** and make the enumeration of vertices efficient. Refer [12] to find a detailed discussion of such a method that we summarise below for the most general case:

Let $Ly \leq b,\ s^T y \leq t$ be a given system of $m+1$-inequalities in d-variables $y = (y_1, y_2, \ldots, y_d)^T$. We want to test whether the subsystem of first m inequalities $Ly \leq b$ implies the last inequality $s^T y \leq t$. If so, the inequality $s^T y \leq t$ is redundant and can be removed from the system. A linear programming (LP) formulation of this checking is rather straightforward:

$$\begin{aligned} f^* = \quad & \text{maximize} \quad s^T y \\ & \text{subject to} \quad Ly \leq b \\ & \qquad\qquad\ \ s^T y \leq t+1. \end{aligned}$$

Then the inequality $s^T y \leq t$ is redundant if and only if the optimal value f^* is less than or equal to t. By successively solving this LP for each untested inequality against the remaining, one would finally obtain a equivalent non-redundant system.

2.4 Vertex Enumeration Method [13]

Before we introduce this method we begin by defining terms which we will use later. These terms are commonly used in the literature of Graph Theory [8].

2.4.1 Definitions

- **Graph**: A graph is a structure amounting to a set of objects (called nodes/vertices) in which some pairs of the objects are "connected" (representatively by an edge).

- **Loop**: A loop (also called a self-loop or a "buckle") is an edge that connects a vertex to itself.
- **Cycle**: A cycle is a path of edges and vertices wherein a vertex is reachable from itself.
- **Degree of a vertex**: The degree (or valency) of a vertex of a graph is the number of edges incident to the vertex, with loops counted twice. In a regular graph, all degrees are the same, and so we can speak of the degree of the graph.
- **Path**: A path in a graph is a finite or infinite sequence of edges which connect a sequence of vertices which, by most definitions, are all distinct from one another.
- **Tree**: A tree is an undirected graph in which any two vertices are connected by exactly one path.
- **Spanning Tree**: An undirected graph G is a subgraph that is a tree which includes all of the vertices of G, with minimum possible number of edges.
- **Forest**: A forest is a disjoint union of trees.
- **Undirected graph**: An undirected graph is a graph in which edges have no orientation.
- **Spanning Forest**: A graph consisting of a spanning tree in each connected component of the graph.
- **Directed graph**: A directed graph (or digraph) is a graph that is a set of vertices connected by edges, where the edges have a direction associated with them.
- **Simple Graph**: Is an undirected graph in which both multiple edges (more than one edge connecting a pair of vertices) and loops are disallowed.

- **Regular Graph**: A graph in which each vertex has the same number of neighbours.

- **Complete Graph**: A graph in which each pair of vertices is joined by an edge. A complete graph contains all possible edges.

- **Root node/vertex**: The vertex at which a directed graph begins.

- **Depth First Search**: It is an algorithm for traversing or searching tree or a graph. The algorithm starts at a root node (selecting some arbitrary node as the root in case of a graph) and explores as far as possible along each branch before bactracking.

2.4.2 Introduction

We will now briefly describe the vertex enumeration method which is used to identify all the extremal vertices of a bounded polyhedron obtained by bounding $\mathbf{Zc} \geq 0$. The relevance of knowing all the extremal vertices of such a bounded polyhedra has already been discussed in section 2.1.5. Let us assume that an inequality of the form $\hat{e_*}^T\mathbf{c} \leq d$ bounds the polyhedra $\mathbf{Zc} \geq 0$, as discussed in section 2.1.5. Let $\hat{e_*}^T\mathbf{c} \leq d$ and $\mathbf{Zc} \geq 0$ together be represented as $\mathbf{b} + \mathbf{Z}_b\mathbf{c} \geq 0$ where, the matrix $\mathbf{Z}_b$ consists of all the row vectors of $\mathbf{Z}$ and also an extra row vector, that is, $\hat{e_*}^T$ that appears as the last row in $\mathbf{Z}_b$. The vector $\mathbf{b}$ is of the form $(0, 0, ..., 0, -d)^T$.

Let $\mathbf{Z}_b$ be a matrix of size $m \times n$ and $\mathbf{b}$ be an m dimensional vector then a polyhedron P is defined as:

$$P = \{\mathbf{c} \in R^n \ : \mathbf{b} + \mathbf{Z}_b\mathbf{c} \geq 0\}$$

Where the inequality in $\mathbf{b} + \mathbf{Z}_b\mathbf{c} \geq 0$ implies a componentwise inequality on the vector $\mathbf{b} + \mathbf{Z}_b\mathbf{c}$.

A point $\mathbf{c} \in P$ is a vertex of P if and only if it is the unique solution to a subset of n inequalities solved as equations. The *vertex enumeration problem* [7] is to output all vertices of a polytope P. A bounded polyhedron

is called a polytope. Formally, a polyhedron P is bounded if there exists $M \geq 0$, such that $|\mathbf{c}| \leq M,\ \forall\ \mathbf{c} \in P$).

The method of finding the vertices is known as Reverse Search for vertex enumeration and is described in detail in [14]. Reverse Search is central to vertex enumeration technique. It works as follows. Suppose we have a system of m linear inequalities defining an n-dimensional polyhedra in R^n and a vertex of that polyhedron given by the indices[22] of n inequalities whose bounding hyperplanes intersect at the vertex. These indices define a *cobasis* for the vertex. The complementary set of $m - n$ indices are called a *basis* (a non-degenerate vertex will have exactly one basis while a degenerate vertex in n-dimensional space will have as many as $\binom{k}{n} = \frac{k!}{n!(k-n)!}$ bases, where, k ($> n$) is the number of inequalities that the degenerate vertex $\mathbf{c}$ satisfies as equations) and $\binom{k}{n}$ represents the set of all n-combinations of a set with cardinality k. For any given linear objective function, the simplex method generates a path between adjacent bases(or equivalently cobases) which are those differing in one index. The path is terminated when a basis of a vertex maximizing this objective function is found. The path is found by pivoting, which involves interchanging one of the hyperplanes defining the current cobasis with one in the basis. The path chosen from the initial basis depends depends on the pivot rule used, which must be finite to avoid cycling[23] in the graph [15]. If we look at the set of all such paths from all bases of the polyhedron, we get a spanning tree of the graph of adjacent bases of the polyhedron. The root of this tree is a basis of an optimum vertex. The reverse search algorithm starts at this root and traces out its nodes in depth first order by reversing the pivot rule. We will illustrate in the next section how it works.

[22]Each index points to a unique inequality in the set of inequalities $\mathbf{b} + \mathbf{Z}_b\mathbf{c} \geq 0$.

[23]Such cycling can prevent the path from terminating and the optimum vertex may never be found.

2.4.3 Illustrations

2.4.3.1 Graph (Nodes and edges) of the bounded polyhedra

Let us assume that the **c**-space is 3-dimensional. Let c_1, c_2 and c_3 in figure 2.14 be the three coordinate directions and $n_0, n_1, n_2, n_3, n_4, n_5, n_6$ be the extremal vertices of the bounded polyhedra obtained as an intersection of $\hat{e_*}^T\mathbf{c} \leq d$ and $\mathbf{Zc} \geq 0$ (see section 2.1.5 for detailed discussion).

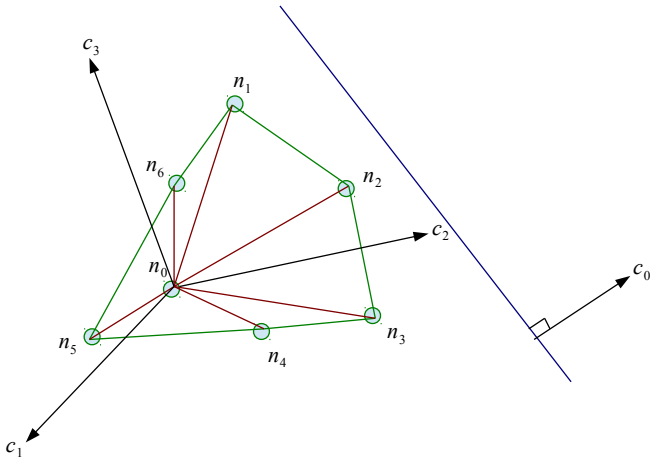

Figure 2.14: Graph of the bounded polyhedra

Note that the plane $\hat{e_*}^T\mathbf{c} = d$ bounds the polyhedra $\mathbf{Zc} \geq 0$. The edges of the bounded polyhedra are shown in solid red and green lines in figure 2.14. All vertices except the origin (n_0) lie in the plane $\hat{e_*}^T\mathbf{c} = d$ ($\hat{e_*}$ is normal to this plane) which bounds the open polyhedra $\mathbf{Zc} \geq 0$. The points, including $n_1, n_2, n_3, n_4, n_5, n_6$, which lie within and on the boundary formed by the green edges in this plane, $\hat{e_*}^T\mathbf{c} = d$, satisfy $\mathbf{Zc} \geq 0$, and the points in this plane which lie outside the boundary do not satisfy $\mathbf{Zc} \geq 0$. The solid green lines also represent the connectivity of $n_1, n_2, n_3, n_4, n_5, n_6$ in this plane. The solid lines in red, the points on which also satisfy $\mathbf{Zc} \geq 0$, are the edges that connect the origin, n_0, to these vertices. Also, consider a plane with a unit normal vector $\mathbf{c_0}$, see figure 2.14, such that $\mathbf{c_0}$ and $\hat{e_*}$ are orthogonal to each other. The plane with $\mathbf{c_0}$ as a surface normal vector, shown in figure 2.14,

appears as a straight line (shown in blue in figure 2.14) when viewed along the normal vector $\hat{e_*}$.

2.4.3.2 Spanning tree on the graph of bounded polyhedra

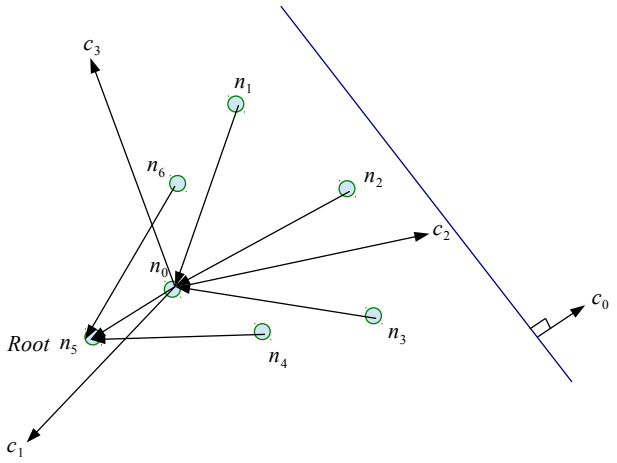

Figure 2.15: Spanning tree on the Graph of the bounded polyhedra

Now, if a linear function of the form $\mathbf{c_0}^T\mathbf{c}$, with $\mathbf{c_0}$ same as described in 2.14, is then minimized subject to the region described by the bounded polyhedra in figure 2.14 then n_5 becomes the optimum vertex. The simplex method turns the connectivity of vertices in figure 2.14 into that of a 'spanning' tree as shown in figure 2.15. The arrows show the path traversed by the simplex method from a given vertex to the optimum vertex. It must be noted that the cycles that existed in the graph of the bounded polyhedra are eliminated due to the simplex method. Degenerate vertices, such as n_0, may prevent the simplex method from reaching the optimum vertex. In such case the simplex method uses a special pivot rule called *Bland's rule* [15] to resolve the problem. Usage of this rule for degenerate vertices ensures the construction of spanning tree as shown in figure 2.15.

The reverse search for vertex enumeration starts at the optimum vertex n_5, also called the root of the search tree, and does a depth first search of

the spanning tree in figure 2.15 to list all the vertices and thus giving us all the extremal vertices that we desire.

For a polyhedron with m inequalities in d variables the vertex enumeration method finds all bases in time $\mathcal{O}(md^2)$ per basis. Thus, for example, if a polyhedron has n vertices, and all non-degenerate, then the vertex enumeration method will list all the vertices in time $\mathcal{O}(nmd^2)$.

2.5 Volume calculation of polytopes

The number of vertices of a bounded polyhedra may sometime become exponentially large as we shall see in section 3.1. In this case we want to devise a method to find a subset of vertices such that they can be used to span 'most' of the region contained in $\mathbf{Zc} \geq 0$, if not all of it.

If we pick points from all the edges of $\mathbf{Zc} \geq 0$ then using these points we can span the entire space of solutions in $\mathbf{Zc} \geq 0$ and hence by the relationship $\mathbf{x}$=$\mathbf{Zc}$ can construct any possible force network that the system might admit. But in figure 3.6 we will see that the number of such points grow exponentially and therefore enumeration of all the points becomes infeasible. We would therefore like to have a smaller subset of points/vertices (since we cannot enumerate an exponential number of points/vertices) using which most of the region in $\mathbf{Zc} \geq 0$ can be covered. This would help us construct most of the force networks that a granular system might admit if not all of them.

In fact in section 3.1.6 we suggest and demonstrate a method to construct a subset of vertices that we believe does well at approximating the volume of the actual/original bounded polyhedra. In order to quantify the success of such an approximation we must be able to calculate what fraction of the actual volume of the bounded polytope is recovered using such subset of vertices. This motivates our volume calculations.

All known algorithms for exact volume computation decompose a given polytope into simplices and thus they all rely on the volume formula of a simplex:

$$\text{Vol}(\Delta(v_0,..,v_d)) = \mid \det(v_1 - v_0,..,v_d - v_0) \mid / \; d!$$

where $\Delta(v_0,..,v_d)$ denotes the simplex in R^d with vertices $v_0,..,v_d \in R^d$. There are two types of methods for volume computation, depending on how a given polytope P is decomposed into simplices. They are the triangulation methods and signed decomposition methods. We use triangulation methods for volume calculation of our polytope.

A *triangulation* of a d-polytope P is a set $(\Delta_i : i = 1, \ldots , s)$ of d-simplices such that

$$P = \bigcup_{i=1}^{s} \Delta_i$$

and no distinct simplices have an interior point in common. Then the volume of P is simply the sum of the volumes of the simplices:

$$Vol(P) = \sum_{i=1}^{s} Vol(\Delta_i)$$

We use the free code developed by authors in [13] for volume calculation of polytopes. It uses a triangulation method to do so. For more on how this triangulation/decomposition of P is achieved please refer [16].

Chapter 3

Results

3.1 Snooker-triangle probem

3.1.1 Introduction to the system

We study the polyhedra corresponding to a system of 2-dimensional identical hard disks contained in an equi-lateral triangular box. The smallest system size would correspond to 3 disks as shown below:

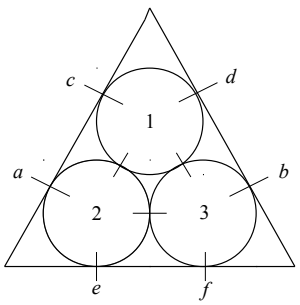

Figure 3.1: Smallest system in a snooker triangle

It is assumed that there exists no friction in the system. The unknown contacts are marked in the figure 3.1. In figure 3.2 we show the scaling of the

number of unknown contacts versus the number of particles in the system. The slope of the curve in figure 3.2 is equal to three which means that the number of unknown contacts, for such a system, is three times the number of particles.

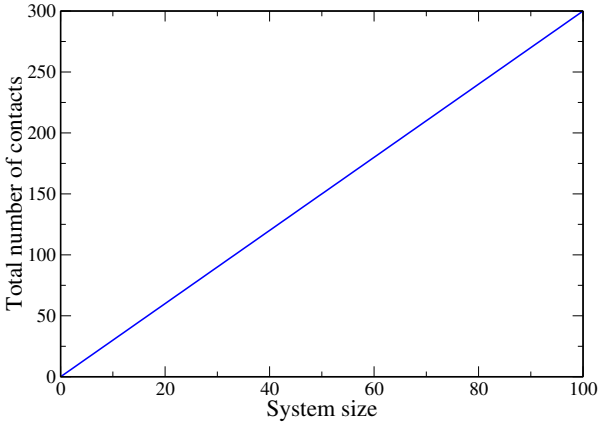

Figure 3.2: Number of contacts vs System size

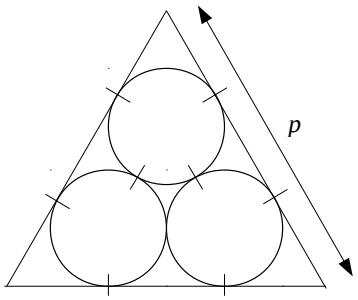

Figure 3.3: Equi-triangular system

If **A** is of the size $m \times n$ and the rank is r then dimensionality of the space of solutions to **Ax=0** is $n - r$. Let the number of particles touching one side of the boundary of the triangle be p, as shown in figure 3.3. We have assumed completely frictionless system in 2-dimensions. Therefore the

number of force balance equations, m, will be $p(p+1)$ and the number of unknown contacts, n, also marked as small dashed lines in figure 3.3, will be $\frac{3p(p+1)}{2}$. Now, the rank r of the contact matrix $\mathbf{A}$ for such system is equal to the number of rows in $\mathbf{A}$, that is, $r = m$[1]. Therefore, the dimensionality of nullspace of $\mathbf{A}$ becomes $n - m$ which is $\frac{3p(p+1)}{2} - p(p+1) = \frac{p(p+1)}{2}$. But $\frac{p(p+1)}{2}$ equals the number of particles in the system. This implies that *the dimensionality of nullspace of* $\boldsymbol{A}$ *equals the number of particles in the system.* And, since the number of unknown contacts equal three times the number of particles, therefore, *the number of unknown contacts equal three times the dimensionality of nullspace of matrix* $\mathbf{A}$.

3.1.2 Number of vertices (and bases) vs System size

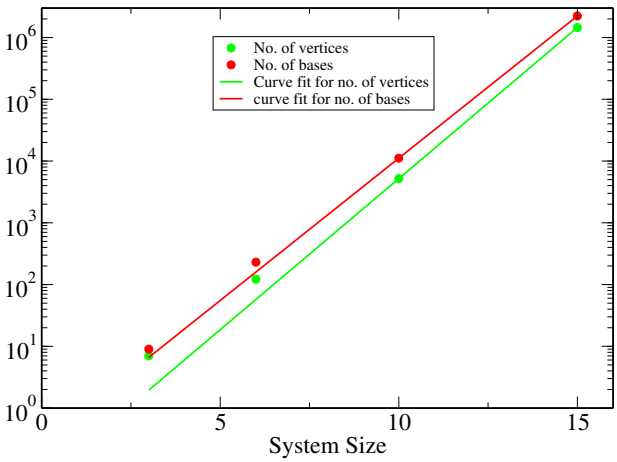

Figure 3.4: Number of vertices (and bases) vs System size

We show in figure 3.4 that, for equi-triangular systems, the number of vertices of the bounded polyhedra, as obtained after bounding $\mathbf{Zc} \geq 0$, scales exponentially with the number of disks/particles in the system.

It was noted in section 3.1.1 that the dimensionality of nullspace of the contact matrix $\mathbf{A}$ equals the number of particles in the system. Since the dimensionality of the nullspace of $\mathbf{A}$ also equals the dimensionality of the

[1]This is an empirical fact that we have noticed during our calculations.

c-space[2], therefore, dimensionality of the **c**-space equals the number of particles. Hence, we can say that for the equi-triangular systems the number of vertices of the bounded polyhedra (note that such a polyhedra lives in **c**-space) grow exponentially with the dimension of **c**-space. A typical polyhedra which shows such kind of exponential scaling is a hypercube the number of vertices of which grow as 2^d, where, d is the dimensionality of the space in which the hypercube lies.

If a vertex in an n-dimensional space, satisfies more than n inequalties as equations then such a vertex is called *degenerate*. A basis of a vertex, in an n-dimensional space, is a distinct set of n number of inequalities which when solved as equations give the coordinates of that vertex in the n-dimensional space. Let us say that a vertex in an n-dimensional space satisfies k ($> n$) number of inequalities as equations then the number of bases for such a vertex will have $\binom{k}{n}$ bases. If a vertex in an n-dimensional space, satisfies exactly n inequalties (not more) as equations then such a vertex is called *non-degenerate*. A non-degenerate vertex will therefore have exactly one basis which is also equal to $\binom{n}{n}$.

In figure 3.4 we see that the number of bases are more than number of vertices because **c=0** is a degenerate vertex which satisfies $\mathbf{Zc} \geq 0$.

We will now give an upper bound on the number of vertices which will serve as an evidence that an exponential scaling of vertices is possible. If the contact matrix **A** is of the size $m \times n$ with rank r then $\mathbf{Zc} \geq 0$ will have n inequalities in $n - r$ variables. Also, let us assume that we have bounded $\mathbf{Zc} \geq 0$ using the methods described in Chapter 2 and obtained a bounded polyhedra. The number of vertices of the bounded polyhedra is then one

[2]Consider the system of force balance equation **Ax=0** and the relationship **x=Zc** as discussed in Chapter 2. Let **A** be of the size $m \times n$ and rank r. Then the solutions **x** to **Ax=0** are n-dimensional vectors (vectors with n-components) which all lie in an $n - r$ dimensional plane. This $n - r$ dimensional plane is called the nullspace of the matrix **A**. The **Z** matrix will consist of the basis vectors of nullspace of **A**. Hence the size of **Z** matrix will be $n \times n - r$. The **c**-space on the other hand would be an $n - r$ dimensional vector space consisting of all the possible vectors with $n - r$ components (a vector in **c**-space is denoted by **c**). A vector in **c**-space has $n - r$ components while a vector in nullspace of **A** has n components. The set of **c** vectors which satisfy the system of linear inequalities $\mathbf{Zc} \geq 0$ is a subset of the **c**-space. The polyhedra obtained after bounding $\mathbf{Zc} \geq 0$ is also a subset of **c**-space therefore.

more (due to the vertex at **c=0**) than the number of edges of the unbounded polyhedra $\mathbf{Zc} \geq 0$[3]. Such an edge is obtained by solving $n-r-1$ inequalities of $\mathbf{Zc} \geq 0$ as equations, which also in addition, satisfy all the remaining $r+1$ inequalities in $\mathbf{Zc} \geq 0$. An upper bound on the number of such edges (such a number can of course serve as an upper bound on the number of vertices as each vertex of the bounded polyhedra, except **c=0**, corresponds to an edge of the unbounded $\mathbf{Zc} \geq 0$) will be obtained if the edges, obtained by solving $n-r-1$ inequalities of $\mathbf{Zc} \geq 0$ as equations, don't necessarily have to satisfy the remaining $r+1$ inequalities in $\mathbf{Zc} \geq 0$. Obviously, this upper bound would be $\binom{n}{n-r-1}$. We calculate this upper bound for different number of particles for the equi-triangular system and obtain the result in figure 3.5 on a log-linear scale.

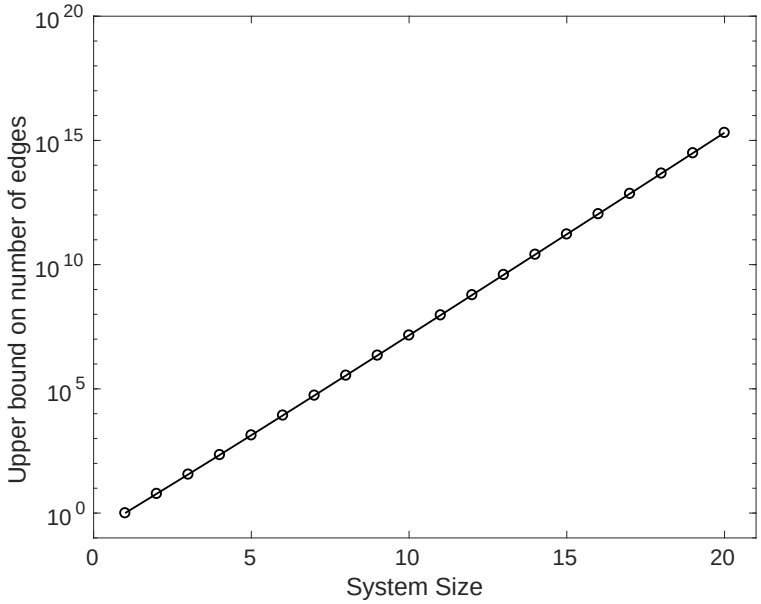

Figure 3.5: Upper bound on the number of edges of $\mathbf{Zc} \geq 0$

Figure 3.5 indeed suggests that the vertices can rise exponentially as a function of system size since the upper bound on such vertices is exponential

[3]Let $\mathbf{Zc} \geq 0$ be n inequalities in $n-r$ variables. As discussed in Chapter 2 we are interested in picking points from each edge of the polyhedra $\mathbf{Zc} \geq 0$. Such an edge is obtained by solving $n-r-1$ inequalities of $\mathbf{Zc} \geq 0$ as equations, which also in addition, satisfy all the remaining $r+1$ inequalities in $\mathbf{Zc} \geq 0$. Except the point **c=0** each point of the bounded polyhedra is a point on such an edge.

in system size. It must be noted that for any given system size the value of the upper bound is more than the corresponding number of vertices observed in figure 3.4 obviously as expected.

Since the number of vertices grow exponentially and the vertex enumeration method of finding vertices is at best linear in time with the number of vertices enumerated [14] it is not feasible to enumerate all the vertices for such systems of large sizes in practical time.

3.1.3 Removal of redundant inequalities in $\mathbf{Zc} \geq 0$

3.1.3.1 Motivation

We know that each inequality in $\mathbf{Zc} \geq 0$ satisfies $\mathbf{c=0}$. This implies that the origin $\mathbf{c=0}$ is degenerate. Such degenerate vertices reduce the efficiency of the vertex enumeration method [19]. We would therefore like to remove those inequalities in $\mathbf{Zc} \geq 0$ which are *redundant* (defined precisely in section 2.3) so that the degeneracy of $\mathbf{c=0}$ is reduced and thus making the enumeration of vertices more efficient than otherwise.

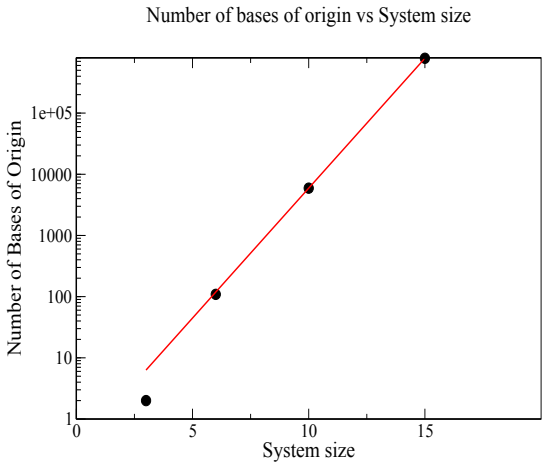

Figure 3.6: Number of bases of origin vs System size

Vertex enumeration method goes through each basis of a vertex before listing it out. Our objective is to enumerate vertices for larger system sizes.

Since the number of basis of the origin, **c=0**, which is a degenerate vertex, grows exponentially, seen in figure 3.6, its important to check if fewer inequalities in $\mathbf{Zc} \geq 0$ can be used (and thus reducing the number of bases of origin) to construct the same polyhedra to make the enumeration possible in practical time. The inequalities which can be removed without affecting the polyhedra formed by $\mathbf{Zc} \geq 0$ are called redundant.

3.1.3.2 Results

The method of obtaining the results in this section has been discussed in section 2.3. We see that the number of redundant inequalities is nearly zero for all system sizes making the enumeration infeasible for large system sizes using the method in [14].

System size	size of A	number of redundant inequalities
3	6x9	2
6	12x18	0
10	20x30	1
15	30x45	0
21	42x63	0
28	56x84	0

Table 3.1: Number of redundant constraints vs sytem size

3.1.4 Volume computation and approximations

3.1.4.1 Motivation

If we pick points from all the edges of $\mathbf{Zc} \geq 0$ then using these points we can span the entire space of solutions in $\mathbf{Zc} \geq 0$ and hence by the relationship $\mathbf{x}=\mathbf{Zc}$ can construct any possible force network that the system might admit (see section 2.1.5 for details). But in figure 3.6 we saw that the number of these points grew exponentially with system size and therefore enumeration of all the points becomes infeasible for larger system size. We would therefore like to have a smaller subset of points/vertices (since we cannot enumerate an exponential number of points/vertices) using which most of the region

in $\mathbf{Zc} \geq 0$ can be covered. This would help us construct most of the force networks that the system might admit if not all of them.

3.1.5 Approximating the volume of the polytope

The goal is to find a subset of vertices of the polytope obtained after bounding the open polyhedra $\mathbf{Zc} \geq 0$ so that using this subset of vertices we can span as much space in $\mathbf{Zc} \geq 0$ if not all. The entire solution space $\mathbf{Zc} \geq 0$ can be sampled by taking convex combination of the vertices of the bounded polyhedra (the polyhedra obtained after bounding $\mathbf{Zc} \geq 0$) and then multiplying the resultant vector by positive scalar. This had been discussed before in section 2.1.5. If only a subset of vertices is known then only a fraction of the region $\mathbf{Zc} \geq 0$ can be spanned.

We demonstrate a method which is based on geometric intuition and show through simple volume calculations that the method does well at approximating the solution space $\mathbf{Zc} \geq 0$.

3.1.6 First strategy: Constructing the box

All the edges of $\mathbf{Zc} \geq 0$ we are interested in emanate from origin. The polyhedra $\mathbf{Zc} \geq 0$ is unbounded. Consider $\mathbf{c}_*^T\mathbf{c} \leq d$ or $\hat{e_*}^T\mathbf{c} \leq d$, whichever bounds $\mathbf{Zc} \geq 0$. Say, $\hat{e_*}^T\mathbf{c} \leq d$ bounds $\mathbf{Zc} \geq 0$[4]. The bounding plane of $\hat{e_*}^T\mathbf{c} \leq d$ that is $\hat{e_*}^T\mathbf{c} = d$ intersects with the edges of $\mathbf{Zc} \geq 0$ in points. We find a special subset of these points with a property that we describe and illustrate now.

3.1.6.1 Method (Box fit method)

(i) Let the polyhedra $\mathbf{Zc} \geq 0$ lie in an n-dimensional $\mathbf{c}$-space. Let us assume that the constraint $\hat{e_*}^T\mathbf{c} \leq d$ bounds the polyhedra $\mathbf{Zc} \geq 0$. Find a set of $n-1$ unit vectors, say $\mathbf{k}_1$,..,$\mathbf{k}_{n-1}$, such that together with $\hat{e_*}$ they form an orthonormal basis of the n-dimensional $\mathbf{c}$-space. Such

[4]The method of approximating volume does *not* depend upon whether $\mathbf{c}_*^T\mathbf{c} \leq d$ bounds $\mathbf{Zc} \geq 0$, or, $\hat{e_*}^T\mathbf{c} \leq d$ bounds $\mathbf{Zc} \geq 0$. So one can simply replace $\hat{e_*}$ with $\mathbf{c}_*$ in the entire discussion of volume approximation of the bounded polyhedra.

an orthonormal basis will not be unique. But we need one, anyone, for our purpose.

(ii) Construct a set of $n-1$ linear functions of the form $\mathbf{k}_1^T\mathbf{c}$,..,$\mathbf{k}_{n-1}^T\mathbf{c}$ (let this set be labelled as S).

(iii) Do a minimisation and a maximisation of each linear function in the set S constrained by $\hat{e_*}^T\mathbf{c} = d$ and $\mathbf{Zc} \geq 0$ (The equality sign in $\hat{e_*}^T\mathbf{c} = d$ put as a constraint will help us keep strictly in the plane which holds all the vertices except the trivial vertex **c=0**). This will give us a pair of points/vertices corresponding to each linear function. The minimisation and maximisation of each $n-1$ linear function generates $2(n-1)$ points/vertices. These $2(n-1)$ vertices of the bounded polyhedra along with the vertex at the origin is a special subset of vertices which we show through our volume calculations a good approximation of the solution space $\mathbf{Zc} \geq 0$.

It must be noted that we do not claim to have found an upper or a lower bound on the percentage of volume of the space of solutions recovered using such an approximation.

3.1.6.2 Illustrating the Method

We now show what the vertices in this special subset look like using geometric illustrations. We show in figure 3.7 a cartoon of the bounded polyhedra obtained as an intersection of $\hat{e_*}^T\mathbf{c} \leq d$ and $\mathbf{Zc} \geq 0$ in a 3-dimensional space[5]. Note that the figure 3.7 has the same description as was used to describe figure 2.14 in section 2.4.3.1. The cartoon picture of the bounded polyhedra has seven vertices $n_0, n_1, n_2, n_3, n_4, n_5$ and n_6. Except the vertex n_0 at origin, all the vertices lie in a plane which is the bounding hyperplane of the half-space that removes the part of the polyhedra extending to infininty, which in other words mean that all the vertices of the bounded polyhedra, except **c=0**, resulting from the intersection of $\hat{e_*}^T\mathbf{c} \leq d$ and $\mathbf{Zc} \geq 0$ are contained in

[5]Figures 3.7 and 3.8 are in 3-dimensional space and the three coordinate directions are marked as c_1, c_2 and c_3.

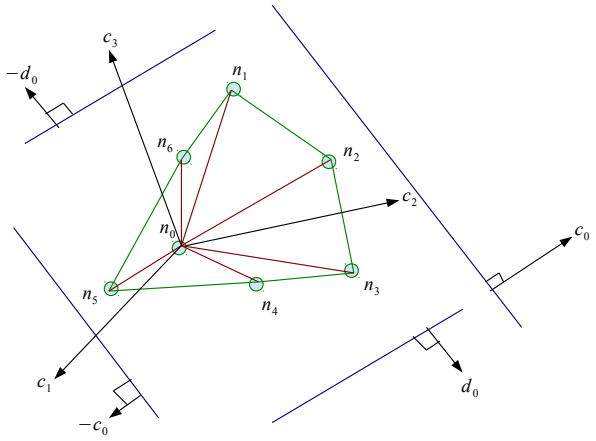

Figure 3.7: Fitting the vertices in a box

the plane $\hat{e_*}^T\mathbf{c} = d$(See section 2.1.5 of Chapter 2 for details). The bounded polyhedra can also be represented as a *convex hull*[6] of its vertices. In figure 3.7 a convex hull of the vertices $n_0, n_1, n_2, n_3, n_4, n_5$ and n_6 is the bounded polyhedra. These two representations are equivalent because the bounded polyhedra is a convex polytope and a convex polytope is a convex hull of its vertices [10].

The box fit method suggests that in 3-dimensions we must have two unit vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ such that $\mathbf{k}_1$, $\mathbf{k}_2$ and $\hat{e_*}$ form an orthonormal basis of the 3-dimensional space. Let's call $\mathbf{k}_1$ and $\mathbf{k}_2$ as c_0 and d_0, respectively, to have notational consistency with figures 3.7 and 3.8.

The minimisation and maximisation of the linear functions $c_0^T\mathbf{c}$ and $d_0^T\mathbf{c}$ against the constraints $\hat{e_*}^T\mathbf{c} = d$ and $\mathbf{Zc} \geq 0$ (In figure 3.7 these constraints will define a convex hull of $n_1, n_2, n_3, n_4, n_5, n_6$ which is nothing but only that part of the bounding plane $\hat{e_*}^T\mathbf{c} = d$ which has non-zero intersection with $\mathbf{Zc} \geq 0$) amounts to saying that the four planes in solid blue color shown in figure 3.8 touch the extreme points of the polytope from outside giving two pairs of vertices. The pair n_1 and n_3 come respectively from minimising and maximising $d_0^T\mathbf{c}$ against $\hat{e_*}^T\mathbf{c} = d$ and $\mathbf{Zc} \geq 0$ and the pair n_5 and n_2 come

[6]The convex hull of a set U of points in a Euclidean space is the smallest convex set that contains U [10].

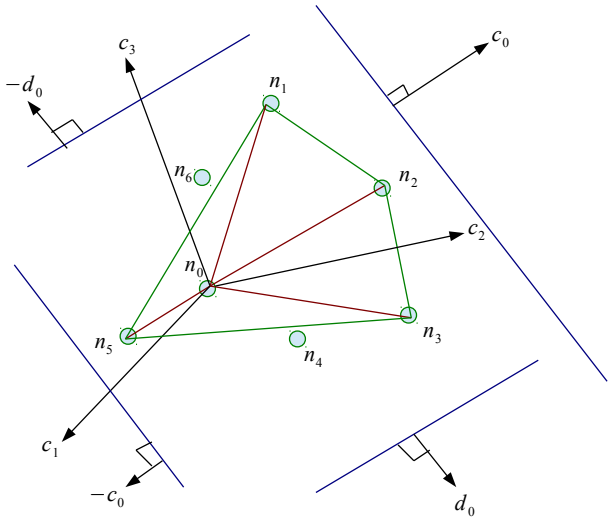

Figure 3.8: A smaller solution space using the box fit

respectively from minimising and maximising $c_0^T\mathbf{c}$ against $\hat{e_*}^T\mathbf{c} = d$ and $\mathbf{Zc} \geq 0$. Thus we obtain a special subset of vertices which include n_1, n_2, n_3 and n_5. Hence our approximation to the bounded polyhedra in figure 3.7 is a convex hull of the special subset of vertices, that is, n_0, n_1, n_2, n_3 and n_5. Of course the volume of the convex hull of n_0, n_1, n_2, n_3 and n_5 is only a fraction of the volume of the convex hull of the vertices $n_0, n_1, n_2, n_3, n_4, n_5$ and n_6.

We obtain the following results based on the box-fit method. These results are for the equilateral-triangular system as discussed in section 3.1.1.

System size	fraction of total vertices	fraction of total volume
3	0.42	0.90
6	0.081	0.254

Table 3.2: Fraction of total vertices vs fraction of total volume

We calculate the volume of the convex hull of a subset of the vertices of the bounded polyhedra obtained through the box-fit method. We see through these results that for a system size of three knowing 42 % of the total number of vertices of the bounded polyhedra through the box fit method helps us cover 90 % of the solution space. And for a system size of six knowing 8.1

% of the total number of vertices of the bounded polyhedra through the box fit method helps us cover 25.4 % of the solution space.

The box in figure 3.7 (shown in solid blue lines) which covers the bounded polyhedra from outside can be rotated about the $\hat{e_*}$ which is the surface normal to the bounding plane $\hat{e_*}^T\mathbf{c} = d$, of $\mathbf{Zc} \geq 0$, to construct more boxes which can then be used to generate more vertices when dealing with larger systems in higher dimensions. This forms the basis of improvement of the box fit method that we mention in next section.

3.1.7 Improvement over first strategy: Using two boxes

It is possible to improve upon the previous results. If we rotate the first box by $45-degrees$ about the surface normal vector of the bounding plane we get a second box (one more rotation by $45 - degrees$ about the surface normal vector will give the original box). The first step of finding the second box is to find a unit vector $\mathbf{k}_1^2$ (the superscript 2 implies that $\mathbf{k}_1^2$ belongs to the second box) such that it is orthogonal to $\hat{e_*}$ and makes an angle of 45-degrees with the unit vector $\mathbf{k}_1$ of the first box. Such a $\mathbf{k}_1^2$ will not be unique, however, we need any one for our purpose. The second and final step of finding the second box is to find a set of $n-2$ unit vectors $\mathbf{k}_2^2$,..,$\mathbf{k}_{n-1}^2$ such that $\mathbf{k}_1^2$,$\mathbf{k}_2^2$,..,$\mathbf{k}_{n-1}^2$ along with $\hat{e_*}$ form an orthonormal basis of the n-dimensional $\mathbf{c}$-space.

Finally, once again, construct a set of $n-1$ linear functions of the form $\mathbf{k}_1^{2T}\mathbf{c}$,..,$\mathbf{k}_{n-1}^{2T}\mathbf{c}$ and do a minimisation and a maximisation of each of the linear function constrained by $\hat{e_*}^T\mathbf{c} = d$ and $\mathbf{Zc} \geq 0$ to obtain another $2(n-1)$ set of points.

Following such an approach we then compute the volume of the convex hull of entire $2(n-1)+2(n-1)$ points to see what percent of solution space is covered for the equi-triangular system of system size six.

System size	fraction of total vertices	fraction of total volume
6	0.139	0.923

Table 3.3: Fraction of total vertices vs fraction of total volume

Significant improvement is observed by using two boxes. The strategy to

approximate the bounded polyhedra then becomes to use multiple boxes in the following way:

Let the number of boxes be L and the **c**-space n-dimensional. Also, let the $n-1$ unit vectors corresponding to the first box be represented by $\mathbf{k}_1^1$,..,$\mathbf{k}_{n-1}^1$ (the superscript 1 implies that $\mathbf{k}_1^1$ belongs to the 1_{st} box). As discussed before, $\mathbf{k}_1^1$,..,$\mathbf{k}_{n-1}^1$ together with $\hat{e_*}$ forms an orthonormal basis of the n-dimensional **c**-space.

(i) Construct the first box using the box-fit method.

(ii) The n_{th} box is then constructed by, first, finding a unit vector $\mathbf{k}_1^n$ (the superscript n implies that $\mathbf{k}_1^n$ belongs to the n_{th} box) such that it is orthogonal to $\hat{e_*}$ and makes an angle of $\theta = (n-1)(\pi/2)/L$ radians with the unit vector $\mathbf{k}_1^1$ of the first box, and then, finding a set of $n-2$ unit vectors $\mathbf{k}_2^n$,..,$\mathbf{k}_{n-1}^n$ such that $\mathbf{k}_1^n$,$\mathbf{k}_2^n$,..,$\mathbf{k}_{n-1}^n$ along with $\hat{e_*}$ form an orthonormal basis of the n-dimensional **c**-space.

(iii) For each box find the $2(n-1)$ vertices using the box-fit method. Then for L boxes we get in total $2L(n-1)$ vertices.

A convex hull of these $2L(n-1)$ vertices and the vertex at origin approximates the solution space which through our volume calculations believe approximates the solution space well.

3.1.8 Conclusion

We finally conclude by listing our observations for systems of frictionless disks in equi-lateral triangular box in 2-dimensions.

(i) The number of vertices of the bounded polyhedra corresponding to equilateral-triangular lattice scale exponentially with system size.

(ii) We use the method of authors in [14] to enumerate the vertices of the bounded polyhedra. Since the method is at best linear in number of vertices enumerated [14], an exponential number of vertices will take exponential time. Enumerating all the vertices of the bounded polyhedra becomes infeasible therefore.

(iii) Since the number of vertices become exponentially large we want to make a choice of a subset of vertices, a convex hull of which, covers as much of the volume of the bounded polyhedra. To do so, we have proposed and demonstrated through preliminary volume calculations that our method performs well.

3.2 Square lattice

We now study square and rhombic lattices with the aim to figure out what causes an exponential rise in number of vertices of the polytope. More precisely we want to find out the role of co-ordination number of particles and their arrangement in determining the number of vertices of the polytope.

3.2.1 Introduction to the system

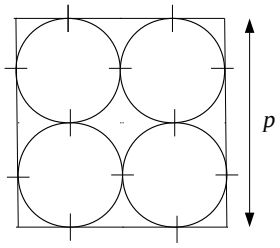

Figure 3.9: Square Lattice

We study the polyhedra corresponding to a system of identical hard disks arranged on a square lattice. The smallest system size would correspond to 4 disks as shown in figure 3.9.

If $\mathbf{A}$ is of the size $m \times n$ and the rank is r then dimensionality of the space of solutions to $\mathbf{Ax=0}$ is $n-r$. Let the number of particles touching one

side of the square boundary be p, as shown in figure 3.9. We have assumed completely frictionless system in 2-dimensions. Therefore the number of force balance equations, m, will be $2p^2$ and the number of unknown contacts, n, also marked as small dashed lines in figure 3.9, will be $2p(p+1)$. Now, the rank r of the contact matrix $\mathbf{A}$ for such system is observed to be equal to the number of rows in $\mathbf{A}$, that is, $r = m$ (an empirical fact that we have noticed during our calculations). Therefore, the dimensionality of nullspace of $\mathbf{A}$ becomes $n-m$ which is $2p(p+1)-2p^2 = 2p$. But p^2 equals the number of particles in the system. Therefore, *the dimensionality of nullspace of* $\boldsymbol{A}$ *equals the twice the square root of the number of particles in the system*, that is, $2\sqrt{N}$, where N is the number of particles in the system.

It is assumed that there exists no friction in the system. Also, since the number of unknown contacts equal $2p(p+1)$ and the number of disks/particles in the system equals p^2, therefore, the number of unknown contacts scales as $2(N+\sqrt{N})$, where, N is the number of particles in the system. This is shown in the figure 3.10:

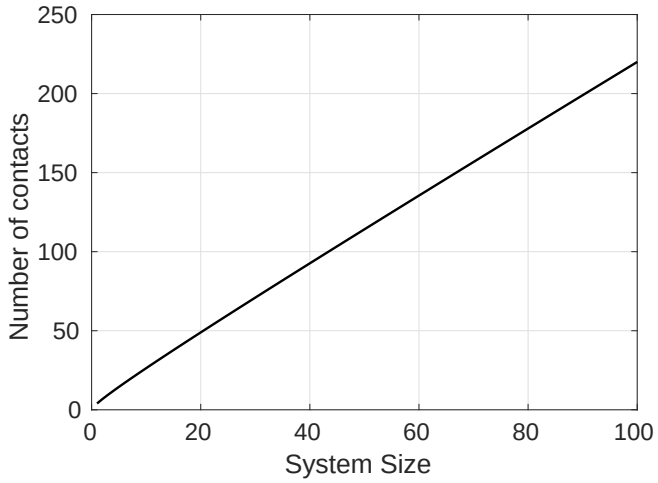

Figure 3.10: Number of contacts vs System size

3.2.2 Number of vertices vs System size

We repeat the same exercise as in the snooker-triangle problem to obtain the following result:

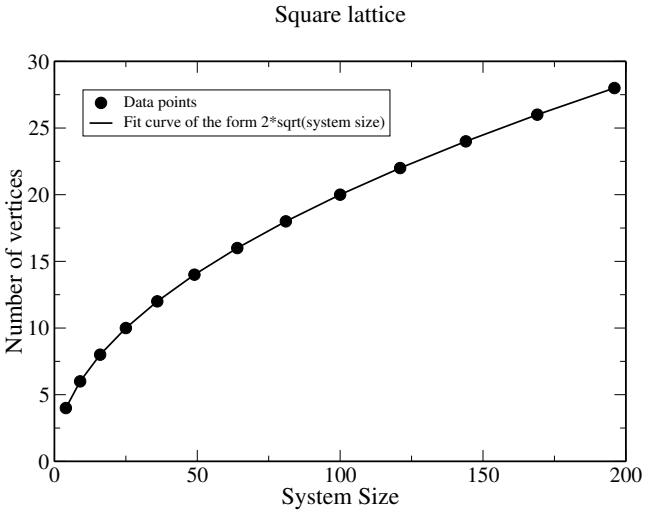

Figure 3.11: Number of vertices vs System size

We see a contrasting change in the number of vertices and it's scaling with system size. The number of vertices scaled exponentially with system size in the snooker triangle problem but in the case of square lattice it scales as square root of the system size.

The force transmitted along a row or a column of the square lattice takes a value that is constant across all the contacts in that row or column. The rows are parallel to each other and so are the columns and since the the rows and columns intersect at 90-degrees therefore the value of the constant can be varied independently for each row or column. This implies that each row or column of the square lattice contributes to one independent solution to the system of force balance equation **Ax=0**. Since the number of rows and columns add to $2\sqrt{N}$, where, N is the number of disks, therefore the number of independent solutions to **Ax=0** for a non-cohesive system of disks on a square lattice must also scale as $2\sqrt{N}$. This explains our result.

3.2.3 Polytope corresponding to square lattice is a simplex

In this section we show the variation of number of vertices with dimension of space in which they lie. We observe in figure 3.12 that the number of

vertices equals the dimension of null space of the contact matrix **A**. This number does not include the trivial vertex at origin. Hence, counting that in we see that polytopes corresponding to square lattices obey the following relationship:

$$Number\ of\ vertices = \{dimensionality\ of\ nullspace\ of\ the\ contact\ matrix\ \mathbf{A}\ \} + 1$$

Hence we conclude that the polytope corresponding to a square lattice is a simplex.

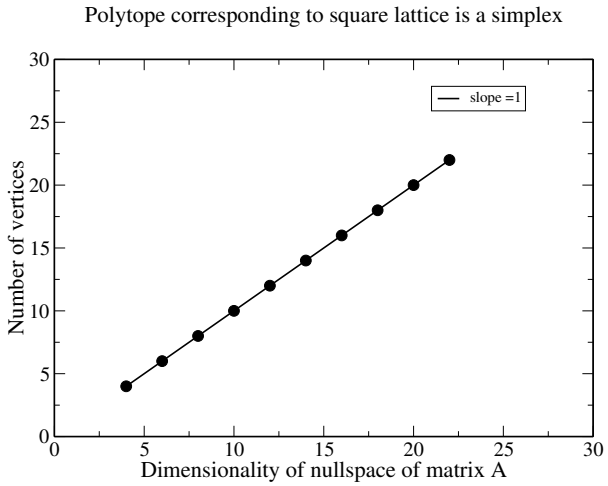

Figure 3.12: Number of vertices vs dimensionality of nullspace of the contact matrix **A**

This result can be explained in the following way. If the contact matrix **A** is of the size $m \times n$ and rank r then dimensionality of the space of solutions to **Ax=0** is $n - r$. It was found empirically that $r = m$ for contact matrix of square lattices. Since the nullspace of **A** is $n - m$ dimensional, therefore, the number of columns in **Z** equal $n - m$. And therefore, a **c** vector by the relationship **x=Zc** must have $n - m$ components. The **c**-space is an $n - m$ dimensional vector space which consists of all the possible vectors **c** with $n - m$ components and not just those **c** vectors which satisfy $\mathbf{Zc} \geq 0$. Therefore, the set of **c** vectors satisfying $\mathbf{Zc} \geq 0$ is a subset of **c**-space. Note that the dimension of nullspace of **A** and dimension of **c**-space both equal

$n - m$ and also that the polyhedra $\mathbf{Zc} \geq 0$ lies in $\mathbf{c}$-space.

Since the number of vertices of the bounded polyhedra depends upon the number of *non-redundant* (see section 2.3 for precise definition) inequalities in $\mathbf{Zc} \geq 0$ we plot and compare the number of inequalities in $\mathbf{Zc} \geq 0$ and the number of non-redundant inequalities in $\mathbf{Zc} \geq 0$ as a function of dimension of nullspace of $\mathbf{A}$ as shown in figure 3.13.

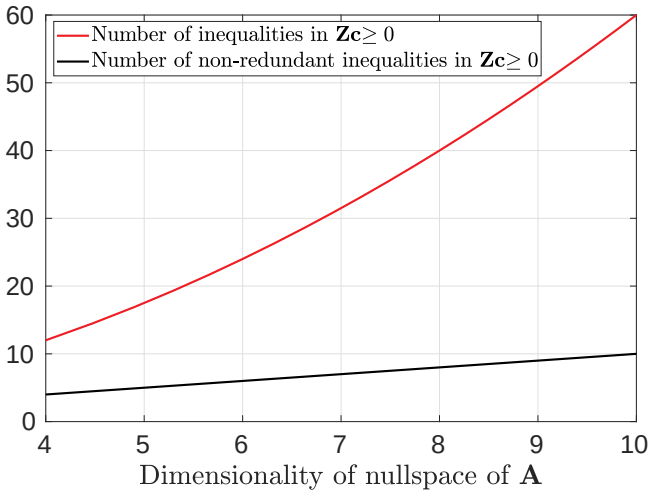

Figure 3.13: Number of inequalties and non-redundant inequalities vs Dimensionality of nullspace of contact matrix $\mathbf{A}$ for square lattices

For the square lattice, we find that the number of non-redundant inequalities in $\mathbf{Zc} \geq 0$ equals the dimensionality of the nullspace of $\mathbf{A}$. Therefore, if a $\mathbf{c}$ vector has $n - m$ components then the minimum number of constraints representing the bounded polyhedra is $n-m+1$, of which, $n-m$ constraints are due to the non-redundant inequalties in $\mathbf{Zc} \geq 0$ and one more due to the inequality constraint $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) which bounds $\mathbf{Zc} \geq 0$. The maximum possible number of vertices that can be obtained using $n - m + 1$ inequalities in $n - m$ variables is equal to $n - m + 1$ which is exactly the number of vertices that would correspond to a simplex. Hence, the number of non-redundant inequalities in $\mathbf{Zc} \geq 0$ corresponding to a square lattice is just enough to form a simplex. This explains our result.

3.2.4 Conclusion

We have shown the following for the system of frictionless hard disks on a square lattice:

1. The polytope corresponding to hard disks on a square lattice is a simplex. Each vertex of this simplex corresponds to a force network which can hold the system in mechanical equilibrium.

2. The number of vertices of the polytope scales as $2\sqrt{N}$, where, N is the number of disks/particles in the system.

3.3 Rhombic lattice

3.3.1 Introduction to the system

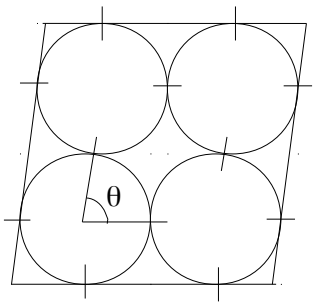

Figure 3.14: Smallest system on a rhombic lattice

Next, we study the polytope corresponding to a system of identical hard disks arranged on a rhombic lattice. We do this to find out the effect of relative arrangement of disks on the number of vertices of the polytope. We achieve this by keeping the coordination number same as that of square lattice which is 4. The smallest system size would correspond to 4 disks as shown in figure 3.14.

Consider a rhombic lattice characterised by an acute angle, θ, enclosed between two adjacent sides of the rhombus. The unknown contacts are marked

in figure 3.14. The number of contacts scale same as that for square lattices with system size shown in figure 3.15.

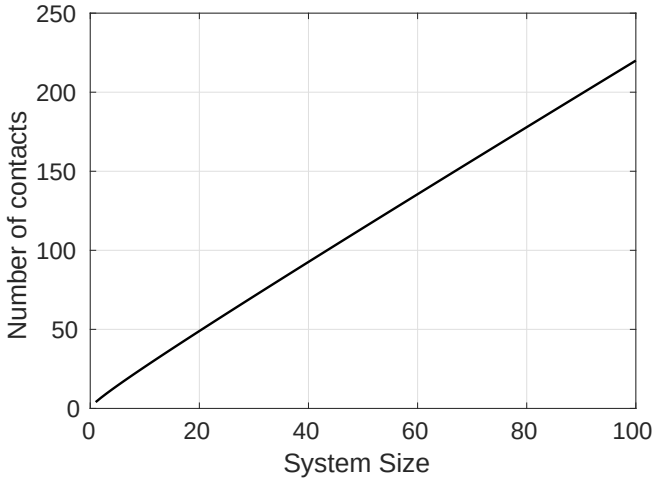

Figure 3.15: Number of contacts vs System size

If the contact matrix **A** is of the size $m \times n$ and rank r then dimensionality of the space of solutions to **Ax=0** is $n - r$. It was found empirically that $r = m$ for contact matrix of rhombic lattices. Since for a given number of particles the number of unknown contacts in rhombic lattice is same as that in a square lattice therefore the same two results hold even in the case of rhombic lattices which are listed below (N is number of disks in the system):

(i) The number of unknown contacts scales as $2(N + \sqrt{N})$.

(ii) The dimensionality of nullspace of **A** equals $2\sqrt{N}$.

3.3.2 Number of vertices vs System size

We will now take rhombic lattices with angles ranging from $\theta = 61 - degrees$ to $85 - degrees$ and see how the number of vertices scale with system size[7].

A rhombic lattice at $\theta = 60 - degrees$ develops extra contacts such that the coordination number of disks in the interior of the rhombic lattice becomes six, same as that of disks in the interior of equi-lateral triangular

[7]Note: The rhombic angle θ cannot be less than $60 - degrees$ because the disks would then overlap.

lattice. Hence, in order to probe if the coordination number is responsible for an exponential rise in the number of vertices of the polytopes it suffices to compare the system size dependence of the number of vertices corresponding to rhombic lattice at $\theta = 60 - degrees$ and $\theta > 60 - degrees$.

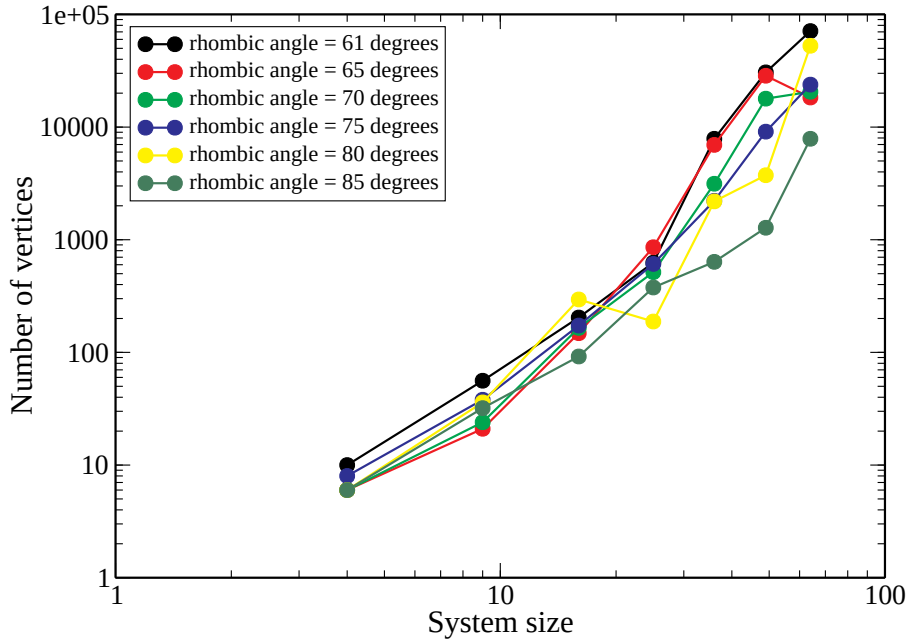

Figure 3.16: Number of vertices vs System size

We observe in figure 3.16, on a log-log scale, that the number of vertices generally increase with system size. The number of vertices for this system scale much slower than a exponential function of system size. The black curve corresponds to a rhombic lattice which is closest to the equi-triangular case (rhombic angle=$61 - degrees$) and could be temptingly close to a polynomial function. When the rhombic angle changes by just another degree and becomes=$60 - degrees$ the coordination number changes from 4 to 6 and for this we have already shown that the increase in number of vertices is exponential. Hence, we conclude that the transition to an increase in vertices with respect to system size to an exponential is caused due to a change in coordination number. Figure 3.16 also tells us that for a system of frictionless particles in 2-dimensions with coordination number of 4 the number of vertices could be bounded polynomially as a function of system size. Hence,

it may be feasible to find all the vertices of bounded polyhedra corresponding to large system sizes which have low coordination numbers.

3.3.3 Number of vertices vs Rhombic Angle

We will now show the effect of the rhombic angle θ on the number of vertices at two different system sizes.

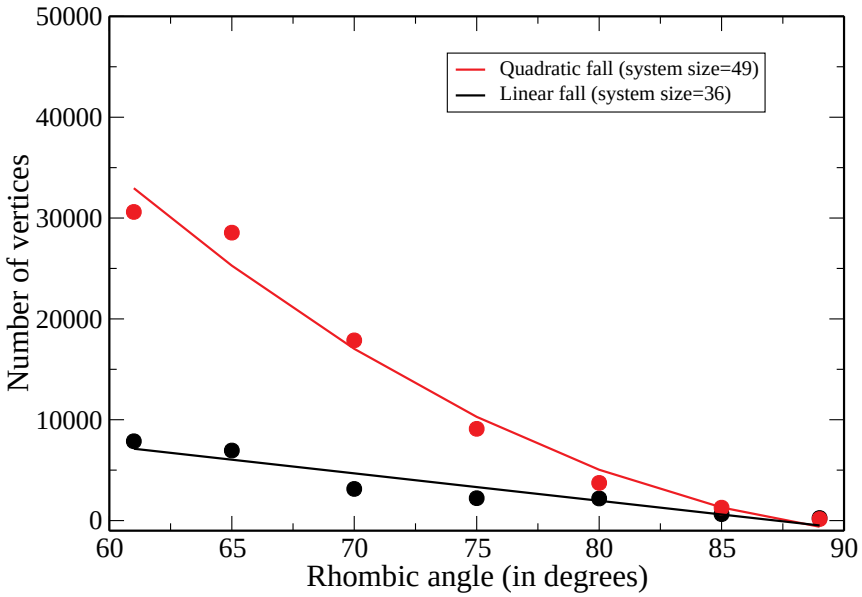

Figure 3.17: Number of vertices vs Rhombic angle

At a given system size we see that as θ approaches $90 - degrees$ the number of vertices converge to that of a square lattice. The number of vertices increase monotonically with decreasing θ.

3.3.4 Conclusion

We have shown the following for a system of frictionless hard disks on a rhombic lattice in 2-dimensions:

1. The number of vertices converge to that of a square lattice as θ approaches $90 - degrees$.

2. As the rhombic angle θ approaches $60 - degrees$ the number of vertices corresponding to a rhombic lattice is significantly different from that of the equilateral-triangular case. This difference is attributed to the difference in coordination number of 4 and 6. In rhombic lattice (coordination number =4) as θ approaches $60 - degrees$ the number of vertices can be bounded by a polynomial function of system size while exactly at $\theta = 60 - degrees$ this becomes exponential which corresponds to the equilateral-triangular case (coordination number =6).

3. For a given rhombic lattice (with $\theta < 90 - degrees$) the number of vertices generally increase with system size. The increase is more rapid for lattices tending towards the equilateral-triangular case.

Chapter 4

Summary and future directions

For a given contact geometry of a system of under-determined granular frictionless particles the force balance equations result into a system of linear equations of the form **Ax=0**. The matrix **A** is called the contact matrix which becomes a known quantity once the contact geometry is fixed. The vector **x** consist of magnitude of unknown forces which exist at contacts between particles. Any solution **x** of **Ax=0** can be written as a linear combination of the basis vectors of the null space of matrix **A**. We represent such an expansion of **x** by the relationship **x=Zc**, where, **Z** is a matrix which consist of column vectors which are orthogonal to each other and form a basis of the null space of **A**. If in addition the constraint of zero cohesion is put at each contact then all the components of the vector **x** must also in addition be constrained to be non-negative. Thus, the problem of solving for an unknown **x**, which satisfies componentwise non-negativity criteria and **Ax=0**, becomes a problem of solving a system of linear inequalities of the form **Zc** ≥ 0 for the unknown variable **c**.

We bound the polyhedra **Zc** ≥ 0 by imposing an inequality of the form $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) on **Zc** ≥ 0. We then identify all the vertices resulting from the intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$, whichever bounds **Zc** ≥ 0) and **Zc** ≥ 0. The knowledge of coordinates of these vertices is sufficient to construct any **c** consistent with **Zc** ≥ 0 and hence by the relationship, **x=Zc**, any force network that the system might admit. As pointed out through the

answer of *Question* 1 posed in section 2.1.5 the value of d in $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) is arbitrary and, since $\mathbf{x}$=$\mathbf{Zc}$, determines the maximum magnitude of contact forces existing in the granular system. And therefore, d sets the force scale in the granular system.

In order to determine the force network of an under-determined system one must generate points, say $\mathbf{c_1}, \mathbf{c_2}, .., \mathbf{c_p}$, which are spread uniformly in the space of intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$, whichever bounds $\mathbf{Zc} \geq 0$) and $\mathbf{Zc} \geq 0$ and then take an average of these points[1], which say is denoted by $\mathbf{c}_{avg}$, to generate, by the relationship $\mathbf{x}$=$\mathbf{Zc}_{avg}$, the force network for the under-determined system. This of course has not been demonstrated in this thesis and forms an important part of future directions of this project. The role of the extremal vertices of the intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) and $\mathbf{Zc} \geq 0$ is to ensure that the uniformly generated points $\mathbf{c_1}, \mathbf{c_2}, .., \mathbf{c_p}$ span over all the feasible space of solutions which is the intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) and $\mathbf{Zc} \geq 0$.

We have considered 2-dimensional frictionless disks on three simple kinds of lattice: Equi-triangular, rhombic and square. It was found that the number of vertices of the polyhedra obtained as an intersection of $\hat{e_*}^T\mathbf{c} \leq d$ and $\mathbf{Zc} \geq 0$ grow exponentially for the equi-triangular case, bounded polynomially for the rhombic case and goes as a square root for the square lattice as a function of system size. We show through comparison of our results of square and rhombic lattices that the exponential growth of vertices in the equi-triangular case is due to a higher coordination number (six). An upper bound on the number of vertices for equi-triangular systems also show exponential behaviour as a function of system size. While we have given concrete explanation for the results corresponding to square lattice case, the results for the rhombic case remain un-explained.

These calculations suggest that finding all the extremal vertices of the bounded polyhedra corresponding to granular systems which are densely packed might not be feasible. The vertex enumeration method as discussed

[1]This of course assumes that the corresponding force networks $\mathbf{x_1}, \mathbf{x_2}, .., \mathbf{x_p}$, which are solutions to $\mathbf{Ax}$=$\mathbf{0}$, obtained using the relationship $\mathbf{x}$=$\mathbf{Zc}$, are *equally likely* to exist in an under-determined granular system. The i_{th} component of $\mathbf{c}_{avg}$ will be equal to the average value of the i_{th} component of the vectors $\mathbf{c_1}, \mathbf{c_2}, .., \mathbf{c_p}$.

in the methods section can not find all the vertices in practical time if their number is exponentially large. Under such situations an approximation of the solution space, which is the intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$, whichever bounds $\mathbf{Zc} \geq 0$) and $\mathbf{Zc} \geq 0$, becomes necessary. We have demonstrated a method to do so. The method which we call a *box fit* method generates a subset of the extremal vertices of the region of intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) and $\mathbf{Zc} \geq 0$ in an attempt to approximate space of solutions corresponding to the intersection of $\hat{e_*}^T\mathbf{c} \leq d$ (or $\mathbf{c}_*^T\mathbf{c} \leq d$) and $\mathbf{Zc} \geq 0$ as closely as possible. We have demonstrated through volume calculations that the approximated space of solutions become better by increasing the number of *boxes*. The cost of finding such a subset of vertices has also been stated.

The box-fit method that we have suggested relies on finding an $\hat{e_*}$ or $\mathbf{c}_*$ so that an inequality of the form $\hat{e_*}^T\mathbf{c} \leq d$ or $\mathbf{c}_*^T\mathbf{c} \leq d$ could bound $\mathbf{Zc} \geq 0$ and an approximation of the resulting volume could be made. The computational cost of finding an $\hat{e_*}$ is high as it requires solving $2n$ LPPs and is known not to succeed for some $\mathbf{Zc} \geq 0$ corresponding to some systems considered in this thesis. On the contrary, finding a $\mathbf{c}_*$ is computationaly very cheap as it requires solving only *one* LPP and has succeeded in bounding $\mathbf{Zc} \geq 0$ for all the systems considered in the thesis. Thus, the general strategy for volume approximation is to use Approach II to bound $\mathbf{Zc} \geq 0$ and then, using the box-fit method, find multiple boxes to find a subset of vertices of the bounded polyhedra.

While our calculations have been performed for frictionless systems the same method can be used in finding the vertices (or a subset of vertices) of polyhedra corresponding to frictional granular systems. The method doesn't depend on if the granular system is composed of 2-dimensional or 3-dimensional particles.

Appendix

Here, we provide information regarding publicly available packages in 'C', called 'lrs' and 'redund' which have been used to compute the following:

1. Finding extremal vertices of a bounded polyhedra: Computed using 'lrs'.
2. Computation of the volume of a bounded polyhedra given by a list of vertices. Computed using 'lrs'.
3. Removal of redundant inequalities from the list of linear inequalities in $\mathbf{Zc} \geq 0$. Computed using 'redund'.

These packages can be found at the following locations:

lrs	author:	David Avis (avis@cs.mcgill.ca)
	ftp site:	http://cgm.cs.mcgill.ca/~avis/C/lrs.html
	directory:	Download/
	file name:	lrslib-061.tar.gz

redund	author:	David Avis (avis@cs.mcgill.ca)
	ftp site:	http://cgm.cs.mcgill.ca/~avis/C/lrs.html
	directory:	Download/
	file name:	lrslib-061.tar.gz

Installation and a guide to the usage of these libraries including the input file format for doing a vertex enumeration, volume computation and finding redundant inequalities can be found at: http://cgm.cs.mcgill.ca/~avis/C/lrslib/USERGUIDE.html#InstallationSection

The calculation involving minimization of linear functions subject to the linear inequality constraints were performed using the 'LINPROG' command in MATLAB. Similarly, the basis of the null space of contact matrices were obtained using the 'null' command in MATLAB.

Bibliography

[1] Nadukuru, Srinivasa S and Michalowski, Radoslaw L, "Arching in distribution of active load on retaining walls," *Journal of geotechnical and geoenvironmental engineering*, 2012.

[2] P.-G. de Gennes, "Granular matter: a tentative view," *Reviews of modern physics*, vol. 71, no. 2, p. S374, 1999.

[3] H. M. Jaeger, S. R. Nagel, and R. P. Behringer, "Granular solids, liquids, and gases," *Reviews of modern physics*, vol. 68, no. 4, p. 1259, 1996.

[4] R. Garcia-Rojo, H. Herrmann, and S. McNamara, "Powders and grains 2005," 2005.

[5] S. Ostojic, E. Somfai, and B. Nienhuis, "Scale invariance and universality of force networks in static granular matter," *Nature*, vol. 439, no. 7078, pp. 828–830, 2006.

[6] Strang, Gilbert, *Linear Algebra and Its Applications*, Hartcourt Brace Jovanovich College Publishers, (1988).

[7] Avis, David and Fukuda, Komei, "Reverse search for enumeration," *Discrete Applied Mathematics*, 1996.

[8] West, Douglas Brent, *Introduction to graph theory*, Prentice hall Upper Saddle River, NJ, 1996.

[9] H. Vinutha and S. Sastry, "Force networks and jamming in shear-deformed sphere packings," *Physical Review E*, 2019.

[10] Boyd, Stephen and Vandenberghe, Lieven, *Convex optimization*, Cambridge university press, 2004.

[11] Dantzig, George, *Linear programming and extensions*, Princeton university press, 2016.

[12] D. Avis, K. Fukuda, and S. Picozzi, "On canonical representations of convex polyhedra," *Mathematical Software, ICMS*, pp. 350–360, 2002.

[13] D. Avis, "A c implementation of the reverse search vertex enumeration algorithm (Computational Geometry and Discrete Geometry)," 1994.

[14] D. Avis and K. Fukuda, "A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra," *Discrete & Computational Geometry*, vol. 8, no. 3, pp. 295–313, 1992.

[15] R. G. Bland, "New finite pivoting rules for the simplex method," *Mathematics of Operations Research*, vol. 2, no. 2, pp. 103–107, 1977.

[16] B. Büeler, A. Enge, and K. Fukuda, "Exact volume computation for polytopes: a practical study," in *Polytopes combinatorics and computation*, pp. 131–154, Springer, 2000.

[17] Klee, Victor and Minty, George J, "How good is the simplex algorithm," *Washington Univ Seattle Dept Of Mathematics*, 1970.

[18] Dantzig, George B and Thapa, Mukund N, *Linear programming 1: introduction*, Springer Science & Business Media, 2006.

[19] Avis, David, "A revised implementation of the reverse search vertex enumeration algorithm," *Polytopes combinatorics and computation*, 2000.