Chapter 1

Force Tiling Algorithm

1.1 Basic Graph Theory Concepts

1.1.1 Graph

A graph is an ordered pair $\mathcal{G}(\mathcal{V}, \mathcal{E})$, which consists of a set of *vertices* or *nodes* $\mathcal{V}$ and a set $\mathcal{E}$ of *edges*. $\mathcal{E}$ is a subset of $\mathcal{V}^2$. Fig. 1.1 shows an example of a *planar graph*.

1.1.2 Undirected and Directed graph

An undirected graph is one in which edges have no orientation. The edge (i, j) is identical to the edge (j, i). A directed graph, on the other hand, has oriented edges and the edge i, j is not equivalent to (j, i). A network of grains connected to each other by some bonds is an example of undirected graph, the order of the grains do not matter in such a case. The graph shown in Fig. 1.1 is an undirected graph. A network of chemical reactions, where the forward rate of some reactions is different than the reverse rate, is an example of directed graph.

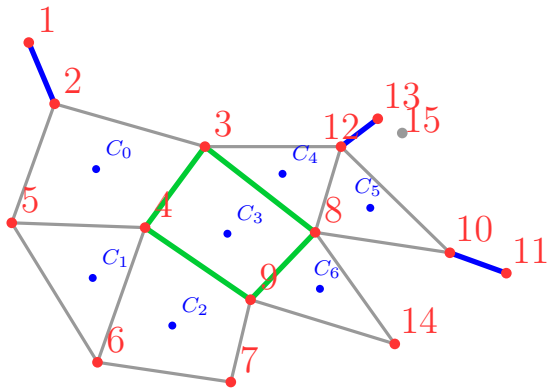


Figure 1.1: An example of an undirected planar graph. This graph conists of vertices $\mathcal{V} = \{1, 2, \ldots, 15\}$ and edges $\mathcal{E} = \{(1,2), (2,3), (2,5), \ldots\}$. It has three basic fundamental structures: (a) Isolated Vertex, (b) Filaments, and (c) Cycles or loops. Isolated vertex has no adjacent vertex; e.g. 15 in above figure (shown in grey). A cycle is a structure which, as the name suggests, closes on itself. For example, in the figure a cycle (3-4-9-8-3) is shown in green. Another cycle will be (3-4-6-7-9-8-12-3). The minimal cycle basis is the set $\{C_i,\ i = 0, 1, \ldots, 6\}$. A filament is defined as a structure which is neither cycle nor an isolated vertex. At least one vertex in the filaments has only one adjacent vertex. In the above figure, three filaments are shown in blue.

1.1.3 Planar Graph

In graph theory, a planar graph is a graph that can be embedded in the plane, i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints. In other words, it can be drawn in such a way that no edges cross each other. Fig. 1.1 shows an example of a planar graph.

1.1.4 Graph Structures and Minimum Cycle Basis

The structure of any undirected graph can be described completely through three basic structures. Namely, (a) Isolated vertices, (b) Cycles, and (c) Filaments. These are called graph primitives and the definitions can be found in Fig.1. A cycle is called simple cycle if it has no repeated vertices other than the starting and ending vertices. For example in the Fig.1.1 (3-4-9-8-3) is a simple cycle and (3-4-9-7-6-4-9-8-3) is not a simple cycle.

The set of all possible simple cycles found in a graph forms a vector space. This is called the binary cycle space. The binary cycle space is spanned by a set of linearly independent cycles. The linear independence is established through symmetric difference of cycles. Three cycles are linearly independent, if one of them cannot be written as the symmetric difference of the vertex set of the other two. For example, (3-4-9-8-3),(8-10-12-8), and (3-8-12) are linearly independent; (3-4-9-8-3),(3-8-12), and (3-4-9-8-12-3) is not. For a graph with n vertices and m edges, the dimension of the binary cycle space (BCS henceforth) is $m-n+1$. There can be multiple basis sets for the BCS. However there is a special set of cycles which is called minimal cycle basis (MCB). This set of cycles cannot be further decomposed into two or more cycles. Hence, MCB is much like the irreducible representation of a group. In the above figure, the set of cycles $\{C_i,\, i = 0, 1, \ldots, 6\}$ forms an MCB. Each cycle is defined as the smallest polygon enclosing that point. An example: C_3 is the cycle (3-4-9-8-3). The

cycles consisting the MCB is also called the *faces* of the graph.

1.2 Construction of Force Tiles

It is often convenient to represent any interacting system as a graph. A graph is described by a set of vertices which represent the constituent elements of the system and the edges connecting the vertices [1]. The edges may be weighted which indicates a nonuniform interaction throughout the system. Also, an edge may have a direction associated with it. In which case, the graph is directed. In a granular network, the grains are in contact with each other. This connectivity can be used to construct a network of the grains. The edges are the contact vectors and thus $R_{ij} = R_{ji}$; $\vec{R}_{ij}$ is the contact vector between i^{th} and j^{th} grain. Hence, the contact network - the real space network - can be represented by an undirected graph, as shown in Fig.1.2. This network does not contain any information about the forces through the contacts. Hence, we need to supply this information additionally. For a frictionless system, this is easy, as the forces are always functions of the contact vector. The detailed method of incorporating the force information is described in the following paragraph.

The additional information about the forces are taken care by the adjacency matrix of the network. The adjacency matrix provides the connectivity of the graph in a logical form. This is an $n \times n$ matrix where n is the number of vertices. If vertices i and j share an edge, then A_{ij} and A_{ji} are updated with the weight of the edge; A is the adjacency matrix. Otherwise, the corresponding elements are set to 0. However, for the present problem this scalar description of connectivity among elements is insufficient. Since the contacts of the granular network are force bearing, the nonzero elements of the adjacency matrix have to be the force vectors. That is, for a two dimensional frictionless network, each element of the adjacency matrix will contain a doublet containing the components of the force or, equivalently, the component of

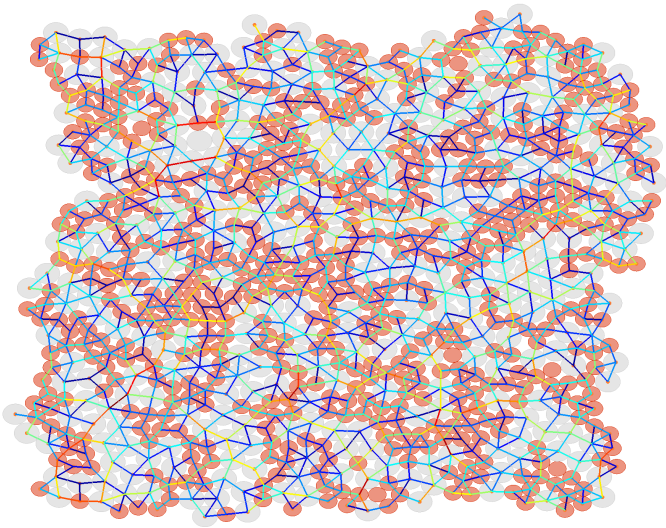


Figure 1.2: A typical contact network of the grains. Force magnitude through the edges are shown through color coding. The magnitude of the force increases from blue to red.

the contact vectors. This vector adjacency matrix along with the coordinates of the grain centers completely describe the real space network (RSN). However, for a frictional system, usually the force vectors can not be mapped to the contact vectors. This means that we'll need additional force information along with the adjacency matrix containing the contact vectors.

The force tile network (FTN) is a dual network of the RSN. Hence, FTN can be completely constructed from the available RSN information. The adjacency matrix for the FTN is obtained by finding the Minimum Cycle Basis of the RSN. Minimum Cycle Basis is the set of all the smallest polygons in the network.[1, 2] For example, in Fig.1.1, all the smallest polygons enclosing points $\{C_i,\, i = 0, 1, \ldots, 6\}$ form a Minimum Cycle Basis (MCB). If any two elements of the MCB share an edge, they are adjacent vertices in the dual network and share a bond between them (To avoid confusion, the dual space edges are referred to as bonds.). The magnitude of the bond is same as that of the edge and the direction is

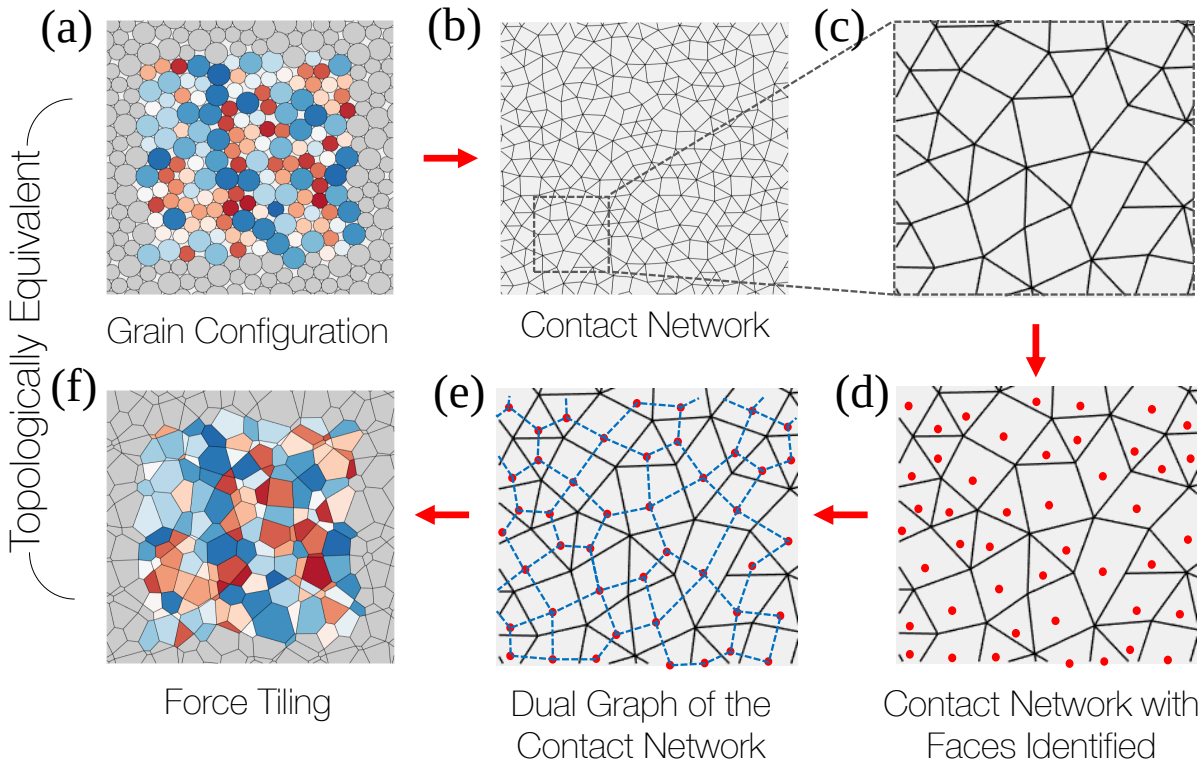


Figure 1.3: (Color online) A schematic of the force tiling algorithm. (a) A typical grain configuration. The colors are used to tag a grain and the corresponding force tile, otherwise they have no physical significance. (b) The real space contact network. (c) A portion of the real space contact network. (d) The faces (marked by the red dot) of this portion of the real space contact network as obtained from the MCB algorithm. (e) The dual graph (orange dashed line) topology obtained from the MCB. (f) The force tiling for this configuration. The color of the force tile is same as the color of the corresponding grain in (a).

perpendicular to that of the edge. Since there can be two perpendicular directions ($\frac{\pi}{2}$ or -$\frac{\pi}{2}$) for a given edge, extra care is taken to choose the correct perpendicular direction.

Once all the bonds are generated in the above mentioned way, the algorithm to construct the FTN can be implemented in the following way:

1. Start with any spatial configurations for the vertex coordinates.

$$\mathcal{V} = \{\mathbf{v}_i\}, \quad i = 1, 2, \ldots, n$$

2. Pick one vertex (master) and find its adjacent vertices (slaves). With the information stored in the adjacency matrix, reconstruct the master coordinate using each of the slaves. Since, each of the slaves generate a different coordinate for the master, we need to take the average of all the reconstructed coordinates as the new coordinate for the master. That is:

$$\mathbf{v}_i^{new} = \frac{1}{N_i} \sum_{\langle j \rangle} (\mathbf{v}_j + \mathbf{f}_{ij})$$
$$\mathbf{f}_{ij} = \mathbf{v}_i - \mathbf{v}_j$$

   Here N_i is the number of neighbors of the i^{th} vertex and $\langle j \rangle$ denotes the sum over neighbors. $\mathbf{f}_{ij}$ is the vector describing the bond which is obtained from the adjacency matrix of the FTN and it does not change.

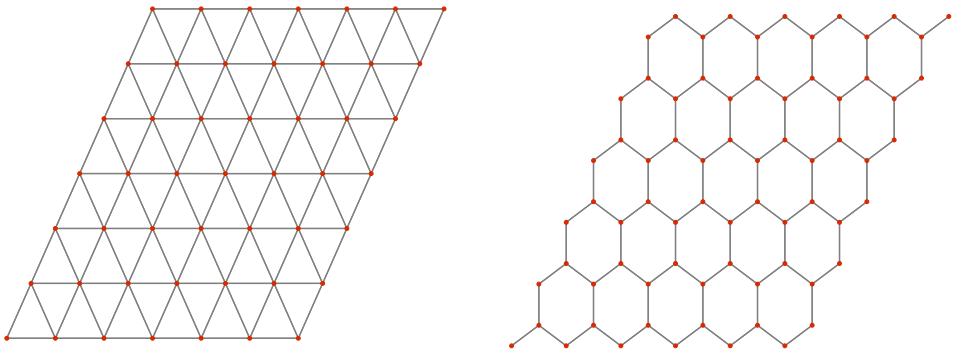
3. Do step 2 for all the vertices serially.

4. Iterate step 2 and 3 until the system converges into a stable structure. So far the choice of iteration steps is completely heuristic, but I wish to modify the algorithm to set some tolerance to end the algorithm.

1.3 Test Cases

As a sanity check for the algorithm, a regular triangular network of the grains is used. It is assumed that the force through each contact is same. The obtained dual network is the regular honeycomb network which is what we expect (Fig.1.4). Similar check with other regular networks has also been done. In all the cases we get the expected dual network.

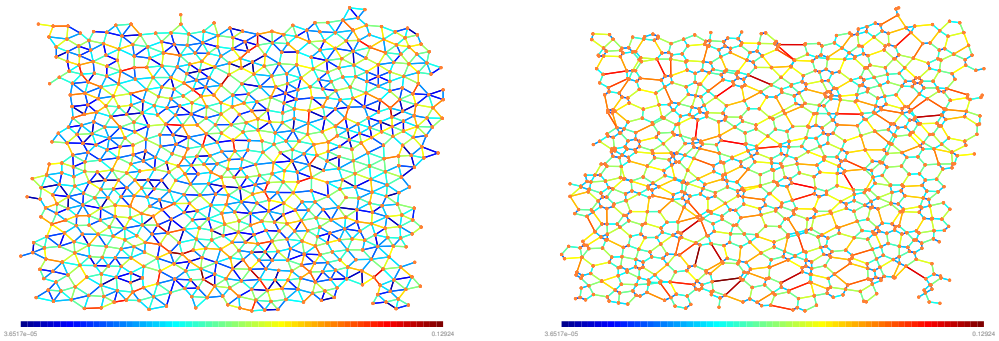
Next, the algorithm is applied to a system of frictionless grains. The grains are isotrop-



(a) Triangular network. (b) Force tile for the triangular network.

Figure 1.4: Regular Triangular Network.

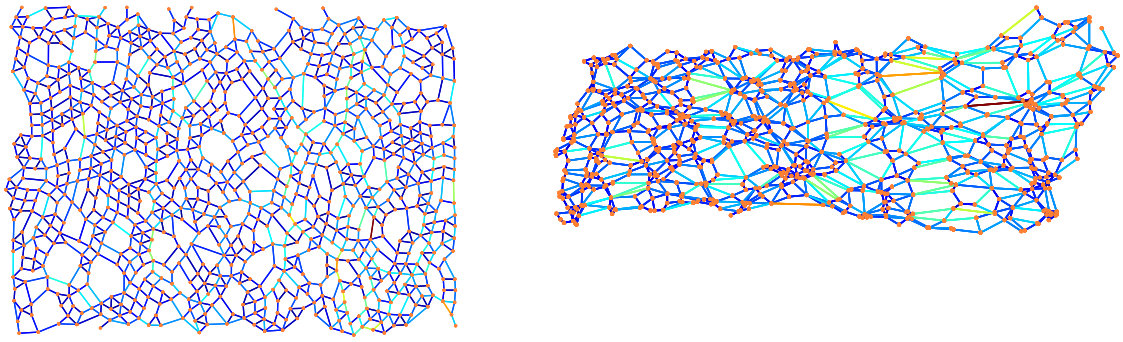
ically jammed and are above ϕ_J. In the following figure, a system of 750 grains is shown (Fig.1.6a). The contacts are color coded according to the magnitude of the force through the contact. The FTN for the system is shown alongside (Fig.1.5b). The color code is the same. Hence a direct qualitative comparison can be done. A more quantitative analysis of the correctness of the construction can be done via comparing the distances between the vertices in the FTN and the actual magnitude of the force through the bond. For the present example, the ratio between these two quantities is exactly one for all the bonds (result not shown). This is true for any completely force balanced network.



(a) The real space network (b) The force tile network

Figure 1.5: Force tile network for isotropically jammed system .

Similarly, the force tile for a shear jammed system can also be constructed. However, there are several challenges. First of all, it is an experimental system and hence susceptible to experimental error. Due to this reason, the smallest forces are indistinguishable from noises and require extra care. Also, Newton's third law was sacrificed to enforce the force balance condition for each grain during data collection. So, we had to take care of the Newton's third law which modified the experimentally obtained forces. Nonetheless, the calculated forces were within a factor of 2 for most of the cases and the major changes in the magnitude happened only for the smaller forces. Another problem was the existence of force bearing grains with only two contacts. We had to treat these grains as rattlers, and replace the grain with a contact between the other two grains carrying equal amount of force. These two assumptions enabled us to construct the experimental force tiles without any concave polygons. Following figure shows an example of the force tile obtained from the experimental data.



(a) The real space network (b) The force tile network

Figure 1.6: Force tile network for experimental shear jammed system .

1.4 Discussion

From an algorithmic point of view the first step, obtaining the RSN, is the easiest. The contacts and force at the granular level are easily obtained from the simulation and experimental data. Some amount of post-processing is required to obtain the RSN for experimental data, since due to experimental limitations, Newton's third law is sacrificed to achieve the force balance criterion for each individual grain. However, this shortcoming can be accommodated within the framework of the algorithm to construct FTN. Since, we already have the contact topology for the FTN, it ensures force balance by default: all the forces acting on a grain are forced to form a closed polygon. Hence, we can enforce Newton's third law on a contact, by taking the average of the $\vec{f}_{ij}$, and $\vec{f}_{ji}$. This allows us to construct an FTN, which is force-balanced and where Newton's third law is satisfied. The forces constructed this way are accurate within the experimental errors. The simulation data are cleaner (margin of error much smaller), and satisfy both force balance and Newton's third law. Hence, no post-processing is required to implement the scheme.

Bibliography

[1] Chris Godsil and Gordon F Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2013.

[2] David Eberly. The minimal cycle basis for a planar graph, May 2014.