A Linear Programming Algorithm to Test for Jamming in Hard-Sphere Packings

Aleksandar Donev,[1,2] Salvatore Torquato,[2,3,*] Frank H. Stillinger,[3] and Robert Connelly[4]

[1]Program in Applied and Computational Mathematics, Princeton University, *Princeton NJ 08540*

[2]Princeton Materials Institute, Princeton University, *Princeton NJ 08540*

[3]Department of Chemistry, Princeton University, *Princeton NJ 08540*

[4]Department of Mathematics, Cornell University, *Ithaca NY 14853*

(Dated: 5th December 2002)

Abstract

Jamming in hard-particle packings has been the subject of considerable interest in recent years. In a paper by Torquato and Stillinger [J.Phys.Chem. B, 105 (2001)], a classification scheme of jammed packings into hierarchical categories of *locally*, *collectively* and *strictly jammed* configurations has been proposed. They suggest that these jamming categories can be tested using numerical algorithms that analyze an equivalent contact network of the packing under applied displacements, but leave the design of such algorithms as a future task. In this work we present a *rigorous* and *efficient* algorithm to assess whether a hard-sphere packing is jammed according to the afformentioned categories. The algorithm is based on linear programming and is applicable to regular as well as random packings of finite size with hard-wall and periodic boundary conditions. If the packing is not jammed, the algorithm yields representative multi-particle unjamming motions. We have implemented this algorithm and applied it to ordered lattices as well as random packings of disks and spheres under periodic boundary conditions. Some representative results for ordered and disordered packings are given, but more applications are anticipated for the future. One important and interesting result is that the random packings that we tested were strictly jammed in three dimensions, but *not* in two dimensions. Numerous interactive visualization models are provided on the authors' webpage.

*Electronic address: `torquato@electron.princeton.edu`

I. INTRODUCTION

Packings of hard particles interacting only with infinite repulsive pairwise forces on contact are applicable as models of complex manybody systems because repulsive interactions are the primary factor in determining their structure. Hard-particle packings are therefore widely used as simple models for granular media [1, 2], glasses [3], liquids [4], and other random media [5], to mention a few examples. Furthermore, hard-particle packings, and especially hard-sphere packings, have inspired mathematicians and been the source of numerous challenging (many still open) theoretical problems [6]. Because the hard-sphere model is characterized by a idealized repulsive potential, one can be precise about the important concept of "jamming," a term which is used in a different sense in the granular media community. Modeling of realistic granular packings requires inclusion of friction, adhesion, particle deformability, etc., and, by definition, hard-sphere systems do not include these effects. In the present work, hard-sphere jamming is presented from a rigorous perspective which focuses on the *geometry* of the *final* packed states.

There are still many important and challenging questions open even for the simplest type of hard-particle packings, i.e., monodisperse packing of smooth perfectly impenetrable spheres. One category of open problems pertains to the enumeration and classification of disordered disk and sphere packings, such as the precise identification and quantitative description of the maximally random jammed (MRJ) state [7], which has supplanted the ill-defined "random close packed" state. Others pertain to the study of ordered systems and finding packing structures with extremal properties, such as the lowest or highest (for polydisperse packings) density jammed disk or sphere packings, for the various jamming categories described below [8, 9]. Numerical algorithms have long been the primary tool for studying random packings quantitatively. In this work we take an important step toward future studies aimed at answering the challenging questions posed above by designing tools for algorithmic assesment of the jamming category of finite packings.

In the first part of this paper, we present the conceptual theoretical framework underlying this work. Specifically, we review and expand on the hierarchical classification scheme for jammed packings into locally, collectively and strictly jammed packings proposed in Ref. [10]. In the second part, we present a randomized linear programming algorithm for finding unjamming motions within the approximation of small displacements, focusing on periodic boundary conditions. Finally, in the third part we apply the algorithm to monodisperse packings under periodic boundary conditions, and present some representative but non-exhaustive results for several periodic ordered lattice packings as well as random packings obtained via the Lubachevsky-Stillinger packing algorithm [11].

Ideas used here are drawn heavily from literature in the mathematical community ([12–15], etc.), and these have only recently percolated into the granular materials community ([15–17], etc.). A separate paper [18] will attempt to give a unified and more rigorous presentation of the mathematical ideas underlying the concepts of stability, rigidity and jamming in sphere packings. Nonetheless, some mathematical preliminaries are given here, and more technical points are deferred to the Appendix.

A. Jamming in Hard-Sphere Packings

The physical intuition behind the word jamming is strong: It imparts a feeling of being frozen in a given configuration. Two main approaches can be taken to defining jamming, kinematic and static. In the **kinematic** approach, one considers the motion of particles away from their current positions, and this approach is for example relevant to the study of flow in granular media[28]. The term *jammed* seems most appropriate here. In the **static**' approach, one considers the mechanical properties of the packing and its ability to resist external forces[29]. The term *rigid* is often used among physicists in relation to such considerations.

However, due to the correspondence between kinematic and static properties, i.e. strains and stresses, these two different views are largely equivalent. A more thorough discussion of this duality is delayed to a later work [18], but is touched upon in section V B. In this paper we largely adopt a kinematic approach, as we focus on the geometry of packings, but the reader should bear in mind the inherent ties to static approaches.

B. Three Jamming Categories

First we repeat, with slight modifications as in Ref. [19], the definitions of several hierarchical jamming categories as taken from Ref. [10], and then make them mathematically specific and rigorous for several different types of sphere packings:

A finite system of spheres is:

Locally jammed: Each particle in the system is locally trapped by its neighbors, i.e., it cannot be translated while fixing the positions of all other particles.

Collectively jammed: Any locally jammed configuration in which no subset of particles can simultaneously be displaced so that its members move out of contact with one another and with the remainder set. An equivalent definition is to state that all finite subsets of particles be trapped by their neighbors.

Strictly jammed: Any collectively jammed configuration that disallows all globally uniform volume-nonincreasing deformations of the system boundary. Note the similarity with collective jamming but with the additional provision of a deforming boundary. This difference and the physical motivations behind it should become clearer in section IV C.

Observe that these are ordered *hierarchically*, with local being a prerequisite for collective and similarly collective being a prerequisite for strict jamming.

The precise meaning of some of the concepts used in these definitions, such as unjamming, boundary and its deformation, depends on the type of packing one considers and on the particular problem at hand, and we next make these specializations more rigorous for specific types of packings of interest. Moreover, we should point out that these do not exhaust all possibilities and various intricacies can arise, especially when considering infinite packings, to be discussed further in Ref. [18].

It should be mentioned that jammed random particle packings produced experimentally or in simulations typically contain a small population of "rattlers", i.e., particles trapped in a cage of jammed neighbours but free to move within the cage. For present purposes we shall assume that these have been removed before considering the (possibly) jammed remainder (section IV A). This idea of excluding rattlers can be further extended to "rattling clusters" of particles, i.e., groups of particles that can be displaced collectively even though the remainder of the packing is jammed. In fact, one can consider any packing which has a jammed subpacking (collectively or strictly, as defined above, with *identical* boundary conditions) to be jammed. The physical meaning and mathematical basis for such a modified approach is more evident from the static perspective, but this aspect will not be discussed in this work. We are currently investigating the application of mathematical programming techniques to identifying the set of jammed subpackings of a given packing.

C. Unjamming Motions

Before discussing the algorithms and related issues, it is important to specify exactly what we mean by unjamming. First, it is helpful to define some terms. Capitalized bold letters will be used to denote dN-dimensional vectors which correspond to the d-dimensional vectors of all N of the particles. A **sphere packing** is a collection of spheres in Euclidian d-dimensional space $\Re^d$ such that the interiors of no two spheres overlap. Here we focus on monodisperse systems, where all spheres have diameter $D = 2R$, but most of the results generalize to polydisperse systems as well. A packing $P(\mathbf{R})$ of N spheres is characterized by the arrangement of the sphere centers $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$, called *configuration*:

$$P(\mathbf{R}) = \left\{ \mathbf{r}_i \in \Re^d, \;\; i = 1, \ldots, N \; : \; \|\mathbf{r}_i - \mathbf{r}_j\| \geq D \; \forall j \neq i \right\}$$

The natural physical definition[30] of what it means to unjam a sphere packing is provided by looking at ways to move the spheres from their current configuration. This leads us to the definition: An **unjamming motion** $\Delta\mathbf{R}(t)$, where t is a time-like parameter, $t \in [0, 1]$, is a *continuous* displacement of the spheres from their current position along the path $\mathbf{R}+\Delta\mathbf{R}(t)$, starting from the current configuration, $\Delta\mathbf{R}(0) = 0$, and ending at the final configuration $\mathbf{R} + \Delta\mathbf{R}(1)$, while observing all relevant constraints along the way, such that some of the contacting spheres lose contact with each other for $t > 0$. This means that impenetrability and any other particular (boundary) conditions must be observed, i.e. $P(\mathbf{R} + \Delta\mathbf{R}(t))$ is a *valid* packing for all $t \in [0, 1]$. If such an unjamming motion does not exist, we say that the packing is *jammed*. By changing the (boundary) constraints we get different categories of jamming, such as local, collective and strict. It can be shown (see references in Ref. [14]) that an equivalent definition[31] is to say that a packing is jammed if it is isolated in the allowed configuration space, i.e., there is no valid packing within some (possibly small) finite region around $\mathbf{R}$ that is not equivalent (congruent) to $P(\mathbf{R})$. We mention this because it is important to understand that although we use a kinematic description based on motion through time, which imparts a feeling of dynamics to most physicists, a perfectly equivalent static view is possible. Therefore, henceforth special consideration will be given to the final displacement $\Delta\mathbf{R}(1)$, so that we will most often just write $\Delta\mathbf{R} = \Delta\mathbf{R}(1)$.

It is also useful to know that we need to only consider *analytic* $\Delta\mathbf{R}(t)$, which gives us the ability to focus only on derivatives of $\Delta\mathbf{R}(t)$. Furthermore, it is a simple yet fundamental fact that we only need to consider first derivatives $\mathbf{V} = \frac{d}{dt}\Delta\mathbf{R}(t)$, which can be thought of as "velocities", and then simply move the spheres in the directions $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_2)$ to obtain an unjamming motion $\Delta\mathbf{R}(t) = \mathbf{V}t$. The formal statement is that *a packing is rigid if and only if it is infinitesimally rigid*, see Refs. [12, 18]. A sphere packing that is not jammed[32] can be unjammed by giving the spheres velocities $\mathbf{V}$ such that no two contacting spheres i and j, $\|\mathbf{r}_i - \mathbf{r}_j\| = D$, have a relative speed $v_{i,j}$ toward each other:

$$v_{i,j} = (\mathbf{v}_i - \mathbf{v}_j)^T \mathbf{u}_{i,j} \leq 0 \tag{1}$$

where $\mathbf{u}_{ij} = \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{r}_j\|}$is the unit vector connecting the two spheres[33]. Of course, some special and trivial cases like rigid body translations ($\mathbf{V} =$ constant) need to be excluded since they do not really change the configuration of the system.

In this paper we will plot unjamming motions as "velocity" fields, and occasionally supplement such illustrations with a sequence of frames from $t = 0$ to $t = 1$ showing the unjamming process. Note that the lengths of the vectors in the velocity fields have been scaled to aid in better visualization. For the sake of clear visualization, only two-dimensional examples will be used. But all of the techniques described here are fully applicable to three-dimensional packings as well. Interactive Virtual Reality Modeling Language (`VRML`) animations which are very useful in getting an intuitive feeling for unjamming mechanisms in sphere packings can be viewed from `Windows` platforms on our webpage (see Ref. [21]).

II. BOUNDARY CONDITIONS

As previously mentioned, the boundary conditions imposed on a given packing are very important, especially in the case of strict jamming. Here we consider two main types of packings, depending on the boundary conditions.

Hard-wall boundaries The packing $P(\mathbf{R})$ is placed in an impenetrable concave ([12]) *hard-wall container* K. Figure 1 shows that the honeycomb lattice can be unjammed inside a certain hard-wall container. Often we will make an effective container out of N_f *fixed spheres* whose positions cannot change. This is because it is often hard to fit a packing into a simple container such as a square box, while it is easy to surround it with other fixed spheres, particularly if a periodic lattice is used to generate the packing. Specifically, one can take a finite sub-packing of an infinite periodic packing and freeze the rest of the spheres, thus effectively making a container for the sub-packing. An example is depicted in Fig. 2.

Periodic boundaries Periodic boundary conditions are often used to emulate infinite systems, and they fit the algorithmic framework of this work very nicely. A periodic packing $\widehat{P}(\mathbf{R})$ is generated by replicating a finite *generating packing* $P(\widehat{\mathbf{R}})$ on a lattice $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_d\}$, where $\boldsymbol{\lambda}_i$ are linearly independent *lattice vectors* and d is the spatial dimensionality. So the positions of the spheres are generated by,

$$\mathbf{r}_{\widehat{i}(\mathbf{n}_c)} = \widehat{\mathbf{r}}_i + \boldsymbol{\Lambda}\mathbf{n}_c \text{ and } \mathbf{n}_c \text{ is integer, } \mathbf{n}_c \in Z^d$$

where we think of $\boldsymbol{\Lambda}$ as a matrix with d^2 elements having the lattice vectors as columns and $\mathbf{n}_c$ is the number of replications of the unit cell along each basis direction. The sphere $\widehat{i}\,(\mathbf{n}_c)$ is the familiar *image sphere* of the *original sphere* $i = \widehat{i}\,(0)$, and of course for the impenetrability condition only the nearest image matters. For true periodic boundary conditions, we wrap the infinite periodic packing $\widehat{P}(\mathbf{R})$ around a flat torus, i.e., we ask that whatever happens to a sphere i also happens to all of the image spheres $\widehat{i}\,(\mathbf{n}_c)$, with the additional provision that the lattice may also change by $\Delta\boldsymbol{\Lambda}$:

$$\Delta\mathbf{r}_{\widehat{i}(\mathbf{n}_c)} = \Delta\widehat{\mathbf{r}}_i + (\Delta\boldsymbol{\Lambda})\mathbf{n}_c \tag{2}$$

A. Using Simple Lattices to Generate Packings

Simple familiar lattices (with a basis) such as the triangular, honeycomb, Kagomé and square in two dimensions, or the simple cubic (SC), body-centered cubic (BCC), face-centered cubic (FCC) and hexagonal-close packed (HCP) in three dimensions, can be used to create a (possibly large) packing taking a subsystem of size $\mathbf{N}_c$ unit cells along each dimension from the infinite lattice packing. The properties of the resulting system can be studied with the tools developed here, provided that we restrict ourselves to finite $\mathbf{N}_c$. Moreover, it is important to specify which lattice vectors are to be used. We will usually take them to be primitive vectors, but sometimes it will be more convenient to use conventional ones (especially for variations on the cubic lattice).

For hard-wall boundary conditions, we can take an infinite packing generated by these simple lattices and then freeze all but the spheres inside the window of $\mathbf{N}_c$ unit cells, thus effectively obtaining a hard-wall container. Figure 2 illustrates an unjamming motion for the honeycomb lattice under these conditions.

For periodic boundary conditions, the generator $P(\widehat{\mathbf{R}})$ can itself be generated using $\mathbf{N}_c$ unit cells of a simple lattice. In this case the lattice $\mathbf{\Lambda}$ is a *sub-lattice* of the underlying (primitive) lattice $\widetilde{\mathbf{\Lambda}}$, i.e., $\mathbf{\Lambda} = \widetilde{\mathbf{\Lambda}}\text{Diag}\{\mathbf{N}_c\}$. This is not only a convenient way to generate simple finite periodic packings, but it is in general what is meant when one asks, for example, to analyze the jamming properties of the Kagomé lattice under periodic or hard-wall boundary conditions. Figure 3 shows a periodic unjamming motion for the Kagomé lattice. Notice though that the jamming properties one finds depend on how many neighboring unit cells $\mathbf{N}_c$ are used as the "base" region (i.e., the generating packing), and therefore, we will usually specify this number explicitly. Some properties are independent of $\mathbf{N}_c$, and tailored mathematical analysis can be used to show this [13, 22]. More systematic approaches based on Bloch wave (Fourier) decompositions of the set of feasible motions are being investigated for periodic packings. We will not consider these issues in detail here, but rather focus on algorithmic approaches tailored for finite and fixed systems (i.e., $\mathbf{N}_c$ is fixed and finite), and postpone the rest of the discussion to Ref. [18].

III. LINEAR PROGRAMMING ALGORITHM TO TEST FOR JAMMING

Given a sphere packing, we would often like to test whether it is jammed according to each of the categories given above, and if it is not, find one or several unjamming motions $\Delta\mathbf{R}(t)$. We now describe a simple algorithm to do this that is exact for *gap-less packings*, i.e., packings where neighboring spheres touch exactly, and for which the definitions given earlier apply directly. However, in practice, we would also like to be able to study *packings with small gaps*, such as produced by various heuristic compression schemes like the Lubachevsky-Stillinger algorithm [11]. In this case the meaning of unjamming needs to be modified so as to fit physical intuition. We do this in such a way as to also maintain the applicability of our efficient randomized linear programming algorithm using what Roux [16] calls the *approximation of small displacements* (ASD).

A. Approximation of Small Displacements

As already explained, an unjamming motion for a sphere packing can be obtained by giving the spheres suitable velocities, such that neighboring spheres do not approach each other. Here we focus on the case when $\Delta\mathbf{R}(t) = \mathbf{V}t + \mathcal{O}(t^2)$ are small finite displacements from the current configuration. Therefore, we will drop the time designation and just use $\Delta\mathbf{R}$ for the displacements from the current configuration $\mathbf{R}$ to the new configuration $\widetilde{\mathbf{R}} = \mathbf{R} + \Delta\mathbf{R}$.

In this ASD approximation, we can linearize the impenetrability constraints by expanding to first order in $\Delta\mathbf{R}$,

$$\|\widetilde{\mathbf{r}}_i - \widetilde{\mathbf{r}}_j\| = \|(\mathbf{r}_i - \mathbf{r}_j) + (\Delta\mathbf{r}_i - \Delta\mathbf{r}_j)\| \geq D \tag{3}$$

to get the condition for the existence of a *feasible displacement* $\Delta\mathbf{R}$,

$$(\Delta\mathbf{r}_i - \Delta\mathbf{r}_j)^T \mathbf{u}_{i,j} \leq \Delta l_{i,j} \text{ for all } \{i,j\} \tag{4}$$

where $\{i,j\}$ represents a *potential contact* between nearby spheres i and j, $\Delta l_{i,j} = \|\mathbf{r}_i - \mathbf{r}_j\| - D$ is the *interparticle gap* (called *interstice* in Ref. [16]), and $\mathbf{u}_{ij} = \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{r}_j\|}$ is the unit vector along the direction of the contact $\{i,j\}$. For a gap-less packing, we have $\Delta l = 0$ and the condition (4) reduces to (1), which as we explained along with the conditions $\Delta\mathbf{R} \neq 0$ and $\Delta\mathbf{R} \neq$ const. is an *exact* condition for the existence of an unjamming motion $\Delta\mathbf{R}$. For packings with finite but small gaps though, condition (4) is only a first-order approximation.

If we had not expanded the impenetrability constraints to first order we would have obtained the non-linear version of Eq. (4):

$$(\Delta\mathbf{r}_i - \Delta\mathbf{r}_j)^T \mathbf{u}_{i,j} - \frac{\|\Delta\mathbf{r}_i - \Delta\mathbf{r}_j\|^2}{2\,\|\mathbf{r}_i - \mathbf{r}_j\|} \leq \Delta l_{i,j}\left(1 - \frac{\Delta l_{i,j}}{2\,\|\mathbf{r}_i - \mathbf{r}_j\|}\right) \text{ for all } \{i,j\} \tag{5}$$

Notice that any displacement feasible under the linearized constraints (4) will also be feasible under the full nonlinear impenetrability constraints (5). Comparison of (4) and (5) also suggests that using the right-hand side of (5) instead of just $\Delta l_{i,j}$ is more appropriate when large gaps are present, and that the linearized constraints become too strict as the magnitude of the displacements becomes comparable to the size of the particles $\|\Delta\mathbf{r}_i - \Delta\mathbf{r}_j\| \approx D$. The complicated issue of how well the ASD approximation works when the gaps are not small enough is illustrated in Fig. 4.

Notice that we only need to consider potential contacts $\{i, j\}$ between nearby, and not all pairs, of spheres; that is we only consider a contact if

$$\|\mathbf{r}_i - \mathbf{r}_j\| \leq (1 + \delta)\, D$$

where δ is some tolerance for how large we are willing to allow the representative $\|\Delta\mathbf{r}\|$ to be. The larger this tolerance, the more possible particle contacts we will add to our constraints, and thus the more computational effort we need. Also, the ASD approximation becomes poorer as we allow larger displacements, and as the discussion above indicates, we are including more redundant and/or stricter-then-necessary linearized impenetrability constraints. Choosing a very small tolerance makes it impossible to treat systems with moderately large interparticle gaps (say of the order of $\delta = 10\%$), since crucial constraints which become relevant as soon as the magnitude of the displacements becomes comparable to δD are omitted. The general rule is that the contacts of each sphere with *all* spheres in (only) its first coordination shell should be included, and of course physical intuition and close examination of the results are very helpful.

By putting the $\mathbf{u}_{ij}$'s as columns in a matrix of dimension $[Nd \times N_e]$, where N_e is the number of contacts in the contact network, we get the important **rigidity matrix** [34] of the packing $\mathbf{A}$. This matrix is sparse and has two blocks of d non-zero entries in the column corresponding to the particle contact $\{i, j\}$, namely, $\mathbf{u}_{ij}$ in the block row corresponding to particle i and $-\mathbf{u}_{ij}$ in the block row corresponding to particle j. Represented schematically:

$$\mathbf{A} = \begin{array}{c} \\ \\ i \rightarrow \\ \\ j \rightarrow \\ \\ \end{array} \begin{array}{c} \{i,j\} \\ \downarrow \\ \left[\begin{array}{c} \vdots \\ \mathbf{u}_{ij} \\ \vdots \\ -\mathbf{u}_{ij} \\ \vdots \end{array}\right] \end{array}$$

For example, for the four-disk packing shown in Fig. 4, and with the numbering of the disks depicted in Fig. 5, we have the following rigidity matrix:

$$\mathbf{A} = \begin{array}{c} \\ D_1 \\ D_2 \\ D_3 \\ D_4 \end{array} \begin{array}{c} \begin{array}{ccc} E_{12} & E_{13} & E_{14} \end{array} \\ \begin{bmatrix} \mathbf{u}_{12} & \mathbf{u}_{13} & \mathbf{u}_{14} \\ -\mathbf{u}_{12} & & \\ & -\mathbf{u}_{13} & \\ & & -\mathbf{u}_{14} \end{bmatrix} \end{array}$$

Using this matrix, we can rewrite the linearized impenetrability constraints as a simple system of linear inequality constraints:

$$\mathbf{A}^T \Delta\mathbf{R} \leq \Delta\mathbf{l} \tag{6}$$

The set of contacts $\{i, j\}$ that we include in (4) form the **contact network** of the packing, and they correspond to a subclass of the class of fascinating objects called *tensegrity frameworks*, namely *strut frameworks* (see Ref. [14] for details, and also [2] for a treatment of more general packings). Figure 6 shows a small random packing with relatively large gaps and the associated contact network.

1. Boundary conditions

Handling different boundary conditions within the above formulation is easy. For example, for usual periodic conditions, handling the boundaries merely amounts to adding a few columns to the rigidity matrix $\mathbf{A}$ with $\mathbf{u}_{i,\widehat{j}(\mathbf{n}_c)} = \frac{\mathbf{r}_{\widehat{j}(\mathbf{n}_c)} - \mathbf{r}_i}{\left\|\mathbf{r}_{\widehat{j}(\mathbf{n}_c)} - \mathbf{r}_i\right\|}$ for all images $\widehat{j}(\mathbf{n}_c)$ which have contacts with one of the original spheres i. These columns correspond to the periodic contacts wrapping the packing around the torus.

For hard-wall boundaries, we would add a potential particle contact to the contact network from each sphere close to a wall to the closest point on the wall, and fix this endpoint. Such fixed points of contact and fixed spheres j, called *fixed nodes* in tensegrity terminology, are simply handled by transferring the corresponding term $\Delta\mathbf{r}_j^T \mathbf{u}_{i,j}$ to the right-hand side of the constraints in (6).

B. Finding Unjamming Motions

We are now ready to explain how one can find unjamming motions for a given packing, if such exist. But first, we need to refine our definition of an unjamming motion to allow for the study of packings with finite gaps.

1. *Unjamming Motions Revisited:* Dealing with Finite Gaps

The problem with packings with small gaps is that according to our previous definition of an unjamming motion in section I C, such packings will never be jammed, since it will always be possible to move some of the spheres at distances comparable to the sizes of the interparticle gaps so that they lose contact with all or most of their neighbors. Clearly we wish to only consider the possibility of displacing some of the spheres such that *considerable* gaps appear, larger then some threshold value $\Delta l_{\text{large}} \gg \overline{\Delta l}$, where $\overline{\Delta l}$ is a measure of the magnitude of the interparticle gaps. Therefore, we have the modified definition: An **unjamming motion** $\Delta\mathbf{R}(t)$, $t \in [0, 1]$, is a *continuous* displacement of the spheres from their current position along the path $\mathbf{R} + \Delta\mathbf{R}(t)$, starting from the current configuration, $\Delta\mathbf{R}(0) = 0$, and observing all relevant constraints along the way, such that some of the spheres lose contact with each other in the final configuration $\mathbf{R} + \Delta\mathbf{R}(1)$ by more then a given Δl_{large}. Alternatively, we could ask that some sphere displace by more than $\Delta r_{\text{large}} \gg \overline{\Delta l}$. Again, we exclude any trivial rigid-body type motions such as a uniform translation of the spheres from consideration.

The problem of whether such an unjamming motion exists and how to find one if it does will become mathematically more precise and interesting if we take the case $\Delta l_{\text{large}} \gg D$. But this problem is extremely complex due to high non-linearity of the impenetrability constraints and we will make no attempt to solve it. It also is not clear that this would be of interest to physicists. Instead, we focus our attention on the case when Δl_{large} is small enough to apply with a reasonable degree of accuracy the approximation of small displacements, but large enough compared to the interparticle gaps so that the exact value is really irrelevant.

Under the ASD, we need only worry about linear displacements from the current configuration, $\Delta\mathbf{R}(t) = \mathbf{V}t$, and so we can focus on $\Delta\mathbf{R} \equiv \Delta\mathbf{R}(1)$. Since here we are really focusing only on $\Delta\mathbf{R} \equiv \Delta\mathbf{R}(1)$, we should change terminology and call Δr an **unjamming displacement**. However, to emphasize the fact that there is a way to continuously move the spheres to achieve this displacement, and also that we can always scale *down* the magnitude of an allowed $\Delta\mathbf{R}$ arbitrarily, we continue to say unjamming motion. Thus, finding an unjamming motion simply reduces to the problem of *feasibility and boundedness* of the solution set of a linear system of inequalities,

$$\mathbf{A}^T\Delta\mathbf{R} \leq \Delta\mathbf{l} \qquad \text{for impenetrability} \tag{7}$$

$$\exists i \text{ such that } \quad \|\Delta\mathbf{r}_i\| \geq \Delta r_{\text{large}} \gg \overline{\Delta l} \quad \text{for unjamming} \tag{8}$$

where we can exclude trivial displacements such as uniform translations by adding additional constraints.

2. Randomized Linear Programming (LP) Algorithm

The question of whether a packing is jammed, i.e., whether the system (8) is feasible, can be answered rigorously by using standard linear programming techniques, as described in Appendix A. If a packing is jammed, then this is enough. But for packings which are not jammed, it is really more useful to obtain a *representative* collection of unjamming motions. A random collection of such unjamming motions is most interesting, and can be obtained easily by solving several linear programs with a random cost vector.

We adopt such a **randomized LP algorithm** to testing unjamming [i.e. studying (8)], namely, we solve several instances of the following LP in the ***displacement formulation***:

$$\max_{\Delta\mathbf{R}} \mathbf{b}^T\Delta\mathbf{R} \quad \text{for virtual work} \tag{9}$$

$$\text{such that} \quad \mathbf{A}^T\Delta\mathbf{R} \leq \Delta\mathbf{l} \quad \text{for impenetrability} \tag{10}$$

$$|\Delta\mathbf{R}| \leq \Delta R_{\max} \quad \text{for boundedness} \tag{11}$$

for *random loads* $\mathbf{b}$, where $\Delta R_{\max} \gg \overline{\Delta l}$ is used to prevent unbounded solutions and thus improve numerical behavior. The physical interpretation of $\mathbf{b}$ as an external load will be elucidated in section V B. Trivial solutions, such as uniform translations of the packing $\Delta\mathbf{R} =$ const. for periodic boundary conditions, can be eliminated *a posteriori*, for example by reducing $\Delta\mathbf{R}$ to zero mean displacement. Alternatively, trivial motions can be handled by introducing extra constraints in (10), for example, by fixing the position of one of the spheres. Finally, trivial components of $\Delta\mathbf{R}$ can also be avoided by carefully choosing $\mathbf{b}$ to be in the null-space of $\mathbf{A}$, which usually means it needs to have zero total sum and total torque (see chapter 15 in Ref. [20]). We will discuss numerical techniques to solve (10) shortly.

We then treat any solution $\Delta\mathbf{R}$ to (10) with components significantly larger then $\overline{\Delta l}$ as an unjamming motion. For each $\mathbf{b}$, if we fail to find an unjamming motion, we apply $-\mathbf{b}$ as a loading also, for reasons detailed in Appendix B. In our tests we usually set $\Delta R_{\max} \sim 10D$ and treated any displacement where some $\|\Delta\mathbf{r}_i\| \geq \Delta R_{\text{unjam}} \sim D$ as unjamming motions, but as should be clear from the discussion in Appendix B, the method is not very sensitive to the exact values. One can then save these individual unjamming motions and visualize them, or try to combine several of these into one "most interesting" unjamming motion, as we have done to obtain the figures in this paper.

We stress that despite its randomized character, this algorithm is almost rigorous when used as a test of jamming, in the sense that it is *strictly rigorous* for gap-less packings and also for packings with small gaps as explained in more detail in Appendix B.

IV. TESTING FOR LOCAL, COLLECTIVE AND STRICT JAMMING: *PERIODIC BOUNDARY CONDITIONS*

A. Local Jamming

Recall that the condition for a packing to be locally jammed is that *each particle be fixed by its neighbors*. This is easy to check. Namely, each sphere has to have at least $d+1$ contacts with neighboring spheres, not all in the same d-dimensional hemisphere. This is easy to test in any dimension by solving a small linear program, and in two and three dimensions one can use more elementary geometric constructions.

We prefer the LP approach because it is in the spirit of this work and because of its dimensional independence, and so we present it here. Take a given sphere i and its set of contacts $\{\mathbf{u}_{i*}\}$, and put these as rows in a matrix $\mathbf{A}_i^T$. Then solve the local portion of (A2) (using the simplex algorithm):

$$\min_{\Delta\mathbf{r}_i} (\mathbf{A}_i\mathbf{e})^T \Delta\mathbf{r}_i \tag{12}$$

$$\text{such that} \quad \mathbf{A}_i^T \Delta\mathbf{r}_i \leq \Delta\mathbf{l}_{i,*} \tag{13}$$

which will have an unbounded solution if the sphere i is not locally jammed, as illustrated in Fig. 4.

Of course we can define higher orders of local jamming by asking that *each set of n spheres be fixed by its neighbors*, called *n-stability* in Ref. [13]. However, for $n > 1$ it becomes combinatorially too difficult to check for this. Computationally, we have found testing for local jamming using (13) to be quite efficient and simple. Notice that checking each sphere for local jamming using (13) only once is not enough under this removal scheme. Namely, once a rattling sphere is removed, this removes some contacts from the packing and can make other spheres not locally jammed. Therefore an implementation in which neighbors of rattlers are recycled on a stack of spheres to be checked is needed. We have observed that often, particularly in two-dimensional systems, *all disks can be removed* on the basis of just the local jamming testing.

B. Collective Jamming

The randomized LP algorithm was designed to test for collective jamming in large packings, and in this case the linear program (10) that needs to be solved is very large and sparse. Notice that boundary conditions are only involved when making the list of contacts in the contact network and deciding if certain spheres or contact points are fixed. In the case of periodic boundary conditions, we simply add the usual contacts between original spheres near the boundary of the unit cell and any nearby image spheres.

We have implemented an efficient numerical solution of (10) [and also (21)], using the primal-dual interior-point algorithm in the `LOQO` optimization library (see Ref. [23]). Illustrations of results obtained using this implementation are given throughout this paper.

We would like to stress that primal-dual interior-point algorithms are very well suited for problems of this type, and should also be intuitive to physicists since in essence they solve a sequence of easier problems in which the perfectly rigid inter-sphere contacts are replaced by stiff (but still deformable) nonlinear (logarithmic) springs, carefully numerically taking the limit of infinitely stiff springs. Physicists have often used similar heuristically designed schemes and hand-tuned them, and even suggested that standard optimization algorithms are not practical. We would like to dispel such beliefs and stress the importance of using robust and highly efficient software developed by applied mathematicians around the world, such as Ref. [23], becoming increasingly more available. Not only are the algorithms implemented theoretically well-analyzed, but they are tested on a variety of cases and often contain several alternative implementations of computation-intensive sections targeting different types of problems. Choice of the correct algorithm and the details are often complex, but well worth the effort.

Nonetheless, for three-dimensional problems the available implementations of interior-point algorithms based on direct linear solvers are too memory demanding and inefficient. Tuned implementations based on conjugate-gradient iterative solvers are needed. We plan to develop efficient parallel algorithms suited for these types of problems and make them publicly available in the very near future.

C. Strict Jamming

To extend the notion of collective jamming to strict jamming we introduced deformations of the boundary. In the case of periodic packings, *the lattice* $\mathbf{\Lambda}$ *is the boundary.* Therefore, the only difference with collective jamming is that we will now allow the lattice to change while the spheres move, i.e., $\Delta\mathbf{\Lambda} \neq 0$ in (2). The **lattice deformations** $\Delta\mathbf{\Lambda}$ will also become unknowns in (10), but since these too enter linearly in (2), we still get a linear program, only with coefficient matrix $\mathbf{A}$ augmented with new (denser) rows in the columns corresponding to contacts across the periodic boundary. The actual implementation now requires more care and bookkeeping, but the conceptual changes should be clear, and the randomized LP algorithm remains applicable.

Obviously, we cannot allow the volume of the unit cell to enlarge, since the unit cell is in a sense the container holding the packing together. Therefore, we only consider *volume-non-increasing continuous lattice deformations* $\Delta\mathbf{\Lambda}(t)$:

$$\det\left[\widetilde{\mathbf{\Lambda}} = \mathbf{\Lambda} + \Delta\mathbf{\Lambda}(t)\right] \leq \det\mathbf{\Lambda} \textit{ for } t > 0 \tag{14}$$

We now think of $[\Delta\mathbf{R}(t), \Delta\mathbf{\Lambda}(t)]$ as an unjamming motion. Through a relatively simple mathematical analysis to be presented in Ref. [18], it can be shown that the principles that applied to unjamming motions of the spheres $\Delta\mathbf{R}$ still remain valid even when we extend the notion of an unjamming motion to include a deforming lattice. That is, we can still only focus on linear motions $\Delta\mathbf{\Lambda}(t) = \mathbf{W}t$, $\mathbf{W} = \text{const.}$ and the final small deformations $\Delta\mathbf{\Lambda} = \Delta\mathbf{\Lambda}(1)$, and need to consider only first-order linearizations of the impenetrability (3) and non-expansion (14) nonlinear constraints[35].

The linearized version of (14) is:

$$\mathrm{Tr}[(\Delta\mathbf{\Lambda})\mathbf{\Lambda}^{-1}] \leq 0 \tag{15}$$

and this is just one extra linear constraint to be added to the linear program (10). An extra condition which needs to be added is that $(\Delta\mathbf{\Lambda})\mathbf{\Lambda}^{-1}$ be symmetric, which is also an added linear constraint,

$$(\Delta\mathbf{\Lambda})\mathbf{\Lambda}^{-1} = \boldsymbol{\varepsilon} \text{ where } \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^T \tag{16}$$

where we add $\boldsymbol{\varepsilon}$ as an unknown in the randomized LP algorithm. This condition in fact does nothing more then eliminate trivial rotations of the lattice.

The motivation for the category of strict jamming and its above interpretation in the periodic case should be clear: Changing the lattice in a volume non-increasing way models *macroscopic non-expansive strain* (i.e., a non-tensile applied stress) and is therefore of great relevance to studying the macroscopic mechanical properties of random packings (see Ref. [19]). In fact, $\boldsymbol{\varepsilon} = (\Delta\mathbf{\Lambda})\mathbf{\Lambda}^{-1}$ can be interpreted as the "macroscopic" *strain tensor*, which explains why it is symmetric and also trace-free for shear deformations. We also again point out that strict jamming is (significantly) stronger then collective jamming for periodic boundary conditions, particularly in two-dimensional packings. This point is illustrated in Fig. 7, which shows an unjamming motion involving a deformation of the lattice, even though this lattice packing is collectively jammed.

D. Shrink-And-Bump Heuristic

The following heuristic test for collective jamming has been suggested in Ref. [11]: Shrink the particles by a small amount δ and then start the Lubachevsky-Stillinger molecular dynamics algorithm with random velocities, and see if the system gets unjammed. One would also slowly enlarge the particles back to their original size while they bump around, so as to allow finite termination of this test (within numerical accuracies). We call this the *shrink-and-bump heuristic.* The idea is that the vector of velocities takes on random values in velocity space and if there is a direction of unjamming, it will be found with a high probability and the system will unjam. Animations of this process can be found at Ref. [21].

This kind of heuristic has the advantage of being very simple and thus easy to implement and use (and also incorporates nonlinear effects), and it is also very efficient. Its disadvantages are its non-rigorous character and indeterminacy, artificial introduction of dynamics into a geometrical problem, and most of all, its strong dependence on the exact value of δ. For example, animations showing how the Kagomé lattice inside a container made of fixed spheres (as in Fig. 2) can be unjammed with a large-enough δ, even though it is actually collectively jammed under these boundary conditions, can be found at Ref. [21]. In fact, many jammed *large* packings will appear unstable under this kind of test, as motivated with the notion of *uniform stability*, defined in Ref. [13].

V. ADDITIONAL APPLICATIONS

A. Compressing Packings using Linear Programming

In this work we emphasize the utility of the randomized linear programming algorithm as a testing tool for jamming, and also for finding representative unjamming motions. The unjamming motions one finds can be used inside compression algorithms. For example, the Lubachevsky-Stillinger algorithm may sometimes get stuck in a particular configuration even though the configuration is not collectively or strictly jammed (particularly in two dimensions, as explained later). The unjamming motion obtained from the linear programming algorithm can then be used to continue the compression. For example, one can displace the spheres by $\Delta\mathbf{R}$ and restart the compression with random velocities or use initial velocities along $\Delta\mathbf{R}$ to unjam the packing and continue the simulation.

Even more can be done with linear programming in this direction. For example, one can ask the question of whether there is an unjamming motion $\Delta\mathbf{R}$ in which all sphere contacts are lost. This can be done for example by solving the LP,

$$\max_{\Delta\mathbf{R},\varepsilon} \alpha \tag{17}$$

$$\text{such that } \mathbf{A}^T\Delta\mathbf{R} \leq -\alpha D \tag{18}$$

$$0 \leq \alpha \leq 1 \qquad \text{for boundedness} \tag{19}$$

By displacing the spheres by such a $\Delta\mathbf{R}$ we would create a "cushion" of free space around each sphere, so that we can actually increase the radius of the spheres by at least αD and thus increase the density, or equivalently, the packing fraction φ. We believe these kinds of approaches to be too inefficient, since solving a large-scale linear program is too expensive to be used iteratively. This is reminiscent of the high-quality but rather expensive compression algorithm of Zinchenko (see Ref. [24]).

Our future work will focus on using mathematical programming algorithms and rigidity theory to design high-quality algorithms for design of packings with target properties, using systems of stiff nonlinear springs as an intermediary.

B. Kinematic/Static Duality

The subject of kinematic/static duality and its physical meaning and implications are discussed at length in Ref. [16], and elsewhere in various degrees of relevance and different perpectives [10, 12, 14, 15, 17]. Here we only comment on it because of its relevance to the randomized LP algorithm for testing jamming, and leave further discussion of this important subject to Ref. [18].

The dual of the displacement formulation LP (10) (excluding the additional practical safeguard constraint $\Delta\mathbf{R} \leq \Delta\mathbf{R}_{\max}$,) also has a very physical interpretation and it gives us the *interparticle repulsive*[36] *forces* f as dual variables, and we call it the ***force formulation*** LP:

$$\begin{aligned} & \max_{\mathbf{f}} (\Delta\mathbf{l})^T\mathbf{f} && \text{for virtual work} && (20)\\ \text{such that} \quad & \mathbf{A}\mathbf{f} = \mathbf{b} && \text{for equilibrium} && (21)\\ & \mathbf{f} \leq 0 && \text{for repulsion only} && (22) \end{aligned}$$

The physical interpretation of the objective function in both the displacement (10) and force formulations (21) is that of (virtual) mechanical work done by the external force load $\mathbf{b}$ applied to the spheres. These two LP's are of great importance in studying the *stress-strain* behavior of granular materials, as explained in Ref. [16], and since they are equivalent to each other, we can call them the *ASD stress-strain LP*.

We wish to emphasize that by using primal-dual interior point algorithms we automatically get both forces and displacements using the same implementation. For example, both `LOQO` and `PCx` ([23]) return both primal and dual solutions to the user. We have emphasized the displacement formulation (10) simply because we based our discussion of jamming on a kinematic perspective, but a parallel static interpretation can easily be given. For example, a random $\mathbf{b}$ used in the randomized LP algorithm that finds an unbounded unjamming motion physically corresponds to a load that the packing cannot support, i.e. the force formulation LP is (dual) infeasible, implying that the displacement formulation LP is (primal) unbounded.

The meaning of jamming within the ASD in the presence of small gaps from a force-based standpoint now becomes clear. A collectively jammed packing can resist (support) *any* non-global loading by (as) small (as possible) rearrangements of the spheres, in which some of the potential contacts are open and others closed, depending on the loading and the interparticle gaps. For strict jamming, the part of $\mathbf{b}$ corresponding to the strain $\boldsymbol{\varepsilon}$ represents the *global* stresses in the packing, and strictly jammed packings can thus support *any* load with a non-tensile global component. Note however that a packing may be able to support all global loads even though it is not strictly jammed, as it may be unstable due to the existence of collective unjamming mechanisms.

In general the stress-strain LP will be highly degenerate and its primal and/or dual solution not unique. However, as Roux points out, the existence of small gaps in random packings is very important in this context. Namely, if $\Delta\mathbf{l}$ is random and nonzero (however small), and $\mathbf{b}$ is also random, theorems on the generic character of linear programs (see the references in Ref. [20]) can be invoked to guarantee that both the primal and dual solutions will be non-degenerate. A non-degenerate solution to (21) corresponds to an isostatic force-carrying contact network, a fact noted and explained in a great many ways by various researchers in the field of granular materials [15–17]. We just mention these points here in order to stimulate interest among the physical community in the very relevant results to be found in the mathematical programming literature.

VI. RESULTS

We have applied the randomized LP algorithm to test for the different jamming categories in practice. The primary aim of this work is not to give exhaustive results, but rather to introduce a conceptual framework and some algorithms. Nonetheless, in this section we present some sample relevant results for both ordered and disordered periodic packings.

A. Periodic Lattice Packings

Table 1 in Ref. [10] gives a classification of some common simple lattice packings into jamming categories for hard-wall boundary conditions. Table I reproduces this for periodic boundary conditions. As we explained in section II, the results in principle will depend on the number of unit cells N_c chosen as the original packing, and also on the unit cell chosen, so the terminology "lattice X is Y jammed" is used loosely here.

Lattice	φ	L	Z	C	$\mathbf{N}_c$	S	$\mathbf{N}_c$	N_s
Honeycomb	0.605	Y	3	N	$(2,1)$, $(1,2)$	N	$(1,1)$	2
Kagomé	0.680	Y	6	N	$(1,1)$	N	$(1,1)$	3
Square	0.785	Y	4	N	$(2,1)$	N	$(1,1)$	1
Triangular	0.907	Y	6	Y		Y		1
Diamond	0.340	Y	4	N	$(1,1,2)$	N	$(1,1,1)$	2
SC	0.524	Y	6	N	$(1,1,2)$	N	$(1,1,1)$	1
BCC	0.680	Y	8	N	$(1,1,2)$	N	$(1,1,1)$	1
FCC	0.741	Y	12	Y		Y		1
HCP	0.741	Y	12	Y		Y		2

Table I: *Classification of some simple lattices into jamming categories.* We give the packing (i.e., covering) fraction φ (to three decimal places), the coordination number Z, and the number of disks/spheres N_s per unit cell, an assessment of whether the lattice is locally (L), collectively (C) or strictly (S) jammed (Y is jammed, N is not jammed), and the "smallest" number of unit cells $\mathbf{N}_c$ on which an unjamming motion exists (illustrated at Ref. [21]).

It turns out that in the cases given in Table I, the packings we have classified as not collectively or strictly jammed will not be so for any large $\mathbf{N}_c$. Here we give the smallest $\mathbf{N}_c$ for which we have found unjamming motions, and illustrate some of these in Figs. 8 and 9.

Moreover, the packings classified as jammed, in this case being the maximal density packings in two dimensions (triangular) and three dimensions (FCP and HCP) will be so for any finite $\mathbf{N}_c$. Here we just point out for the curious that the triangular lattice is not the only strictly jammed ordered disk packing; two other examples are shown in Fig. 10 (see Ref. [8]).

B. Periodic Random Packings

We also tested a sample of periodic random packings in two and three dimensions produced via the Lubachevsky-Stillinger compression algorithm [11] at different compression rates. This algorithm often tends to produce a certain number of rattlers, i.e., spheres which are not locally jammed, which we *remove* before testing for jamming. We would like to stress that these are not comprehensive tests, but they do illustrate some essential points, and so instead of giving tables with statistics, we give some representative illustrations. The tolerances for the interparticle gaps used were in the range $\delta \in [0.25, 0.50]$, and, as explained earlier, the results in some cases depend on this chosen tolerance, but not strongly.

All random disk (i.e., two-dimensional) packings we tested were not strictly jammed. At the typical LS end states of roughly $\varphi \approx 0.82$, we generally found that the packings were collectively jammed (with some exceptions such as the packing shown in Fig. 11), although not strictly jammed, as with the packing depicted in Fig. 12. However, even at very high densities ($\varphi \approx 0.89$), the packings were only collectively jammed, as illustrated in Fig. 13. Note that quite different properties were observed for the three-dimensional packings: *All random sphere (i.e., three-dimensional) packings we tested were strictly (and thus collectively) jammed.*

VII. CONCLUSIONS AND DISCUSSION

Our results have important implications for the classification of random disk and sphere packings and suggest a number of interesting avenues of inquiry for future investigations. Random disk packings are less well-understood then sphere packings. The tendency of disk packings to "crystallize" (to form ordered, locally dense domains) at sufficiently high densities is well established. For example, Quickenden and Tan [25] experimentally estimated the packing fraction of the "random close packed" (RCP) state to be $\varphi \approx 0.83$ and found that the packing fraction could be further increased until the maximum value of $\varphi = 0.906$ is achieved for the triangular lattice packing. By contrast, random sphere packings at φ in the range $0.63 - 0.66$ cannot be further densified.

Our recent understanding of the ill-defined nature of random close packing and of jamming categories raise serious questions about previous two-dimensional studies, particularly the stability of such packings. Our present study sugests that random disk packings are not strictly jammed at $\varphi \approx 0.83$; at best they may be collectively jammed. Of course, the old concept of the RCP state incorrectly did not account for the jamming category of the packing. Previous attempts to estimate the packing fraction of the "random loose" state [26] are even more problematic, given that this term is even less well-defined than the RCP state. The best way to categorize random disk packings is to determine the maximally random jammed (MRJ) state (see Ref. [7]) for each of the three jamming categories (local, collective and strict). Such investigations will be carried out in the future, and we have some some preliminary results and promising avenues of approach.

The identification of the MRJ state for strictly jammed *disk* packings is an intriguing open problem. On the one hand, we have shown that random packings exist with densities in the vicinity of the maximum possible value ($\varphi = \frac{\pi}{2\sqrt{3}}$) that are not strictly jammed, and on the other hand, there is a conjectured achievable lower bound $\varphi \geq \frac{\sqrt{3}\pi}{8}$ corresponding to the "reinforced" Kagomé lattice (see Fig. 10). For random sphere packings, an initial study undertaken in [27], using the LP algorithm described in this work, found that maximally disordered random packings around $\varphi \approx 0.63$ were strictly jammed, suggesting a close relation between the conventionally accepted RCP state and the MRJ state for strictly jammed packings. The conventionally accepted RCP state in two dimensions may be closely related to the MRJ state for *collectively* jammed packings. Much less obvious is what the MRJ state for collectivelly jammed sphere packings is. Finally, a completely unexplored question concerns the identification of the MRJ state for locally jammed disk and sphere packings.

In this work we have proposed, implemented, and tested a practical algorithm for verifying jamming categories in finite sphere packings based on linear programming, demonstrated its simplicity and utility, and presented some representative results for ordered lattices and random packings. Interestingly, the random packings that we tested were strictly jammed in three dimensions, but *not* in two dimensions. Future applications of the randomized linear-programming algorithm are to be expected. We will further present and explore the theoretical connections between rigidity and jamming, kinematic and static rigidity, rigidity and energy minima, rigidity and stability, and finite, periodic and infinite packings in Ref. [18], and work is already under way to provide highly efficient implementations of various optimization algorithms for linear and nonlinear programming on large-scale (contact) networks.

Acknowledgments

The authors would like to thank Robert Vanderbei for numerous helpful discussions on mathematical programming issues and for providing us with the `LOQO` optimization library.

Appendix A: NON-RANDOMIZED LP TESTING FOR JAMMING

Determining whether the system (8) is feasible is an interesting mathematical programming problem. The gap-less case $\Delta \mathbf{l} = 0$, $\Delta l_{\text{large}} \rightarrow 0^+$, can be studied rigorously. When gaps are present, of course, the unjamming condition is mathematically ambiguous and also the ASD approximation becomes inexact. One approach is the following: Solve the following linear program aimed at maximizing the sum of the (positive) gap dilations $\left(\Delta \mathbf{l} - \mathbf{A}^T \Delta \mathbf{R}\right)_{i,j}$,

$$\min_{\Delta \mathbf{R}} \sum_{\{i,j\}} (\mathbf{A}^T \Delta \mathbf{R})_{i,j} = \min \left(\mathbf{A}\mathbf{e}\right)^T \Delta \mathbf{R} \tag{A1}$$

$$\text{such that} \qquad \mathbf{A}^T \Delta \mathbf{R} \leq \Delta \mathbf{l} \tag{A2}$$

where $\mathbf{e}$ is the unit vector, and then look at the magnitudes of the gap dilations and the displacements (these may be unbounded) and decide if they are large enough to consider the solution an unjamming motion. Otherwise the packing is jammed. Notice that this will usually produce a single unjamming motion, which we have found to be rather uninteresting for lattice packings in the sense that it is highly sensitive to $\mathbf{N}_c$.

Appendix B: THE GEOMETRY OF THE SET OF UNJAMMING MOTIONS

The linearized impenetrability constraints $\mathbf{A}^T \Delta \mathbf{R} \leq \Delta \mathbf{l}$ define a polyhedral set $\mathcal{P}_{\Delta \mathbf{R}}$ of *feasible (linearized) displacements*. Every such polyhedron consists of a finite piece $\mathcal{P}_{\Delta \mathbf{R}}^{\text{hull}}$, the convex hull of its extreme points, and possibly an unbounded piece $\mathcal{C}_{\Delta \mathbf{R}}$, a finitely generated *polyhedral cone*. In some cases this cone will be empty (i.e. $\mathcal{C}_{\Delta \mathbf{R}} = \{0\}$), but in others it will not, as can be seen in Fig. 4. The full nonlinear impenetrability constraints given by (5) define the true set of feasible displacements $\mathcal{P}_{\Delta \mathbf{R}}^{\text{NL}}$, which always relaxes the linearization: $\mathcal{P}_{\Delta \mathbf{R}} \subseteq \mathcal{P}_{\Delta \mathbf{R}}^{\text{NL}}$.

A mathematically very well defined formulation is to take any ray in the cone $\mathcal{C}_{\Delta \mathbf{R}}$ as an unjamming motion, and exclude others; however, as Fig. 4 shows, the elongated corners of this polyhedron are in fact very likely to be unbounded in the true non-linear feasible set of displacements, so we prefer to take any "long" direction in $\mathcal{P}_{\Delta\mathbf{R}}$ as an unjamming motion. It is possible to also treat the nonlinear constraints without linearizing them. However, this significantly increases the computational effort and requires more careful considerations beyond the scope of this paper. We are currently investigating such nonlinear extensions of the randomized LP algorithm for testing for jamming.

We note that the randomized LP algorithm proposed here strictly answers the question of whether the polyhedral set of feasible displacements contains an unbounded ray just by applying two (nonzero) loads $\mathbf{b}$ and $-\mathbf{b}$. This is because an attempt to find such a ray will be unsuccessful only if $-\mathbf{b} \in \mathcal{C}^*_{\Delta\mathbf{R}}$, where $\mathcal{C}^*_{\Delta\mathbf{R}}$ is the conjugate cone of $\mathcal{C}^*_{\Delta\mathbf{R}}$, and in this case $\mathbf{b} \notin \mathcal{C}^*_{\Delta\mathbf{R}}$, so that using the load $-\mathbf{b}$ will find a ray if such a ray exists. Also, we note that one cannot hope to fully characterize the cone of first-order unjamming motions $\mathcal{C}^*_{\Delta\mathbf{R}}$ (i.e. find its convex hull of generating rays), as this is known to be an NP complete problem related to the full enumeration of the vertices of a polyhedron. Our randomized approach essentially finds a few sample rays in $\mathcal{C}^*_{\Delta\mathbf{R}}$.

---

[1] A. Mehta, ed., *Granular Matter* (Springer-Verlag, New York, 1994).

[2] S. F. Edwards and D. V. Grinev, Chem. Eng. Sci. **56**, 5451 (2001).

[3] R. Zallen, *The Physics of Amorphous Solids* (Wiley, New York, 1983).

[4] J. P. Hansen and I. R. McDonald, *Theory of Simple Liquids* (Academic Press, New York, 1986).

[5] S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties* (Springer-Verlag, New York, 2002).

[6] T. Aste and D. Weaire, *The Pursuit of Perfect Packing* (IOP Publishing, 2000).

[7] S. Torquato, T. M. Truskett, and P. G. Debenedetti, Phys. Rev. Lett. **84**, 2064 (2000).

[8] F. H. Stillinger, S. Torquato, and H. Sakai (2002), submitted to Phys. Rev. E.

[9] L. F. Toth, *Regular figures* (Pergamon Press, 1964).

[10] S. Torquato and F. H. Stillinger, J. Phys. Chem. B **105**, 11849 (2001).

[11] B. D. Lubachevsky and F. H. Stillinger, J. Stat. Phys. **60**, 561 (1990); B. D. Lubachevsky, F. H. Stillinger, and E. N. Pinson, J. Stat. Phys. **64**, 501 (1991).

[12] R. Connelly, Structural Topology **14**, 43 (1988), see also [22].

[13] R. Connelly, K. Bezdek, and A. Bezdek, Discrete and Computational Geometry **20**, 111 (1998).

[14] R. Connelly and W. Whiteley, SIAM Journal of Discrete Mathematics **9**, 453 (1996).

[15] M. F. Thorpe and P. M. Duxbury, eds., *Rigidity Theory and Applications*, Fundamental Materials Research (Kluwer/Plenum, 1999).

[16] J. N. Roux, Phys. Rev. E **61**, 6802 (2000).

[17] C. F. Moukarzel, Phys. Rev. Lett. **81**, 1634 (1998).

[18] R. Connelly, A. Donev, S. Torquato, and F. H. Stillinger (2002), in preparation.

[19] S. Torquato, A. Donev, and F. H. Stillinger (2002), submitted for publication.

[20] R. Vanderbei, *Linear Programming:* Foundations and Extensions (Kluwer, 1997).

[21] `http://atom.princeton.edu/donev/Packing`, homepage for the sphere packing project, with useful supplementary materials.

[22] R. Connelly, Structural Topology **16**, 57 (1991), second part of [12].

[23] The general purpose interior-point `LOQO` optimization library is not public-domain, but can be tried at `http://orfe.princeton.edu/~loqo/`. The public-domain `PCx` library implements interior point linear programming algorithms and can be found at `http://www-fp.mcs.anl.gov/otc/Tools/PCx/`.

[24] A. Zinchenko, J. Comp. Phys. **114**, 298 (1994).

[25] T. J. Quickenden and G. K. Tan, J. Colloid Interface Sci. **48**, 382 (1974).

[26] W. Uhler and R. Schilling, J. Phys. C **18**, L979 (1985).

[27] A. R. Kansal, S. Torquato, and F. H. Stillinger, Phys. Rev. E **66**, 041109 (2002)

[28] In particular, the cessation of flow as jamming is approached.

[29] In particular, the infinite elastic moduli near jamming.

[30] This is the second definition (definition b) in section 2.1 of Ref. [14].

[31] This the first definition (definition a) in section 2.1 of Ref. [14].

[32] This is the third definition (definition c) in section 2.1 of Ref. [14].

[33] The sign notation may be a bit unorthodox but is taken from Ref. [20].

[34] This is in fact the negative transpose of what is usually taken to be the rigidity matrix, and is chosen to fit the notation in Ref. [20].

[35] The condition (16) needs to hold also.

[36] We choose a negative sign for repulsive forces here in agreement with mathematical literature [14].

III. FIGURE CAPTIONS

Figure 1: *Unjamming the honeycomb lattice* inside a hard-wall simple box container. The arrows in the figures given here show the direction of motion of the spheres $\mathbf{V}$ in the linear unjamming motion, scaled by some arbitrary constant to enhance the figure. $\mathbf{N}_c = (3, 2)$ unit cells make this small packing.

Figure 2: *Unjamming the honeycomb lattice.* A sub-packing of size $\mathbf{N}_c = (3, 3)$ of an infinite honeycomb lattice packing is pinned by freezing all neighboring image disks. A representative unjamming motion is shown as a sequence of several frames between times $t = 0$ and $t = 1$. The unshaded disks represent the particles in the generating packing $P(\widehat{\mathbf{R}})$, while the shaded ones are image disks that touch one of the original disks.

Figure 3: *Unjamming the Kagomé lattice.* Periodic boundary conditions are used with $\mathbf{N}_c = (2, 2)$.

Figure 4: *Feasible displacements polyhedron.* The figures show three stationary (dark gray) disks surrounding a mobile disk (light gray). For each of the three stationary disks, we have a nonlinear impenetrability constraint that excludes the mobile disk from a disk of radius D surrounding each stationary disk (dark circles). Also shown are the linearized versions of these constraints (dark lines), which are simply tangents to the circles at the point of closest approach, as well as the region of feasible displacements bounded by these lines (shaded gray).

This region is a polyhedral set, and in the left figure it is bounded, meaning that within the ASD the mobile disk is locally jammed (trapped) by its three neighbors, while on the left it is unbounded, showing the cone of locally unjamming motions (escape routes). Notice that with the true nonlinear constraints, the mobile disk can escape the cage of neighbors in both cases, showing that the ASD is not exact. However, it should also be clear that this is because we have relatively large interparticle gaps here.

Figure 5: The packing from Fig. 4 shown again with a numbering of the disks. D_i denotes particle i and E_{ij} denotes the contact between the ith and jth particles, i.e., the contact $\{i, j\}$.

Figure 6: *Contact network* of a random packing of 100 disks with periodic boundary conditions and $\delta = 0.5$. Periodic contacts with neighboring image spheres are also shown. All disks are locally jammed within the rather large gap tolerance employed.

Figure 7: *Example of a lattice deformation.* The above periodic packing (packing 3 in Ref. [22]) is collectively jammed, but not strictly jammed. It can be continuously sheared toward the triangular lattice by deforming the lattice in a volume-preserving manner, as shown here.

Figure 8: *Simple collective mechanisms in the Kagomé and honeycomb lattices*, respectively. These lattices are not collectively jammed with periodic boundary conditions, as the sample unjamming motions periodic with $\mathbf{N}_c = (1, 1)$ for Kagomé (left) and $\mathbf{N}_c = (1, 2)$ for honeycomb (right) shown here illustrate.

Figure 9: *Shearing the honeycomb lattice.* The honeycomb lattice is not strictly (or collectively) jammed, and an example of a lattice deformation with $\mathbf{N}_c = (1, 1)$ is shown, replicated on several unit cells to illustrate the shear character of the strain $\boldsymbol{\varepsilon} = (\Delta\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{-1}$ [c.f. (16)]. Note that only three (original) spheres are involved in the actual calculation of this unjamming motion, the rest are image spheres.

Figure 10: *Examples of strictly jammed lattices* in two dimensions. The 6/7th lattice (packing number 2 in Ref. [22] and the last packing in Ref. [8]), left, is obtained by removing every 7th disk from the triangular lattice. The reinforced Kagomé lattice, right, is obtained by adding an extra "row" and "column" of disks to the Kagomé lattice and thus has the same density in the thermodynamic limit, namely, it has every 4th disk removed from the triangular packing (see also Ref. [8]).

Figure 11: A random packing ($\varphi = 0.82$) of 1000 disks that is *not collectively jammed*, and a representative periodic unjamming motion.

Figure 12: A random packing ($\varphi = 0.83$) of 1000 disks that is *collectively jammed but not strictly jammed*, and a representative unjamming motion. Though it is hard to see from this figure, this is indeed a shearing motion that induces an unjamming mechanisms. A more insightful animation can be found at the webpage [21].

Figure 13: A *dense* ($\varphi = 0.89$) random packing of 1000 disks that is *collectively jammed but not strictly jammed*, and a representative unjamming motion. One can see the grains gliding over each grain boundary due to the shear, bringing this packing closer to a triangular lattice.

IV. FIGURES

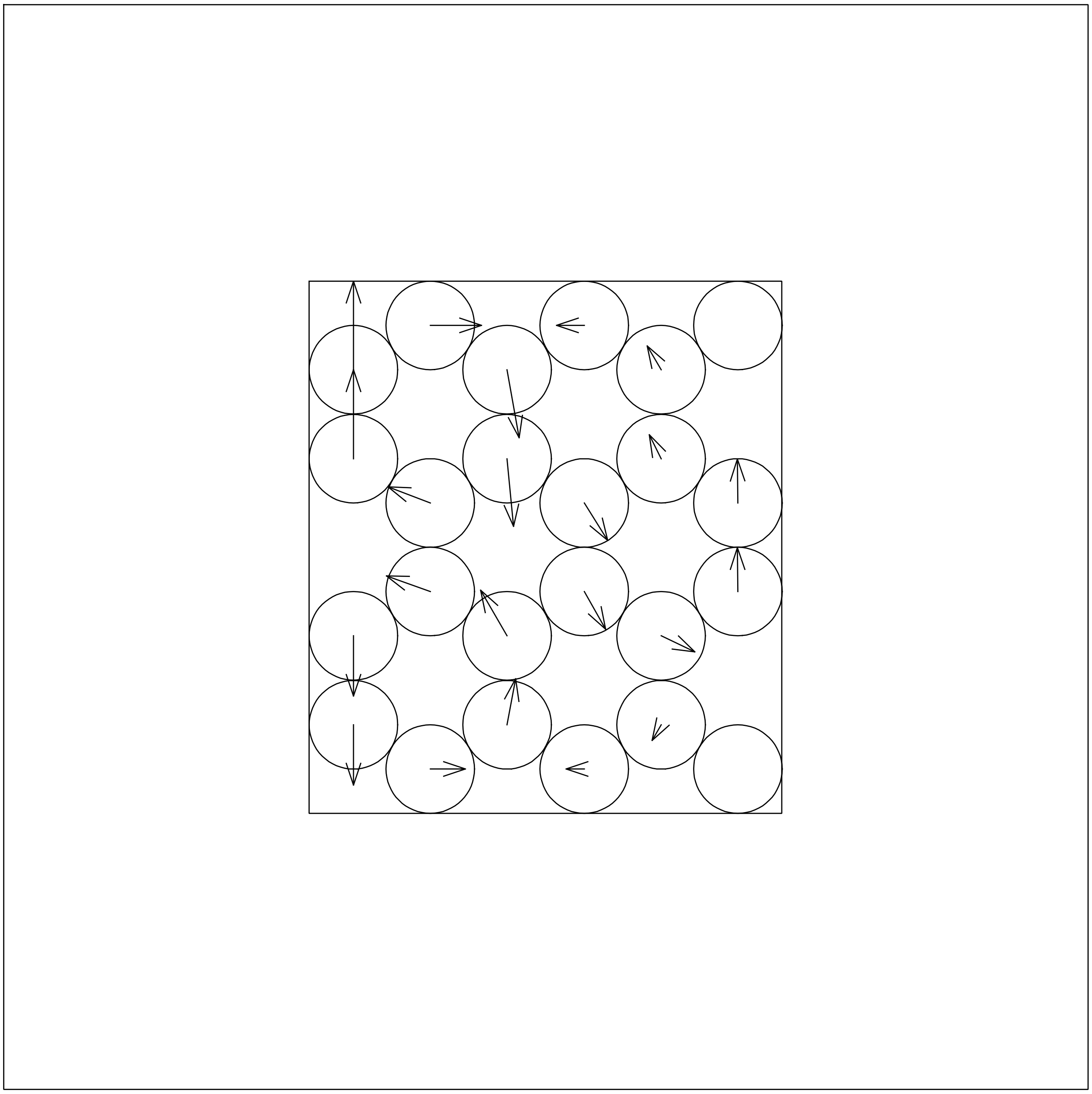

Figure 1: Donev *et al.*

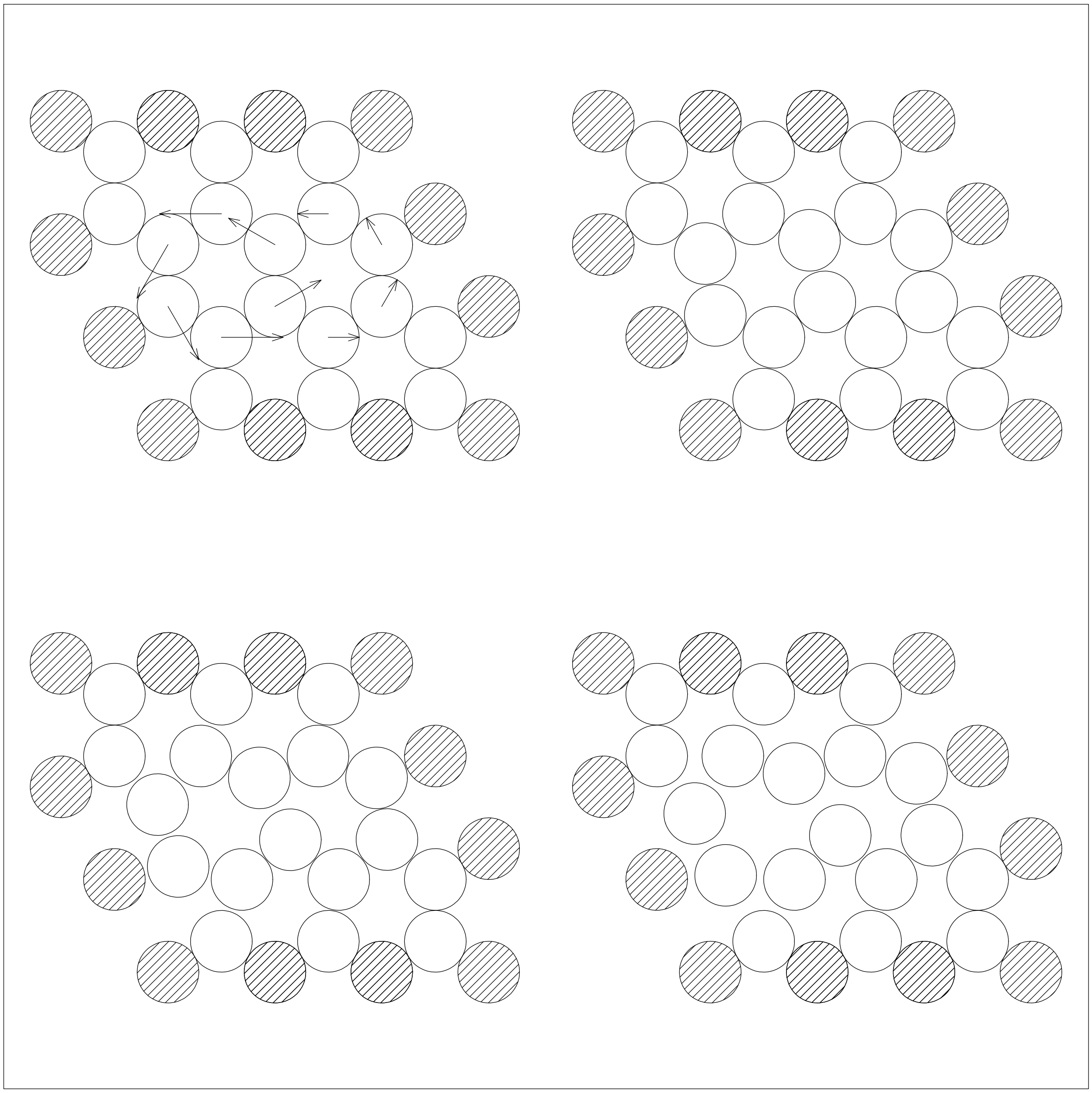

Figure 2: Donev *et al.*

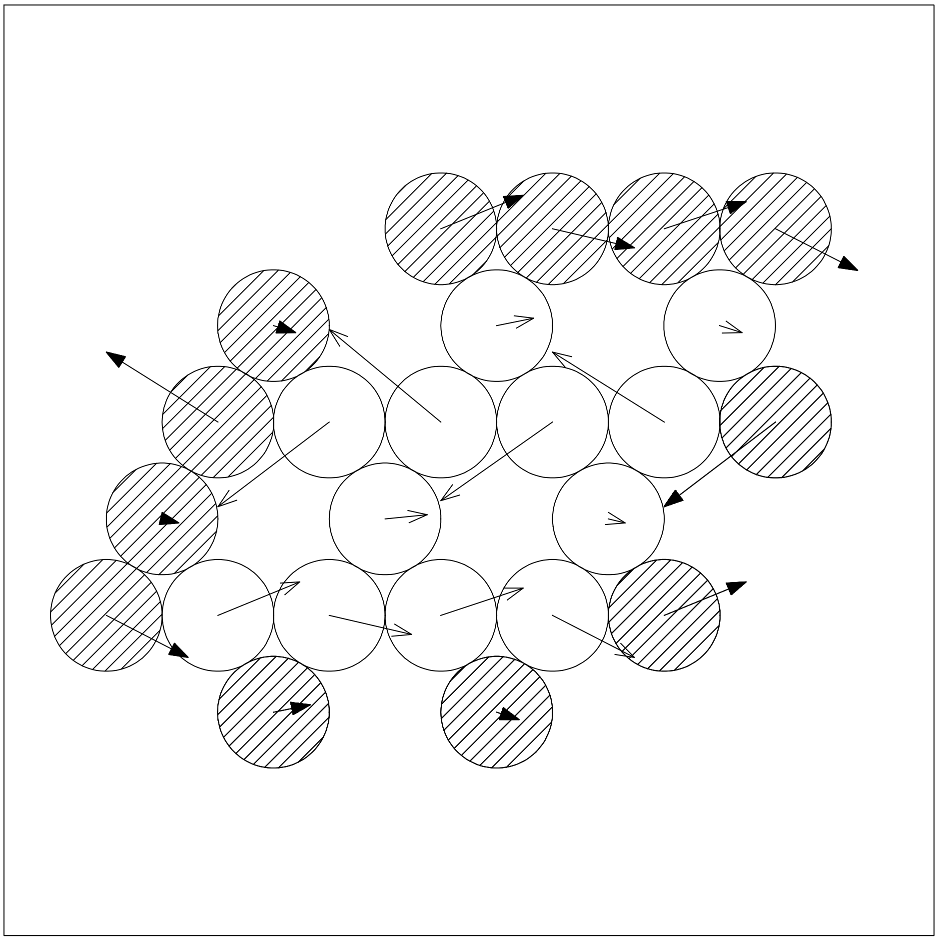

Figure 3: Donev *et al.*

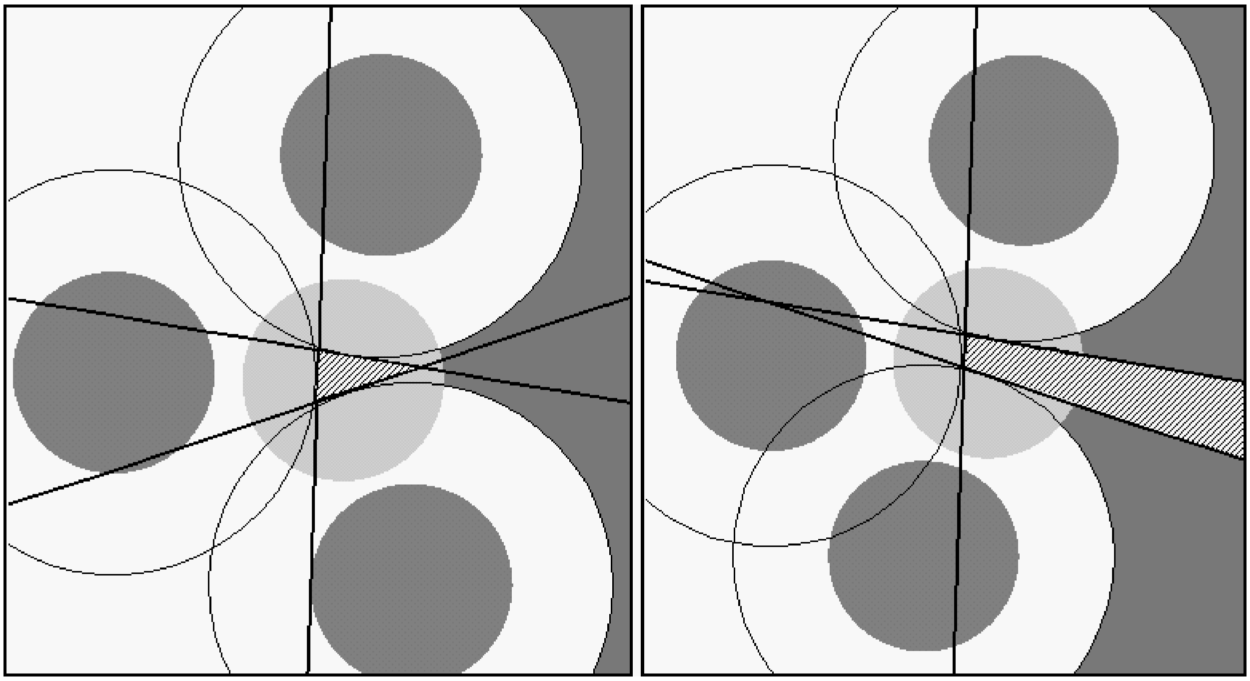

Figure 4: Donev *et al.*

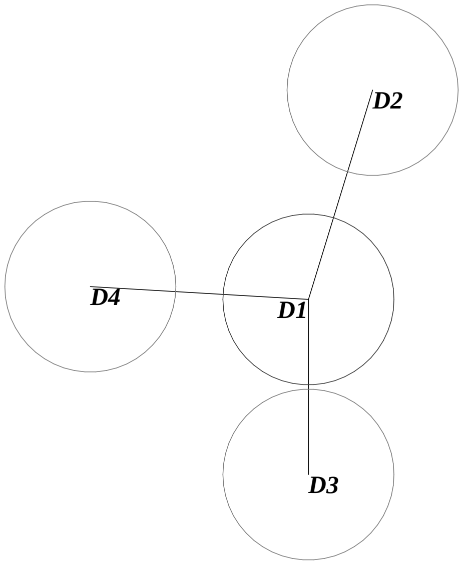

Figure 5: Donev *et al.*

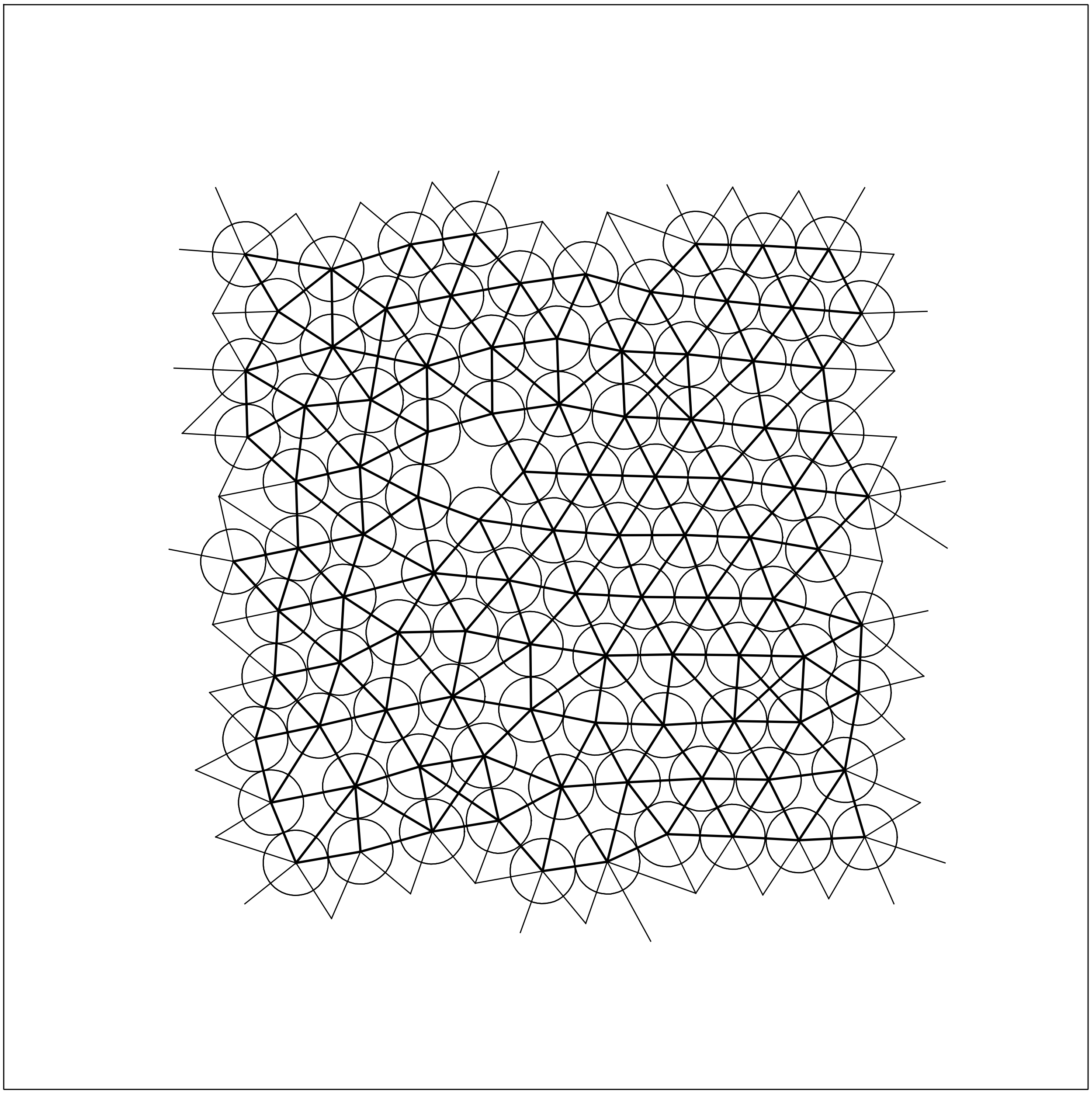

Figure 6: Donev *et al.*

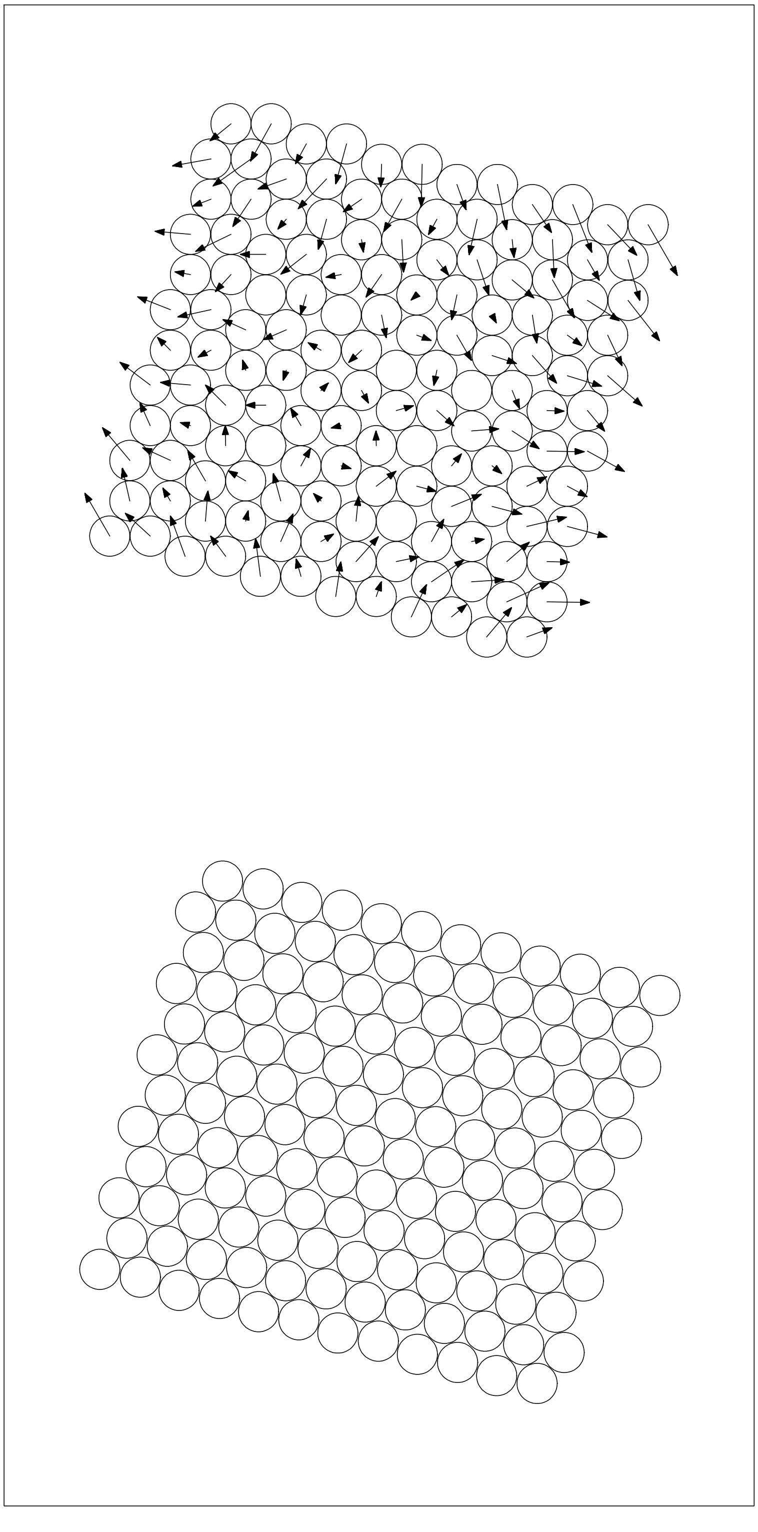

Figure 7: Donev *et al.*

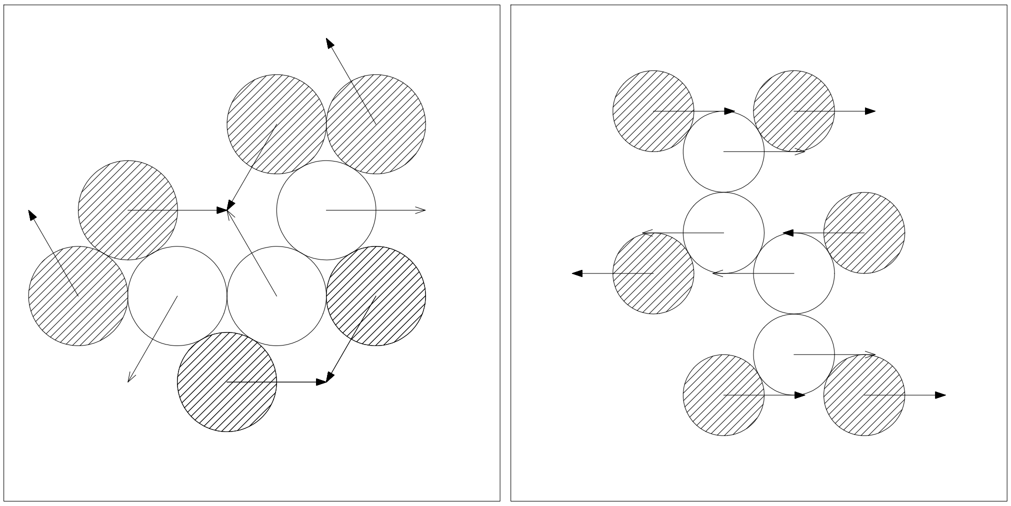

Figure 8: Donev *et al.*

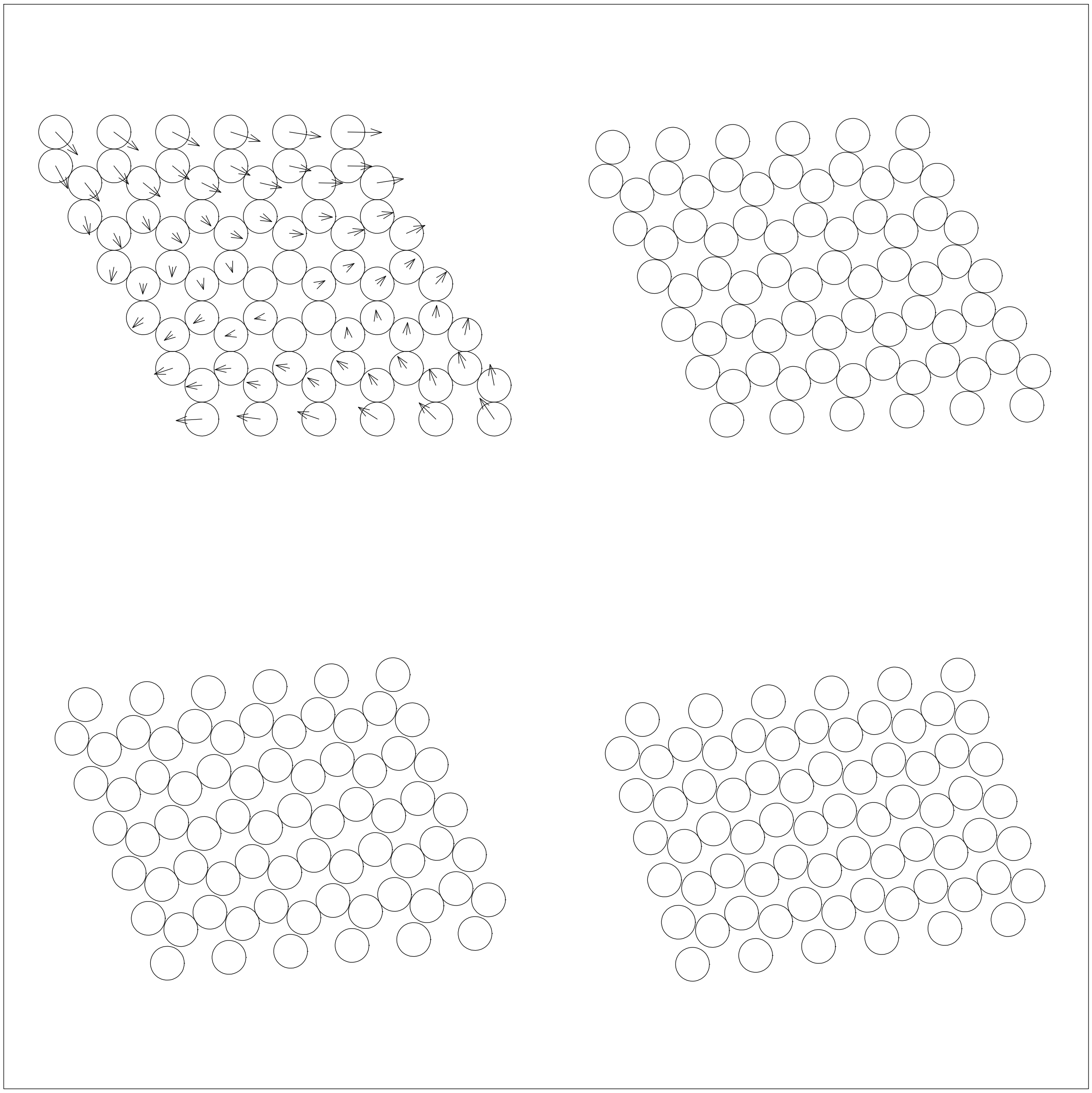

Figure 9: Donev *et al.*

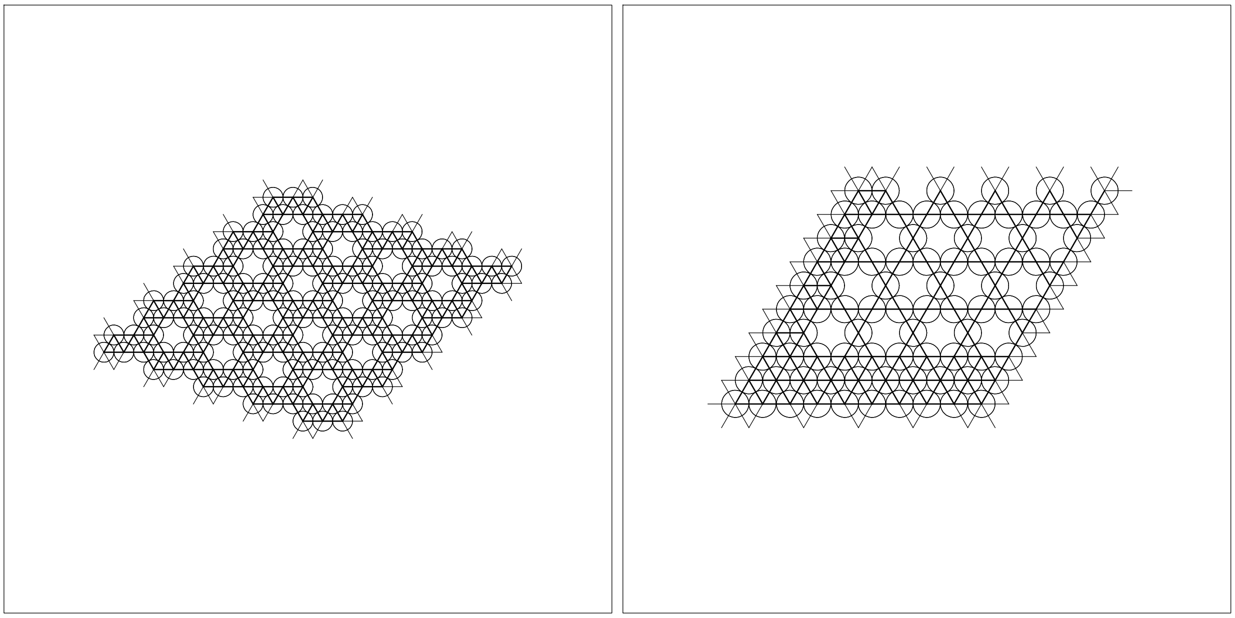

Figure 10: Donev *et al.*

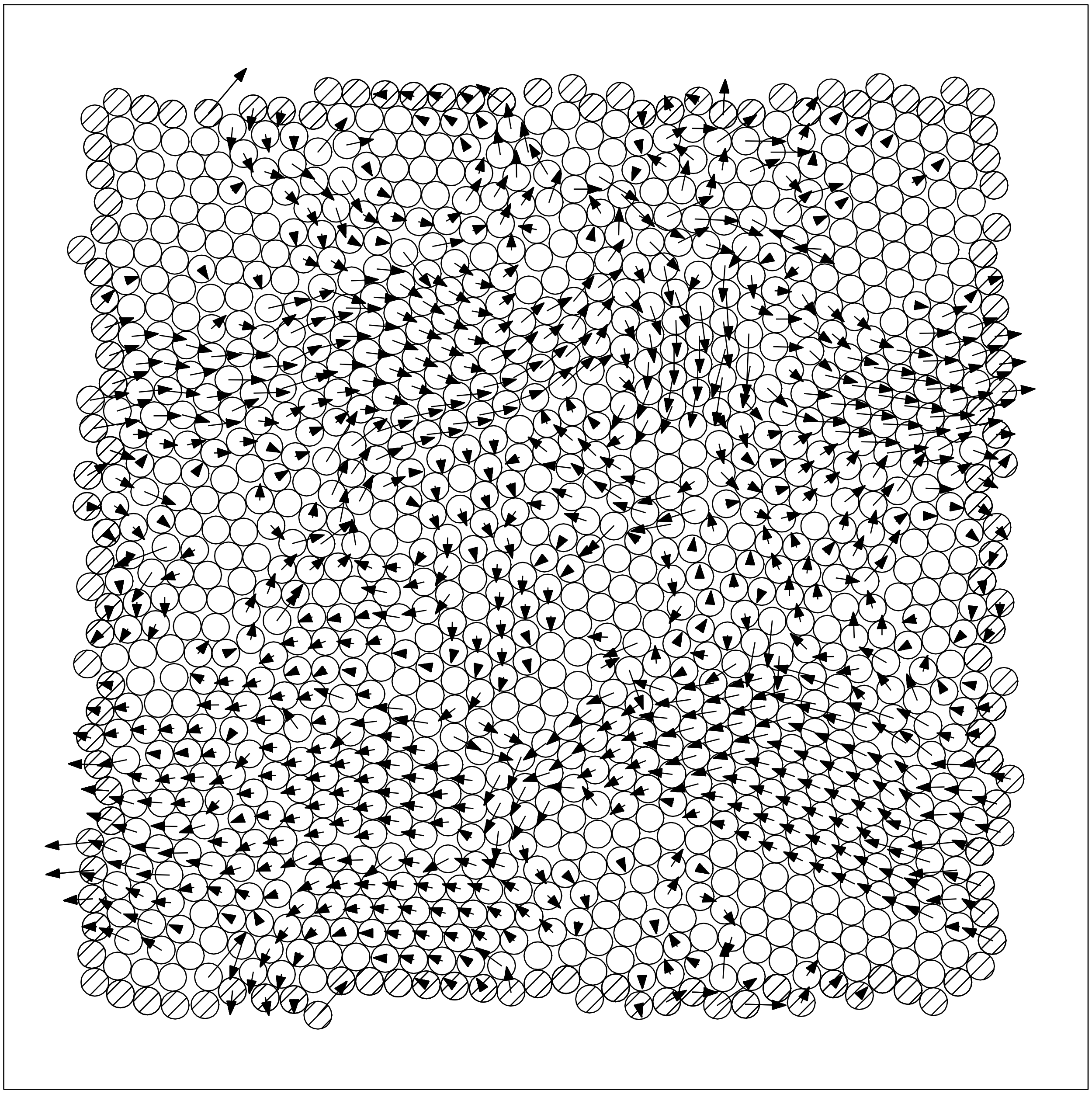

Figure 11: Donev *et al.*

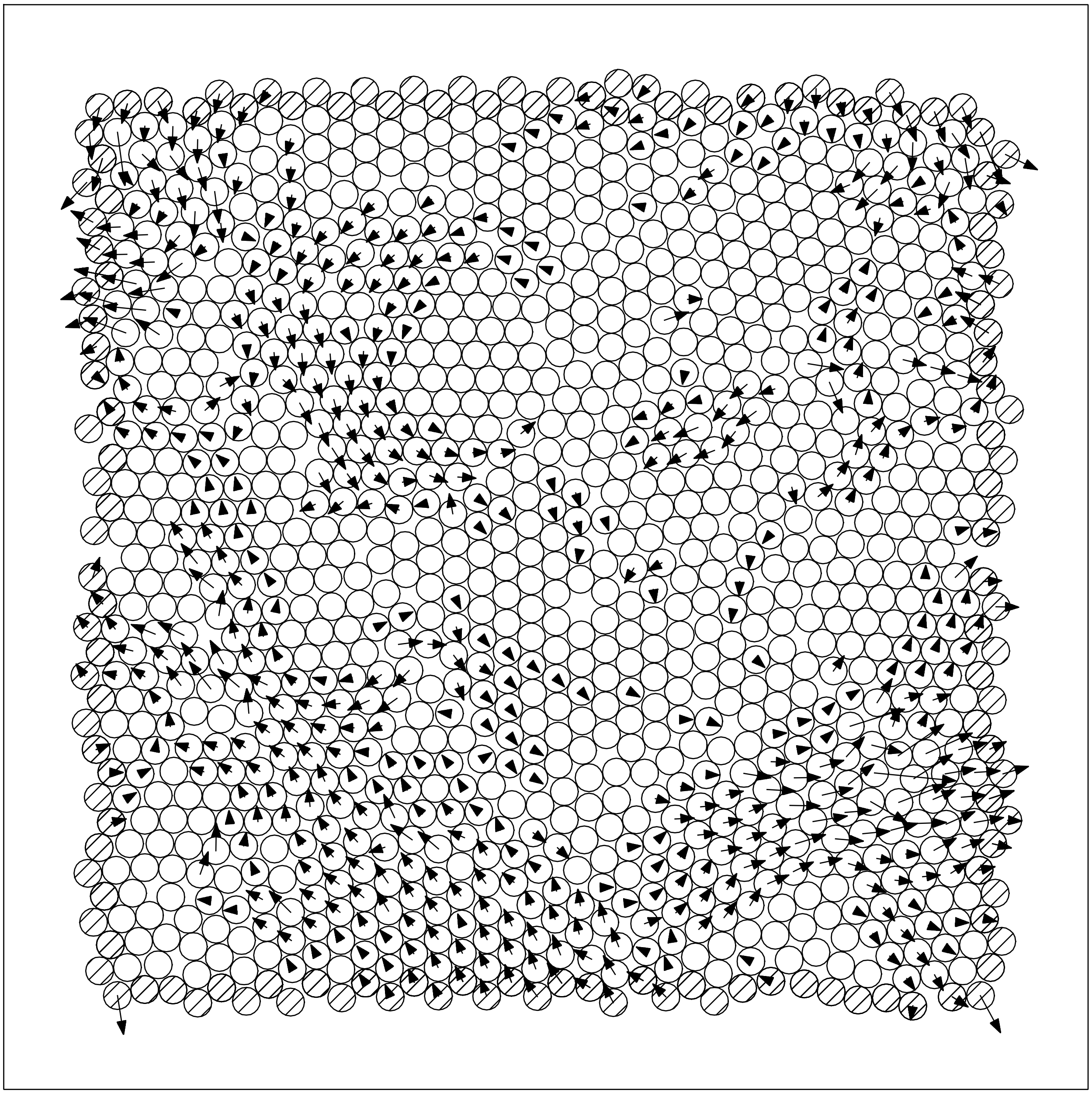

Figure 12: Donev *et al.*

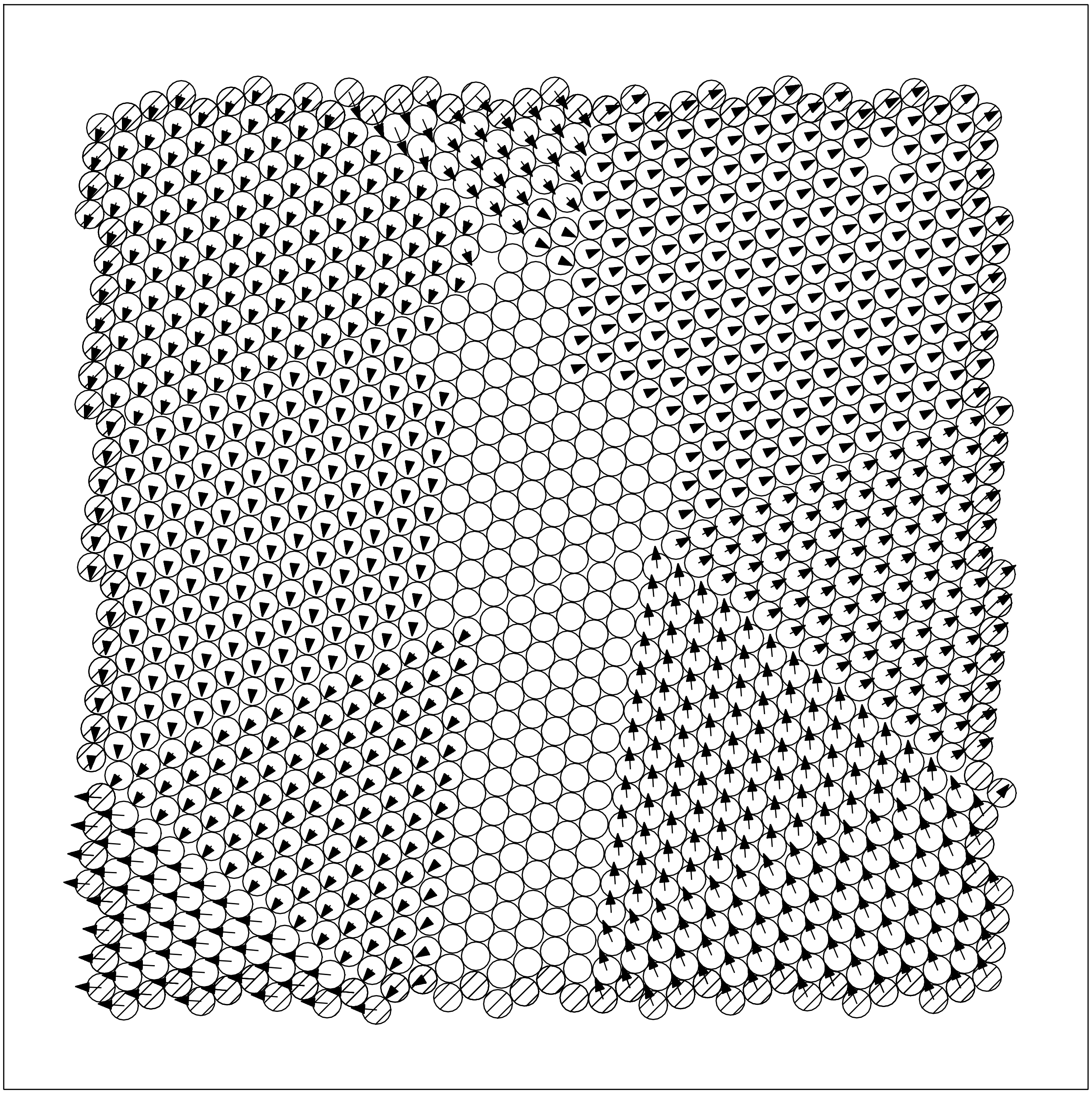

Figure 13: Donev *et al.*