Hard-core configurations
on a triangular lattice
and Eisenstein primes

A. Mazel\textsuperscript{1}, I. Stuhl\textsuperscript{2}, Y. Suhov\textsuperscript{2−4}

Abstract

We study the Gibbs statistics of high-density hard-core random configurations
on a triangular lattice. Depending on certain arithmetic properties of the repulsion
diameter $D$ (related to Eisenstein integers), we identify, for a large fugacity, the
extreme periodic Gibbs measures and analyze their properties. (a) For the values of
$D$ belonging to certain arithmetic classes a complete phase diagram is established.
(b) For the remaining values of $D$ we prove non-uniqueness of a pure phase and
provide some additional information. We argue that in general the list of extreme
periodic Gibbs measures can vary according to the arithmetic structure of $D$. This
argument is supported by the analysis of several specific values of $D$ (outside the
classes from (a)) for which the complete phase diagram is established, using in part
a computer assistance. The proofs are achieved by applying Zahradník’s extension
of the Pirogov–Sinai theory.

1 Introduction

High-density hard-core configurations are of interest in a number of theoretical and
applied disciplines, including Pure Mathematics (number theory, algebraic geometry),
Applied Mathematics (computer/information sciences), Mathematical and Theoretical
Physics (as an interesting example of a phase transition), Theoretical Chemistry and
Theoretical Biology (as a model of allelopathy). Notable examples are configurations of
a maximal density in a plane \cite{5}–\cite{8}, 3D-space \cite{7} and higher dimensions \cite{15}, \cite{2} (see also
the biblio therein).

In this paper we focus on random hard-core configurations on a triangular lattice $L$
in a high-density/large-fugacity regime. The hard-core exclusion is imposed in the
Euclidean metric and is described via the hard-core diameter $D$. Without loss of generality

\textit{2010 Mathematics Subject Classification:} primary 60G60, 82B20
\textit{Key words and phrases:} triangular lattice, hard-core configuration, Eisenstein prime, Gibbs measure,
high density, ground state, contour representation of the partition function, Peierls bound, Pirogov-Sinai
theory, stable ground state, Zahradník’s argument, inclined and non-inclined sub-lattices, computer-assisted enumeration

\textsuperscript{1} AMC Health, New York, NY, USA;
\textsuperscript{2} Math Department, Penn State University, PA, USA;
\textsuperscript{3} DPMMS, University of Cambridge and St John’s College, Cambridge, UK;
\textsuperscript{4} IITP RAS, Moscow, RF

\vspace{1cm}

1
we assume that the value $D$ is attainable, meaning that $D^2$ admits the representation $D^2 = a^2 + b^2 + ab$, where $a$ and $b$ are integers. In the literature, the sums $a^2 + b^2 + ab$ are known as Löschian numbers (see A003136 in the on-line Encyclopedia of integer sequences). We use some classical results regarding these integers which are presented in a convenient form in [10, 11]. The emerging probability measures on $(0, 1)^2$ describing random configurations are identified as Gibbs, or DLR, measures. Their structure appears to be dependent on arithmetic properties of $D^2$ (for the arithmetic properties of lattices see the monograph [4]).

We distinguish the following 3 cases.

- **Case 1:** Either $D$ or $D/\sqrt{3}$ is an integer whose prime decomposition does not contain factors of the form $6k + 1$. That is, each prime factor of $D$ or $D/\sqrt{3}$ is either 3 or an Eisenstein prime. The first fifty values of $D^2$ falling in this category are $D^2 = 1, 3, 4, 9, 12, 16, 25, 27, 36, 48, 64, 75, 81, 100, 108, 121, 144, 192, 225, 243, 256, 289, 300, 324, 363, 400, 432, 484, 529, 576, 625, 675, 729, 768, 841, 867, 900, 972, 1024, 1089, 1156, 1200, 1296, 1452, 1587, 1600, 1681, 1728, 1875, 1936.

- **Case 2:** $D^2$ is an integer whose prime decomposition contains (i) a single prime of the form $6k+1$, and (ii) other primes, if any, in even powers, except for the prime 3. (The factor 3 can enter with any power.) The first fifty values of $D^2$ with this property are $7, 13, 19, 21, 28, 31, 37, 39, 43, 52, 57, 61, 63, 67, 73, 76, 79, 84, 93, 97, 103, 109, 111, 112, 117, 124, 127, 129, 139, 148, 151, 156, 157, 163, 171, 172, 175, 181, 183, 189, 193, 199, 201, 208, 211, 219, 223, 228, 229, 237.

- **Case 3:** All remaining attainable values of $D$. Here we have $D^2 = 49, 91, 133, 147, 169, 196, 217, 247, 259, 273, 301, 343, 361, 364, 399, 403, 427, 441, 469, 481, 507, 511, 532, 553, 559, 588, 589, 637, 651, 676, 679, 703, 721, 741, 763, 777, 784, 793, 817, 819, 868, 871, 889, 903, 931, 949, 961, 973, 988, for the initial list of values of this type.

Correspondingly, we refer to the value of $D$ of type 1, type 2 and type 3.

We prove that for large fugacities there are exactly $D^2$ extreme space-periodic Gibbs measures (pure phases) in Case 1 and exactly $2D^2$ in Case 2. See Theorems 1, 2 in Section 4. Following the canons of the Pirogov–Sinai (PS) theory [12], the above measures are constructed by means of polymer expansions around the corresponding ground states, i.e., dense-packing configurations. A symmetry between ground states plays a crucial role in our considerations and implies a similar symmetry between the corresponding Gibbs measures. In Case 1 the extreme periodic Gibbs measures are obtained from each other by lattice shifts. In Case 2 they are obtained from each other by lattice shifts and reflections.

For a general $D$ belonging to the Case 3, we provide a somewhat reduced information on Gibbs measures; see Theorem 3 in Section 5. For specific values $D^2 = 49, 147, 169$ the situation is clarified to the level of Theorems 1 and 2; see Theorems 4, 5, and 6 in Section 5. The approach adopted in Theorems 4, 5, 6 is applicable to any given value $D$ of type 3 but the amount of calculations needed to be performed grows rapidly with $D$.

The property enabling us to achieve these results is that the ground states are ‘rigid’: each of them is uniquely determined by the position of few particles, one in Case 1 and two in other cases. A similar model on a square lattice $\mathbb{Z}^2$ behaves differently: here the
ground states may form a continuous family (a ‘sliding’ phenomenon) which prevents the use of the PS theory. (In fact, a natural hypothesis is that on $\mathbb{Z}^2$, when the hard-core diameter admits sliding, the large-fugacity Gibbs measure is unique.) Cf. [9].

The rigidity property of the ground states on $\mathbb{L}$, together with an appropriate definition of a *contour* [12], leads to a convenient representation of the partition function. In turn, this implies straightforward *Peierls bounds* and allows us to use a well-established Zahradník’s argument [10] without major changes and provide a short proof. We also exploit the fact that a dense-packing configuration of circles in $\mathbb{R}^2$ is a triangular array [5] [8].

2 Preliminaries and basic facts

**Particle/spin configurations.** In this paper we refer to a two-dimensional triangular lattice $\mathbb{L}$ as a set in $\mathbb{R}^2$ formed by points $x = (x_1; x_2)$ (sites of the lattice) with Euclidean co-ordinates $x_1 = m + \frac{1}{2}n$ and $x_2 = \sqrt{3}n$, where $m, n \in \mathbb{Z}$. Accordingly, we write $x \simeq (m, n) \in \mathbb{L}$. Every site $x \in \mathbb{L}$ has six neighboring sites $x'$ such that the distance $\rho(x, x')$ equals 1. Here $\rho(= \rho_2)$ stands for the 2D Euclidean metric: for $x = (x_1; x_2), y = (y_1; y_2) \in \mathbb{R}^2$, the distance $\rho(x, y) = \|p_1(x_1, y_1)^2 + p_1(x_2, y_2)^2\|^{1/2}$, where $p_1(x, y) = |y - x|$, $x, y \in \mathbb{R}$. Given $u \in \mathbb{L}$, we designate $S_u : \mathbb{L} \to \mathbb{L}$ to be a lattice shift by $u$, with $S_u x = x + u, x \in \mathbb{L}$. Similarly, $R : \mathbb{L} \to \mathbb{L}$ stands for the reflection map about the lattice diagonal: $Rx \simeq (n, m)$ when $x \simeq (m, n) \in \mathbb{L}$.

We fix a value $D \geq 1$ and consider admissible configurations $\phi : x \in \mathbb{L} \mapsto \phi_L(x) \in \{0, 1\}$ (or shortly $\phi : \mathbb{L} \to \{0, 1\}^\mathbb{L}$) such that for any two occupied sites $x$ and $y$ with $\phi(x) = \phi(y) = 1$ the distance $\rho(x, y) \geq D$. (We can think that $\phi(x) = 1$ means site $x$ is occupied by a ‘physical’ particle.) The value $D$ is interpreted as a hard-core diameter or hard-core distance. The set of admissible configurations is denoted by $A(=A(D))$.

Similar definitions can be introduced for a subset $V \subset \mathbb{L}$; in particular, the set of admissible configurations in $V$ is denoted by $A_V(=A_V(D))$. The restriction of configuration $\phi \in A$ to $V$ is denoted by $\phi |_V$.

As was said in Introduction, we can assume that

$$D^2 \in \mathbb{N} \quad \text{and} \quad D^2 = a^2 + b^2 + ab \quad \text{where} \quad a, b \in \mathbb{Z}; \quad (2.1)$$

physically, it means that $D$ is attainable, i.e., there are sites $x, y \in \mathbb{L}$ with $\rho(x, y) = D$. Assumption (2.1) does not restrict generality, as any other $D'$ can be replaced by the smallest $D \geq D'$ satisfying (2.1) without changing the set $A$.

The partition function with empty boundary conditions. For a finite $V \subset \mathbb{L}$ the corresponding partition function with empty boundary conditions is defined as

$$Z(V; u) = \sum_{\phi_V \in A_V} \prod_{x \in V} u^{\phi_V(x)}; \quad (2.2)$$

here $u > 0$ is the fugacity of an occupied lattice site. The summand $\prod_{x \in V} u^{\phi_V(x)}$ is called the statistical weight of admissible configuration $\phi_V$.

Despite a straightforward (and appealing) form of the partition function $Z(V; u)$ from (2.2), it is not always convenient (or at least not the most convenient) for the rigorous
analysis in the thermodynamic limit (for a sequence of ‘volumes’ $V_k$, with $V_{k+1} \supset V_k$ and $\bigcup V_k = \mathbb{L}$). The corresponding limit Gibbs distribution (if it exists) depends on the particular shape of volumes $V_k$ which can be in a ‘good’ or ‘bad’ agreement with symmetries of the hard-core model on $\mathbb{L}$. In this paper we concentrate on the partition function $Z_\varphi(V, u)$ with a ground-state boundary condition $\varphi$; see equation (2.4) below. We also analyse a periodic version of (2.2). Cf. [3].

Ground states and arithmetic properties of $D$. Consider an arbitrary set $E \subset \mathbb{R}^2$ such that $\rho(x, y) \geq D$ for any two distinct points $x, y \in E$. For each $x \in E$ define a local (Voronoi) cell $C_E(x)$ as the set of points $z \in \mathbb{R}^2$ satisfying $\rho(x, z) \leq \rho(y, z) \forall y \in E \setminus \{x\}$.

It is known [3] that a local cell with the minimal possible area is a perfect hexagon with the side length $\frac{D}{\sqrt{3}}$ and area $S = \frac{D^2 \sqrt{3}}{2}$. We call it a perfect local cell. A set of points $E \subset \mathbb{R}^2$ is called perfect if it contains only perfect local cells. The only perfect $E \subset \mathbb{R}^2$ are triangular lattices with the distance $D$ between neighboring lattice sites (see again [3]).

Lemma 2.1 For each $D \geq 1$ with $D^2 \in \mathbb{N}$ there exist a number $s(D) > 0$ such that the minimal area of non-perfect local cell is larger than $S + s(D)$.

Proof. A perfect cell is defined by the occupied site, say the origin, and 6 occupied sites situated at distance $D$ from the origin. A direct consequence of results in [3] is that the next-to-optimal local cell is given by a shift to some neighboring lattice site of one of 6 occupied sites above.

We focus on the case where $E$ is the set of all occupied sites in an admissible configuration $\phi_L$, which is denoted by $E(\phi_L) \subset \mathbb{L}$. Of a particular interest are dense-packing admissible configurations $\varphi \in A$, for which every occupied site $x \in E(\varphi)$ has exactly six sites $y \in E(\varphi)$ with $\rho(x, y) = D$. The collection of dense-packing configurations is denoted by $D(=D(D))$. In terms of the PS theory, every $\varphi \in D(D)$ is a ground state for the hard-core diameter $D$ (see Figure 1), provided that $u > 1$. It means that there is no admissible $\phi \in A(D)$ differing from $\varphi$ at finitely many sites with the property that $\phi$ has the same or larger amount of occupied sites counted inside any large volume that contains all sites $x$ where $\phi(x) \neq \varphi(x)$.

Given an admissible configuration $\phi \in A(D)$ and a dense-packing configuration $\varphi \in D(D)$, we say that a site $x \in \mathbb{L}$ is $\varphi$-correct in $\phi$ if $\phi(x) = \varphi(x) = 1$ and $\phi(y) = \varphi(y) = 1 \forall y \in \mathbb{L}$ with $\rho(x, y) = D$. This is a step towards the specification of the definition in [16], P. 561, for our model.

Now we turn to arithmetic properties of $D$. Any ordered pair of integers $(a, b)$ which is a solution to equation (2.1) defines a sublattice of $\mathbb{L}$ containing the origin and the following 6 sites:

$$(a, b); (-b, a + b); (-a - b, a); (-a, -b); (b, -a - b); (a + b, -a),$$ (2.3)

which all are the solutions to (2.1) as ordered pairs of integers. If $a \neq b, a \neq 0, b \neq 0$ then the pair $(b, a)$ also defines a sublattice of $\mathbb{L}$ which is a reflection (by $\mathbb{R}$) of the sublattice defined by $(a, b)$ (cf. Case 2). If $a = b$ or $ab = 0$ then the pair $(a, b)$ defines a single
Figure 1: Fragments of ground-state configurations $\varphi$ for $D = \sqrt{3}$ (diagonal), $D = 2$ (non-inclined) and $D = \sqrt{7}$ (inclined). Note that for $D = \sqrt{7}$ there are two such configurations.

sublattice of $\mathbb{L}$ which is mapped into itself under the reflection $R$ (cf. Case 1). For each sublattice of $\mathbb{L}$ generated by a solution to (2.1) there are exactly $D^2$ distinct shifts $S_u$ as there are exactly $D^2$ lattice sites inside the fundamental parallelogram of the sublattice. All shifted configurations are also ground states. Moreover, all space-periodic ground states corresponding to the given $D$ are obtained as a shift of a sublattice generated by a solution to (2.1).

Consider Case 1. For an integer $D$, the theory of Eisenstein integers (see, e.g., [10]) implies that this is precisely the case where the equation (2.1) has only trivial solutions $a = D, b = 0$ and $a = 0, b = D$. An alternative situation is where $D/\sqrt{3}$ is integer: then $a = b$ in (2.1). In Case 1 the set $\mathcal{D}(D)$ of ground states contains exactly $D^2$ configurations, and they are lattice shifts of each other. (We will stress this fact by saying that there is lattice-shift symmetry between the ground states.) More precisely, the ground states are shifts of a dense-packing configuration $\varphi^0$ for which set $E(\varphi^0)$ represents either a non-inclined $D$-sublattice with $E(\varphi^0) = \{(Dm, Dn)\}$ or a diagonal $D$-sublattice where $E(\varphi^0) = \{D(m - n)/\sqrt{3}, D(m + 2n)/\sqrt{3}\}$, where $(m, n) \in \mathbb{L}$. In addition, each $\varphi \in \mathcal{D}(D)$ is $R$-invariant.

In Case 2, again invoking the theory of Eisenstein integers [10], equation (2.1) has a unique non-trivial solution, where $a, b \in \mathbb{N}$ and $a > b \geq 1$. In this case there are exactly two inclined sublattices, $E(\varphi^0) = \{(m(a + b) + nb, mb - na)\}$ and $E(R\varphi^0) = \{(m(a + b) + na, ma - nb)\}$, which are mapped into each other by the reflection $R$. Accordingly, the ground states are organized into pairs where the members of a pair are images of each other under the reflection $R$. The number of such pairs equals $D^2$, and the pairs are transformed to each other by lattice shifts. The cardinality of $\mathcal{D}(D)$ in this case equals $2D^2$.

The fundamental parallelograms for different sublattices are different but common
parallelograms

\[ F_{k,l} = \{ (m, n) \in \mathbb{L} : kD^2 \leq m < (k+1)D^2, lD^2 \leq n < (l+1)D^2 \}, \quad k, l \in \mathbb{Z}, \quad (2.4) \]

are independent of the choice of a sublattice. The family \( \{ F_{k,l} \} \) forms a partition of \( \mathbb{L} \).

The parallelogram \( F_{0,0} \), treated as a \( D^2 \times D^2 \)-torus, is partitioned into \( D^2 \) congruent parallelograms (one of which is a fundamental parallelogram for a sublattice), one partition for each of the sublattices. Accordingly, for each of the sublattices there are exactly \( D^2 \) sublattice sites inside every \( F_{k,l} \). We frequently omit the indices \( k, l \) in the notation of the common parallelogram when their values are not important or are evident from the context. Figure 2 demonstrates examples of common parallelograms.

![Figure 2: Common (gray) and fundamental (thick lines) parallelograms for \( D = \sqrt{7} \). Marked sites belong to the common parallelogram. The larger mark identifies the origin.](image)

Applying the definition of correctness on p. 561 in [16] we say that a common parallelogram \( F_{k,l} \) is \( \varphi \)-correct in \( \phi \) if \( \phi(x) = \varphi(x) \) for every site \( x \) lying in 9 common parallelograms \( F_{k+i,l+j} \), where \( i, j = -1, 0, 1 \). Figure 3 demonstrates various types of common parallelograms.

**Partition functions with dense-packing boundary conditions.** Suppose that \( V \subset \mathbb{L} \) is a finite union of common parallelograms; such a set \( V \) is called a basic lattice polygon. The partition function with the boundary condition \( \varphi \in \mathcal{D} \) is defined as

\[
Z_{\varphi}(V, u) = \sum_{\phi_V \in \mathcal{A}_{V, \varphi}} \prod_{x \in V} u^{\phi_V(x) - \varphi(x)}. \quad (2.5)
\]

Here the sum extends over the set \( \mathcal{A}_{V, \varphi} \subset \mathcal{A}_V \) of admissible configurations \( \phi_V \) in \( V \) which are compatible with the boundary condition \( \varphi \). The latter means that in the combined configuration \( \phi_V \lor \varphi \rvert_{\mathbb{L}\setminus V} \) over \( \mathbb{L} \) all not \( \varphi \)-correct common parallelograms are located inside \( V \). This definition of compatibility is more restrictive then just requiring that \( \phi_V \lor \varphi \rvert_{\mathbb{L}\setminus V} \in \mathcal{A} \). Essentially, we require that all occupied sites in \( \phi_V \) which are at distance \( D^2 \) or less from \( \mathbb{L} \setminus V \) belong to \( E(\varphi) \). In contrast with (2.2), the statistical weight of a configuration \( \phi_V \) in (2.5) is measured relatively to that of \( \varphi \rvert_{V} \). The definition (2.5) is a special case of Eqn (1.12) in [16].

**Contours.** A contour support in a configuration \( \phi \in \mathcal{A} \) is defined as an \( \mathbb{L} \)-connected component of the union of common parallelograms that are not \( \varphi \)-correct in \( \phi \) for any \( \varphi \in \mathcal{D} \). A contour in \( \phi \) is defined as a pair \( \Gamma = (\text{supp} \Gamma, \phi \rvert_{\text{supp}}) \) consisting of a contour
Figure 3: Common parallelograms which are $\varphi$-correct (light gray) and not $\varphi$-correct (medium and dark gray) in the admissible configuration $\phi$ for $D = \sqrt{7}$. Black balls indicate occupied sites in $\phi$. White balls indicate not occupied sites in $\phi$ which would be occupied in $\varphi$. The light gray indicates $\varphi$-correct parallelograms. The dark gray indicates not $\varphi$-correct parallelograms which contain some defects ($\phi \neq \varphi$). For the medium gray parallelograms $\phi = \varphi$ but these parallelograms are still not $\varphi$-correct as they have a neighboring parallelogram with defects.

Figure 4: A contour support (the union of gray and dark gray common parallelograms). Here the internal area $\text{Int}(\Gamma)$ includes three components $\text{Int}_{\varphi_i}(\Gamma)$. The boundary layers are shown as the union of gray common parallelograms.
support suppΓ and the restriction φ | suppΓ. These definitions are specifications of general definitions on P. 561 in [16] for our model. Accordingly, we define sets intΓ, intϕΓ and extΓ by using Eqn (1.5) from [16].

Inside each extΓ, suppΓ, and intϕΓ we can specify a boundary layer: it is an L-connected set of common parallelograms with each parallelogram in the set having a neighboring common parallelogram outside of the corresponding extΓ, suppΓ, or intϕΓ. Each of extΓ and intϕΓ has a single corresponding boundary layer while suppΓ has several of them. Every boundary layer in suppΓ has a corresponding (dual) boundary layer inside extΓ or intϕΓ. Moreover, in every boundary layer all occupied sites belong to the same E(ϕ) (which justifies the notation intϕΓ). Finally, following Eqn (1.5) from [16], a contour for which a boundary layer of extΓ belongs to the ground state ϕ is called a ϕ-contour. See Figure 5.

3 The Peierls bound

For any basic lattice polygon V ⊂ L and configurations ϕV ∈ D and ϕV ∈ AV,ϕ we have the identity

$$\bigcup_{x \in E(\phi_V)} C_{E(\phi_V)}(x) = \bigcup_{x \in E(\phi_{V|V})} C_{E(\phi_{V|V})}(x),$$

(3.1)

where the RHS is obtained from the LHS by replacing of ϕV with ϕ_{V|V}. Therefore, for the partition function (2.5) we can write

$$Z_\phi(V, u) = \sum_{\phi_V \in A_{V,\phi}} \prod_{x \in E(\phi_V)} u^{-S^{-1}(|C_{E(\phi_V)}(x)| - S)}.$$  

(3.2)

Here, and in what follows, |C_{E(\phi_V)}(x)| stands for the area of polygon C_{E(\phi_V)}(x), and S = \sqrt{3/2}D^2 is the area of the perfect hexagon.

**Lemma 3.1 (The Peierls Bound)** Define the statistical weight w(Γ) of a contour Γ by

$$w(\Gamma) = \prod_{x \in E(\phi_\Gamma)} u^{-S^{-1}(|C_{E(\phi_\Gamma)}(x)| - S)}.$$  

(3.3)

Here and below, \phi_\Gamma stands for the restriction φ | suppΓ. Then

$$w(\Gamma) \leq u^{-p(D)||supp\Gamma||}$$

(3.4)

where p(D) > 0 and ||supp\Gamma|| is the number of lattice sites in suppΓ.

**Proof.** Observe that

if |C_{E(\phi_\Gamma)}(x)| - S ≥ S then |C_{E(\phi_\Gamma)}(x)| - S ≥ \frac{1}{2}|C_{E(\phi_\Gamma)}(x)|.
On the other hand, by the Lemma 2.1,

\[
\text{if } |C_{E(\varphi^T)}(x)| - S < S \text{ then } |C_{E(\varphi^T)}(x)| - S \geq s(D) \geq \frac{s(D)}{2S} |C_{E(\varphi^T)}(x)|.
\]

According to the definition of a \(\varphi\)-correct common parallelogram, we have an inequality

\[
\sum_{x \in E(\varphi^T)} |C_{E(\varphi^T)}(x)| \geq \frac{1}{gD^2} |\mathrm{supp}\Gamma|. \quad \text{Also, } ||\mathrm{supp}\Gamma|| = \frac{1}{\sqrt{3}} |\mathrm{supp}\Gamma|.
\]

Thus, we can take \(p(D) = \frac{2}{9D^2\sqrt{3}} \min\left(\frac{1}{2}, \frac{s(D)}{\sqrt{3}}D^{-2}\right).\)

**Remark.** The value \(p(D) = \frac{2}{9D^2\sqrt{3}} \min\left(\frac{1}{2}, \frac{s(D)}{\sqrt{3}}D^{-2}\right)\) is by no means optimal and is chosen here for simplicity.

In the PS theory, the Peierls bound is used for assessing the probability of a contour, which in turn is used to distinguish extreme Gibbs measures generated by different ground state boundary conditions. Note that taking the logarithm of both sides in (3.4) transforms bound (3.4) into Eqn (1.9) from [16].

### 4 The limit Gibbs measures: Cases 1 and 2

Given a ground state \((\text{GS}) \varphi \in \mathcal{D}\) and a basic lattice polygon \(V \subset \mathbb{L}\), the partition function \(Z_\varphi(V, u)\) in (2.5) and (3.2) gives rise to a Gibbs probability distribution \(\mu_{V, \varphi}\) on \(\{0, 1\}^V\) concentrated on \(A_{V, \varphi}\). In a standard way, \(\mu_{V, \varphi}\) is treated as a probability distribution on \(\{0, 1\}^L\), and we can consider measures that are weak limit points for \(\mu_{V_k, \varphi}\) where \(V_k\) is an increasing sequence of basic lattice polygons. Such measures are examples of Gibbs, or DLR, measures for the hard-core model on \(\mathbb{L}\) with diameter \(D\); they sit on the set \(\mathcal{A}\). If for a given \(\varphi \in \mathcal{D}\) there exists the limiting measure \(\mu_\varphi = \lim_{k \to \infty} \mu_{V_k, \varphi}\), common for any basic lattice polygon sequence \(V_k\), we say that \(\varphi\) generates a Gibbs measure (GM) \(\mu_\varphi\). For the theory of GMs, see the monograph [G], Chapters 3,4, 5–8. In particular, we refer to the set \(\mathcal{G} = \mathcal{G}(D)\) of GMs sitting on \(\mathcal{A}(D)\); geometrically, \(\mathcal{G}\) is a Choquet simplex. Physically, measures \(\mu \in \mathcal{G}(D)\) describe random hard-core configurations on \(\mathbb{L}\) with diameter \(D\). An extreme point \(\nu\) of \(\mathcal{G}\) is interpreted as a pure phase for the random hard-core model; in the situations occurring in Theorems 1 − 6 below it will be a measure with an exponential decay of correlation (exponential mixing). We also write \(S_u \nu\) for the lattice shift of a measure \(\nu\) on \(\{0, 1\}^L\) and \(R \nu\) for the image of \(\nu\) under the reflection \(R\).

A specific construction of a GM exploits periodic boundary conditions, in toric volumes \(V_k = T_k, k = 1, 2, \ldots\)

\[
T_k = \{(m, n) \in \mathbb{L} : -kD^2 \leq m < kD^2, -kD^2 \leq n < kD^2; \text{ with indentionition (}kD^2, n) \equiv (-kD^2, n) \text{ and (}m, kD^2) \equiv (m, -kD^2)\}. \quad (4.1)
\]

To determine the admissible configurations in a torus we use the condition that \(\rho^{(k)}(x,y) \geq D\). Here the metric \(\rho^{(k)}\) is defined by \(\rho^{(k)}(x,y) = \left[\rho^{(k)}_1(x_1,y_1)^2 + \rho^{(k)}_2(x_2,y_2)^2\right]^{1/2}\), where \(x = (x_1, x_2), y = (y_1, y_2)\). In turn, \(\rho^{(k)}_1\) is a metric on the interval \([-kD^2, kD^2]\), with
\( \rho_1^{(k)}(x, y) = \min \{ y - x, x + 2kD^2 - y \} \) for \(-kD^2 \leq x \leq y < kD^2\). The set of admissible configurations in \( T_k \) is denoted by \( A_{\text{per}, k} \). In the same spirit as \((2.2)\) and \((2.5)\), the partition function in \( T_k \) with periodic boundary condition is determined by

\[
Z_{\text{per}}(T_k, u) = \sum_{\phi_{T_k} \in A_{\text{per}, k}} \prod_{x \in T_k} u^{\phi_{T_k}(x)}. \tag{4.2}
\]

This in turn defines the Gibbs distribution \( \mu_{\text{per}, k} \). The limiting measure \( \lim_{k \to \infty} \mu_{\text{per}, k} \), if it exists, is denoted by \( \mu_{\text{per}} \).

Our results in this section are the following two theorems.

**Theorem 1.** Let \( D \) or \( D/\sqrt{3} \) be an integer not divisible by primes of the form \( 6k+1 \) (Case 1). There exists \( u_0 = u_0(D) \in (1, \infty) \) such that for \( u > u_0(D) \) the assertions (a–c) below hold true.

- (a) Each GS \( \varphi \in \mathcal{D} \) generates a GM \( \mu_\varphi \) which is a pure phase. The measures \( \mu_\varphi \) are pair-wise distinct. However, if \( S_u \varphi = \varphi' \) then \( S_u \mu_\varphi = \mu_{\varphi'} \). In particular, each measure \( \mu_\varphi \) is \( D \)-periodic relative to lattice shifts.

- (b) Any shift-periodic GM \( \mu \in \mathcal{G} \) is a convex combination of measures \( \mu_\varphi, \varphi \in \mathcal{D} \).

- (c) The GM \( \mu_{\text{per}} \) exists for sets \( V_k = T_k \); cf. \((1.2)\). It is a uniform mixture of the \( D^2 \) measures \( \mu_\varphi, \varphi \in \mathcal{D} \). Consequently, measure \( \mu_{\text{per}} \) is shift-invariant.

**Theorem 2.** Let \( D^2 \) be an integer whose prime decomposition contains a single prime of the form \( 6k+1 \) (Case 2). There exists \( u_0 = u_0(D) \in (1, \infty) \) such that for \( u > u_0(D) \) the following assertions (a–c) hold true.

- (a) Each of the \( 2D^2 \) GSs \( \varphi \in \mathcal{D}(D) \) generates a GM \( \mu_\varphi \) which is a pure phase. The measures \( \mu_\varphi \) are pair-wise distinct. However, if \( S_u \varphi = \varphi' \) then \( S_u \mu_\varphi = \mu_{\varphi'} \) and if \( R \varphi = \varphi' \) then \( R \mu_\varphi = \mu_{\varphi'} \). In particular, each measure \( \mu_\varphi \) is \( D \)-periodic relative to lattice shifts.

- (b) Any shift-periodic GM \( \mu \in \mathcal{G} \) is a convex combination of measures \( \mu_\varphi, \varphi \in \mathcal{D}(D) \).

- (c) The GM \( \mu_{\text{per}} \) exists for \( V_k = T_k \); cf. \((4.2)\). It is a uniform mixture of the \( 2D^2 \) measures \( \mu_\varphi, \varphi \in \mathcal{D}(D) \). Therefore, measure \( \mu_{\text{per}} \) is shift- and reflection-invariant.

**Proof of Theorem 1.** Assertion (a) follows from Corollary on P. 565 in \[16\]. Assertion (b) follows from Corollary on P. 578 in \[16\]. In both cases the argument relies on the observation that, because of the shift-symmetry between the GSs, the partition functions \((2.5)\) do not depend on the specific choice of the GS \( \varphi \) used as the boundary condition. Therefore, all \( \varphi \)-contours \( \Gamma \) are stable in a sense of Definition 3, P. 564, in \[16\], \( \forall \varphi \in \mathcal{D} \). The Eqn (1.9) in \[16\] follows from \((3.4)\) and the rest of the requirements in \[16\] are automatically satisfied for the partition function \((2.5)\).
Passing to case (c), the main difference is the existence of contours winding around the torus \( T_k \). However, the \( \mu_{\text{per},k}^{(1)} \) probability of the event \( \mathcal{W}^{(1)}_k \subset A^{(1)}_{\text{per},k} \) that such contour is present in an admissible configuration \( \phi^{(1)}_k \) becomes negligible as \( k \to \infty \), as winding contours are too large. On the remaining event, \( \overline{\mathcal{W}}^{(1)}_k = A_{\text{per},k} \setminus \mathcal{W}^{(1)}_k \), the statistics of the random configuration is described in terms of the ensemble of external contours. Furthermore, by the shift-symmetry, event \( \overline{\mathcal{W}}^{(1)}_k \) can be partitioned into \( D^2 \) parts, \( \overline{\mathcal{W}}^{(1)}_{k,\varphi}, \varphi \in \mathcal{D} \), of equal probability \( \frac{1}{D^2} \left[ 1 - \mu_{\text{per},k}^{(1)}(\mathcal{W}^{(1)}_k) \right] \), so that on \( \overline{\mathcal{W}}^{(1)}_{k,\varphi} \) all external contours are \( \varphi \)-contours. This leads to the formula \( \lim_{k \to \infty} \mu_{\text{per},k}^{(1)} = \frac{1}{D^2} \sum_{\varphi \in \mathcal{D}} \mu_\varphi \). ■

**Proof of Theorem 2.** The difference between Theorems 1 and 2 is that for Theorem 2 we do not have a lattice-shift symmetry between all GSs. Therefore we can’t claim that the partition function \( Z_\varphi(V, u) \) defined in (2.5) (which corresponds to Eqns (1.12) and (1.18) in [16]) is independent of the boundary condition \( \varphi \in \mathcal{D} \). Nevertheless, given a \( \varphi \)-contour \( \Gamma \), the corresponding crystallic partition function (see (1.10) and (1.19) in [16]) remains intact under lattice shifts \( S_u \) or reflection \( R \): these maps transform \( \Gamma \) into a \( \varphi' \)-contour \( \Gamma' \) for some other GS \( \varphi' \). (Recall that the lattice \( \mathbb{L} \) is transformed into itself under these maps, implying that \( S_u \) and \( R \) take \( \mathcal{A} \) to \( \mathcal{A} \).)

To exploit the invariance of crystallic partition functions, we need to go one step deeper into the analysis of truncated contour models (see section 1.7 in [16]) and the corresponding polymer models (see section 2.1 in [16] and also section I.3.a in [13]). For any GS \( \varphi \) the polymers contributing to the polymer series for the free energy of the corresponding truncated model are constructed from \( \varphi \)-contours. For each contributing polymer, a similar series for any other ground state \( \varphi' \) contains either exactly the same polymer (if \( S_u \varphi = \varphi' \) for some \( u \)) or a reflected polymer (if \( RS_u \varphi = \varphi' \) for some \( u \)). Similarly to lattice shifts, the reflection does not change the statistical weight of the polymer as for each constituting contour the crystallic partition functions coincide for the original and the reflected contours. Therefore, at each step of the inductive (in the size of \( V_k \)) proof of Main Lemma in [16] (see Section 2.2 in [16]) we obtain that the truncated free energies for all GSs are the same. Consequently, all contours are stable in the sense of Definition 6 on P. 573 in [16] and by induction all contours are stable in the sense of Definition 2 on P. 564 from [16], for all GSs \( \varphi \). Thus, the truncated free energy coincides with the complete free energy (as no contour is actually truncated), and all GSs are stable in the sense of Definition 5; see P. 564 in [16]. (Another term used in this situation is dominant GSs; see [1].)

Similarly, for part (c) of the Theorem 2, after excluding the highly unlikely winding contours, the statistics of random configurations \( \phi^{(1)}_{T_k} \in A_{\text{per},k} \) is described through the ensemble of external contours. As in the proof of Theorem 1, the combined shift/reflection-symmetry implies that the limit GM \( \lim_{k \to \infty} \mu_{\text{per},k} \) exists and is represented as the sum

\[
\frac{1}{D^2} \sum_{\varphi \in \mathcal{D}} \mu_\varphi. \quad \square
\]
5 The limit Gibbs measures: a generic hard-core diameter

For a generic $D$ (Case 3) there are at least three sublattices among GSs. It is convenient to pair the R-symmetric sublattices and group their shift-generated GSs into classes. Geometrically, different classes of GSs are different because the sublattices from different classes are only mapped into each other by a rotation which is not mapping underlying lattice $\mathbb{L}$ into itself. Consequently, there is no simple way to establish equality between partition functions with GS boundary conditions $\varphi$ and $\varphi'$ originating from two different classes, as there is no obvious map between sets $\mathcal{A}_{V,\varphi}$ and $\mathcal{A}_{V,\varphi'}$ defining the partition functions $Z(V, u)$ in (2.5). Moreover, due to the difference in their geometry (caused by arithmetic properties of $D^2$), some classes of dense-packing configurations do not survive in the thermodynamic limit (i.e., do not generate pure phases). To be more specific, the general results in [16] ensure only the following theorem.

**Theorem 3.** Suppose that the prime decomposition of $D^2$ contains at least two primes of the form $6k + 1$ (Case 3). There exists $u_0 = u_0(D) \in (1, \infty)$ such that for $u > u_0(D)$ the subsequent assertions (a-c) hold true.

- (a) For at least one class of GSs (containing either $D^2$ or $2D^2$ elements) each of the GSs $\varphi$ in the class generates a GM $\mu_{\varphi}$ which is a pure phase. The measures $\mu_{\varphi}$ are pair-wise distinct. However, if $S_{u}\varphi = \varphi'$ then $S_{u}\mu_{\varphi} = \mu_{\varphi'}$ and if $R\varphi = \varphi'$ then $R\mu_{\varphi} = \mu_{\varphi'}$. In particular, each measure $\mu_{\varphi}$ is $D$-periodic relative to lattice shifts.

- (b) Any shift-periodic GM $\mu \in \mathcal{G}$ is a convex combination of measures $\mu_{\varphi}$ constructed in (a).

- (c) The limit GM $\mu_{\text{per}}$ exists for $V_k = T_k$ as in (4.1). Furthermore, $\mu_{\text{per}}$ is a uniform mixture of the measures $\mu_{\varphi}$ constructed in (a). Consequently, measure $\mu_{\text{per}}$ is shift-invariant and $R$-invariant.

**Proof of Theorem 3.** The proof is simply a repetition of arguments already presented in the proof of Theorems 1 and 2.

In Theorems 4 – 6 we present simple examples where not all classes of the GSs generate the corresponding pure phases, i.e., some of the GSs are not stable in the sense of Definition 5 on page 564 in [16]. As in the preceding theorems, we consider the setting where fugacity $u$ is large enough; to avoid repetitions, this fact is not stressed explicitly. We start with the value $D = 7$.

**Theorem 4.** For $D = 7$ there are 147 GSs divided in two classes: one inclined and one non-inclined. The non-inclined class consists of 49 GSs, and these are the only GSs that generate pure phases.

Our next result is that for the value of $D = 13$ the selection of the stable ground states is inverted.
Theorem 5. For $D = 13$ there are 507 GSs divided in two classes: one inclined and one non-inclined. The inclined class consists of 338 GSs, and these are the only GSs that generate pure phases.

Finally, we discuss the case of $D = \sqrt{147}$. This case is interesting since we have two inclined classes of GSs. One is generated by the $R$-symmetric $D$-sublattice comprising the site $(7, 7)$ and the other by two $D$-sublattices comprising the sites $(11, 2)$ and $(2, 11)$ respectively. (We bear in mind that both sublattices pass through the origin.) We refer to these classes as the $(7, 7)$-one (diagonal) and the $(11, 2)$-one.

Theorem 6. For $D = \sqrt{147}$ there are 441 GSs divided in two inclined classes, $(7, 7)$ and $(11, 2)$. The diadonal $(7, 7)$-class consists of 147 GSs, and these are the only GSs generating pure phases.

Proof of Theorem 4. For $D = 7$, the two classes (inclined and non-inclined) are determined by the sublattices containing sites $(3, 5)$ or $(5, 3)$ for the inclined class and site $(0, 7)$ for the non-inclined class. This implies the count of the corresponding GSs. Our proof follows closely the approach from [16]: we define an appropriate family of small contours (see Definition 1 on page 566 in [16]) and then compare the free energies of the corresponding truncated models to see that only non-inclined GSs are stable.

The 'smallest' contour in a GS is generated by the removal of a single particle. The statistical weight of such a contour is $u^{-1}$. The density of such contours is the same in each of the GSs $\varphi \in D$. Similarly, the removal of two particles at distance $D$ from each other generates a contour with statistical weights $u^{-2}$. Again the density of such contours is the same in every GS $\varphi \in D$.

The next type of a small contour is generated when three particles at the vertices of the equilateral $D$-triangle are removed and one particle is inserted at a site inside the $D$-triangle such that it repels only the 3 vertices of this triangle (it is at the distance $\geq D$ from any other sublattice site). Again such a contour has statistical weight $u^{-2}$. Geometrically, the inserted position belongs to a closed concave circular triangle inside a $D$-triangle. (Cf. the light-gray areas in Figure 5.) In all above examples we can speak of a single-particle 'defect'. In examples below we will deal with pair (or double) defects.

The argument so far holds for any attainable $D$. For $D = 7$, any inclined triangle has 12 sites in the inscribed closed concave circular triangle (see Figure 5). E.g. the light-gray triangle with vertices

$$(0, 0), (5, 3), \text{ and } (-3, 8)$$

covers the following lattice sites

$$(-1, 5), \quad (-1, 6), \quad (0, 2), \quad (0, 3), \quad (0, 4), \quad (0, 5),$$
$$\quad (1, 2), \quad (1, 3), \quad (1, 4), \quad (2, 3), \quad (2, 4), \quad (3, 3).$$

A particle inserted into any of these sites generates a contour of statistical weight $u^{-2}$. (Three of these sites: $(0, 2), (1, 2)$ and $(0, 3)$ are shown in Figure 5 with black balls placed at them, for the purpose explained below.)
Figure 5: The structure of pair defects for the distinguishing contour used in the proof of Theorem 4, with $D = 7$. The contour exhibits 4 removed and 2 added particles. Both inclined (left) and non-inclined (right) ground states are shown: the white balls indicate the removed ground-state particles (repelled sites) at the vertices of a $D$-parallelogram. The origin $0 = (0,0)$ coincides with the site of the bottom left white ball. The light-gray areas (closed circular triangles) cover the inserted lattice sites repelling 3 white balls but not repelling any other ground-state particles. The inserted sites falling in dark gray areas (open bi-convex lenses) repel 4 white balls but no other ground-state particles. Any two black balls connected by a straight line can be selected as particles forming a pair defect: there are 6 pairs for the inclined ground state (left) and 7 for the non-inclined one (right). In particular, a larger black ball can be paired with every ball from the opposite stack.
Similarly, for \( D = 7 \) any non-inclined closed concave circular \( D \)-triangle also covers 12 sites. E.g. a closed concave circular triangle with vertices 

\[(0, 0), (7, 0), \text{ and } (0, 7),\]

includes the following sites:

\[(1, 1), (1, 2), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (3, 2), (4, 1), (4, 2), (5, 1).\]

These are exactly the lattice sites falling in the left light-gray closed circular triangle from the right picture in Figure 5. A particle inserted into any of these sites generates a contour of statistical weight \( u^{-2} \). (Again, three of these sites: \( (1, 1), (1, 2) \) and \( (2, 1) \) are shown, marked with black balls; see below.) Thus, both inclined and non-inclined GSs have the same density of small contours of this type and therefore are still indistinguishable.

The small contour which makes the difference is constructed when four particles at the vertices of a \( D \)-parallelogram are removed and two particles inside the parallelogram are added, maintaining admissibility. The statistical weight of this contour also equals \( u^{-2} \). For any inclined parallelogram, e.g. with vertices 

\[(0, 0), (5, 3), (-3, 8), \text{ and } (2, 11),\]

there are 6 pairs of sites to place the particles:

\[[(0, 2), (1, 9)], [(0, 2), (2, 8)], [(0, 2), (2, 9)], [(0, 3), (2, 9)], [(1, 2), (1, 9)], [(1, 2), (2, 9)].\]

See pairs of black balls connected by straight lines in the left picture in Figure 5.

In contrast, for any non-inclined parallelogram, e.g. with vertices 

\[(0, 0), (7, 0), (0, 7), \text{ and } (7, 7),\]

there are 7 pairs of sites to place the particles:

\[[(1, 1), (5, 6)], [(1, 1), (6, 5)], [(1, 1), (6, 6)], [(1, 2), (6, 5)], [(1, 2), (6, 6)], [(2, 1), (5, 6)], [(2, 1), (6, 6)].\]

see pairs of black balls connected by straight lines in the right picture in Figure 5. We conclude that the non-inclined GSs win in the number (and density) of contours of weight \( u^{-2} \).

Any other contour in the truncated model has statistical weight at most \( u^{-3} \). This seemingly obvious statement actually requires a certain effort to verify. We present it as Lemma A.1 in Appendix I. With Lemma A.1 at hand one can use in a standard way the polymer expansions for the free energies of the truncated contour models (e.g. see Sections 1.7, 2.1 in [10] or Section 3.a in [13]) and conclude that the free energy for the non-inclined class is smaller than that for the inclined class by at least \( cu^{-2} \), where \( c > 0 \) is an absolute constant. Therefore, only the non-inclined class contains stable GSs. ■

**Proof of Theorem 5.** The argument in the proof of Theorem 5 is similar to that of Theorem 4, except for the specific enumeration of small contours. The first difference
Figure 6: The structure of pair defects for a distinguishing contour of weight $u^{-2}$ used in the proof of Theorem 5, with $D = 13$. The contour exhibits 4 removed and 2 added particles, affecting the vertices of a $D$-parallelogram. Both inclined (left) and non-inclined (right) ground states are shown: white circles indicate the removed ground-state particles. The origin $0 = (0,0)$ is at the position of the bottom left white ball. The light-gray areas (closed circular triangles) cover the lattice sites which repel 3 out of 4 white balls but do not repel any other ground-state particles. The gray area (an open bi-convex lense) includes the sites that repel all 4 white balls but no other ground-state particles. Each black circle from one triangular stack, together with at least one black circle from the opposite stack, can be selected to form a pair defect. There are 113 pairs that can be formed for the inclined $D$-sublattice and 78 pairs for the non-inclined one (out of 625 and 484 nominal pairs, respectively). Again, a larger black ball can be paired with any ball from the opposite stack.
with Theorem 4 is in the types of contours having the statistical weight $u^{-2}$. As before, we have pair defects where two inserted particles remove four vertices of a $D$-parallelogram: see Figure 6. In addition, we also can place three particles in a trapezoid that is the union of a $D$-parallelogram and a $D$-triangle and four particles in a $2D$-triangle: see Figure 7. In all 3 categories the inclined sublattice dominates distinctively with 113 vs 78 pairs in a $D$-parallelogram, 61 vs 20 triples in a trapezoid and 39 vs 3 quadruples in a $2D$-triangle. The enumeration of cases above can be performed manually but we also present as Listing L.1 in Section 7 below the Java routine which automates this task. The verification that no other contours with statistical weight smaller than $u^{-3}$ exist for the non-inclined sublattice is presented as Lemma A.2 in the Appendix I; it requires more massive enumeration and therefore relies on computer-assisted argument. The rest of the proof can be borrowed from Theorem 4 verbatim. 

**Proof of Theorem 6.** As above, the argument repeats that for Theorems 4, 5, and is based on an exact count of types of small contours of weight $u^{-2}$. There are 3 categories of distinguishing contours: all give a heavy preference to the diagonal $(7,7)$ class. First, there are contours with two-particle defects where two added particles remove four vertices of a $D$-parallelogram: here the $(7,7)$ sublattice admits 86 pairs against 51 in the $(11,2)$-sublattice. Cf. Figure 8. Second, there are contours with three-particle defects: here three added particles remove five sites from a trapezoid. Again, the $(7,7)$ sublattice admits 39 such triples while the $(11,2)$ sublattice admits only 1. Finally, there are admissible quadruples of added particles: they remove six sites from a $2D$-triangle. The number of such quadruples in the $(7,7)$ sublattice is 16 whereas the $(11,2)$ sublattice does not admit them at all. See Figure 9. The proof is completed by using Lemma A.3
Figure 8: The structure of pair defects for a distinguishing contour of weight $u^{-2}$ used in the proof of Theorem 6, with $D = \sqrt{147}$. The contour exhibits 4 removed and 2 added particles, affecting the vertices of inclined $D$-parallelogram. Both $(11, 2)$ (left) and $(7, 7)$ (right) ground states are shown: white circles indicate the removed ground-state particles. The origin $0 = (0, 0)$ is at the position of the bottom left white ball. The light-gray areas (closed circular triangles) cover the lattice sites which repel 3 out of 4 white balls but do not repel any other ground-state particles. The gray area (an open bi-convex lens) includes the sites that repel all 4 white balls but no other ground-state particles. Each black circle from one triangular stack, together with at least one black circle from the opposite stack, can be selected to form a pair defect. There are 51 pairs that can be formed for the $(11, 2)$ $D$-sublattice and 86 pairs for the diagonal $(7, 7)$ one (out of 144 and 441 nominal pairs, respectively). As before, a larger black ball can be paired with any ball from the opposite stack.
Figure 9: Examples of double, triple and quadruple defects of statistical weight $u^{-2}$ for $D = \sqrt{147}$. As before, pairs of inserted sites (black balls) remove four repelled sites (the vertices of a $D$-parallelogram). Triples of inserted sites remove five repelled sites (white balls forming a trapezoid). A quadruple of inserted sites removes six repelled sites from the diagonal $(7,7)$ ground state (white balls forming a $2D$-triangle); for the $(11,2)$ such a quadruple is impossible

guaranteeing that the above types are the only ones with weight $u^{-2}$.

Going back to general $D$, it is possible to verify that the two-particle small contours of the type we used in Theorems 4, 5 and 6 are the smallest possible contours which can distinguish classes of GSs. The only other candidate is the triangular contour defined in the proof of Theorem 4 (the contour with 3 removed particles and 1 added particle). The amount of lattice sites to place a particle inside such a triangle (see gray triangles in Figures 5 and 6) is actually the same for all GSs generated by given $D$. For the case $D = 7, 13, \sqrt{147}$ this was verified by direct enumeration in the proofs of Theorems 4, 5 and 6.

For an arbitrary attainable $D$ this amount is exactly one half of the count of lattice sites inside the centered at the origin hexagon with the side length $D$ minus the amount of lattice sites inside the centered at the origin disc of radius $D$. The above difference is calculated under the following conditions. The disk is open (in $\mathbb{R}^2$). The hexagon is half-open (also in $\mathbb{R}^2$), i.e., 3 hexagon vertices and 3 open (in $\mathbb{R}$) sides of the hexagon boundary are included while the other 3 sides and 3 vertices are not. Additionally, the vertices of the hexagon are placed at the sites of the sublattice that generates given class of GSs. It is evident that under the conditions above both quantities participating in the difference are independent on a choice of the sublattice.

We do not know if the contours with the statistical weight $u^{-2}$ are enough to always determine stable phases and if the stable class is always unique. The numeric calculation for all $D^2 \leq 100000$ shows that there is always only one sublattice which dominates in the amount of parallelogram contours. Presumably this is the sublattice for which the corresponding closed concave circular $D$-triangle contains a site at the shortest distance from the triangle vertex with respect to other sublattices.
6 Appendix 1. A brief review of Eisenstein primes

Let \( \omega \) stand for a complex number \( \omega = -\frac{1}{2} + \frac{\sqrt{3}}{2} i \). Then \( \overline{\omega} = \omega^2 \) and \( 1 + \omega + \omega^2 = 0 \). An Eisenstein integer (EI) is a complex number \( m + n \omega \) where \( m, n \in \mathbb{Z} \) are integers. The set of EIs is a ring (in fact, an entire domain) \( \mathbb{E} \) with 6 units: \( \pm 1, \pm \omega, \pm \omega^2 \) (cf. [10] [11]).

Define a set of canonical EIs as a subset of EIs which have the form \( m - n \omega \) or \( m - n \omega^2 = -\omega^2(n - m \omega) \) with \( m \geq n \geq 0 \). Each canonical EI multiplied by a unit EI different from 1 yields a different EI and the entire domain \( \mathbb{E} \) is generated in this way from the set of canonical EIs. Enumerating all 6 products of a given EI and a unit EI it is not hard to see that a canonical form of given EI is defined uniquely. Canonical EIs of the form \( m - n \omega^2 \) are complex conjugates for the canonical EIs of the form \( m - n \omega \).

An EI is called Eisenstein prime (EP) if it cannot be decomposed into a product of two not unit EIs. If a prime \( p \) is not an EP then it can be decomposed into a product of two non-unit factors. These two factors are, of course, complex conjugates of each other with non-0 real and imaginary parts, i.e. \( p = (a - b \omega)(a - b \omega^2) = a^2 + ab + b^2 \), \( a \neq 0 \), \( b \neq 0 \). It is not hard to see that one can always select two complex conjugate factors to be a canonical EIs.

Theorem 5 in [10] implies that the above decomposition of \( p \) is possible only for \( p = 3 = (1 - \omega)(1 - \omega^2) = -\omega^2(1 - \omega)^2 \) and \( p \) of the form \( p = 6k + 1 \). Moreover, for \( p = 6k + 1 \), or equivalently \( p = 3k + 1 \) (see [M]), the integers \( a > b \) are defined uniquely and are mutually prime. The remaining primes \( p \) are either \( p = 2 \) or have the form \( p = 6k - 1 \), or equivalently they are of the form \( p = 3k - 1 \). These primes are also EPs (see [10]).

According to Theorem 8 in [10], each EI has a unique prime decomposition

\[
m + n \omega = \varepsilon(1 - \omega)^\alpha \prod_j p_j^{\beta_j} \prod_k (a_k - b_k \omega)^{\gamma_k} \prod_l (a_l - b_l \omega^2)^{\delta_l}. \tag{A1.1}
\]

Here \( \varepsilon \) is a unit, each index \( j, k \) or \( l \) runs over a finite (possibly empty) set, power \( \alpha \) is non-negative, powers \( \beta_j, \gamma_k \) and \( \delta_l \) are positive integers, all \( p_j \) are different primes of the form \( 3k - 1 \), and all \( (a_k - b_k \omega), (a_l - b_l \omega^2) \) are different canonical EPs.

Consider an integer \( D^2 \) of the form

\[
D^2 = 3^\alpha \prod_j p_j^{2\beta_j}, \tag{A1.2}
\]

where \( \alpha \in \mathbb{Z}_+ \) is a non-negative integer, \( j \) takes finite number of values, \( \beta_j \in \mathbb{N} \) is a positive integer and \( p_j \) are distinct primes of the form \( 3k - 1 \). Suppose that there exists a representation

\[
D^2 = c^2 + cd + d^2 = (c - d \omega)(c - d \omega^2),
\]

where \( c \) and \( d \) are integers.

Then the unique prime decomposition for \( (c - d \omega) \) and \( (c - d \omega^2) \) must be

\[
(c - d \omega) = \varepsilon_1(1 - \omega)^\alpha \prod_j p_j^{\beta_j} \quad \text{and} \quad (c - d \omega^2) = \varepsilon_2(1 - \omega)^\alpha \prod_j p_j^{\beta_j}
\]

20
as both \((c - d\omega)\) and \((c - d\omega^2)\) have the same absolute value (as complex numbers).

If \(\alpha\) is even then \((1 - \omega)^\alpha = (-3\omega)^{\alpha/2}\) and we have

\[
(c - d\omega) = \varepsilon 3^{\alpha/2} \prod_j p_j^{\beta_j} \quad \text{and} \quad (c - d\omega^2) = \varepsilon^{-1} 3^{\alpha/2} \prod_j p_j^{\beta_j}.
\]

Inspecting all possible values of \(\varepsilon\) we obtain the following 6 solutions:

\[
\begin{align*}
\varepsilon &= 1, \; c = D, \; d = 0 & \varepsilon &= -1, \; c = -D, \; d = 0 & \varepsilon &= \omega, \; c = 0, \; d = D \\
\varepsilon &= -\omega, \; c = 0, \; d = -D & \varepsilon &= \omega^2, \; c = -D, \; d = D & \varepsilon &= -\omega^2, \; c = D, \; d = -D,
\end{align*}
\]

none of which satisfies the condition \(c > d > 0\).

If \(\alpha\) is odd we have

\[
(c - d\omega) = \varepsilon(1 - \omega)3^{(\alpha-1)/2} \prod_j p_j^{\beta_j} \quad \text{and} \quad (c - d\omega^2) = \varepsilon^{-1}(1 - \omega^2)3^{\alpha/2} \prod_j p_j^{\beta_j},
\]

Listing all possible values of \(\varepsilon\) we obtain the following 6 solutions:

\[
\begin{align*}
\varepsilon &= 1, \; c = D/\sqrt{3}, \; d = D/\sqrt{3} & \varepsilon &= -1, \; c = -D/\sqrt{3}, \; d = -D/\sqrt{3} \\
\varepsilon &= \omega, \; c = D/\sqrt{3}, \; d = -2D/\sqrt{3} & \varepsilon &= -\omega, \; c = -D/\sqrt{3}, \; d = 2D/\sqrt{3} \\
\varepsilon &= \omega^2, \; c = -2D/\sqrt{3}, \; d = D/\sqrt{3} & \varepsilon &= -\omega^2, \; c = 2D/\sqrt{3}, \; d = -D/\sqrt{3},
\end{align*}
\]

none of which satisfies the condition \(c > d > 0\).

Now consider an integer \(D^2\) of the form

\[
D^2 = p3^\alpha \prod_j p_j^{2\beta_j}, \quad (A1.3)
\]

where \(\alpha\) is non-negative integer, \(j\) takes finite number of values, \(\beta_j\) is positive integer, \(p_j\) are distinct primes of the form \(3k - 1\), and \(p\) is a prime of the form \(6k + 1\).

Let \(p = (a - b\omega)(a - b\omega^2)\) be a decomposition of \(p\) into canonical EPs and suppose again that there exists a representation

\[
D^2 = c^2 + cd + d^2 = (c - d\omega)(c - d\omega^2),
\]

where \(c\) and \(d\) are integers. Then the unique prime decomposition for \((c - d\omega)\) and \((c - d\omega^2)\) must be

\[
(c - d\omega) = \varepsilon_1(1 - \omega)^\alpha (a - b\omega) \prod_j p_j^{\beta_j} \quad \text{and} \quad (c - d\omega^2) = \varepsilon_2(1 - \omega)^\alpha (a - b\omega^2) \prod_j p_j^{\beta_j}
\]

as both \((c - d\omega)\) and \((c - d\omega^2)\) have the same modulus.

As before if \(\alpha\) is even then we write

\[
(c - d\omega) = \varepsilon(a - b\omega)3^{\alpha/2} \prod_j p_j^{\beta_j} \quad \text{and} \quad (c - d\omega^2) = \varepsilon^{-1}(a - b\omega^2)3^{\alpha/2} \prod_j p_j^{\beta_j}
\]

or

\[
(c - d\omega) = \varepsilon(a - b\omega^2)3^{\alpha/2} \prod_j p_j^{\beta_j} \quad \text{and} \quad (c - d\omega^2) = \varepsilon^{-1}(a - b\omega)3^{\alpha/2} \prod_j p_j^{\beta_j}.
\]
Inspecting all possible values of $\varepsilon$ we obtain the following 12 solutions:

$$\varepsilon = 1, \ c = aD/\sqrt{p}, \ d = bD/\sqrt{p}, \ \ldots$$

among which there is a unique solution $\varepsilon = 1, \ c = aD/\sqrt{p}, \ d = bD/\sqrt{p}$ with $c > d > 0$.

If $\alpha$ is odd then we have

$$(c-d\omega) = \varepsilon(1-\omega)(a-b\omega)3^{(\alpha-1)/2} \prod_j p_j^{\beta_j}, \ (c-d\omega^2) = \varepsilon^{-1}(1-\omega^2)(a-b\omega^2)3^{(\alpha-1)/2} \prod_j p_j^{\beta_j}$$
or

$$(c-d\omega) = \varepsilon(1-\omega)(a-b\omega^2)3^{(\alpha-1)/2} \prod_j p_j^{\beta_j}, \ (c-d\omega^2) = \varepsilon^{-1}(1-\omega^2)(a-b\omega)3^{(\alpha-1)/2} \prod_j p_j^{\beta_j}.$$ 

Again listing all possible values of $\varepsilon$ we obtain the following 12 solutions:

$$\ldots, \ \varepsilon = -\omega^2, \ c = (a+2b)D/\sqrt{3p}, \ d = (a-b)D/\sqrt{3p}, \ \ldots$$

among which there is a unique solution $\varepsilon = -\omega^2, \ c = (a+2b)D/\sqrt{3p}, \ d = (a-b)D/\sqrt{3p}$ with $c > d > 0$.

7 Appendix 2. Technical lemmas

In this section we verify that the the small contours with the statistical weight larger than or equal to $u^{-2}$ which were defined in the previous section are the only ones with this statistical weight, and any other contour has the statistical weight $\leq u^{-3}$.

**Lemma A2.1.** For $D=7$ any contour other than the contours described in the proof of Theorem 4 has statistical weight at most $u^{-3}$.

**Proof of Lemma A2.1.** Without loss of generality we can assume that the underlying GS $\varphi$ has $\varphi(0) = 1$ (i.e., $\mathbb{E}(\varphi)$ is a $D$-sublattice). Any contour can be obtained by adding finitely many particles (at inserted sites), and then removing the particles from $\mathbb{E}(\varphi)$ which are repelled by the inserted particles (removed sites). The resulting admissible configuration is denoted by $\phi$. One can also remove from $\mathbb{E}(\phi)$ any additional particles which are not forced to be repelled by inserted sites. However, an unforced removal of particles can only decrease the statistical weight and therefore can be disregarded.

If $\sharp = \sharp(\phi)$ is the difference between the numbers of removed and inserted sites then the statistical weight of the contour is $u^{-\sharp}$. As we saw earlier, every inserted site repels from $\mathbb{E}(\varphi)$ either 3 or 4 removed sites. The inserted sites which repel 3 removed sites are located inside closed concave circular triangles identified in the proof of Theorem 4 (the light-gray areas in Figure 5). The complement (in $\mathbb{R}^2$) to these triangles consists of mutually disjoint open circular bi-convex lenses (see the gray areas in Figure 5). A particle inserted in a lattice site belonging to a lens repels 4 removed sites.

Consider a $D$-connected component $\Delta$ of the set of removed sites (together with the corresponding inserted sites). As above, let $\sharp(\Delta)$ denote the difference between the
numbers of removed and inserted sites in \( \Delta \). To prove the lemma, it suffices to verify that the statistical weight \( u^{-\sharp(\Delta)} \) of any such component is at most \( u^{-3} \), i.e., \( \sharp(\Delta) \geq 3 \). Note that a contour support has been defined in Section 2 by using a different notion of connectedness; hence it can include more than one \( \Delta \). In that case the statistical weight of the contour is the product of the statistical weights of constituting \( D \)-connected components, and for our purposes it is enough to estimate the statistical weight of a single component \( \Delta \).

It is convenient to introduce a total repelling force \( F(x) = F(x, \phi) \) acting upon a removed site \( x \) (in the resulting configuration \( \phi \in \mathcal{A} \)). This force is accumulated from all inserted sites \( y_i \) that repel site \( x \) in \( \phi \): \( F(x) = \sum_i F(x, y_i, \phi) \). We require that every summand \( F(x, y_i, \phi) \) is non-negative and depends only on the Euclidean distance \( \rho(x, y_i) \) between \( x \) and \( y_i \). The square of this distance \( r = \rho(x, y_i)^2 \) is always a positive integer, and we use a shorthand notation \( f_r \) for \( F(x, y_i, \phi) \), with \( r = \rho(x, y_i)^2 \). With this notation,

\[
F(x) = \sum_{r \in \mathbb{N}} f_r \sum_{y \in \mathcal{L}} 1 \left( y \text{ repels } x \text{ in } \phi, \text{ and } r = \rho(y, x)^2 \right). \tag{A2.2}
\]

The term \( f_r \) are referred to as the local repelling force at distance \( \sqrt{r} \).

An additional restriction on \( f_r \) is that if a given inserted site \( y \) repels 3 or 4 removed sites \( x_j \) then

\[
\sum_j f_{\rho(y, x_j)^2} = 1, \tag{A2.3}
\]

i.e., the total repelling force 1 generated by the inserted site \( y \) is split between removed sites \( x_j \) it repels. Under this approach, if the deficit \( \delta(x) = \delta(x, \phi) \) of the removed site \( x \) is calculated as \( 1 - F(x) \) then \( \sharp(\phi) \) is the sum \( \sum \delta(x, \phi) \) over all sites \( x \in \mathcal{E}(\varphi) \) removed in the course of passage from \( \varphi \) to \( \phi \). From now on we assume that the configuration \( \phi \) has a single component \( \Delta \), and the rest of the proof deals with this \( \Delta \). Accordingly, we write that \( \sharp(\phi) = \sharp(\Delta) \).

The family \( \{f_r\} \) of local repelling forces is called proper if for any \( \Delta \) and each removed site \( x \in \Delta \) the deficit \( \delta(x) \geq 0 \). A proper family of repelling forces is a convenient tool for estimating the value of \( \sharp(\Delta) \) as the sum of deficits \( \delta(x) \) collected over some removed sites in \( \Delta \) can only increase when the remaining removed sites in \( \Delta \) are also taken into account.

Each connected component \( \Delta \) consists of internal sites for which all 6 sublattice neighbors also belong to \( \Delta \) and boundary sites which have at least one occupied \( D \)-sublattice neighbor (obviously, not belonging to \( \Delta \)). Each \( D \)-connected component of the boundary sites in \( \Delta \) defines a closed broken line in \( \mathbb{R}^2 \), and the set \( \Delta \) can be understood as \( \mathbb{R}^2 \)-polygon with the boundary \( \partial \Delta \) formed by these broken lines. In general, the boundary \( \partial \Delta \) of \( \Delta \) can have several connected components: one external component and zero or more internal components. An ambiguous situation arises when 4 \( D \)-segments from \( \partial \Delta \) meet at the same boundary site (i.e., this site has 2 opposite \( D \)-neighbors that are occupied). In that case we fictitiously cut this site along the short line segment (length less than 1) which passes through this site and has both ends inside \( \Delta \) (viewed as an open polygon in \( \mathbb{R}^2 \)). This removes the ambiguity, and the exterior and the interior of \( \Delta \) become defined uniquely.
It is clear that, as an $\mathbb{R}^2$-polygon, $\Delta$ can only have vertices with angles $\pi/3$, $2\pi/3$ and $4\pi/3$. We say that the corresponding removed sites from $\partial \Delta$ are of the type $\pi/3$, $2\pi/3$ and $4\pi/3$ respectively. The remaining sublattice sites from $\partial \Delta$ correspond to the angle $\pi$; we say that such a site has type $\pi$.

If a vertex $x \in \partial \Delta$ is repelled only by a single inserted site $y$ then imagine this particle being deleted. Then vertex $x$ also disappears from $\Delta$ (as nothing repels it anymore) and the value $\sharp(\Delta)$ does not increase. (Actually, $\sharp(\Delta)$ remains intact if $y$ repels a single vertex $x$ in $\Delta$.) Therefore, every polygon $\Delta$ which can be reduces to a single $D$-parallelogram by the above deletion process has $\sharp(\Delta) \geq 3$. Indeed, for $D = 7$ it is not hard to verify (see Figure 5) that adding a repelling site to a $D$-parallelogram always produces a polygon with $\sharp \geq 3$. Reversing the deletion process we obtain a sequence of increasing polygons with a non-decreasing $\sharp$, where the starting polygon in the sequence has $\sharp \geq 3$ and the final polygon is the original $\Delta$.

The above deletion trick does not work in the case where we have 2 inserted sites inside each $D$-parallelogram supporting a type $2\pi/3$ vertex in $\partial \Delta$. In that case there are at least 6 such $D$-parallelograms in $\Delta$. Our next step is to find a collection of proper repelling forces $\{f_r\}$ such that the deficit $\delta(x) > 1/3$ for any $x \in \partial \Delta$ supported by a non-deletable $D$-parallelogram. This would imply the lemma as $6\delta(x) > 2$.

To specify appropriate values $f_r$, consider an open $\mathbb{R}^2$-disk of radius 7 centered at the origin $0$. Inside this disk one can only have lattice sites at the following squared Euclidean distances from $0$:

$$1, 3, 4, 7, 9, 12, 13, 16, 19, 21, 25, 27, 28, 31, 36, 37, 39, 43, 48.$$  \hfill (A2.4)

These are all possible values of $r$ to consider.

Define the repelling type of an inserted site $y$ as an ordered triple $(r_1, r_2, r_3)$, with $r_1 \leq r_2 \leq r_3$, or an ordered quadruple $(r_1, r_2, r_3, r_4)$, with $r_1 \leq r_2 \leq r_3 \leq r_4$, of squared distances $r_i$ from $y$ to each of the removed sites $x_i$ it repels. For the non-inclined GS $\varphi$ there are only 3 distinct repelling types $(3, 31, 31)$, $(7, 21, 28)$ and $(12, 19, 19)$ for the inserted site located inside a closed concave circular triangle. The corresponding values $f_r$ must obey the following equalities:

$$f_3 + f_{31} + f_{31} = 1, \quad f_7 + f_{21} + f_{28} = 1, \quad f_{12} + f_{19} + f_{19} = 1.$$  \hfill (A2.5)

For the inclined GS $\varphi$ there are only 4 such types which imply the following equalities:

$$f_4 + f_{27} + f_{31} = 1, f_7 + f_{21} + f_{28} = 1,$$
$$f_9 + f_{19} + f_{25} = 1, f_{12} + f_{13} + f_{16} + f_{21} = 1.$$  \hfill (A2.6)

Similarly, for the non-inclined GS there are only 4 distinct repelling types for the inserted site inside an open bi-concave circular lense, which imply the following equalities:

$$f_1 + f_{36} + f_{36} + f_{36} = 1, f_4 + f_{25} + f_{39} + f_{39} = 1,$$
$$f_9 + f_{16} + f_{37} + f_{37} = 1, f_{13} + f_{13} + f_{13} + f_{13} = 1.$$  \hfill (A2.7)

For the inclined GS there are only 4 such types which imply the following equalities:

$$f_1 + f_{37} + f_{39} + f_{48} = 1, f_3 + f_{28} + f_{37} + f_{43} = 1,$$
$$f_7 + f_{19} + f_{36} + f_{39} = 1, f_{12} + f_{13} + f_{31} + f_{43} = 1.$$  \hfill (A2.8)
For our purposes we select the following collection \( \{ f_r \} \) solving the system of linear equations (A2.5)–(A2.8).

\[
\begin{align*}
    f_1 &= 44/56, & f_3 &= 40/56, & f_4 &= 40/56, & f_7 &= 31/56, & f_9 &= 31/56, \\
    f_{12} &= 22/56, & f_{13} &= 22/56, & f_{16} &= 17/56, & f_{19} &= 17/56, & f_{21} &= 17/56, \\
    f_{25} &= 8/56, & f_{27} &= 8/56, & f_{28} &= 8/56, & f_{31} &= 8/56, & f_{36} &= 4/56, \\
    f_{37} &= 4/56, & f_{39} &= 4/56, & f_{43} &= 4/56, & f_{48} &= 4/56,
\end{align*}
\]

\( \text{(A2.9)} \)

This choice of \( f_r \) has an advantage of being monotonically non-increasing with \( r \); see below.

Similarly to the repelling type of an inserted site, we can define a dual notion of the repelling type of a removed site. Let \( x \) be a removed site and suppose it is repelled by \( m \) inserted sites \( y_1, \ldots, y_m \). The repelling type of \( x \) is an ordered \( m \)-tuple \((r_1, \ldots, r_m)\) of squared distances from \( x \) to \( y_1, \ldots, y_m \). It is possible to verify that for \( D = 7 \) there are in total 430 different repelling types, which correspond to \( m = 1, 2, 3, 4, 5 \). However, since the values \( f_r \) in (A2.9) are monotone, the number of cases to analyse is greatly reduced.

To this end, we define a partial order on \( m \)-tuples \( \mathbf{n} \in \mathbb{N}^m \) of positive integers for which \( \mathbf{n}' \succ \mathbf{n} \) iff the entry \( n'_i \geq n_i \), \( 1 \leq i \leq m \), and \( \mathbf{n}' \neq \mathbf{n} \). The monotonicity of \( f_r \) implies that if \((r'_1, \ldots, r'_m) \succ (r_1, \ldots, r_m)\) then \( \sum_{j=1}^m f_{r'_j} \leq \sum_{j=1}^m f_{r_j} \). As a result, we need to focus only on minimal (in the sense of \( \succ \)) types of removed sites.

A direct inspection of all possible minimal repelling types re-produces 14 relations (A2.5)–(A2.8) as well as 22 additional equalities:

\[
\begin{align*}
    f_4 + f_{28} + f_{36} + f_{48} &= 1, & f_7 + f_{25} + f_{27} &= 47/56, \\
    f_7 + f_{28} + f_{36} + f_{37} &= 47/56, & f_9 + f_{28} + f_{31} + f_{48} &= 51/56, \\
    f_9 + f_{31} + f_{33} + f_{37} &= 51/56, & f_{12} + f_{16} + f_{28} &= 47/56, \\
    f_{12} + f_{21} + f_{28} + f_{43} &= 51/56, & f_{12} + f_{27} + f_{31} + f_{33} &= 23/28, \\
    f_{13} + f_{19} + f_{39} + f_{39} &= 13/14, & f_{13} + f_{19} + f_{31} + f_{39} &= 51/56, \\
    f_{13} + f_{25} + f_{27} + f_{39} &= 3/4, & f_{13} + f_{37} + f_{39} + f_{43} + f_{48} &= 19/28, \\
    f_{16} + f_{21} + f_{28} + f_{37} &= 23/28, & f_{16} + f_{28} + f_{28} + f_{36} &= 37/56, \\
    f_{19} + f_{19} + f_{25} + f_{36} &= 23/28, & f_{19} + f_{25} + f_{27} + f_{31} &= 41/56, \\
    f_{21} + f_{21} + f_{28} + f_{28} &= 25/28, & f_{25} + f_{25} + f_{27} + f_{27} &= 4/7, \\
    f_{25} + f_{37} + f_{39} + f_{48} &= 3/7, & f_{28} + f_{28} + f_{36} + f_{43} + f_{48} &= 1/2, \\
    f_{28} + f_{36} + f_{37} + f_{39} + f_{43} &= 3/7, & f_{31} + f_{31} + f_{37} + f_{37} + f_{48} &= 1/2.
\end{align*}
\]

\( \text{(A2.10)} \)

This reduces the amount of cases to consider from 430 to 36 and makes possible manual enumeration and analysis. The result of this analysis is presented as the right hand sides of the equalities (A2.5)–(A2.8) and (A2.10). It demonstrates that the family \( \{ f_r \} \) suggested in (A2.8) is proper. Moreover, the minimal \( \delta(x) \) for a non-deletable site \( x \) (see Figure 5) is

\[
\delta(x) = 1 - f_{19} - f_{31} = 1 - f_{21} - f_{27} = 31/56 > 1/3,
\]

\( \text{(A2.11)} \)

which implies the assertion of Lemma A2.1.

\[ \blacksquare \]

**Lemma A2.2.** For \( D = 13 \) and the non-inclined GS \( \varphi \), every inserted contour other than the contours described in the proof of Theorem 5 has statistical weight at most \( u^{-3} \).
Proof of Lemma A2.2. We follow the argument used in the proof of Lemma A2. The proposed family \( \{f_r\} \) of local repelling forces is

\[
\begin{align*}
f_1 &= 131/135, \quad f_3 = 127/135, \quad f_4 = 251/270, \quad f_7 = 241/270, \quad f_9 = 116/135, \\
f_{12} &= 37/45, \quad f_{13} = 221/270, \quad f_{16} = 7/9, \quad f_{19} = 133/180, \quad f_{21} = 383/540, \\
f_{25} &= 179/270, \quad f_{27} = 19/30, \quad f_{28} = 169/270, \quad f_{31} = 317/540, \quad f_{36} = 281/540, \\
f_{37} &= 14/27, \quad f_{39} = 131/270, \quad f_{43} = 41/90, \quad f_{48} = 37/90, \quad f_{49} = 43/108, \\
f_{52} &= 103/270, \quad f_{57} = 35/108, \quad f_{61} = 53/180, \quad f_{63} = 151/540, \quad f_{64} = 5/18, \\
f_{67} &= 7/27, \quad f_{73} = 119/540, \quad f_{75} = 11/54, \quad f_{76} = 109/540, \quad f_{79} = 11/60, \\
f_{81} &= 22/135, \quad f_{84} = 83/540, \quad f_{91} = 2/15, \quad f_{93} = 31/270, \quad f_{97} = 29/270, \\
f_{100} &= 1/9, \quad f_{103} = 4/45, \quad f_{108} = 2/27, \quad f_{109} = 2/27, \quad f_{111} = 7/108, \\
f_{112} &= 1/15, \quad f_{117} = 2/45, \quad f_{121} = 11/270, \quad f_{124} = 23/540, \quad f_{127} = 11/270, \\
f_{129} &= 4/135, \quad f_{133} = 4/135, \quad f_{139} = 2/135, \quad f_{144} = 1/54, \quad f_{147} = 2/135, \\
f_{148} &= 2/135, \quad f_{151} = 1/270, \quad f_{156} = 1/270, \quad f_{157} = 1/180 \quad f_{163} = 0.
\end{align*}
\] (A2.12)

The value of \( f_{157} \) in (A2.12) breaks the monotonicity of \( f_r \). But it is not important as even the number of minimal repelling types of removed sites is too large to be inspected manually. Using a computer it is possible to analyze all rather than minimal repelling types of removed sites. The enumeration of all repelling types of inserted sites turns out to be as follows:

\[
\begin{align*}
f_1 + f_{144} + 2f_{157} &= 1, \quad f_1 + f_{147} + f_{148} = 1, \\
f_3 + f_{127} + f_{148} + f_{151} &= 1, \quad f_3 + 2f_{133} = 1, \\
f_4 + f_{121} + 2f_{147} &= 1, \quad f_4 + f_{127} + f_{129} = 1, \\
f_7 + f_{108} + f_{133} + f_{151} &= 1, \quad f_7 + f_{109} + f_{129} + f_{156} = 1, \\
f_7 + f_{111} + f_{124} + f_{163} &= 1, \quad f_7 + f_{112} + f_{127} + f_{163} = 1, \\
f_9 + f_{100} + 2f_{139} &= 1, \quad f_9 + f_{109} + f_{112} = 1, \\
f_{12} + f_{91} + f_{133} + f_{139} &= 1, \quad f_{12} + 2f_{103} = 1, \\
f_{13} + f_{91} + f_{117} + f_{156} &= 1, \quad f_{13} + f_{93} + f_{112} + f_{163} = 1, \\
f_{16} + f_{81} + 2f_{133} &= 1, \quad f_{16} + f_{93} + f_{97} = 1, \\
f_{19} + f_{75} + f_{124} + f_{139} &= 1, \quad f_{19} + f_{76} + f_{117} + f_{147} = 1, \\
f_{20} + f_{84} + f_{97} &= 1, \quad f_{21} + f_{73} + f_{112} + f_{151} = 1, \\
f_{21} + f_{76} + f_{103} + f_{163} &= 1, \quad f_{21} + f_{79} + f_{97} = 1, \\
f_{25} + f_{64} + 2f_{129} &= 1, \quad f_{25} + f_{79} + f_{84} = 1, \\
f_{27} + f_{61} + f_{124} + f_{133} &= 1, \quad f_{27} + 2f_{79} = 1, \\
f_{28} + f_{61} + f_{111} + f_{147} &= 1, \quad f_{28} + f_{63} + f_{103} + f_{157} = 1, \\
f_{28} + f_{67} + f_{93} &= 1, \quad f_{31} + f_{57} + f_{109} + f_{148} = 1, \\
f_{31} + f_{63} + f_{91} &= 1, \quad f_{31} + f_{67} + f_{84} = 1, \\
f_{36} + f_{49} + 2f_{127} &= 1, \quad f_{36} + f_{67} + f_{73} = 1, \\
f_{37} + f_{48} + f_{121} + f_{133} &= 1, \quad f_{37} + f_{49} + f_{111} + f_{144} = 1, \\
f_{37} + f_{63} + f_{76} &= 1, \quad f_{39} + f_{49} + f_{100} + f_{157} = 1, \\
f_{39} + f_{52} + f_{91} &= 1, \quad 2f_{43} + f_{108} + f_{147} = 1, \\
f_{43} + f_{52} + f_{81} &= 1, \quad f_{43} + f_{57} + f_{73} = 1, \\
f_{48} + 2f_{61} &= 1, \quad 2f_{49} + f_{75} = 1, \\
f_{49} + f_{57} + f_{64} &= 1.
\end{align*}
\] (A2.13)

Cf. Figure 6 and compare with (A2.5)–(A2.8)).
The verification of the fact that the collection \( \{ f_r \} \) in (A2.12) is proper requires a computer assistance as there are 84943 distinct repelling types of removed sites (see Listing L.2 for the Java routine verifying that a family of repelling forces is proper).

After that it is not hard to see (by using a computer or just analyzing the right picture in Figure 6) that for non-inclined sublattice the the minimal \( \delta(x) \) for a non-deletable site \( x \) is

\[
\delta(x) = 1 - f_{28} - f_{133} = 93/270 > 1/3, \tag{A2.14}
\]

which implies the lemma. ■

**Lemma A2.3.** For \( D = \sqrt{147} \) and the \((11,2)\) inclined GS \( \varphi \), every inserted contour other than the contours described in the proof of Theorem 6 has statistical weight at most \( u^{-3} \).

**Proof of Lemma A2.3.** Again we follow the argument used in the proof of Lemma A2. The proposed monotone family \( \{ f_r \} \) of local repelling forces is

\[
\begin{align*}
  f_1 &= 24/24, & f_4 &= 24/24, & f_7 &= 23/24, & f_9 &= 22/24, \\
  f_{12} &= 21/24, & f_{13} &= 21/24, & f_{16} &= 20/24, & f_{19} &= 19/24, & f_{21} &= 18/24, \\
  f_{25} &= 16/24, & f_{27} &= 15/24, & f_{28} &= 15/24, & f_{31} &= 14/24, & f_{36} &= 12/24, \\
  f_{37} &= 12/24, & f_{39} &= 11/24, & f_{43} &= 10/24, & f_{48} &= 9/24, & f_{49} &= 8/24, \\
  f_{52} &= 7/24, & f_{57} &= 6/24, & f_{61} &= 5/24, & f_{63} &= 4/24, & f_{64} &= 4/24, \\
  f_{67} &= 4/24, & f_{73} &= 3/24, & f_{75} &= 3/24, & f_{76} &= 2/24, & f_{79} &= 2/24, \\
  f_{81} &= 2/24, & f_{84} &= 2/24, & f_{91} &= 1/24, & f_{93} &= 1/24, & f_{97} &= 1/24,
\end{align*}
\]

with \( f_r = 0 \) for \( r > 97 \).

The computer analysis of all repelling types of inserted sites leads to the following equations:

\[
\begin{align*}
  f_1 + f_{124} + f_{133} + f_{139} &= 1, & f_1 + f_{127} + f_{127} &= 1 \\
  f_3 + f_{108} + f_{129} + f_{129} &= 1, & f_3 + f_{111} + f_{117} + f_{144} &= 1 \\
  f_4 + f_{103} + f_{121} + f_{133} &= 1, & f_4 + f_{109} + f_{109} &= 1 \\
  f_7 + f_{91} + f_{112} + f_{133} &= 1, & f_7 + f_{97} + f_{100} &= 1 \\
  f_9 + f_{84} + f_{111} + f_{129} &= 1, & f_9 + f_{93} + f_{93} &= 1 \\
  f_{12} + f_{75} + f_{117} + f_{117} &= 1, & f_{12} + f_{81} + f_{93} &= 1 \\
  f_{13} + f_{73} + f_{109} + f_{124} &= 1, & f_{13} + f_{76} + f_{97} + f_{139} &= 1 \\
  f_{13} + f_{79} + f_{91} &= 1, & f_{16} + f_{67} + f_{103} + f_{127} &= 1 \\
  f_{16} + f_{79} + f_{79} &= 1, & f_{19} + f_{61} + f_{103} + f_{124} &= 1 \\
  f_{19} + f_{64} + f_{91} + f_{139} &= 1, & f_{19} + f_{73} + f_{76} &= 1 \\
  f_{21} + f_{57} + f_{108} + f_{117} &= 1, & f_{21} + f_{63} + f_{84} &= 1 \\
  f_{25} + f_{52} + f_{97} + f_{127} &= 1, & f_{25} + f_{67} + f_{67} &= 1 \\
  f_{27} + f_{48} + f_{111} + f_{111} &= 1, & f_{27} + f_{57} + f_{75} &= 1 \\
  f_{28} + f_{49} + f_{91} + f_{133} &= 1, & f_{28} + f_{61} + f_{67} &= 1 \\
  f_{31} + f_{43} + f_{109} + f_{112} &= 1, & f_{31} + f_{49} + f_{79} &= 1 \\
  f_{31} + f_{52} + f_{73} &= 1, & f_{36} + f_{39} + f_{93} + f_{129} &= 1
\end{align*}
\]
and
\[
\begin{align*}
\begin{align*}
& f_{36} + f_{57} + f_{57} = 1, & f_{37} + f_{37} + f_{100} + f_{121} = 1 \\
& f_{37} + f_{43} + f_{76} = 1, & f_{37} + f_{52} + f_{61} = 1 \\
& f_{39} + f_{39} + f_{81} + f_{144} = 1, & f_{39} + f_{48} + f_{63} = 1 \\
& f_{43} + f_{43} + f_{64} = 1, & f_{49} + f_{49} + f_{49} = 1.
\end{align*}
\end{align*}
\]

The verification of the fact that the collection \( \{ f_r \} \) in (A2.12) is proper requires a computer assistance as there are 54400 distinct repelling types (and 300 minimal repelling types) of removed sites (see Listing L.2 for the Java routine checking that a family of local repelling forces is proper).

After that it is not hard to see (using a computer or just analyzing the left picture in Figure 8) that for the (11, 2) inclined sublattice the minimal \( \delta(x) \) for a non-deletable site \( x \) is
\[
\delta(x) = 1 - f_{37} - f_{100} = 1/2 > 1/3,
\]
which implies the assertion of the lemma.

Remark. In essence, our Lemmas A2.1 – A2.3 allow us to check that the hard-core repulsion potential for \( D = 7 \), \( D = 13 \) and \( D = \sqrt{147} \) generate \( M \)-potentials in the terminology adopted in [I].

8 Appendix 3. Listings.

This section collects the listings of Java routines used to assist the proofs of Lemmas in Appendix I. They can be compiled and executed on any computer hosting Java Development Kit version 1.4 and later. The routine from Listing 2 requires 3GB of RAM and takes few minutes to execute. The other routines are not resource-hungry.

Listing 1 specifies a routine calculating, for a given sublattice, the amount of distinguishing small contours of the types used in the proofs of Theorems 4, 5 and 6. The execution results are presented only for \( D = 7 \), \( D = 13 \) and \( D = \sqrt{147} \) and the sublattices used in the proof of Lemmas A2.1, A2.2 and A2.3.

Listing 2 contains a routine that determines, for any \( D \) and any family of local repelling forces \( \{ f_r \} \), if this family is proper. First, the routine enumerates all repelling types of inserted and removed sites. Then, for each type it verifies if the given family of local repelling forces is proper. As above, the execution results are presented for \( D = 7 \), \( D = 13 \) and \( \sqrt{147} \) only.

Listing 3 contains a routine that calculates, for a given sublattice and a local repelling force family, the corresponding minimal value of \( \delta \) for non-deletable \( D \)-parallelograms (see the proofs of Lemmas A2.1, A2.2 and A2.3). Again, the execution results are presented only for \( D = 7 \), \( D = 13 \) and \( D = \sqrt{147} \), and the sublattices used in the proof of Lemmas A2.1, A2.2 and A2.3.
Listing 1.

class CountExcitations {
    
    public static int distanceSquared(int[] siteA, int[] siteB) {
        return (siteA[0] - siteB[0]) * (siteA[0] - siteB[0])
                + (siteA[0] - siteB[0]) * (siteA[1] - siteB[1])
                + (siteA[1] - siteB[1]) * (siteA[1] - siteB[1]);
    }

    public static int[] triangleSiteC(int[] siteA, int[] siteB) {
        return new int[]{siteA[0] + siteA[1] - siteB[1], siteB[0] + siteB[1] - siteA[0]};
    }

    public static int[][] sitesInTriangle(int[] siteA, int[] siteB) {
        int sizeSquared = distanceSquared(siteA, siteB);
        int[] siteC = triangleSiteC(siteA, siteB);
        int left = Math.min(Math.min(siteA[0], siteB[0]), siteC[0]);
        int right = Math.max(Math.max(siteA[0], siteB[0]), siteC[0]);
        int bottom = Math.min(Math.min(siteA[1], siteB[1]), siteC[1]);
        int top = Math.max(Math.max(siteA[1], siteB[1]), siteC[1]);
        int[][] repellingSites = new int[][]{(siteB, siteA), triangleSiteC(siteA, siteC), triangleSiteC(siteC, siteB)};
        int count = 0;
        for (int i = left+1; i < right; i++)
            for (int j = bottom+1; j < top; j++)
                if (distanceSquared(new int[]{i, j}, repellingSites[0]) >= sizeSquared
                    && distanceSquared(new int[]{i, j}, repellingSites[1]) >= sizeSquared
                    && distanceSquared(new int[]{i, j}, repellingSites[2]) >= sizeSquared
                ) count++;
        int[][] sites = new int[count][2];
        count = -1;
    }
}
for (int i = left+1; i < right; i++)
for (int j = bottom+1; j < top; j++)
if (distanceSquared(new int[]{i,j}, repellingSites[0]) >= sizeSquared
 && distanceSquared(new int[]{i,j}, repellingSites[1]) >= sizeSquared
 && distanceSquared(new int[]{i,j}, repellingSites[2]) >= sizeSquared
 ) {count++; sites[count][0] = i; sites[count][1] = j;}

return sites;
}

public static int enumeratePairs(int[][] sitesInTriangleA, int[][] sitesInTriangleB, int sizeSquared )
{
int count = 0;
for (int i = 0; i < sitesInTriangleA.length; i++)
for (int j = 0; j < sitesInTriangleB.length; j++)
if (distanceSquared(sitesInTriangleA[i], sitesInTriangleB[j]) >= sizeSquared) count++;
return count;
}

public static int enumerateTriples(int[][] sitesInTriangleA,
     int[][] sitesInTriangleB, int[][] sitesInTriangleC, int sizeSquared )
{
int count = 0;
for (int i = 0; i < sitesInTriangleA.length; i++)
for (int j = 0; j < sitesInTriangleB.length; j++)
for (int k = 0; k < sitesInTriangleC.length; k++)
if ( distanceSquared(sitesInTriangleA[i], sitesInTriangleB[j]) >= sizeSquared
 && distanceSquared(sitesInTriangleA[i], sitesInTriangleC[k]) >= sizeSquared
 && distanceSquared(sitesInTriangleB[j], sitesInTriangleC[k]) >= sizeSquared ) count++;
return count;
}
public static int enumerateQuadruples(int[][] sitesInTriangleA, int[][] sitesInTriangleB,  
    int[][] sitesInTriangleC, int[][] sitesInTriangleD, int sizeSquared)
{
    int count = 0;
    for (int i = 0; i < sitesInTriangleA.length; i++)
        for (int j = 0; j < sitesInTriangleB.length; j++)
            for (int k = 0; k < sitesInTriangleC.length; k++)
                for (int l = 0; l < sitesInTriangleD.length; l++)
                    if (distanceSquared(sitesInTriangleA[i], sitesInTriangleB[j]) >= sizeSquared
                        && distanceSquared(sitesInTriangleA[i], sitesInTriangleC[k]) >= sizeSquared
                        && distanceSquared(sitesInTriangleA[i], sitesInTriangleD[l]) >= sizeSquared
                        && distanceSquared(sitesInTriangleB[j], sitesInTriangleC[k]) >= sizeSquared
                        && distanceSquared(sitesInTriangleB[j], sitesInTriangleD[l]) >= sizeSquared
                        && distanceSquared(sitesInTriangleC[k], sitesInTriangleD[l]) >= sizeSquared ) count++;
    return count;
}

public static void enumerateForSublattice(int a, int b) {
    int sizeSquared = a * a + a * b + b * b;
    int[][] sitesInTriangleA = sitesInTriangle(new int[]{0,0}, new int[]{a,b});
    int[][] sitesInTriangleB = sitesInTriangle(new int[]{a,b}, new int[]{0,0});
    System.out.println("Number of admissible pairs for (" + a + "," + b + ") sublattice= "
        + enumeratePairs(sitesInTriangleA, sitesInTriangleB, sizeSquared));
    int[][] sitesInTriangleC = sitesInTriangle(new int[]{-b,a+b}, new int[]{a,b});
    System.out.println("Number of admissible triples for (" + a + "," + b + ") sublattice= "
        + enumerateTriples(sitesInTriangleA, sitesInTriangleB, sitesInTriangleC, sizeSquared));
    int[][] sitesInTriangleD = sitesInTriangle(new int[]{0,0}, new int[]{-b,a+b});
    System.out.println("Number of admissible quadruples for (" + a + "," + b + ") sublattice= "
        + enumerateQuadruples(sitesInTriangleA, sitesInTriangleB, sitesInTriangleC, sitesInTriangleD, sizeSquared));
}

public static void main(String[] args) {
enumerateForSublattice(8, 7);
enumerateForSublattice(13, 0);
enumerateForSublattice(11, 2);
enumerateForSublattice(7, 7);
enumerateForSublattice(5, 3);
enumerateForSublattice(7, 0);
}
}

Number of admissible pairs for (8,7) sublattice= 113
Number of admissible triples for (8,7) sublattice= 61
Number of admissible quadruples for (8,7) sublattice= 39
Number of admissible pairs for (13,0) sublattice= 78
Number of admissible triples for (13,0) sublattice= 20
Number of admissible quadruples for (13,0) sublattice= 3
Number of admissible pairs for (11,2) sublattice= 51
Number of admissible triples for (11,2) sublattice= 1
Number of admissible quadruples for (11,2) sublattice= 0
Number of admissible pairs for (7,7) sublattice= 86
Number of admissible triples for (7,7) sublattice= 39
Number of admissible quadruples for (7,7) sublattice= 16
Number of admissible pairs for (5,3) sublattice= 6
Number of admissible triples for (5,3) sublattice= 0
Number of admissible quadruples for (5,3) sublattice= 0
Number of admissible pairs for (7,0) sublattice= 7
Number of admissible triples for (7,0) sublattice= 0
Number of admissible quadruples for (7,0) sublattice= 0

Listing 2.

import org.apache.commons.math3.fraction.BigFraction;

public class VeririfyRepellingForces
{ //the family of repelling forces
    public static BigFraction[] f = new BigFraction[169];

    //it is assumed that each site is a pair of lattice coordinates
    //site[0] and site[1]
    public static int distSquared(int[] siteA, int[] siteB) {
        return (siteA[0] - siteB[0]) * (siteA[0] - siteB[0])
            + (siteA[0] - siteB[0]) * (siteA[1] - siteB[1])
            + (siteA[1] - siteB[1]) * (siteA[1] - siteB[1]);
    }

    //it is assumed a >= b >= 0
    public static boolean verifyTriangleTuples(int a, int b) {
        int[][] cornerSites = new int[][]{{0,0}, {a,b}, {-b,a+b}};
        int[][] repellingSites = new int[][]{{a-b,a+2*b}, {-a-b,a}, {a+b,-a}};
        int sizeSquared = a * a + a * b + b * b;

        for (int i = -b+1; i < a; i++)
            for (int j = 1; j < a+b; j++)
                if (distSquared(new int[]{i,j}, repellingSites[0]) >= sizeSquared
                    && distSquared(new int[]{i,j}, repellingSites[1]) >= sizeSquared
                    && distSquared(new int[]{i,j}, repellingSites[2]) >= sizeSquared
                    && f[distSquared(new int[]{i,j}, cornerSites[0])]
                        .add(f[distSquared(new int[]{i,j}, cornerSites[1])])
                        .add(f[distSquared(new int[]{i,j}, cornerSites[2])])
                        .compareTo(BigFraction.ONE) != 0) return false;

        return true;
    }

    //it is assumed a >= b >= 0
public static boolean verifyLensTuples(int a, int b)
{
  int[][] cornerSites = new int[][]{{0,0}, {a,b}, {-b,a+b}, {a-b,a+2*b}};
  int[][] attractingSites = new int[][]{{0,0}, {a-b,a+2*b}};

  int sizeSquared = a * a + a * b + b * b;

  for (int i = -b+1; i < a; i++)
    for (int j = b+1; j < a+b; j++)
      if (distSquared(new int[] {i,j}, attractingSites[0]) < sizeSquared
          && distSquared(new int[] {i,j}, attractingSites[1]) < sizeSquared
          && f[distSquared(new int[] {i,j}, cornerSites[0])]
              .add(f[distSquared(new int[] {i,j}, cornerSites[1])])
              .add(f[distSquared(new int[] {i,j}, cornerSites[2])])
              .add(f[distSquared(new int[] {i,j}, cornerSites[3])])
              .compareTo(BigFraction.ONE) != 0) return false;
  return true;
}

//transform angles in [-pi, pi] into angles in [0, 2 pi]
public static double trueAngle(double angle)
{
  if (angle < 0)
    return 2 * Math.PI + angle;
  else
    return angle;
}

//compare two lattice sites according to the order
//based on polar coordinates
//it is assumed that each site is a pair of lattice coordinated site[0], site[1]
//and the corresponding squared distance from the origin site[3]
public static boolean isSiteALarger(int[] siteA, int[] siteB)
{
    if (siteA[2] > siteB[2]) return true;
    if (siteA[2] < siteB[2]) return false;
    if (trueAngle(Math.atan2(siteA[1] * Math.sqrt(3)/2, siteA[0] + siteA[1]/2)) >=
        trueAngle(Math.atan2(siteB[1] * Math.sqrt(3)/2, siteB[0] + siteB[1]/2))) return true;
    return false;
}

//enumerates all admissible ordered tuples of sites
//inside a disk centered at origin
//and verify that the force family is proper
public static boolean verifyRepellingTuples(int sizeSquared)
{
    int size = (int)Math.sqrt(sizeSquared * 4 / 3) + 1;
    int maxTupleSize = 5;

    //the tuples are stored as an array with the following indexes
    //[][][][] - specifies the size of the tuple
    //[][][] - specifies the tuple in given array of tuples of given size
    //[][] - specifies the site inside given tuple
    //[][][] - specifies coordinates of given site (the last coordinate is squared distance from the origin)
    int[][][] tuples = new int[maxTupleSize][][[]][];

    //count all sites inside D-disk
    int count = 0;
    for (int i = -size; i < size; i++)
        for (int j = -size; j < size; j++)
            if (!(i == 0 && j == 0) && distSquared(new int[]{0,0}, new int[]{i,j}) < sizeSquared) count++;

    //enumerate all sites inside D-disk
    int[][] sites = new int[count][3];
count = -1;
for (int i = -size; i < size; i++)
    for (int j = -size; j < size; j++)
        if (!(i == 0 && j == 0) && distSquared(new int[]{0,0}, new int[]{i,j}) < sizeSquared) {
            count++;
            sites[count] = new int[]{i, j, i*i+i*j+j*j};
        }

//sort sites
for (int i = 0; i < sites.length; i++)
    for (int j = i+1; j < sites.length; j++)
        if (isSiteALarger(sites[i], sites[j])) {
            int[] site = {sites[i][0], sites[i][1], sites[i][2]};
            for (int k = 0; k < 3; k++) {sites[i][k] = sites[j][k]; sites[j][k] = site[k];}
        }

//start iterative (in tuple size) enumeration process
//by specifying 1-tuples
tuples[0] = new int[sites.length][1][3];
for (int i = 0; i < sites.length; i++) tuples[0][i][0] = sites[i];

//enumerate inside the disk all tuples of size t+1
//assuming that all tuples of size t are already known
for (int t = 1; t < maxTupleSize; t++ )
{
    //count all admissible (t+1)-tuples of sites inside D-disk
    count = 0;
    for (int i = 0; i < tuples[t-1].length; i++)
        for (int j = 0; j < sites.length; j++) {
            boolean isAllowed = true;
            for (int s = 0; s < t; s++)
                if (distSquared(tuples[t-1][i][s], sites[j]) < sizeSquared) {isAllowed = false; break;}
            if (isAllowed) count++;
        }
}
if (isAllowed && isSiteALarger(sites[j], tuples[t-1][i][t-1])) count++;
}

//Enumerate all admissible (t+1)-tuples of sites inside D-disk
tuples[t] = new int[count][t+1][3];
count = -1;
for (int i = 0; i < tuples[t-1].length; i++) {
  for (int j = 0; j < sites.length; j++) {
    boolean isAllowed = true;
    for (int s = 0; s < t; s++)
      if (distSquared(tuples[t-1][i][s], sites[j]) < sizeSquared) {isAllowed = false; break;}
    if (isAllowed && isSiteALarger(sites[j], tuples[t-1][i][t-1])) {
      count++;
      tuples[t][count] = new int[t+1][];
      for (int s = 0; s < t; s++)
        tuples[t][count][s] = tuples[t-1][i][s];
      tuples[t][count][t] = sites[j];
    }
  }
}

//verify the family of repelling forces
for (int t = 1; t < maxTupleSize; t++)
  for (int i = 0; i < tuples[t].length; i++) {
    BigFraction force = BigFraction.ZERO;
    for (int j = 0; j < tuples[t][i].length; j++) force = force.add(f[tuples[t][i][j][2]]);
    if (force.compareTo(BigFraction.ONE) > 0) return false;
  }
return true;
public static void main(String[] args)
{
    for (int i = 0; i < f.length; i++) f[i] = BigFraction.ZERO;

    f[1] = new BigFraction(131, 135); f[3] = new BigFraction(127, 135); f[4] = new BigFraction(251, 270);
    f[7] = new BigFraction(241, 270); f[9] = new BigFraction(116, 135); f[12] = new BigFraction(37, 45);
    f[13] = new BigFraction(221, 270); f[16] = new BigFraction(7, 9); f[19] = new BigFraction(133, 180);
    f[21] = new BigFraction(383, 540); f[25] = new BigFraction(179, 270); f[27] = new BigFraction(19, 30);
    f[28] = new BigFraction(169, 270); f[31] = new BigFraction(317, 540); f[36] = new BigFraction(281, 540);
    f[37] = new BigFraction(14, 27); f[39] = new BigFraction(131, 270); f[43] = new BigFraction(41, 90);
    f[48] = new BigFraction(37, 90); f[49] = new BigFraction(43, 108); f[52] = new BigFraction(103, 270);
    f[57] = new BigFraction(35, 108); f[61] = new BigFraction(53, 180); f[63] = new BigFraction(151, 540);
    f[64] = new BigFraction(5, 18); f[67] = new BigFraction(7, 27); f[73] = new BigFraction(119, 540);
    f[75] = new BigFraction(11, 54); f[76] = new BigFraction(109, 540); f[79] = new BigFraction(11, 60);
    f[81] = new BigFraction(22, 135); f[84] = new BigFraction(83, 540); f[91] = new BigFraction(2, 15);
    f[93] = new BigFraction(31, 270); f[97] = new BigFraction(29, 270); f[100] = new BigFraction(1, 9);
    f[103] = new BigFraction(4, 45); f[108] = new BigFraction(2, 27); f[109] = new BigFraction(2, 27);
    f[111] = new BigFraction(7, 108); f[112] = new BigFraction(1, 15); f[117] = new BigFraction(2, 45);
    f[121] = new BigFraction(11, 270); f[124] = new BigFraction(23, 540); f[127] = new BigFraction(11, 270);
    f[129] = new BigFraction(4, 135); f[133] = new BigFraction(4, 135); f[139] = new BigFraction(2, 135);
    f[144] = new BigFraction(1, 54); f[147] = new BigFraction(2, 135); f[148] = new BigFraction(2, 135);
    f[151] = new BigFraction(1, 270); f[156] = new BigFraction(1, 270); f[157] = new BigFraction(1, 180);

    System.out.println("Sublattice (13,0):");
    System.out.println("The force family is proper for inserted types which are triples - "+ verifyTriangleTuples(13,0));
    System.out.println("The force family is proper for inserted types which are quadruples - "+ verifyLensTuples(13,0));
    System.out.println("Sublattice (8,7):");
    System.out.println("The force family is proper for inserted types which are triples - "+ verifyTriangleTuples(8,7));
    System.out.println("The force family is proper for inserted types which are quadruples - ");
System.out.println("Both sublattices (13,0) and (8,7). The force family is proper for all removed types - " + verifyRepellingTuples(169));

for (int i = 0; i < f.length; i++) f[i] = BigFraction.ZERO;

f[1] = new BigFraction(1); f[3] = new BigFraction(1); f[4] = new BigFraction(1);
f[7] = new BigFraction(23, 24); f[9] = new BigFraction(11, 12); f[12] = new BigFraction( 7, 8);
f[13] = new BigFraction( 7, 8); f[16] = new BigFraction( 5, 6); f[19] = new BigFraction(19, 24);
f[21] = new BigFraction( 3, 4); f[25] = new BigFraction( 2, 3); f[27] = new BigFraction( 5, 8);
f[28] = new BigFraction( 5, 8); f[31] = new BigFraction( 7, 12); f[36] = new BigFraction( 1, 2);
f[37] = new BigFraction( 1, 2); f[39] = new BigFraction(11, 24); f[43] = new BigFraction( 5, 12);
f[48] = new BigFraction( 3, 8); f[49] = new BigFraction( 1, 3); f[52] = new BigFraction( 7, 24);
f[57] = new BigFraction( 1, 4); f[61] = new BigFraction( 5, 24); f[63] = new BigFraction( 1, 6);
f[64] = new BigFraction( 1, 6); f[67] = new BigFraction( 1, 6); f[73] = new BigFraction( 1, 8);
f[75] = new BigFraction( 1, 8); f[76] = new BigFraction( 1, 12); f[79] = new BigFraction( 1, 12);
f[81] = new BigFraction( 1, 12); f[84] = new BigFraction( 1, 12); f[91] = new BigFraction( 1, 24);
f[93] = new BigFraction( 1, 24); f[97] = new BigFraction( 1, 24);

System.out.println("Sublattice (7,7): ");
System.out.println("The force family is proper for inserted types which are triples - " + verifyTriangleTuples(7,7));
System.out.println("The force family is proper for inserted types which are quadruples - " + verifyLensTuples(7,7));
System.out.println("Sublattice (11,2): ");
System.out.println("The force family is proper for inserted types which are triples - " + verifyTriangleTuples(11,2));
System.out.println("The force family is proper for inserted types which are quadruples - " + verifyLensTuples(11,2));
System.out.println("Both sublattices (7,7) and (11,2). The force family is proper for all removed types - " + verifyRepellingTuples(147));

for (int i = 0; i < f.length; i++) f[i] = BigFraction.ZERO;
System.out.println("Sublattice (7,0):");
System.out.println("The force family is proper for inserted types which are triples - "+ verifyTriangleTuples(7,0));
System.out.println("The force family is proper for inserted types which are quadruples - "+ verifyLensTuples(7,0));
System.out.println("Sublattice (5,3):");
System.out.println("The force family is proper for inserted types which are triples - "+ verifyTriangleTuples(5,3));
System.out.println("The force family is proper for inserted types which are quadruples - "+ verifyLensTuples(5,3));
System.out.println("Both sublattices (7,0) and (5,3).
The force family is proper for all removed types - "+ verifyRepellingTuples(49));
Both sublattices (13,0) and (8,7).
The force family is proper for all removed types - true

Sublattice (7,7):
The force family is proper for inserted types which are triples - true
The force family is proper for inserted types which are quadruples - true

Sublattice (11,2):
The force family is proper for inserted types which are triples - true
The force family is proper for inserted types which are quadruples - true

Both sublattices (7,7) and (11,2).
The force family is proper for all removed types - true

Sublattice (7,0):
The force family is proper for inserted types which are triples - true
The force family is proper for inserted types which are quadruples - true

Sublattice (5,3):
The force family is proper for inserted types which are triples - true
The force family is proper for inserted types which are quadruples - true

Both sublattices (7,0) and (5,3).
The force family is proper for all removed types - true

Listing 3.

import org.apache.commons.math3.fraction.BigFraction;

public class CountMinDelta {

    public static BigFraction[] f = new BigFraction[169];
public static int distanceSquared(int[] siteA, int[] siteB) {
    return (siteA[0] - siteB[0]) * (siteA[0] - siteB[0])
        + (siteA[0] - siteB[0]) * (siteA[1] - siteB[1])
        + (siteA[1] - siteB[1]) * (siteA[1] - siteB[1]);
}

public static int[] triangleSiteC(int[] siteA, int[] siteB) {
    return new int[]{siteA[0] + siteA[1] - siteB[1], siteB[0] + siteB[1] - siteA[0]};
}

public static int[][] sitesInTriangle(int[] siteA, int[] siteB) {
    int sizeSquared = distanceSquared(siteA, siteB);
    int[] siteC = triangleSiteC(siteA, siteB);
    int left = Math.min(Math.min(siteA[0], siteB[0]), siteC[0]);
    int right = Math.max(Math.max(siteA[0], siteB[0]), siteC[0]);
    int bottom = Math.min(Math.min(siteA[1], siteB[1]), siteC[1]);
    int top = Math.max(Math.max(siteA[1], siteB[1]), siteC[1]);
    int[][] repellingSites = new int[][]{
        triangleSiteC(siteB, siteA),
        triangleSiteC(siteA, siteC),
        triangleSiteC(siteC, siteB)};
    int count = 0;
    for (int i = left+1; i < right; i++)
        for (int j = bottom+1; j < top; j++)
            if (distanceSquared(new int[] {i,j}, repellingSites[0]) >= sizeSquared
                && distanceSquared(new int[] {i,j}, repellingSites[1]) >= sizeSquared
                && distanceSquared(new int[] {i,j}, repellingSites[2]) >= sizeSquared)
                count++;
    int[] sites = new int[count][2];
    count = -1;
    for (int i = left+1; i < right; i++)
        for (int j = bottom+1; j < top; j++)
            if (distanceSquared(new int[] {i,j}, repellingSites[0]) >= sizeSquared
                && distanceSquared(new int[] {i,j}, repellingSites[1]) >= sizeSquared
                && distanceSquared(new int[] {i,j}, repellingSites[2]) >= sizeSquared)
& \& \text{distanceSquared(new int\[\] \{i,j\}, repellingSites[2]) >= sizeSquared}
 ) {count++; sites[count][0] = i; sites[count][1] = j;}

return sites;
}

public static String enumeratePairs(int\[\]\[\] sitesInTriangleA, int\[\]\[\] sitesInTriangleB, int\[] removedSite, int sizeSquared) {
    BigFraction maxForce = BigFraction.ZERO;
    int distancesSquaredA = 0;
    int distancesSquaredB = 0;

    for (int i = 0; i < sitesInTriangleA.length; i++)
        for (int j = 0; j < sitesInTriangleB.length; j++)
            if (distanceSquared(sitesInTriangleA[i], sitesInTriangleB[j]) >= sizeSquared) {
                BigFraction force = f[distanceSquared(sitesInTriangleA[i], removedSite)].add(f[distanceSquared(sitesInTriangleB[j], removedSite)]);
                if (force.compareTo(maxForce) > 0) {
                    maxForce = force;
                    distancesSquaredA = distanceSquared(sitesInTriangleA[i], removedSite);
                    distancesSquaredB = distanceSquared(sitesInTriangleB[j], removedSite);
                }
            }

    return "minimal delta = 1 - f[" + distancesSquaredA + "] - f[" + distancesSquaredB + "] = "
           + BigFraction.ONE.subtract(maxForce);
}

public static void main(String\[] args) {

    for (int i = 0; i < f.length; i++) f[i] = BigFraction.ZERO;
for (int i = 0; i < f.length; i++) f[i] = BigFraction.ZERO;

// inclined sublattice
int[][] sitesInTriangleA = sitesInTriangle(new int[]{0,0}, new int[]{8,7});
sitesInTriangleB = sitesInTriangle(new int[]{8,7}, new int[]{0,0});
System.out.println("For inclined sublattice (8,7) "+ enumeratePairs(sitesInTriangleA, sitesInTriangleB, new int[]{8, 7}, 169));

//non-inclined sublattice
sitesInTriangleA = sitesInTriangle(new int[]{0,0}, new int[]{13,0});
sitesInTriangleB = sitesInTriangle(new int[]{13,0}, new int[]{0,0});
System.out.println("For non-inclined sublattice (13,0) "+ enumeratePairs(sitesInTriangleA, sitesInTriangleB, new int[]{13,0}, 169));
\[ f[1] = \text{new BigFraction}(1); \quad f[3] = \text{new BigFraction}(1); \quad f[4] = \text{new BigFraction}(1); \]
\[ f[7] = \text{new BigFraction}(23, 24); f[9] = \text{new BigFraction}(11, 12); f[12] = \text{new BigFraction}(7, 8); \]
\[ f[13] = \text{new BigFraction}(7, 8); f[16] = \text{new BigFraction}(5, 6); f[19] = \text{new BigFraction}(19, 24); \]
\[ f[21] = \text{new BigFraction}(3, 4); f[25] = \text{new BigFraction}(2, 3); f[27] = \text{new BigFraction}(5, 8); \]
\[ f[28] = \text{new BigFraction}(5, 8); f[31] = \text{new BigFraction}(7, 12); f[36] = \text{new BigFraction}(1, 2); \]
\[ f[37] = \text{new BigFraction}(1, 2); f[39] = \text{new BigFraction}(11, 24); f[43] = \text{new BigFraction}(5, 12); \]
\[ f[48] = \text{new BigFraction}(3, 8); f[49] = \text{new BigFraction}(1, 3); f[52] = \text{new BigFraction}(7, 24); \]
\[ f[57] = \text{new BigFraction}(1, 4); f[61] = \text{new BigFraction}(5, 24); f[63] = \text{new BigFraction}(1, 6); \]
\[ f[64] = \text{new BigFraction}(1, 6); f[67] = \text{new BigFraction}(1, 8); f[73] = \text{new BigFraction}(1, 8); \]
\[ f[75] = \text{new BigFraction}(1, 8); f[76] = \text{new BigFraction}(1, 12); f[79] = \text{new BigFraction}(1, 12); \]
\[ f[81] = \text{new BigFraction}(1, 12); f[84] = \text{new BigFraction}(1, 12); f[91] = \text{new BigFraction}(1, 24); \]
\[ f[93] = \text{new BigFraction}(1, 24); f[97] = \text{new BigFraction}(1, 24); \]

```
// inclined sublattice
sitesInTriangleA = sitesInTriangle(new int[]{0,0}, new int[]{11,2});
sitesInTriangleB = sitesInTriangle(new int[]{11,2}, new int[]{0,0});
System.out.println("For inclined sublattice (11,2) "
    + enumeratePairs(sitesInTriangleA, sitesInTriangleB, new int[]{11, 2}, 147));

// diagonal sublattice
sitesInTriangleA = sitesInTriangle(new int[]{0,0}, new int[]{7,7});
sitesInTriangleB = sitesInTriangle(new int[]{7,7}, new int[]{0,0});
System.out.println("For diagonal sublattice (7,7) "
    + enumeratePairs(sitesInTriangleA, sitesInTriangleB, new int[]{7,7}, 147));
```

```
for (int i = 0; i < f.length; i++) f[i] = BigFraction.ZERO;
```
f[37] = new BigFraction( 4, 56); f[39] = new BigFraction( 4, 56); f[43] = new BigFraction( 4, 56);
f[48] = new BigFraction( 4, 56);

//inclined sublattice
sitesInTriangleA = sitesInTriangle(new int[]{0,0}, new int[]{5,3});
sitesInTriangleB = sitesInTriangle(new int[]{5,3}, new int[]{0,0});
System.out.println("For inclined sublattice (5,3) "+ enumeratePairs(sitesInTriangleA, sitesInTriangleB, new int[]{5, 3}, 49));

//non-inclined sublattice
sitesInTriangleA = sitesInTriangle(new int[]{0,0}, new int[]{7,0});
sitesInTriangleB = sitesInTriangle(new int[]{7,0}, new int[]{0,0});
System.out.println("For diagonal sublattice (7,0) "+ enumeratePairs(sitesInTriangleA, sitesInTriangleB, new int[]{7,0}, 49));

}

For inclined sublattice (8,7) minimal delta = 1 - f[21] - f[148] = 149 / 540
For non-inclined sublattice (13,0) minimal delta = 1 - f[133] - f[28] = 31 / 90
For inclined sublattice (11,2) minimal delta = 1 - f[37] - f[100] = 1 / 2
For diagonal sublattice (7,7) minimal delta = 1 - f[127] - f[21] = 1 / 4
For inclined sublattice (5,3) minimal delta = 1 - f[28] - f[21] = 31 / 56
For diagonal sublattice (7,0) minimal delta = 1 - f[28] - f[21] = 31 / 56
Acknowledgement  IS and YS thank the Math Department, Penn State University, for hospitality and support. YS thanks St John’s College, Cambridge, for long-term support.

References

[1] Bricmont, J., Slawny, J. Phase transitions in systems with a finite number of dominant ground states. *J. Stat. Physics*, **54**, (1989), 89-161.

[2] Cohn, H., Kumar, A.D., Miller, S.D., Radchenko, D., Viazovska, M. The sphere packing problem in dimension 24. *Annals of Math.*, **185** (2017), 1017-1033.

[3] Connelly, R., W. Dickinson. Periodic planar disc packings. *Phil. Trans. Roy. Soc.*, **A372**, 20120039. [http://dx.doi.org/10.1098/rsta.2012.0039](http://dx.doi.org/10.1098/rsta.2012.0039)

[4] Conway, J., Sloane, N. *Sphere packings, lattices and groups*, 3rd Ed (With contributions by E. Bannai, R.E. Borcherds, J. Leech, S.P. Norton, A.M. Odlyzko, R.A. Parker, L. Queen and B. B. Venkov.) New York: Springer, 1999.

[5] Fejes Tóth, L. Some packing and covering theorems. *Acta Sci. Math.*, **12A** (1950), 62-67.

[6] Georgii, H.-O. *Gibbs measures and phase transitions*. De Gruyter, 2011.

[7] Hales, T.C. A proof of the Kepler conjecture. *Annals of Math.*, **162** (2005), 1065-1185.

[8] Hsiang, W-Y. Simple proof of a theorem of Thue on the maximal density of circle packings in $E^2$, *L’Enseignement Mathématique*, **38** (1992), 125-131.

[9] Jauslin, I., Lebowitz, J.L. High-fugacity expansion, Lee-Yang zeros and order-disorder transitions in hard-core lattice systems. arXiv: 1708.0912 (2017).

[10] Misaghian, M. Factor rings and their decompositions in the Eisenstein integers ring $\mathbb{Z}[\omega]$. *Armenian J. of Mathematics*, **5**:1 (2013), 58–68.

[11] Nair, U.P. Elementary results on the binary quadratic form $a^2 + ab + b^2$. arXiv: math.0408107v1, 2004

[12] Pirogov, S.A., Sinai, Ya.G. Phase diagrams of classical lattice systems. *Teor. Mat. Fiz.* **25** (1975), 1185-1192; **26** (1976), 61-76.

[13] Seiler, E. *Gauge theories as a problem of Constructive Quantum field theory and Statistical mechanics*. Lecture Notes in Physics, **159**. Berlin: Springer, 1982.

[14] Sinai, Ya. G. *Theory of phase transitions: rigorous results*. Oxford et al.: Pergamon Press, 1982.

[15] Vyazovskaya, M. The sphere packing problem in dimension 8. *Annals of Math.*, **185** (2017), 991-1015.

[16] Zahradnik, M. An alternate version of Pirogov-Sinai theory. *Comm. Math Phys.* **93** (1984) 559-581.