Computing the bridge length: the key ingredient in a continuous isometry classification of periodic point sets

Jonathan McManus[a]* and Vitaliy Kurlin[b]

[a]*Computer Science department and Materials Innovation Factory, University of Liverpool, Liverpool L69 3BX UK. E-mail: {j.d.mcmanus,vkurlin}@liverpool.ac.uk*

Abstract

The fundamental model of any periodic crystal is a periodic set of points at all atomic centres. Since crystal structures are determined in a rigid form, their strongest equivalence is rigid motion (composition of translations and rotations) or isometry (also including reflections). The recent classification of periodic point sets under rigid motion used a complete invariant isoset whose size essentially depends on the bridge length, defined as the minimum 'jump' that suffices to connect any points in the given set.

We propose a practical algorithm to compute the bridge length of any periodic point set given by a motif of points in a periodically translated unit cell. The algorithm has been tested on a large crystal dataset and is required for an efficient continuous classification of all periodic crystals. The exact computation of the bridge length is a key step to realising the inverse design of materials from new invariant values.

1. Introduction: practical motivations and the problem statement

All solid crystalline materials can be modelled at the atomic level as periodic sets of points (with the chemical attributes if desired) at all atomic centres, defined below.

Definition 1 (lattice, unit cell, motif, periodic point set)**.** *Any vectors* $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$ *that form a linear basis of* $\mathbb{R}^n$ *generate the* lattice $\Lambda = \{\sum_{i=1}^{n} c_i \boldsymbol{v_i} \mid c_i \in \mathbb{Z}\}$ *and the* unit cell $U = \{\sum_{i=1}^{n} t_i \boldsymbol{v_i} \mid 0 \le t_i < 1\}$. *A* motif *is any finite set of points* $M \subset U$*, which can represent centres of atoms in a real crystal. The* motif size $|M|$ *is the number of points in* M*. A* periodic point set $S = \Lambda + M = \{\boldsymbol{v} + p \mid \boldsymbol{v} \in \Lambda, p \in M\}$ *is a union of* $|M|$ *lattices whose origins are shifted to all points* p *of the motif* M*, see Fig. 1 (left).*

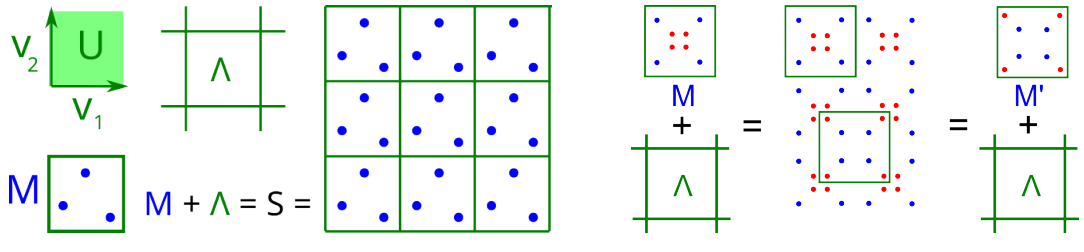


Fig. 1. **Left**: the orthonormal basis $\boldsymbol{v_1}, \boldsymbol{v_2}$ generates the green lattice Λ and the unit cell U containing the blue motif M of three points. The periodic point set $S = \Lambda + M$ is obtained by periodically repeating M along all vectors of Λ. **Right**: different motifs M, M' in the same cell generate periodic sets that differ by only translation.

Any unit cell U is a parallelepiped on basis vectors $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$. If we translate the unit cell U by all vectors $v \in \Lambda$, the resulting cells tile $\mathbb{R}^n$ without overlaps. Motif points represent atomic centres in a real crystal. The same lattice can be generated by infinitely many different bases that are all related under multiplication by $n \times n$ matrices with integer elements and determinant 1. Even if we fix a basis of $\mathbb{R}^n$ and hence a unit cell U, different motifs in U can define periodic point sets that differ only by Euclidean *isometry* defined as any distance-preserving transformation of $\mathbb{R}^n$.

Since crystal structures are determined in a rigid form, their slightly stronger equivalence is *rigid motion* defined as any orientation-preserving isometry without reflections or as a composition of translations and rotations. After many years of discussing definitions of a "crystal" (Brock, 2021), a *crystal structure* was recently defined in the

periodic case as a class of periodic sets under rigid motion (Anosova *et al.*, 2024).

Any such class consists of all (infinitely many) periodic point sets that are equivalent to each other under some rigid motions. However, almost any perturbation of atoms disturbs some inter-atomic distances and hence the isometry class with all cell-based descriptors such as symmetry groups. Even in dimension 1, for any integer $m > 0$ and threshold $\epsilon > 0$, the sequence $\mathbb{Z}$ with period 1 is pointwise ϵ-close to the sequence with the motif $M = \{0, 1+\epsilon, \dots, m+\epsilon\}$ and arbitrarily large period $m+1$.

This inherent discontinuity of all cell-based descriptors was resolved by Pointwise Distance Distributions (PDD) in (Widdowson *et al.*, 2022; Widdowson & Kurlin, 2022; Widdowson & Kurlin, 2021), which defined geographic-style coordinates on the Cambridge Structural Database (CSD) in (Widdowson & Kurlin, 2024). Though PDDs distinguish all periodic crystals in the CSD within minutes on a modest desktop, the only theoretically complete and continuous invariant descriptor that uniquely identifies any periodic point set under isometry in $\mathbb{R}^n$ is the *isoset* (Anosova & Kurlin, 2021), see (Kurlin, 2022) for complete and continuous invariants of 1-periodic sets in $\mathbb{R}^n$. The isoset invariant requires the bridge length whose definition is reminded below.

Definition 2 (bridge length $\beta(S)$)**.** *For any finite or periodic set of points $S \subset \mathbb{R}^n$, the* bridge length $\beta(S)$ *is the minimum distance such that any points $p, q \in S$ can be connected by a finite sequence of points $p = p_1, p_2, \dots, p_k = q$ in S, such that every Euclidean distance has the upper bound $|p_i - p_{i+1}| \le \beta(S)$ for all $i = 1, \dots, k-1$.*

Equivalently, the *bridge length* $\beta(S)$ is the minimum double radius such that the union of the closed balls of the radius $\frac{1}{2}\beta(S)$ around all points of S is connected. The lattice $\Lambda = \mathbb{Z}^3$ of all points with integer coordinates has $\beta(\Lambda) = 1$. If we add to $\mathbb{Z}^3$ all points whose all coordinates are half-integer, the resulting BCC (body-centred cubic) periodic point set has $\beta = \frac{\sqrt{3}}{2}$ equal to the half-diagonal of the unit cube in $\mathbb{R}^3$.

Problem 3. *Design an algorithm to compute the bridge length $\beta(S)$ in polynomial time of the motif size for any periodic point set S with a fixed unit cell in $\mathbb{R}^n$.*

The bridge length of a finite set can be computed via a Minimum Spanning Tree below, but the periodic case does not easily reduce to a finite one as shown in Fig. 2.

Definition 4 (Minimum Spanning Tree)**.** *For any finite set M of points in $\mathbb{R}^n$, a* Minimum Spanning Tree MST(M) *is a tree that has the vertex set M and a minimum total length of straight-line edges with lengths measured by Euclidean distance.*

MST(M) is uniquely defined if all distances between points of M are distinct. By Definition 2 the bridge length $\beta(M)$ equals the length of the longest edge of MST(M).

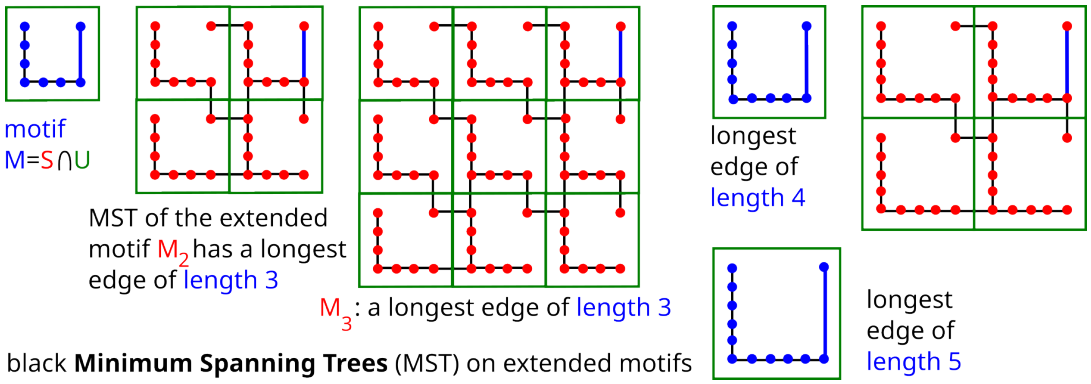


Fig. 2. All Minimum Spanning Trees on extended motifs of a periodic point set S have the longest edge (in blue) of length 3, which could be made arbitrarily long, relative to a preserved minimum inter-point distance of 1 and bridge length $\beta(S) = 2$ due to shorter edges from the top right point in every cell across a cell boundary.

For any periodic point set S with a unit cell U on a basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$ in $\mathbb{R}^n$, one can consider the extended motifs $M_k = S \cap U_k$, where the extended cell U_k is defined by the basis $k\boldsymbol{v_1}, \dots, k\boldsymbol{v_n}$ for any integer $k > 1$. The Minimum Spanning Trees provide the upper bounds $\beta(S) \le \beta(M_k)$ for $k > 1$, which can be unnecessarily high, see Fig. 2, so Problem 3 is much harder for periodic sets than for finite sets of points.

For any periodic point set $S \subset \mathbb{R}^n$, Lemma 3.7(a) in (Anosova *et al.*, 2022) proved the upper bound $\beta(S) \leq \min\{r(U), 2R(S)\}$ in terms of parameters below.

Definition 5 (parameters $r(U)$, $R(S)$, $a(U)$)**.** *Let $S \subset \mathbb{R}^n$ be periodic point set whose a unit cell U has a basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$. Set $r(U) = \max\{b, \frac{d}{2}\}$, where d is the length of the longest diagonal of U and $b = \max\limits_{i=1,\dots,n} |\boldsymbol{v_i}|$. The* covering radius *$R(S)$ is the smallest radius R such that the union of closed balls of the radius R around all $p \in S$ covers $\mathbb{R}^n$. The* height *is $h(U) = \mathrm{vol}(U) / \max\limits_{i=1,\dots,n} \mathrm{vol}(U_i)$, where U_i is the subcell of U spanned by all basis vectors except $\boldsymbol{v_i}$. The* aspect ratio *is $a(U) = r(U)/h(U)$.*

Main Theorem 6 below guarantees an exact computation of the bridge length $\beta(S)$ in a time that only quadratically depends on the motif size m of a periodic set S.

Theorem 6. *For any periodic point set $S \subset \mathbb{R}^n$ with a motif of m points in a unit cell U, the bridge length $\beta(S)$ can be computed in time $O(m^2 a(U)^n N)$, where N is the time complexity of the Smith Normal Form, $a(U)$ is the aspect ratio from Definition 5.*

As the time complexity is proportional to the aspect ratio $a(U)$ of a cell U, an initial reduction of U to a smaller cell will speed up the computation of the bridge length by minimising further cell extensions, namely *supercell_size* in Algorithm 16.

The Smith Normal Form and its time complexity are reminded in sections 3 and 4, respectively. Section 5 discusses computations on experimental and simulated crystals.

2. Auxiliary concepts of graph theory for the bridge length algorithm

This section introduces a few auxiliary concepts to describe the exact algorithm for the bridge length in section 3 and to prove main Theorem 6 at the end of section 4.

Definition 7 ($G \subset \mathbb{R}^n$)**.** *Let $S \subset \mathbb{R}^n$ be a periodic point set with a lattice Λ. A* periodic Euclidean graph *$G \subset \mathbb{R}^n$ is an infinite graph with the vertex set S and straight-line*

edges such that the translation by any vector $\boldsymbol{v} \in \Lambda$ *defines an* automorphism *of* G*, which is a bijection* $S \to S$ *that also induces a bijection on the edges of* G*, see Fig. 3.*

If straight-line edges meet at interior points, they are not considered vertices of G.

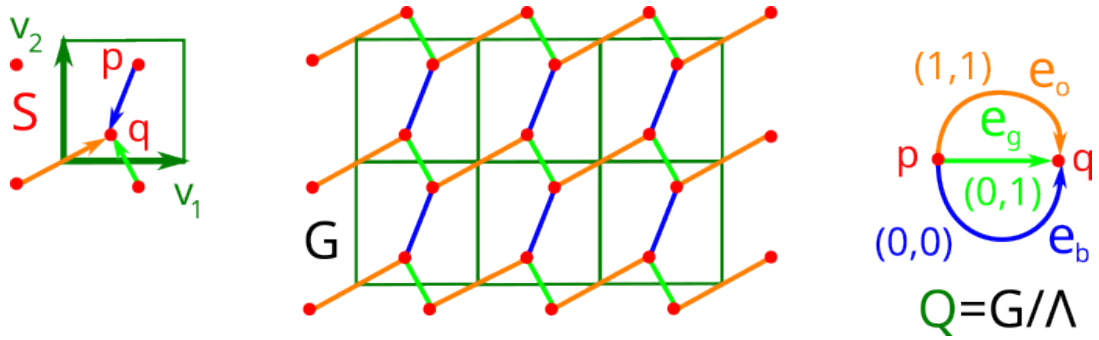


Fig. 3. **Left**: the periodic point set S with the basis vectors $\boldsymbol{v_1} = (5, 0)$, $\boldsymbol{v_2} = (0, 5)$ and motif points $p = (2, 1)$, $q = (3, 4)$. **Middle**: the periodic Euclidean graph $G \subset \mathbb{R}^2$ with three types of straight-line edges: green, blue, orange of lengths $\sqrt{5}, \sqrt{10}, \sqrt{20}$, respectively. **Right**: the labelled quotient graph Q has directed edges e_g, e_b, e_o with translational vectors indicating integer shifts of cells, see Definitions 7, 8, 9.

Fig. 3 shows a connected periodic graph G but G can also be disconnected. For example, let S be the square lattice $\mathbb{Z}^2$, then the graph G consisting of all horizontal edges connecting the points (m, n) and $(m + 1, n)$ for $m, n \in \mathbb{Z}$ is periodic but not connected. If we add to G all vertical edges connecting (m, n) and $(m, n + 1)$ for $m, n \in \mathbb{Z}$, the resulting infinite square grid is a connected periodic graph on $\mathbb{Z}^2$.

Definition 8 (quotient graph)**.** *Let* G *be a periodic graph on a periodic point set* S *with a lattice* Λ *in* $\mathbb{R}^n$*. Two points of* S *(also vertices or edges of* G*) are called* Λ*-equivalent if they are related by a translation along a vector* $\boldsymbol{v} \in \Lambda$*. The* quotient graph G/Λ *is an abstract undirected graph obtained as the quotient of* G *under the* Λ*-equivalence. Then* G *is called a* lifted graph *of* G/Λ*. Any vertex of* G/Λ *is a* Λ*-equivalence class* $p + \Lambda$ *represented by a point* $p \in S$*. Any edge* e *of the quotient graph*

G/Λ is a Λ-equivalence class $[p,q]+\Lambda$ represented by a straight-line edge $[p,q]$ of G. We define the length of any edge e in G/Λ as the Euclidean distance $|p-q|$.

The quotient graph G/Λ can have multiple edges between the same pair of vertices as shown in Fig. 3, which all can be distinguished by the labels defined below.

Definition 9 (labelled quotient graph)**.** *Let $S \subset \mathbb{R}^n$ be a periodic point set with a lattice Λ defined by a basis $\boldsymbol{v_1},\dots,\boldsymbol{v_n}$. Let G be a periodic graph on S. For an edge e of the quotient graph G/Λ, choose any of two directions and a representative edge $[p,q]$ in the lifted graph G. Let $U(p), U(q)$ be the unit cells containing p, q, respectively. There is a unique vector $\boldsymbol{v} = \sum_{i=1}^{n} c_i \boldsymbol{v_i} \in \Lambda$ such that $U(q) = U(p) + v$ and $c_i \in \mathbb{Z}$.*

A labelled quotient graph (LQG) *is G/Λ whose every edge e has a direction (say, from the Λ-equivalence class of p to Λ-equivalence class of q) and the* translational vector $\boldsymbol{v}(e) = (c_1,\dots,c_n) \in \mathbb{Z}^n$*, see Fig. 3. Changing the direction of e multiplies each coordinate of $\boldsymbol{v}(e)$ by (-1). An equivalence of LQGs is a composition of a graph isomorphism and changes in edge directions that match all translational vectors.*

Translational vectors $\boldsymbol{v}(e)$ are also called *voltages* if G/Λ is considered a *voltage graph* or a *gain graph* in topological graph theory. In crystallography, labelled quotient graphs have been studied by many authors. Section 6 in (Chung *et al.*, 1984) generated 3-periodic nets by considering LQGs whose translational vectors have entries from $\{-1,0,1\}$. Section 2 in (Cohen & Megiddo, 1990) described an algorithm to find connected components of a fixed periodic graph in terms of its LQG. Proposition 5.1 in (Eon, 2011) showed how to reconstruct a periodic graph up to translations from LQG and a lattice basis, which we also prove in Lemma 10 in our notations for completeness. Section 3 in (Eon, 2016*a*) described surgeries on building units of LQGs. Theorem 6.1 in (Eon, 2016*b*) characterised 3-connected minimal periodic graphs (with a slightly different definition of 'minimal'). (McColm, 2024) initiated a search for systematic periodic graphs realisable by real crystal nets, see also (Edelsbrunner & Heiss, 2024).

The labelled quotient graph G/Λ in Fig. 3 has two vertices p, q. If we orient the three edges of $Q = G/\Lambda$ from p to q, the translational vector $(0, 0)$ of the blue edge e_b in G/Λ means that the corresponding straight-line blue edge in the lifted graph $G \subset \mathbb{R}^2$ connects points of S within the same unit cell U with the basis $\boldsymbol{v_1}, \boldsymbol{v_2}$. The orange edge with the translational vector $(1, 1)$ means that each of its infinitely many liftings in $G \subset \mathbb{R}^2$ joins a point in a cell U to another point in the cell $U + \boldsymbol{v_1} + \boldsymbol{v_2}$.

Lemma 10 (lifting)**.** *Let G be a periodic Euclidean graph on a periodic point set S with a motif M in a unit cell U defined by a basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$ in $\mathbb{R}^n$. Let Q be a labelled quotient graph of G. Then $G \subset \mathbb{R}^n$ can be reconstructed from Q, the basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$, and a bijection between all vertices of Q and all points of the motif $M \subset U$.*

Proof. The basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$ is needed to define a unit cell U with the given points of M, which are in 1-1 correspondence with all vertices of Q. The periodic point set S, which is the vertex set of the periodic graph G, is obtained from M by translations along the vectors $\sum_{i=1}^{n} c_i \boldsymbol{v_i}$ for all $c_i \in \mathbb{Z}$. By Definitions 8 and 9, every edge e of the labelled quotient graph Q has a translational vector $\boldsymbol{v}(e) = (c_1, \dots, c_n)$ and is a Λ-equivalence class $[p, q] + \Lambda$ for some $p, q \in S$ whose unit cells $U(p), U(q)$ are related by the translation along $\sum_{i=1}^{n} c_i \boldsymbol{v_i}$. Then we can lift the edge e to the periodically translated straight-line edges $[p + \boldsymbol{v}, q + \boldsymbol{v} + \sum_{i=1}^{n} c_i \boldsymbol{v_i}]$ in the periodic graph G for all $\boldsymbol{v} \in \Lambda$. $\square$

Definition 11 (path/cycle sum)**.** *For a path (sequence of consecutive edges) in a labelled quotient graph Q, we make all directions of edges consistent in the sequence and define the* path sum *in $\mathbb{Z}^n$ as the sum of the resulting translational vectors along the path. If the path is a closed cycle, the path sum is called the* cycle sum*.*

In the language of *voltage graphs*, a path sum may equivalently be referred to as the *net voltage* over the path. In the labelled quotient graph in Fig. 3, the upper cycle

consisting of the directed orange edge (from p to q) and the inverted green edge (from q to p) has the cycle sum $(1,1)+(0,-1)=(1,0)$. This cycle sum means that a lifting of the cycle to the periodic graph G in $\mathbb{R}^2$ produces a polygonal path connecting a point to its translate by the vector $\boldsymbol{v_1}=(1,0)$ in the next cell to the right.

Definition 12 (minimal tree MST(S/Λ))**.** *For a periodic point set $S\subset\mathbb{R}^n$ with a lattice Λ, a* minimal tree *is a Minimum Spanning Tree* MST(S/Λ) *(Definition 4) on the set S/Λ of Λ-equivalence classes of points, where the distance between any classes in S/Λ is the minimum Euclidean distance between their representatives in the set S.*

In Fig. 3, a minimal tree MST(S/Λ) consists of one shortest green edge in G/Λ.

3. Algorithm for the bridge length of a periodic point set

This section will describe main Algorithm 16 for solving Problem 3, which will call auxiliary Algorithm 13 several times. Algorithm 13 starts from a conventional representation of a periodic set $S\subset\mathbb{R}^n$ with a motif M of points given by coordinates in a basis $\boldsymbol{v_1},\dots,\boldsymbol{v_n}$ of a lattice Λ as in a Crystallographic Information File (CIF).

At every call, Algorithm 13 returns the next shortest edge e between points of S in increasing order of length. Although S is a set of points rather than a graph, we will use the term 'edge', because e can be considered an edge from a complete graph with the vertex set S and with the 'next shortest edge' being up to Λ-equivalence.

Any edge e between points of S will be represented by an ordered pair of points $p,q\in M$ and a translational vector $(c_1,\dots,c_n)\in\mathbb{Z}^n$ so that the actual straight-line edge in the lifted periodic graph $G\subset\mathbb{R}^n$ is from p to the point $q+\sum\limits_{i=1}^{n}c_i\boldsymbol{v_i}$. For convenience, we record the Euclidean distance $d=|q-p+\sum\limits_{i=1}^{n}c_i\boldsymbol{v_i}|$ between these endpoints. Then Algorithm 13 outputs any edge e as a tuple $(p,q;c_1,\dots,c_n;d)$.

Algorithm 13 maintains the list of already found edges in increasing order of length.

If the next required edge e is already in the list, Algorithm 13 simply returns e. This shortcut is implemented in Python with the keyword 'Yield', see the documentation at https://docs.python.org/3/glossary.html#term-generator-iterator. Rather than starting from line 1, every time when Algorithm 13 is called, each call 'Yield e' returns an edge e, then temporarily suspends processing, remembering the location execution state including all local variables. When 'Yield e' is called again, Algorithm 13 picks up where it left off in contrast to functions that start fresh on every invocation.

If the next edge e is not yet found, Algorithm 13 adds more points from a shell of unit cells surrounding the previously considered cells. This *shell* contains the extended motif M_k without the smaller motif M_{k-1} for $k > 1$, see Fig. 2. For any new point p, it suffices to consider only edges to points $q \in M \subset U$ because any edge e can be periodically translated by $\boldsymbol{v} \in \Lambda$ so that one of the endpoints of e belongs to U. In Algorithm 13, the *Chebyshev distance* D_∞ in line 3 is the maximum absolute difference of corresponding coordinates, while d in line 7 is the usual *Euclidean distance*.

Algorithm 13. *Input: a basis* $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$ *defining a unit cell* U*, a motif* $M \subset U$*.*
next_edge ***runs only until the next*** **Yield*****, and outputs the yielded edge.***

```
1:  supercell_size=0, current_batch=[], next_batch=[], next_batch_min_len=infinity
2:  while True do
3:      for transl_vector in ℤ^n s.t. D_∞(0⃗, transl_vector) = supercell_size do
4:          for source in the motif M do
5:              for dest in the motif M do
6:                  true_dest = dest + basis · transl_vector
7:                  length = d(source, true_dest)
8:                  next_batch.append((length, source, dest, transl_vector))
9:                  next_batch_min_len = minimum(length,next_batch_min_len)
10:             end for
```

11: ***end for***

12: ***end for***

13: ***while*** *current_batch is not empty* ***do***

14: *next = minimum(current_batch)*

15: ***if*** *next* $\geq$ *next_batch_min_len* ***then*** *Break*

16: ***end if***

17: *current_batch.remove(next)*

18: *Yield(next)*

19: ***end while***

20: *current_batch = concatenate(current_batch, next_batch)*

21: *next_batch=[]*

22: *supercell_size = supercell_size + 1*

23: ***end while***

There is a faster way of checking a condition equivalent to *next_batch_min_len* by using the cell geometry. Then in the vast majority of cases the algorithm can stop at a supercell one size smaller, which dramatically speeds up the calculation. This calculation is described in Remark 14. However, due to the possibility of that not being the case (upon which the algorithm would just default to the same supercell size), we will keep this simpler idea and use it for the time complexity calculations.

Remark 14 (a faster way to compute *next_batch_min_len* in Algorithm 13)**.** *For a unit cell with a basis* $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$*, let* $\boldsymbol{a}_i$ *and* $\boldsymbol{b}_i$ *be the shortest vectors parallel and antiparallel to* $\boldsymbol{v_i}$ *from any point of a motif* $M \subset U$ *to the opposite boundary faces of the unit cell* U*. Then the faster alternative for next_batch_min_len is*

$$\min_{i=1,\ldots,n} (|\boldsymbol{a_i}| + |\boldsymbol{b_i}| + supercell\_size * |\boldsymbol{v_i}|).$$

As all the vector lengths $|\boldsymbol{a_i}|, |\boldsymbol{b_i}|$*,* $i = 1, \ldots, n$ *can be pre-computed, we get a massive*

improvement over the calculation of next_batch_min_len in Algorithm 13.

Algorithm 16 will be building a labelled quotient graph Q by adding (or ignoring) edges found by Algorithm 13 and monitoring the connectivity of the growing lifted graph G whose quotient G/Λ is Q. For a basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$ of a unit cell U of the lattice Λ of S, the edge e between points p and $q + \sum_{i=1}^{n} c_i \boldsymbol{v_i} \in S$ is added to Q as the edge between the Λ-equivalence classes of p and q, with the translational vector $\boldsymbol{v}(e) = (c_1, \dots, c_n) \in \mathbb{Z}^n$. As soon as G becomes connected, the length of the last added edge is the bridge length $\beta(S)$, which will be proved in Theorem 26 later.

In comparison with a Minimum Spanning Tree of a finite set of points, verifying the connectivity of the lifted periodic graph requires a much more complicated check that translational vectors with integer coordinates form a basis in $\mathbb{Z}^n$ (not $\mathbb{R}^n$), which can include more than n vectors. Fig. 4 shows a basis of $\mathbb{Z}^2$ consisting 3 vectors, where no vector can be dropped without losing the connectivity of all integer points in $\mathbb{Z}^2$.

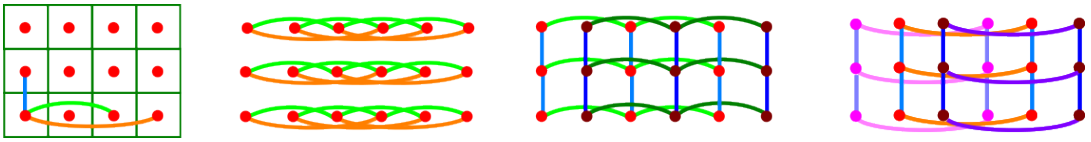


Fig. 4. **Left**: the 3 vectors $\boldsymbol{v_1} = (0,1)$, $\boldsymbol{v_2} = (2,0)$, $\boldsymbol{v_3} = (3,0)$ form a basis of $\mathbb{Z}^2$. **Other images**: none of the 3 pairs $(\boldsymbol{v_2}, \boldsymbol{v_3})$, $(\boldsymbol{v_1}, \boldsymbol{v_2})$, $(\boldsymbol{v_1}, \boldsymbol{v_3})$ form a basis (insufficient for full connectedness) of $\mathbb{Z}^2$. Some straight edges are shown curved for better visibility.

Algorithm 16 will use the Smith Normal Form (SNF) of a matrix of vectors $(c_1, \dots, c_n)$ in $\mathbb{Z}^n$, see p. 26 in (Newman, 1972), (Cohn, 1985), and chapter 3.6 in (Van der Waerden, 2003) for finitely generated modules over a Principal Ideal Domain (PID).

Definition 15 (Smith Normal Form and invariant factors)**.** *For integers $m, n \geq 1$, let A be a non-zero $n \times m$ matrix over a Principal Ideal Domain P, for example, $P = \mathbb{Z}$. Then there exist invertible $n \times n$ and $m \times m$-matrices L,R, respectively, with*

coefficients in P, such that the product LAR is an $n \times m$ matrix whose only non-zero entries are diagonal elements a_i such that a_i divides a_{i+1} for $i = 1, \ldots, j-1$, and $a_i = 0$ for $i = j, \ldots, n$ for some $1 < j \leq n$. This diagonal matrix LAR is the Smith Normal Form SNF(A). *The diagonal elements a_i are called the* invariant factors *of A.*

Let 1 denote the unit element of a Principal Ideal Domain P. If $P = \mathbb{Z}$, then 1 is the usual integer 1. The simplest SNF has all invariant factors equal to 1, which happens if and only if the last factor $a_n = 1$ because all previous factors a_i divide a_n.

Algorithm 16 (Finding the bridge length $\beta(S)$ of any periodic point set $S \subset \mathbb{R}^n$)**.**
Initialisation*. A labelled quotient graph Q and a forest $F \subset Q$ initially consist of m isolated vertices, each representing a Λ-equivalence class of a point of the motif of S. We will build a* translational matrix *A with columns in $\mathbb{Z}^n$, which is initially empty.*

Loop stage*. Consider the next edge $e = next\_edge()$ found by Algorithm 13.*

Case 1*. If adding the edge e to the current forest F would not form a closed cycle (ignoring all edge directions), then add e to F and Q as an edge with an arbitrarily chosen direction and corresponding translational vector $\boldsymbol{v}(e)$ found by Algorithm 13.*

Case 2*. If adding the edge e to F does form a cycle, find its cycle sum $c \in \mathbb{Z}^n$ from Definition 11. If c is not $0 \in \mathbb{Z}^n$ and cannot be expressed as an integer linear combination of the columns from the current translational matrix A, then add e to Q as in Case 1 (but not to the forest F) and add the vector c as a new column to A.*

Termination*. Stop if both conditions below hold, otherwise continue the loop.*

(1) the labelled quotient graph Q (hence the forest F) becomes connected; and

(2) the translational matrix A (whose columns are cycle sums of cycles created by adding edges) has n invariant factors equal to 1, see Definition 15.

The necessity of termination condition 1 in Algorithm 16 means that if the lifted periodic graph G is connected then so is its quotient $Q = G/\Lambda$. The inverse implication

(sufficiency) may not hold. For example, in Fig. 3, the minimal tree MST(S/Λ) is a single green edge e_g, whose preimage under the quotient map $G \to G/\Lambda$ is the disconnected set of all green straight-line edges in the periodic graph $G \subset \mathbb{R}^2$.

Example 17 (running Algorithm 16 on the periodic point set S in Fig. 3)**.** *The first addition to the quotient graph Q and forest F, which initially had two isolated vertices p, q, is the shortest green edge e_g from p to q (case 1 in the loop stage) with the translational vector $c(e_g) = (0,1) \in \mathbb{Z}^2$. The translational matrix A remains empty.*

Adding the next (by length) blue edge e_b with $c(e_b) = (0,0)$ to $F = \{e_g\}$ creates a cycle with the cycle sum $c = c(e_g) - c(e_b) = (0,1)$. According to case 2 in the loop stage, the quotient graph Q becomes the cycle of two edges $e_g \cup e_b$ but the forest remains $F = \{e_g\}$. The translational matrix A becomes one column $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and does not yet have two invariant factors 1. The 2nd termination condition is not yet satisfied, and the current lifted graph consisting of all green and blue segments is still disconnected.

Adding the orange edge e_o with $c(e_o) = (1,1)$ to F creates another cycle with the cycle sum $c' = c(e_g) - c(e_o) = (-1,0)$. The quotient graph $Q = e_g \cup e_b \cup e_o$ is now full but $F = \{e_g\}$ is still one edge. The matrix A becomes $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ whose SNF $= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ *shows that A has 2 invariant factors equal to 1. Both termination conditions hold and the lifted periodic graph $G \subset \mathbb{R}^2$ of all green, blue, and orange edges is connected. The bridge length $\beta(S) = 2\sqrt{5}$ equals the length of the last (orange) edge as expected.*

4. Correctness and time complexity of the bridge length algorithm

This section proves the correctness of Algorithm 16 in Theorem 26 about the bridge length and main Theorem 6 about its time complexity. Lemmas 20-21 will prove the necessity of termination condition 2 in Algorithm 16. Both conditions 1 and 2 will guarantee the connectedness of the lifted periodic graph G due to Lemma 23.

Lemma 18 is a partial case of the splitting lemma on page 147 in (Hatcher, 2002).

Lemma 18 (splitting)**.** *A short sequence of linear maps* $0 \to \mathbb{Z}^{m-n} \xrightarrow{f} \mathbb{Z}^m \xrightarrow{g} \mathbb{Z}^n \to 0$ *is called* exact *if the image of each map coincides with the* kernel *(subspace mapping to 0) of the next map, i.e.* $\mathrm{Ker}(f) = 0$, $\mathrm{Im}(f) = \mathrm{Ker}(g)$, $\mathrm{Im}(g) = \mathbb{Z}^n$. *If there is a map* $h : \mathbb{Z}^n \to \mathbb{Z}^m$, *such that* $g \circ h$ *is the identity on* $\mathbb{Z}^n$, *then* $\mathbb{Z}^m \cong f(\mathbb{Z}^{m-n}) \oplus h(\mathbb{Z}^n)$, *where* $f(\mathbb{Z}^{m-n})$ *and* $h(\mathbb{Z}^n)$ *are linearly independent subspaces of* $\mathbb{Z}^m$ *for* $m \geq n$.

Example 19 (finding a Smith Normal Form)**.** *In the notations of Lemma 18, Fig. 4 defines the map* $g : \mathbb{Z}^3 \to \mathbb{Z}^2$ *given by the matrix* $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 3 \end{pmatrix}$ *whose 3 columns generate* $\mathbb{Z}^2$. *Then* $\mathrm{Ker}(g) \subset \mathbb{Z}^3$ *consists of all vectors* $f(k) = k \begin{bmatrix} 0 \\ 3 \\ -2 \end{bmatrix}$ *for* $k \in \mathbb{Z}$, *which defines* $f : \mathbb{Z} \to \mathbb{Z}^3$ *with* $\mathrm{Ker}(f) = 0$ *and* $\mathrm{Im}(f) = \mathrm{Ker}(g)$ *as required in Lemma 18. Since* $g : \mathbb{Z}^3 \to \mathbb{Z}^2$ *is surjective, we can find a map* $h : \mathbb{Z}^2 \to \mathbb{Z}^3$ *satisfying* $g \circ h = \mathrm{id}$, *e.g.* h *can be given by* $M = \begin{pmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix}$, *then* $AM = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, *denoted by* I_2. *After extending the* 3×2 *matrix* M *by the extra column with a basis vector of* $\mathrm{Im}(f)$, *we get the matrix* $R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 3 \\ 0 & 1 & -2 \end{pmatrix}$ *such that* $AR = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$.

Lemma 18 implies that the constituent blocks of R *are linearly independent to each other; all columns of* R *are linearly independent, and* R *is invertible. Hence,* I_2AR *is a Smith Normal Form of* A *with* $n = 2$ *invariant factors equal to 1 by Definition 15.*

Lemma 20 (matrix generating $\mathbb{Z}^n \Leftrightarrow n$ invariant factors equal to 1)**.** *The columns of any* $n \times m$ *matrix* A *generate* $\mathbb{Z}^n$ *if and only if* A *has* n *invariant factors equal to 1.*

Proof. Let the m columns of A generate $\mathbb{Z}^n$. Then A defines the surjection $g : \mathbb{Z}^m \to \mathbb{Z}^n$ whose $\mathrm{Ker}(g)$ can be obtained as the image of a map $f : \mathbb{Z}^{m-n} \to \mathbb{Z}^m$. So $\mathrm{Ker}(g)$ is generated by $f(\boldsymbol{e_1}), \ldots, f(\boldsymbol{e_{m-n}})$, where $\boldsymbol{e_1}, \ldots, \boldsymbol{e_{m-n}}$ form an orthonormal basis of $\mathbb{Z}^{m-n}$. Since $g : \mathbb{Z}^m \to \mathbb{Z}^n$ is surjective, orthonormal basis vectors $\boldsymbol{u_1}, \ldots, \boldsymbol{u_n}$ of $\mathbb{Z}^n$ are images $g(\boldsymbol{v_1}), \ldots, g(\boldsymbol{v_n})$, respectively, of some vectors $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n} \in \mathbb{Z}^m$. We can define the linear map $h : \mathbb{Z}^n \to \mathbb{Z}^m$, $h(\boldsymbol{u_i}) = \boldsymbol{v_i}$ for $i = 1, \ldots, n$, so that $g \circ h = \mathrm{id}$ on $\mathbb{Z}^n$.

Then h has the $m \times n$ matrix M such that $AM = I_n$, where I_n is the $n \times n$ identity matrix. Extending M by the $m - n$ columns $f(\boldsymbol{e_1}), \ldots, f(\boldsymbol{e_{m-n}})$ gives the invertible $m \times m$ matrix R such that AR equals the $n \times m$ matrix obtained by extending I_n with $m - n$ zero columns. Again, R is an invertible matrix over $\mathbb{Z}$, so $I_m AR = AR$ is the Smith Normal Form of A with all invariant factors equal to 1 by Definition 15.

Conversely, let the Smith Normal Form SNF $= LAR$ of the matrix A in Definition 15 have all invariant factors equal to 1. Then the n columns of the $n \times m$ matrix AR and hence the n columns of A form a basis of $\mathbb{Z}^n$. Indeed, transforming the m columns of AR by the invertible $n \times n$ matrix L gives the standard orthonormal basis of $\mathbb{Z}^n$. □

Lemma 21 (connected periodic graph $G \subset \mathbb{R}^n \Rightarrow n$ invariant factors equal 1)**.** *In Algorithm 16, if the lifted periodic graph $G \subset \mathbb{R}^n$ becomes connected, then the translational matrix A has n invariant factors equal to 1.*

Proof. By Lemma 20 it suffices to show that any vector $\boldsymbol{v} \in \mathbb{Z}^n$ is an integer linear combination of columns of A. Choose any point $p \in S$. Then the points p and $p + \boldsymbol{v}$ are connected in the lifted periodic graph $G \subset \mathbb{R}^n$ by a polygonal path of straight-line edges. Under $G \to G/\Lambda$, this path projects to a closed cycle C at the vertex (Λ-equivalence class) $p + \Lambda$ in the labelled quotient graph $Q = G/\Lambda$.

Let the cycle C pass through edges $e_1, \ldots, e_k$ (with integer multiplicities) in the complement $Q - F$ of the forest F in the quotient graph Q. These edges were added only to Q in case 2 of the loop stage. When we tried to add every edge e_j to F, the edge e_j created a cycle C_j whose cycle sum appeared as a column in the translational matrix A (if this cycle sum was not yet an integer combination of the previous columns). Then the vector $\boldsymbol{v}$ equals the sum of the cycle sums of all the cycles C_j for $j = 1, \ldots, k$, which is an integer combination of the columns of A as required. □

Lemma 22 (connected quotient graph $G/\Lambda \Rightarrow \exists$ a tree of representatives $T \subset G$)**.** *If*

a labelled quotient graph $Q = G/\Lambda$ is connected, its lifted graph $G \subset \mathbb{R}^n$ on a periodic point set S with a motif of m points and a lattice Λ includes a straight-line tree of representatives $T \subset G$ *with m vertices that are not Λ-equivalent to each other.*

Proof. Since Q is connected, we can choose a spanning tree $F \subset Q$ on the m vertices of Q. A required tree $T \subset G$ will be a connected union of straight-line edges of G that map 1-1 to all edges of F under the quotient $G \to Q$. Start from any point $p \in S$ and take any edge e at the vertex (Λ-equivalence class) $p + \Lambda$ of $F \subset Q$. The preimage of e under $G \to Q$ contains a unique straight-line edge $[p, q] \subset G$, which we add to T. After adding to T all edges at p that project to all edges of F at the vertex $p + \Lambda$, choose another point $p' \in T$ such that the vertex $p' + \Lambda$ has an edge of F not yet covered by T under $G \to Q$. We continue adding edges to T by using their projections in $F \subset Q$ until we get a tree $T \subset G$ that spans m points of S that are not Λ-equivalent. The final T has no cycle, else this cycle projects under $G \to Q$ to a cycle in a forest F. $\square$

Lemma 23 (termination conditions in Algorithm 16 $\Rightarrow$ connected graph $G \subset \mathbb{R}^n$)**.** *Let Q be a labelled quotient graph with a translational matrix A and a lifted graph G on a periodic point set $S \subset \mathbb{R}^n$ with a lattice Λ. If Q is connected and the matrix A has n invariant factors equal to 1, then the lifted periodic graph $G \subset \mathbb{R}^n$ is connected.*

Proof. For any points $p, q \in S$, we will find a path of straight-line edges in G as follows. By Lemma 22 the connectedness of the quotient graph $Q = G/\Lambda$ guarantees the existence of a tree $T \subset G$ whose vertices represent all Λ-equivalences classes of points of S. Let p', q' be the vertices of T that are Λ-equivalent to p, q, respectively.

Since p', q' are connected by a path in T, it suffices to find a path from p to its Λ-translate $p' = p + \boldsymbol{v}$ (then similarly from q to q') in the graph G for any $\boldsymbol{v} \in \Lambda$. By Lemma 20 the columns of A form a basis of $\mathbb{Z}^n$, so $\boldsymbol{v}$ is an integer combination of these columns. It suffices to find a path in G by assuming that $\boldsymbol{v}$ is one column of A

because a path for any sum $\sum_i \boldsymbol{v}_i$ can be obtained by concatenating paths for $\boldsymbol{v}_i$. A column $\boldsymbol{v}$ can appear in A only in case 2 of the loop stage in Algorithm 16 as a cycle sum of a cycle $C \subset Q$ that was created by trying to add an edge e from Algorithm 13 to a forest $F \subset Q$. If we order all edges of C from the vertex $p + \Lambda$ as $e_1, \dots, e_k$, the sum of their translation vectors equals $\boldsymbol{v}$. We build a path from p to $p + \boldsymbol{v}$ in G by finding a unique edge $[p, p_1] \subset G$ that projects to e_1, then a unique edge $[p_1, p_2] \subset G$ that projects to e_2 and so on until we cover all $e_1, \dots, e_k$ and arrive at $p + \boldsymbol{v}$. □

Remark 24. *The paper (Onus & Robins, 2022) discusses connected components of a periodic graph K in terms of homology, namely Theorem 1(1) proves that $H_0(K)$ has a basis of $\sum_{i=1}^{N} [\mathbb{Z}^d : W_{Q_i}]$ elements, see details in their section 3.1, but without describing an algorithm for finding such a basis. Our results complement their approach by proving the time complexity for checking the connectivity of a dynamic periodic Euclidean graph in Theorem 6 whilst keeping track of its connected components.*

Lemma 25 (ignored edges)**.** *Let an edge e be a Λ-equivalence class of a straight-line edge $[p, q] + \Lambda$ in a lifted periodic graph G for some points $p, q \in S$. If Algorithm 16 does not add the edge e to a labelled quotient graph Q, then the points p, q are already connected by a path in the graph $G \subset \mathbb{R}^n$ lifted from Q by Lemma 10.*

Proof. The loop stage in Algorithm 16 ignores an edge e in the cases below.

Case 1. The edge e forms a cycle in Q whose cycle sum is the zero vector in $\mathbb{Z}^n$.

Case 2. The edge e forms a cycle whose cycle sum equals an integer linear combination of pre-existing *cycle sums* from the translational set B.

In both cases, we have either one cycle (in case 1) containing e, whose cycle sum is $0 \in \mathbb{Z}^n$, or several cycles (in case 2), one (up to multiplicity) of which contains e, whose total sum of translational vectors is $0 \in \mathbb{Z}^n$. By Definition 9 each edge of Q involved in this zero sum can be lifted to a straight-line edge in the graph $G \subset \mathbb{R}^n$.

If we start from the given point $p \in S$, a cycle in Q and its sum 0 of translational vectors guarantees that the sequence of the lifted edges in G finishes at the same point p and hence forms a cycle C. This cycle C has the edge $[p, q]$ whose exclusion keeps the points $p, q \in S$ connected by the path in C that is complementary to $[p, q]$. □

Theorem 26. *Algorithm 16 finds the bridge length $\beta(S)$ from Definition 2 for any periodic point set $S \subset \mathbb{R}^n$ with a motif M of points given in a basis $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$.*

Proof. Within Algorithm 16, let d be the length of the last added edge e after which both termination conditions finally hold. By Lemma 25 all ignored edges do not create extra connections in the graph G. By Lemmas 21 and 22 the graph G obtained before adding the last edge e is disconnected. Lemma 23 guarantees that, when e is added, the graph G becomes connected. Because Algortihm 13 yields edges in increasing order, e is the shortest edge that could have this property, so the bridge length is $\beta(S) = d$. □

Theorem 6 has a rough upper bound assuming that the Smith Normal Form $\mathrm{SNF}(A)$ of an integer $n \times m$ matrix A is re-computed for every iteration in time $O(N)$. This time was estimated in (Giesbrecht, 1995) as $O^{\sim}(n^{\omega-1}m \cdot M(n \log ||A||))$, where $||A|| = \max_{i,j} |A_{ij}|$, $M(t)$ bounds the cost of multiplying two t-bit integers, and $\omega \leq 2.372$ is the exponent for matrix multiplication: two $n \times n$ matrices can be multiplied in time $O(n^{\omega})$, see (Williams *et al.*, 2024). The "soft-Oh" simplifies the complexity up to logarithmic factors, so $f = O^{\sim}(G)$ if and only if $f = O(g \log^c g)$ for a constant $c > 0$.

To speed up Algorithm 16 in practice, the Smith Normal Form can be updated at every iteration instead of recomputing from scratch, see details in appendix A.2.

***Proof of Theorem** 6.* Algorithm 16 solves Problem 3 by Theorem 26. It remains to show that the time complexity of Algorithm 16 is $O(m^2 a(U)^n N)$. Algorithm 16 has the initialisation of a constant time $O(1)$ and the loop stage. We will multiply an upper bound for the number of loops by the time complexity of each loop.

One loop in Algorithm 16 contains at most the following checks.

- *(Cycle)* Does adding an edge e to a forest F create a cycle?
- *(Combination)* Is the cycle sum an integer combination of previous cycle sums?
- *(Termination)* After appending a cycle sum c to the translational matrix A and calculating SNF(A), does A have n invariant factors equal to 1?

The condition *Cycle* is checked traditionally by a depth-first search $O(m), see$ (Sedgewick, 1983). The condition *Combination* is equivalent to 'Has SNF(A) changed?', and *Termination* is equivalent to 'Is the product of invariant factors of A equal to 1?'. So both conditions can be jointly checked in time $O(N)$ needed to compute SNF(A).

The time complexity of SNF(A) dominates all other steps in Algorithm 16, so we will use $O(N)$ to represent the complexity of a single loop iteration of Algorithm 16.

Every loop iteration calls Algorithm 13. If we consider all calls to Algorithm 13 as running sequentially, then the main loop will run at most $a(U)+1$ times, where $a(U)$ is the aspect ratio from Definition 5. Each loop runs through the unit cells that are '*supercell_size*' away from the central cell U_1. By the end, we will have run through and *yielded* $(a(U)+1)^n$ unit cells. For each unit cell U_i, we find all distances between the m points in U_i and m points in the central cell. The required time is $O(m^2)$ for two cells and hence $O(m^2 a(U)^n)$ for all cells. Algorithm 16 does not actually run for every edge found by Algorithm 13 but we assume this for simplicity. The worst-case complexity for the naive implementation of Algorithm 16 is $O(m^2 a(U)^n N)$. □

5. Experiments on real and simulated crystals, and a discussion

This section discusses experiments computing the exact bridge length $\beta(S)$ for 5679 simulated and 5 real nanoporous crystals in Fig. 5 reported in Nature paper (Pulido *et al.*, 2017). Table 1 contains the bridge lengths computed by Algorithm 16 on the

crystals from Fig. 5 given by their codes in the Cambridge Structural Database (CSD). The names of T2 polymorphs refer to the crystalline forms $\alpha, \beta, \gamma, \delta, \epsilon$ based on the same molecule T2. The crystal IDs starting from 6-letter codes in the first column of Table 1 refer to the Cambridge Structural Database (Taylor & Wood, 2019).

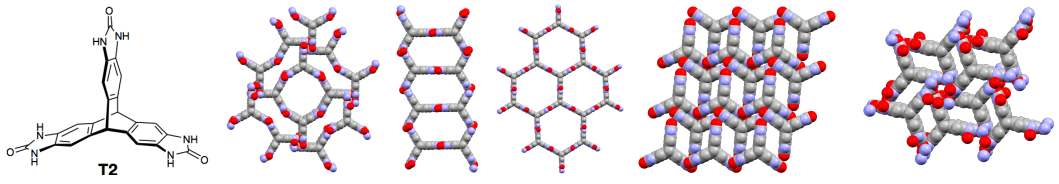


Fig. 5. T2 molecule and 5 crystals synthesized from T2. The first four T2-α, T2-β, T2-γ, T2-δ were reported in (Pulido *et al.*, 2017), the last T2-ϵ in (Zhu *et al.*, 2022).

Note that the polymorph T2-γ contains four slightly different versions in the CSD (DEBXIT01...04) because their crystal structures were determined at different temperatures. The seven versions DEBXIT01...07 with the same 6-letter code may look similar even for experts. T2-δ (SEMDIA) was deposited later than others because even the original authors confused this polymorph with earlier crystals, which was detected by invariants from (Edelsbrunner *et al.*, 2021), computed by (Smith & Kurlin, 2022).

Table 1 includes the upper bounds $\beta(S) \leq \min\{r(U), 2R(S)\}$ from Lemma 3.7(a) in (Anosova *et al.*, 2022), see $r(U)$ and $R(S)$ in Definition 5. The run times in Table 1 were recorded on a laptop with Intel i5, one 1GHz core, 8Gb RAM.

The final row contains the averages for 5,679 simulated T2 crystals, which are publicly available in the supplementary materials of (Pulido *et al.*, 2017) and were used for predicting the 5 experimental polymorphs represented by 9 entries in the CSD. For all crystals in Table 1, the translational matrix size never exceeded 3 columns.

Table 1. *The exact bridge length $\beta(S)$ computed by Algorithm 16 and its upper bounds for the 9 experimental and 5679 simulated T2 crystals reported by (Pulido* et al.*, 2017).*

CSD ref codes of experimental and simulated crystals	number of atoms in a cell	bridge length $\beta(S)$, Å	upper bound $r(U)$, Å	upper bound $2R(S)$, Å	best upper bound over exact $\beta(S)$	running time, seconds
T2-α NAVXUG	184	2.028	22.325	15.609	7.695	4.337
T2-β DEBXIT05	92	3.163	20.665	12.906	4.080	0.664
T2-β DEBXIT06	92	3.188	20.694	12.884	4.042	0.657
T2-γ DEBXIT01	92	1.879	23.224	23.366	12.358	0.706
T2-γ DEBXIT02	92	1.926	23.226	23.375	12.061	0.636
T2-γ DEBXIT03	92	1.902	23.230	23.373	12.216	0.653
T2-γ DEBXIT04	92	1.970	23.290	23.448	11.824	0.649
T2-δ SEMDIA	92	2.713	14.401	8.350	3.077	0.671
T2-ϵ DEBXIT07	92	2.062	12.608	5.707	2.768	0.641
average for all 5679 simulated T2 crystals	295.8	2.293	15.203	9.110	3.973	31.653

The real T2 crystals in the CSD have smaller motifs consisting of only 2 or 4 T2 molecules, while simulated T2 crystals contain up to 32 molecules, which makes the running times slower in comparison with real ones, see the last column in Table 1.

More importantly, the exact bridge length $\beta(S)$ is 4 times smaller (on average) than its upper bound $\min\{r(U), 2R(S)\}$. The bridge length $\beta(S)$ provides the upper bound $\beta(S) + 2R(S) > \alpha(S)$ in Lemma 3.7(b) from (Anosova *et al.*, 2022) for a stable radius α of atomic clouds that suffices for a complete and continuous isoset invariant of S.

This isoset uniquely identifies any periodic crystal S under rigid motion and has a continuous distance metric that has detected thousands of near-duplicate crystals. Decreasing the upper bound of $\alpha(S)$ from $4R(S)$ to the smaller value $\beta(S) + 2R(S)$ by a factor of about 2 decreases the size m of atomic clouds by a factor of $2^3 = 8$ in $\mathbb{R}^3$. This size reduction speeds up by several orders of magnitude the algorithms for isosets and their distance metric, which have complexity $O(m^3 \log m)$ and $O(m^6)$ in $\mathbb{R}^3$, respectively; see the conclusions of section 5 in (Anosova *et al.*, 2022).

The next open problem is an exact computation of the minimal stable radius $\alpha(S)$.

Acknowledgements. We thank Jean-Guillaume Eon and Gregory McColm for their helpful advice on the early draft of this paper. This work was supported by the Royal Society APEX fellowship APX/R1/231152 and EPSRC New Horizons EP/X018474/1.

Appendix A

A.1. Pathsum Matrix Example

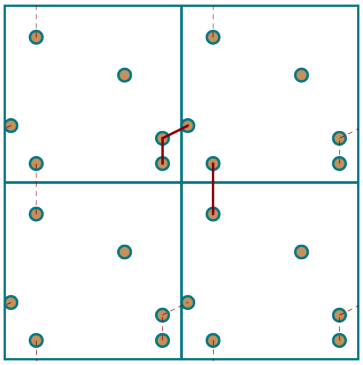
With unit cell:

$$[28, 0], [0, 28]$$

And ordered motif:

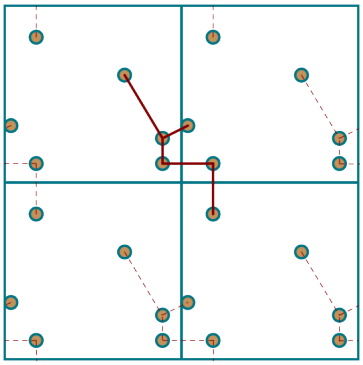
$$(5, 3), (25, 3), (25, 7), (1, 9), (19, 17), (5, 23)$$

We get the incomplete and complete Pathsum Matrices:



$$\begin{pmatrix} (0,0) & NaN & NaN & NaN & NaN & (0,-1) \\ NaN & (0,0) & (0,0) & (1,0) & NaN & NaN \\ NaN & (0,0) & (0,0) & (1,0) & NaN & NaN \\ NaN & (-1,0) & (-1,0) & (0,0) & NaN & NaN \\ NaN & NaN & NaN & NaN & (0,0) & NaN \\ (0,1) & NaN & NaN & NaN & NaN & (0,0) \end{pmatrix}$$

This is only one example of an incomplete Tree and Pathsum Matrix



$$\begin{pmatrix} (0,0) & (-1,0) & (-1,0) & (0,0) & (-1,0) & (0,-1) \\ (1,0) & (0,0) & (0,0) & (1,0) & (0,0) & (1,-1) \\ (1,0) & (0,0) & (0,0) & (1,0) & (0,0) & (1,-1) \\ (0,0) & (-1,0) & (-1,0) & (0,0) & (-1,0) & (0,-1) \\ (1,0) & (0,0) & (0,0) & (1,0) & (0,0) & (1,-1) \\ (0,1) & (-1,1) & (-1,1) & (0,1) & (-1,1) & (0,0) \end{pmatrix}$$

This is the only possible complete Tree and Pathsum Matrix for a Minimal Tree.

As can be seen, we have used NaN for the $Null$(disconnected) values. This is because one powerful way to calculate the matrix is to leverage IEEE 754's NaN to simply add and subtract rows and columns when including new edges in the tree. What may also be seen, is that not only can we now quickly determine which equivalance classes of motif points are already connected (i.e., not NaN), we can also quickly determine

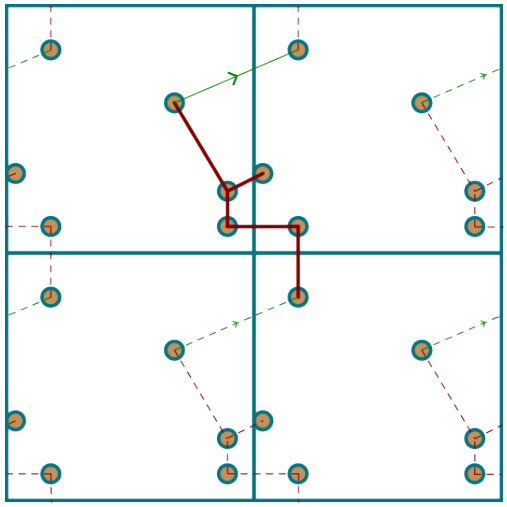
519 cyclesums for any additions to the tree.

520 For example, if we were to add an edge e, formally described as:

$$(source\_index, dest\_index, voltage) \;=\; (4, 5, (1, 0))$$

521 (with zero-based numbering for motif point indices):

522



523

524 Fig. h. Creating a cycle

525 We can see that e has the voltage $(1, 0)$. To get the cyclesum of the cycle formed
526 from adding the edge to the tree (consistent with the direction of e), we simply add v
527 to the Matrix element $R_{5,4}$ as so:

$$v + R_{5,4} \;=\; (1, 0) + (-1, 1) \;=\; (0, 1)$$

528 Which coincides with the figure.

529 *A.2. A faster 'online' algorithm for the Smith Normal Form*

530 A different way of checking the *Termination* condition is to append columns to A
531 in an 'online' fashion. This avoids the need to calculate the Smith Normal Form from
532 scratch every time (or often at all), and reduces the complexity to a time close to $O(m^{\omega} \cdot$

$E \cdot n$), where $\omega \leq 2.372$ is the exponent for matrix multiplication (Williams *et al.*, 2024), and $O(E)$ is the complexity of the Extended Euclidean Algorithm (Baladi & Vallée, 2005). As this reduction in complexity is dominated by the price of populating the edges with Algorithm 13, this will be irrelevant for most use cases (and is not used in the experiments shown later). As a use case involves, say, a larger or higher-dimensional pre-populated set of edges, this algorithm becomes more necessary.

Recalling from Definition 15 that the diagonal of the Smith Normal Form $\mathrm{SNF}(A) = LAR$ is made up of the invariant factors of A. To progressively calculate $\mathrm{SNF}(A)$, we must only keep track of the right-multiplying unimodular matrix R, and the invariant factors themselves, which form a vector $\boldsymbol{f} = (f_1, \ldots, f_n) \in \mathbb{Z}^n$. To run the main algorithm here, we do have to begin with a matrix with n integer linearly independent rows. 'Adding' a vector $\boldsymbol{v}$ to $\boldsymbol{f}$ is where the process changes. We treat R and $\boldsymbol{f}$ as mutable, meaning each value is not necessarily fixed to its original assignment. The first step is to define $x := \boldsymbol{v} * R$, then we find $g_i = gcd(x_i, f_i)$. If $f_i = g_i$ (i.e., f_i *divides* x_i), we can continue with $i := i + 1$, with no need to change R as it only keeps track of columns (for context, if we were keeping track of L, too, we would have to subtract the i-th row from the last row x_i/f_i times).

If f_i divides x_i for all i, we would know that including the vector changes nothing, therefore the relative edge is also irrelevant and can be discarded (this reduces the complexity of most of the *Termination* condition from $O(N)$ to $O(n^\omega + log^2(n))$.

However, if $g_i < f_i$, then f_i not only becomes g_i, but we also know that $\mathrm{SNF}(A)$ will change and that we must add the edge relative to $\boldsymbol{v}$. We must also alter R, accounting for the fact that F represents the diagonal of a matrix. We can do this by any typical process of 'changing the pivot' in the SNF algorithm, ensuring that we update R in tandem. As accounting for the previous values of i is trivial, it is the worst-case equivalent to calculating the SNF of an $(n-i) \times (n-i)$ matrix in time $O(N_{n-i})$, which

improves upon the naive calculation of SNF from scratch upon every alteration of A.

Lemma 27. *Updating the Smith Normal Form as above preserves its properties.*

Proof. As we only alter with elementary row and column operations, this preserves the Smith Normal Form. By multiplying the to-be-added row $\boldsymbol{v}$ by R before concatenating it as a new row to F, it is the same as performing those same elementary column operations upon a new matrix: $[\boldsymbol{A_0}, .., \boldsymbol{A_n}, \boldsymbol{v}]$ (i.e. $\boldsymbol{v}$ concatenated as a row onto A).

We then continue to perform only elementary row and column operations, and we end with a matrix that satisfies the conditions of an SNF noted in Definition 15. □

To discuss this process any further is beyond the scope of this paper, though there are still some small tricks that take advantage of the way the 'new' rows for consideration are intrinsically related to $\boldsymbol{v}$, and how f_{i+1} divides f_i.

References

Anosova, O. & Kurlin, V. (2021). In *Proceedings of Discrete Geometry and Mathematical Morphology*, pp. 229–241.

Anosova, O., Kurlin, V. & Senechal, M. (2024). *IUCrJ*, **11**, 453–463.

Anosova, O., Widdowson, D. & Kurlin, V. (2022). *arXiv:2205.15298 (latest version at https://kurlin.org/projects/periodic-geometry/near-duplicate-periodic-patterns.pdf)*.

Baladi, V. & Vallée, B. (2005). *Journal of Number Theory*, **110**(2), 331–386.

Brock, C. P., (2021). Change to the definition of "crystal" in the IUCr Online Dictionary of Crystallography. https://www.iucr.org/news/newsletter/etc/articles?issue=151351&result_138339_result_page=17.

Chung, S. J., Hahn, T. & Klee, W. (1984). *Acta Crystallographica Section A: Foundations of Crystallography*, **40**(1), 42–50.

Cohen, E. & Megiddo, N. (1990). In *Applied geometry and discrete mathematics*, pp. 135–146. Citeseer.

Cohn, P. M. (1985). *Free rings and their relations*. Academic Press.

Edelsbrunner, H. & Heiss, T. (2024). *arXiv:2408.16575*.

Edelsbrunner, H., Heiss, T., Kurlin, V., Smith, P. & Wintraecken, M. (2021). In *Proceedings of SoCG*, pp. 32:1–32:16.

Eon, J.-G. (2011). *Acta Crystallographica Section A*, **67**(1), 68–86.

Eon, J.-G. (2016*a*). *Acta Cryst A*, **72**(3), 268–293.

Eon, J.-G. (2016*b*). *Acta Cryst A*, **72**(3), 376–384.

Giesbrecht, M. (1995). In *Proceedings of the 1995 international symposium on Symbolic and algebraic computation*, pp. 110–118.

Hatcher, A. (2002). *Algebraic topology*. Cambridge: Cambridge University Press.

Kurlin, V. (2022). *arXiv:2205.04388*.

McColm, G. (2024). *Acta Crystallographica Section A*, **80**(1).

Newman, M. (1972). *Integral matrices*. Academic Press.

Onus, A. & Robins, V. (2022). *arXiv:2208.09223*.

Pulido, A. *et al.* (2017). *Nature*, **543**, 657–664.

Sedgewick, R. (1983). *Algorithms*. Addison-Wesley series in computer science and information processing. Addison-Wesley.

Smith, P. & Kurlin, V. (2022). In *Lecture Notes in Computer Science (Proceedings of ISVC)*, vol. 13599, pp. 377–391.

Taylor, R. & Wood, P. A. (2019). *Chemical reviews*, **119**(16), 9427–9477.

Van der Waerden, B. L. (2003). *Algebra*, vol. 1. Springer Science & Business Media.

Widdowson, D. & Kurlin, V. (2021). *arxiv:2108.04798v3*.

Widdowson, D. & Kurlin, V. (2022). *Advances in Neural Information Processing Systems (NeurIPS)*, **35**, 24625–24638.

Widdowson, D. & Kurlin, V. (2024). *Crystal Growth and Design*, **24**, 5627–5636.

Widdowson, D., Mosca, M. M., Pulido, A., Cooper, A. I. & Kurlin, V. (2022). *MATCH Commun. Math. Comput. Chem.* **87**, 529–559.

Williams, V. V., Xu, Y., Xu, Z. & Zhou, R. (2024). In *Symposium on Discrete Algorithms*, pp. 3792–3835. SIAM.

Zhu, Q. *et al.* (2022). *J Amer. Chem. Soc.* **144**, 9893–9901.

Synopsis

We describe an efficient algorithm to compute the bridge length estimating the size of a complete isoset invariant, which classifies all periodic point sets under Euclidean motion.