A new approach to generating derivative structures

Gus L. W. Hart and Rodney W. Forcade

Department of Physics & Astronomy, Brigham Young University, Provo Utah 84602 USA
Department of Mathematics, Brigham Young University, Provo Utah 84602 USA

(Dated: May 1, 2008)

We present an algorithm for generating all derivative superstructures—for arbitrary parent structures and for any number of atom types. This algorithm enumerates superlattices and atomic configurations in a geometry-independent way. The key concept is to use the quotient group associated with each superlattice to determine all unique atomic configurations. The run time of the algorithm scales linearly with the number of unique structures found. We show several applications demonstrating how the algorithm can be used in materials design problems. We predict an altogether new crystal structure in Cd-Pt and Pd-Pt, and several new ground states in Pd-rich and Pt-rich binary systems.

I. WHY DERIVATIVE STRUCTURES?

Derivative superstructures\textsuperscript{1} play an important role in different materials phenomenon such as chemical ordering in alloys, spin ordering in magnets, and vacancy ordering in non-stoichiometric materials. Similarly, derivative superlattices\textsuperscript{2–4} are important in problems such as twinning. What is a derivative superstructure? A derivative superstructure is a structure whose lattice vectors are multiples of those of a ‘parent lattice’ and whose atomic basis vectors correspond to lattice points of the parent lattice. Many structures of intermetallic compounds can be classified as fcc-derived superstructures. These superstructures have atomic sites that closely correspond to the sites of an fcc lattice but some of the translational symmetry is broken by a periodic arrangement of different kinds of atoms. The structures shown in Fig. 2 comprise the set of all fcc-derived binary superstructures with unit cell sizes 2, 3, and 4 times larger than the parent lattice.

Large sets of derivative superstructures are often used in (practically) exhaustive searches of binary configurations on a lattice to determine ground state properties of intermetallic systems. The approach is not limited to searches of configurational energies, but other physical observables can also be targeted if an appropriate Hamiltonian is available. For example, Kim et al.\textsuperscript{5} used an empirical pseudopotential Hamiltonian and a large list of derivative superstructures to directly search semiconductor alloys for desirable band-gaps and effective masses. The set of derivative superstructures is useful in any situation where the physical observable of interest depends on the atomic configuration.

For the aforementioned reasons, an algorithm for systematically generating all superstructures of a given parent structure is useful. Such an algorithm has been presented in the literature only once\textsuperscript{6,7} (FWZ below). The FWZ algorithm generates superlattices using a geometric, ‘smallest first’ approach\textsuperscript{8} (see page 44 of Ref. 6) and then generates all unique atomic configurations. Many of the resulting structures are equivalent by rotational and/or translational symmetry. FWZ removes duplicates by calculating interatom correlations (pairs, triplets, etc.) averaged over the structure, eliminating structures with matching correlations. Because of the limited range of correlations used in the original FWZ implementation, some formally inequivalent structures are eliminated, though in practice the list may be sufficient.

Though the original presentation of the FWZ algorithm is restricted to fcc- and bcc-based superstructures and to binary cases only, the superlattice generation can be extended to the general case and it was implemented in several alloy modeling packages. The code of Ref. 9 overcomes the deficiencies of finding duplicates via correlations by using a rigorously complete list of correlations.\textsuperscript{10} The ATAT package\textsuperscript{11,12} handles the issue by directly comparing the structures geometrically.

The purpose of this paper is to present a general algorithm that generates a formally complete list of two- or three-dimensional superstructures, and that works for any parent lattice and for arbitrary $k$-nary systems (binary, ternary, etc.). This algorithm is conceptually distinct from FWZ and related implementations. Instead of using a geometrical approach, it takes advantage of known group theoretical properties of integer matrices. The algorithm is orders of magnitude faster than FWZ, more general, and formally complete. A Fortran 95 implementation of the algorithm is included with this paper as supplementary material.\textsuperscript{13}

Mathematically, we can describe the purpose of the

FIG. 1: An example parent lattice (left) and a superstructure (right). The parent lattice is fcc and the superlattice is defined by the (doubled) unit cell outlined in gray. The two interior points of the superlattice are occupied by one black atom and one gray atom. The superlattice + atoms constitute a derivative structure. The superstructure of this example is that of CuAu.
algorithm as this: for a given parent lattice, enumerate all possible superlattices and all rotationally- and translationally-unique ‘colorings’ or labelings of each superlattice. In presenting the algorithm in the next section, we shall refer to superlattices and labelings rather than referring to crystal structures or atomic sites.

II. ENUMERATING ALL DERIVATIVE STRUCTURES

Here is a brief outline of the algorithm.

1. For each superlattice of size \(n\), generate all Hermite Normal Form (HNF) matrices.\(^3\)\(^,\)\(^4\) (In what follows, we refer to \(n\) as the index of the superlattice.)

2. Use the symmetry of the parent lattice to remove rotationally-equivalent superlattices, thus shrinking the list of HNF matrices.

3. For each superlattice index \(n\), find the Smith Normal Form (SNF) of each HNF in the list, and:
   - (a) Generate a list of possible labelings (atomic configurations) for each SNF, essentially a list of all \(k^n\) numbers in a base \(k\), \(n\)-digit system. For the labels, we use the first \(k\) letters of the alphabet, \(a, b, \cdots\).
   - (b) Remove incomplete labelings where each of the \(k\) labels \((a, b, \cdots)\) does not appear at least once.\(^15\)
   - (c) Remove labelings that are equivalent under translation of the parent lattice vectors. This reduces the list of labelings by a factor of \(\approx n\).
   - (d) Remove labelings that are equivalent under an exchange of labels, i.e., \(a \leftrightarrow b\), so that, e.g., the labeling \(aabb\) is removed from the list because it is equivalent to \(bbaa\).
   - (e) Remove labelings that are super-periodic, i.e., labelings that correspond to a non-primitive superstructure. This can be done without using the geometry of the superlattice.

4. For each HNF, remove labels that are permuted by symmetry operations (of the parent lattice) that leave the superlattice fixed.

An important feature of the algorithm is that the list of possible labelings, generated in step 3a, form a minimal hash table with a perfect hash function. Eliminating all duplicate labelings in a list of \(N\) can be accomplished in \(O(N)\) time. Coupled with the group theoretical approach, this results in an extremely efficient algorithm that is orders of magnitude faster than FWZ. Enumerating fcc-derived structures up to \(n = 20\) \((N \sim 2^{20} \approx 10^6)\) takes a dozen minutes with the present algorithm but more than one day with FWZ. The case of \(n = 24\) \((N \sim 2^{24} \approx 20^7)\) takes less than two hours but about one month with FWZ. More significantly, the run time of the algorithm scales linearly. That is, the time to find \(N\) unique structures is proportional to \(N\)—the best possible scaling for this type of problem. This scaling for binary superstructures of an fcc parent lattice is shown in Fig. 3.

A. Generating all superlattices

Given a ‘parent’ cell (any lattice), the first step in finding all derivative structures of that cell is to enumerate all derivative superlattices. Consider the transformation \(B = \mathcal{A}H\) where \(A\) is the basis for the original lattice (the basis vectors listed column-wise), \(H\) is a matrix with all...
integer elements, and $\mathbb{B}$ is the matrix of the transformed lattice. If the determinant of the transformation matrix, $|\mathbb{H}|$, is $\pm 1$, then $\mathbb{H}$ is merely a change of basis that leaves the lattice represented by $\mathbb{A}$ unchanged. Matrices $\mathbb{A}$ and $\mathbb{B}$ are merely two different choices of basis for the same lattice. On the other hand, if the elements of $\mathbb{H}$ are all integers but the determinant of $\mathbb{H}$ is 2, say, then the lattice of $\mathbb{B}$ is a superlattice of (i.e., a subgroup) of the lattice defined by $\mathbb{A}$, but with twice the volume of the original lattice.

Two different matrices, $\mathbb{H}_1$ and $\mathbb{H}_2$, with the same determinant, will generate different bases for the same lattice if and only if $\mathbb{H}_1$ can be reduced to $\mathbb{H}_2$ by elementary integer column operations. The canonical form for such operations is lower triangular Hermite Normal Form (HNF). Thus, if we use only matrices $\mathbb{H}$ which are in HNF, we will produce exactly one representation of each superlattice.\(^4\) In three dimensions, the lower-triangular Hermite Normal Form is

$$
\begin{pmatrix}
a & 0 & 0 \\
b & c & 0 \\
d & e & f
\end{pmatrix}
$$

with $0 \leq b < c$, $0 \leq d, e < f$. \(^1\)

In this form, the product of the integers on the diagonal alone, $a \times c \times f$, fixes the determinant. Again, we refer to the superlattice size, or the determinant, as the index $n$. Generating all HNF matrices of a given index can be done then by finding each unique triplet, $acf = |\mathbb{H}|$, and then generating all values of $b, d,$ and $e$ that obey the conditions in \(^1\).

The algorithm for generating all possible HNF matrices of a given index $|\mathbb{H}|$ is rather simple, comprising just two steps. In the first step, find all possible diagonals: find all values $a$, $1 \leq a \leq |\mathbb{H}|$, that evenly divide $|\mathbb{H}|$; for each of these values, find all $c$, $1 \leq c \leq |\mathbb{H}|/a$, that evenly divide $|\mathbb{H}|/a$. For each value of $c$, let $f = |\mathbb{H}|/(ac)$. For example, consider the case of $|\mathbb{H}| = 6$. We execute two nested loops over the possible values of $a$ and $c$; each loop runs over all integers between 1 and the $|\mathbb{H}|$, testing the above conditions at each iteration. The loops run from 1 to 6, and the algorithm finds eight cases that meet the above conditions. They are:

$$
\begin{array}{cccccccc}
 a & 1 & 1 & 1 & 1 & 2 & 2 & 3 \\
 b & 1 & 2 & 3 & 1 & 3 & 1 & 2 \\
 c & 1 & 2 & 3 & 1 & 3 & 1 & 2 \\
 d & 6 & 3 & 2 & 3 & 1 & 2 & 1 \\
 e & f & & & & & &
\end{array}
$$

The set of $acf$ triplets generated during this first step comprises all possible diagonals of the HNF matrices for the case of $n = |\mathbb{H}| = 6$. The second step, generating each set of values of $b, d,$ and $e$ for each diagonal (set of $acf$ triplets), can be accomplished simply by three nested loops that start at zero and terminate at $b < c$ and $d, e < f$.

As an example of both steps, consider the case where the index is merely double that of the original lattice, i.e., where $|\mathbb{H}| = 2$. The factors of 2 are just the set $\{1, 2\}$, so the first step finds only three cases: $(2,1,1)$, $(1,2,1)$, and $(1,1,2)$. Then, generating the off-diagonal terms for each of these three cases, we find seven HNF matrices:

- **case 1:** \((2\ 0\ 0)\)
- **case 2:** \((1\ 0\ 0)\)
- **case 3:** \((0\ 1\ 0)\)

For increasing index, $n = |\mathbb{H}| = 1, 2, 3, \ldots$, the number of HNF matrices generates an interesting sequence: 1, 7, 13, 35, 31, 91, \ldots. We find the closed-form expression for $n$-th term in the series is

$$
\sum_{d|n} d \cdot \sigma(d) = \prod_{i=1}^{k} \left( \frac{(p_i^{e_i+2} - 1)(p_i^{e_i+1} - 1)}{(p_i - 1)^2(p_i + 1)} \right),
$$

where $\sigma$ is the sum of divisors function, and where the $p_i$ and $e_i$ are the prime factors and powers of $n$: $n = p_i^{e_i} \cdots p_k^{e_k}$. This expression is the same as that given for Sloane’s A001001.\(^16\) The sequence appears in the crystallography literature as well as several other contexts.\(^19–22\)

Significantly, because we have an expression for the number of superlattices, the implementation of the HNF-generating algorithm can be rigorously checked. Also note that this step of the algorithm is independent of the choice of parent lattice.

### B. Reducing the HNF list by parent lattice symmetry

The set of HNF matrices defines the set of all derivative superlattices of a parent cell via the transformation
mentioned above, \( \mathbb{B} = A \mathbb{H} \). However, not all of the superlattices in this set will be geometrically different. Some distinct lattices will be equivalent under symmetries of the parent lattice, illustrated in the example below.

Such duplicate superstructures must be eliminated by the algorithm. At the end of the algorithm we want all derivative structures to be unique from a materials point of view. So we wish to exclude from the list any superstructures that are related to others already in the list simply by a rotation, reflection or change of basis.

As an illustration, consider a two-dimensional parent lattice that is square, that is, \( A = \mathbb{I} \) (the two-by-two identity matrix). There are three HNF matrices for which \(|H| = 2\) and three corresponding superlattices, \( \mathbb{B} = A H = \mathbb{I} \mathbb{H} = \mathbb{H} \):

\[
\begin{pmatrix}
2 & 0 \\
0 & 1
\end{pmatrix}, \quad \begin{pmatrix}
1 & 0 \\
0 & 2
\end{pmatrix}, \quad \begin{pmatrix}
1 & 0 \\
1 & 2
\end{pmatrix}
\]

The parent lattice itself is indicated by dots (filled and unfilled) while the superlattice is indicated by filled dots. The vectors defined by the matrices are shown as arrows. The first two lattices are clearly equivalent under a 90\(^\circ\) rotation, one of the eight symmetry operations of a square lattice.

To enumerate the distinct superlattices of a given index \( n \) then, we must check that each new superlattice that is added to the list is not a rotated duplicate of a previous superlattice. More precisely, we must check that each new basis \( B_k \) is not equivalent, under change-of-basis, to some symmetric image \( R B_j \) of a basis \( B_j \) already in the list. In other words, we want to avoid the relation \( B_j = R B_j \mathbb{H} \) where \( B_j \) is a candidate superlattice, \( R \) is any of the rotations of the parent lattice, \( B_j \) is a superlattice already in the list of distinct superlattices, and \( \mathbb{H} \) is any unimodular matrix of integers. (Since \( B_i \) and \( B_j \) have the same determinant, we will only need to check that \( B_j^{-1} R^{-1} B_i \) is a matrix of integers.)

For the case of cubic symmetry, the seven superlattices for the \( \mathbb{H} = 2 \) case mentioned above reduce to only two symmetrically-distinct superlattices. The corresponding derivative superstructures are \( L_{10} \) and \( L_{11} \), both well-known structures in intermetallic compounds. The fact that these are the only two 2-atom/cell fcc structures is not coincidence or an accident of chemistry; no other 2-atom/cell structures are possible geometrically. The hierarchy of physically-observed structures uncovered for fcc and bcc lattices as the index is increased is discussed in Refs. 23, 24.

C. Find the unique labelings for all superlattices

1. Generate all possible labelings

For each HNF, each superlattice, we start by generating all possible labelings of that superlattice. In other words, given \( k \) colors (types of atoms), represented by the labels \( a, b, \cdots \), we generate all possible ways of labeling (coloring) the superlattice. Each HNF matrix of determinant size \( n \) represents a superlattice with \( n \) interior points to be decorated. If the number of colors is \( k \), then the list of all possible labelings is easily represented by the list of all \( n \)-digit, base-\( k \) numbers. So, from a combinatorial point of view, there are \( k^n \) distinct labelings. For example, in the case of a binary system \( (k = 2) \) with 4 interior points (index \( n = 4 \)), there are \( 2^4 = 16 \) possible labelings (see Table I).

2. Concept of eliminating duplicate labelings

The rest of the algorithm deals with just one conceptual issue—given the \( k^n \) labelings/colorings of the superlattice, eliminate the duplicates. In the FWZ algorithm and its extended implementations, duplicate structures are eliminated by comparing\(^{25}\) one candidate structure to another, necessitating an expensive \( O(N^2) \) search. We eliminate the duplicates via group theory rather than checking the structures themselves. Although this approach is more abstract than the geometric approach, it is much more efficient—eliminating the duplicates in a list becomes a strictly \( O(N) \) procedure.

One-dimensional example: We start with a simple illustration and then discuss the essential group theoretical concepts in the context of that example. Consider the one-dimensional case of Fig. 4. The first line (I) is a parent lattice, an infinite collection of equally-spaced points, identified with the set of integers, denoted \( \mathbb{Z} \). The second line (II) is a superlattice, a subset of the parent lattice (every fourth point; those colored black). The third line (III) is a superstructure, a ‘labeling’ or ‘coloring’ of the parent lattice that has the same periodicity as the superlattice. The points of the lattice play the role of positions...
in a crystal, and the colors/labels play the role of atoms placed at those positions.

There are labelings that are distinct yet physically equivalent, as shown in Fig. 5. If Fig. 5 note that line (II) is obtained from line (I) by shifting the colors two units to the right, and line (III) is obtained from line (I) by shifting the numbers two units to the left—with the same result. Lines (II) and (III) are the same labeling, obtained in different ways from (I). Both are physically equivalent to (I). The fact that we can obtain such a shifted labeling either by shifting the numbers or by shifting the labels explains why we can do much of our equivalence checking within a finite group, instead of geometrically within an infinite lattice. By this method, we will identify these equivalent labelings and remove them from the original list of \( k^n \) labelings.

We may illustrate the group theory approach using Fig. 4. The parent lattice (I) is the set of integers \( \mathbb{Z} \), which is a group under the addition operation. We refer to this group as \( L \). The superlattice is the set of multiples of 4, denoted \( 4\mathbb{Z} \). We refer to this subgroup of \( L \) as \( S \). We label the parent lattice \( L \) in a manner which is periodic with respect to the superlattice \( S \), and note that if two points differ by an element of the superlattice, they must receive the same label. We use colors as labels in line (III) of Fig. 4 and note that every fourth point has the same color.

Notice that the green points are our superlattice are \( 4\mathbb{Z} \), and the yellow points are a copy of \( 4\mathbb{Z} \), but translated one unit to the right. Thus, we may denote the latter set (the yellow points) by the set \( 1 + 4\mathbb{Z} \). Similarly, the red points are the set \( 2 + 4\mathbb{Z} \), and the blue points are \( 3 + 4\mathbb{Z} \). These four sets, \( 4\mathbb{Z}, 1 + 4\mathbb{Z}, 2 + 4\mathbb{Z}, 3 + 4\mathbb{Z} \), are mutually disjoint (they don’t overlap), and their union is the entire parent lattice \( L \). They are translations of \( S \); and thus are the cosets of the subgroup \( S \). This means we can use them to form a new group, called the quotient group \( G = L/S \) (see Table II). This new group is finite, having only four elements. For notational convenience we shall also refer to these 4 elements of \( G \) as \( (0,1,2,3) \). We need only label the four elements of our quotient group in order to label the entire parent lattice.

Suppose we wish to translate a labeling (in order to identify and eliminate equivalent structures). As shown in line (I) of Fig. 5 we have labeled the elements of the quotient group as follows (using \( g, y, r \) and \( b \) for the colors):

\[
\begin{align*}
0 & : \quad 4\mathbb{Z} \rightarrow g \\
1 & : \quad 1 + 4\mathbb{Z} \rightarrow y \\
2 & : \quad 2 + 4\mathbb{Z} \rightarrow r \\
3 & : \quad 3 + 4\mathbb{Z} \rightarrow b.
\end{align*}
\]

In Fig. 5, we see that translating the labels by two is the same as simply adding 2 to each coordinate, thus:

\[
\begin{align*}
4\mathbb{Z} & \rightarrow 2 + 4\mathbb{Z} & 0 & \rightarrow 2 \\
1 + 4\mathbb{Z} & \rightarrow 3 + 4\mathbb{Z} & 1 & \rightarrow 3 \\
2 + 4\mathbb{Z} & \rightarrow 4 + 4\mathbb{Z} = 4\mathbb{Z} & 2 & \rightarrow 0 \\
3 + 4\mathbb{Z} & \rightarrow 5 + 4\mathbb{Z} = 1 + 4\mathbb{Z} & 3 & \rightarrow 1
\end{align*}
\]

The effect is the same as if we had assigned the colors differently:

\[
\begin{align*}
0 & : \quad 4\mathbb{Z} \rightarrow r \\
1 & : \quad 1 + 4\mathbb{Z} \rightarrow b \\
2 & : \quad 2 + 4\mathbb{Z} \rightarrow g \\
3 & : \quad 3 + 4\mathbb{Z} \rightarrow y.
\end{align*}
\]

Translating the lattice by adding +2 to every point (moving the origin by 2 units) has the same effect on the labeling as if we had merely labeled the four elements of the quotient group, and then added +2 to every element of the group, producing the permutation \( 0 \rightarrow 2, 1 \rightarrow 3, 2 \rightarrow 0 \) and \( 3 \rightarrow 1 \), denoted \( (2,0,3,1) \).

Instead of determining that two labelings of the (infinite) lattice are equivalent by translation, we may simply check that the corresponding labelings of our finite quotient group \( G = \mathbb{Z}_4 \) are equivalent. We do this by just adding a fixed element to every element in the group, effecting a permutation of the cyclic group \( \mathbb{Z}_4 \). This idea—of labeling the quotient group instead of the lattice elements, and checking equivalence within the group instead of by translating the lattice—may seem unduly abstract and unnecessary in one dimension, but it becomes much
TABLE II: Two representations of the Cayley table for the quotient group $G$ (cosets of the subgroup $S$). On the right, the elements of the group have been denoted $(0,1,2,3)$ for notational convenience.

<table>
<thead>
<tr>
<th></th>
<th>$Z$</th>
<th>$1 + Z$</th>
<th>$2 + Z$</th>
<th>$3 + Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z$</td>
<td>$1 + Z$</td>
<td>$2 + Z$</td>
<td>$3 + Z$</td>
<td>$Z$</td>
</tr>
<tr>
<td>$1 + Z$</td>
<td>$1 + Z$</td>
<td>$2 + Z$</td>
<td>$3 + Z$</td>
<td></td>
</tr>
<tr>
<td>$2 + Z$</td>
<td>$2 + Z$</td>
<td>$3 + Z$</td>
<td>$1 + Z$</td>
<td></td>
</tr>
<tr>
<td>$3 + Z$</td>
<td>$3 + Z$</td>
<td>$1 + Z$</td>
<td>$2 + Z$</td>
<td></td>
</tr>
</tbody>
</table>

TABLE III: Listing the number of Hermite normal form (HNF) matrices and Smith normal form (SNF) matrices as a function of index $n$ (determinant size). The number of HNFs is a rapid increasing function of $n$ (see Eq. 2) whereas the number of SNFs grows very slowly.

<table>
<thead>
<tr>
<th>$n$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>HNFs</td>
<td>7</td>
<td>13</td>
<td>35</td>
<td>91</td>
<td>157</td>
<td>213</td>
<td>271</td>
<td>333</td>
<td>399</td>
<td>465</td>
<td>531</td>
<td>597</td>
<td>663</td>
<td>729</td>
<td>895</td>
</tr>
<tr>
<td>SNFs</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FIG. 6: A superlattice whose SNF is non-trivial (non-cyclic). Because the quotient group of the superlattice is non-cyclic, translations of the lattice are not cyclic permutations. Instead the sites are permuted in groups of two or four with each translation (rather than permuting as a single group of 8). In the figure, three different pictures of the same superlattice are shown. In each case a different origin is chosen. In the second case, the translation permutes two groups of 4 sites, in the third, 4 pairs are permuted.

Hence, we show that the determinant size.

3. Eliminating translation duplicates

Because of its periodicity, the choice of origin of a superlattice is arbitrary. A change in origin implies a permutation of the labels that nonetheless defines the same superstructure (compare lines I and III in Fig. 5). As stated previously, by examining the quotient group instead of directly comparing the structures, the duplicate labelings can be readily identified. For example, consider the case for $n = 4$. Adding each member to the quotient group $Z_4 = (0, 1, 2, 3)$ effects 4 permutations as follows:

<table>
<thead>
<tr>
<th>member</th>
<th>mapping</th>
<th>permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:</td>
<td>0 → 0, 1 → 1, 2 → 2, 3 → 3</td>
<td>(0,1,2,3)</td>
</tr>
<tr>
<td>1:</td>
<td>0 → 1, 2 → 3, 3 → 0</td>
<td>(1,2,3,0)</td>
</tr>
<tr>
<td>2:</td>
<td>0 → 2, 1 → 3, 3 → 1</td>
<td>(2,0,3,1)</td>
</tr>
<tr>
<td>3:</td>
<td>0 → 3, 2 → 1, 1 → 0</td>
<td>(3,0,1,2)</td>
</tr>
</tbody>
</table>

If we take the 14 complete labelings of Table I and the three non-trivial permutations above, we find that 10 are duplicates (colored purple in Table I):

<table>
<thead>
<tr>
<th>duplicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaab</td>
</tr>
<tr>
<td>aabb</td>
</tr>
<tr>
<td>abab</td>
</tr>
<tr>
<td>abbb</td>
</tr>
</tbody>
</table>

Of the original $2^4 = 16$ labelings, two were discarded immediately because they were incomplete. Of the re-
But there are still more duplicates to remove. Some of the cause they are identical under at least one permutation. Naturally, there is no difference between a superlattice and a derivative superlattice, (II) a derivative superlattice (index \( n = 4 \)). The superstructure of line III would have been found as an index \( n = 2 \) derivative structure and is therefore a duplicate.

remaining 14, 10 are translation duplicates, leaving 4 that are translationally inequivalent (left column above).

4. **Remove ‘label-exchange’ duplicates**

The next step in the algorithm is to remove labelings that are equivalent under exchange of labels. Structurally, there is no difference between a superlattice whose interior points are labeled \( aaab \) versus \( bbba \). Although the energy of an isostructural compound with composition \( X_3Y_1 \) is different from one with composition \( X_1Y_3 \), we only wish to include one entry in our list of derivative superstructures because the full composition list can always be recovered by making all possible label exchanges (i.e., \( a \leftrightarrow b \)). In the example above, four labelings were unique under translations:

- \( aaab \)
- \( aabb \)
- \( abab \)
- \( abbb \)

But the first and the fourth are equivalent by exchanging \( a \leftrightarrow b \) and applying the permutation \((1,2,3,0)\).

5. **Remove super-periodic labelings (non-primitive structures)**

At this point of the algorithm, many of the duplicate labelings have been removed from the original \( k^n \) list. But there are still more duplicates to remove. Some of the labelings in the list will represent superstructures that are not primitive. In other words, the labelings will be super-periodic—they will have periods shorter scale than the superlattice.\(^{26}\)

The super-periodic duplicates are easily identified because they are identical under at least one permutation.

The quotient group \( G \) dictates a set of permutations under which the labelings are duplicate. One of these permutations will leave the labeling unchanged if it is super-periodic. For example, continuing the example above, three unique labelings are still in the list: \( aaab, aabb, \) and \( abab \). One of the permutations of the quotient group \( G = Z_4 \) is \((2,0,3,1)\). Under this permutation, the labeling \( abab \) is unchanged. Thus it is super-periodic, as depicted in Fig. 7. It is a duplicate in the sense that the algorithm would have already enumerated this structure with the index \( n = 2 \) structures.

6. **For each HNF: remove ‘label-rotation’ duplicates**

The previous three steps of the algorithm yield a list of distinct labelings for each SNF of index \( n \). Three kinds of the duplicate labelings, translation duplicates and label-exchange duplicates and super-periodic duplicates, have already been removed. One kind of duplicate remains, however, and these are eliminated in the current step.

This step removes labelings which are permuted by the rotations of the parent lattice. Whereas the preceding steps were applied to generate a list of unique labelings for each SNF, the current step must be applied to each HNF. In other words, this step must be applied to the surviving labelings separately for each superlattice.

Superlattices which are not fixed by rotations of the parent lattice were already eliminated as duplicates in step 2 of the algorithm. But rotations which leave the superlattice unchanged may still permute the labeling itself. Such permutations are physically equivalent (merely rotated with respect to one another). So any two labelings which are equivalent under rotations that fix the superlattice are duplicate and one must be removed from the list. Figure 8 illustrates the situation in two dimensions.

Here again, the group theory approach and the SNF makes the search extremely efficient. Label rotation duplicates can be identified easily using the properties of the quotient group and the SNF transformation. The row and column operations required to transform the

![FIG. 8: (color on-line) Two identical 4 × 4 superlattices (dotted lines) with rotationally-equivalent labelings (red and blue circles). Although the superlattices themselves are unchanged, a 90° rotation applied to the left labeling yields that shown on the right. Thus the second is a duplicate of the first and should be removed from the list of labelings.](image-url)
HNF matrix of a superlattice into its SNF can be represented by two integer transform matrices, \( L \) and \( R \), so that \( LHR = S \), where \( S \) is the SNF. This step of the algorithm is implemented using the left transformation matrix \( L \).

Let \( G \) be a \( 3 \times n \) matrix where each member of the quotient group is represented as a column\(^{27} \) in \( G \), and let \( R \) be one of the rotations that fixes the superlattice. Then the permutation of the labels (which is the same as the permutations of the quotient group) enacted by the rotation \( R \) is given by:

\[
G' = L A^{-1} R (L A^{-1})^{-1} G
\]  

(3)

where columns of \( A \) are the lattice vectors of the parent lattice, and \( L \) is the left transformation matrix for the SNF.

The power of this expression is that it allows the label-rotation duplicates to be identified by working entirely within the quotient group, without requiring any explicit reference to the geometry of the superlattice. Thus, as in the other steps, duplicates labelings can be eliminated in a time proportional only to the number of labelings in the list.

### III. EXAMPLES

In this section, we give several example derivative structure lists enumerated by the algorithm. We discuss the symmetry reduction of the structure lists and then give results for several cases. First, we compare the fcc/bcc binary list to that generated by the FWZ algorithm. We also show the small-unit-cell binary structures for a simple-cubic parent lattice and the small-unit-cell ternary structures for an fcc parent lattice.

#### A. Symmetry reduction of superlattices

In step B of the algorithm, the complete list of HNF matrices is reduced to those that are unique under the symmetry operations of the parent lattice. Asymptotically, the factor by which the list is reduced is one half the order of the rotation group of the parent lattice. For example, for cubic parent lattices (simple cubic, face-centered-, or body-centered-cubic), the point group contains 48 rotations (proper and improper). For superlattices with large index \( n \), the number of HNFs is reduced by a factor of \( 48/2 = 24 \). Because every lattice is symmetric under inversion, only the proper rotations (i.e., not reflections) need be considered in the reduction (thus the factor of \( 1/2 \)). Figure 9 shows the fraction of symmetrically distinct superlattices for determinant sizes up to 100, while Fig. 10 shows the actual number of fcc-based superlattices compared to the total number of distinct HNF matrices.

For an fcc or bcc parent lattice (the numbers are the same), the number of unique lattices as a function of index \( n \) (cell size) appears to be equivalent to the Sloane sequence A045790.\(^{28} \) For the sequences generated for other parent lattices, which accordingly have a different symmetry group, there are no known number-theoretic connections. Surprisingly, this is even true for the simple cubic lattice. For the simple cubic lattice, the sequence is identical to the fcc/bcc one for odd values of the index \( n \) but larger for the even values. (See Table IV.)

#### B. Number of structures of different parent lattices

The number of superstructures increases much faster than the number of superlattices as a function of \( n \). In general, each superlattice has many different unique labelings. Table V shows the number of fcc/bcc derivative
TABLE IV: Number of symmetrically unique superlattices (HNFs) for several different parent lattices as a function of the index size \( n \). Note that fcc/bcc are the same as simple cubic (sc) only for the odd values of \( n \), and always smaller for the even values. Hexagonal (hex) and simple tetragonal parent lattices have more unique lattices than the cubic systems because of their lower symmetry.

<table>
<thead>
<tr>
<th>( n )</th>
<th>fcc/bcc</th>
<th>sc</th>
<th>hex</th>
<th>tetragonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>13</td>
<td>19</td>
<td>29</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>24</td>
<td>34</td>
<td>51</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>14</td>
<td>23</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>23</td>
<td>33</td>
<td>53</td>
</tr>
</tbody>
</table>

TABLE V: Number of unique fcc derivative structures as a function of the index \( n \). The second and fifth columns show the number of unique structures for each \( n \), while the third and sixth columns show the cumulative total.

<table>
<thead>
<tr>
<th>( n )</th>
<th>structures cumulative</th>
<th>( n )</th>
<th>structures cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>13</td>
<td>2624</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>14</td>
<td>9628</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>15</td>
<td>16584</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>16</td>
<td>49764</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>17</td>
<td>42135</td>
</tr>
<tr>
<td>7</td>
<td>52</td>
<td>18</td>
<td>212612</td>
</tr>
<tr>
<td>8</td>
<td>229</td>
<td>19</td>
<td>174104</td>
</tr>
<tr>
<td>9</td>
<td>252</td>
<td>20</td>
<td>867893</td>
</tr>
<tr>
<td>10</td>
<td>685</td>
<td>21</td>
<td>1120708</td>
</tr>
<tr>
<td>11</td>
<td>682</td>
<td>22</td>
<td>2628180</td>
</tr>
<tr>
<td>12</td>
<td>3875</td>
<td>23</td>
<td>3042732</td>
</tr>
</tbody>
</table>

TABLE VI: Simple cubic superstructures for \( n \leq 4 \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>HNF SNF superlattices labelings</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7 1 3 3 3</td>
</tr>
<tr>
<td>3</td>
<td>13 1 3 3</td>
</tr>
<tr>
<td>4</td>
<td>35 2 9 15</td>
</tr>
<tr>
<td>total</td>
<td>55 4 15 21</td>
</tr>
</tbody>
</table>

TABLE VII: Number of ternary and quaternary derivative structures for an fcc parent lattice. Compare to the number of binary structures of Table V. As the number of labels \( k \) is increased, the number of derivative structures increases rapidly.

<table>
<thead>
<tr>
<th>( n )</th>
<th>ternary structures cumulative</th>
<th>quaternary structures cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>23</td>
<td>39</td>
</tr>
<tr>
<td>6</td>
<td>130</td>
<td>169</td>
</tr>
<tr>
<td>7</td>
<td>197</td>
<td>366</td>
</tr>
<tr>
<td>8</td>
<td>1267</td>
<td>1633</td>
</tr>
<tr>
<td>9</td>
<td>2322</td>
<td>3955</td>
</tr>
<tr>
<td>10</td>
<td>9332</td>
<td>13287</td>
</tr>
</tbody>
</table>

IV. SUMMARY

We developed an algorithm for enumerating derivative structures. The results of such an algorithm are useful in a variety of physics and materials science contexts. The algorithm first generates all unique superlattices by enumerating all Hermite normal form matrices and using the symmetry operations of the parent lattice to eliminate rotationally equivalent superlattices. Next, the algorithm generates all possible atomic configurations (labelings) of each superlattice and eliminates duplicates using a group-theoretical approach rather than geometric analysis.

The algorithm is exceptionally efficient due to the use of (i) perfect, minimal hash tables and (ii) a group-theoretical approach to eliminating duplicate structures.
These two features result in a linearly scaling algorithm that is orders of magnitude faster than the previous method. Moreover, the method can be applied to any parent lattice and to arbitrary \( k\)-ary systems (binary, ternary, quaternary, etc.). The method is formally complete (does not undercount) and key parts of the algorithm (and its implementation) can be rigorously checked by number theory results and Burnside’s lemma.

V. ACKNOWLEDGMENTS

G. L. W. H. gratefully acknowledges financial support from the National Science Foundation through Grant No. DMR-0650406. G. L. W. H. thanks Martin L. Searcy and Bronson S. Argyle for useful discussions concerning algorithm development and implementation of hash tables and hash functions. We also wish to thank Axel van de Walle whose input was helpful in testing the code.

VI. AUTHOR CONTRIBUTIONS

G. L. W. H. conceived the project, implemented the algorithm in Fortran 90, and wrote the first draft of the paper. The algorithm was developed jointly but key components were invented by R. W. F, who also implemented the algorithm in Maple as a check on the Fortran implementation. Both authors contributed to the manuscript.

VII. APPENDIX

Hermite Normal Form: If \( L \) is a lattice, with basis given by the columns of a square matrix \( A \), and \( S \) is a superlattice, then \( S \) will have basis \( AM \) where \( M \) is a square matrix of integers. Furthermore, all bases of \( S \) will have the form \( AMN \) where \( N \) is an integer matrix with determinant \( \pm 1 \). Thus, to find a canonical basis for \( S \), we may use elementary integer column operations on \( M \) to make it lower triangular, with positive entries down the diagonal. Furthermore, we can arrange that the lower triangular matrix \( H = MN \) have the property that every off-diagonal element is less than the diagonal element in its row. Such a matrix \( H \) is said to be in Hermite Normal Form, and is unique with respect to the matrix \( M \).

Thus, if the determinant of \( M \) is \( n \), then the number of superlattices \( S \) of \( L \) with index \( n \) is equal to the number of distinct HNF matrices with determinant \( n \). In 3 dimensions, that number is

\[
\sum_{d \mid n} d \sigma(d) = \prod_{i=1}^{k} \left( \frac{(p_i^{e_i+2}-1)(p_i^{e_i+1}-1)}{(p_i-1)^2(p_i+1)} \right),
\]

where \( n = \prod p_i^{e_i} \) is the prime factorization of \( n \).

Smith Normal Form: Using elementary integer row and column operations (adding or subtracting an integer multiple of one row or column to another, multiplying a row or column by \( \pm 1 \), or exchanging two rows or columns), we may reduce the integer matrix \( M \) to a diagonal matrix \( D \) with the following properties:

(i) Each diagonal entry of \( D \) divides the next one down.  
(ii) The product of the diagonal entries of \( D \) is the absolute value of the determinant of \( M \).

This is called the Smith Normal Form of \( M \). In the lattice case, where \( L \) is a lattice with basis \( A \), and \( S \) is a superlattice (subgroup) with basis \( AM \), then \( D \) describes the quotient group \( L/S \) as a direct sum of cyclic groups. The diagonal entries of \( D \) are the orders of the cyclic direct summands of the quotient group (as in the Fundamental Theorem of Finite Abelian Groups). For example, using the notation \( Z_n = \mathbb{Z}/n\mathbb{Z} \), if \( D = \begin{pmatrix} D_{11} & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & D_{33} \end{pmatrix} \), then

\[
L/S \cong G = \mathbb{Z}_{D_{11}} \oplus \mathbb{Z}_{D_{22}} \oplus \mathbb{Z}_{D_{33}}.
\]
FIG. 12: Two-dimensional superlattices of index $n = 8$. The first and third share the same SNF while the second has the trivial SNF, implying a purely cyclic quotient group. In contrast to the first and third superlattices, the lattice points in layers parallel to the superlattice edge (dotted line) must all have the same label.

A simple, two-dimensional example, may help to show how this affects our lattice labeling problem. Consider the three matrices (all in HNF form)

$$
\mathbb{H}_0 = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}, \quad \mathbb{H}_1 = \begin{pmatrix} 2 & 0 \\ 1 & 4 \end{pmatrix}, \quad \mathbb{H}_2 = \begin{pmatrix} 2 & 0 \\ 2 & 4 \end{pmatrix}.
$$

The matrices $\mathbb{H}_0$ and $\mathbb{H}_2$ both reduce to the SNF matrix

$$
\begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix},
$$

which corresponds to a quotient group which is Abelian, but not cyclic, while the middle matrix, $\mathbb{H}_1$, reduces to SNF matrix

$$
\begin{pmatrix} 1 & 0 \\ 0 & 8 \end{pmatrix},
$$

corresponding to the cyclic group of order 8. Thus, if we take $\mathbb{A}$ to be the identity matrix, so $L = \mathbb{Z}^2$, and let $S_i$ be the lattice with basis $\mathbb{H}_i$, then $L/S_0$ and $L/S_2$ are each isomorphic to the group $\mathbb{Z}_8 \oplus \mathbb{Z}_4$, while $L/S_1$ is isomorphic to the cyclic group of order 8.

The fact that the latter is cyclic means we can layer the parent lattice in such a way that each parallel layer consists of points which all must get the same label (see Fig. 12). We can arbitrarily label each layer passing through the interior points of the basis parallelogram, and label the rest of the lattice cyclically, as if labeling a one-dimensional lattice. The quotient group for $S_0$ or $S_2$ is not cyclic but can just as easily be used to determine equivalent labelings.

In general, SNF provides a natural homomorphism from the parent lattice $L$ onto the direct-sum group $G = \mathbb{Z}_{D_{11}} \oplus \mathbb{Z}_{D_{22}} \oplus \mathbb{Z}_{D_{33}}$, with kernel $S$. By the First Homomorphism Theorem, it effectively gives an isomorphism from $L/S$ to $G$. Since we do only elementary integer row and column operations, we may write $D = \mathbb{F} \mathbb{M} \mathbb{Q}$, where the transition matrices $\mathbb{P}$ and $\mathbb{Q}$ are integer matrices with determinant $\pm 1$. Note that $\mathbb{M} \mathbb{Q}$ is another basis for $S$, so an element $x \in S$ iff $\mathbb{M} \mathbb{Q} z = A_i^{-1} x$ for some integer column vector $z$, which is true if $\mathbb{D} z = \mathbb{P} A_i^{-1} x$.

So the map

$$
x \mapsto \mathbb{P} A_i^{-1} x \pmod{D_i},
$$

(meaning that each row of the resulting column matrix is reduced modulo the corresponding diagonal element of $D$) maps from $L$ into the direct sum group $G$, with its kernel being the superlattice $S$.

As for computing the SNF of a matrix, there are special algorithms designed to compute it efficiently when $\mathbb{M}$ is very large but the simplest algorithm, effective for small (e.g., $3 \times 3$) matrices, is basically an extension of Euclid’s algorithm for finding the greatest common divisor of two numbers. One subtructs multiples of elements in the matrix from other elements in the same row or column (using column or row operations respectively) until the greatest common divisor of all the elements of $\mathbb{M}$ is exposed. That element is then moved to the upper left corner of the matrix and used to zero out all other elements in the first row and in the first column. Then one applies the same algorithm to the 2 by 2 submatrix in the lower right. Thus, in particular, the upper left entry in $D$ is always the greatest common divisor of all the entries in $\mathbb{M}$.

Note that the number of $3 \times 3$ SNF matrices with determinant $n$ is given by $\prod P_3(c_i)$, where $n = \prod p_i^{c_i}$ (the prime factorization) and $P_3(k)$ is the number of partitions of an integer $k$ using at most 3 summands.20

---

2 In the mathematical literature, and in some of the crystallography literature, these derivative lattices are referred to as sublattices. Although this nomenclature is more correct from a mathematical/group theory point of view, we follow the nomenclature typically seen in the physics literature where a lattice (or a structure) whose volume is larger than that of the parent is referred as a superlattice (or a superstructure).
7 We recently found a body of work by M. Hosoya that is related to our enumeration problem. Hosoya’s algorithm enumerates all possible crystal structures of any parent lattice of which derivative structures are a subset. Though this approach is not useful for the normal applications of

---

[^2]: In the mathematical literature, and in some of the crystallography literature, these derivative lattices are referred to as sublattices. Although this nomenclature is more correct from a mathematical/group theory point of view, we follow the nomenclature typically seen in the physics literature where a lattice (or a structure) whose volume is larger than that of the parent is referred as a superlattice (or a superstructure).
[^7]: We recently found a body of work by M. Hosoya that is related to our enumeration problem. Hosoya’s algorithm enumerates all possible crystal structures of any parent lattice of which derivative structures are a subset. Though this approach is not useful for the normal applications of
In some applications it is useful to include the incomplete labelings. For example, in a ternary alloy, the incomplete labelings represent the binary structures on the three edges of the phase diagram. Removal of the incomplete labelings can be disabled in the included implementation of the algorithm. See Ref. 13.


M. Baake, Mathematics of Long-Range Aperiodic Order (Kluwer Academic, 1997), chap. Solution of coincidence problem in dimensions $d \leq 4$, pp. 9–44.


An analytical check on the counting for these kinds of duplicates can be found in Rutherford’s work.

In three dimensions, each member of the quotient group formally has three components, even though in many cases only one is non-trivial. For example, if the diagonal elements of the SNF of a superlattice are (1, 1, 4), then the 4 members of the image group are (0,0,0), (0,0,1), (0,0,2), and (0,0,3). On the other hand, if the diagonal elements of the SNF are (1,2,2), then the 4 members of the image group are (0,0,0), (0,0,1), (0,1,0), and (0,1,1).


The partition idea was introduced into the crystallographic literature in Ref. 32.

