A robust algorithm for \(k\)-point grid generation and symmetry reduction

Gus L. W. Hart, Jeremy J. Jorgensen, and Wiley S. Morgan
Dept. of Physics and Astronomy, Brigham Young University, Provo Utah 84602 USA

Rodney W. Forcade
Dept. of Mathematics, Brigham Young University, Provo Utah 84602 USA

We develop an algorithm for computing generalized regular \(k\)-point grids\(^1,2\) and a related algorithm for symmetry-reducing a \(k\)-point grid to its symmetrically distinct points. The algorithm exploits the connection between integer matrices and symmetric groups, which leads to a computational complexity that is linear with the number of \(k\)-points, rather than the standard quadratic convergence of most electronic structure codes. The favorable scaling means that \(k\)-point grids of \(\sim 10^6\) points can be generated and reduced quicker and with lower memory requirements than current DFT codes. More importantly, the integer nature of the algorithm eliminates potential finite precision problems. An implementation of the algorithm is available as open source software. This algorithm is applicable to any numerical problem that requires a very dense, uniform sampling or that can benefit from symmetry reduction.

I. INTRODUCTION

Codes that simulate materials using density functional theory (DFT) create uniform grids over the Brillouin zone in order to calculate the total electronic energy, among other material properties. The total electronic energy is calculated by numerically integrating the occupied regions of the electronic bands. For metallic systems, there exist surfaces of discontinuities at the boundary between occupied and unoccupied states, collectively known as the Fermi surface. These discontinuities cause the accuracy in the calculation of the total electronic energy to converge extremely slowly and erratically with respect to grid density. This is demonstrated in Fig. 1 where we compare the convergence of an insulator with a metal.

The poor convergence of the electronic energy means that DFT codes must generate extremely dense grids for a target accuracy of several meV/atom\(^2\). It is common practice to increase computational efficiency by using the crystal’s symmetry to reduce the number of electronic structure evaluations: grid points that are symmetrically equivalent have the same eigenenergies, so only one has to be evaluated.

In this brief report, an algorithm for generating, and subsequently symmetry-reducing, \(k\)-point grids is explained. This algorithm relies heavily on concepts used in the derivative structure enumeration algorithm of the enumlib code.\(^3\) The algorithm has been implemented in an open-source code available at \(https://github.com/msg-byu/kgridGen\). The algorithm has been incorporated in version 6 of the VASP code.\(^5\)

In most DFT codes, even for very dense grids, the setup and symmetry reduction of the grid takes a few seconds at most. Our motivation for an improved algorithm, despite the speed of current routines, is two-fold: 1) eliminate (or at least greatly reduce) the probability of incorrect symmetry reduction as the result of finite precision errors (the danger of these increases as the density of the integration grid increases), and 2) enable an automatic grid-generation technique that allows us to scan over thousands of candidate grids, in a few seconds, to find one with the best possible symmetry reduction (in other words, to enable a \(k\)-point generation method in the same spirit as that of Ref. 1 but for which the grid generation can be done on-the-fly).

**FIG. 1.** Total energy error vs. \(k\)-point density for the cases of silicon and aluminum. Silicon does not have a Fermi surface so there is no discontinuity in the occupied bands; convergence is super-exponential. (See the discussion of example 1 in Ref. 3.) In contrast, the total energy of aluminum converges very slowly, and the convergence is quite erratic. For typical target accuracies in the total energy, around \(10^{-3}\) eV/atom, metals require 10–50 times more \(k\)-points than semiconductors.
II. GENERATING GRIDS

Every uniform sampling of a Brillouin zone can be expressed through the simple integer relationship

\[ \mathbb{R} = \mathbb{K} \mathbb{N}. \]  

In Eq. (1), \( \mathbb{R} \), \( \mathbb{K} \), and \( \mathbb{N} \) are \( 3 \times 3 \) matrices; the columns of \( \mathbb{R} \) are the reciprocal lattice vectors, the columns of \( \mathbb{K} \) are the \( k \)-point grid generating vectors, and \( \mathbb{N} \) is a non-singular integer matrix. Simply, \( \mathbb{N} \) describes the integer linear combination of vectors of \( \mathbb{K} \) that is equivalent to \( \mathbb{R} \). One example of the concept, illustrated in two dimensions, is shown in Fig. 2. We refer to the infinite lattice of points defined by integer linear combinations of \( \mathbb{R} \) as \( \mathbb{R} \), and the lattice of points defined by \( \mathbb{K} \) as the lattice \( \mathbb{K} \).

With no loss of generality, a new basis for the grid \( \mathbb{K} \) can be chosen (a different, but equivalent, \( \mathbb{K} \)) so that \( \mathbb{N} \) is a lower triangular matrix in Hermite normal form. (HNF; see Sec. II-A of Ref. 4 for a brief introduction to Hermite normal form. HNF is a lower-triangular canonical matrix form, where the non-zero, off-diagonal terms are non-negative and less than the diagonal entry in the same row. Code for generating the HNF (and SNF, see below), is available at https://github.com/msg-byu/symlib.)

The integration grid is the set of points of the lattice \( \mathbb{K} \) that lie inside one unit cell (one fundamental domain) of the reciprocal lattice \( \mathbb{R} \). We refer to this finite subset of \( \mathbb{K} \) as \( \mathbb{K}_n \) (See Fig. 2; black dots are \( K \), dots inside the blue parallelogram comprise \( \mathbb{K}_n \)). The number of points that lie within one unit cell of \( \mathbb{R} \) is given by \( |\det(\mathbb{N})| = n \).

How then does one generate these \( n \) points? If \( \mathbb{N} \) is in HNF, then the diagonal elements of \( \mathbb{N} \) are three integers (call them \( a \), \( c \), and \( f \)) such that \( a \cdot c \cdot f = n \). A set of \( n \) translationally distinct points of the lattice \( \mathbb{K} \) can be generated by taking integer linear combinations of the basis of \( \mathbb{K} \) (columns of the matrix \( \mathbb{K} \)):

\[ \mathbf{k} = p\mathbf{\kappa}_1 + q\mathbf{\kappa}_2 + r\mathbf{\kappa}_3, \]  

where \( p \), \( q \), and \( r \) are nonnegative integers such that

\[ 0 \leq p < a, \]
\[ 0 \leq q < c, \]  
\[ 0 \leq r < f, \]

and \( \mathbf{\kappa}_1 \), \( \mathbf{\kappa}_2 \), and \( \mathbf{\kappa}_3 \) are the columns of \( \mathbb{K} \). The \( n \) points generated this way will not generally lie inside the same unit cell, but they can be translated into the same cell by expressing them in “lattice coordinates” (fractions of the columns of \( \mathbb{R} \), instead of Cartesian coordinates) and then modding the coordinates by 1 so that they all lie within the range \([0, 1)\). This is illustrated by the dashed arrow in Fig. 3.

Expressed as fractions of the lattice vectors of \( \mathbb{R} \), these four points are:

\[ \mathbf{\kappa}_1 = (0, 0) \]
\[ \mathbf{\kappa}_2 = \left( \frac{1}{2}, 0 \right) \]
\[ \mathbf{\kappa}_3 = \left( -\frac{1}{3}, \frac{1}{2} \right) \mod 1 \rightarrow \left( \frac{2}{3}, \frac{1}{2} \right) \]
\[ \mathbf{\kappa}_4 = \left( 1, \frac{1}{2} \right) \]

Initially, \( \mathbf{\kappa}_3 \) is not in the same unit cell as the other three points; its first coordinate is not between 0 and

\[ \begin{pmatrix} -3 & 6 \\ 2 & 4 \end{pmatrix} = \begin{pmatrix} 0 & 3 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -1 & 2 \end{pmatrix} \]
be as follows: For each point \( (\mathcal{O}(N_k) \text{ loop}) \), compare all rotations of that point (a loop of \( \mathcal{O}(|G|) \)) to all other points (another \( \mathcal{O}(N_k) \text{ loop} \)) to find a match; for a total computational complexity of \( \mathcal{O}(N_k^2 N_G) \) (where \( N_G \) is the number of rotation and reflection symmetries). In pseudocode, this algorithm would be the following:

**Algorithm 1**

```plaintext
uniqueCount ← 0
for each \( k_i \in K_n \)
    orbitLength; ← true
    for each \( k_j \in K_n \)
        for each \( g \in G \)
            if \( k_j = g \cdot k_i \)
                orbitLength; ++
                unique ← false
        end
    end
    if (unique)
        uniqueCount ++
        add \( k_i \) to list of unique points
end
```

\( N_G \) will never be larger than 48, but \( N_k \) may be as large as \( 50^3 \) for extremely dense grids, so the \( N_k^2 \) complexity of this naive approach is undesirable. But using group theory concepts (see the Appendix for details), we can construct a hash table for the points that reduces the complexity from \( \mathcal{O}(N_k^2 N_G) \) to \( \mathcal{O}(N_k N_G) \) by eliminating the \( k_i \) loop in the above algorithm. The hash table makes a one-to-one association between the ordinal counter (i.e., the index) for each point and its coordinates.

The three coefficients \( p, q, r \) in Eq. (2) can be conceptualized as the three “digits” of a 3-digit mixed-radix number \( pqr \) or as the three numerals shown on an odometer with three wheels. The ranges of the values are \( 0 \leq p < d_1 \), \( 0 \leq q < d_2 \), and \( 0 \leq r < d_3 \), where \( d_1, d_2, d_3 \) are the “sizes” of the wheels, or in other words, the base of each digit. Then the mixed-radix number is converted to base 10 as

\[
x = p \cdot d_2 \cdot d_1 + q \cdot d_1 + r. \tag{3}
\]

The total number of possible readings of the odometer is \( d_3 \cdot d_2 \cdot d_1 \). So it must be the case that the number of \( k \)-points in the cell is \( n = d_3 \cdot d_2 \cdot d_1 \). Each reading on the odometer is a distinct point of the \( n \) points that are contained in the reciprocal cell. Via Eq. (3) it is simple to convert a point given in “lattice coordinates” as \( (p, q, r) \) to a base-10 number \( x \). The concept of the hash table is to use this base-10 representation as the index in the hash table. Without the hash table, comparing two points is an \( \mathcal{O}(N_k) \) search because one point must be compared to every other point in the list to check for equivalency. But with the hash function, mapping \( (p, q, r) \) to \( x \) only requires a single comparison.

It is not generally the case that the coefficients \( p, q, r \) for every interior point of the unit cell obey conditions:

![Diagram](image-url)
it is possible to transform the matrix $N$ in Eq. (1) into its Smith Normal Form, $D = A N B$. By elementary row and column operations, represented by unimodular matrices $A$ and $B$, it is possible to transform $N$ into a diagonal matrix $D$, where each diagonal element divides the ones below it:

$$d_{11} | d_{22} | d_{33},$$

and

$$d_{11} \cdot d_{22} \cdot d_{33} = n = |N|$$

(the notation $i|j$ means that $i$ is divisible by $j$). As explained in the appendix (Sec. VII), when $N$ is expressed in Smith normal form (SNF) and the interior points of the reciprocal cell are expressed as linear combinations of the grid generating vectors $K$, then the coordinates (coefficients) of the interior points will obey Eq. (2). When these conditions are met, the hashing algorithm discussed above (in particular, Eq. (3)) becomes possible. This enables the $O(N_k)$ algorithm.

**Algorithm 2**

```plaintext```
uniqueCount ← 0
hashTable[:] ← − 0
First[: ] ← − 0
Wt[: ] ← − 0
for each $k_i \in K$
    indx ← $K^{-1} A D \cdot k_i$
    if hashTable[indx] ≠ 0 cycle #this #point and all its symmetry #equivalent points has already been #indexed
    uniqueCount++
    hashtable[indx] ← uniqueCount
    First[uniqueCount] ← i
    Wt[uniqueCount] ← 1
# Now mark all equivalent points
for each $g \in G$
    $k_{rot} ← \frac{g \cdot k_i}{K^{-1} A D \cdot k_{rot}}$
    indx ← $K^{-1} A D \cdot k_{rot}$
    if hashtable[indx] == 0
        hashtable[indx] ← uniqueCount
        Wt[uniqueCount]++
```

At the end of the algorithm, the array $Wt$ will be a list of the weights for each symmetrically unique point, and the index of each unique point in $K$ will be stored in the array $First$.

### IV. MOVING POINTS INTO THE FIRST BRILLOUIN ZONE

For accurate DFT calculations, it is best if the energy eigenvalues (electronic bands) are evaluated at $k$-points inside the first Brillouin zone, so our algorithm includes a step that finds the translationally equivalent grid points in the Brillouin zone. (In principle, the electronic structure should be the same in every unit cell, but numerically the periodicity of the electronic bands is only approximate, becoming less accurate for $k$-points in unit cells farther from the origin.)

The first Brillouin zone of the reciprocal lattice is simply the Voronoi cell centered at the origin—all $k$-points in the first Brillouin zone are closer to the origin than to any other lattice point. Conceptually, an algorithm for translating a $k$-point of the integration grid into the first zone merely requires one to look at all translationally equivalent “cousins” of the $k$-point and select the one closest to the origin. The number of translationally equivalent cousins is countably infinite, of course, so in practice, the set of cousins must be limited only to those near the origin.

How can we select a set of cousins near the origin that is guaranteed to include the closest cousin? The key idea is illustrated in two-dimensions in Fig. 5. In three-dimensions, if the basis vectors of the reciprocal unit cell are as short as possible (the so-called Minkowski-reduced basis), then the eight unit cells that all share a vertex at the origin must contain every point that is closer to the origin than any other lattice point. In other words, the boundary of this union of eight cells is guaranteed to circumscribe the first Brillouin zone (i.e., the Voronoi cell containing the origin). A proof of this “8 cells” conjecture is given in the appendix.

The steps for moving $k$-points into the Brillouin zone are as follows:

1. Minkowski-reduce the reciprocal unit cell (i.e., find the basis with the shortest basis vectors)
2. For each $k$-point in the reduced grid, find the translation-equivalent cousin in each of the eight unit cells that have a vertex at the origin.
3. From these eight cousins, select the one closest to the origin.

![FIG. 5. In 2D, the Brillouin zone, shown in blue, is a subset of the union of 4 basis cells around the origin, shown in red, when the basis vectors are chosen to be as short as possible (the so-called Minkowski basis). If the basis is not Minkowski reduced, regions of the Brillouin zone may lie outside the union of the 4 basis cells, which is depicted by the cell in green.](image-url)
V. CONCLUSION

We have developed an algorithm for generating generalized regular \( k \)-point grids, symmetry reducing those grids, and mapping each point of the reduced grid into the first Brillouin zone. This algorithm scales linearly with the total number of \( k \)-points. Mapping the grid to the Brillouin zone benefits from a proof that limits the search for translationally equivalent \( k \)-points to the eight unit cells having a vertex at the origin. For dense grids, the application of the algorithm can reduce time and memory requirements for DFT calculations. The algorithm is also useful because it relies primarily on integer-based operations, making it more robust than typical floating point-based algorithms that are prone to finite precision errors. This algorithm has been incorporated into version 6 of VASP.

VI. ACKNOWLEDGMENTS

We are grateful to Martijn Marsmann who found a bug in an early version of the code that spoiled the \( O(N_k) \) scaling and who implemented the algorithm in VASP, version 6. Tim Mueller also contributed by pointing out a simpler approach to generating all the \( k \)-points interior to the unit cell of \( R \) than we originally proposed. GLWH, WSM, and JJJ are grateful for financial support from the Office of Naval Research (MURI N00014-13-1-0635).

VII. APPENDIX

A. Proof limiting Brillouin zone location

Given a point \( x \) in the space, we will use the term cousin for a point \( x' \) which differs from \( x \) by an element of the lattice—i.e., a coset representative or a lattice-translation equivalent point.

Let \( R \) be a basis. Let \( U_R \) denote the union of \( 2^d \) basis cells around the origin—the set of points which are expressible in terms of the basis \( R \) with all coefficients having absolute value \( < 1 \). Let \( V \) denote the Voronoi cell (Brillouin zone)—the set of all points in the space which are closer to the origin than to any other lattice point. Note that \( U_R \) depends on the basis \( R \), but \( V \) depends only on the lattice. Note also that both \( U_R \) and \( V \) are convex sets.

We claim (in two and three dimensions) that if \( R \) is a Minkowski basis, then \( V \subseteq U_R \). We shall argue by contrapositive— if \( V \nsubseteq U_R \), then the basis is not Minkowski reduced.

If \( V \nsubseteq U_R \) then \( V \) must intersect the boundary of \( U_R \), so there exist points on the boundary of \( U_R \) which are closer to the origin than to any other lattice points. Equivalently, those points are closer to the origin than any of their cousins.

Note that among the cousins of any point on the boundary of \( U_R \), there is always a closest to the origin. But usually points on the boundary will have closer cousins in the interior. But if \( V \nsubseteq U_R \) there must be points on the boundary which have no closer cousins in the interior of \( U_R \). In other words, there are points (at least one) on the boundary such that \( all \) of its cousins in the interior of \( U_R \) are farther from the origin.

1. 2D argument

Let \( \vec{r}_1 \) and \( \vec{r}_2 \) be basis elements of \( R \). Assuming that \( V \nsubseteq U_R \) there must be a point \( x \) on the boundary of \( U_R \) whose cousins are all farther from the origin than \( x \).

Without loss of generality (re-label the basis if necessary), we may express one of the bounding edges of \( U_R \) as \( x = \vec{r}_1 + \lambda \vec{r}_2 \) where \( \lambda \in [-1, 1) \). One of its interior cousins is \( x' = \lambda \vec{r}_2 \), which is illustrated in Fig. 5. We have (since \( x' \) must be farther from the origin)

\[
\begin{align*}
  x^2 &< x'^2 \\
  (\vec{r}_1 + \lambda \vec{r}_2)^2 &< (\lambda \vec{r}_2)^2 \\
  \vec{r}_1^2 &+ 2\lambda \vec{r}_1 \cdot \vec{r}_2 + \lambda^2 \vec{r}_2^2 &< \lambda^2 \vec{r}_2^2 \\
  \vec{r}_1^2 &< -2\lambda \vec{r}_1 \cdot \vec{r}_2
\end{align*}
\]

Since the expression on the left-hand side is greater than zero, the expression on the right-hand side must be also and taking the absolute value of both sides does not change the inequality:

\[
|\vec{r}_1^2| < -2\lambda |\vec{r}_1 \cdot \vec{r}_2| \implies |\vec{r}_1|^2 < 2|\lambda||\vec{r}_1 \cdot \vec{r}_2|.
\]

Considering the worst case scenario of \( \lambda = 1 \) gives

\[
\frac{|\vec{r}_1|}{2} < \frac{|\vec{r}_1 \cdot \vec{r}_2|}{|\vec{r}_1|}, \quad (5)
\]

which violates the condition of a Minkowski basis \( |\vec{r}_1 \cdot \vec{r}_2|/|\vec{r}_1| < |\vec{r}_1|/2 \). The remaining three boundaries are similar to the one just considered, the only differences being permutations of the basis elements \( \vec{r}_1 \) and \( \vec{r}_2 \) and possibly changes of sign. When applying the same reasoning to the other edges we arrive at the same contradiction. Hence, the points on the boundary of \( U_R \) are closer to the origin than interior cousins, \( V \nsubseteq U_R \), only when the basis \( R \) is not Minkowski reduced. If \( R \) is Minkowski reduced, all points on the boundary of \( U_R \) have interior cousins that lie closer to the origin and \( V \subseteq U_R \).

2. 3D argument

Let \( \vec{r}_1 \), \( \vec{r}_2 \), and \( \vec{r}_3 \) be the basis elements of \( R \), and suppose (relabeling the basis vectors if necessary) that \( x = \vec{r}_1 + \lambda \vec{r}_2 + \delta \vec{r}_3 \) (where \( \lambda \) and \( \delta \) are elements of \([-1, 1]\)) is a point on the boundary of \( U_R \) which is closer to the origin than any of its interior cousins.
One of those cousins is a plane through the origin $x' = \lambda \vec{r}_2 + \delta \vec{r}_3$. The boundary and cousin planes are shown in Fig. 7. Thus

\[
x^2 < x'^2
\]
\[
(\vec{r}_1 + \lambda \vec{r}_2 + \delta \vec{r}_3)^2 < (\lambda \vec{r}_2 + \delta \vec{r}_3)^2.
\]

Simplifying this expression gives

\[
\vec{r}_1^2 < -2\lambda \vec{r}_1 \cdot \vec{r}_2 - 2\delta \vec{r}_1.
\]

Since the expression on the left-hand side is greater than zero, the expression on the right-hand side must be also and taking the absolute value of both sides does not change the inequality:

\[
|\vec{r}_1|^2 < |\lambda \vec{r}_1 \cdot \vec{r}_2 - 2\delta \vec{r}_1 \cdot \vec{r}_3| \quad \Rightarrow \quad |\vec{r}_1|^2 < |\lambda \vec{r}_1 \cdot \vec{r}_2 + 2\delta \vec{r}_1 \cdot \vec{r}_3|
\]

Since using the triangle inequality makes it more likely that the inequality in Eq. 7 is true, we can use it to simplify the right-hand side:

\[
|\vec{r}_1|^2 < 2\lambda |\vec{r}_1 \cdot \vec{r}_2| + 2\delta |\vec{r}_1 \cdot \vec{r}_3|
\]

\[
|\vec{r}_1|^2 < 2|\lambda| |\vec{r}_1 \cdot \vec{r}_2| + 2|\delta| |\vec{r}_1 \cdot \vec{r}_3|
\]

(8)

Combining Eqns. 8 and 9 gives

\[
|\vec{r}_1|^2 < (|\lambda| + |\lambda - 1|)|\vec{r}_1 \cdot \vec{r}_2| + (|\delta| + |\delta - 1|)|\vec{r}_1 \cdot \vec{r}_3|
\]

\[
|\vec{r}_1|^2 < \frac{|\vec{r}_1 \cdot \vec{r}_2| + |\vec{r}_1 \cdot \vec{r}_3|}{|\vec{r}_1|}
\]

(10)

Assuming \{\vec{r}_1, \vec{r}_2, \vec{r}_3\} forms a Minkowski basis, and plugging in the largest possible values for all quantities under this assumption on the right-hand side of Eqn. 10 gives the contradiction $|\vec{r}_1| < |\vec{r}_1|$. The remaining seven bounding planes are similar to the one just considered, the only differences being permutations of the basis elements $\vec{r}_1, \vec{r}_2$, and $\vec{r}_3$ and changes of sign. We arrive at the same contradiction when applying the same reasoning to the other planes. Hence, the points on the boundary of $U_R$ are closer to the origin than interior cousins, $V \not\subseteq U_R$, only when the basis $R$ is not Minkowski reduced. If $R$ is Minkowski reduced, all points on the boundary of $U_R$ have interior cousins that lie closer to the origin and $V \subseteq U_R$.

**B. Groups, Matrices, and Lattices in Smith Normal Form.**

The discussion below is limited to three-dimensions though the arguments easily generalize to higher dimensions. The purpose of the discussion below is to help the reader make the connection between groups and integer...
matrices. The Smith Normal Form is a key concept to make this connection.

In this discussion, we show that we can associate a single, finite group with the lattice sites within one tile (i.e., one unit cell) of a superlattice. In our application, this tile is the unit cell of the grid generating vectors and the superlattice is the reciprocal cell. The association between the group and the lattice sites is a homomorphism that maps each lattice site to an element of the group. If two points are translationally equivalent (same site but in two different tiles) they will map to the same element of the group. This homomorphism is the key ingredient to constructing the hash function (see Eq. 3) that enables a perfect hash table where points are listed consecutively, from 1 to \(N\). In what follows, we explain in detail how this association is made, i.e., we detail how one finds this homomorphism.

1. Groups in Smith Normal Form

Begin with the simplest case. Let \(N\) be a non-singular \(3 \times 3\) matrix of integers. Its columns represent the basis for a subgroup \(L_N\) of the group \(\mathbb{Z}^3\) (where \(\mathbb{Z}\) is the set of all integers, and the group operation is addition). The two lattices whose symmetries are represented by these two groups are the “simple cubic” lattice of all points with all integer coordinates and its superlattice whose basis is given by the columns of \(N\). Since \(\mathbb{Z}^3\) and its subgroups are Abelian, we know that all the subgroups are normal so there exists a quotient group \(G = \mathbb{Z}^3/L_N\), and that group is finite.

Note that the cosets which form the elements of that quotient group are simply the distinct translates of the lattice \(L_N\) within \(\mathbb{Z}^3\). In fact, each coset has exactly one representative in each unit cell, so the order of \(G\) is equal to the volume of a unit cell (the absolute value of the determinant of \(N\)). Since the quotient group \(G\) is finite, and Abelian, it must be a direct sum of cyclic groups (by the Fundamental Theorem of Finite Abelian Groups).

One canonical form for direct sums of groups is called Smith Normal Form, where the direct summands are ordered so that each summand divides the next. In other words, \(G \cong \mathbb{Z}_{m_1} \oplus \mathbb{Z}_{m_2} \oplus \cdots \oplus \mathbb{Z}_{m_k}\) where \(m_1|m_2|\cdots|m_{k-1}|m_k\) and (of course) \(\prod m_i = |G|\). Any finite Abelian group can be uniquely written in this form. (Isomorphic groups will yield the same “invariant factors” \(m_1, m_2, \ldots, m_k\) when written in this form.)

Note that, since \(G = \mathbb{Z}^3/L_N\), there must be a homomorphism from \(\mathbb{Z}^3\) onto \(G\), having \(L_N\) as its kernel. In other words, \(L_N = \{ p \in \mathbb{Z}^3 : \psi(p) = 0 \}\). Our task is to find the direct-sum representation of the quotient group \(\mathbb{Z}^3/L_N\), and also to find the homomorphism \(\psi\) which maps the points of \(\mathbb{Z}^3\) onto the group (in such a way that \(\psi(p) = 0\) iff \(p \in L_N\)). This allows us to work with the elements of the group as proxies for the \(k\)-points inside the reciprocal cell.

2. Matrices in Smith Normal Form

There is a useful connection between the SNF for Abelian groups and the SNF for integer matrices. As the reader may infer, the SNF form of the basis matrix \(N\) effectively tells one how to represent the quotient group \(\mathbb{Z}^3/L_N\) as a direct sum of cyclic groups in Smith Normal Form, and, as shown in the following section, the row operations used to create the SNF of \(N\) give the homomorphism \(\psi\) suggested above.

3. The connection between SNF groups and SNF matrices

In the matrix case, since the operations are elementary row and column operations, we have \(D = A'N'B\) where \(A\) and \(B\) are integer matrices with determinant \(\pm 1\) representing the accumulated row operations and column operations respectively. The matrix \(D\) is completely determined by \(N\), but the matrices \(A\) and \(B\) depend on the algorithm used to arrive at the Smith Normal Form of \(N\). A different implementation might yield \(D = A'N'B'\) (same \(N\) and same \(D\), but different \(A\) and \(B\)).

Note that, since \(B\) represents elementary column operations, the product \(N'B\) simply represents a change of basis from \(N\) to a new basis \(N' = N'B\). In other words, the columns of \(N'\) are still a basis for \(L_N\). But the new basis has the property that \(AN' = D\). That means that every element \(\bar{w} = N'\bar{z}\) of \(L_N\) (where \(\bar{z}\) is some element of \(\mathbb{Z}^3\)) will satisfy the equation \(A\bar{w} = D\bar{z}\).

Thus we have shown \(\bar{w} \in L_N\) iff \((A\bar{w})^* = (0, 0, 0)\) (the zero-element in the group \(G_0 = \mathbb{Z}_{D_{11}} \oplus \mathbb{Z}_{D_{22}} \oplus \mathbb{Z}_{D_{33}}\)).

To put it another way, define \* to be the operation that maps \(\bar{z} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}\) in \(\mathbb{Z}^3\) to \(\bar{z}^* = \begin{pmatrix} x_1 \text{ (mod } D_{11}) \\ x_2 \text{ (mod } D_{22}) \\ x_3 \text{ (mod } D_{33}) \end{pmatrix}\).

Then we have shown \(\bar{w} \in L_N\) iff \((A\bar{w})^* = (0, 0, 0)\) (the zero-element in the group \(G_0 = \mathbb{Z}_{D_{11}} \oplus \mathbb{Z}_{D_{22}} \oplus \mathbb{Z}_{D_{33}}\)).

That suggests we let \(\psi(\bar{w}) = (A\bar{w})^*\), a homomorphism from \(\mathbb{Z}^3\) onto the direct-sum \(G_0\). Then, since that homomorphism is easily shown to be onto, and its kernel is \(L_N\), we see (by the First Isomorphism Theorem of group theory) that \(G_0 \cong \mathbb{Z}^3/L_N\), and \(\psi\) is precisely the homomorphism we sought.

Thus we have connected the two versions of SNF. The matrix algorithm provides the SNF description of the quotient group by the diagonal entries in \(D\), and the transition matrix \(A\) provides the homomorphism which maps the parent lattice onto the group.

An example. Let \(N = \begin{pmatrix} 1 & 2 & -1 \\ 1 & 4 & -3 \\ 0 & 2 & 4 \end{pmatrix}\). This describes a
lattice $\mathcal{L}_N$ which contains the points $\tilde{p}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$, $\tilde{p}_2 = \begin{pmatrix} 2 \\ -4 \\ 2 \end{pmatrix}$, and $\tilde{p}_3 = \begin{pmatrix} -1 \\ -3 \\ 4 \end{pmatrix}$, and all the points which are integer linear combinations of those three points. The matrix $\mathbb{N}$ has determinant 12, which must be the volume of each lattice tile—and it is also the order of the quotient group $\mathbb{Z}^3/\mathcal{L}_N$.

Using the SNF algorithm to diagonalize this basis matrix, we find $\mathbb{D} = \mathbb{N} \mathbb{B}$ where $\mathbb{D} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 6 \end{pmatrix}$, with

$$\mathbb{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -2 \end{pmatrix} \text{ and } \mathbb{B} = \begin{pmatrix} 1 & 7 & 11 \\ 0 & -1 & -2 \\ 0 & 1 & 1 \end{pmatrix}.$$ 

Thus we now know that the quotient group is $G = \mathbb{Z}^3/\mathcal{L}_N \cong \mathbb{Z}_1 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_6 \cong \mathbb{Z}_2 \oplus \mathbb{Z}_6$.

Further, from the matrix $\mathbb{A}$, we may obtain the homomorphism projecting $\mathbb{Z}^3$ onto the quotient group, with kernel $\mathcal{L}_N$. If $\bar{w} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ then $\mathbb{A}\bar{w} = \begin{pmatrix} y \\ z \\ x - y - 2z \end{pmatrix}$ and thus

$$\psi(\bar{w}) = (\mathbb{A}\bar{w})^* = \begin{pmatrix} y \pmod{1} \\ z \pmod{2} \\ x - y - 2z \pmod{6} \end{pmatrix} = \begin{pmatrix} z \pmod{2} \\ x + 5y + 4z \pmod{6} \end{pmatrix}$$

(noting that anything mod 1 is zero).

Note that this homomorphism provides a different, but convenient, way to describe the superlattice. Since $\mathcal{L}_N$ is the kernel of $\psi$, it is comprised of the points $(x, y, z) \in \mathbb{Z}^3$ which satisfy the simultaneous congruences $z \equiv 0 \pmod{2}$ and $x + 5y + 4z \equiv 0 \pmod{6}$. We note that all three basis points $p_1, p_2, p_3$ satisfy these congruences, and thus so will all their integer linear combinations (all points in $\mathcal{L}_N$).

**Algorithmic variation.** In the example we computed above, a different application of the SNF matrix algorithm, with the same $\mathbb{N}$, might have yielded the same diagonal matrix $\mathbb{D} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 6 \end{pmatrix}$, but different

$$\tilde{p}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \tilde{p}_2 = \begin{pmatrix} 5 \\ 3 \\ 1 \end{pmatrix}, \text{ and } \tilde{p}_3 = \begin{pmatrix} -1 \\ 2 \\ 1 \end{pmatrix},$$

which would change the homomorphism to $(x, y, z) \mapsto (-5x + 3y + z \pmod{2}, -2x + 2y + z \pmod{6}) = (x + y + z \pmod{2}, 4x + 2y + z \pmod{6})$. The new homomorphism is different, since $(1, 0, 1) \mapsto (0, 5)$ now, where previously $(1, 2, 3) \mapsto (1, 5)$ (for example), but the kernel is the same. In fact the two homomorphisms are related via an automorphism of the group $G$.

### 4. Non-integer lattices

Now, what about the more complicated situation, where $\mathbb{N}$ represents a (possibly INN) matrix describing the change from some lattice other than the simple integer lattice $\mathbb{Z}^3$ to one of its subgroups (superlattice)?

Then we have a basis $\mathcal{V}$ and lattice $\mathcal{L}_V$ and a basis $\mathcal{W} = \mathbb{V} \mathbb{N}$ for a (super) lattice $\mathcal{L}_W$. Again, the quotient group $G = \mathcal{L}_V/\mathcal{L}_W$ is Abelian of order $\det(\mathbb{N})$. Again, $G$ is a direct sum of cyclic groups corresponding to the diagonal entries of $\mathbb{D} = \mathbb{A}\mathbb{B}$ (where $\mathbb{D}$ is the SNF of $\mathbb{N}$).

The only difference here is that the homomorphism $\psi$ provided by $\mathbb{A}$ must depend on the basis $\mathcal{V}$ (which might even be irrational). Every point in $\mathcal{L}_V$ has the form $\bar{x} = \mathbb{V}\bar{w}$ where $\bar{w}$ is a column of integers. Then $\psi(\bar{x}) = \mathbb{A}\bar{w}$ (modded by the corresponding entries from $\mathbb{D} = \mathbb{A}\mathbb{B}$). We could write it as $\psi(\bar{x}) = (\mathbb{A}\mathbb{V}^{-1}\bar{x})^*$ (with the entries appropriately modularly reduced and transposed to a horizontal vector).

### 5. Example: general (non-integer) lattices

Suppose $\mathcal{L}_V$ is the lattice defined by (columns of) the basis matrix $\mathcal{V} = \begin{pmatrix} 1 & 1/2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$, and $\mathcal{L}_W$ is the subgroup lattice defined by the basis matrix $\mathcal{W} = \mathbb{V}\mathbb{N}$ where $\mathbb{N} = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 2 & 2 \end{pmatrix}$. In other words, one basis for $\mathcal{L}_W$ is given by

the columns of $\mathcal{W} = \begin{pmatrix} 5 & 3 & 3 \\ \sqrt{3} & \sqrt{3} & \sqrt{3} \end{pmatrix}$.

Reducing $\mathbb{N}$ to SNF yields

$$\mathbb{D} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 1 & -1 & -1 \\ -6 & 4 & 5 \end{pmatrix} \mathbb{N} \begin{pmatrix} -2 & -3 & -2 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Thus our quotient group is $G = \mathcal{L}_V/\mathcal{L}_W \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4$ and $\mathbb{A} = \begin{pmatrix} 1 & 0 & -1 \\ 1 & -1 & -1 \\ -6 & 4 & 5 \end{pmatrix}$ so

$$\mathbb{A}\mathbb{V}^{-1} = \begin{pmatrix} 1 & -\sqrt{3}/2 & -1/2 \\ 1 & -\sqrt{3} & -1/2 \\ -6 & 14\sqrt{3}/3 & 5/2 \end{pmatrix}.$$
which provides our homomorphism \( \psi(\vec{x}) = (A^V_1)^* \) from \( L_V \) onto \( G \).

If we let \( \vec{x} = \begin{pmatrix} 2 \\ \sqrt{3}/2 \end{pmatrix} \) which is an element of \( L_V \) but not of \( L_W \), then \( A^V_1 \vec{x} = \begin{pmatrix} 0 \\ -2/7 \end{pmatrix} \) and \( \psi(\vec{x}) = (0, 0, 3) \in G \) (after modding the elements by 2, 2 and 4 respectively).

On the other hand, if we let \( \vec{x} = \begin{pmatrix} 7 \\ \sqrt{3}/8 \end{pmatrix} \) which is an element of \( L_W \), then \( A^V_1 \vec{x} = \begin{pmatrix} 2 \\ 0 \\ -8 \end{pmatrix} \) and so \( \psi(\vec{x}) = (0, 0, 0) \), and \( \vec{x} \) is in the group.

By this function \( \psi \), the elements of \( L_V \) are all mapped to elements of the group \( G \) and, in particular, the elements of \( L_W \) are mapped to the zero element of the group. Stated in terms of the cosets, the entire set \( L_W \) is mapped to the zero element of the group, and each of the distinct translates of \( L_W \) (within \( L_V \)) gets mapped to a different element of the group. We might think of this as decorating or labeling the elements of \( L_V \) in a periodic manner, using \( L_W \) to define the period, and using the elements of the group \( G \) as the labels.

\[ \begin{align*}
1 & \text{ P. Wisessa, K. A. McGill, and T. Mueller, Phys. Rev. B 93, 155109 (2016).} \\
2 & \text{W. S. Morgan, J. J. Jorgensen, B. C. Hess, and G. L. Hart, Comput. Mater. Sci 153, 424 (2018).} \\
3 & \text{J. A. C. Weideman, The American mathematical monthly 109, 21 (2002).} \\
4 & \text{G. L. Hart and R. W. Forcade, Phys. Rev. B 77, 224115 (2008).} \\
5 & \text{G. Kresse and J. Hafner, Phys Rev B 47, 558 (1993).} \\
6 & \text{H. J. Monkhorst and J. D. Pack, Phys. Rev. B 13, 5188 (1976).} \\
7 & \text{If two points are translationally distinct, their difference cannot be an integer linear combination of the reciprocal cell vectors; that is, } \vec{k}_i - \vec{k}_j \neq n\vec{r}_1 + m\vec{r}_2 + \ell\vec{r}_3, \text{ for all integer values } n, m, \ell. \text{ (} \vec{r}_i \text{ are the columns of } \mathbb{R}). \\
8 & \text{In addition to the rotations/reflections of the crystal, inversion symmetry is also included. Even when the crystal does not have inversion symmetry, the electronic bands will.} \\
9 & \text{P. Q. Nguyen and D. Stehle, ACM Transactions on Algorithms (TALG) 5, 46 (2009), this paper explains how to Minkowski reduce a lattice basis.} \\
10 & \text{Our Fortran code for computing the Minkowski reduced basis is available at https://github.com/msg-byu/symlib.} \\
11 & \text{In the mathematical literature, and in some of the crystallography literature, these “superlattices” are referred to as sublattices. The group associated with a “superlattice” is a subgroup of the group associated with the parent lattice. Although this nomenclature (subgroups, sublattices) is more correct from a mathematical or group theory point of view, we follow the nomenclature typically seen in the physics literature where a lattice or a structure whose volume is larger than that of the parent is referred as a superlattice.} \\
\end{align*} \]