

PAPER • OPEN ACCESS

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

To cite this article: Gus L W Hart *et al* 2019 *J. Phys. Commun.* **3** 065009

View the [article online](#) for updates and enhancements.



PAPER

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

OPEN ACCESS

RECEIVED
1 April 2019REVISED
3 June 2019ACCEPTED FOR PUBLICATION
12 June 2019PUBLISHED
25 June 2019

Original content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Gus L W Hart¹, Jeremy J Jorgensen¹ , Wiley S Morgan¹ and Rodney W Forcade²¹ Dept. of Physics and Astronomy, Brigham Young University, Provo Utah 84602 United States of America² Dept. of Mathematics, Brigham Young University, Provo Utah 84602 United States of AmericaE-mail: gus.hart@gmail.comKeywords: k -point symmetry reduction, density functional theory, electronic structure, generalized regular grids, Monkhorst–Pack grids

Abstract

We develop an algorithm for i) computing *generalized regular* k -point grids, ii) reducing the grids to their symmetrically distinct points, and iii) mapping the reduced grid points into the Brillouin zone. The algorithm exploits the connection between integer matrices and finite groups to achieve a computational complexity that is linear with the number of k -points. The favorable scaling means that, at a given k -point density, all possible commensurate grids can be generated (as suggested by Moreno and Soler) and quickly reduced to identify the grid with the fewest symmetrically unique k -points. These optimal grids provide significant speed-up compared to Monkhorst–Pack k -point grids; they have better symmetry reduction resulting in fewer irreducible k -points at a given grid density. The integer nature of this new reduction algorithm also simplifies issues with finite precision in current implementations. The algorithm is available as open source software.

1. Introduction

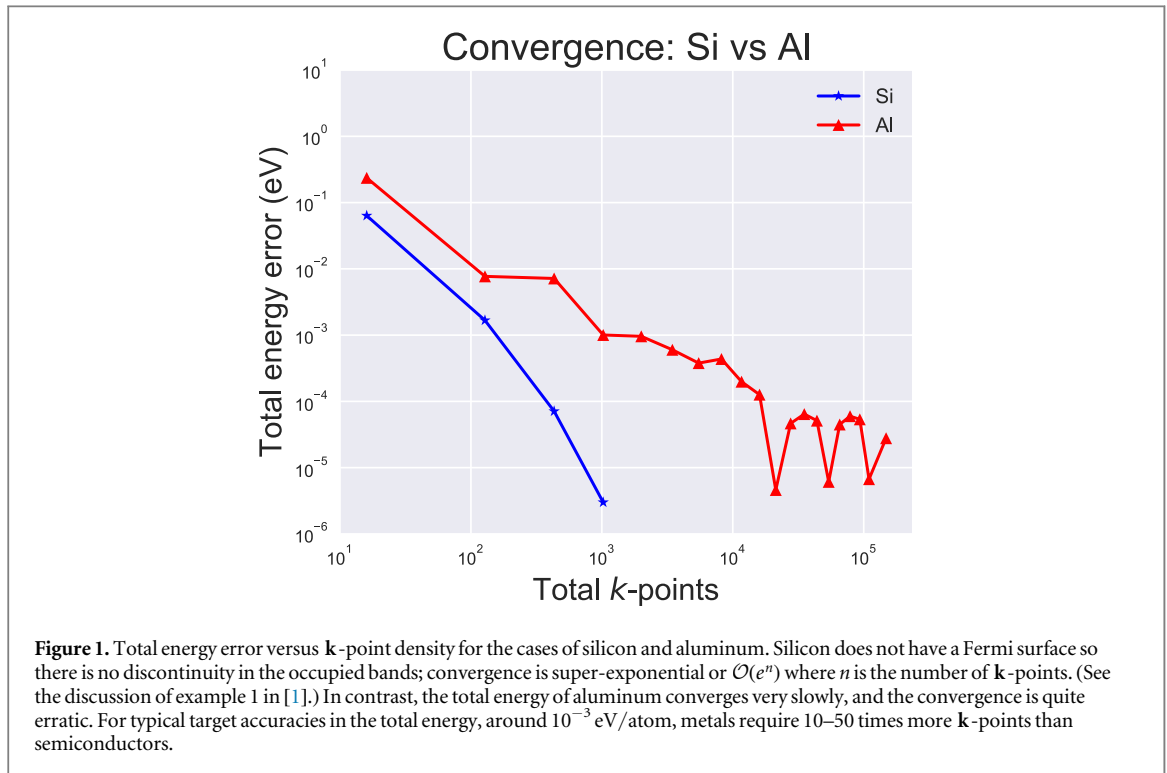
Codes that solve the many-body problem using density functional theory (DFT) use uniform grids over the Brillouin zone in order to calculate the total electronic energy, among other material properties. The total electronic energy is calculated by numerically integrating the occupied electronic bands. 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 figure 1 where we compare the convergence of an insulator (silicon) with a metal (aluminum).

The poor convergence of the electronic energy means that DFT codes must use extremely dense grids [2, 3] to achieve an accuracy of several meV/atom. To reduce computation time, it is common practice to evaluate eigenvalues at symmetrically equivalent k -points only once. This is the essence of ‘symmetry reducing’ a k -point grid.

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) 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 [4, 5] (in other words, enable a k -point generation method in the same spirit as that of [2] but have the grid generation done *on-the-fly* [6]), and 2) eliminate (or at least greatly reduce) the probability of incorrect symmetry reduction³ as the result of finite precision errors (the danger of these increases as the density of the integration grid increases).

In this brief report, an algorithm for generating, and subsequently symmetry-reducing, k -point grids is explained. This algorithm builds on concepts such as Hermite Normal Form, Smith Normal Form, and the connection between finite groups and integer matrices. These concepts are briefly explained in the main text; for more details, see the [appendix](#) and [8]. The algorithm has been implemented in an open-source code available at

³ Such errors are not uncommon in k -point reduction, but are not documented in the literature. The same errors are known to affect symmetry analysis as discussed at length in [7].



<https://github.com/msg-byu/kgridGen> and incorporated in version 6 of the VASP code [9]. The algorithm has been incorporated into a code for generating generalized regular grids <https://github.com/msg-byu/GRkgridgen> [6].

2. Generating grids

As demonstrated in figure 3, every uniform sampling of a reciprocal unit cell can be expressed through the simple integer relationship

$$\mathbb{R} = \mathbb{K}\mathbb{N} \quad (1)$$

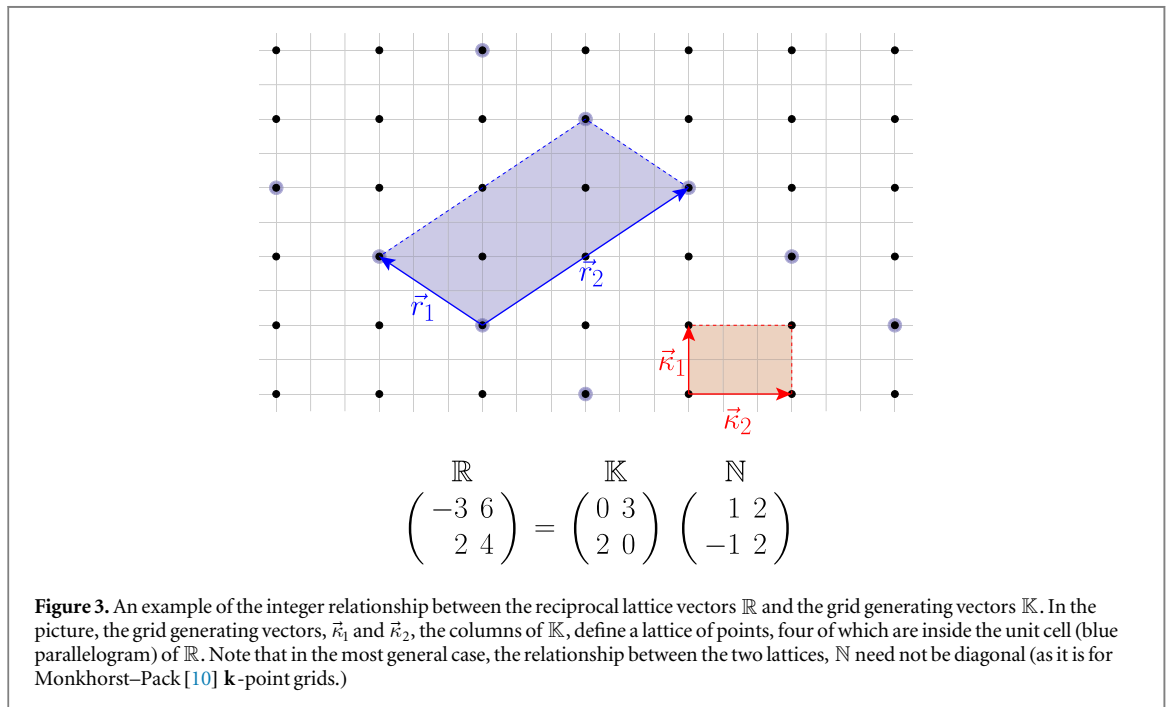
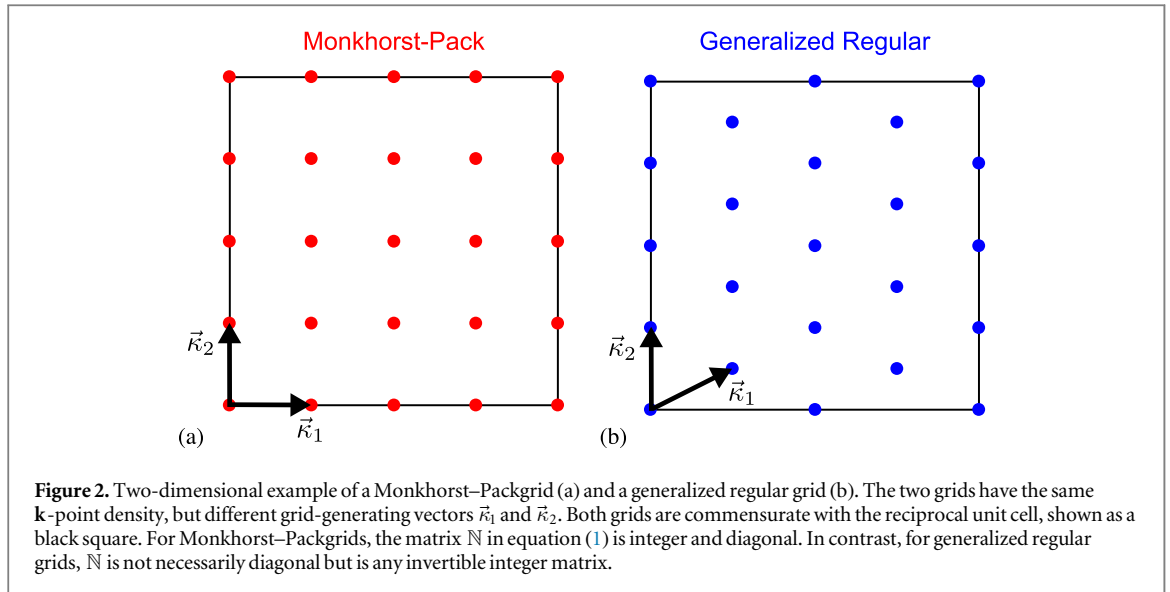
where \mathbb{R} , \mathbb{K} , and \mathbb{N} are 3×3 matrices; the columns of \mathbb{R} are the reciprocal lattice vectors, and the columns of \mathbb{K} are the \mathbf{k} -point grid generating vectors. Put simply, \mathbb{N} describes the integer linear combination of vectors of \mathbb{K} that are equivalent to \mathbb{R} . One obtains Monkhorst–Packgrids (regular grids) when \mathbb{N} is an integer, diagonal matrix. More generally, when \mathbb{N} is an invertible, integer matrix, one obtains *generalized regular* grids. Examples of Monkhorst–Pack and generalized regular grids are given in figure 2. We use R to refer to the infinite lattice of points defined by integer linear combinations of \mathbb{R} , and K to refer to the lattice of points defined by \mathbb{K} .

With no loss of generality, a new basis for the lattice K can be chosen (a different, but equivalent, \mathbb{K}) so that \mathbb{N} is a lower triangular matrix in Hermite normal form (HNF) [11]. (See section II-A of [8] for a brief introduction to HNF.) HNF is a lower-triangular canonical matrix form, where the entries below the diagonal are non-negative and strictly less than the diagonal entry in the same row. Code for converting integer matrices to Hermite Normal Form is available at <https://github.com/msg-byu/symlib> in the `rational_mathematics` module.

A \mathbf{k} -point integration grid is the set of points of the lattice K that lie inside one unit cell (one fundamental domain) of the reciprocal lattice R . We refer to this finite subset of K as K_α (See figure 3; black dots are K , dots inside the blue parallelogram comprise K_α .) The number of points that lie within one unit cell of 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, a , c , and f , such that $a \cdot c \cdot f = n$. A set of n translationally distinct⁴ points of the lattice K can be generated by taking integer linear combinations of the columns of \mathbb{K} :

⁴ If two points are translationally distinct, their difference cannot be an integer linear combination of the *reciprocal* cell vectors; that is, $\vec{k}_i - \vec{k}_j \neq n\vec{r}_1 + m\vec{r}_2 + \ell\vec{r}_3$, for all integer values n, m, ℓ . (\vec{r}_i are the columns of \mathbb{R} .)

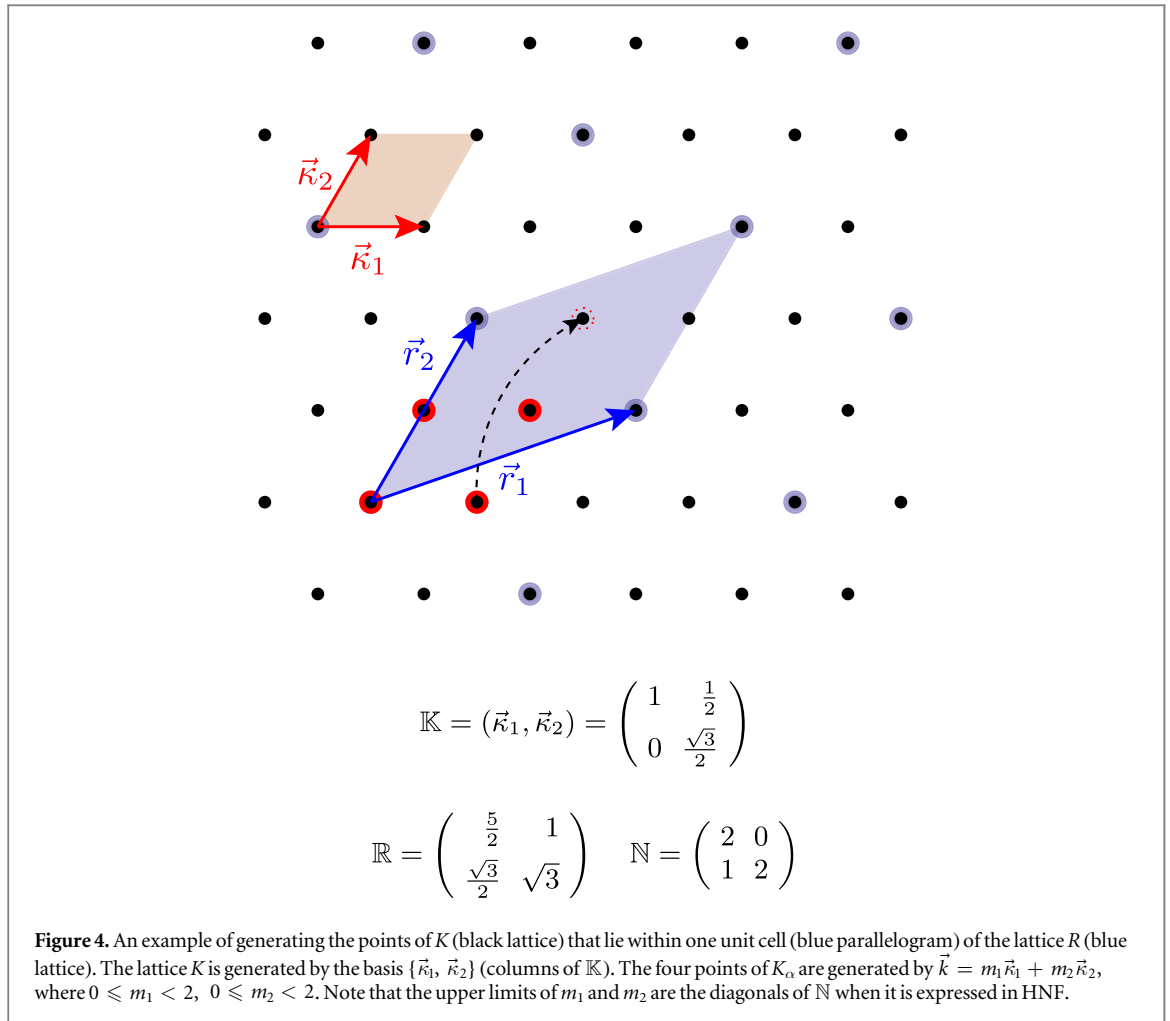


$$\vec{k} = p\vec{k}_1 + q\vec{k}_2 + r\vec{k}_3, \tag{2}$$

where $p, q,$ and r are nonnegative integers such that

$$\begin{aligned} 0 &\leq p < a \\ 0 &\leq q < c \\ 0 &\leq r < f. \end{aligned}$$

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 reducing the coordinates modulo 1 so that they all lie within the range $[0, 1)$. This is illustrated by the dashed arrow in figure 4.



Expressed as fractions of the lattice vectors of R , these four points are:

$$\begin{aligned} \vec{k}_1 &= (0, 0) \\ \vec{k}_2 &= \left(0, \frac{1}{2}\right) \\ \vec{k}_3 &= \left(\frac{1}{2}, -\frac{1}{4}\right) \xrightarrow{\text{mod } 1} \left(\frac{1}{2}, \frac{3}{4}\right) \\ \vec{k}_4 &= \left(\frac{1}{2}, \frac{1}{4}\right). \end{aligned}$$

Initially, \vec{k}_3 is not in the same unit cell as the other three points; its first coordinate is not between 0 and 1. After reducing the first coordinate modulo 1, \vec{k}_3 moves to an equivalent position in the same unit cell as the other three points.

In summary, this first part of the algorithm generates n translationally distinct points and translates them all into the first unit cell of R . (It is not necessary to translate all the points into the first unit cell, but it is convenient to do so as a first step to translating them into the first Brillouin zone. The translation into the first Brillouin zone is less trivial and is discussed in section 4.)

3. Symmetry reduction of the grid

In many cases the crystal will have some point group symmetries, and these can be exploited to reduce the number of \mathbf{k} -points where the energy eigenvalues and corresponding wavefunctions need to be evaluated. The grid is reduced by applying the point group symmetries⁵ of the crystal to each point in the grid. For example, in

⁵ In addition to the rotations, reflections, and improper rotations of the crystal, inversion symmetry is also included by default. Even when the crystal itself does not have inversion symmetry, the electronic bands generally will. If, as in the case of magnetism, the inversion symmetry is broken, the inversion symmetry can be disabled in the code.

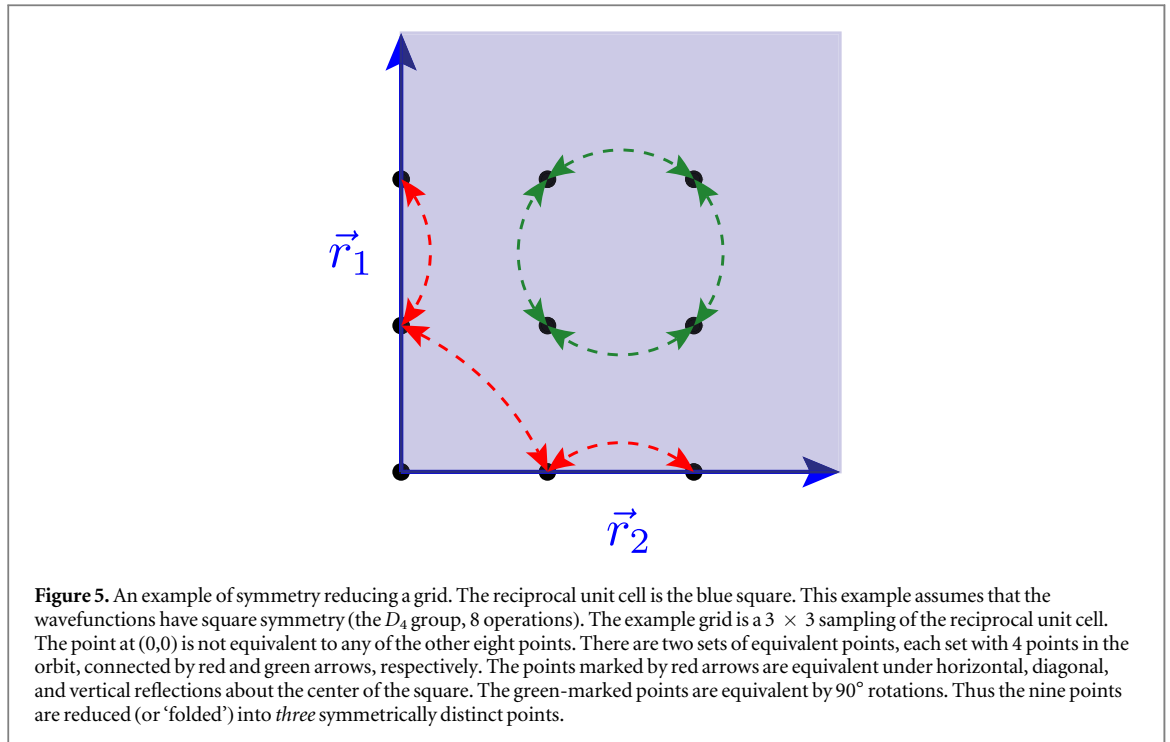


figure 5, the points connected by green arrows will be mapped onto one another by successive 90° rotations. These four symmetrically equivalent points lie on a 4-fold ‘orbit’ (as do the points marked by the red arrows). The point at the origin maps onto itself under all symmetry operations and has an orbit of length 1.

For a grid containing N_k points and a group (of rotation and reflection symmetries) with N_G operations, a naive algorithm for identifying the symmetrically equivalent points and counting the length of each orbit would be as follows: For each point ($\mathcal{O}(N_k)$ loop), compare all rotations of that point (a loop of $\mathcal{O}(|G|)$) to all other points (another $\mathcal{O}(N_k)$ loop) to find a match; for a total computational complexity of $\mathcal{O}(N_k^2 N_G)$. The algorithm is shown in pseudocode in figure 6. 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_j loop in algorithm 1. The hash table makes a one-to-one association between the ordinal counter (the index) of each point and its coordinates.

The three coefficients p, q, r in equation (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 by

$$x = p \cdot d_2 \cdot d_1 + q \cdot d_1 + r. \quad (3)$$

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 \mathbf{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 equation (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, no search is necessary—one simply maps the point (p, q, r) to the index x of the equivalent \mathbf{k} -point in the hash table.

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

$$0 \leq p < d_1, \quad 0 \leq q < d_2, \quad 0 \leq r < d_3 \quad (4)$$

(Figure 4 shows an example where the interior points do not meet these conditions.) These conditions hold only for a certain choice of basis. That basis is found by transforming the matrix \mathbb{N} in equation (1) into its Smith Normal Form [11], $\mathbb{D} = \mathbb{A}\mathbb{N}\mathbb{B}$. By elementary row and column operations, represented by unimodular matrices \mathbb{A} and \mathbb{B} , it is possible to transform \mathbb{N} into a diagonal matrix \mathbb{D} , where each diagonal element divides the ones below it: $d_{11}|d_{22}|d_{33}$, and $d_{11} \cdot d_{22} \cdot d_{33} = n = |\mathbb{N}|$ (the notation $i|j$ means that i is divisible by j). As explained in the appendix (section A.1), when \mathbb{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 \mathbb{K} , then the coordinates

Algorithm 1

```

uniqueCount ← 0
First[:] ← 0
Wt[:] ← 0
unique[:] ← 1
for each  $\mathbf{k}_i \in K_\alpha$ 
    if unique[i] ≠ 1 cycle #this
    #point and all its symmetry equivalent
    #points have already been indexed
    uniqueCount++
    First[uniqueCount] ← i
    unique[i] ← 0
    Wt[uniqueCount] ← 1
    # Now mark all equivalent points
    for each  $\mathbf{k}_j \in K_\alpha$ 
        for each  $g \in G$ 
            if  $\mathbf{k}_j = g \cdot \mathbf{k}_i$ 
                Wt[uniqueCount]; ++
                unique[j] ← 0

```

Figure 6. The typical algorithm for reducing a grid by symmetry. In the algorithm, uniqueCount is a running counter of the number of unique points and serves as the index of the orbit, First is a list of the indices of the unique points, Wt (weight) is the number of symmetrically equivalent versions of each \mathbf{k} -point in First, unique is an array of ones and zeros where each element corresponds to a \mathbf{k} -point in K_α . An element gets set to zero when the corresponding \mathbf{k} -point is equivalent to another \mathbf{k} -point, or when the \mathbf{k} -point becomes the representative \mathbf{k} -point of an orbit. This algorithm scales quadratically with the number of points in K_α and requires floating point comparisons between \mathbf{k}_i and \mathbf{k}_j .

(coefficients) of the interior points will obey equation (4). When these conditions are met, the hashing algorithm discussed above (in particular, equation (3)) becomes possible. This enables the $\mathcal{O}(N_k)$ algorithm, shown in figure 7.

4. Moving points into the first Brillouin zone

For accurate DFT calculations, it is best if the energy eigenvalues (electronic bands) are evaluated at \mathbf{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 $E(\mathbf{k})$ should be the same in every unit cell, but numerically the periodicity of the electronic bands is only approximate, becoming less accurate for \mathbf{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 \mathbf{k} -points in the first Brillouin zone are closer to the origin than to any other lattice point. Conceptually, an algorithm for translating a \mathbf{k} -point of the integration grid into the first zone merely requires one to look at all translationally equivalent ‘cousins’ of the \mathbf{k} -point and select the one closest to the origin. But the number of translationally equivalent cousins is countably infinite, 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 figure 8. In three-dimensions, if the basis vectors of the reciprocal unit cell are as short as possible (the so-called Minkowski-reduced basis [12]), then the eight unit cells that all share a vertex at the origin *must* contain the Brillouin zone. 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 (section A.1). The steps for moving \mathbf{k} -points into the Brillouin zone are as follows:

1. Minkowski-reduce the reciprocal unit cell [12] (i.e., find the basis with the shortest basis vectors⁶)

⁶ Our Fortran code for computing the Minkowski reduced basis is available at <https://github.com/msg-byu/symlib> in the vector_matrix_utilities module.

Algorithm 2

```

uniqueCount ← 0
hashTable[:] ← 0
First[:] ← 0
Wt[:] ← 0
for each  $\mathbf{k}_i \in K_\alpha$ 
  indx ←  $\mathbb{K}^{-1}\mathbb{A}\mathbb{D} \cdot \mathbf{k}_i$ 
  if hashTable[indx]  $\neq$  0 cycle #this
  #point and all its symmetry equivalent
  #points have already been indexed
  uniqueCount++
  hashtable[indx] ← uniqueCount
  First[uniqueCount] ← i
  Wt[uniqueCount] ← 1
  # Now mark all equivalent points
  for each  $g \in G$ 
     $\mathbf{k}_{\text{rot}} \leftarrow g \cdot \mathbf{k}_i$ 
    indx ←  $\mathbb{K}^{-1}\mathbb{A}\mathbb{D} \cdot \mathbf{k}_{\text{rot}}$ 
    if hashtable[indx] == 0
      hashtable[indx] ← uniqueCount
      Wt[uniqueCount]++

```

Figure 7. Our algorithm that reduces a grid to a set of symmetrically distinct \mathbf{k} -points. In the algorithm, `uniqueCount` is a running counter of the number of unique points and serves as the index of the orbit, `First` is a list of the indices of the unique points, `Wt` (weight) is the number of symmetrically equivalent versions of each \mathbf{k} -point in `First`, and `hashtable` is a hash table that points from the position of a \mathbf{k} -point in K_α to the index of its orbit. In contrast to algorithm 1, this algorithm scales linearly with the number of points in the grid K_α and does not require floating point comparisons.

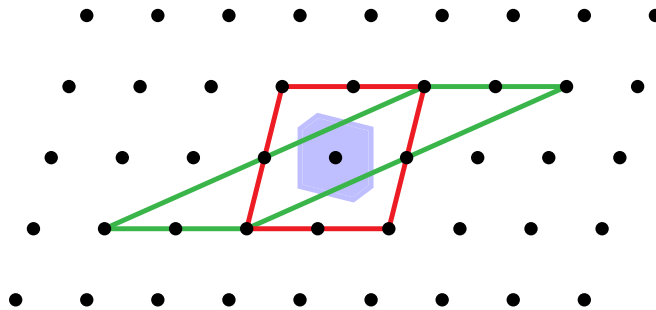


Figure 8. A two-dimensional example of the ‘closest cousin’ guarantee. The Brillouin zone (blue) will be completely contained in the union of 4 basis cells around the origin (shown in red) when the basis vectors are chosen to be as short as possible (the so-called Minkowski basis). On the other hand, if the basis is not Minkowski reduced, regions of the Brillouin zone may lie outside the union of the 4 basis cells (depicted by the cell in green). A proof is given in the appendix (section A.1).

2. For each \mathbf{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.

5. Conclusion

We have developed an algorithm that i) generates generalized regular \mathbf{k} -point grids, ii) reduces the grids by symmetry, and iii) maps the points of the reduced grids into the first Brillouin zone. Whereas the typical algorithm for generating and reducing \mathbf{k} -point grids scales quadratically with the number of \mathbf{k} -points, this algorithm scales linearly. The improved scaling becomes essential when one generates and symmetry reduces all combinatorically possible generalized regular grids at a given \mathbf{k} -point density, in order to select the one with the fewest number of reduced \mathbf{k} -points [6].

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. Mapping the grid to the first Brillouin zone is more efficient due to a proof that limits the search for translationally equivalent \mathbf{k} -points to the eight unit cells having a vertex at the origin. The algorithm has been incorporated into version 6 of VASP [9].

Acknowledgments

We are grateful to Martijn Marsmann who found a bug in an early version of the code that spoiled the $\mathcal{O}(N_k)$ scaling and who implemented the algorithm in VASP, version 6. Tim Mueller also contributed by pointing out a simpler approach to generating all the \mathbf{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).

Appendix

A.1. 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 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 $V \not\subseteq U_R$, then the basis is not Minkowski reduced.

If $V \not\subseteq U_R$ 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 $V \not\subseteq U_R$ 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.

A.1.1. 2D argument. Let \vec{r}_1 and \vec{r}_2 be basis elements of R . Assuming that $V \not\subseteq U_R$ 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 figure A1. We have (since x' must be farther from the origin)

$$\begin{aligned} x^2 &< x'^2 \\ (\vec{r}_1 + \lambda \vec{r}_2)^2 &< (\lambda \vec{r}_2)^2 \\ \vec{r}_1^2 + 2\lambda \vec{r}_1 \cdot \vec{r}_2 + \lambda^2 \vec{r}_2^2 &< \lambda^2 \vec{r}_2^2 \\ \vec{r}_1^2 &< -2\lambda \vec{r}_1 \cdot \vec{r}_2 \end{aligned}$$

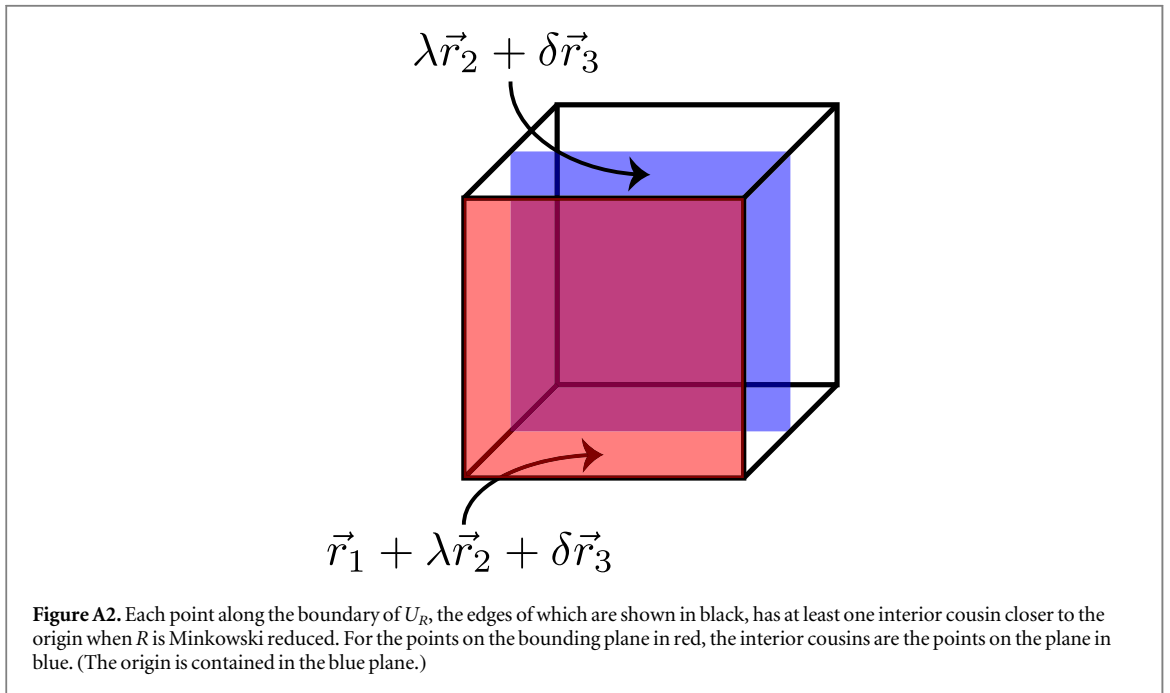
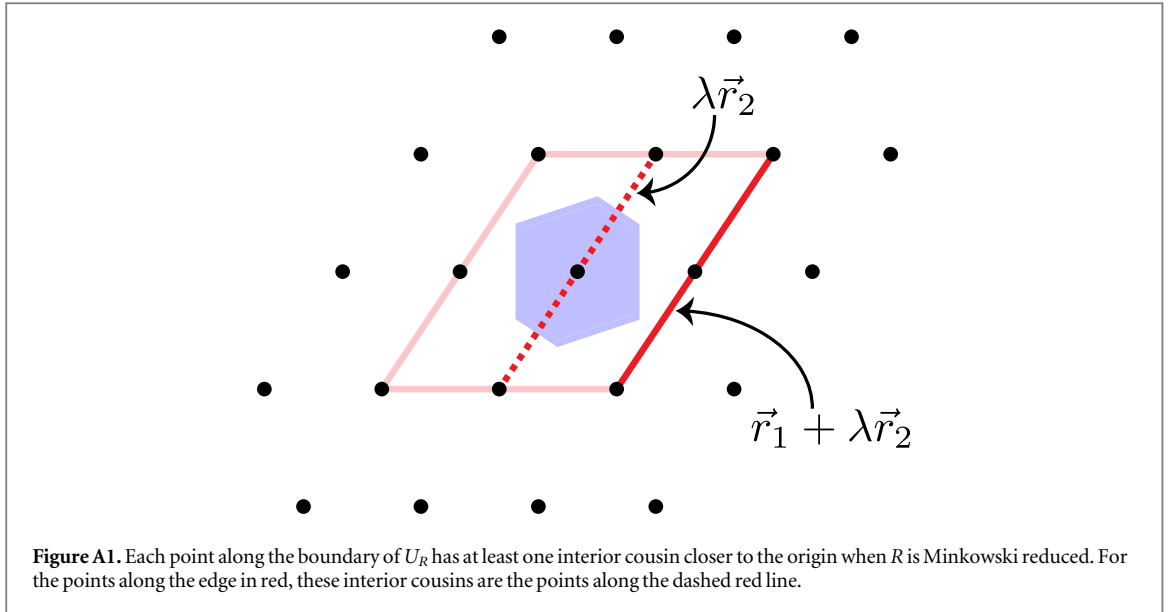
Since the expression on the left-hand side is greater than zero, the expression on the right-hand side must be also and taking the absolute value of both sides does not change the inequality:

$$|\vec{r}_1^2| < |-2\lambda \vec{r}_1 \cdot \vec{r}_2| \Rightarrow |\vec{r}_1|^2 < 2|\lambda| |\vec{r}_1 \cdot \vec{r}_2|.$$

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, $V \not\subseteq U_R$, 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 $V \subseteq U_R$.



A.1.2. 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 λ and δ are elements of $[-1, 1]$) is a point on the boundary of U_R which is closer to the origin than are 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 figure A2. Thus

$$\begin{aligned} x^2 &< x'^2 \\ (\vec{r}_1 + \lambda\vec{r}_2 + \delta\vec{r}_3)^2 &< (\lambda\vec{r}_2 + \delta\vec{r}_3)^2. \end{aligned}$$

Simplifying this expression gives

$$\vec{r}_1^2 < -2\lambda\vec{r}_1 \cdot \vec{r}_2 - 2\delta\vec{r}_1 \cdot \vec{r}_3. \tag{6}$$

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:

$$|\vec{r}_1^2| < |-2\lambda\vec{r}_1 \cdot \vec{r}_2 - 2\delta\vec{r}_1 \cdot \vec{r}_3| \Rightarrow |\vec{r}_1|^2 < |2\lambda\vec{r}_1 \cdot \vec{r}_2 + 2\delta\vec{r}_1 \cdot \vec{r}_3| \tag{7}$$

To simplify the right-hand side of equation (7) the triangle inequality is used, making it more likely that the inequality is satisfied:

$$\begin{aligned} |\vec{r}_1|^2 &< |2\lambda\vec{r}_1 \cdot \vec{r}_2 + 2\delta|\vec{r}_1 \cdot \vec{r}_3| \\ |\vec{r}_1|^2 &< 2|\lambda||\vec{r}_1 \cdot \vec{r}_2| + 2|\delta||\vec{r}_1 \cdot \vec{r}_3| \end{aligned} \quad (8)$$

Since the expression in equation (8) does not depend on the sign of λ or δ , we can restrict λ and δ to positive values within $[0,1]$ without loss of generality. Consider now *another* cousin that lies within U_R and on the same plane as x' : $x'' = (\lambda - 1)\vec{r}_2 + (\delta - 1)\vec{r}_3$. Repeating the same process for x' with x'' gives

$$|\vec{r}_1|^2 < 2|\lambda - 1||\vec{r}_1 \cdot \vec{r}_2| + 2|\delta - 1||\vec{r}_1 \cdot \vec{r}_3| \quad (9)$$

Combining equations (8) and (9) gives

$$\begin{aligned} |\vec{r}_1|^2 &< (|\lambda| + |\lambda - 1|)|\vec{r}_1 \cdot \vec{r}_2| + (|\delta| + |\delta - 1|)|\vec{r}_1 \cdot \vec{r}_3| \\ |\vec{r}_1| &< \frac{|\vec{r}_1 \cdot \vec{r}_2|}{|\vec{r}_1|} + \frac{|\vec{r}_1 \cdot \vec{r}_3|}{|\vec{r}_1|} \end{aligned} \quad (10)$$

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 equation (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 $V \subseteq U_R$.

A.2. 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 equation (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.

A.2.1. Groups in smith normal form. Begin with the simplest case. Let \mathbb{N} be a non-singular 3×3 matrix of integers. Its columns represent the basis for a subgroup \mathcal{L}_N of the group \mathbf{Z}^3 (where \mathbf{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*⁷ whose basis is given by the columns of \mathbb{N} . Since \mathbf{Z}^3 and its subgroups are Abelian, we know that all the subgroups are *normal* so there exists a quotient group $G = \mathbf{Z}^3/\mathcal{L}_N$, 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 \mathcal{L}_N within \mathbf{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 \mathbb{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, $G \cong \mathbf{Z}_{m_1} \oplus \mathbf{Z}_{m_2} \oplus \dots \oplus \mathbf{Z}_{m_k}$ where $m_1|m_2|\dots|m_{k-1}|m_k$ 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, \dots, m_k when written in this form.)

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

⁷ 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.

A.2.2. 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 \mathbb{N} effectively tells one how to represent the quotient group $\mathbb{Z}^3/\mathcal{L}_N$ 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 \mathbb{N} give the homomorphism ψ suggested above.

A.2.3. The connection between SNF groups and SNF matrices. In the matrix case, since the operations are elementary row and column operations, we have $\mathbb{D} = \mathbb{A}\mathbb{N}\mathbb{B}$ where \mathbb{A} and \mathbb{B} are integer matrices with determinant ± 1 representing the accumulated row operations and column operations respectively. The matrix \mathbb{D} is completely determined by \mathbb{N} , but the matrices \mathbb{A} and \mathbb{B} depend on the algorithm used to arrive at the Smith Normal Form of \mathbb{N} . A different implementation might yield $\mathbb{D} = \mathbb{A}'\mathbb{N}\mathbb{B}'$ (same \mathbb{N} and same \mathbb{D} , but different \mathbb{A} and \mathbb{B}).

Note that, since \mathbb{B} represents elementary column operations, the product $\mathbb{N}\mathbb{B}$ simply represents a change of basis from \mathbb{N} to a new basis $\mathbb{N}' = \mathbb{N}\mathbb{B}$. In other words, the columns of \mathbb{N}' are still a basis for \mathcal{L}_N . But the new basis has the property that $\mathbb{A}\mathbb{N}' = \mathbb{D}$. That means that every element $\vec{w} = \mathbb{N}'\vec{z}$ of \mathcal{L}_N (where \vec{z} is some element

of \mathbb{Z}^3) will satisfy the equation $\mathbb{A}\vec{w} = \mathbb{D}\vec{z} = \begin{pmatrix} \mathbb{D}_{11}z_1 \\ \mathbb{D}_{22}z_2 \\ \mathbb{D}_{33}z_3 \end{pmatrix}$. In other words, $\mathbb{A}\vec{w}$ will be a vector whose entries are multiples of the corresponding diagonal entries in \mathbb{D} .

To put it another way, define $*$ to be the operation that maps $\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ in \mathbb{Z}^3 to $\vec{x}^* = \begin{pmatrix} x_1 \pmod{\mathbb{D}_{11}} \\ x_2 \pmod{\mathbb{D}_{22}} \\ x_3 \pmod{\mathbb{D}_{33}} \end{pmatrix}^T$.

Then we have shown $\vec{w} \in \mathcal{L}_N$ iff $(\mathbb{A}\vec{w})^* = (0, 0, 0)$ (the zero-element in the group $G_0 = \mathbb{Z}_{\mathbb{D}_{11}} \oplus \mathbb{Z}_{\mathbb{D}_{22}} \oplus \mathbb{Z}_{\mathbb{D}_{33}}$).

That suggests we let $\psi(\vec{w}) = (\mathbb{A}\vec{w})^*$, 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 \mathcal{L}_N , we see (by the First Isomorphism theorem of group theory) that $G_0 \cong \mathbb{Z}^3/\mathcal{L}_N$, and ψ 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 \mathbb{D} , and the transition matrix \mathbb{A} provides the homomorphism which maps the parent lattice onto the group.

A.2.3.1. An example

Let $\mathbb{N} = \begin{pmatrix} 1 & 2 & -1 \\ 1 & 4 & -3 \\ 0 & 2 & 4 \end{pmatrix}$. This describes a lattice \mathcal{L}_N which contains the points $\vec{p}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$, $\vec{p}_2 = \begin{pmatrix} 2 \\ 4 \\ 2 \end{pmatrix}$, and

$\vec{p}_3 = \begin{pmatrix} -1 \\ -3 \\ 4 \end{pmatrix}$, 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 $\mathbb{D} = \mathbb{A}\mathbb{N}\mathbb{B}$ 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 $\vec{w} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ then $\mathbb{A}\vec{w} = \begin{pmatrix} y \\ z \\ x - y - 2z \end{pmatrix}$ and thus

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

(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 ψ , 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 and p_3 satisfy these congruences, and thus so will all their integer linear combinations (all points in \mathcal{L}_N).

A.2.3.2. *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 $\mathbb{A} = \begin{pmatrix} 1 & 0 & 0 \\ -5 & 3 & 1 \\ -2 & 2 & 1 \end{pmatrix}$ and $\mathbb{B} = \begin{pmatrix} 0 & -1 & 2 \\ 0 & 0 & 1 \\ -1 & -1 & 4 \end{pmatrix}$, 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 .

A.2.4. *Non-integer lattices.* Now, what about the more complicated situation, where \mathbb{N} represents a (possibly HNF) 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 \mathbb{V} and lattice \mathcal{L}_V and a basis $\mathbb{W} = \mathbb{V}\mathbb{N}$ 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 $\mathbb{D} = \mathbb{A}\mathbb{N}\mathbb{B}$ (where \mathbb{D} is the SNF of \mathbb{N}).

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

A.2.5. *Example: general (non-integer) lattices.* Suppose \mathcal{L}_V is the lattice defined by (columns of) the basis matrix

$$\mathbb{V} = \begin{pmatrix} 1 & 1/2 & 0 \\ 0 & \sqrt{3}/2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \text{ and } \mathcal{L}_W \text{ is the subgroup lattice defined by the basis matrix } \mathbb{W} = \mathbb{V}\mathbb{N} \text{ where}$$

$$\mathbb{N} = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 2 & 2 \\ 4 & 0 & 4 \end{pmatrix}. \text{ In other words, one basis for } \mathcal{L}_W \text{ is given by the columns of } \mathbb{W} = \begin{pmatrix} 5 & 3 & 3 \\ \sqrt{3} & \sqrt{3} & \sqrt{3} \\ 8 & 0 & 8 \end{pmatrix}.$$

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

$$\mathbb{A}\mathbb{V}^{-1} = \begin{pmatrix} 1 & -\sqrt{3}/3 & -1/2 \\ 1 & -\sqrt{3} & -1/2 \\ -6 & 14\sqrt{3}/3 & 5/2 \end{pmatrix},$$

which provides our homomorphism $\psi(\vec{x}) = (\mathbb{A}\mathbb{V}^{-1}\vec{x})^*$ from \mathcal{L}_V onto G .

If we let $\vec{x} = \begin{pmatrix} 2 \\ \sqrt{3} \\ 2 \end{pmatrix}$ which is an element of \mathcal{L}_V but not of \mathcal{L}_W , then $\mathbb{A}\mathbb{V}^{-1}\vec{x} = \begin{pmatrix} 0 \\ -2 \\ 7 \end{pmatrix}$ and $\psi(\vec{x}) = (0, 0, 3) \in G$

(after reducing the elements modulo 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 \mathcal{L}_W (the kernel), then $\mathbb{A}\mathbb{V}^{-1}\vec{x} = \begin{pmatrix} 2 \\ 0 \\ -8 \end{pmatrix}$ and so $\psi(\vec{x}) = (0, 0, 0)$, and \vec{x} is in the group.

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

ORCID iDs

Jeremy J Jorgensen  <https://orcid.org/0000-0003-0629-7549>

References

- [1] Weideman J A C 2002 *The American Mathematical Monthly* **109** 21
- [2] Wisesa P, McGill K A and Mueller T 2016 *Phys. Rev. B* **93** 155109
- [3] Morgan W S, Jorgensen J J, Hess B C and Hart G L 2018 *Comput. Mater. Sci.* **153** 424
- [4] Moreno J and Soler J M 1992 *Phys. Rev. B* **45** 13891
- [5] Froyen S 1989 *Phys. Rev. B* **39** 3168
- [6] Morgan W S, Christensen J E, Hamilton P K, Jorgensen J J, Campbell B J, Hart G L and Forcade R W 2019 arXiv:1902.03257
- [7] Hicks D, Oses C, Gossett E, Gomez G, Taylor R H, Toher C, Mehl M J, Levy O and Curtarolo S 2018 *Acta Crystallogr. A* **74** 184
- [8] Hart G L and Forcade R W 2008 *Phys. Rev. B* **77** 224115
- [9] Kresse G and Hafner J 1993 *Phys. Rev. B* **47** 558
- [10] Monkhorst H J and Pack J D 1976 *Phys. Rev. B* **13** 5188
- [11] Storjohann A 1994 Computation of Hermite and Smith normal forms of matrices *PhD Thesis* University of Waterloo
- [12] Nguyen P Q and Stehlé D 2009 *ACM Transactions on algorithms (TALG)* **5** 46