

# Visualising higher order Brillouin zones with applications

R C Andrew<sup>1</sup>, T Salagaram<sup>2,\*</sup> and N Chetty<sup>1,3</sup>

<sup>1</sup> Department of Physics, University of Pretoria, Pretoria 0002, South Africa

<sup>2</sup> Department of Physics, University of Cape Town, Rondebosch 7701, South Africa

<sup>3</sup> National Institute for Theoretical Physics, Johannesburg, 2000, South Africa

\*Author to whom any correspondence should be addressed.

E-mail: Richard.Andrew@up.ac.za, trisha.salagaram@uct.ac.za and Nithaya.Chetty@up.ac.za

## Abstract

A key concept in material science is the relationship between the Bravais lattice, the reciprocal lattice and the resulting Brillouin zones (BZ). These zones are often complicated shapes that are hard to construct and visualise without the use of sophisticated software, even by professional scientists. We have used a simple sorting algorithm to construct BZ of any order for a chosen Bravais lattice that is easy to implement in any scientific programming language. The resulting zones can then be visualised using freely available plotting software. This method has pedagogical value for upper-level under-graduate students since, along with other computational methods, it can be used to illustrate how constant-energy surfaces combine with these zones to create van Hove singularities in the density of states. In this paper we apply our algorithm along with the empirical pseudopotential method and the 2D equivalent of the tetrahedron method to show how they can be used in a simple software project to investigate this interaction for a 2D crystal. This project not only enhances students' fundamental understanding of the principles involved but also improves transferable coding skills.

Keywords: Fermi surface, visualisation, Brillouin zone, van Hove singularities, density of states

## 1. Introduction

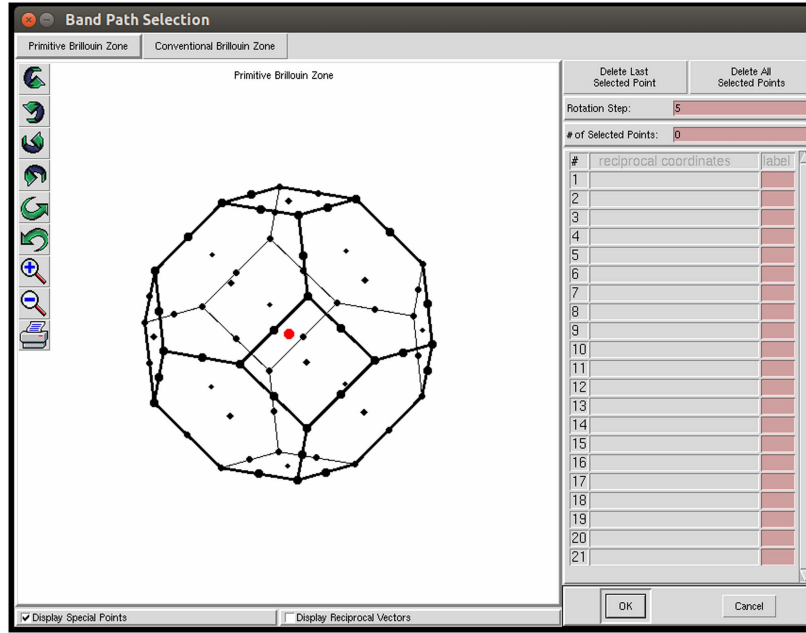
It is all too often the case in materials science (as well as in other areas of science) that modern modelling codes are used as ‘black-boxes’ without a full understanding of the underlying theoretical and computational principles. This is mostly due to the fact that most research groups no longer involve their postgraduate students in the development of in-house production codes; they often use open-source software produced by other research groups or commercially available codes. Their sole understanding of the software lies often mainly in the graphical user interface which presents a variety of pull down menus and check-box options as well as sophisticated visualisations. Students lack the fundamental programming skills that would allow them to not only produce useful data processing tools but also the ability to read, modify and improve existing codes.

This state of affairs poses a problem since in both academia and private industry, advertised positions increasingly require knowledge of advanced computational techniques beyond mere proficiency in software use; those that lack these skills will have ever decreasing employment opportunities. In addition, the quality of their research will suffer due to a fundamental lack of understanding of the methods used in modelling codes. As academics, any postgraduate students they supervise would have a greater chance of inheriting these limitations. However, with a larger theoretical knowledge base and a more complete computational skill set, they would greatly improve the effectiveness of their research as well as their marketability in an increasingly competitive field.

To achieve this, great care must be taken to train undergraduate students pursuing the computational stream. Numerical coding concepts must be introduced early and developed as students work through the theoretical material. Later when they specialise, fundamental aspects of typical modelling code must be covered along with more advanced computational principles; both the theoretical basis of these techniques and their implementation must be investigated. One method to do this is to have students complete simplified computational projects [1], using an appropriate scientific computer language, that illustrate these key principles and techniques. The benefits are that the students’ conceptual understanding of the underlying theoretical principles is enhanced, and they can see how these techniques are implemented into production code. The resulting computational graduates will become more effective applied researchers with the added ability to add functionality to the software they use. The main benefit will be a scientific profession with better trained, well rounded practitioners.

A fundamental aspect of all electronic structure modelling code used by material scientists is the concept of the Brillouin zone (BZ). In the 1930s, scientists applied quantum mechanics to study electrons in a periodic potential in order to calculate various material properties. During this time, Leon Brillouin [2] made an important contribution by formulating geometric regions of momentum space (also called reciprocal space). These BZ provide a means of determining the electronic energies and wavefunctions for a crystal at every point in reciprocal space by only considering the points inside the first zone.

Higher order BZ also play a role in other physical properties. The Fermi surface of metals affects such phenomena such as the de Haas–van Alphen, cyclotron-resonance, and anomalous-skin effects [3]. It can extend into the higher BZ with the crossing of Bragg-planes resulting in surface deformations due to off-diagonal potential terms in the Hamiltonian matrix [4]. Extremum points in the energy band dispersion relation  $\varepsilon_n(\mathbf{k})$  result in the creation of van Hove singularities in the density of states (DOS) function [5]. The location of these points can be deduced by comparing the features of the DOS to where the corresponding constant energy surface interacts with the BZ boundaries.



**Figure 1.** First Brillouin zone for the face-centred cubic real space lattice produced with XCrySDen [8].

In this paper we present a simple software project to illustrate many of these concepts. We use a simple sorting algorithm to divide a grid of  $\mathbf{k}$ -points into their respective BZ. We present example images in 2D and 3D produced with easy-to-use plotting software showing intricate visualisations of the first BZ as well as those of higher orders. The algorithm is then used to show the free-electron Fermi surface, likewise divided, with higher order sections folded back into the first BZ. We then use the empirical pseudopotential method (EPM) [6] and the 2D equivalent of the tetrahedron method [7] to look at the relationship between constant energy surfaces and van Hove singularities in the DOS.

## 2. The reciprocal lattice

For every real crystal lattice, there is a reciprocal lattice constructed from the real lattice spanning vectors that define the unit cell. The three spanning vectors  $\mathbf{a}$ ,  $\mathbf{b}$  and  $\mathbf{c}$  define the three reciprocal lattice spanning vectors  $\mathbf{a}^*$ ,  $\mathbf{b}^*$  and  $\mathbf{c}^*$  using the following relationships between the two sets of vectors:

$$\begin{aligned} \mathbf{a} \cdot \mathbf{a}^* &= 2\pi & \mathbf{a} \cdot \mathbf{b}^* &= 0 & \mathbf{a} \cdot \mathbf{c}^* &= 0 \\ \mathbf{b} \cdot \mathbf{b}^* &= 2\pi & \mathbf{b} \cdot \mathbf{a}^* &= 0 & \mathbf{b} \cdot \mathbf{c}^* &= 0 \\ \mathbf{c} \cdot \mathbf{c}^* &= 2\pi & \mathbf{c} \cdot \mathbf{a}^* &= 0 & \mathbf{c} \cdot \mathbf{b}^* &= 0. \end{aligned}$$

Reciprocal lattice points are then defined by the following vectors:

$$\mathbf{G} = i \mathbf{a}^* + j \mathbf{b}^* + k \mathbf{c}^*,$$

$$i = 0, \pm 1, \pm 2, \dots, \quad j = 0, \pm 1, \pm 2, \dots, \quad k = 0, \pm 1, \pm 2, \dots$$

The vectors  $\mathbf{c}$  and  $\mathbf{c}^*$  are not considered for 2D crystals.

For 2D, one choice of reciprocal unit cell is the parallelogram created by the two reciprocal spanning vectors. For 3D it is the parallelepiped created by the three spanning vectors. Another choice of unit cell is the Wigner–Seitz cell defining the first BZ.

### 3. The sorting algorithm

There have been many presentations on the calculation and construction of BZ's over the years. Modern, sophisticated software packages, such as XCrySDen [8, 9] (see figure 1), use GUI toolkits and APIs, such as Tcl/Tk and OpenGL to visualise the first BZ. Simple codes have also been presented such as that of Shirer and Merrill [10]. They made available a computer programme written in BASIC which calculated the BZ and Fermi circle for any 2D lattice given the spanning vectors and number of electrons per unit cell as inputs. The programme generated the reciprocal lattice points, two points of each perpendicular bisector and the Fermi radius. These points defined the BZ boundaries which could then be plotted.

The simple algorithm presented here focuses not on the boundary points of each BZ but rather on the interior points and visualising the BZ can be done on simple plotting software such as `gnuplot`. The algorithm follows from the following definition [11]:

- The  $n$ th BZ contains those points reached from the origin while crossing exactly  $(n - 1)$  Bragg planes.

A Bragg plane is the plane located halfway between the origin and the reciprocal lattice point defined by  $\mathbf{G}$ , that has its normal plane vector orientated along  $\mathbf{G}$ . By crossing this plane, a  $\mathbf{k}$ -point is further from the origin than it is to the lattice point defined by  $\mathbf{G}$ . This enables an alternate definition for the  $n$ th BZ:

- A  $\mathbf{k}$ -point is in the  $n$ th BZ if the mathematical relation

$$|\mathbf{k}| < |\mathbf{k} - \mathbf{G}| \tag{1}$$

fails for exactly  $(n - 1)$   $\mathbf{G}$  vectors.

The sorting algorithm uses the  $\mathbf{G}$  vectors that define the reciprocal lattice so that:

1. A  $\mathbf{k}$ -point is chosen.
2. The relation (1) is tested for each  $\mathbf{G}$  vector where  $n$  is the total number of fails.
3. The point is in the  $(n + 1)$ th BZ.

In practice, only non-zero  $\mathbf{G}$  vectors within a suitably large radius  $G_{\text{cut}}$  centred at the origin need to be considered. This finite set of vectors effectively places an upper-bound on the highest order of BZ that can be classified. A suitable number of  $\mathbf{G}$  vectors must be chosen to ensure a thorough search.

The algorithm is easy to implement using any programming language that supports basic mathematics with indexed data types. For the purposes of this paper, Python and Fortran were used.

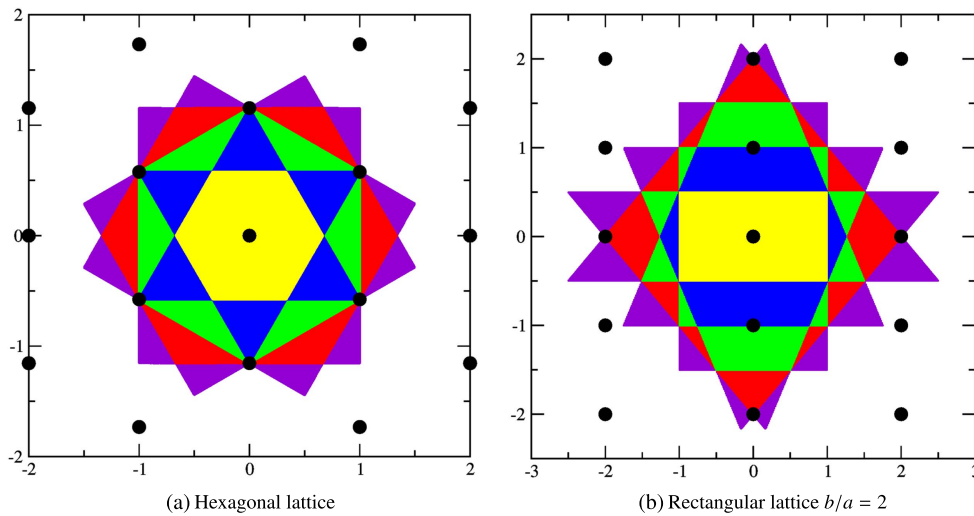


Figure 2. Brillouin zones of the hexagonal and rectangular lattices.

## 4. Visualising BZ

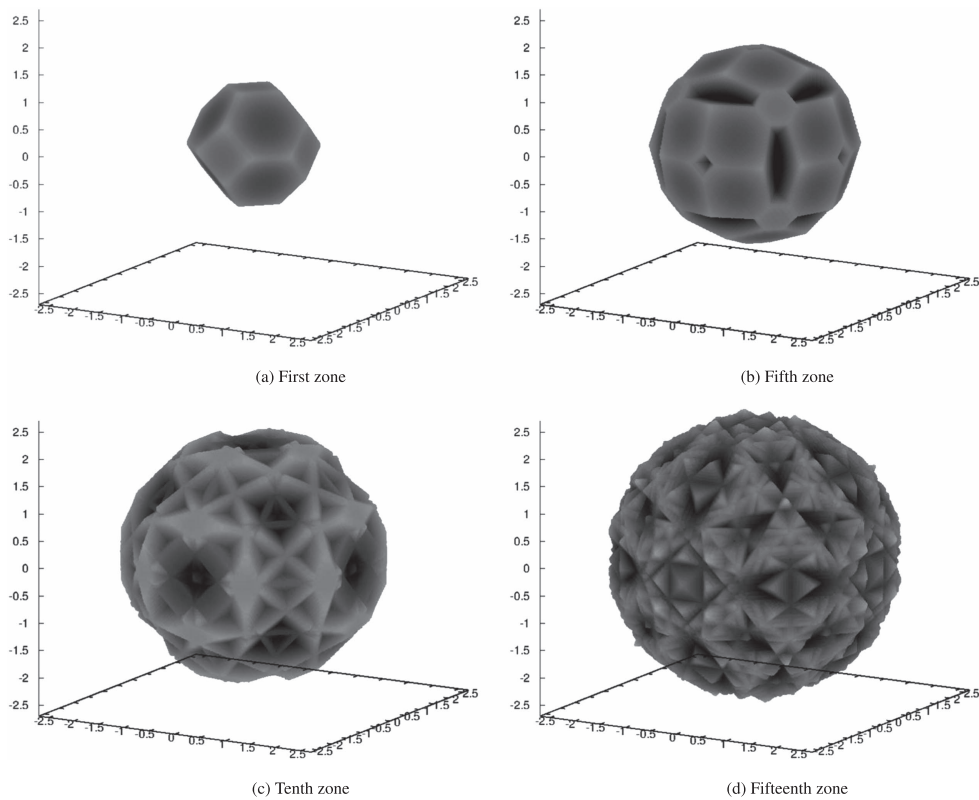
### 4.1. 2D case

The 2D BZ can easily be worked out on paper. Bragg lines are drawn for each  $\mathbf{G}$  vector and the various bounded areas define each zone. These are labelled according to the first definition for the  $n$ th BZ. Students can use these pen-on-paper constructions to test their code.

Figure 2 shows the BZ for the hexagonal and rectangular lattices constructed using the sorting algorithm. For convenience, points were plotted in units of  $2\pi/a$  where  $a$  is the lattice spacing. Using OpenMP-parallelised code, a dense  $2000 \times 2000$  grid of points defined within a centred  $4 \times 4$  supercell were sorted and stored along with their BZ number labels. A separate post-processing programme was used to extract the points for a particular BZ of interest and store them in a datafile. These were then plotted using the `xmgrace` plotting programme. A different colour was assigned to each BZ; yellow for the first BZ surrounded by higher order zones of blue (2nd zone), green (3rd zone), red (4th zone) then purple (5th zone).

From these plots, students can confirm using a ruler that mathematical relation (1) fails for exactly  $(n - 1)$   $\mathbf{G}$  vectors for any point in the  $n$ th zone. They can also verify that the area covered by each BZ is the same as that for the first zone or any other unit cell. Discrepancies in area size for increasing BZ order indicate artificially distorted shapes where a zone is either incomplete or has added parts. The beginning of an increasing size trend indicates that not enough  $\mathbf{G}$  vectors were used while decreased size indicates that the higher order BZ have extended past the boundaries of the chosen gridded area. The 2D case easily shows students that all the BZ taken together fully cover reciprocal space and each BZ is a unique ‘cut’. These simple observations enhance a students’ understanding of BZ and their relationship to the reciprocal lattice points and the surrounding space.

So far, the 2D case has been used as a verification step to ensure the students’ code is working properly and some general properties of BZ have been noted. The strength and usefulness of the algorithm is more clearly evident in the 3D case.

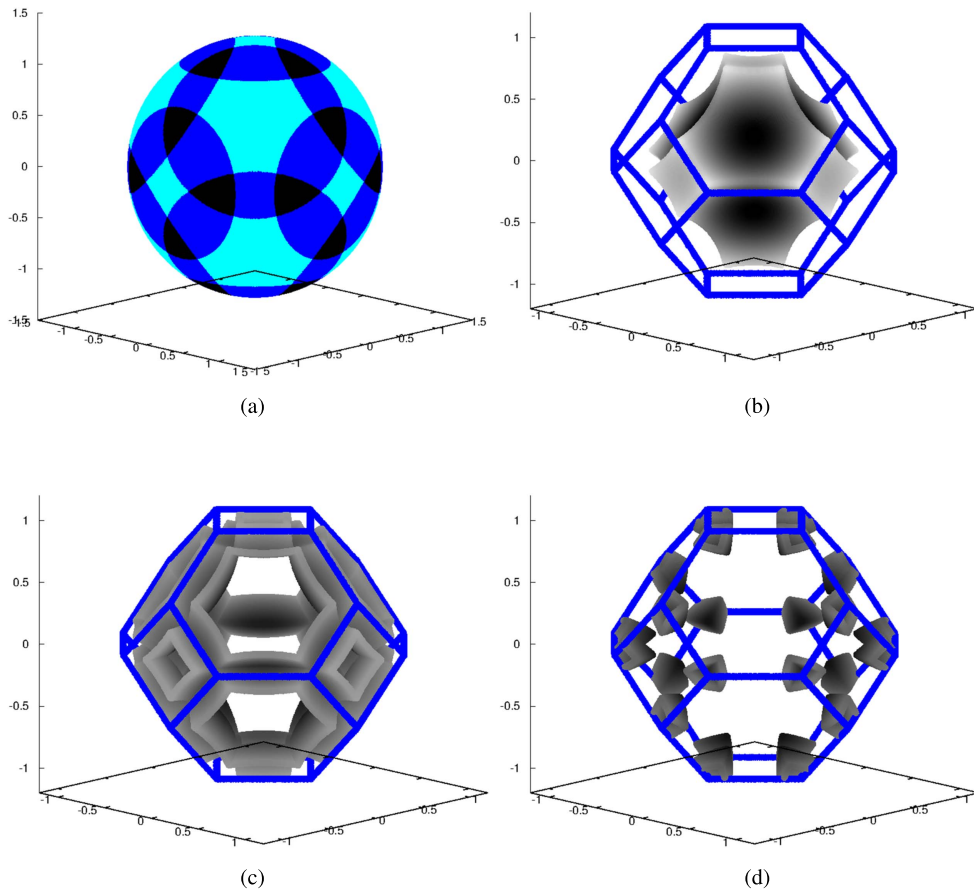


**Figure 3.** Brillouin zones of the fcc lattice.

#### 4.2. 3D case

Pen-on-paper constructions for the BZ of a 3D crystal lattice are non-trivial and usually only the first BZ can be sketched if the shape is known before hand. On the other hand, extending the 2D code to handle the extra dimension required for 3D lattices is trivial. The results for the first, fifth, tenth and fifteenth BZ for the face-centred cubic real lattice (fcc) are shown in figure 3. A grid of  $500 \times 500 \times 500$  points contained within a centred  $6 \times 6 \times 6$  supercell was used. Points in each BZ datafile were plotted using `gnuplot` with `splot`. To emphasise the features of each BZ, a grey scale was mapped onto the radius of each  $\mathbf{k}$ -point whereby points closer to the origin are darker than those further away. This effectively shaded each surface to reveal the intricate features. The increasing complexity of each BZ shown in figure 3 is clearly evident. Students will also notice that with increasing BZ order, the effective ‘radius’ of each zone increases. Plane-cuts can also be done to show that while the first zone is a solid volume, all higher order BZ have a hollow centre. As with the 2D case, students will appreciate that each BZ is a ‘cut’ of reciprocal space so that the combined effect of all BZ taken together is to fully cover that space.

Whereas the properties of 2D BZ were easy to verify visually, equivalent properties for 3D sometimes have to be verified by indirect means. For instance, students can verify that each BZ has the same size (volume) as the first zone by looking at their respective datafile sizes. Since each file should have almost the same number of points, they should all have a very similar size. Discrepancies in file size for increasing BZ order indicate artificially



**Figure 4.** The free-electron Fermi surface for a fcc lattice with  $N = 4$ . (a) Sorted Fermi surface with second zone regions (cyan), third zone regions (blue) and fourth zone regions (black), (b) second zone holes, (c) third zone open, (d) fourth zone electrons.

distorted shapes where a file either contains an incomplete zone or a complete zone with added parts. The beginning of an increasing file size trend indicates that not enough  $\mathbf{G}$  vectors were used while decreased or zero size indicates that the ever expanding higher order BZ have extended past the boundaries of the chosen gridded volume. These simple observations show a student that for the 3D case, the same relationships found for the 2D case exist between the BZ, the reciprocal lattice points and the surrounding space.

## 5. The free-electron Fermi surface

The free-electron model is used in all introductory solid state courses to introduce students to the use of basic quantum mechanics to describe material properties. Electrons are free since no periodic potential is present in the lattice. Electronic states are populated with available electrons according to the Pauli exclusion principle until a maximum energy state is reached. For 2D, this defines a circle of fixed radius while for a 3D lattice this defines a sphere. The relationship given in Hartree atomic units [12] between this Fermi energy and the reciprocal

space  $\mathbf{k}$ -vector that defines the radii is:

$$E_f = \frac{1}{2} k_f^2.$$

Depending on the dimensionality of the lattice, this radius is also related to the valence electron volume density  $n$  as:

$$k_f = (2\pi n)^{\frac{1}{2}} \quad 2\text{D},$$

$$k_f = (3\pi^2 n)^{\frac{1}{3}} \quad 3\text{D}.$$

The Fermi surface (the boundary of the Fermi circle/sphere) at radius  $k_f$  can extend into higher-order BZ if there are enough electrons per real lattice unit-cell to define a Fermi radius larger than the outermost boundary of the first BZ. In the reduced-zone scheme, these sections of the Fermi surface are folded back into the first BZ. Students can use the sorting algorithm to paint each Fermi surface  $\mathbf{k}$ -point according to which BZ it lies in. By finding the nearest  $\mathbf{G}$  vector to each extended point in a particular BZ, the point can be folded back into the first BZ using:

$$\mathbf{k}_{\text{folded}} = \mathbf{k}_{\text{extended}} - \mathbf{G}_{\text{nearest}}.$$

For a 3D fcc lattice with  $N = 4$  electron per unit cell, the Fermi surface extends in the higher order zones with the first BZ being completely encased within the sphere. Figure 4(a) shows the sphere sectioned by the sorting algorithm to show which regions are enclosed by which BZ. The lightest colour represents those points within the 2nd zone. The darker colours successively represent those within the 3rd and 4th zones. These are then folded back into the first BZ to give the plots in figures 4(b)–(d). The reduced-zone regions from each higher order BZ define either enclosed occupied states (electrons) or empty states (holes). The reduced region from the 3rd BZ (shown in in figure 4(c)) defines an open region that is not fully enclosed by surfaces.

NOTE: The blue outlined first zones in the reduced-zone plots were created by another programme that looks for zone edges in the reciprocal lattice Wigner–Seitz cell. This can be part of a separate software project for students when they first learn about the Wigner–Seitz cell for real lattices. The algorithm for this method is not suitable for higher order BZ.

The plots generated by this method compare well with those found in upper level solid state textbooks and on-line sources. This application of the sorting algorithm complements the explanations given by these traditional sources thereby enhancing the students' knowledge and understanding of how these plots are generated and what they mean.

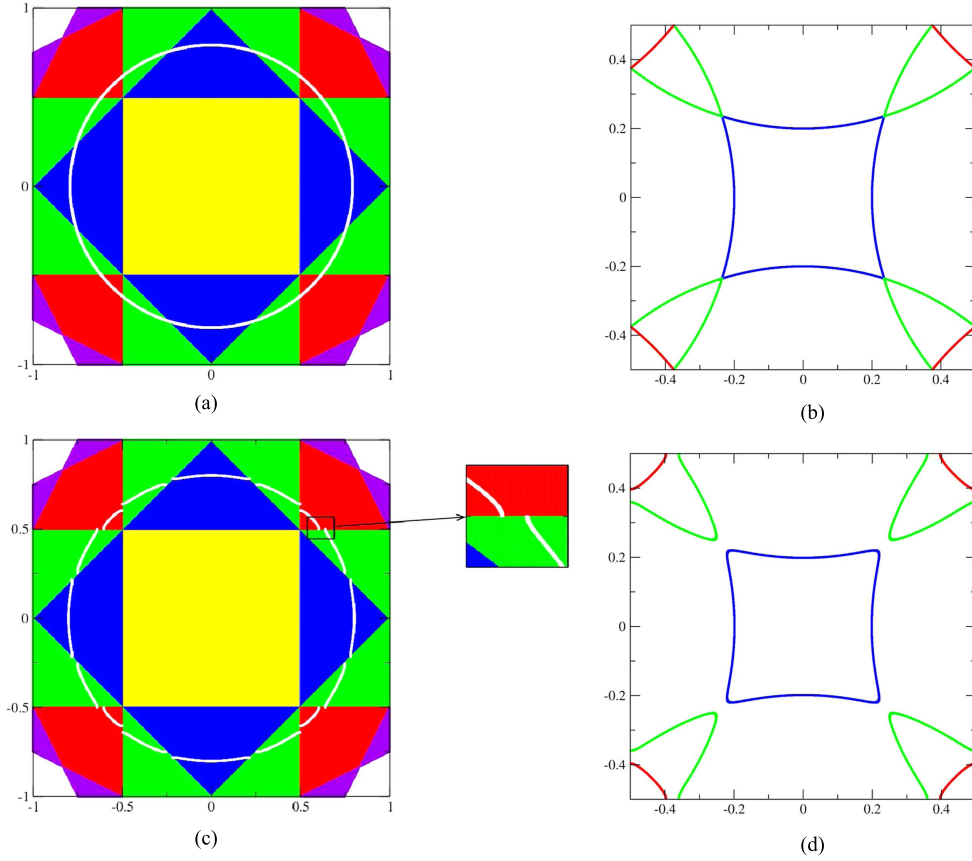
## 6. The perturbed Fermi surface

In reality, electrons in a crystal lattice experience the effects of a periodic potential  $V(\mathbf{r})$  so that the single-electron Schrödinger equation is given in Hartree atomic units as:

$$\left[ -\frac{1}{2}\nabla^2 + V(\mathbf{r}) \right] \phi_{\mathbf{k}}^n(\mathbf{r}) = \varepsilon_{\mathbf{k}}^n \phi_{\mathbf{k}}^n(\mathbf{r}). \quad (2)$$

The effect of the potential is to perturb the shape of the free-electron Fermi surface. This can be investigated using the EPM. In this method, Bloch's theorem is used to perform plane-wave expansions of both the wavefunction and the potential to transform equation (2) into a





**Figure 5.** The square lattice with  $E = 0.318$ . (a) Unperturbed (free-electron) extended-zone Fermi surface, (b) unperturbed (free-electron) reduce-zone Fermi surface, (c) perturbed Fermi extended-zone Fermi surface with expanded view of BZ crossing, (d) perturbed reduce-zone Fermi surface.

matrix equation in reciprocal space:

$$\sum_{\mathbf{G}'} \left[ \frac{1}{2} |\mathbf{k} + \mathbf{G}'|^2 \delta_{\mathbf{G}, \mathbf{G}'} + V(\mathbf{G} - \mathbf{G}') \right] U_{\mathbf{k}}^n(\mathbf{G}') = \varepsilon_{\mathbf{k}}^n U_{\mathbf{k}}^n(\mathbf{G}).$$

The rows and columns of the matrix are labelled by the set of  $\mathbf{G}$  vectors. The first term in the summation defines the diagonal kinetic terms and the second term the off-diagonal Fourier components of the periodic potential. Diagonalizing the matrix for all points in a reciprocal lattice unit cell defines the energy-band dispersion relation  $\varepsilon_n(\mathbf{k})$  where  $n$  is the band index. In practice only a finite number of  $\mathbf{G}$  vectors within a chosen cut-off radius are used. This gives a finite size to the matrix therefore a limit to the total number of energy bands (eigenvalues). It also places a limit on the number of off-diagonal terms used to describe the potential.

To find the extended-zone constant energy Fermi surface for a particular energy  $E_f$ , one just has to plot the  $\mathbf{k}$ -points that satisfy  $\varepsilon_1(\mathbf{k}) = E_f$  for the first band (the first eigenvalue). These points can be labelled by the sorting algorithm and folded back into the first zone using the method described earlier. This reduced zone surface can be directly plotted by only considering those points in any single unit cell. Here one plots the  $\mathbf{k}$ -points from that cell that

satisfy  $\varepsilon_n(\mathbf{k}) = E$  for all bands (not just the first one). Unfortunately with this direct method, the student can no longer label each point with the extended BZ it comes from.

Students can generate plots like those shown in figure 5. Here the extended-zone Fermi surface plot was made using a square lattice. The  $\mathbf{G}$  vector limiting radius satisfied  $G_{\text{cut}}^2 = 8$  which gave a  $25 \times 25$  Hamiltonian matrix. For the free-electron case, all off-diagonal terms were set to zero. To switch on the potential for the perturbed case, some of these terms are given non-zero values. Noting that the off-diagonal terms are just functions of  $|\mathbf{G} - \mathbf{G}'|$ , all off-diagonal terms in the Hamiltonian were set to zero except those for

$$|\mathbf{G} - \mathbf{G}'| = 1 \quad |\mathbf{G} - \mathbf{G}'| = \sqrt{2} \quad |\mathbf{G} - \mathbf{G}'| = 2$$

which were set to  $V(|\mathbf{G} - \mathbf{G}'|) = 0.01$

NOTE. All energy values are in units of  $\frac{\hbar^2}{m} \left(\frac{2\pi}{a}\right)^2$  which is a consequence of coding the Hamiltonian in Hartree units with the  $\mathbf{G}$  vectors and  $\mathbf{k}$ -points in units of  $\frac{2\pi}{a}$ .

The constant energy surface is for  $E = 0.318$ .

The plot in figure 5(a) clearly shows a free-electron Fermi circle. Students will note that the surfaces are continuous when they cross each BZ boundary. The reduced-zone surface is shown in figure 5(b) with its sharp edge features for the second, third and fourth zone curves. The blue curve is from the second BZ and defines an enclosed area of unoccupied states. The green and red curves enclose occupied states and are from the third and fourth zones respectively (this becomes more evident if the student considers these reduced zones in the periodic zone scheme). Figure 5(c) shows the perturbed Fermi surface in the extended-zone scheme. The curve is clearly a perturbed circle with discontinuities at the BZ edges. The expanded view shows that at these distortions, the constant energy curves are normal to the BZ boundaries. Figure 5(d) shows the reduced-zone surface. Here, the perturbation of the free-electron surface is even more evident showing a smoothing out of the sharp features of the free-electron case. These observations verify those given in typical Solid State Physics textbooks such as the one by Kittel [13].

## 7. Van Hove singularities

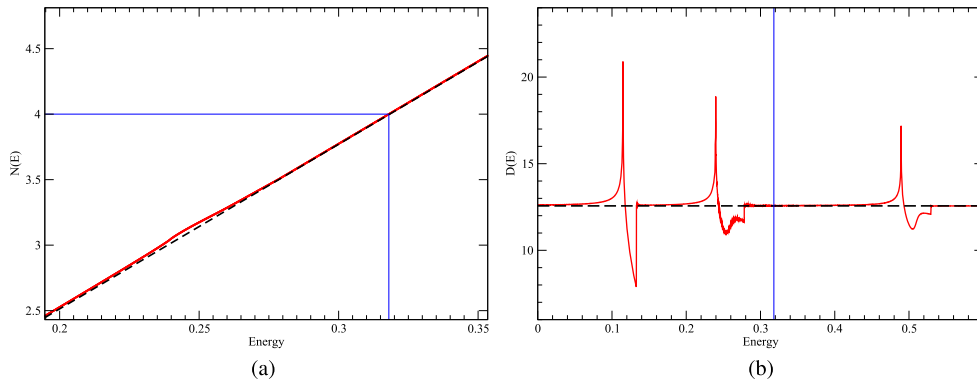
Another topic in solid state physics is the concept of Van Hove singularities in the DOS. Since these features only exist when electrons are under the influence of a periodic potential, the EPM is a very useful way for students to numerically generate them. Using the sorting algorithm, students can explore the constant energy surfaces around these singularities to gain insight into their cause.

### 7.1. The free-electron 2D DOS

Students are introduced to the concept of the DOS by considering the free-electron case. They can prove on paper that for the 2D box of size  $a \times a$  (using our energy units):

$$N(E) = 4\pi E, \quad D(E) = \frac{dN(E)}{dE} = 4\pi,$$

where  $N(E)$  is the number of electrons per unit cell when the constant-energy Fermi surface has energy  $E$  and  $D(E)$  is the DOS at that energy. It is no coincidence that a value of  $E = 0.318$  was chosen for the plots in figure 5 since this is just the case for  $N = 4$ . A



**Figure 6.** Number of electrons  $N(E)$  and density of states  $D(E)$  for the free-electron case (dashed black line) and the perturbed square lattice (solid red line). (a)  $N(E)$ —blue line at the Fermi energy for  $N = 4$ , (b)  $D(E)$ —blue line at the Fermi energy for  $N = 4$ .

numerical method has to be used to calculate  $N(E)$  and the DOS for a crystal with a periodic potential. These plots can then be compared to the free-electron case to see how they deviate.

### 7.2. The perturbed 2D DOS

Once a student has calculated  $\varepsilon_n(\mathbf{k})$  for a crystal with a periodic potential (using the EPM for example), they can numerically calculate  $N(E)$  and  $D(E)$  for a range of energy values and plot them. The numerical method to be used [7] is based on the 3D tetrahedron method [14, 15]. Students can verify this method by seeing if they can replicate the free-electron plots when all form factors are set to zero.

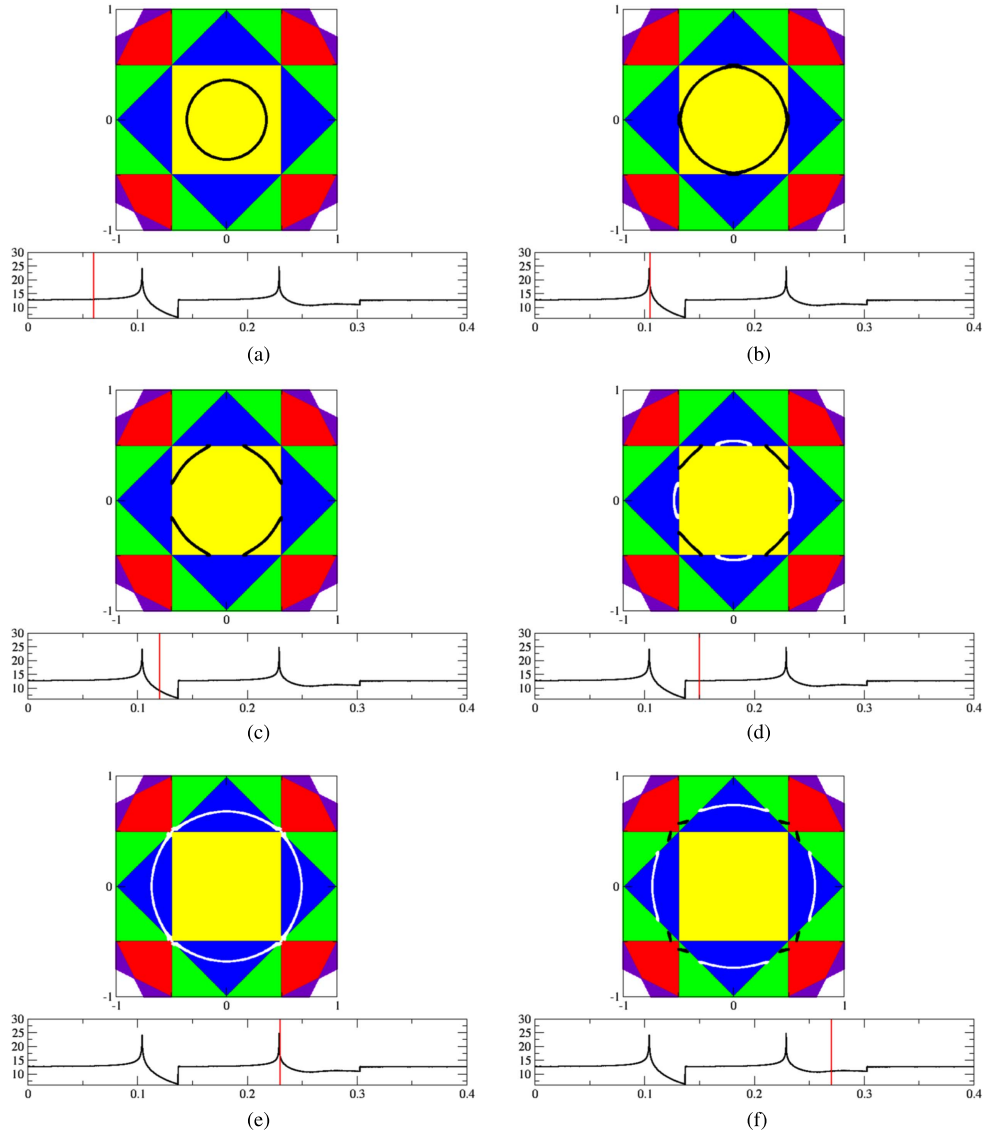
Figure 6 shows the plots obtained for a square lattice using the same parameters for the EPM (solid red lines) as before along with the theoretical free-electron plots (dashed black lines). Figure 6(a) clearly shows how the perturbed case deviates slightly from the straight line of the free-electron case. Students will note that these deviations yield slightly different energies for a particular  $N$  such as that observed for  $N = 3$ . For  $N = 4$ , the energy is  $E = 0.318$  for both cases since the lines overlap. Students will also notice that these deviations give an intricate structure for  $D(E)$  due to the fact that  $D(E)$  is equal to the derivative of  $N(E)$ . These features are called van Hove singularities and are caused by extremum  $\mathbf{k}$ -points on the constant energy curves defined by the band structure.

### 7.3. Types of Van Hove singularities

Formally, the 2D DOS is given by an integral over a constant energy curve so that:

$$D(E) = 2 \frac{A_{\text{cell}}}{(2\pi)^2} \int_{\varepsilon(\mathbf{k})=E} \frac{ds}{|\nabla \varepsilon(\mathbf{k})|}.$$

It can be easily seen that singularities will occur when the gradient of the dispersion relation vanishes at certain  $\mathbf{k}$ -points along the constant energy curve. Students can prove on paper that min (max) points give rise to upward (downward) step discontinuities while a saddle point gives rise to an upward pointing logarithmic discontinuity. These features are seen in figure 6(b). Students can devise which  $\mathbf{k}$ -points give rise to these features by plotting constant energy surfaces in the various BZ. Figure 7 shows how this is possible.



**Figure 7.** The perturbed constant energy surfaces for a square lattice illustrating the relationship between extrema  $\mathbf{k}$ -points and van Hove singularities in the DOS. (a) Continuous energy surface enclosed by the first zone, (b) continuous energy surface touching the first zone boundary, (c) discontinuous energy surface touching the first zone boundary, (d) discontinuous energy surface projecting into the second zone, (e) continuous energy surface enclosed within the second zone, (f) discontinuous energy surface projecting into the third zone.

In figure 7(a) we see the constant energy surface for an energy value just before the first logarithmic divergence feature in the DOS. The constant energy surface is completely enclosed within the first BZ. In figure 7(b) the constant energy for an energy at the discontinuity touches the BZ boundary at the saddle points  $(0, \pm \frac{1}{2})$  and  $(\pm \frac{1}{2}, 0)$ . Figure 7(c) shows the the surface becomes discontinuous along the boundary. In figure 7(d) we see that

part of the energy surface projects into the second zone (coloured white) after the step up feature in the DOS. Students can deduce that these projections begin to emanate at the same points as before and that these are minimum points. Figure 7(e) shows that the energy surface forms a closed curve enclosed by the second BZ when the sections in the second zone and the first zone meet at the saddle points given by  $(\pm\frac{1}{2}, \pm\frac{1}{2})$ . Figure 7(f) shows that the surface then becomes discontinuous extending into the third zone (coloured black).

Of course students can verify the nature of these points by taking various cuts in the band dispersion surface  $\varepsilon_n(\mathbf{k})$  to see if these points are in fact minima or saddle points at the energy values that define each singularity.

## 8. Conclusion

Our sorting algorithm provides an easy method to visualise higher order BZ for any 2D or 3D real lattice. It can also be generalised to higher dimensions. The algorithm is simple enough that an undergraduate student possessing elementary programming skills can easily implement it. In the case of 2D lattices, it complements the traditional pen and paper constructions as these can be compared to the plots generated from the algorithm output. The code for the 2D case can then be altered to include the added dimension of 3D lattices. The hands-on coding involved for this educational project ingrains the students' conceptual understanding of the construction of the reciprocal lattice and its relation to the construction of the associated BZ.

The code can be used in conjunction with other educational projects such as the visualisation of the metallic Fermi surface. This can include not only the simple case of the free-electron model but also the surface defined by energy bands empirically generated using pseudopotential form factors of the EPM. In the latter case, the effect of non-zero interaction potential terms on the shape of the Fermi surface can be investigated.

Lastly, students can see how these surfaces interact with the BZ to find the extremum  $\mathbf{k}$ -points that give rise to Van Hove singularities in the DOS.

## Acknowledgments

The authors would like to thank the National Research Foundation and the University of Pretoria for providing funding for this work. NC is grateful to the National Institute for Theoretical Physics for support.

## References

- [1] Salagaram T, Andrew R C and Chetty N 2015 *Comput. Sci. Eng.* **17** 46
- [2] Brillouin L 1930 *J. Phys. Radium* **1** 377
- [3] Ashcroft N W and Mermin N D 1976 *Solid State Physics* (London: Thomson Learning, Inc.) ch 14
- [4] Harrison W A 1960 *Phys. Rev.* **118** 1190
- [5] Hove L V 1953 *Phys. Rev.* **89** 1189
- [6] Chelikowsky J R and Cohen M L 1974 *Phys. Rev. B* **10** 5095
- [7] Lee J-H, Shishidou T and Freeman A J 2002 *Phys. Rev. B* **66** 233102
- [8] Kokalj A 1999 *J. Mol. Graph. Modelling* **17** 1761179
- [9] Kokalj A 2003 *Comput. Mater. Sci.* **28** 1551168
- [10] Shirer D L and Merrill J R 1972 *Am. J. Phys.* **40** 1704
- [11] Ashcroft N W and Mermin N D 1976 *Solid State Physics* (London: Thomson Learning, Inc.) ch 9, p 163

- [12] Hartree D R 1928 *Math. Proc. Camb. Soc.* **24** 89
- [13] Kittel C 2005 *Introduction to Solid State Physics* (New York: Wiley) ch 9
- [14] Gilat G and Raubenheimer L J 1966 *Phys. Rev.* **144** 390
- [15] Jepson O and Anderson O K 1971 *Solid State Commun.* **9** 1763