Brillouin Zones and AFLOW implementation

Jake LaNasa and Robert Hanson

January 1, 2016

Abstract

Overview and discussion of the progress and results made by the Hanson Research Group in the fall of 2015. Research specifically focused on Brillouin zones three dimensional binary alloys found within the AFLOW database. The result of the group's research was a Jmol algorithm for generating n^{th} level Brillouin zones for any general binary model.

1 Statement of Purpose

1.1 Background

The Hanson Research Group, in collaboration with the Curtarolo Group based at Duke University, started work in the summer of 2015 with the goal of enhancing visualization of the AFLOW database using Jmol and its JavaScript counterpart, JSmol.

The AFLOW database (*http://www.aflowlib.org*) is a globally accessible database of real and theoretical material properties computed though density functional theory. The database is designed as a standardized repository utilized by experimental material scientists in the discovery and construction of new materials. Jmol is an open-source, Java based tool used to visualize and analyze complex structures in 3D such as molecules, proteins, and crystals.

Through the course of the year, the Hanson Research Group has aimed to expand Jmol's capabilities in order to improve visualization and UI capabilities within the AFLOW database and enhance data accessibility within the AFLOW website. Initially, the capabilities of both Jmol and the AFLOW database were evaluated with the intent of optimizing their interaction. Once deficiencies were found, AFLOW specific macroscripts were implemented into the functionality of Jmol to enable interaction with the AFLOW API through the Jmol UI. With Jmol's improved functionality, the research group was able to construct a proof-of-concept website that was able to access, interpret, and visualize selected binary alloy data accessed from the AFLOW API at command of the user. The progress made during the summer of 2015 provides a basis for the following development conducted during the first semester of the academic year.

1.2 Brillouin Zones

Brillouin zone visualization became the primary focus of the academic year, as it had been a suggested enhancement of Jmol during the Hanson Group's collaboration with the Curtarolo Group at Duke University in July.

Physically, the Brillouin zones are used to describe and analyze the electron energy within in the band energy structure of crystals. The discrete nuclei located at the crystal's lattice points create a discrete set of potential wells for electrons in a solid, and these wells establish a countably infinite set of electronic states. The Brillouin zone is used primarily to express the discrete nature of the electronic wave function in a crystalline solid. The importance of the Brillouin zone stems from the Bloch wave description of waves in a periodic medium. Essentially, solutions to the Bloch equation can be completely characterized by their behavior inside the first Brillouin zone.

Geometrically, The first Brillouin zone is the Voronoi cell around the origin of the reciprocal lattice, Γ. To find the first zone, one must draw vectors from Γ to all the nearest reciprocal lattice points. Then, these vectors must be bisected by perpendicular planes. The smallest volume resulting from the intersection of these perpendicular planes is the first Brillouin zone.

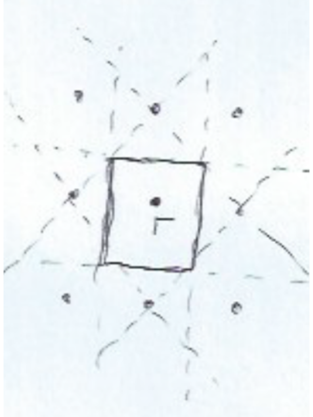


Figure 1: First Brillouin zone of a general rectangular lattice

There are also second, third, etc., Brillouin zones, corresponding to a sequence of disjoint regions at increasing distances from the origin. An interesting property of all of these zones is that they all have the same volume.

The benefit of considering higher level Brillouin zones is that it allows for harmonics within the set of wave functions – a concept very familiar to chemists in terms of molecular orbitals and excited states in conjugated systems. These are single-electron solutions to the wave equations that are equally valid but have higher and higher energies.

It is common to see band diagrams where electronic energies at particularly interesting points along designated line segments through and on the surface of the Brillouin zone are depicted. These zones appear ”folded back” into the first Brillouin zone for convenience in visualization.

While one could pack the first zone with all the pieces of the other zones, it could also be beneficial to visualize the second zone, third zone, etc. directly. A state can be selected and the energy can be visualized without folding. From this, one could plot the Fermi surface, for example, and see how it ”penetrates” into higher BZs.

During preliminary research and investigation, it was noted that there was a lack of comprehensive online or educational visualization for 2D and 3D Bravais lattice structures’ first Brillouin zone. The initial intent then, was to fill this need and provide a readily available and comprehensive visualization of Brillouin zones for all 14 3D Bravais lattice structures. Specifically, the Hanson Research Group set a goal to develop an algorithm that used Jmol’s interface to generate the n^{th} level Brillouin zone of a general lattice structure found within the binary compounds of the AFLOW database.

2 Methods

Development was segmented into three distinct phases. The first step was to investigate the characteristics of higher order Brillouin zones within the 2D Bravais lattices. From there, knowledge of Brillouin zones could be generalized and expanded to accommodate 3D lattices. Lastly, the accumulation of findings would result in an algorithm concept that could be implemented into Jmol and would generate an n^{th} level Brillouin zone for a general binary alloy model taken from the AFLOW Database.

2.1 2D Investigation

The first step of investigation was to study higher level Brillouin zones for each of the five Bravais lattices in two dimensions. The intent of 2D investigation was to minimize the complexity and understand the evolution of the zones as more points were taken into account. To do this, diagrams of 2D lattice structures were created and analyzed. Some of the sketches used are shown below.

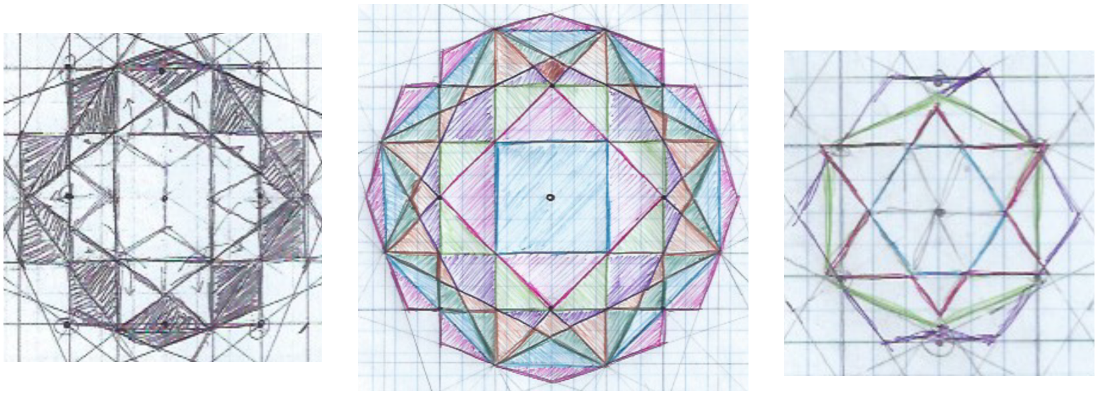


Figure 2: Right: Brillouin zones for rectangular lattice; Center: Brillouin zones for square lattice Left: Brillouin zones for hexagonal lattice

These drawings were able to effectively illustrate the interaction of a large number of lattice points and their corresponding planes. Furthermore, it was made apparent what contributed to a lattice point becoming significant to the zone, how a lattice point can be significant to multiple zones, and how subsections "folded out" from a previous zone to create a new zone. After studying these structures the group discovered corollaries that could be applicable to a computational algorithm.

A specific example of this was a selection algorithm in order to determine points of significance for the first Brillouin zone. In Jmol, it would be incredibly difficult to draw a plane for every single nearby lattice point and account for the interaction of planes. Therefore, the group needed a selection process that limited the number of lattice points taken into consideration. As a result, a circle selection algorithm was developed that determines if a random point is significant based on its neighboring points and its relation to Γ.

2.1.1 Circle Selection Algorithm

The algorithm is discussed in this section. It is by no means a rigorous proof, it lays the discusses the theory behind the algorithm which is later applied to the Jmol implementation.

We start with a central point in a periodic lattice structure, Γ. From Γ, a line segment can be drawn to every other lattice point in the structure. For these line segments, we know that a perpendicular bisecting plane (or line in 2D) would be located at the midpoint of the line segment. That is, considering "plane points" that represent the point of intersection between these line segments and their corresponding bisecting plane is the same as reducing the scale of the lattice structure by $\frac{1}{2}$. Therefore, for conceptual clarity, we are going to assume that the lattice points around Γ actually are "plane points" of the lattice.

From there, we consider another point in space α. The distance $d(\Gamma, \alpha)$ between Γ and α is defined as $\sqrt{(\Gamma_x - \alpha_x)^2 + (\Gamma_y - \alpha_y)^2}$. The selection algorithm defines $d(\Gamma, \alpha)$ to be a diameter of a circle (with radius $\frac{d(\Gamma, \alpha)}{2}$) and centered at $(\frac{\alpha_x - \Gamma_x}{2}, \frac{\alpha_y - \Gamma_y}{2})$.

From this, the algorithm determines if α is significant. What exactly makes a point significant? Recall that these "points" are the actually the nearest points to Γ on planes that stretch off indefinitely in a direction perpendicular to the segment connecting Γ and the plane P_α. Therefore, for the first zone, a plane is significant if it is not blocked completely from a direct path to Γ by neighboring planes. There must exist a point p on that plane P_α where a straight line segment can be drawn from p to Γ without being intersected by a neighboring plane.

This is where the circle comes in. The suggestion is that a point α is significant if and only if there are no points within the area of the corresponding circle.

Let's suppose that there exists another distinct point, β that is within the circle of α defined above so that $\beta \neq \alpha$. Both α and β are points on the planes P_α,P_β. These two planes are either going to have one intersection, or no intersection.

If the planes have no intersection, then by definition P_α and P_β are parallel. Because β is within the circle, and $d(\Gamma, \alpha)$ is the diameter of the circle $d(\Gamma, \beta) < d(\Gamma, \alpha)$ for all β that are within the circle, so that there cannot exist a line segment between Γ and a point p on the plane P_α that is not intersected by the plane P_β.

Note: the rest of this is not rigorous. I believe it works in a discrete space like a lattice structure but not in a continuous space. I'm not familiar enough with this type of proof to give a rigorous explanation. This is mostly speculation that I think is on the right track.

If the planes have one intersection, there is going to be a second point ω that is a translation of one of our three already defined points Γ, β, α due to the periodicity of the lattice. The planes that correspond with ω and β are going to both completely block off P_α. The point ω could either also be within the circle, or outside of the circle. Either way, I think that P_ω and P_β are going to intersect at some point ρ and the distance $d(\Gamma, \rho) \leq d(\Gamma, p)$ where p is a point on p_α. This then blocks P_α for all points $p \in P_\alpha$ so that there cannot exist a line segment connecting Γ to a point p where the line segment does not intersect either P_ω or P_β.

For the other direction, if there is no point within the circle of α, then α is significant.

Let's suppose there is a second point β so that $\beta \neq \alpha$ and β is not within the circle of α as originally defined. The point β, is again, a point on a plane P_β. Which extends indefinitely and either has one intersection with P_α or no intersection. The case that there is no intersection is a similar argument to the argument given above.

In the case that there is one intersection of P_β and P_α, we would again need a second point ω and a corresponding plane P_ω to intersect with P_β and block off P_α completely. However, due to the periodic structure of the lattice, this restricts the possible points ω to be a translation of Γ, β, α. Since there is an assumption that there is no point within the circle of α, this point has to be far enough away to not be within the circle. With those restrictions, P_β and P_ω are not going to intersect at a point ρ where $d(\Gamma, \rho) \leq d(\Gamma, p)$ for all p on the plane P_α. There will be some interval on P_α so that a line segment originating at Γ can be drawn.

While this isn't a rigorous proof, the concept seems to work and was a simple enough to expand.

2.2 3D Expansion

The next step involved expanding the knowledge gathered from the 2D investigation so that it would be applicable in three dimensions. Specifically, this materialized within the concept of the selection algorithm that determines significant points for the first Brillouin zones. Whereas in 2D, this selection algorithm involved a circle, the 3D version would involve a sphere in 3D. The rest of the concept followed similarly: a lattice point was deemed significant only if the corresponding plane point's sphere did not contain an additional plane point.

It was also during this time that we started to think about higher zones. The selection algorithm discussed so far only selects the lattice points that are significant for the first zone, and therefore needed to be expanded to higher levels. This required careful thought, as significant lattice points for the first zone can also be significant for higher zones. Essentially, in order to determine significant points, already used points would need to be included in a way that did not interfere with the selection of new significant points. To do this, it was decided that previously significant points could be temporarily taken out while new significant points were found, and added back in to the total collection after new points were determined. The result would be an appended list of points that is still efficient and can be used to determine the subzones of the brillouin zone of interest.

2.3 Jmol Implementation

Applying the knowledge gained from our investigation of 2D and 3D Brillouin zones, the next step was to expand the functionality of Jmol to create an algorithm that would generate and display a general reciprocal lattice structure's Brillouin zone of n^{th} level. This algorithm is all stored within a single script named *bz6.spt* that can be selected and dragged into the Jmol UI.

The algorithm begins by loading a binary model. Once the POSCAR file of a single model of a binary alloy is selected and loaded, Jmol then utilizes the primitive unit cell that is automatically displayed for all alloy models and generates the reciprocal lattice in order to work within reciprocal space. Therefore, all "points" referred to in this section are to be understood as points in reciprocal space.

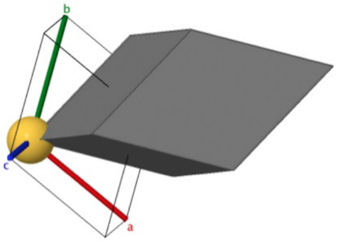


Figure 3: Lattice unit cell and reciprocal lattice vectors

From there, the script executes the function *createAllBzs(n)* in the command line of Jmol, where n is the desired level of Brillouin zone. This function calls more functions that set the generation in motion. Essentially, the function creates an array that contains a range of 3D reciprocal lattice points and a second array to store each lattice point's corresponding midpoint with respect to the central point, Γ. From there, *createAllBzs()* starts with the first Brillouin zone and works its way out to the n^{th} Brillouin zone. A new structure is allocated to store information pertinent to each respective zone and is pushed to a zone array. From there, the function *createBZ(zone,zone-1)* goes through the entire zone array and builds up to the desired zone.

The function *createBZ()* is the heart of the algorithm, as it is what is continually called to generate each individual zone. As the function progresses through the array of zones, it builds the current zone from the preceding zone. The function determines all the significant lattice points for a desired zone by looping through all possible points and using the half-distance sphere test discussed earlier. As all significant points are found, the global lattice and plane points are redefined to contain only the remaining points that have not yet been used. Once *createBZ()* has all the new lattice points, the function draws the zone centers, which are points located on the bisecting plane of the significant lattice points, for later use.

Next, the function begins to determine the subzones or faces that will potentially compose the intended Brillouin zone. If the Brillouin zone being drawn is the first zone, new subzone structures are created and initialized from the zone planes determined earlier. However, if the Brillouin zone is higher level, the function must also consider all subzones that were considered in generating previous zones. Specifically, each subzone of the previous zone possesses a set of planes. If the previous zone was the first zone, the function uses all previous planes. Otherwise, the function always skips the first plane in the set (as it originated two zones prior). Now, each of these planes is a starting point for a new subzone. The initial plane of the new subzone will have its normal vector pointing outward from its subzone, which will mean that the plane's normal will be negative to the normal of the previous zones plane. Once these initial planes are determined, all of the previously used, unused, and new planes are added without inversion for consideration.

From this, the function *createBZ()* will loop through all of the planes to create a pmesh (composition of triangles) for each face of the zone. The function uses a resolution of .001 for the pmesh so that the minimum number of triangles are used to build pmesh plane. Since we are dealing with planes, is unnecessary to have a pmesh with high resolution. Next, the function uses the slab command in Jmol and slabs each plane by every other plane. Slab will treat the intersection of two planes as a new boundary to the plane the slab is being applied to. After all planes are slabbed, the remaining pieces of planes form the faces of the zone.

From these slabbed pmesh faces, the function then creates subzone polyhedron. The function creates the polyhedron originating at Γ and extends to edges of these faces. After this, the zone is finalized, which involves removing redundant sections, calculating the zone and subzone volumes, and calculating the symmetry of the zone.

3 Results

3.1 N^{th} level Brillouin zones

The initial purpose of this investigation was to provide a way to visualize Brillouin zones for all Bravais lattices and view higher level zones. The algorithm implemented into Jmol accomplishes this. Furthermore, this algorithm is designed to be fully compatible with the POSCAR files stored in the AFLOW database for binary alloys. As seen in the images below, Jmol can generate higher level zones for any general binary alloy found within the AFLOW database. Animation and comparison of multiple zones enable the user to see the zone evolution. Once the n^{th} zone is generated, user input can instantly display any of the $n-1$ zones preceding it, as each successive zone is dependent on the generation of previous zones. All the user needs to do is define the boolean *showbz n* as true or false. As each zone covers its predecessor, it is important for visualization purposes that the higher level zones are turned off in order to visualize the zone of interest.

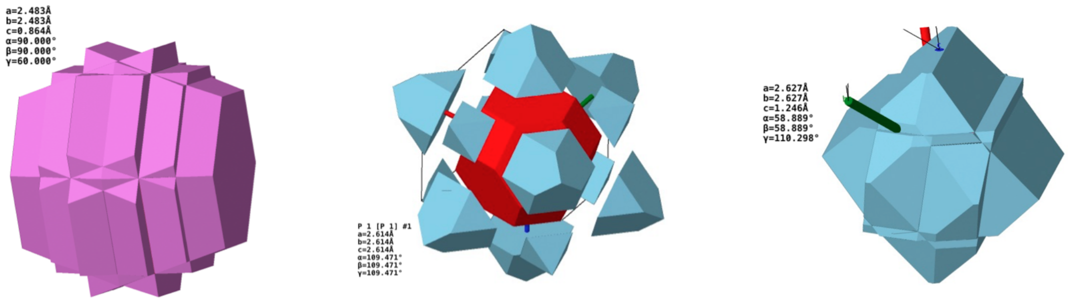
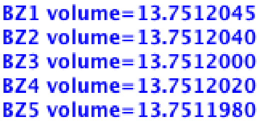


Figure 4: 3rd Brillouin zone of hexagonal lattice ; Center: 1st and 2nd Brillouin zones of the face centered cubic lattice ; Left: 2nd Brillouin zone of monoclinic lattice

Furthermore, the polyhedra features of Jmol developed over the summer and expanded during the academic year gives the algorithm added flexibility in calculation in visualization. From the polyhedra in Jmol, we can calculate the total volume of each subzone and sum the polyhedra together to determine the total volume of each zone. As predicted by theory, the volume for each zone should remain constant as n increases. Volume then, became a convenient and reliable quality of each zone that could be used to verify the accuracy of each zone. If the algorithm ignored a significant point, or added an insignificant point, volume would be altered significantly. However, as can be seen below, for a crystal composed of 100% Ag, Brillouin zone volumes through the fifth zone agree with the initial volume to within 4 significant digits.



```
BZ1 volume=13.7512045
BZ2 volume=13.7512040
BZ3 volume=13.7512000
BZ4 volume=13.7512020
BZ5 volume=13.7511980
```

Figure 5: Volume calculations for the first model of AgAu POSCAR file

These volume calculations are an inherit part of the algorithm, and are iterated for each volume of each alloy model the algorithm is applied to. As the algorithm finishes each zone, it will print out the volume of every finished zone thus far within the Jmol console. Testing of the algorithm on multiple binary files has resulted in consistent volume calculations similar to the one shown above. The small deviation between volumes can be attributed to the limits of float numbers during generation.

3.2 packBZ()

Jmol actually can illustrate the conservation of volume with a unique feature called *packBZ()*. If the volumes of zones are indeed conserved, the subzone polyhedra of a particular zone should be able to fit inside the confines of the first zone. Using the command *packBZ()*, the user can actually pack the second Brillouin zone into the first zone. The user can either do this by selecting individual subzones to be translated to its position within the first zone, or the entire process can be completed in one step (see images below).

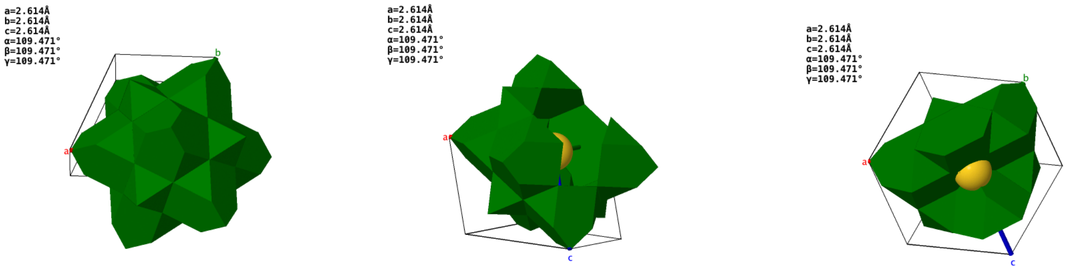


Figure 6: Right: Unpacked second Brillouin zone; Center: second zone with polyhedra translated into the first zone; Left: continuation of the packing with Γ represented by the yellow sphere

As the images illustrate, the second brillouin zone can be packed into the first brillouin zone entirely by translating subzones across the lattice. The final result of this will be a set of subzone polyhedra packed into an identical structure as the first zone for the lattice. As evidence by the images below, the second zone's final packed structure looks identical to the lattices first Brillouin zone shown in red.

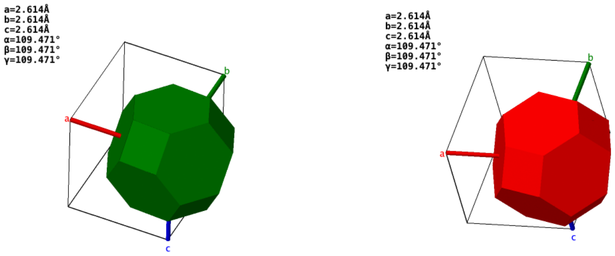


Figure 7: Left: Fully packed second zone of first model of AgAu POSCAR; Right: First Brillouin zone of first model of AgAu POSCAR

While this can be done for the second Brillouin zone, Jmol does not currently have the capability of using the *packBZ()* function for higher zones. This is because the complexity of zones increases as n increases. For the second zone, polyhedra are only translated across the lattice. As n increases, subzone polyhedra would need to be inverted, rotated, translated, split apart, etc. To implement this function to higher zones would require a packing algorithm that solves each specific zone for any general lattice.

3.3 Areas of Improvement

As of right now, there are a few areas where the current algorithm could be improved. For one, as the user asks for higher level zones, the computation time increases substantially. This is because the current version of the algorithm takes all previous planes used into consideration as it slabs pmesh faces. As the algorithm progresses to higher zones, the number of previous planes goes up as well. This could be improved, since realistically it is unnecessary to consider every single plane as some planes will be parallel or will slab far enough away where to be inconsequential to the final pmesh face. A reworking of the higher level selection process that limits the number of planes to be slabbed could decrease the time of computation. However, it is not absolutely imperative that this improvement be made, as the algorithm can generate up to 6 zones within a minute, and there isn't much practical need to go higher to higher zones from that. For implementation on a website however, this may be too slow as users tend to click through websites at a very fast pace will not want to wait for a minute for a web page to load.