Abstract

Department of Chemistry Degree in Bachelor of Arts

Structural Changes in Oxygen-Deficient Double Perovskite, $Sr_2CaSbO_{5.5}$

by Megha Patel

In order to develop a more efficient cathode oxygen vacancy conduction fuel cell, it is important to understand the oxygen vacancy structure and conduction in perovskites and double perovskites. Previous research using a neutron pair distribution analysis looked at double perovskite Sr_2CaSbO_5.5 and found that the local geometry changes significantly when oxygen vacancies are introduced across multiple structures. Here, we looked at energetically different computationally generated structures by employing methods like density functional theory (DFT) using a PBE functional and a generalized gradient approximation (GGA). These methods were implemented in Vienna ab initio simulations package (VASP) to obtain final configurations and their energies. To obtain the lowest energy final configuration, the conjugate gradient (CG) method was utilized. First, Glazer octahedral tilting was performed on cubic Sr_2CaSbO_6 to generate 23 possible distorted structures. The minimum energy configurations of these structures were found using CG. Then, to the lowest energy Glazer, oxygen vacancies were introduced into the system with either no distortion, a trigonal bipyramid distortion around Sb and Ca ions, a 45 degree rotation around SbO_5 or CaO_5, or 45 degree octahedral rotation on SbO_6 or CaO_6. All resulting structures were optimized on VASP. The global minimum energy configuration was determined to have trigonal bipyramidal geometry around the Sb ion. The edge of the SbO_5 polyhedra lined up with the edges of the polyhedra around the M ion. Lastly, molecular dynamic trajectories on the Born-Oppenheimer surface showed no conduction or rotation at 1200K for 2000 fs.

Acknowledgements

One my greatest hesitations was to join a research lab, let alone a theoretical chemistry lab. I reached out to Professor Maria Gomez, who is now my current thesis advisor, for research my freshman year in hopes to see if research was a field for me. I remember starting in her lab first year and struggling the entire time. At multiple points in my semester of research, I questioned if research was for me. Throughout this time, Professor Gomez continued to be patient and helped me learn from my mistakes. By my second year, I fell in love with research and chemistry thanks to her. She has continued to be my inspiration as she has influenced my love for solving for the unknown and for teaching.

Without Jiayun Zhong, this project wouldn't have been possible. Through her, we learned the value of good teamwork. I would also like to thank Konrad Gomez-Haibach, and Fred Haibach for helping out in our group publication and for the lovely group outings. I also greatly cherish the times spent with the group: Jiayun Zhong, Shiyun Lin, and Janey Lin. There were a lot of good laughs, especially in South Carolina.

Most importantly, I want to express my warmest gratitude to my friends and family. They have continued to support me, no matter the endeavor and have had to hear me vent about thesis writing constantly.

Funding Resources

The research was supported by the National Science Foundation under grant DMR-1709975 and the Mount Holyoke College Department of Chemistry. Computational resources were provided in part by the MERCURY consortium under NSF grant CHE-1626238.

Contents

List of Figures

List of Tables

Abbreviations

CG	Conjugate Gradient
DFT	Density Function Theory
GGA	Generalized Gradient Approximation
M4	M-centered Tetrahedron
M5	M-centered Square Pyramid or Trigonal Bipyramid
M(6,10)	M-centered Pentagonal Pyramid
M(6,11)	M-centered Pentagonal Bipyramid w/ one base corner missing
M(6,12)	M-centered Octahedron
M7	M-centered Pentagonal Bipyramid
MD	Molecular Dynamics
PAW	Projector-Augmented Wave
PBE	Perdew Burke Ernzerhof
PDF	Paired Distribution Function
Sb(5,8)	Sb-Centered Square Pyramid
Sb(5,9)	Sb-centered Trigonal Bipyramid
SE	Schrodinger Equation
SOFC	Solid Oxide Fuel Cells
VASP	Vienna ab-initio Simulation Package

Chapter 1

INTRODUCTION

1.1 PEROVSKITES AND USES IN CATHODES

In the search for an efficient but sustainable energy, it is important to look at the means for generating energy in the most effective manner. Thus, we turn to look for more efficient fuel cells, devices that generate energy via chemical reactions. These chemical reactions take place at the electrodes of a fuel cell. Although there are many types of fuel cells, solid oxide fuel cells (SOFCs) which generate electricity efficiently, cleanly, and with low pollution rates. In addition to this, their modularity, reliability, and low emission rates of NO_x and SO_x make SOFCs a more efficient as opposed to classical energy generation systems. SOFCs contain two porous electrodes. The cathode supplies oxygen which then reacts with all the electrons that are from the external circuit, forming many oxide ions. These oxide ions move towards the anode. Once at the anode, the oxide combines with H_2 or CO to form either H_2O or CO_2. This frees the electrons causing the electrodes to migrate via the external circuit to the cathode.[3] A diagram of an SOFC can be seen in Fig. 1.1.

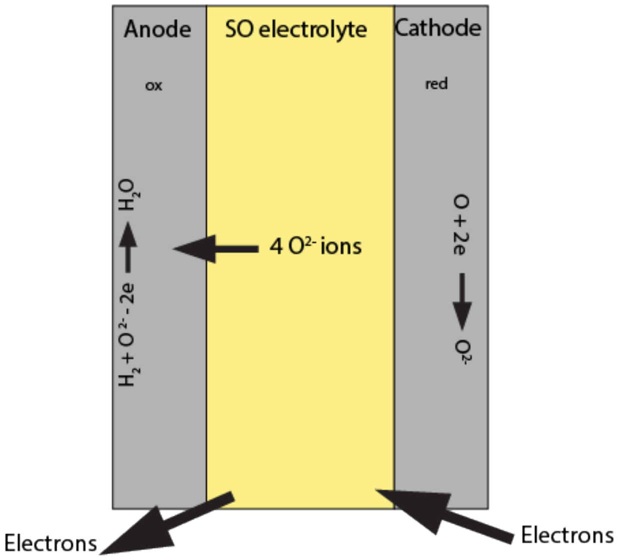
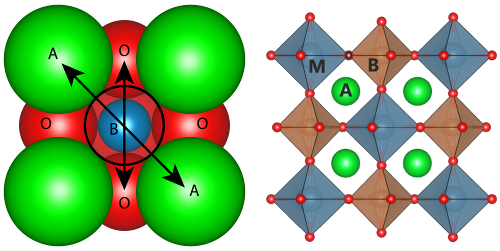


FIGURE 1.1: Diagram of a solid oxide fuel cell showing the movement of oxygen ions from the cathode to the anode.



(a) (b)

FIGURE 1.2: (a) shows a space filling model of a single unit of a ABO_3 perovskite. The oxygen ions around the B ion form an octahedron. (b) shows a double perovskite A_2MBO_6 in polyhedral view to highlight the octahedron around B and M ions. The A ions are in green. The B ions are in brown. The M ions are in blue and the red ions represent oxygen anions forming an octahedra around the M and B ions. In this study, M=Ca and B=Sb.

Conducting perovskites make great cathode materials because they have suitable electrical conducting properties, structural and chemical stability even at high temperatures.[4] Perovskites shown in Fig. 1.2 are crystal structures with a cubic unit cell. In this unit cell, the twelve O anions surround the A cation, which is located at the eight corners of the cell. The B cation is located at the center of the cube and surrounded by 6 O anions.

Perovskites are shown to be chemically stable within the fuel cell environment. Mixed ionic-electronic conductivity in oxides has been discovered to be highly helpful for the use of cathodes in fuel cells. How the material used for the cathode performs is fundamentally crucial to how the fuel cell as a whole operates. Thus, there has been a recent surge in the amount of time expended on developing suitable perovskite materials. In order to understand ionic conductivity, we need be able to distinguish where the conduction ions are located and the initial energy requisite for conduction, requiring us to find out the coordinates of each species.

Double perovskites have an A_2MBO_6 structure, where there is a network of octahedra of BO_6 and MO_6 that alternate. The six vertices of BO_6 octahedron are connected to six vertices of adjacent MO_6 octahedra via shared oxygen ions, while the A ion sits at larger twelve coordinate sites between eight octahedra. The front four octahedra for each A ion are shown in Fig. 1.2 (b). The double perovskite of interest in this work is Sr_2CaSbO_6 since studies have shown that this perovskite goes through phase transitions at both low and high temperatures. This double perovskite formula is charge balanced when A ions have +2 valency and M and B ions have +4 valency since each oxygen ion has -2 valency. However, $Sr_2MSbO_{5.5}$ with M=Ca and Sr has +2 valency for A and M ions in addition to Sb^{5+} ions. As a result, only 5.5 oxygen ions are needed to balance the positive charge.

1.2 FORMATION OF DISTORTIONS

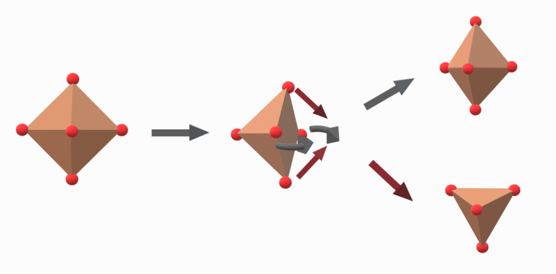


FIGURE 1.3: To screen the central ion charge, pairs of oxygen ions near the oxygen vacancy can move toward the vacancy transforming the SbO_5 or MO_5 polyhedron from a square pyramid into a trigonal bipyramid. The red spheres are oxygen ions and brown were Sb or M centered polyhedra.

In order to understand the reduction of oxygen at the cathode as well as the movement of the oxygen, studying structures that lack oxygen anions is critical. In the past, studies were performed where an oxygen vacancy is created and conduction is studied by oxygen anions moving from one vacancy to another, but because these studies were centered around single perovskites, there was little distortion around the site of the oxygen vacancy. But in certain instances perovskites with vacancies significantly distort the structure. These distortions can be structural changes on the oxygen deficient polyhedra around the central atom.

A neutron pair distribution analysis was performed to study the geometry of three double perovskite systems, $Sr_2MSbO_{5.5}$(M=Ca, Sr, Ba) and found that the geometry significantly changes upon the introduction of oxygen vacancies [1]. Specifically, when a vacancy is created, neighboring oxygen ions move to shield the positive charge of the central cation, converting the geometry from an octahedron to a trigonal bipyramid as can be seen in Fig. 1.3. Another significant change found in the study involved a 45 degree rotation of SbO_6 octahedron. These changes cause the oxygen ions to move so that pentagonal bipyramids in the M centered polyhedron can form.

To date, there has not been a computational study to verify the implications of the pair distribution analysis of the neutron scattering analysis to perovskite geometry around vacancy sites. These geometric changes upon adding a vacancy open up new potential pathways for oxygen ions to move, allowing for conduction to occur. Computational verification would open the way to including these new pathways.

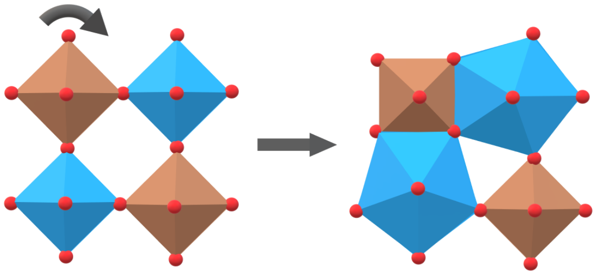


FIGURE 1.4: The 45 degree rotation of the SbO_6 polyhedron could lead to edge sharing between M centered polyhedra and Sb centered polyhedra forming M centered pentagonal bipyramids. A vacancy in the MO_7 polyhedron can be either at the pyramid apex forming a pentagonal pyramid or at one of the base corners forming a very distorted octahedron. The red atoms were oxygen atoms, blue were M-centered polyhedra, and brown were Sb centered polyhedra.

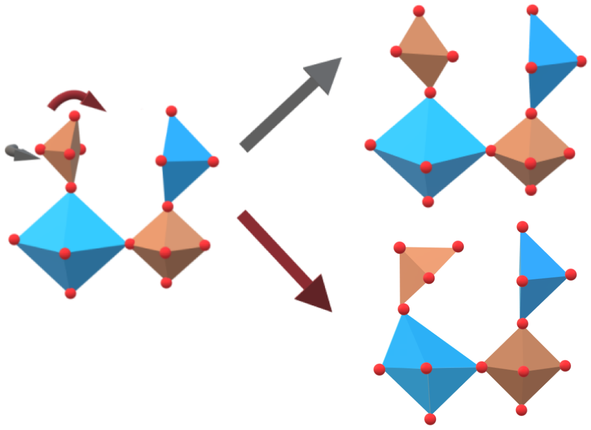


FIGURE 1.5: The 45 degree rotation of the SbO_5 polyhedron along two axes could better share the SbO_5 oxygen ions with the adjacent MO_5 polyhedron.

In this study, the double perovskites of interest were inspired by past experimental studies done on $Sr_2CaSbO_{5.5}$ [1]. Optimizations were performed to obtain low energy configurations after vacancies were introduced in order to shed light on the most likely geometric arrangements. This work will start by talking about methods used to obtain ground state configurations in Chapter 2.1 and then go onto explain how those methods were used to generate structures in Chapter 2.3. Computational methods to obtain transition states and how they are used to explore conduction pathways is discussed in Chapter 2.4. Resulting configurations of the structure generation will be analyzed in comparison to the structures of neutron pair distribution analysis for the $Sr_2CaSbO_{5.5}$ as will be seen in Chapter 3.

Chapter 2

METHODS

This chapter will discuss electronic methods for obtaining minimum energy configurations, methods used to implement distortions in the $Sr_2CaSbO_{5.5}$, and additional computational methods used to study conduction. Several distortions were created in the system were optimized by using electronic methods in order to find minimum energy configurations. This chapter will start by discussing how the potential energy surface for the nuclei to move on is found, specifically the Born-Oppenheimer approximation and density functional theory. Then, movement of the nuclei along the energy surface will be explored by primarily focusing on the conjugate gradient algorithm as a means for optimizing configurations. Specific parameters determine the electronic structure for optimization using conjugate gradient method. These parameters will be explained. Next, structure generation starting from a perfectly cubic Sr_2CaSbO_6 system will be talked about, wherein each structure is optimized to relax into a minimum energy configuration. Lastly, molecular dynamics will be discussed in how it was used to effectively study oxygen vacancy motion for the minimum energy configurations generated.

2.1 Finding the potential energy surface

In order to find ground energy states of configurations of the $Sr_2CaSbO_{5.5}$ system, the Schrodinger equation (SE) must be solved within a set of approximations. The equation is shown in Equation 2.1, where $\hat{H}$ is a Hamiltonian operator or total kinetic and potential energy of the system as shown in Eq. 2.2, E is the energy eigenvalue of the Hamiltonian, ψ is the system wave function which contains the information to describe the system. [5]

$$\hat{H}\Psi = E\Psi \tag{2.1}$$

$$\hat{H} = -\frac{\hbar^2}{2m_e}\sum_{i=1}^{N_{elec}}\nabla_i^2 + -\frac{\hbar^2}{2}\sum_{i=1}^{N_{nuclei}}\frac{1}{m_i}\nabla_i^2 - \sum_{i=1}^{N_{elec}}\sum_{j=1}^{N_{nuclei}}\frac{Z_j e^2}{|\overrightarrow{r_i} - \overrightarrow{R_j}|} + \sum_{i=1}^{N_{elec}}\sum_{j>i}^{N_{elec}}\frac{e^2}{|\overrightarrow{r_i} - \overrightarrow{r_j}|} \tag{2.2}$$

The first two terms in the Hamiltonian contain the kinetic energy of the electrons and nuclei, respectively. The remaining terms describe the potential energy for electron-nuclei interaction. The second term describes one electron attraction to the nucleus, and the third term describes inter-electronic interactions. The second and third terms have the position based vector of $\hat{H}$. R and r represent vector positions of nuclei and electrons, respectively. This section will begin by exploring two methods, the Born Oppenheimer approximation and density functional theory, used to approximate the solution to the Schrodinger equation.

2.1.1 Born Oppenheimer approximation

The Born Oppenheimer approximation to SE aids in finding precise ground energy states as a function of nuclear coordinates. This approximation is based on the fact that nuclei have a significantly larger mass compared to electrons. Therefore, electrons are able to move much faster and with greater ease compared to the nuclei. These drastic differences allow us to separate the SE into two separate SE - one for electrons and one for nuclei.[5] The electronic SE can be solved for fixed nuclear positions. The energy eigenvalue obtained plus the nuclei-nuclei potential energy is the potential energy for that nuclear configuration. Solving the electronic SE for many nuclear configurations allows a mapping of the potential energy surface on which the nuclei move. In our study, nuclei move classically and we will look for minima on the potential energy surface to characterize the most likely nuclear configurations.

2.1.2 Density functional theory

Density functional theory (DFT) is an alternative approach to approximating the solution to SE [6]. When F (the universal functional) is added to the potential energy term, the ground state energy of the system is obtained. But since F can not be exactly determined, approximations must be made by either a generalized gradient approximation or a local density approximation. Generalized gradient approximation (GGA) is more realistic than the local density approximation because it uses the electron density at each point as well its gradient. GGA also includes changes in exchange correlation energy at each point. This allows the approximation to determine a correlation energy between the electron density and the ground state energy. Here, Perdew–Burke-Ernzerhof (PBE) functional was used along with projected augmented waves (PAW) and a plane wave basis set to expand electronic wave functions. A maximum cut off energy was chosen to limit the plane wave used.

Vienna ab Initio Simulation Package, or VASP, is a package that uses projector-augmented wave (PAW) method and a plane wave basis set to describe interaction between ions and electrons and allows full stress tensor and forces to relax atoms into their ground-state configurations. [7] VASP can also be used for ab initio quantum mechanical molecular dynamics using pseudopotentials. [8] In order to do this, VASP implements density functional theory (DFT), which allows for calculations of the potential energy at a high accuracy. These calculations rely on the Born-Oppenheimer approximation to find the potential energy surface. Once this surface is established, VASP then allows for the positions of the atoms to change so that a lower energy configuration is obtained. The next section explores the actual movement along the energy surface to obtain minimum energy configurations.

2.2 Moving along the potential energy surface

Once a potential energy surface is built as described above by using the Born Oppenheimer approximation to map out the potential energy at each of the nuclear positions, we can then turn our attention to moving along the surface by determining the changes in energy between one configuration on the surface and another. Because waves expand in solid, it becomes possible to approximate changes in solid particles. In the past, these approximations of energy changes were done by implementing a trial move and determining if it is lower in energy. If it is lower, it will continue down that path until there are no trial moves that result in an even lower energy. Thereby, a decision is made about which direction to go based on current input that changes with each iteration. At this point, an energy minimum is found.[9] Because of the several recursive iterations to obtain a single minimum energy state, this method has been deemed ineffective. A more effective method is the conjugate gradient (CG) method.

The CG method serves to be more effective because it is not only making a decision about a future step based on the new gradient, but it is also storing and using information about the direction that was taken in the previous step.[9] This way, the search for the direction to take to lead to a step that has a lower energy, is more efficient. In this study, the conjugate gradient method was used by VASP.

2.2.1 Parameters and obtaining accurate calculations

For the optimization of the smaller 2x2x2 system, a 2x2x2 Monkhorst-Pack k-points mesh was used to create the plane wave basis, and the energy cutoff was set at 600 eV, a value higher than the default because of the higher accuracy. A global break condition of $1 * 10^{-4}$ was used for the electronic SC-loop. The maximum number of ionic steps was set to 400. For the optimization of the larger 4x4x4 replicated system along with the PDF configurations, a 1x1x1 gamma k-points mesh was used to create a plane wave basis. The same energy cutoff was used since it proved to be accurate even in the replicated system and the same global break condition was used. However, these structures were optimized using a single gamma point,

2.3 Structure Generation

This section will go through the various distortions that were implemented by first talking about octahedral tilting, then go into adding vacancies and rotating octahedra. Finally, this section goes into the replication of the system. The parameters used at each stage will also be discussed.

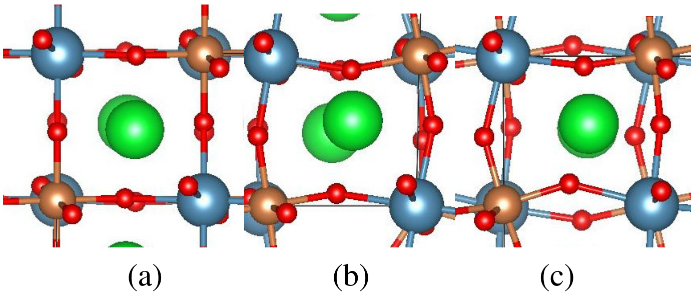


FIGURE 2.1: Structures were given labels based on the consistency of the tilting directions between two layers of octahedra adjacent to each other. (a) shows no tilting, (b) shows the result when two octahedra are tilted in the same direction, and (c) shows the result when two octahedra are tilted in opposite directions.

2.3.1 Octahedral Tilting

Lattice distortions on the 2x2x2 were first implemented by tilting octahedron. Tilting distorts the octahedra around the central atom by moving the cation, allowing for oxygen to take up that extra space. There are 23 different possible distortions in 2x2x2 system that come from tilting. Tilting may occur when each octahedron tilts in the same direction or in different directions. Glazer labels can be assigned to categorize the different tilting patterns. A $+$ label indicates that the top and bottom octahedra tilted in the same direction, while a $-$ label indicates that they tilted in opposite directions. A 0 label indicates that no tilting has occurred, and that the lattice is still perfectly cubic.

Figure 2.1 shows how the different labels look like structurally. All 23 configurations were optimized using VASP to allow for atom positions and lattice parameters to relax into their most energetically favorable state. They were optimized with a 0.01 eV Å and a 0.1 scaling on constant forces.

2.3.2 Normal mode calculations

Normal mode calculations were performed on optimized structures to ensure that the configurations is at a real energy minimum. In this study in particular, normal modes analysis was done on all 23 possible Glazer distortions.

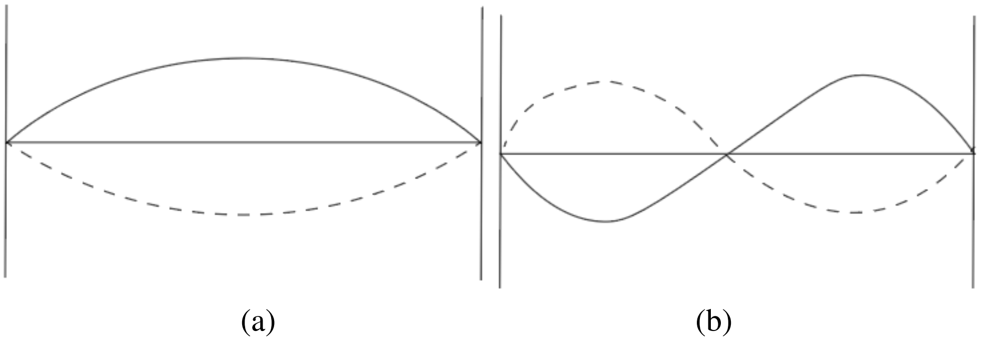


FIGURE 2.2: The first two normal modes for a vibrating string are shown. (a) has no nodes and (b) has one node

In order for this calculation to be performed, harmonic oscillation has to occur at potential energy wells to reach to a saddle point, a maximum point on a reaction coordinate diagram. Each normal mode is in correlation with the vibration spectrum absorbed. Figure 2.2 shows the first two normal modes. Where N is number of particles and 3 is the number of degrees of freedom, the modes of linear molecules are described by 3N-5, whereas the modes of nonlinear molecules are described by 3N-6[10]. During each oscillation, the molecule passes through the equilibrium, which is denoted by the horizontal line running through the graph in Figure 2.2.

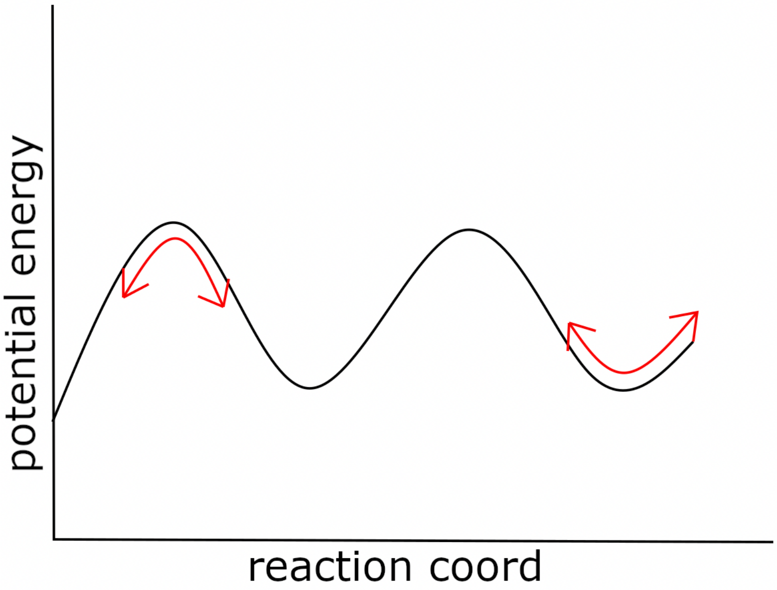


FIGURE 2.3: A reaction coordinate diagram is shown. The arrow is showing harmonic oscillation that is needed to get from a energy well to a saddle point.

Components of the dynamical matrix can be calculated by taking the second derivative of the total potential energy. These eigenvalues of the matrix describe oscillation frequencies while the eigenvectors of the matrix describe the displacement of the atoms. When an eigenvalue is negative, an imaginary frequency has been reached and following the imaginary frequency can lead to movement out of the potential energy well to reach a saddle point. A diagram of the reaction coordinate diagram is shown in Figure 2.3. If the normal mode analysis revealed a curvature in any direction characteristic of a maximum instead of a minimum, the resulting highest imaginary frequency mode is followed to and the structure is re-optimized until any remaining imaginary frequencies are less than 120 cm^{-1} or 0.01 eV. In this scenario, the imaginary frequencies which are indicative of concave down curvature have energy that lower than the accuracy cutoff and thus not significant.

2.3.3 Adding Vacancies and Rotating Octahedron

Once the lowest energy Glazer configuration had been decided, an oxygen was removed at each of the possible sites, thereby creating an oxygen vacancy at that site. The oxygen vacancy was introduced in several ways. The first way involved removing the oxygen and leaving the surrounding atoms undisturbed. The second way involved creating a vacancy and rotating a SbO_5 octahedron with the vacancy by 45 degrees towards the site of the vacancy in two ways from two different axes. The same thing was done but on the MO_5 octahedron, containing the vacancy. This rotation can be better visualized in Figure. 1.5. Another way a vacancy was added was by rotating a SbO_6 octahedron without a vacancy by 45 degrees for the x, y, and z axes. The same thing was done to the MO_6 octahedron without a vacancy. As can be seen in Figure. 1.4 The final way a vacancy was created was by moving pairs of oxygen ions closer to the oxygen vacancy in two different ways, thereby converting the surrounding SbO_5 or MO_5 from a square pyramid to a trigonal bipyramid. All these structures were once again optimized using VASP to allow for atom positions and lattice parameters to relax into their most energetically favorable state by employing an energy cutoff 600 eV on a 2x2x2 Monkhorst kpoint grid. An additional vacancy is added according to the same methods as described before to now form $Sr_2CaSbO_{5.5}$.

2.3.4 Expanding system and PDF configurations

King[1] had provided sample electronic configurations for $Sr_2CaSbO_{5.5}$ and $Sr_3SbO_{5.5}$ systems, which were created using a neutron pair distribution function (PDF) analysis, where a sine Fourier transform is taken of the whole scattering pattern. This yields the radial distribution of inter-atomic distance, and using that, the scattering data was analyzed. Rietveld refinements were done to determine the site location of different ion species. Visualization of the two different structures that had come from the PDF analysis [1] revealed that both structures had a robust pentagonal motif around the calcium. However, one of the structures had 56 percent of its Sb atoms taking on a trigonal bipyramidal shape in conjunction with a much lower percentage of pentagonal pyramids around Ca ions. The two structures are shown in Figure 3.7.

In order to properly compare the structures generated in this study, these initial given structures were also optimized with a 600 eV energy cutoff first under fixed cell shape and volume (ISIF=2)[7], then under relaxing conditions (ISIF=3), and once again optimized under fixed conditions (ISIF=2). All final structures were analyzed for composition of various polyhedral shapes.

Two distinct low energy structures for the computationally generated system were determined for both M=Ca and M=Sr. But for M=Sr, because these structures are to eventually undergo oxygen vacancy motion and if the simulation area is not large enough, the motion of the atoms would be compromised. Thus, it was determined that the original 2x2x2 system had to be replicated two times in the x, y, and z directions to create a 4x4x4 system, which is eight times larger than what it was previously. To this expanded structure, 32 oxygen atoms had to be removed instead of two atoms in order to fashion a $Sr_2CaSbO_{5.5}$ system because the system is larger. Then, the SbO_6 polyhedra were rotated to give rise to the pentagonal shape.

These 4x4x4 distorted M=Sr structures and the 4x4x4 PDF structures were optimized with a energy cutoff of 600 eV, first with a fixed cell shape and volume, then optimized again so that atoms are allowed to relax, and lastly optimized once more with a fixed cell shape and volume.

2.4 Oxygen vacancy conduction

Once the minimum energy configurations containing significant structural distortions were found after generation and optimizations, potential oxygen vacancy motions were considered using the nudged elastic band method as well as molecular dynamics to shed light on oxygen vacancy conduction.

Because molecular dynamics mimics what atoms do in real life, ab-initio molecular dynamics was also used to not only locate potential sites of conduction, but to also trace a potential conduction pathway. A given potential energy function is needed to run a molecular dynamics simulation because it takes information on the coordinates of other atoms to determine the force an atom is experiencing. We can then use Newton's laws to characterize the motion that an atom undergoes given the forces it experiences. Time is divided into time steps in the order of femtoseconds. At each time step, the forces that each atom in the system is experiencing is determined by using a force field.

Determining the forces allows for the position and the velocity of each atom to thus be calculated.

In this study, a molecular dynamics simulation was performed only on the lowest energy structure of the system.

Chapter 3

RESULTS

In this chapter, resulting structures of the computational methods explained in the prior sections will be presented. Specifically, results from adding vacancies to a cubic $2 \times 2 \times 2$ system, shown in Fig. 3.1 will first be shown. Then, structures that were obtained from adding vacancies to the expanded $4 \times 4 \times 4$ cubic system will be discussed. Additional results from running a molecular dynamics simulation will be shown. Lastly, results for these generated $Sr_2CaSbO_{5.5}$ will be compared to that of the $Sr_2CaSbO_{5.5}$ and $Sr_3SbO_{5.5}$ structures from a neutron pair distribution as well as to generated $Sr_3SbO_{5.5}$. These results have also been published in other places [11].

3.1 Generated M=Ca results

After Glazer distortions were implemented to create 23 unique structures, it was found that the lowest energy structure had (+,-,-) in the x, y, and z direction respectively.

The lattice size of the structure, which can be seen in Fig.3.2 was 8.57 in all directions and the angles were 89.9° 88.2° 90.1° in the x, y and z directions.

Out of the 24 oxygen atoms in the 2x2x2 Sr_2CaSbO_6, one was removed to produce an $Sr_2CaSbO_{5.75}$, resulting in 72 possibilities. The 72 resulting structures resulted from the 24 oxygen atoms that were removed and the three different movements of nearby oxygen atom to shield the positive charge of the central ion. Of the 72 structures, the lowest energy configuration, shown in Fig. 3.3 had a lattice size of 8.47 Å, 8.55 Å, 8.47 Å. Fig. 3.3 shows that a trigonal bipyramid forms around the Sb ions (in brown) that have the vacancy, and a square pyramid forms on the Ca ion (in blue) that has the vacancy. The remainder of the ions continued to have a octahedral arrangement after optimization using VASP. Sr-O bonds are hidden to make it easier to see the polyhedral arrangements.

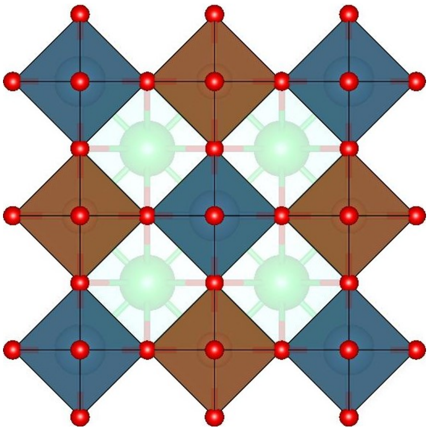


FIGURE 3.1: $Sr_2CaSbO_{5.5}$ system before any octahedral tilting, causing it to have a cubic structure. The polyhedra in brown surround the Sb ions and the polyhedra in blue surround the Ca ions. A sites are occupied by Sr ions shown in green.

After rotation and addition of a second vacancy to Fig. 3.3, the lowest energy configuration was determined from a large number of generated structures. It was found that the lowest energy structure for this system came the lowest energy structure without

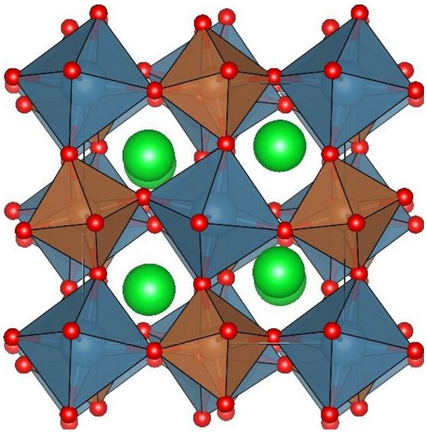


FIGURE 3.2: Lowest energy structure $Sr_2CaSbO_{5.5}$ system from the optimization of Glazer tilt possibilities. The structure has a +,-,- Glazer label

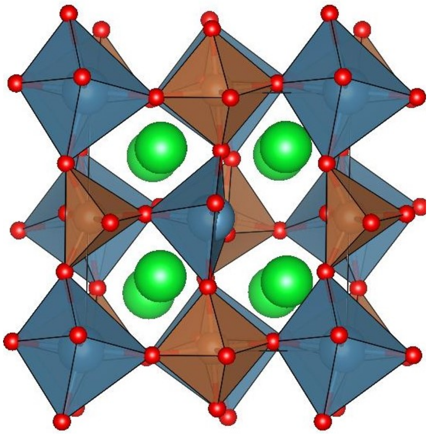
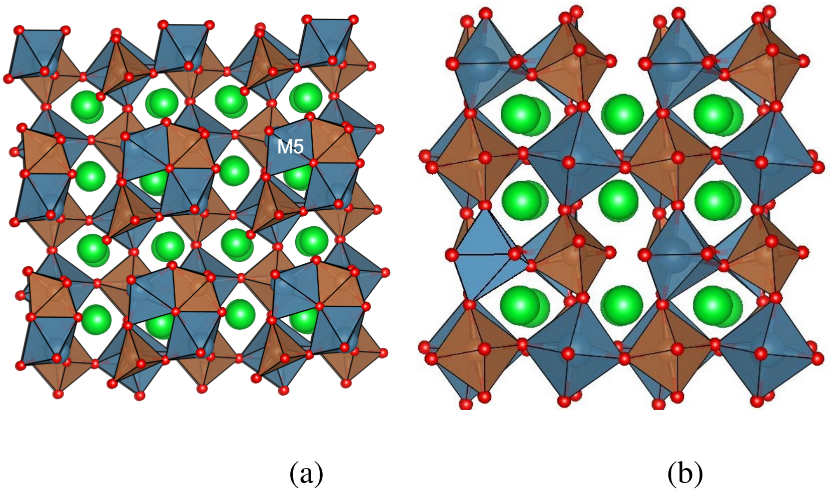


FIGURE 3.3: . Lowest energy structure of 72 possibilities from the removal of one oxygen from the movement of nearby oxygen atoms to shield positive charge of the central atoms to form $Sr_2CaSbO_{5.75}$

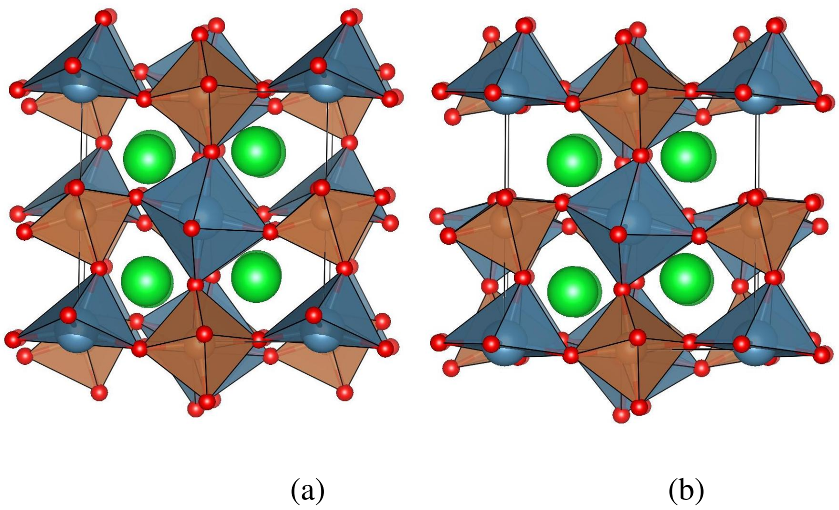
rotation of the polyhedra. Even though a number of configurations were generated after the removal of an additional oxygen, not all of the configurations were distinct. Many structures would have energies that were within 0.2 eV away from each other but would have polyhedral arrangements that would be very similar to others of similar energies. In this way, many structures would not be distinct. However, if two structures were more than 0.2 eV away even though they had a similar polyhedral arrangement, they would be classified as distinct. In addition, to the lowest energy structure shown in Fig. 3.4 (a), the second lowest distinct structure is shown in Fig. 3.4 (b).



(a) (b)

FIGURE 3.4: The low energy structures for $Sr_2CaSbO_{5.5}$ structures shown in (a) and (b), where (a) is the lowest energy structure and (b) is a second distinct structure.

The lowest energy structure exhibited edge sharing, where the edge of the polyhedra surrounding the calcium ion touches the edge of the polyhedra surround the antimony ion. The polyhedral arrangements around the calcium relaxed into an arrangement that is not quite a square pyramid or trigonal bipyramid, as can be seen in Fig. 3.4 (a). On the other hand, the polyhedra arrangement around the antimony ion relaxed into forming a trigonal bipyramid. This is consistent to the lowest energy one vacancy structure in Fig. 3.3



(a) (b)

FIGURE 3.5: Additional 2 × 2 × 2 $Sr_2CaSbO_{5.5}$ structures shown in (a) and (b).

Structure	Energy (eV)	Relative Energy (eV)
Fig. 3.4(a)	$-1,922.00$	0.00
Fig. 3.4(b)	$-1,920.39$	1.60
PDF	2.04	8.34

TABLE 3.1: Absolute and relative energies for all $4 \times 4 \times 4$ expanded M=Ca structures, either generated or from PDF analysis (as indicated).

Additional structures (Fig. 3.5 for the M=Ca system, that were higher in energy compared to the minima, displayed similar geometrical characteristics. Even in slightly higher energy structures, antimony ions adjacent to site of oxygen vacancy displayed preference for trigonal bipyramidal arrangement. Likewise, calcium ions next to the oxygen-deficient site exhibited square pyramid structures.

3.2 Comparison to similar structures

Comparing results from the computationally generated $Sr_2CaSbO_{5.5}$ to other systems could shed light on common reoccurring patterns among structural arrangements that could inform conduction pathways.

Upon completion of the same methods to the M=Sr system, the lowest energy structure for this system was determined in the same way as for the M=Ca system. The polyhedral arrangement for the lowest energy structure for the $Sr_2CaSbO_{5.5}$ system parallels very nicely with that of the $Sr_3SbO_{5.5}$. The M=Sr structure shown in Fig. 3.6 also shows edge sharing between the M polyhedra and the Sb polyhedra.

Like the polyhedra surrounding the Ca ion at the B site, the polyhedra around the Sr ion at B sites also took the form of an arrangement that is somewhere between a trigonal bipyramid and a square pyramid. Both of these lowest energy structures also had trigonal bipyramids around the Sb ion.

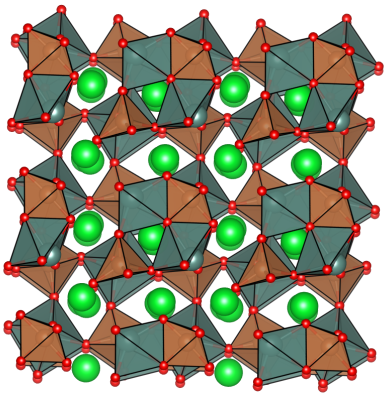
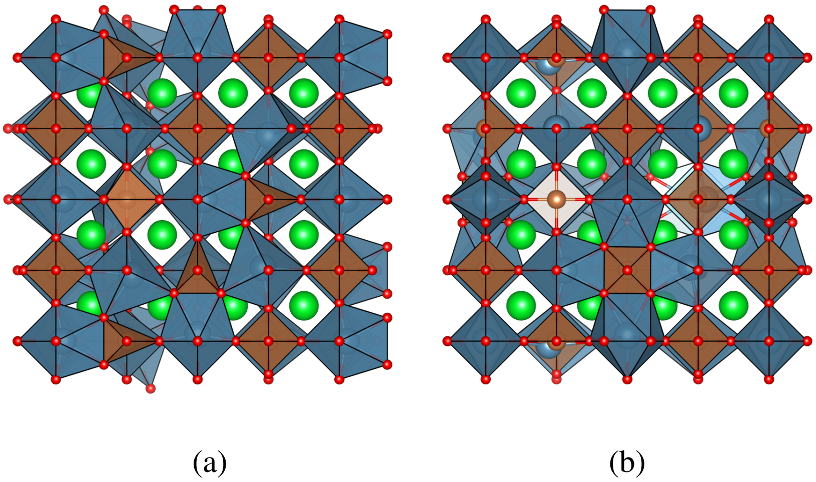


FIGURE 3.6: The lowest energy structure for $Sr_3SbO_{5.5}$ system is shown, where blue polyhedra represent M polyhedra, in this case M=Sr, and green spheres are the A atoms (A=Sr)



(a) (b)

FIGURE 3.7: Two of the PDF generated structures[1] for the $Sr_2CaSbO_{5.5}$ system before any optimizations.

However, in order to draw more direct comparisons about types of polyhedra between systems, a category system was created and implemented. Polyhedra were classified according the atom it surrounds, the coordination number of the central ion, and the the number of edges the polyhedra have. For instance, a designation of Sb(5,9) would mean that the Sb ion has a coordination number of 5 and the polyhedra has 9 edges, which fits a trigonal bipyramid shape. However, sometimes, the number of edges will

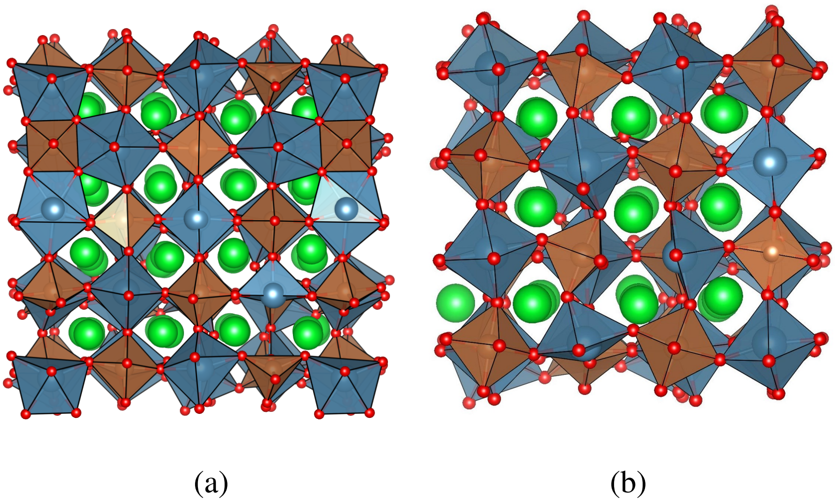


FIGURE 3.8: Two of the PDF generated structures for the $Sr_2CaSbO_{5.5}$ system after optimization.

3.7 3.8

Structure	Energy (eV) M=Ca	Lattice (Å) M=Ca	Energy (eV) M=Sr	Lattice (Å) M=Sr
Experimental[1, 12]	-	8.20-8.23	-	8.31-8.32
Generated	0.00	8.41	0.09	8.59
PDF	2.04	8.34	0.00	8.47

TABLE 3.2: Relative energies and average lattice sizes of the lowest energy generated structures optimized in this study and the lowest energy structure optimized starting from the sample neutron pair distribution function (PDF) analysis structures for M=Ca and Sr, respectively. The experimental lattice size is also shown for comparison.

always be the same given a coordination number. For example, a tetrahedron has a coordination number of four, which will always a produce a shape that has 6 straight edges. Because of this, there are categories that have only one number in the name. In generated structures, there were some polyhedra that could not be distinguished between a distorted square pyramid and a trigonal bipyramid like, as indicated in white, in Fig. 3.4. Because they are not able to be differentiated, they were both designated with an M5 label.

Based on this, it was found that 50 percent of the M polyehdra in the computationally generated M=Ca structure were M(6,12), while the other 50 percent were M5

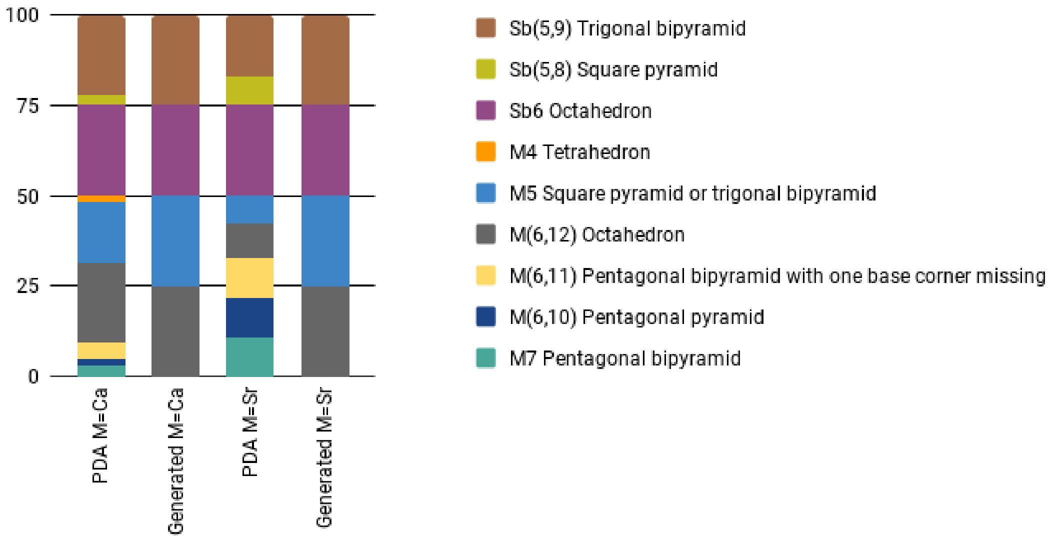


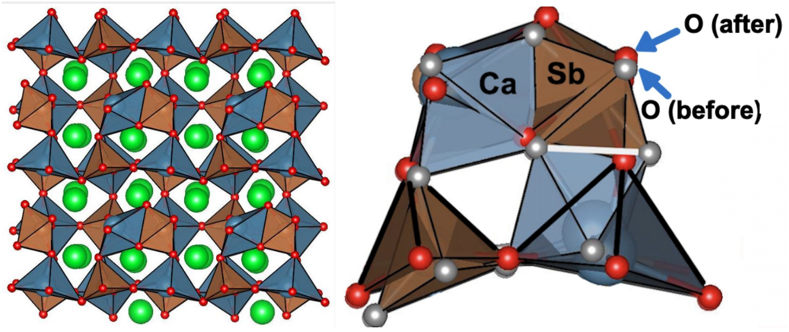
FIGURE 3.9: Percentages of the polyhedra around the M and B sites are shown for all systems. Structures with opt in their name were experimentally fitted structures after optimization. Experimental fit rot structures are structures that were originally rotated 45 degrees around the SbO_5 polyehdra. Edge sharing in the small systems effectively allowed for the formation of pentagonal pyramid-like structures made from trigonal bipyramids around the Sb sharing edges with the square pyramids around the M atoms.

(either square pyramid or trigonal bipyramid). Half the Sb ions were Sb6 and the other half were Sb(5,9).

The lowest energy minimum found from neutron pair distribution analysis for the $Sr_2CaSbO_{5.5}$ system displayed some similarity, where majority of the polyehdra around the M atom were either M(6,12) or M5. Also, the majority of the polyehdra around the Sb atom were either Sb6 or Sb(5,9). However, in this structure there also showed Sb(5,8), M4, M(6,11), M(6,10) and M7 polyhedra but in relatively small amounts.

The polyhedral distribution of the computationally generated M=Sr structure, identically resembled that of M=Ca. M centered polyhedra were evenly distributed between M(6,12) and M5, while Sb centered polyehdra took on Sb6 or Sb(5,9) configurations. Polyhedral distribution across systems are described in Fig. 3.9

A simple comparison of energies revealed that the computationally generated M=Ca minimum was 2.04 eV lower than the minima from the paired neutron distribution analysis. Direct energetic comparisons to the M=Sr system cannot be made because the entire system is different. Table 3.2 shows relative energies and lattice sizes across the systems.



(a) (b)

FIGURE 3.10: Post-MD configuration at 1200 K are shown for M=Ca system. (a) shows the full end result of the simulation and (b) shows a zoomed in version that highlights areas of significant changes. The polyhedra that significantly changed are highlighted in black. The shared edge that disappeared during the simulation is shown white.

3.3 Testing for oxygen vacancy conduction

The generated $Sr_2CaSbO_{5.5}$ minima were run using molecular dynamics with velocity rescaling to sample configurations with temperature set to 1200K. The resulting structure of the molecular dynamics simulation had a rotation of an Ca5 polyhedra, causing some loss of edge sharing to occur as can be seen in Fig. 3.10. A similar trend where edge sharing lost due to a clockwise rotation of polyhedra is visible in the M=Sr case, indicating that such rotations may actually be critical for conduction to occur.

Chapter 4

DISCUSSION

In this study, it was found that oxygen deficient systems mainly took on either trigonal bipyramid, square pyramid, or pentagonal pyramid geometry. Edge sharing was also present in our minima. In this chapter, arguments that explain the geometric arrangements seen in our results will be presented. Firstly, this chapter will go into how the size of the central cation can impose physical limitations, driven by Coulombic forces, on the shape of the polyhedra. Then, lattice arguments for observed structures will be discussed. Lastly, oxygen vacancy results will be analyzed.

4.1 Central ion size impact on polyhedra shape

The results of this study show that deficient polyehdra took on a range of different types of polyhedra. Specifically, either trigonal bipyramids, square pyramids, or pentagonal pyramids surrounded the central cation when a vacancy of present. The type of geometry around the ions is influenced by the the size of the central ion. Pauling suggests that there are several parameters that go in to helping us predict a crystal structure. [13] These principles could be applied to our oxygen vacant systems in order to understand properly why the crystal structures shown in this paper look the way they do.

The cation (the central ion) has a natural attraction to the anions (oxygen ions) that surround it. The cation approaches a distance away from the anions such that there is a balance in attractive and repulsive Coulombic forces. This balance could be determined by calculating the radius ratio, which is the ratio of the cation radius to the anion radius. If the distance between the anion and the cation is increased just even slightly higher than the radius sum, the structure would be high in energy and thus, quite unstable. Pauling had calculated the minimum radius ratio for different types of polyhedra and used the cation radius and its potential coordination number to predict exactly how the crystal structure would look. In the crystallographic sense, coordination number (CN) refers to the atoms that are the nearest neighbors of the species of interest, which in this case would be the central cation [14]. In this study, we can do the same and see if these geometrical predictions hold true to our resulting structures.

In order to effectively predict the polyhedral shape, we must first calculate cation to anion radius ratio. To do this, we split the polyhedron into isosceles triangles at the vertices as shown in Fig.4.1. We can further split these triangles to form right triangles as seen by the dotted blue lines drawn in Fig.4.1. Splitting the triangles this way allows us to use the Pythagorean theorem to solve for the hypotenuse, which is $r_{cation} + r_{anion}$. The central angle of θ can be rewritten as $\frac{2\pi}{2n}$, which can be further simplified to just

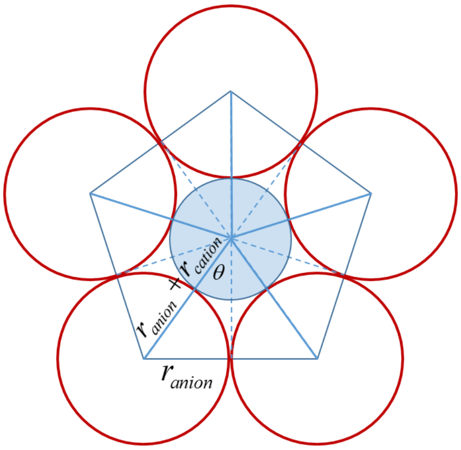


FIGURE 4.1: A schematic of how the cation to anion radius ratios can be calculated. The red circles are the anions at the corners of the polyhedra and the blue circle represents the central cation.

Geometry	CN	Min. Ratio
Tetrahedral	CN=4	0.225
Trigonal bipyramid	CN=5	0.155
Octahedra	CN=6	0.414
Cubes	CN=8	0.701

TABLE 4.1: Calculated minimum ratios for common polyhedral geometry

$\frac{\pi}{n}$. Further mathematical manipulation yields r_{anion}=(r_{cation} + r_{anion})$sin(\frac{\pi}{n})$. Solving for the ratio, $\frac{r_{cation}}{r_{anion}} = csc\left(\frac{\pi}{n}\right) - 1$.

With this equation, the minimum ratios for tetrahedra (CN=4), trigonal bipyramids (CN=5), octahedra (CN=6), pentagonal bipyramids (CN=7), and cubes (CN=8) were determined (Table 4.1).

Ion	**Ionic Radius(Å)[2]**	$r_{ion}/r_{O^{2-}}$	Pauling CN
Sb^{5+}	0.60	0.43	6
Ca^{2+}	1.00	0.71	7
Sr^{2+}	1.18	0.84	7

TABLE 4.2: The Shannon ionic radii [2] for ions in the centers of octahedra are shown above in the second column. The likely coordination number of the ions were predicted based on the minimum cation to anion ratio.

Central cations that exist in the holes of octahedra have a certain ionic radii that can influence the shape of the polyehdra surrounding it. The Shannon ionic radius[2] for these cations are shown in Table 4.2. The largest coordination number for the minimum cation to anion (oxygen) radius was determined to be 6, 7, and 7 for Sb^{5+}, Ca^{2+}, and Sr^{2+}, respectively according to Pauling [13].

These calculations support the results shown in Fig. 3.9. Specifically, the observation of pentagonal motifs existing at only M sites is confirmed by the fact that the cation to anion ratio (Sr^{2+} radius O^{2-} radius) for M=Sr is the highest of the two. Conversely, a smaller cation to anion ratio implies that it is less likely for pentagons to surround a Ca^{2+} ion. This could provide part of a potential explanation as to why pentagons around Ca ions did not form in our generated structures, even after forced rotations of adjacent octahedra.

Additionally, in our generated structures it was found that smaller ions (Sb^{5+} and Ca^{2+}) adjacent to the oxygen vacancy encompassed trigonal bipyramids and distorted square pyramids. However, these smaller polyhedra shared their edges such that the shared complexes were variations of pentagonal bipyramids as shown in Fig. 3.4 and Fig. 3.6. Since these shared complexes were energy minima, it could be that having some variation of a pentagonal bipyramid can stabilize the energy of the crystal structure. This could explain the fact that pentagonal bipyramids, pentagonal pyramids, and/or pentagonal bipyramids with one base corner missing were observed in the PDF analysis structures, both before (Fig. 3.7) and after optimization (Fig. 3.8). Because Ca is a smaller ion than Sr, this same argument explains why the PDF analysis structures had smaller percentages of pentagonal motifs compared to that of the M=Sr system.

4.2 Charge exposure and lattice arguments

The radius of all the M cations was larger than that of the Sb^{5+} making the MO_6 octahedra larger than the SbO_6 octahedra. The larger the M cation's radius, the larger the disparity between the two octahedral sizes in each double perovskite and potentially greater lattice changes when oxygen vacancies are introduced.

Though generated $2 \times 2 \times 2$ structures has overall higher lattices compared to the PDF analysis structures across all structures (Table. 3.2), the same general trend in previous literature [1][12] was observed. M=Ca system had a smaller lattice than the M=Sr system.

The charge exposure created by a vacancy at a corner site of the octahedron can be compensated by movement of the remaining oxygen atoms to form a trigonal bipyramidal structure, causing a delocalization of the vacancy as shown in Fig. 1.3. Both the results adapted from the experimental structures and the computationally generated structures demonstrated a large portion of trigonal bipyramidal structure as shown in Fig. 3.9. A larger sized positive central ion can be better surrounded by negative charge when a 45 degree rotation of an adjacent octahedron in either x, y, or z direction allows an increased coordination forming pentagonal bipyramids, as shown in Fig. 1.4. By this argument, the appearance of polyehdra with greater coordination numbers may serve to lower down the lattice size of our generated structures to approach experimental values.

Thus, exploring more structures starting from a $4 \times 4 \times 4$ may be crucial in finding configurations with smaller lattice sizes that better match experimental values.

4.3 Testing for oxygen vacancy motion

Molecular dynamics run via *ab initio* showed that the most probably form of oxygen vacancy conduction is related to polyhedral rotation such that the oxygen deficient site “moves” (Fig. 3.10). Previous research found that rotation of the polyehdra in such manner may allow proton conduction to occur if a proton is introduced into the system [15]. However, the structures considered in this paper did not possess significant local distortions and complex edge sharing.

Chapter 5

CONCLUSION

This study showed that there is a preference for certain local geometry over other depending on the central ion for double perovskite systems $Sr_2CaSbO_{5.5}$ and $Sr_3SbO_{5.5}$. Specifically, there was a large preference for trigonal bipyramid and square pyramid geometry around smaller ions. Around larger ions, pentagonal bipyramids or other variations of pentagons dominated. It was also found that rotation of crystal structures created room for edge sharing that was found to be important for lowering the energy for the overall system. Molecular dynamics simulation revealed that rotations may be critical to oxygen vacancy motion. Further kinds of simulations that characterizes oxygen movement would need to be performed to see if similar results occur. Additionally, a greater survey of distinct $4 \times 4 \times 4$ rotated structures would need to be performed to see if there are other variations that can not only lower the energy of the system, but also to see if other types of oxygen vacancy movement can be observed in these sample double perovskite systems.

Appendix A

45 Degree Rotation on 4x4x4 System

This script removes sixteen oxygen atoms from a 4 × 4 × 4 A_2MBO_6 system as mentioned in Chapter 2.3.4. The script reads in the coordinates of the starting structure and then randomly chooses 16 oxygen atoms to remove, such that there is only one vacancy per polyhedra at maximum. This script then randomly chooses an Sb polyhedra and an M polyhedra that does not have a vacancy adjacent. The coordinates of the oxygen ions around the polyhedra are altered such that a 45 degree rotation takes place about the central ion. This script then prints the resulting coordinate files. The script was written by me using Python with assistance from my project advisor. A large portion of the script was created by collaborative effort with the help of Professor Gomez, Jiayun Zhong, and Konrad Gomez-Haibach.

```
  #!/usr/bin/env python
"""
```

```
"""

__copyright__ = "GPL"
__license__ = "Python"

import os
import time
import sys
import re
import numpy
import math
import operator
import random

def periodic(arg1,arg2):
        #returns periodic difference
      dx=arg2-arg1
      while (dx<-0.5):
                 dx=dx+1.0
      while (dx>0.5):
                 dx=dx-1.0
      return dx;

def zeroCheck(difference):
    #returns 0 if the difference in arg1 is close to 0 or 1 and 1 if it's
    #different from both

    #also changes difference so its between -0.5 and 0.5

    while (difference<-0.5):

                 difference=difference+1.0

    while (difference>0.5):

                 difference=difference-1.0

    if (abs(difference)<0.1):

     check=0

    else:

     check=1

    return check;


def periodicDiff(difference):

    #returns periodic difference from difference

    while (difference<-0.5):

                 difference=difference+1.0

    while (difference>0.5):

                 difference=difference-1.0

    return difference;


def axisConnectingSbVoM(posClosestSb,posClosestM):

        #delta is the difference vector

delta=numpy.zeros(3)

max=0.0

for i in range(0,3):
```

```
delta[i]=periodic(posClosestSb[i],posClosestM[i])

if (abs(delta[i])>max):

max=abs(delta[i])

axis=i

print "Difference Vector is",delta

print "axes connectingSbVoM ",axis

return axis


def RotationMatrix(axisOfRotation,alpha):

#initially rotation matrix as identity matrix

rotation=[[1,0,0],[0,1,0],[0,0,1]]

if axisOfRotation==0:

   rotation[1][1]=math.cos(alpha)

   rotation[1][2]=math.sin(alpha)

   rotation[2][1]=-math.sin(alpha)

   rotation[2][2]=math.cos(alpha)

if axisOfRotation==1:

   rotation[0][0]=numpy.cos(alpha)

   rotation[0][2]=-numpy.sin(alpha)

   rotation[2][0]=numpy.sin(alpha)

   rotation[2][2]=numpy.cos(alpha)

        if axisOfRotation==2:

   rotation[0][0]=numpy.cos(alpha)

   rotation[0][1]=numpy.sin(alpha)

   rotation[1][0]=-numpy.sin(alpha)
   rotation[1][1]=numpy.cos(alpha)
return rotation


def MatrixTimesVector(matrix,vector):
   finalVector=numpy.zeros(3)
   for i in range(0,3):
for j in range(0,3):
finalVector[i]=finalVector[i]+matrix[i][j]*vector[j]
   return finalVector;


lattice=numpy.zeros((3,3))
initialBackbonePositions=numpy.zeros((324,3))
MatomDifference = numpy.zeros((7,3))
MatomDifferenceTransform = numpy.zeros((7,3))
SbDifference = numpy.zeros((7,3))
SbDifferenceTransform = numpy.zeros((7,3))
oDistToMatom = numpy.zeros((196))
oNumToMatom = numpy.zeros((196))
oDistToSb = numpy.zeros((196))
oNumToSb = numpy.zeros((196))
axesOfRotation = numpy.zeros((3),dtype=numpy.int)
axisLabel=['x','y','z']
counter=0
counterBackbone=0
theta=0.785 # 45*3.14/180.
```

```
for line in open("POSCAR"):

        if counter>1 and counter <5:

                dummy=line.split()

                lattice[counter-2][0]=eval(dummy[0])

                lattice[counter-2][1]=eval(dummy[1])

                lattice[counter-2][2]=eval(dummy[2])

        if counter>7:

                dummy=line.split()

                initialBackbonePositions[counterBackbone][0]=eval(dummy[0])

                initialBackbonePositions[counterBackbone][1]=eval(dummy[1])

                initialBackbonePositions[counterBackbone][2]=eval(dummy[2])

                counterBackbone=counterBackbone+1

        counter=counter+1


print "counter is "+str(counter)

print "counterBackbone is "+str(counterBackbone)


#Checking that we can print out what we read in

#print "Double Perovskite"

#print "8.2200000000000"

#for k in range(0,3):

#  print lattice[k][0],lattice[k][1],lattice[k][2]

#print " Sr Sb   O"

#print "  12 4 24"
```

```
#print "Direct"

#for k in range(0,counterBackbone):



#calculate atom-atom distances

print "atom-atom distances"

print "n | m | distances[n][m]"

distances=numpy.zeros((counterBackbone,counterBackbone))

for n in range(0,counterBackbone):

for m in range(0,counterBackbone):

dx=periodic(initialBackbonePositions[n][0],initialBackbonePositions[m][0])

dy=periodic(initialBackbonePositions[n][1],initialBackbonePositions[m][1])

dz=periodic(initialBackbonePositions[n][2],initialBackbonePositions[m][2])

dxC=lattice[0][0] * dx + lattice[1][0] * dy + lattice[2][0] * dz

dyC=lattice[0][1] * dx + lattice[1][1] * dy + lattice[2][1] * dz

dzC=lattice[0][2] * dz + lattice[1][2] * dy + lattice[2][2] * dz

distances[n][m]=8.22*((dxC ** (2)) + (dyC ** (2)) + (dzC ** (2))) ** (0.5)

print n+1,m+1,distances[n][m]




#  positions 0 to 7 are Sr

#  positions 8 to 11 are Matom

#  positions 12 to 15 are Sb

#  positions 16 to 38 are O


#goal is to rotate octahedron of one M and one Sb three different ways and for each
```

```
#print out POSCAR files removing each possible oxygen


NO=190

O_offset=128



#Need to be sure that Original.vasp does not have oxygen vacancies round the

#chosen atoms

#If it does, choose others

chosenMatom=64

chosenSb=96

SbsConnected =[]

MsConnected=[]

ORemoved=[]

flagPrintMFile=True

flagPrintSbFile=True


for z in range(0,16):

OtoRemove=random.randint(129,304)

closestMatom=64

#need to find closest oxygens to the chosen atoms



#get list of distances of oxygens from chosenMatom

for j in range(O_offset+1, O_offset + 1 + NO):
```

```
oDistToMatom[j - O_offset] = distances[chosenMatom][j]

oNumToMatom[j - O_offset] = j - O_offset


print "\nUnsorted chosen Matom to O array"

print "O index O | distance from chosen Matom"

for l in range(1, NO+1):

print str(oNumToMatom[l]) + " " + str(oDistToMatom[l])

print


# bubble sort from book "Discrete Mathematics"

#end of list is NO and the range function goes one less than the end of the list

for j in range(1, NO):

#end of list if NO

# one j less than the end of list is NO-j but range goes one less so end of

#range had to be NO-j+1

for m in range(1, NO - j+1):

if oDistToMatom[NO + 1 - m] < oDistToMatom[NO - m]:

#putting in earlier larger distance into a temporary variable and

# switching order

Temp = oDistToMatom[NO - m]

oDistToMatom[NO - m] = oDistToMatom[NO + 1 - m]

oDistToMatom[NO + 1 - m] = Temp

# same switching for label

iTemp = oNumToMatom[NO - m]

oNumToMatom[NO - m] = oNumToMatom[NO + 1 - m]
```

```
oNumToMatom[NO + 1 - m] = iTemp


print "Sorted O vacancy to O array Matom"

print "O index O | distance from chosenMatom"

lndx=1

for l in range(1, NO+1):

print str(l)+" "+str(oNumToMatom[l]) + " " + str(oDistToMatom[l])

if (l<6):

  if (oDistToMatom[l]>3.0):

flagPrintMFile=False

print "There is a vacancy around the chosen M atom.  Change your choice.\n"

exit()

#get list of distances of oxygens from chosenSb


for j in range(O_offset+1, O_offset + 1 + NO):

oDistToSb[j - O_offset] = distances[chosenSb][j]

oNumToSb[j - O_offset] = j - O_offset


print "\nUnsorted chosen Sb to O array"

print "O index O | distance from chosen Sb"

for l in range(1, NO+1):

print str(oNumToSb[l]) + " " + str(oDistToSb[l])

print


# bubble sort from book "Discrete Mathematics"
```

```
#end of list is NO and the range function goes one less than the end of the list

for j in range(1, NO):

#end of list if NO

# one j less than the end of list is NO-j but range goes one less so end of

#range had to be NO-j+1

for m in range(1, NO - j+1):

if oDistToSb[NO + 1 - m] < oDistToSb[NO - m]:

#putting in earlier larger distance into a temporary variable and

# switching order

Temp = oDistToSb[NO - m]

oDistToSb[NO - m] = oDistToSb[NO + 1 - m]

oDistToSb[NO + 1 - m] = Temp

# same switching for label

iTemp = oNumToSb[NO - m]

oNumToSb[NO - m] = oNumToSb[NO + 1 - m]

oNumToSb[NO + 1 - m] = iTemp


print "Sorted O vacancy to O array"

print "O index O | distance from chosen Sb"

for l in range(1, NO+1):

print str(l)+" "+str(oNumToSb[l]) + " " + str(oDistToSb[l])

                if (l<6):

                  if (oDistToSb[l]>2.5):

                        flagPrintSbFile=False

print "There is a vacancy by your chose Sb.  Change your choise\n"
```

```
exit()

maxOforM=7
maxOforSb=7

#subtracting M center from Os around it
print "O differences from Matom center"
for a in range(1, maxOforM):
for j in range(0, 3):
MatomDifference[a][j] = initialBackbonePositions[O_offset+oNumToMatom[a]][j] -
initialBackbonePositions[chosenMatom][j]
MatomDifference[a][j]=periodicDiff(MatomDifference[a][j])
print a, j, MatomDifference[a][j],oNumToMatom[a]+O_offset,oNumToMatom[a]

print "O differences from Sb center"
for a in range(1, maxOforSb):
for j in range(0, 3):
SbDifference[a][j] = initialBackbonePositions[O_offset+oNumToSb[a]][j] -
initialBackbonePositions[chosenSb][j]
SbDifference[a][j]=periodicDiff(SbDifference[a][j])
print a, j, SbDifference[a][j],oNumToSb[a]+O_offset,oNumToSb[a]

#Rotations about Matom and Sb
for irot in range(0,3):
axesOfRotation[irot]=irot
```

```
#rotations about Matom octahedron

for a in range(1,maxOforM):

print RotationMatrix(axesOfRotation[irot],theta)

print MatomDifference[a]

MatomDifferenceTransform[a]=MatrixTimesVector(RotationMatrix(axesOfRotation[irot],

theta),MatomDifference[a])

print MatomDifferenceTransform[a]

#Rotations about Sb

for a in range(1,maxOforSb):

print RotationMatrix(axesOfRotation[irot],theta)

print SbDifference[a]

SbDifferenceTransform[a]=MatrixTimesVector(RotationMatrix(axesOfRotation[irot],theta),

SbDifference[a])

print SbDifferenceTransform[a]




# i loops over oxygens to be removed or vacancies

 #zero index in array! first O is at 16

if (oNumToMatom[1]!=

OtoRemove and oNumToMatom[2]!=OtoRemove and oNumToMatom[3]!=OtoRemove and oNumToMatom[4

  #writing the POSCAR file

  nameoffile = "POSCAR_RotM"+str(irot)+"_"+str(OtoRemove-128)+".vasp"

  poscar = open(nameoffile,’w’)

  poscar.write("Double Perovskite\n")

  poscar.write("8.2200000000000\n")
  for k in range(0,3):
poscar.write(str(lattice[k][0])+" "+str(lattice[k][1])+" "+str(lattice[k][2])+"\n")
  poscar.write (" Sr Y  Sb  O"+"\n")
  poscar.write (" 64 32 32 160"+"\n")
  poscar.write("Direct"+"\n")

  for k in range(0,counterBackbone):
flagPrint=True
if k==OtoRemove:
flagPrint=False #don't print the vacancy
for a in range(1,maxOforM):
  if (k==oNumToMatom[a]+O_offset):
flagPrint=False  #don't print the other oxygens in the octahedron
if flagPrint==True:
  poscar.write(str(initialBackbonePositions[k][0])+" "+
  str(initialBackbonePositions[k][1])+" "+str(initialBackbonePositions[k][2])+"\n")

else:
  for a in range(1,maxOforM):#printing rotated other oxygens in octahedron
if k==oNumToMatom[a]+O_offset:
           poscar.write(str(MatomDifferenceTransform[a][0]
           +initialBackbonePositions[chosenMatom][0])+" "+
            str(MatomDifferenceTransform[a][1]+
            initialBackbonePositions[chosenMatom][1])+" "+str(MatomDifferenceTransform[
           +initialBackbonePositions[chosenMatom][2])+"\n")
  poscar.close()


  #writing the script files

  nameofscript="scriptM"+str(irot)+"_"+str(OtoRemove-128)

  script = open(nameofscript, 'w')

  script.write("#PBS -N


  M_Rot"+str(irot)+"_"+str(OtoRemove-128)+"\n")

  script.write("#PBS -l nodes=2:ppn=16:E5-2680\n")

  script.write("#PBS -M patel25m@mtholyoke.edu\n")

  script.write("#PBS -m bea\n")

  script.write("#PBS -j oe\n")

  script.write("cat $PBS_NODEFILE\n")

  script.write("cd $PBS_O_WORKDIR\n")

  script.write("module unload compiler64/pgi12.8 openmpi/openmpi-1.4.5.pgi12.8-64\n")

  script.write("module load intel/12.1.6 mpi/openmpi-1.4.5_intel-12.1.6\n")

  if OtoRemove-128 <10 :

nameDir="SrRot"+str(irot)+"V0"+str(OtoRemove-128)+"VOonNoRotOct"

  else:

nameDir="SrRot"+str(irot)+"V"+str(OtoRemove-128)+"VOonNoRotOct"

  script.write("mkdir "+nameDir+"\n")

  script.write("cp INCAR "+" "+nameDir+"\n")

  script.write("cp KPOINTS"+" "+nameDir+"\n")

  script.write("cp POTCAR"+" "+nameDir+"\n")

  script.write("mv "+nameoffile+" "+nameDir+"/POSCAR\n")
  script.write("cd "+nameDir+"\n")
  script.write("mpiexec /home/chem_programs/bin/vasp5.3gamma>vasp.out\n")
  script.write("rm CHG CHGCAR EIGENVAL
  vasprun.xml IBZKPT OSZICAR WAVECAR XDATCAR DOSCAR PCDAT\n")
  #commands to be issued on medusa
  print "qsub "+nameofscript

if (oNumToSb[1]!=OtoRemove and
oNumToSb[2]!=OtoRemove and oNumToSb[3]!=OtoRemove and
oNumToSb[4]!=OtoRemove and oNumToSb[5]!=OtoRemove and oNumToSb[6]!=OtoRemove):
  #writing the POSCAR file
  nameoffile = "POSCAR_RotSb"+str(irot)+"_"+str(OtoRemove-128)+".vasp"
  poscar = open(nameoffile,'w')
  poscar.write("Double Perovskite\n")
  poscar.write("8.2200000000000\n")
  for k in range(0,3):
poscar.write(str(lattice[k][0])+" "+str(lattice[k][1])+" "+str(lattice[k][2])+"\n")
  poscar.write (" Sr  Y  Sb  O"+"\n")
  poscar.write (" 64 32  32  160"+"\n")
  poscar.write("Direct"+"\n")

  for k in range(0,counterBackbone):
flagPrint=True
if k==OtoRemove:
flagPrint=False #don't print the vacancy
```

```
for a in range(1,maxOforSb):

if (k==oNumToSb[a]+O_offset):

flagPrint=False  #don't print the other oxygens in the octahedron

if flagPrint==True:

poscar.write(str(initialBackbonePositions[k][0])+" "+

str(initialBackbonePositions[k][1])+"

"+str(initialBackbonePositions[k][2])+"\n")

else:

for a in range(1, maxOforSb):

if k==oNumToSb[a]+O_offset:

poscar.write(str(SbDifferenceTransform[a][0]+

initialBackbonePositions[chosenSb][0])+" "+ str(SbDifferenceTransform[a][1]+

initialBackbonePositions[chosenSb][1])+" "+str(SbDifferenceTransform[a][2]

+initialBackbonePositions[chosenSb][2])+"\n")

  poscar.close()


  #writing the script files

  nameofscript="scriptSb"+str(irot)+"_"+str(OtoRemove-128)

  script = open(nameofscript, 'w')

  script.write("#PBS -N Sb_Rot"+str(irot)+"_"+str(OtoRemove-128)+"\n")

  script.write("#PBS -l nodes=2:ppn=16:E5-2680v4\n")

  script.write("#PBS -M patel25m@mtholyoke.edu\n")

  script.write("#PBS -m bea\n")

  script.write("#PBS -j oe\n")

  script.write("cat $PBS_NODEFILE\n")
  script.write("cd $PBS_O_WORKDIR\n")

  script.write("module unload compiler64/pgi12.8 openmpi/openmpi-1.4.5.pgi12.8-64\n")

  script.write("module load intel/12.1.6 mpi/openmpi-1.4.5_intel-12.1.6\n")

  if OtoRemove-128 <10 :

nameDir="SbRot"+str(irot)+"V0"+str(OtoRemove-128)+"VOonNoRotOct"

  else:

nameDir="SbRot"+str(irot)+"V"+str(OtoRemove-128)+"VOonNoRotOct"

  script.write("mkdir "+nameDir+"\n")

  script.write("cp INCAR "+" "+nameDir+"\n")

  script.write("cp KPOINTS"+" "+nameDir+"\n")

  script.write("cp POTCAR"+" "+nameDir+"\n")

  script.write("mv "+nameoffile+" "+nameDir+"/POSCAR\n")

  script.write("cd "+nameDir+"\n")

  script.write("mpiexec /home/chem_programs/bin/vasp5.3gamma>vasp.out\n")

  script.write("rm CHG CHGCAR EIGENVAL

  vasprun.xml IBZKPT OSZICAR WAVECAR XDATCAR DOSCAR PCDAT\n")

  #commands to be issued on medusa

  print "qsub "+nameofscript
```

APPENDIX

Bibliography

[1] G King, K J Thomas, and A Llobet. Drastic differences between the local and the average structures of Sr2MSbO$_{5.5}$ (M = Ca, Sr, Ba) oxygen-deficient double perovskites. *Inorganic Chemistry*, 51:13060–13068, 2012.

[2] R. D. Shannon. Revised effective ionic radii and systematic studies of interatomic distances in halides and chalcogenides. *Acta Crystallographica Section A*, 32: 757–761, 1976.

[3] S C Singhal and K Kendall. *High temperature solid oxide fuel cells: Fundamentals, design, and applications*. Elsevier, Oxford, UK, 2003.

[4] S C Singhal. Advances in solid oxide fuel cell technology. *Solid State Ionics*, 135: 305–315, 2000.

[5] D McQuarrie and J Simon. *Physical Chemistry: A molecular approach*. University Science Books, California, US, 2008.

[6] K Burke. *The ABC of DFT*. Department of Chemistry, University of California, Irvine, California, US, 2007.

[7] G. Kresse and J. Furthmüller. Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, 54:11169–11186, Oct

1996. doi: 10.1103/PhysRevB.54.11169. URL `http://link.aps.org/doi/10.1103/PhysRevB.54.11169`.

[8] G. Kresse and D. Joubert. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B*, 59:1758–1775, Jan 1999. doi: 10.1103/PhysRevB.59.1758. URL `http://link.aps.org/doi/10.1103/PhysRevB.59.1758`.

[9] W Press. *Numerical Recipes*. Cambridge University Press, Cambridge, UK, 1986.

[10] J G Henkelman and Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics*, 13:9901–9904, 2000.

[11] M Patel, J Zhong, K Gomez-Haibach, M Gomez, and G King. Lowest energy $Sr_2MSbO_{5.5}$ (M=Ca and Sr) structures show significant distortions near oxygen vacancies. *Submitted to International Journal of Quantum Chemistry*.

[12] Q Zhou, B J Kennedy, and M J Avdeev. Thermal expansion behaviour in the oxygen deficient perovskites $Sr_2BSbO_{5.5}$ (B=Ca, Sr, Ba). competing effects of water and oxygen ordering. *Solid State Ionics*, 184:2559–2565, 2011.

[13] L Pauling. The principles determining the structure of complex ionic crystals. *American Chemical Society*, 51:1010–1026, 1929.

[14] J. Toofan. A simple expression between critical radius ratio and coordination number. *Journal of Chemical Education*, 71(2):147, 1994.

[15] P. V. Ong, Y. G. Du, and P. V. Sushko. Low-dimensional oxygen vacancy ordering and diffusion in $SrCrO_3$-delta. *Journal of Physical Chemistry Letters*, 8(8):1757–1763, 2017.