Modeling of Silver-Platinum Alloy Configuration Energies

Lincoln B. Houghton
Brigham Young University - Idaho
(Dated: November 17, 2021)

I. INTRODUCTION

New metal alloys are constantly being developed and implemented in science and industry. The difficulty is in producing a useful alloy and determining its properties. Rather than manufacturing every conceivable alloy in a laboratory, each alloy's properties can be determined computationally. Complex quantum models can be used to generate atomic configuration energies and eventually detailed phase diagrams. A phase diagram is a chart used to show the distinct phases (solid, liquid, gaseous, etc.) of a material for a given temperature and pressure. These alloys can then be evaluated and, if found to be superior, fabricated for use in aircrafts, bridges, batteries, etc.

These quantum models come at a high computational cost, often making data collection a long and drawn out process. It would be useful to construct a simpler model that could be quickly trained and used to predict the atomic configuration energies of a given alloy. Such a model would help alleviate the stress on the main bottleneck to finding novelty alloys, computational power.

The purpose of this research was to investigate the credibility of this concept by attempting to produce a model of this quality. Sections II and III will cover the math and modeling concepts required to understand the actual model construction in Section IV and the analysis of its results in Section V.

II. BACKGROUND

Testing – part 2 – part 3

A. Linear Algebra

In order to understand how to build the models we want, we must first have a knowledge of linear algebra, data analysis, and signal processing. Imagine you are presented with the data points in Figure 1 as samples from a secret function. How would you recreate this secret function?

Let us first make a guess of the form of the secret function. Because it looks like a simple polynomial, let us work with a power series of the form

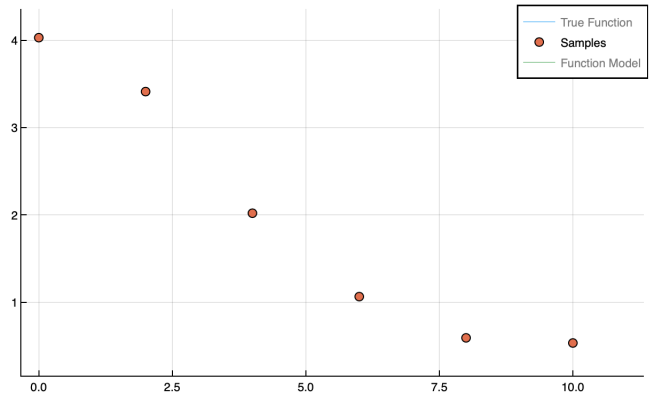


FIG. 1: An unknown function sampled by 6 data points.

$$f(x) = \sum_{n=0}^{\infty} a_n x^n \tag{1}$$

$$= a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots \ . \tag{2}$$

The coefficients, a_n, will determine the shape of the polynomial. Rewriting equation (2) as

$$x^0 b_0 + x^1 b_1 + x^2 b_2 + \dots = f(x), \tag{3}$$

will help prepare us to understand the linear algebra we will need. Now we can write equation (3) with vectors as

$$\begin{bmatrix} x^0 & x^1 & x^2 & x^3 & x^4 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} f(x) \end{bmatrix}. \tag{4}$$

Clearly with equation (4) we can only input one point $(x_i, f(x_i))$ which does not provide enough information to solve for reasonable values in $\vec{b}$, the vector of coefficients. In figure 1 we have 6 samples, so we can repeat equation (4) for each sample with the same coefficients. We can combine all six equations into one like so

$$\begin{bmatrix} x^0 & x^1 & x^2 & x^3 & x^4 \\ x^0 & x^1 & x^2 & x^3 & x^4 \\ x^0 & x^1 & x^2 & x^3 & x^4 \\ & & \vdots & & \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} f(x) \\ f(x) \\ f(x) \\ \vdots \end{bmatrix}. \tag{5}$$

Now we can see that each row in the x matrix corresponds with a row in the $f(x)$ vector. Each row represents one of our sample points, thus denoting 6 rows. This relationship between rows and individual sample points can be better visualized if subscripts are added to each respective value of x,

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 \\ & & \vdots & & \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \end{bmatrix}. \tag{6}$$

And finally with the true values of A and $\vec{y}$ inserted,

$$\begin{bmatrix} 1 & 0 & 0^2 & 0^3 & 0^4 \\ 1 & 2 & 2^2 & 2^3 & 2^4 \\ 1 & 4 & 4^2 & 4^3 & 4^4 \\ 1 & 6 & 6^2 & 6^3 & 6^4 \\ 1 & 8 & 8^2 & 8^3 & 8^4 \\ 1 & 10 & 10^2 & 10^3 & 10^4 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} f(0) \\ f(2) \\ f(4) \\ f(6) \\ f(8) \\ f(10) \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 4 & 16 & 64 & 256 \\ 1 & 6 & 36 & 216 & 1296 \\ 1 & 8 & 64 & 512 & 4096 \\ 1 & 10 & 100 & 1000 & 10000 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 4.06427214 \\ 3.26415222 \\ 2.01261424 \\ 1.31046021 \\ 0.94640926 \\ 0.77709839 \end{bmatrix}. \tag{8}$$

Equation 6 is the expanded form of the fundamental linear algebra equation,

$$A\vec{b} = \vec{y}. \tag{9}$$

If the A matrix were square and invertible, solving for $\vec{b}$ would be as simple as $\vec{b} = A^{-1}\vec{y}$. Because there are more equations than unknowns, this will be an *over-determined* system; the alternative would be an *under-determined* system, one with more unknowns than equations. In either case, the solutions to the linear systems become non-unique. This means there is not a single set of values for $\vec{b}$ that can solve Equation 9. In the example given, calculating one sample is quite easy, but in modeling more complicated systems, each sample becomes computationaly expensive. It will be later seen that the systems used will generally be underdetermined. Solving for the vector of coefficients can then be done by

$$A\vec{b} = \vec{y} \tag{10}$$

$$(A^T A)\vec{b} = A^T\vec{y} \tag{11}$$

$$\vec{b} = A^T\vec{y}, \tag{12}$$

where A^T denotes the transpose of A.

Now that $\vec{b}$, the vector of coefficients, has been solved for, the model can be used to predict the y position of any x value within the range of sample points. To produce a continuous visual of the model's fit, the model can take in x values for every point in the sample range and plot its respective evaluation. The result of this process can be seen in Figure 2.

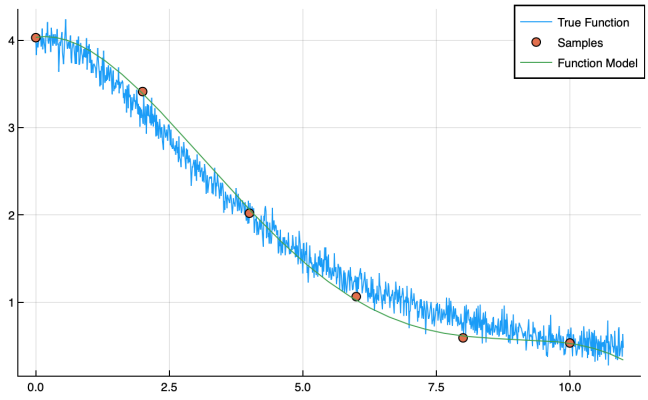


FIG. 2: The true function is shown by the blue line and the points are the locations of our samples. The green line shows the function created by our model to predict the true function.

B. Basis Functions and Sample Size

As seen in the previous example, and in particular Equations 6, 7, and 8, each row of matrix A and vector $\vec{y}$ represents a single sampling and each column is a unique basis function. It should be understood that as either of these two variables increases, thus increasing the length or width of A, so should the accuracy and precision of the model increase. In theory, there is no limit to the number of samples or basis vectors that could be used to construct a model, but in reality, there is a balance between the former variables and computational capabilities. In the example given, an increase in the number of samples or basis functions will not dramatically affect the computational power required, but becomes a greater concern for models of systems with increasing complexity.

As discussed, the quantity of basis functions can make a large impact, but another important factor is quality. Though the choice of basis functions in the example above was simple, it can often be a difficult choice. Consider, for example, a Fourier basis. The equivalent of Equation 6 in this basis would be

$$\begin{bmatrix} \sin(x_1) & \sin(2x_1) & & \\ \sin(x_2) & \sin(2x_2) & \dots & \dots \\ \sin(x_3) & \sin(2x_3) & & \\ \vdots & & \ddots & \\ \sin(x_n) & \dots & & \sin(mx_n) \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_n) \end{bmatrix}, \tag{13}$$

for an $n \times m$ matrix.

When used in the proper circumstance, this Fourier basis can be an excellent choice for modeling a function, as in Figure 3.

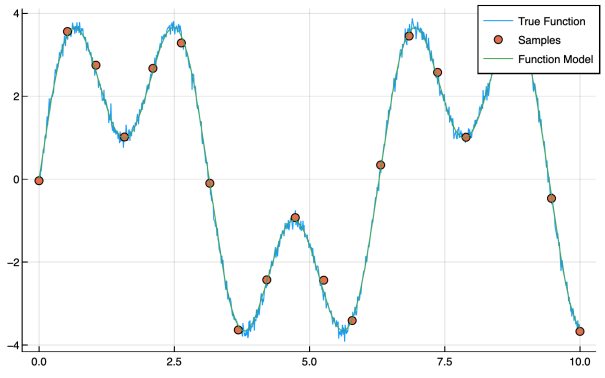


FIG. 3: The sample points shown were used to build a model of the function in green, and the true function is displayed in blue. As can be seen, a Fourier basis resulted in an excellent fit.

When chosing sample points, there are again two things to consider: number and breadth. As will be seen later, the number of samples can have a significant impact on the time it takes to construct a model and the accuracy of said model. The breadth of samples is similarly important. If all samples from Figure 3 were taken between 0 and 1, the model produced would poorly estimate $f(2.5)$, as in 4.

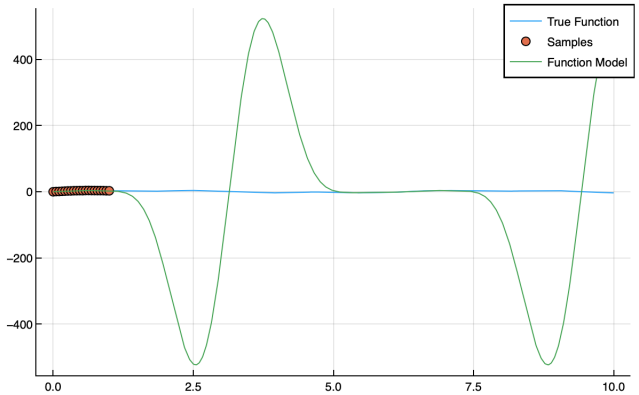
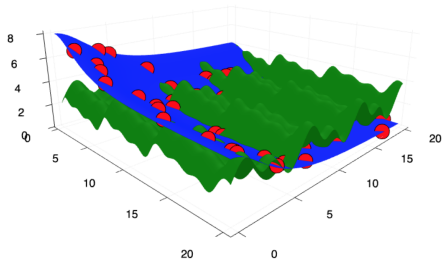
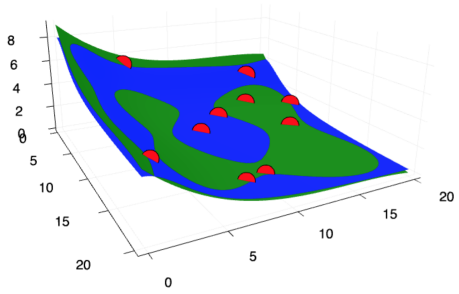


FIG. 4: A model with an insufficient breadth of samples will naturally produce a poor model. In this case, samples were only taken from the interval 0 to 1. If the model were evaluated at $x = 2.5$, the fit would produce a highly inaccurate result when compared to the true function.

By this point it should be obvious to the reader that the decision of number and breadth of sample points as well as quantity and quality of basis functions is critical to the model's performance. With well chosen



(a) Fourier basis. A is a 50×7 matrix.



(b) Power basis. A is a 10×7 matrix.

FIG. 5: A three dimensional function in blue with a constructed model in green. The red points show the sampling of the true function. (a) In this case, a Fourier basis is a poor choice and resulted in an inaccurate model of the true function. (b) A Power series makes a convincingly better fit even with significantly fewer sample points.

basis functions but poor breadth of samples, a model's accuracy can be greatly limited, as in Figure 4. Figure 5a shows the reverse situation, good number and breadth of samples, but poorly chosen basis functions.

C. Julia

The figures provided throughout this paper were produced using the Julia programming language. Though encouraged to use Python for the majority of my formal education, Julia is a relatively new and fast growing language in terms of popularity.

III. PRELIMINARY MODELING

A. Lennard-Jones Potential

Before jumping to quantum mechanical data, this math will be tested to construct models on a simplified potential, the Lennard-Jones potential. Though the Lennard-Jones potential is a simplification of reality, it does a excellent job representing real intermolecular forces of attraction and repulsion. The potential is a function of distance between two particles given by

$$V_{LJ}(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{14}$$

where ε and σ are constants for a given particle interaction. Figure 6 shows the plot of this potential.

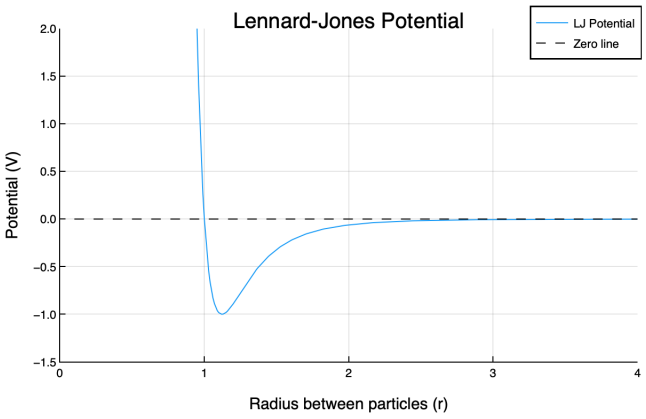


FIG. 6: The Lennard-Jones potential. A simple yet realistic model of intramolecular forces.

It can be recalled that the force from a potential is given by

$$F = -\nabla U, \tag{15}$$

and thus the force between two particles is zero at the bottom of the potential well. With that location as a reference, distances any greater will produce a force that is attractive and at any lesser distances, the force is intensely repulsive.

Calculating the potential between two particles is not difficult, but finding the total potential energy of a system of several particles becomes increasingly computationaly expensive. Because of the simplicity of the Lennard-Jones potential, the computational power required to solve for the system's energy is still relatively small. In preparation for using real, quantum-mechanical data, each system's energy will be treated as expensive to compute.

B. Constructing a Model

Ensuring a sufficient number and breadth of samples as well as reasonable basis functions becomes difficult as the number of dimensions goes beyond 2 or 3. From studying a variety of potential basis functions, bessel functions of the second kind, $Y_\alpha(x)$, have the potential to be useful.

Rewriting Equation 9 in component form yields

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1m} \\ A_{21} & & & \\ \vdots & & & \vdots \\ A_{n1} & \dots & & A_{nm} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \vdots \\ V_n \end{bmatrix}. \tag{16}$$

Recognizing each row of A as a unique sample, Equation 16 can be written as a system of linear equations

$$A_{11}b_1 + A_{12}b_2 + A_{13}b_3 + ... + A_{1n}b_n = V_1 \tag{17}$$
$$A_{21}b_1 + A_{22}b_2 + A_{23}b_3 + ... + A_{2n}b_n = V_2. \tag{18}$$

Each element of A will be populated with

$$A_{11} = Y_0(\alpha_{01}r_{12}) + Y_0(\alpha_{01}r_{13}) + Y_0(\alpha_{01}r_{23}) + \dots \tag{19}$$
$$A_{12} = Y_0(\alpha_{02}r_{12}) + Y_0(\alpha_{02}r_{13}) + Y_0(\alpha_{02}r_{23}) + \dots \tag{20}$$
$$A_{1n} = Y_0(\alpha_{0n}r_{12}) + Y_0(\alpha_{0n}r_{13}) + Y_0(\alpha_{0n}r_{23}) + \dots \tag{21}$$

where α_{0n} is the n-th zero of J_0, the zeroth bessel function of the first kind.

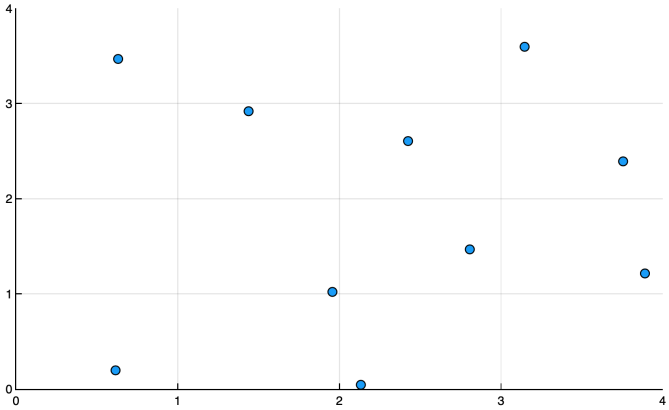


FIG. 7: A random assortment of 10 particles in a 4x4 square. No two particles are allowed to be within a pre-specified distance of each other.

- Minimum separation distance
- Number of particles in box
- Sample/function num
- Why separation vector norms?
-

C. Multi-Type Particle Systems

Now that a simple monotomic model has been constructed, it can be adjusted to handle two different types of particles. The different particles can be labeled type-A and type-B. This diatomic system will now require *three* different potential equations, one describing each type of interaction. The three unique interactions are type-A interacting with another type-A, a type-A interacting wtih a type-B, and a type-B with another type-B. Because these are arbitrary interactions, they can be defined by choosing reasonable values of ε and σ from Equation 24.

$$V_{AB}(r) = 4\left[\left(\frac{1}{r}\right)^{12} - \left(\frac{1}{r}\right)^{6}\right] \tag{22}$$

$$V_{AA}(r) = 4(0.7)\left[\left(\frac{0.8}{r}\right)^{12} - \left(\frac{0.8}{r}\right)^{6}\right] \tag{23}$$

$$V_{BB}(r) = 4(0.4)\left[\left(\frac{1.1}{r}\right)^{12} - \left(\frac{1.1}{r}\right)^{6}\right] \tag{24}$$

The graph of each potential can be seen in Figure 8. Each of the equilibrium positions and energies is slightly different, but all in the same neighborhood.

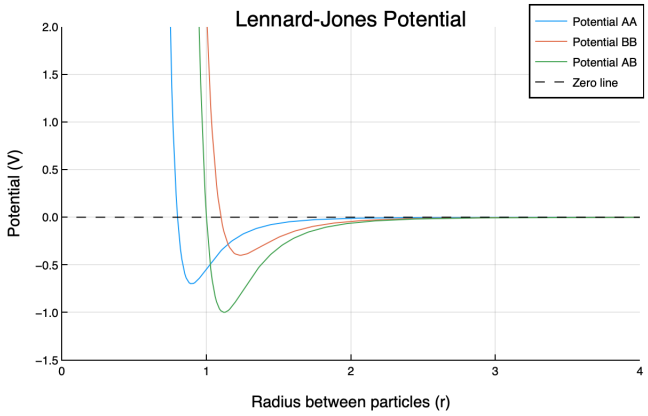


FIG. 8: A Lennard-Jones potential for each particle interaction type (AA, AB, and BB). Each interaction potential has a slightly different equilibrium position and energy.

To handle this increase in complexity, matrix A will need to contain *three* columns where the previous model had only one. This is again because of the three different interaction types, one column for each. Another obstacle arises in the decision for particle type ratio. If the particle number (10) and box size (4×4) remain constant, how many type-A versus type-B particles should there be? The question is really just a general question specified to this case; what is a sufficient breadth of samples for this model? As previously discussed, this depends greatly upon the desired range of predictions. As will be seen in Chapter **??**, the configurations vary in particle number as well as ratio. Therefore, this model should be trained and tested using a variety of particle ratios. Each training and testing set will thus be given a random number and ratio of particles to be populated in and calculated.

Once a model has been properly constructed and trained, how can its accuracy be easily determined and visualized?

- Particle type ratio
- Show model accuracy

IV. PROCEDURE

A. Quantum Mechanical Data

Brother Nelson gave me a bunch of data...

Because effective models could be contructed with the Lennard-Jones potential, a similar model could potentially be used to train from and predict useful data faster than current models. The data discussed and used in this section was generated by and received from Lance Nelson. It was generated using a particularly complicated model that incorporated principles of math and quantum mechanics that are beyond the scope of this paper. It is important to understand that the process of obtaining said data was computationally expensive and difficult. The culmination of this research is to produce a simpler model that can be trained on the given data then reproduce expected energies within a range of reasonable error. For a reference, Figure 9 shows a histogram of all the configuration energies in the data set.

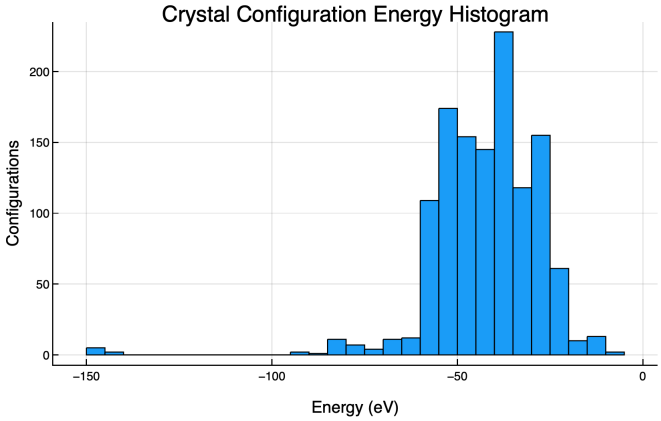


FIG. 9: A histogram showing all the configuration energies in the original data file. The majority lie within the -20eV to -60eV range, with a few outliers around -150eV.

Each configuration in the data represents a unique primitive unit cell. A unit cell is the building block of any crystal structure. Each unit cell is an identical copy of another, with the same shape, size, and contents.

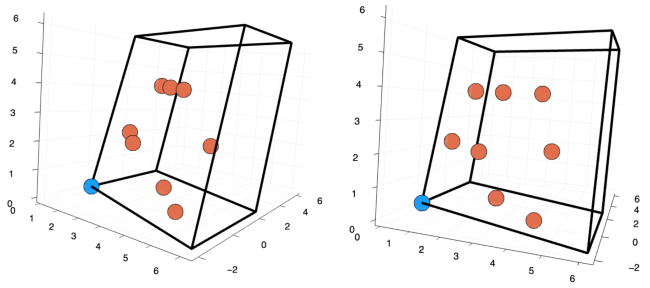


FIG. 10: The primitive unit cell for configuration 65 from two different perspectives. The blue point is the single Ag atom while each red point shows each Pt atom in said configuration.

A *primitive* unit cell is the smallest possible unit cell which contains only one of each uniquely positioned atoms in the crystal. An example of a primitive unit cell configuration can be seen in Figure 10.

In order to use the data in the *.txt* file, it needs to be parsed into vectors that can be easily manipulated. An example of the data being parsed can be seen in Figure 11. The "str" can be identified in the first line to begin each new system. The number on the second line is the lattice parameter followed by three lattice vectors in (i, j, k) coordinates. The following line contains two numbers, the first is the number of silver (Ag) atoms in the unit cell and the second is the number of platinum (Pt) atoms. In direct coordinates (in terms of the lattice vectors) we are given the positions of each silver and platinum atom in order. The last line in this system tells the total potential energy of the unit cell configuration.

```
Ag-Pt str #: 100
   4.06600000000000
     0.4897375027131054    0.4897375027131054    0.0000000000000000
    -0.9857506444128054    0.9857506444128054    0.0001480742989182
     0.4911874202245589   -0.0014499175114533    1.5317387854017932
   3   3
Direct
  0.9962945910437568  0.0016543423460078  0.0074108179124863
  0.3399413516957068  0.8293793395082005  0.3201172866085926
  0.9989260447531194  0.5020634528287573  0.0021479104937541
  0.6671639334432413  0.6696967385982830  0.6656721431135040
  0.3328857305662609  0.3335677384713236  0.3342285288674844
  0.6647883484979076  0.1636383882474277  0.6704233130041786
 Energy:-26.60832
```

FIG. 11: An example of the data given in AgPtdata.txt that needs to be used to build the model.

B. Model Construction

With the file parsed and the important data retrieved, the process of building a model can commence. The potential energy of a single particle is due to its interactions with all its surrounding particles, thus the unit cell must be propagated outwards in all three dimensions. Then the relative position of each affecting particle can be determined and the vector separating the particle pair can be calculated. These separation vectors are the information that will be passed into the basis function to construct the model.

Because the affect two particles have on each other drop off as a function of distance, the unit cell does not need to be propogated infinitely in each direction, only out to a sphere of reasonable influence, beyond which all interactions are negligible. Clearly the choice of radius for said sphere will make a significant impact on the quality of the model. It is also obvious that the one arbitrarily chosen radius cannot be applied effectively to each unique system. A simple solution is to make the radius of this sphere of influence can be chosen to be the magnitude of the largest lattice vector multiplied by a constant. Later the effect of the constant on the model's precision can be tested, but for now will be chosen to be 1.2.

This simplification from the "sphere of influence" gives a clear upper bound to our unit cell propagation. It is expected that the unit cell will be propagated out further than the radius in each direction, thus encapsulating said sphere. An example of this iterated unit cell can be seen in Figure 12.

Now each of the unit cells in Figure 12 can be populated with all the atoms contained within. From that fully populated unit cell can be extracted only the atoms inside the sphere. When these atoms are stored into a new vector, they can be re-plotted as in Figure 13. When this process is repeated for every system in the given data, the result is a vector containing these "important positions" for each unique silver and platinum atom in each unit cell. The separation vectors can be quickly calculated and their norms stored in three separate vectors, one for each type of interaction (Ag-Ag, Pt-Pt, and Ag-Pt).

With all the pertinent information from the data set organized into vectors, composition of the A matrix can begin. As in the corollary Lennard-Jones example, each row represents a crystal structure and each column corresponds to a unique basis function. Also as in the Lennard-Jones code, a single basis function will result in three columns of A due to each type of interaction. With 300 basis functions, the first 300 columns will be calculated using 300 unique basis functions with separation vector norms from Ag-Ag interactions. The following 300 columns from Pt-Pt interactions and the final 300 from Ag-Pt. The size of the training set is limited to 1224, the number of systems given in the data set. Whatever systems are not included in the holding set will comprise the validation set.

The basis functions will again be chosen to be bessel functions of the second kind. This is because of the similarity in potential energy between two atoms as a function of distance and the shape of the graph of said bessel functions.

Once the model is fully constructed, tests can be run to determine the effect of radius, sample number, and

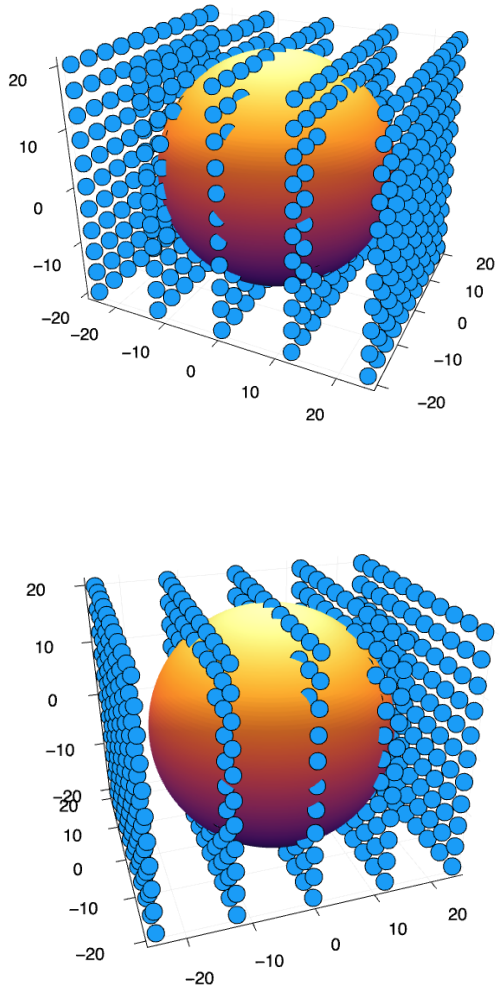


FIG. 12: The sphere of influence for one silver atom is completely encapsulated by the unit cell's iterations from two different perspectives. Only one particle from each unit cell is shown.

number of basis functions on the precision and accuracy of the model. An increase of radius, sample number, or number of basis functions will make significant impacts on the program runtime. As the radius is increased, the unit cell must be propagated out further and more atom interactions considered. Thus effect of the radius will drastically impact the total runtime. For every basis function added, there will be three columns added to the A matrix, drawing out its composition and calculation times.

Generally, as the number of basis functions is increased and the radius of influence extended, the model's predictions will become increasingly reliable. On the other hand, as those factors increase accuracy and precision, they also increase the computational costs. It

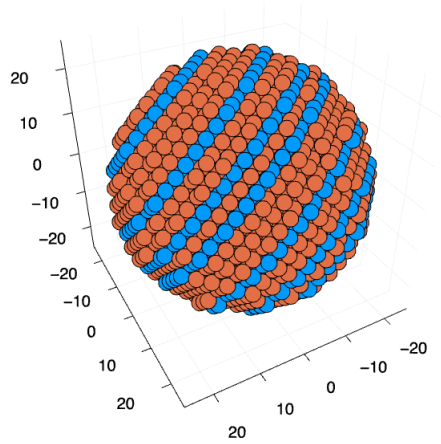


FIG. 13: Every silver and platinum atom inside the sphere of interest. According to the model, these are the only atoms interacting with the central atom in question. This data is taken from the first silver atom in the third sample system.

would be ideal to find a manageable trade off between the program's runtime and reliability. To investigate the quality of the model as a function of cutoff radius, sample number, and number of basis functions, a sweep can be conducted over a range of values for each and record the details of the fit for each combination. Rather than run this code for weeks on a normal desktop computer, it was completed on Mary Lou, BYU's supercomputer.

V. RESULTS AND ANALYSIS

Introduction lines... Test

The quality of a single fit is given in Figure 14. For some models, there are predictions that differ significantly, more than 100eV, from their true values. In those cases, the number of outlying crystals has been counted then excluded from the average error calculation. In this way, the average error can become independent of the model's outliers. To understand a given model's accuracy and precision visually, each prediction energy can be plotted against the actual energy for each configuration. The model referenced in Figure 14 can be seen again in Figure 15.

```
Number of samples:                  800
Number of basis functions:          500
Radius r:                           1.5
Number of energy predictions:       424
Number of errors greater than 100.0: 3
Average error:                      3.365961170977803
```

FIG. 14: An example of the data in the output file.

The method of analysis in Figure 15 works very well for observing one model at a time but interpreting each

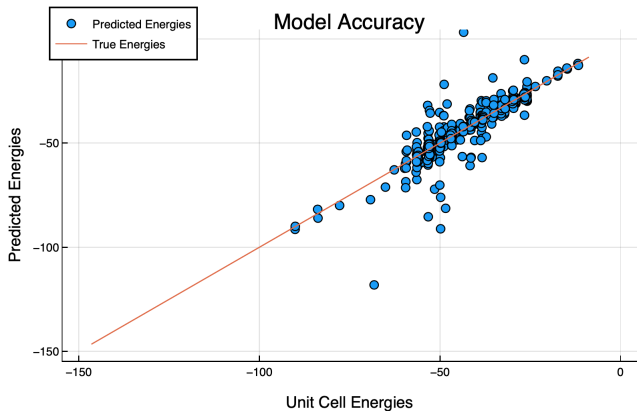


FIG. 15: Predicted energies versus the actual energies. The model tested is the same as the one shown in Figure 14.

of the 270 unique models this way would difficult and time consuming. It would be preferable if several data sets could be interpreted at one time.

One solution could be a series of heatmaps showing the average error of each model with it's respective parameters. But as discussed above, the average error does not tell the whole story, the number of large errors must also be accounted. Figure 16 shows five heatmaps, one for each size of training set. The color in each square represents the average error for a specific model. The scale for each subplot is set to be identical to make its analysis easier. A similar cluster of plots can be seen in Figure 17 showing the number of large errors for each model.

The true accuracy and precision of each model can only be realized when comparing its results from Figures 16 *and* 17. One without the other does not show the full picture. For example, the lower left corner of Figure 16a appears to be surprisingly accurate but when compared to Figure 17a, it can be seen that these models actually produce a considerable number of large errors. These particular models are not of interest. The models that *are* of interest will be the squares from corresponding subplots in Figures 16 and 17 that are both blue or blue-ish. It should be noted that one of the reasons the lower left corner of Figure 16a is so blue is because the large errors, as seen in Figure 17a, are all excluded from the overall average error.

Figure 17a includes a large red clump in the bottom right of the plot. This red patch then appears to move upward in each ensuing figure. .....

The most accurate and precise model was trained on 1000 sets, used 40 basis functions, and had an effective radius of 1.0. This model produced 0 large errors and had an average error of only 1.77eV. The second and third best models had very similar input parameters, changing by one step either the size of the training set or the number of basis functions.

Why did this model work in some places, but not in others? .....

Because this model was produced using only pair-interactions, implementation of three body interactions has the potential to greatly increase the accuracy of this model. This is, unfortunately, beyond the scope of this research.

The success of this simple model may be unique to this silver platinum bond and other particular crystals. But frankly, I have no clue. .....

VI. CONCLUSION

The models produced were satisfactorily successful. Each provided great insight into which parameters build useful models. Though some models are useful, none of them are without errors. Using the "best" model, as mentioned in Section V, new data on silver-platinum crystal configurations can be produced. The use of this data will be discussed in section VI A.

A. Future Work

The code writen to produce the model in Section IV B can be easily adapted to construct models with new data. Such research could be used to determine which alloy energies can be accurately predicted using a simple model. This model, as well as other potentially successful models, can be used to create theoretical phase-diagrams for these alloys without the need to chart them through experimental means [1] [2]. This exercise has been left for the reader to complete (haha!) .....

[1] H. T. Stokes, *Solid State Physics for Advanced Undergraduates*, 4th ed. (Brigham Young University, 2007).
[2] Lincoln, Physics Journal (2021).

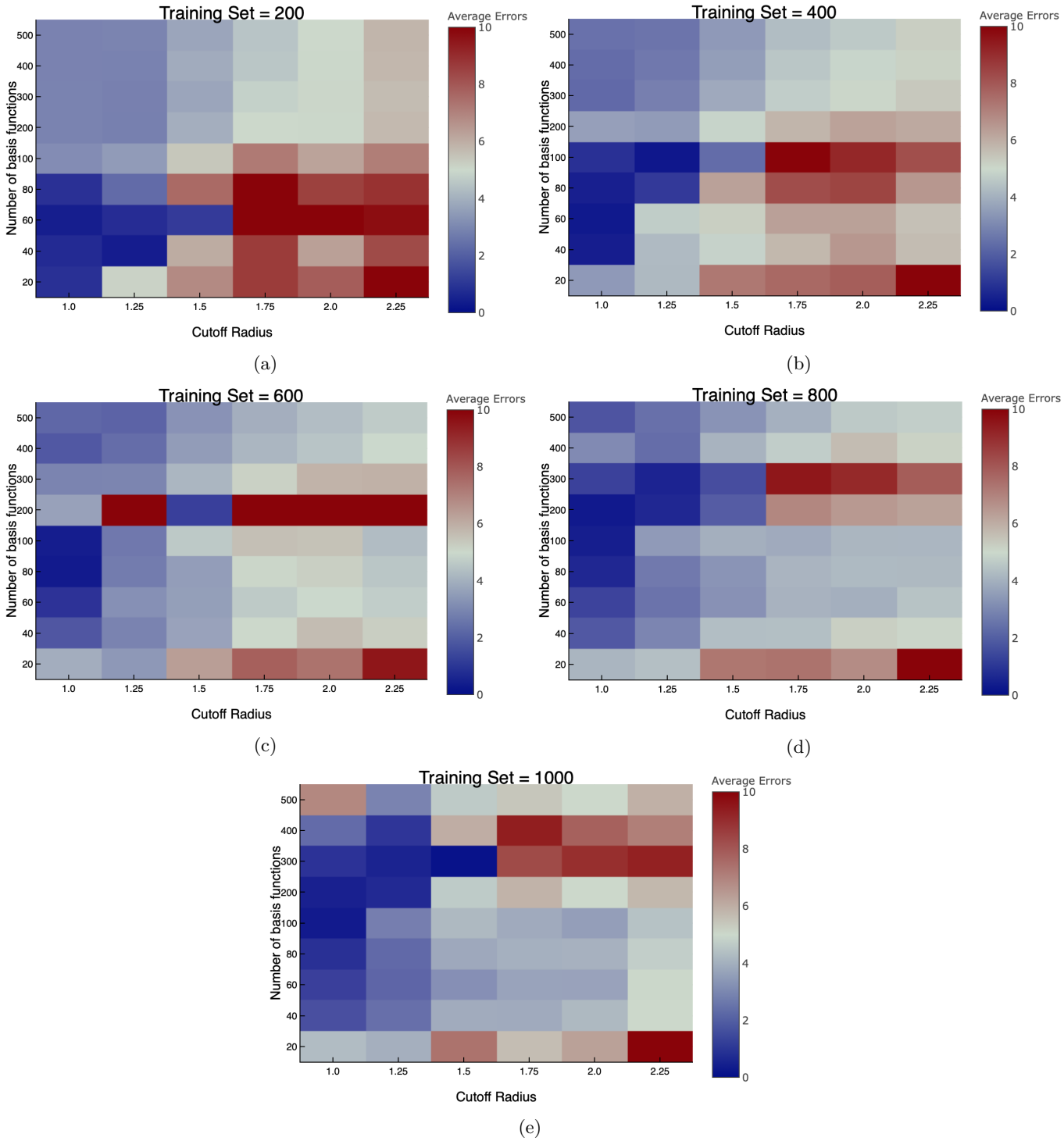


FIG. 16: Heatmaps showing the average error of each model produced. The scale for each is fixed from 0eV to 10eV. The effect from removing the large errors from the average can be seen by comparing the more red areas from Figure 17 with the dark blue areas here.

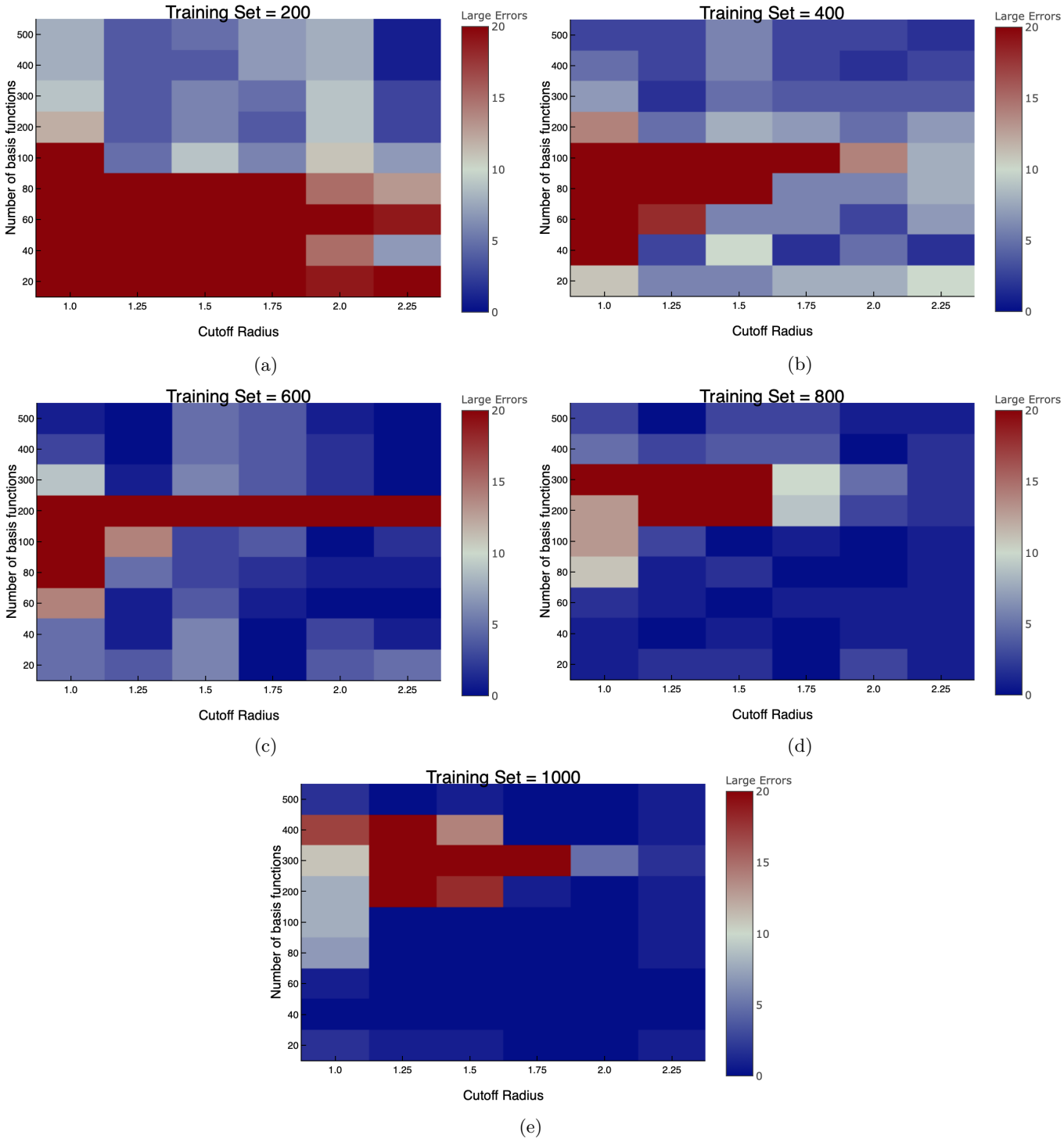


FIG. 17: Heatmaps showing the number of large errors of each model produced. The scale for each is fixed from 0 to 20 errors. Because the scale is fixed, any square with a deep red color has a minimum of 20 errors.