

PHYS 760 Class Project Adaptive Quantum Design

Gautam Rai

Department of Physics and Astronomy, University of Southern California, Los Angeles, California 90089-0484, USA
(Dated: Summer 2020)

This project is aimed at introducing modern search algorithms and machine learning techniques to the physics community via the problem of adaptive quantum design—searching the space of configurations of a physical model for the optimal configuration that exhibits a desired response. Following Thalken et al.¹, we find the optimal positions of N atoms on a 2D plane such that the density of states of the corresponding tight-binding Hamiltonian takes a particular desired form. We show that this can be achieved using a local search scheme. Further, we explore using the Kernel Polynomial method to calculate the density of states in the inner loop. This improves resource consumption scaling to $\mathcal{O}(N^2)$ from $\mathcal{O}(N^3)$.

I. INTRODUCTION

The laws of physics allow one to map a given physical system to its response—the signal a scientist would see if they were to probe it with a particular perturbation. When building devices, the inverse question is more relevant—what are the details of a physical system that would produce a desired response. The most general version of the problem presents the following challenges: 1) The inverse map from response to system is only implicitly defined. One cannot input the desired response into an equation, one must *a priori guess* the correct system. 2) The inverse map is not unique. There can be many degenerate solutions; the energy landscape is flat. 3) The space of possible physical systems is huge. Brute force search is implausible.

In this project, we aim to use modern search algorithms to treat this problem in the context of two-dimensional (2D) tight binding models. First, we develop a local search scheme that approximates the solution well. Second, we improve the scaling of the algorithm by using an approximate estimate for the objective function. Finally, we discuss embellishments to the basic local search core to further improve performance.

II. TIGHT BINDING MODELS

In 1989, researchers at IBM showed that they could use a scanning tunneling microscope (STM) to move individual Xenon atoms with remarkable precision to spell out the initials of their employers’ name.

The tight binding model is one way to model the electronic properties of such arrangements of atoms. Consider a solid made up of a regular arrangement of one species of atoms. For example, diamond made up of Carbon atoms arranged in a tetrahedral structure, or crystalline Aluminium made up of Aluminium atoms arranged in an FCC lattice. The electronic properties of such a material depends primarily on two factors: the wavefunctions of the outermost or valence electrons of the constituent atoms, and the details of the way the

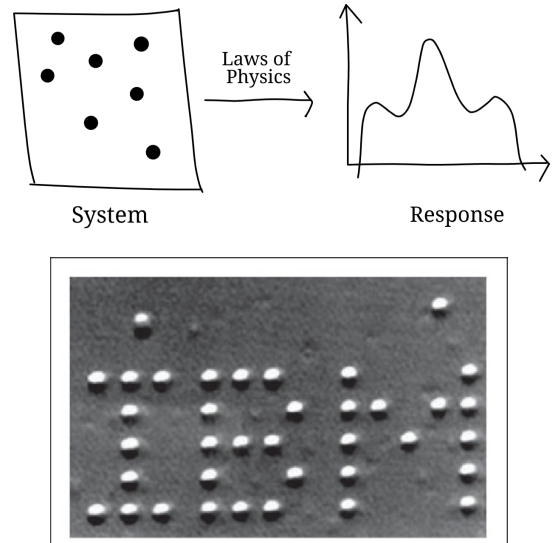


FIG. 1. (top) A typical physics problem. (bottom) An STM image of Xenon atoms on a Nickel substrate manipulated to spell out “IBM”.

atoms are arranged. The first one contains information about the way each electron interacts with its host ion. The second one contains information on how the electron is influenced by all the other ions. In the tight-binding model, these effects are collectively described using an $N \times N$ matrix Hamiltonian for N atoms that has two kinds of contributions to the energy: an on-site potential that goes in the diagonal of the matrix, and the off-diagonal *hopping* terms that encode the overlap between given sites.

In our simplified scenario, the diagonal terms are all zero, and the hopping terms depend on the distance between two atoms in simply as $t_{ij} = \text{dist}(i, j)^{-3}$. This means that the system, is fully described if the locations of all the atoms are known. A given configuration of N atoms is then encoded by a list of N tuples representing the positions of each of the atoms. The configuration space of the problem depends on the size of the plane

and the resolution of the grid. As the grid gets finer, it approaches the limit of atoms in continuous space.

$$\hat{H} = \underbrace{\begin{pmatrix} 0 & t_{01} & t_{02} & & \\ t_{10} & 0 & t_{12} & & \\ t_{20} & t_{21} & 0 & \ddots & \\ & & \ddots & \ddots & \\ & & & & \ddots \end{pmatrix}}_{N \times N \text{ matrix}} \quad (1)$$

In this introduction, we have used monoatomic crystalline materials, there is a straightforward but tedious generalization to polyatomic (containing atoms of different types) and disordered (ion locations are irregular) materials.

III. ELECTRONIC DENSITY OF STATES

The density of states (DOS), $\rho(E)$, describes how many states an electron can occupy at a certain energy in a given system. It is defined such that $\rho(E)dE$ is the number of states in the energy range $[E, E+dE]$. It is mathematically represented by a probability distribution function. In a bounded system, with a finite number of total states, the density distribution is discrete. Since every state has a well-defined energy, the density of states is made up of Dirac delta functions at the energy of every state, E_i .

$$\rho(E) = \frac{1}{N} \sum_i \delta(E - E_i) \quad (2)$$

the sum runs over all the N states available to the system.

In the tight-binding model, the energies of the available electronic levels, E_i , are simply the eigenvalues of the matrix Hamiltonian (1).

A. Smoothed DOS

$\rho(E)$ can be measured by way of the tunneling currents that are induced when a doped scanning tunneling microscope (STM) tip is brought close to the system. However, the experimentally measured density of states takes a form different from (2). In place of infinitely tall delta function peaks, the device measures smeared out peaked functions. The smoothed density of states can be obtained from (2) by drawing peaked functions such as Gaussians or Lorentzians centered around the eigenvalues as shown in Fig. 2. The spread of the peaked functions must be chosen with care. The correct choice depends on the size of the system being measured, the resolution of the measurement, and the size of the system simulated. The smoothed density of states is closely related to the density of states obtained from the dispersion relation $E(k)$ using:

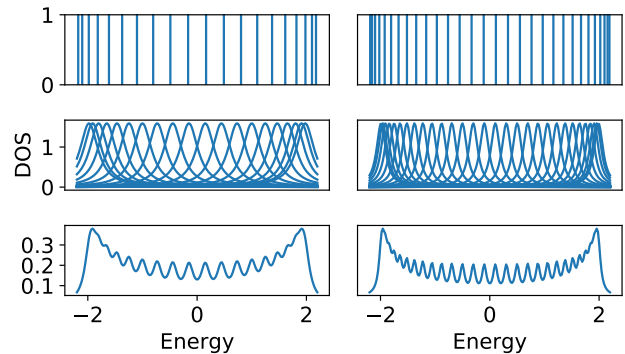


FIG. 2. Bare and smoothed representations of the discrete density of states of finite systems—linear nearest neighbor chain with 20 sites (left) and 30 sites (right). The vertical lines on the top panel represent the delta functions in (2). The middle panel shows the delta functions replaced by Lorentzians with width 0.2 (left) and 0.1 (right). The bottom panel shows the smoothed density of states obtained by summing up the Lorentzians in the middle panel.

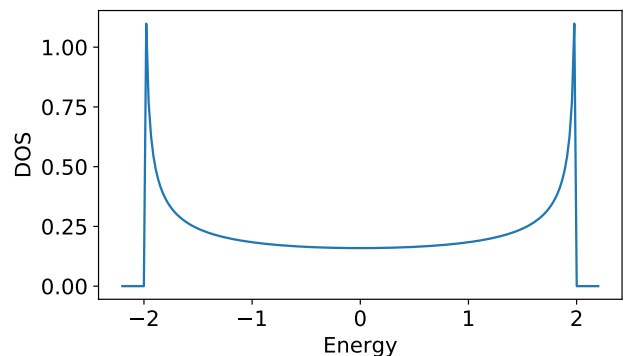


FIG. 3. The density of states obtained from the dispersion relation of the linear nearest neighbour chain using (3). This is equivalent to considering (2) for an infinitely long chain and then repeating the smoothing procedure describe in Fig. 2 with Lorentzians of infinitesimal width.

$$\tilde{\rho}(E) = \frac{\partial \mathcal{I}[E(k)]}{\partial E} \quad (3)$$

where $\mathcal{I}[E(k)]$ is the inverse of $E(k)$.

$\tilde{\rho}(E)$ can be interpreted as the density of states of the same system if it were scaled up to be infinitely large. $\tilde{\rho}(E)$ for the linear nearest neighbor chain is shown in Fig. 3.

B. A note on implementation

In this study, I've used the tophat function as my target density of states (see top panel of Fig. 4). It is natural to interpret it as $\tilde{\rho}(E)$, the scale-independent infinite

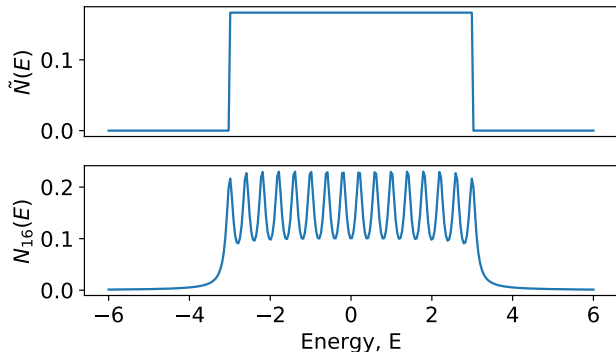


FIG. 4. The map (4) applied to the tophat target function with $N = 16$

limit density of states. In the practical implementation, one must apply a map, σ_N , that converts this to a finite smoothed density of states that depends on the number of particles in the system.

$$\sigma_N : \tilde{\rho}(E) \rightarrow \rho_N(E) \quad (4)$$

For instance, applying this map to the infinite limit density of states in Fig. 3 would recover the left (right) plot in the bottom panel of 2 if we set $N = 20(30)$. As a practical matter, applying σ_N to the target before running the search algorithm results in markedly improved time to convergence.

IV. LOCAL SEARCH

A first approach to the problem is to search the configuration space using a randomized local search algorithm. First, all atoms are placed in randomly chosen sites. The main loop involves calculating the density of states for all the configurations that can be visited by moving one randomly chosen atom to each of the vacant sites in its neighbourhood. The size of the neighborhood R is the most significant parameter choice determining the performance of the algorithm. The rough scheme is as follows:

```

randomly pick  $N$  occupied sites
do  $X$  times{
  randomly pick one of the atoms
  list all vacant sites up to  $R$  distance away
  move atoms to the site in the neighbourhood
  with lowest residual
}
end

```

V. RESULTS

In the top panel of Fig. 5, I have shown the initial, final, and target density of states for a run with $R = 4$

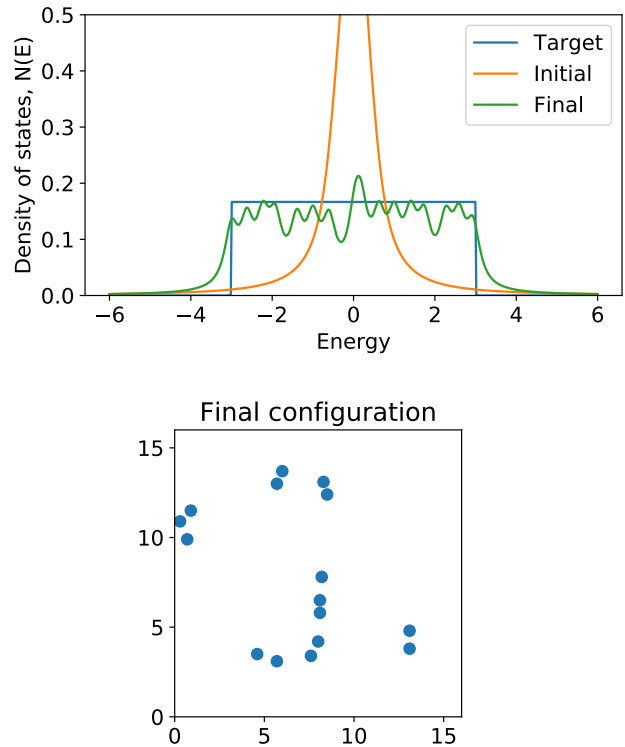


FIG. 5. (top) The density of states of the initial and final configurations after 500 iterations with $R = 4$. (bottom) The final configuration corresponding to the green curve above. Notice that atoms tend to pair up into dimers.

and 500 iterations. We see reasonably good convergence. Fig. 6 shows how the error varies with each iteration step. An inspection by eye reveals that $R = 4$ is a fairly good compromise between fast convergence and resource consumption as the number of operations scales as R^2 . In general, the optimal value of R will be problem (target) dependent. A quick glance at the $R = 5$ curve in Fig. 6 illustrates a major pitfall of the local search method. This run has hit a local minima and appears to be stuck in it. Finally, the bottom panel of Fig. 5 shows the final configuration corresponding to the green curve in the top panel. Notice how the atoms tend to pair up.

VI. THE KERNEL POLYNOMIAL METHOD

One of the main points of improvement of this algorithm is the resource scaling with size of the system. Representing N atoms using the tight-binding formalism requires an $N \times N$ matrix Hamiltonian. Every time the Density of states is calculated, the matrix Hamiltonian is diagonalized which takes $\mathcal{O}(N^3)$ operations. Since the density of states is calculated approximately R^2 times per iteration, the total time consumption of the program is $\mathcal{O}(N^3 R^2 X)$. R and X are system size independent, so improving the efficiency of the program relies largely on

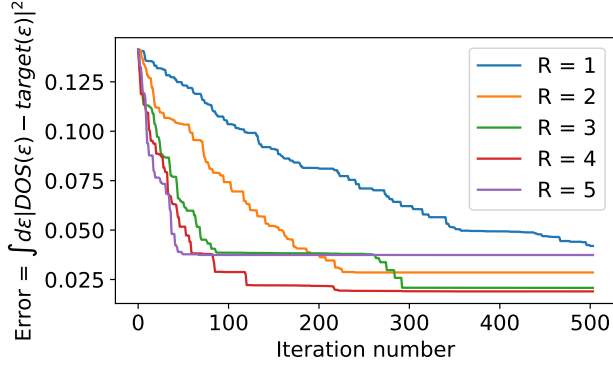


FIG. 6. The error as a function of the iteration number for various choices of R . The case $R = 5$ showcases an important pitfall—getting stuck in local minima.

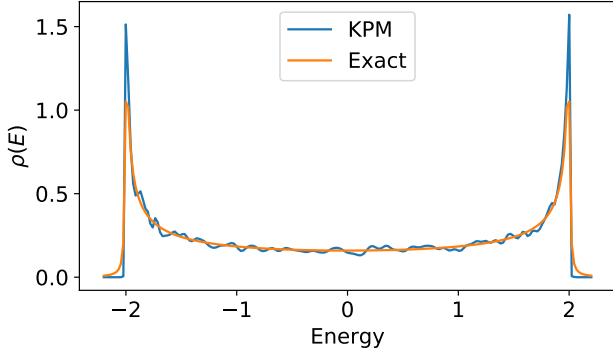


FIG. 7. Density of states of a linear nearest-neighbour chain with 1000 sites. Comparison of calculation by exact diagonalization and kernel polynomial method.

finding quicker ways to calculate the density of states.

The Kernel polynomial method (KPM)² is an approximate method that provides one such improvement. The core of the approximation is to expand the density of states in the basis of Chebyshev polynomials. Good convergence properties of the Chebyshev polynomials allow us to truncate the series early. The problem morphs into that of calculating a small number of coefficients (called Chebyshev moments). These Chebyshev moments can be calculated efficiently by taking advantage of two facts: The Chebyshev polynomials have certain recursion relations with which higher moments can be built off of lower ones. Traces of large matrices can be evaluated stochastically.

A. A brief description of the method

In this section, I've provided a brief summary of the steps that go into calculating the density of states with KPM. More details can be found in the review by Weisse et al.².

Chebyshev polynomials are only defined on the interval $[-1, 1]$. Therefore, the first step is to rescale the Hamiltonian $\hat{H} \rightarrow \tilde{H}$, such that all of its eigenvalues have magnitude less than 1.

Then, we can expand the rescaled density of states $\tilde{\rho}(E)$ in Chebyshev polynomials of the first kind.

$$\tilde{\rho}(E) = \frac{1}{\pi\sqrt{1-E^2}} \left[\mu_0 + 2 \sum_{n=1}^{\infty} \mu_n T_n(E) \right] \quad (5)$$

Here, the μ_i are simply the coefficients of the expansion. They can be calculated by:

$$\mu_n = \int_{-1}^1 \tilde{\rho}(E) T_n(E) dE = \frac{1}{N} \sum_i T_n(\tilde{E}_i) \quad (6)$$

$$= \frac{1}{N} \sum_i \langle i | T_n(\tilde{H}) | i \rangle = \frac{1}{N} \text{Tr}(T_n(\tilde{H})) \quad (7)$$

Here, we make our first approximation. Instead of calculating the trace in (6) exactly, we evaluate it stochastically based on the average over a small number $M \ll N$ of randomly chosen states $|r\rangle$.

$$\mu_n \approx \frac{1}{M} \sum_r \langle r | T_n(\tilde{H}) | r \rangle \quad (8)$$

The relative error of the trace estimate is of order $\mathcal{O}(1/\sqrt{MN})$. This implies that the number of vectors used for sampling can be kept constant or even reduced as the system size grows which makes this method very attractive for large calculations.

The second approximation comes from reconstructing $\tilde{\rho}(E)$ from a finite number of moments. While Chebyshev expansions have good convergence properties, a naive reconstruction will result in Gibbs oscillations around points with a discontinuous derivative. These oscillations can be controlled by modifying the moments slightly by applying the Jackson kernel.

Calculating the moments is simplified by the Chebyshev recursion relations. For convenience define $|r_n\rangle = T_n(\tilde{H})|r\rangle$. In the context of (8), the recursion relations amount to:

$$|r_0\rangle = |r\rangle \quad (9)$$

$$|r_1\rangle = \tilde{H} |r_0\rangle \quad (10)$$

$$|r_{i+1}\rangle = 2\tilde{H} |r_i\rangle - |r_{i-1}\rangle \quad (11)$$

As a consequence of this, generating a new moment only requires one application of a matrix to M vectors.

Finally, the reconstruction of $\rho(E)$ from the μ_n amounts to a cosine transform which can be performed efficiently with standard FFT distributions followed by a rescaling back to the original energy scale.

B. Results with KPM

Fig. 8 shows the results of the local search algorithm using KPM to evaluate the objective function. Unfortunately, I have not been as successful with recreating the

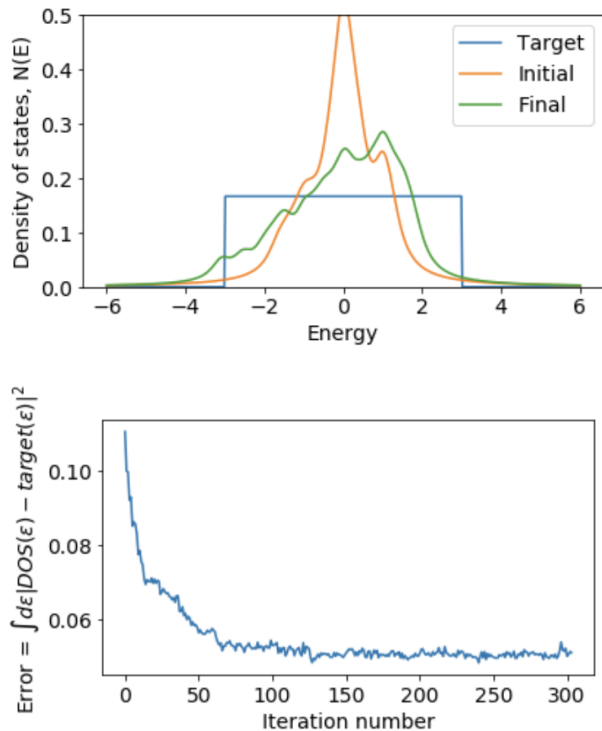


FIG. 8. (top) A comparison of the initial, final, and target density of states using the KPM version of the algorithm. (bottom) The error through the iterations

target state. There is however plenty of scope here as I did not have time to perform a proper investigation. The KPM method introduces several new control parameters such as the number of moments used and the number of vectors used. In all likelihood, there is more fine-tuning required before the algorithm is ready to make good predictions.

VII. FUTURE DIRECTIONS

In this section, I will discuss some observations I made through the course of working on this project. Each of these represents a potential improvement to the algorithm. The first is simply the fact that as with most

local search algorithms, this one is also vulnerable to getting stuck in local minima. This algorithm would benefit immensely with some manner of annealing—temporarily increasing the error in order to jump out of local minima. There is a small amount of annealing implicitly built into the algorithm—the atom must move to a neighbouring site even if the current site has the lowest error, but this is naturally not enough. Some potential ways to achieve this would be rapid randomized restarts and introducing a finite probability for moving to sites with large error. The second observation is that the tophat target function results in a tendency for atoms to pair up and form dimers. Under the assumption that any typical target function will lead to clustering of some form, it may be used to our advantage. One possibility is to periodically—in an outer loop—identify clusters and then to move the clusters rather than the individual atoms. Finally, every iteration of the local search involves recomputing the density of states for a configuration that is the same in every regard except for the position of one atom. It may be worth it to check if there is a way to calculate the change in the density of states rather than the entire function. I do not know if such a calculation is possible, but it could greatly reduce the resources consumption as the scaling with R would become non-existent.

VIII. CONCLUSION

Our local search based algorithm is potentially a good predictor of the best configuration of tight-binding atoms to achieve a target density of states assuming annealing is properly implemented. We tested kernel polynomial method approximate methods and we weren't fully successful but we showed that it should be possible to write an algorithm based on it. Finally, we discussed some other directions this project could go in. These include clustering analysis, and approximate density of state difference calculations. Finally, the density of states is one of a large class of spectral functions that can be calculated using KPM. If we could develop an algorithm to allow any of these spectral functions to be used as the target response, we would have a general-purpose tool on our hands.

¹ J. Thalken, Y. Chen, A. F. J. Levi, and S. Haas, Physical Review B **69**, 195410 (2004).

² A. Weisse, G. Wellein, A. Alvermann, and H. Fehske, Reviews of Modern Physics **78**, 275 (2006), arXiv: cond-