Computing the One-Electron Energy Band Structure within Metal

Leo Huang
Carnegie Mellon University
Pittsburgh Supercomputing Center

June 9, 2017

1 Introduction

This is a physics topic calculating the one-electron energy band structure within metal. I'm mainly in charge of doing the math part of it. In the frame of problem I will be working on, I have a given function

$$M : (-1.0, 1.0) \times \mathcal{B} \to \mathbb{R}^{n \times n}$$

M has two inputs: ϵ which stands for the one-electron energy and $k \in \mathcal{B} \subseteq \mathbb{R}^3$ which is the momentum of a free electron. The domain $\mathbb{B}$ is the first Brillouin zone, which is the smallest periodic unit cell in k-space. One important property of M is that it is twice differentiable. The problem I will try to solve is that given any $k \in \mathcal{B}$, we want to find ϵ such that $\det(M(\epsilon, k)) = 0$. With enough sampling of k, we can graph the band struct of ϵ against k.

2 Approaches

There are three ways to solve this problem:

The first way is simply by conducting a brute force binary search for all $\epsilon \in (-1.0, 1.0)$. Such process is tedious and takes really long time.

The second way is to triagonalize the matrix, so the determinant the the product of all entries on the diagonal. Then, simply by sampling $\epsilon \in (-1.0, 1.0)$ and observe the sign changes along the diagonal, we can find those ϵ that make the determinant 0.

The third way is most efficient and controls the error within an acceptable range. Note that we

can Taylor expand M given its twice differentiable. By some physics tradition, we can write

$$M(\epsilon_0 + \Delta, k) = H(\epsilon_0, k) + O(\epsilon_0, k)\Delta + A((\epsilon_0, k)\Delta^2 + \xi(\Delta)$$

where $\xi(\Delta)$ is the error function. By sampling enough $\epsilon_0 \in (-1.0, 1.0)$, we can control Δ to be small enough so that $\xi(\Delta)$ is negaligible. Then, we can reduce finding $\epsilon_0 + \Delta$ s.t. $\det(M(\epsilon_0 + \Delta, k)) = 0$ into find the solving Δ for

$$\det\left(\begin{bmatrix} -O^{-1}H + AO & -O^{-1}HO^{-1}H \\ I_n & -O^{-1}H \end{bmatrix} - \Delta I_{2n}\right) = 0$$

which is solving for eigenvalues.

My job is to implement all three approaches and benchmark all of them to show that the third way is much faster than the first two ways.

3 Implementation via High Performance Computing (HPC)

Since we are sampling a lot of $k \in \mathcal{B}$, and for each k, the calculations are independent, I will be using Message Passing Interface (MPI) to parallelize calculations for all different k. Also for the second and third method, sampling all ϵ or ϵ_0 can be also done in parallel, thus we the can further use MPI to speed up the process. Since the calculation involves a lot of simple identical operations across different threads, I will also consider using CUDA to speed up the process.