

论文简读

前言

本文首先简单解读我参与合作的论文(Lattice Vibration and Field Model)，然后简述我在整个合作过程中发挥的作用（主要是编程），并且谈一谈整个过程中遇到的问题以及体会。

论文简读

这篇文章发展了一个简单的场模型，显示了标量场和声子场的等价性。经过简短的计算，我们得到了和经典动力学矩阵方法相同的结果。（照搬summary部分的）

对于晶格振动问题，我们早已有简正模等理论来描述。

$$H = \sum_i \frac{\vec{p}_i^2}{2m} + V(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N) = V(\vec{x}_i - \vec{x}_j) = \sum_{i,j} \frac{\vec{p}_i^2}{2m} + V(\vec{x}_i - \vec{x}_j)$$

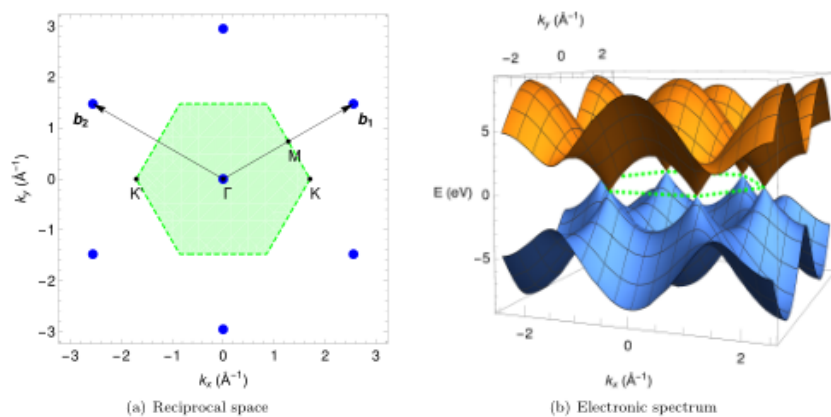
建立多个晶格的耦合哈密顿量，取低阶势能项；由平移对称性试探简谐波的解。这些办法相当完备。但是文章通过一些二次量子化的方法，在不同的格点模型中（一维链，二维六角格子，Kagome格子等等）成功导出了和经典方法一致的结果。

我的贡献

实际上由于理论水平的限制，我并没能提供太多理论上的想法。但是我有相当的编程能力，合作者描述清楚他需要计算哪些数，然后我可以找出合适的编程方法编出运行速度相对较快的程序，这是需要一点知识和技巧的。在整个过程中，我做得最多的基本上是求矩阵本征值，绘制能带图。其次是计算贝里曲率之类的物理量（但是我现在还不知道这些物理量的真实含义，我只是知道如何计算它）。再次之是求解一些解析解，但是实际上只有非常少的情况求出方程/矩阵的解析解/解析特征值。

计算哈密顿算符在六角格子的本征值

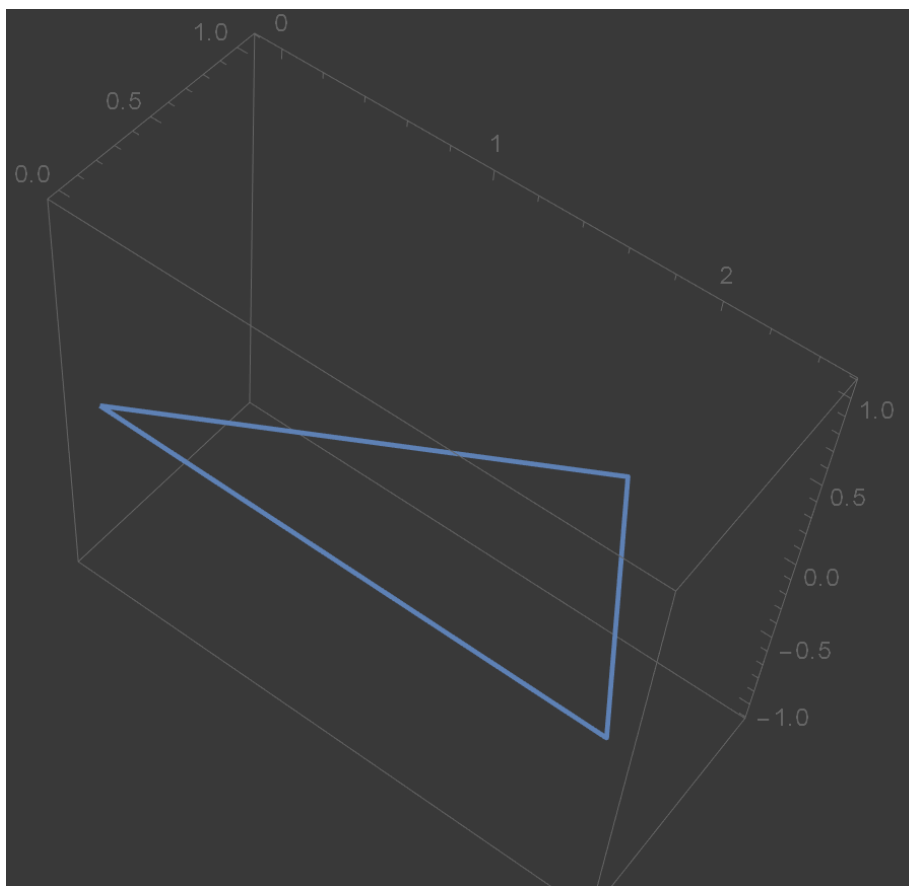
这部分实际上是我们工作初期要复现一篇论文（[Twisted bilayer graphene-electronic and optical properties](#)）的图。我们需要编写出程序来计算以下左图格子下的能量本征值，得到右图。事实上这还不够。当时我们还需要得到一个路径上的图像（而不是整个平面上），相当于我们要得到一个二维能带。



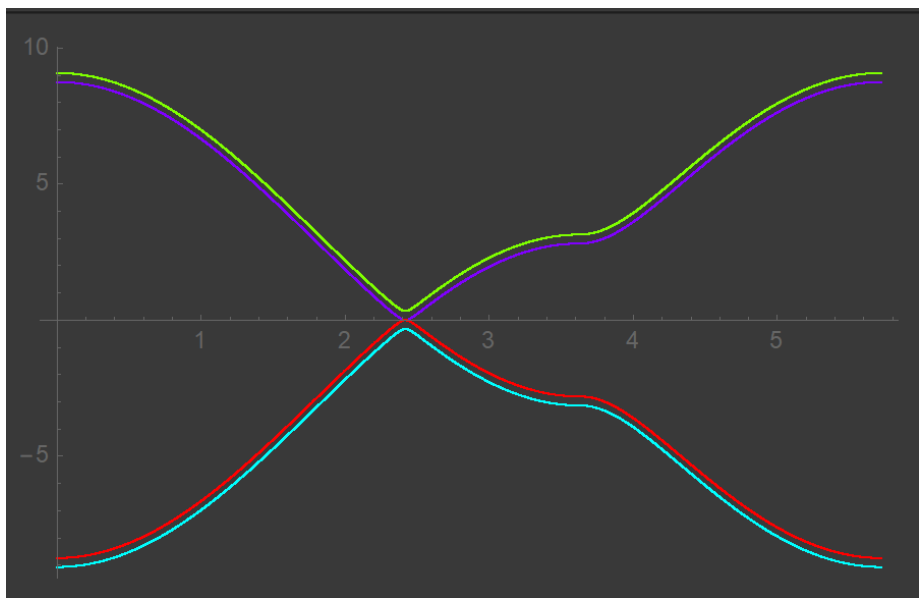
哈密顿矩阵形式如下：

$$\begin{pmatrix} 0 & -2.97 \left(e^{i \left(-\frac{\sqrt{3}}{2} \frac{y}{T} \right)} + e^{i \left(\frac{\sqrt{3}}{2} \frac{y}{T} \right)} + e^{iky} \right) & 0 & -0.33 \\ -2.97 \left(e^{-i \left(-\frac{\sqrt{3}}{2} \frac{y}{T} \right)} + e^{-i \left(\frac{\sqrt{3}}{2} \frac{y}{T} \right)} + e^{-iky} \right) & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.97 \left(e^{i \left(-\frac{\sqrt{3}}{2} \frac{y}{T} \right)} + e^{i \left(\frac{\sqrt{3}}{2} \frac{y}{T} \right)} + e^{iky} \right) \\ -0.33 & 0 & -2.97 \left(e^{-i \left(-\frac{\sqrt{3}}{2} \frac{y}{T} \right)} + e^{-i \left(\frac{\sqrt{3}}{2} \frac{y}{T} \right)} + e^{-iky} \right) & 0 \end{pmatrix}$$

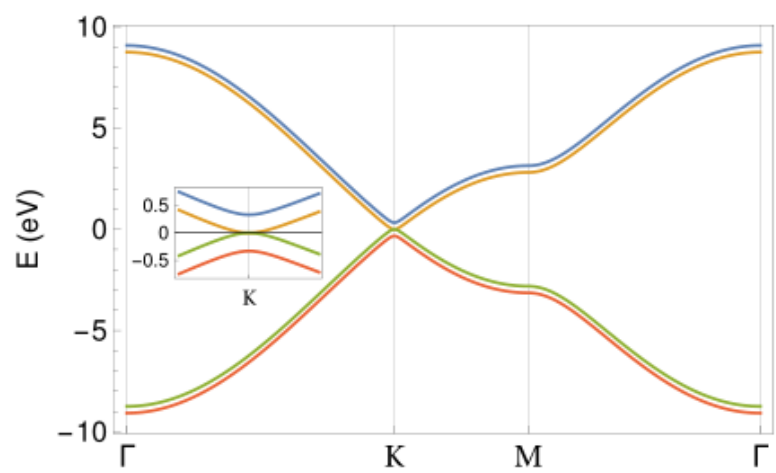
我们的路径大致是这个样子的：



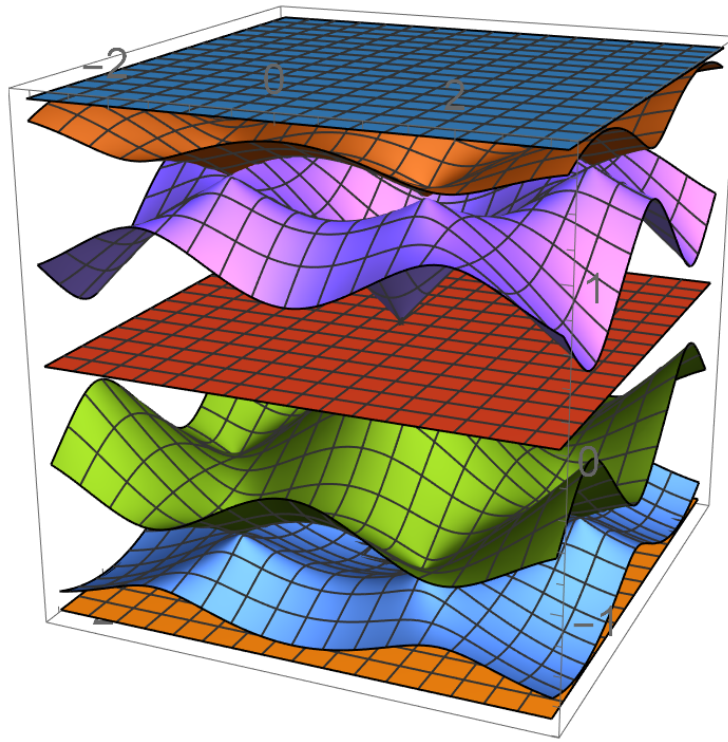
事实上计算本征值有相当完备的函数（无论是数值的还是解析的，当然这些软件函数并不一定可信，尤其是当它出现了令人意外的结果的时候：事实上我已经遇到过一些用内置函数算出来很难理解的结果的情况，而且这时候我无法得知其内部算法，这使得我愈加倾向于开源软件.....）。最困难的部分是编写可复用的能计算任意给定路径本征值的函数；经过一段时间的思考我做到了；并且复现了论文上的图片：



(以下是论文图)



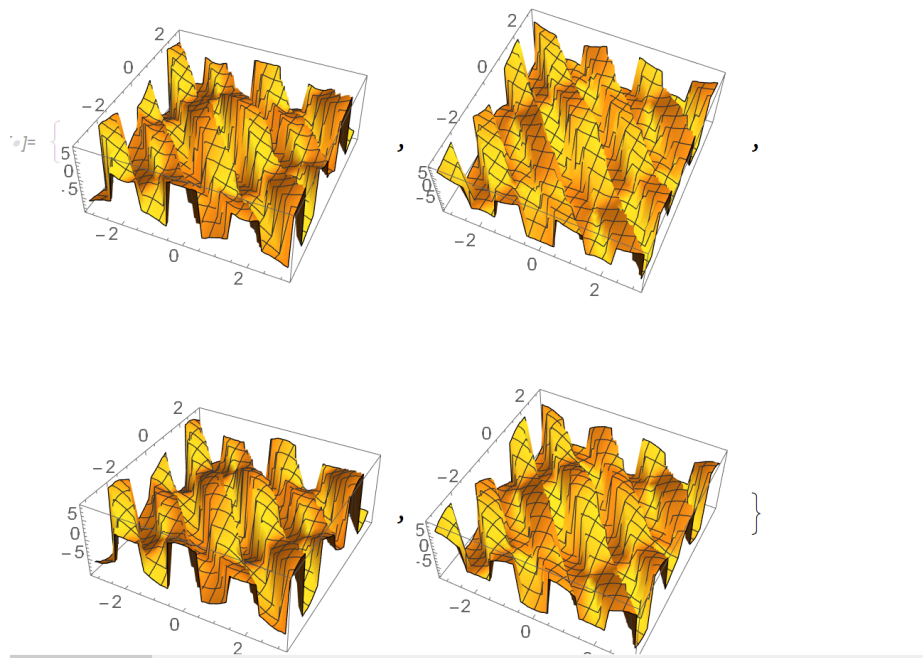
事实上，我们计算过不同阶数的矩阵的三维能带，下面是我们计算过的其中一个矩阵的三维能带图：



值得注意的是，这个矩阵是可以求出本征值解析式的；但是在研究更多的矩阵时，我们发现解析解法并非总是给出很好的结果。当我们使用**Mathematica**对矩阵强行求符号本征值的时候，它给出来一些异常巨大的式子，像下面这样：

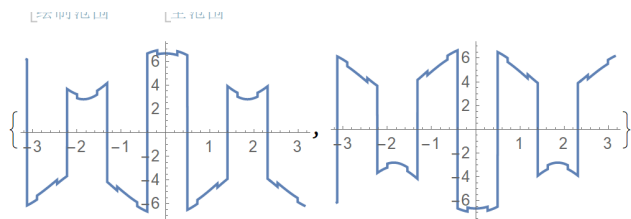
[illegible]

尽管看上去这些解的形式都是光滑可导的，但是实际上它们作出来的图是这样的：

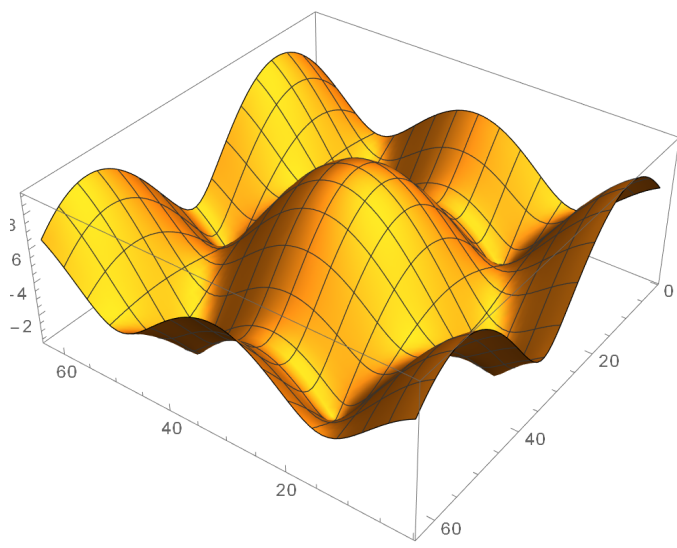


取一个截面看：

它们表现得像是跳跃不连续的函数：



然而，如果我们用数值解法来解算每一支能带时，得出来的图非常漂亮：



事实上，数值解和解析解都给出来了正确的本征值。我们探究过出现异常的原因。例如你要解算一个4阶方阵的本征值，那么它将会给出四个本征值；由于随着倒空间上 k_x k_y 的不同，这些本征值也不一样，因此就有四支能带。而Mathematica解析求解本征值的时候，必定是通过求解行列式方程来实现的；4阶的方程意味着它将会给出四支解析解。不幸的是，这四支解析解实际是交错的，也就是说每一支解析解上面，都交错地分布着四支能带的部分值，所以图像

看上去像是割裂的。然而令人疑惑的是软件确实给出了一个形式看上去光滑的解函数，所以“割裂”现象在此处是令人意外的。目前位置我们还在理解这一现象。

对kagome格子计算贝里曲率

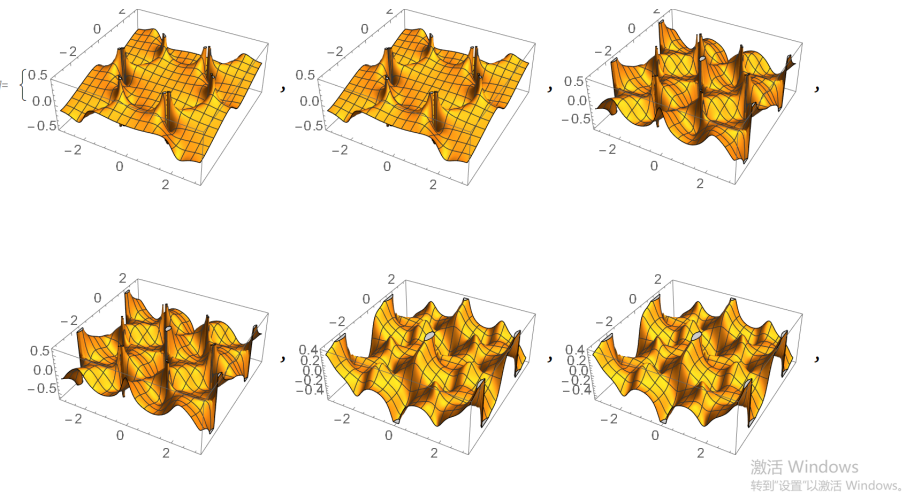
贝里曲率如下式：

$$\Omega_{nk}^{\mu\nu} = -2\sum_{n'\neq n}\frac{\text{Im}\langle n|\partial H/\partial k_\mu|n'\rangle\langle n'|\partial H/\partial k_\nu|n\rangle}{(\varepsilon_n - \varepsilon'_n)^2}, \quad (8)$$

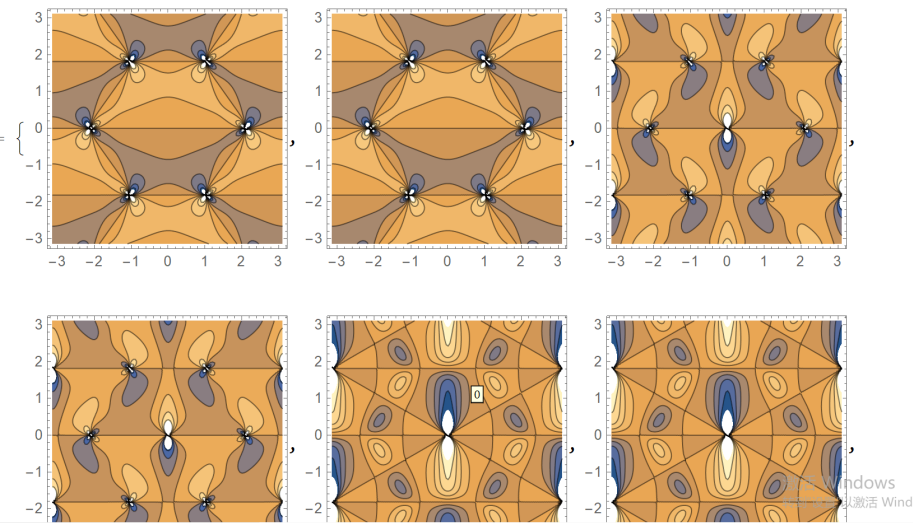
我们还计算过不同格子的贝里曲率。我们做计算的时候总是采用两种思路：

- 先符号求解，再对于整个倒空间进行计算
- 不符号计算，直接对倒空间取离散点阵进行数值计算

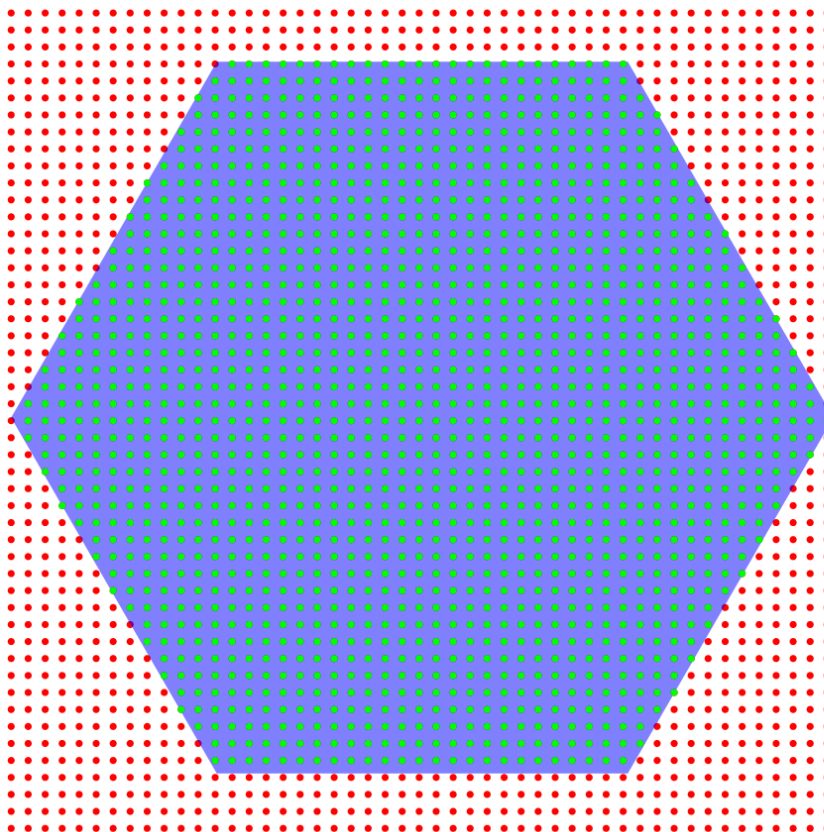
事实上在我们整个研究过程中数值计算几乎总是最有效的；依赖于Mathematica给出来的符号解则是庞杂而无法得出有用信息的。例如下面的图形就是我们数值计算kagome格子贝里曲率（哈密顿矩阵为12阶）的结果：



等高线图如下：



我们还计算过石墨烯的陈数：



一如既往地，我们离散化区域，然后取六角格子上的点计算某些量，然后求和，最终验证了文献（[Chern Numbers in Discretized Brillouin Zone: Efficient Method of Computing \(Spin\) Hall Conductances](#)）上的结论（+ - 1）。

```
In[149]= F[kx_, ky_] := Log[U1[kx, ky] * U2[kx + dk, ky] * U1[kx, ky + dk] ^ (-1) * U2[kx, ky] ^ (-1)];  
                                         [对数  
  
求和  
  
In[150]= Total[Map[F@@# &, set]] / (2 * Pi * I)  
          [总计] [映射] [虚数]  
Out[150]= -1. - 6.27276 × 10-15 i  
  
In[153]= N[%150, 30]  
          [数值运算]  
Out[153]= -1. - 6.27276 × 10-15 i
```

当然这部分在这里还无法展开细说（其实是我还没理清楚），但是我将把代码放在开源库上，并且写一篇相对完整的说明。

更多

我们合作的过程中遇到过不少问题，也解决其中的很多，当然还有一些正在想办法。我认为将这些代码以及思路放在开源平台上面是个好的选择；我看论文的时候如果有计算部分的话通常那篇论文不会给出代码.....使得复现验证有一定的阻碍；只有少数作者会将过程代码分享，我们会是其中之一。建立一个学术主页以及代码库是好事情。

我相当擅长**Mathematica**，用它做过一定的工作；有些是正经的研究需求，另一些则是在“玩”它的特性，但无论如何，我将写过的好的代码以及参考放在下面的开源库上：

[我的Mathematica代码库](#)

同时欢迎访问我的个人主页（虽然那上面基本啥也没有，但我空闲时将会丰富里面的内容）：

[我的个人主页](#)