

# Picturing Hofstadter's Butterfly<sup>\*</sup>

叶常青

(LSEC, 中国科学院, 数学与系统科学研究院, 计算数学与科学工程计算研究所, 北京 100190)

## 摘 要

本文从 Hofstadter's butterfly 出发引入对称三对角矩阵的特征值求解问题。我们介绍了针对这一问题的分而治之算法的框架, 给出了相应的并行化方案, 对并行带来的加速效果作了一定的分析。最后我们绘制了 Hofstadter's butterfly。本文所有的代码托管在Github上。

**关键词:** 分而治之算法 特征值计算 并行计算

## 1. 预备知识

“Hofstadter's butterfly”或直译为“侯世达的蝴蝶”来源于 D. Hofstadter 的著名文章<sup>1)</sup>。这篇文章研究电子在磁场中量子化行为, 并且将一些物理量可视化, 其图像出现了复杂的分形结构, 这些分形结构与张开翅膀的蝴蝶十分相似, 故由此而得名。这样一幅图像背后隐藏着深奥的物理和数学, 直接或间接与 1988 年 Nobel 奖和 2014 年 Fields 奖获得者的工作相联系。

数学上, “Hofstadter's butterfly”指一组定义在 Hilbert 空间  $l^2(\mathbb{Z})$  上的自共轭算子  $\{H_\alpha\}_{\alpha \in (0,1]}$

$$(H_\alpha u)_n = u_{n+1} + u_{n-1} + 2 \cos(2\pi\alpha n)u_n$$

其谱集  $\sigma(H_\alpha)$  在  $\mathbb{R}$  上的分布情况。显然我们可以证明  $\|H_\alpha\| \leq 4$ , 于是  $\sigma(H_\alpha) \in [-4, 4]$ 。令人惊讶的是如下结果:

**命题 1.** 当  $\alpha$  为无理数时,  $\sigma(H_\alpha)$  为康托集, 即无处稠密的完备集; 当  $\alpha = p/q$  为有理数时,  $\sigma(H_\alpha)$  由  $q$  个闭区间组成, 这些区间的端点可能重合。

对于无穷维的 Hilbert 空间  $l^2(\mathbb{Z})$ , 一般而言其上算子的谱是无法进行数值计算的。这里需要注意这里的谱不一定是特征值(或称点谱), 特别的当算子是紧算子的时候, 谱就一定是特征值, 但在这里  $H_\alpha$  显然是非紧的。我们的计算方法是将原来的  $H_\alpha$  对应的

<sup>\*</sup> 收到。

<sup>1)</sup> Hofstadter D R. Energy levels and wave functions of Bloch electrons in rational and irrational magnetic fields[J]. Physical review B, 1976, 14(6): 2239.

无穷维矩阵进行中心截断, 得到  $2k+1$  阶的对称三对角矩阵  $H_{\alpha,k}$ :

$$-k \begin{pmatrix} -k & \cdots & 0 & \cdots & k \\ 2\cos(2\pi\alpha k) & 1 & & & \\ \vdots & \cdots & \ddots & \ddots & \\ 0 & & 1 & 2 & 1 \\ \vdots & & & \ddots & \ddots & \cdots \\ k & & & & 1 & 2\cos(2\pi\alpha k) \end{pmatrix}$$

然后计算  $H_{\alpha,k}$  的特征值  $\lambda(H_{\alpha,k})$  来代替  $\sigma(H_\alpha)$ 。然而我们必须承认没有理论基础支撑这样的计算结果, 这样的中心截断只是为了能够计算。

## 2. 分而治之算法

在这一部分我们将介绍一种使用分而治之的思想求解对称三对角矩阵的特征值的算法, 这里我们只介绍这一算法的框架, 详细的推导见<sup>1)</sup>。

我们记  $n$  阶三对角对称矩阵  $T = [\beta_{i-1}, \alpha_i, \beta_i]$  为:

$$T = \begin{bmatrix} \alpha_0 & \beta_0 & & & \\ \beta_0 & \alpha_1 & \beta_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-3} & \alpha_{n-2} & \beta_{n-2} \\ & & & \beta_{n-2} & \alpha_{n-1} \end{bmatrix}$$

这里我们假定  $\beta_i > 0$ , 于是可以证明

**命题 2.** 当  $\beta_i > 0$ , 三对角矩阵  $T$  的特征值都是简单的, 即对应的特征多项式没有重根。

我们求解特征值的方法实际上是求解特征多项式的根, 记  $T$  的特征多项式为  $p_T(\lambda) = \det(T - \lambda I)$ 。对于三对角矩阵, 我们可以利用 Sturm sequence 使得以  $O(n)$  的计算代价得到  $P_T(x), P'_T(x)$  和  $P''_T(x)$ 。具体如下:

$$\begin{cases} \rho_0 = 1 & \rho_1 = \alpha_0 - x \\ \rho_i = (\alpha_i - x)\rho_{i-1} - \beta_{i-1}^2\rho_{i-2} & i = 2, 3, \dots, n-1 \end{cases}$$

最终有  $p_T(x) = \rho_{n-1}$ 。对于  $p'_T(x)$  和  $p''_T(x)$  也能推导出相应的递推式, 并且这个递推式也可以告诉我们小于  $x$  的特征值的数目。一旦我们可以得到  $p_T(x)$  的二阶导数信息, 便可以使用高阶的多项式求根算法, 这里我们使用的是 Laguerre 迭代法:

$$L_{\pm}(x) = x + \frac{n}{\left(-\frac{f'(x)}{f(x)}\right) \pm \sqrt{(n-1) \left[(n-1) \left(-\frac{f'(x)}{f(x)}\right) - n \frac{f''(x)}{f(x)}\right]}}$$

<sup>1)</sup> Li T Y, Zeng Z. The Laguerre iteration in solving the symmetric tridiagonal eigenproblem, revisited[J]. SIAM Journal on Scientific Computing, 1994, 15(5): 1145-1173.

注意到这里实际上只需要得到  $-p'_T(x)/p_T(x)$  和  $p''_T(x)/p_T(x)$ , 原来的迭代法可以进一步改造, 详细公式可见原文献。Laguerre 迭代法有如下性质:

**命题 3.** 若初始点  $x$  在特征值  $\lambda_i$  和  $\lambda_{i+1}$  之间, 即  $x \in (\lambda_i, \lambda_{i+1})$ , 于是有:

$$\lambda_i < L_-^{(n)}(x) < \cdots < L_-(x) < x < L_+(x) < \cdots < L_+^{(n)}(x) < \lambda_{i+1}$$

并且 Laguerre 迭代法在初始点选择良好的情况下, 可以以三次速率收敛到多项式的根。于是剩下的问题便在于如何选择合适的初始点, 分而治之的思想也在此体现。对于  $T = [\beta_{i-1}, \alpha_i, \beta_i]$ , 可以分裂矩阵:

$$\hat{T} = \begin{bmatrix} T_0 & \mathbf{0} \\ \mathbf{0} & T_1 \end{bmatrix}$$

其中

$$T_0 = \begin{bmatrix} \alpha_0 & \beta_0 & & & \\ \beta_0 & \alpha_1 & \beta_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-3} & \alpha_{k-2} & \beta_{k-2} \\ & & & \beta_{k-2} & \alpha_{k-1} \end{bmatrix} \quad T_1 = \begin{bmatrix} \alpha_k & \beta_k & & & \\ \beta_k & \alpha_{k+1} & \beta_{k+1} & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-3} & \alpha_{n-2} & \beta_{n-2} \\ & & & \beta_{n-2} & \alpha_{n-1} \end{bmatrix}$$

我们有如下命题:

**命题 4.** 记  $\lambda_0 < \lambda_1 < \cdots < \lambda_{n-1}$  为对称三对角矩阵  $T$  的特征值;  $\hat{\lambda}_0 \leq \hat{\lambda}_1 \leq \cdots \leq \hat{\lambda}_{n-1}$  为分块对称三对角矩阵  $\hat{T}$  的特征值, 注意这里无法保证  $\hat{T}$  的特征值都是简单的。于是有:

- $\hat{\lambda}_1 \in (\lambda_1, \lambda_2)$ ,  $\hat{\lambda}_{n-1} \in (\lambda_{n-2}, \lambda_{n-1})$ ,  $\hat{\lambda}_i \in (\lambda_{i-1}, \lambda_{i+1})$ ,  $1 \leq i \leq n-2$ 。
- $\lambda_0 \in [\hat{\lambda}_1 - |\beta_k|, \hat{\lambda}_1)$ ,  $\lambda_{n-1} \in (\hat{\lambda}_{n-1}, \hat{\lambda}_{n-1} + \beta_k]$ ,  $\lambda_i \in (\hat{\lambda}_{i-1}, \hat{\lambda}_{i+1})$ ,  $1 \leq i \leq n-2$ 。

当得到  $T_0$  和  $T_1$  的特征值后, 使用上述定理, 我们可以准确的预测  $T$  的特征值所在区间。通过二分法得到合适的初始值  $x$  后, 再使用高收敛速度的 Laguerre 迭代法得到特征值, 通常只需迭代 3 到 4 次即可收敛。整个算法实际上表现为二叉树结构, 当  $T$  分裂至的阶数为 1 或 2 时, 我们有特征值的显式的表达式, 或者设定当  $T$  分裂至的阶数小于特定值时, 直接使用 LAPACK 的 QR 算法求解特征值。

### 3. 并行性能测试

这里我们测试平台信息如下: CPU 为 intel i7-6700, 内存为 16GB, 系统为 Window Subsystem Linux (ubuntu 18.04), 编译器为 icc, 并开启 -O3 优化, 使用 Math Kernel Library 中的 QR 算法。

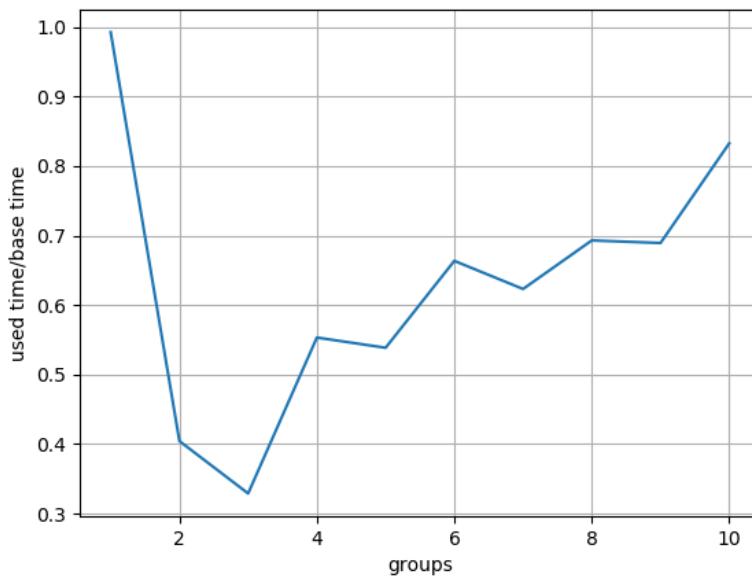
由于我们的分而治之算法相对的结构复杂, 并且在优化上不及 Math Kernel Library (MKL) 中的 QR 算法, 使得当矩阵的规模在 1000 到 10000 时, 我们计算耗时大约为

MKL 的 QR 算法的 10 倍，虽然这个结果已经比 Numpy 中的 QR 算法要快 2 倍左右。使用 MKL LAPACK 测试的代码见 `lapacke_test.c`，python numpy 测试代码见 `test.py`。测试对象为对称三对角矩阵  $H_{k,\alpha}$ ， $k$  取为 4096， $\alpha$  取 0.34。记录下关键求解步骤的墙上时间耗时，这里所有实验均测试 10 次后取平均值。

我们使用 openMP 进行并行化，为了充分利用 4 核处理器的架构，我们使用 8 个 openMP 线程进行测试。在我们的算法中可以有两种并行方案，一种是并行求解第 0 到第  $2k$  个特征值，这点在 `TriMatEigen.c` 中函数 `get_eigenvalues` 中体现；另一种是根据分而治之算法结构特点，先分组求解再汇总求解，相应的代码见 `TriMatEigen_groups.c`。我们简记不使用 openMP 的方法为 a，使用第一种并行方案为 b，使用第二种并行方案结果为 c。以下是测试结果：

测试方案	耗时 (s)
a	5.44
b	3.60
c groups=64	4.03

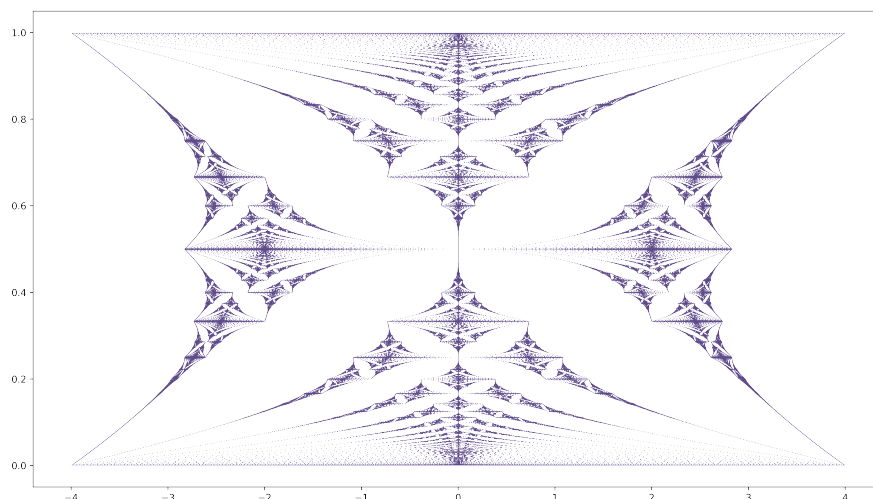
这里可以看出方案 b 取得了约 0.34 倍的加速，而方案 c 取得了 0.26 倍的加速。我们的加速实际上指减少的耗时占基准耗时的比例，基准耗时始终取方案 a 的耗时。注意到此时分组求解的加速效果并不明显，甚至不及简单的并行求特征值效果。仔细分析可以发现，分组过细使得并行带来的收益减少，因为计算耗时始终集中在分而治之算法的根部。于是尝试使用小分组的方案进行测试，结果绘制如下图：



可以看出当  $groups = 3$  时，加速效果最好，此时取得了约 0.67 倍的加速，也与使用 3 个分组相符。

#### 4. 结果展示

前文已经说明，我们的算法在实现上的效果不及 MKL LAPACK 中的 QR 算法，尽管并行可以取得一定的加速。在结果展示部分我们实际上使用 QR 算法进行计算，计算对象为一系列对称三对角矩阵  $H_{\alpha,k}$ ，其中  $k = 1024$ ， $\alpha$  取  $1/1025 : 1024/1025 : 1/1025$ 。源代码见 PicHofButterfly.c，下面是最终我们绘制的 Hofstadter's butterfly。



从上图中可以观察到不断出现的蝴蝶翅膀的结构，这与Wikipedia上的样图也相符。

#### 5. 总结

对称三对角矩阵的特征值求解是一个应用广泛的问题，对于这个问题可以使用一般的 QR 算法，算法复杂度大约为  $O(n^3)$ ，也可以使用利用三对角矩阵的特殊性质的分而治之算法。QR 算法已经有成熟的软件包可以调用，性能调校也非常优越，但是其并行化难度较高。而分而治之算法有着天然的并行结构，并行化潜力巨大。