# 汤姆逊问题

段元兴

2020 年 6 月 9 日

# Content

# 1 $V_{min}$

## 1.1 计算结果

以下是计算结果和与其他实现 (由于不同来源不一样, 所以$N$=2-50采用论文 [1]的结果, 而$N$=51-64采用网站 [2]的结果) 的差值. 考虑到论文实现的精度只有小数点后9位, 而自己计算的结果多次计算结果都表明在$N$较小的时候能达到$10^{-12}$, 在$N$较大的时候能达到$10^{-11}$, 且所有和参考值差的绝对值都在$5 \times 10^{-10}$以内, 所以基本可以认定精度达到了小数点后10位. 由于初始化为球面均匀随机分布, 所以很容易发生收敛到局域极小值的情况, 这时候根据参考计算结果判断是否需要重启. 所以总计算时间会由于重启的次数有很大不同, 而计算完$N$=2-64的总耗时在80s左右.

表 1: $V_{min}$计算结果(CPU: Ryzen9 3950x, 单线程)(误差为计算结果减去参考结果)

| $N$ | $V_{min}$ | $\Delta/10^{-10}$ | $N$ | $V_{min}$ | $\Delta/10^{-10}$ | $N$ | $V_{min}$ | $\Delta/10^{-10}$ |
|---|---|---|---|---|---|---|---|---|
| 2 | 0.50000000000 | 1.110e-06 | 23 | 203.93019066288 | -1.215 | 44 | 807.17426308463 | -3.720 |
| 3 | 1.73205080757 | -4.311 | 24 | 223.34707405181 | -1.949 | 45 | 846.18840106108 | 0.7788 |
| 4 | 3.67423461417 | 1.748 | 25 | 243.81276029877 | -2.342 | 46 | 886.16711363919 | 1.912 |
| 5 | 6.47469149469 | -3.118 | 26 | 265.13332631736 | 3.565 | 47 | 927.05927067971 | -2.904 |
| 6 | 9.98528137424 | 2.386 | 27 | 287.30261503304 | 0.3917 | 48 | 968.71345534379 | -2.125 |
| 7 | 14.45297741422 | 2.213 | 28 | 310.49154235820 | 2.019 | 49 | 1011.55718265357 | -4.284 |
| 8 | 19.67528786123 | 2.328 | 29 | 334.63443992042 | 4.156 | 50 | 1055.18231472630 | 2.963 |
| 9 | 25.75998653127 | 2.698 | 30 | 359.60394590376 | -2.365 | 51 | 1099.81929031890 | -1.019 |
| 10 | 32.71694946015 | 1.476 | 31 | 385.53083806330 | 2.995 | 52 | 1145.41896431928 | 2.790 |
| 11 | 40.59645050819 | 1.906 | 32 | 412.26127465053 | -4.707 | 53 | 1191.92229041622 | 2.242 |
| 12 | 49.16525305763 | -3.712 | 33 | 440.20405744765 | -3.527 | 54 | 1239.36147472916 | 1.589 |
| 13 | 58.85323061170 | -2.976 | 34 | 468.90485328134 | 3.433 | 55 | 1287.77272078271 | -2.913 |
| 14 | 69.30636329663 | -3.736 | 35 | 498.56987249065 | -3.547 | 56 | 1337.09494527566 | -3.429 |
| 15 | 80.67024411429 | 2.939 | 36 | 529.12240837541 | 4.137 | 57 | 1387.38322925284 | -1.585 |
| 16 | 92.91165530254 | -4.550 | 37 | 560.61888773104 | 0.4366 | 58 | 1438.61825064040 | 4.013 |
| 17 | 106.05040482862 | -3.813 | 38 | 593.03850356645 | 4.512 | 59 | 1490.77333527870 | -3.029 |
| 18 | 120.08446744749 | 4.923 | 39 | 626.38900901682 | -1.771 | 60 | 1543.83040097638 | 3.786 |
| 19 | 135.08946755668 | -3.207 | 40 | 660.67527883462 | -3.777 | 61 | 1597.94183019899 | -0.1091 |
| 20 | 150.88156833376 | -2.435 | 41 | 695.91674434189 | -1.129 | 62 | 1652.90940989830 | 3.013 |
| 21 | 167.64162239927 | 2.705 | 42 | 732.07810754367 | -3.266 | 63 | 1708.87968150325 | 2.494 |
| 22 | 185.28753614931 | 3.076 | 43 | 769.19084645916 | 1.584 | 64 | 1765.80257792730 | 3.035 |

## 1.2 实现方法

1. 求梯度的方法是手动计算函数值而非数值差分. 原因是在计算过程中发现当搜索步长降低到一定程度的时候, 发现使用差分求导会有不稳定和不收敛的情况.

2. 求极值的方法是梯度使用改良的Davidon三次插值线搜索的共轭梯度法. 改进之处为: 在求得得根与区间两端非常接近的时候利用二倍步长法重新确定一个相对更小的区间而不是仍然利用Davidon法给出的区间. 经过检验, 这种策略极大的加快了收敛速度, 而且原生方法有很多时候是不收敛的, 改进后基本上能做到保证收敛. 求新的搜索方向的$\beta$采用的是论文 [3]的结果.

3. 为了加速收敛, 采取了根据梯度的模动态调整初始搜索步长的策略, 而为了避免收敛至局域极值, 当与参考极小值误差较大而梯度的模已经非常小的时候重启程序. 对于重设搜索方向为梯度方向的条件, 有 $d^T g > 0$ (即沿当前方向的导数为正)和搜索次数已经达到$n$次.

# 2 简正频率和简并度

## 2.1 计算结果

对于$N = 12$, 验证为正二十面体的方法是求出最短棱的长度方差, 并验证是30条最短棱. 计算结果如下: 最短棱条数为30, 方差为$3.7 \times 10^{-15}$, 故可以认定为正二十面体. 而简正频率, 相应的简并度和计算出的特征值标准差如下表:

表 2: 简正频率, 简并度和标准差(CPU: Ryzen9 3950x, 单线程, 耗时20ms)

| 简正频率 | 简并度 | $\sigma(\omega^2)/10^{-14}$ |
|---|---|---|
| $< 5 \times 10^{-8}$ | 3 | 0.4 |
| 0.95245151590457 | 5 | 2.0 |
| 1.4664913060554 | 4 | 5.3 |
| 2.2716553285895 | 3 | 0.8 |
| 2.5701967498198 | 4 | 4.9 |
| 2.6396337463619 | 5 | 3.1 |

## 2.2 实现方法

1. 首先提高求平衡位置的精度, 原来是$||\nabla U||_2^2 < 10^{-20}$, 现调整为$10^{-26}$ (不然对求 Hessian 矩阵有影响), 而求Hessian矩阵采用的仍然是求出解析式然后求值.

2. 对于广义本征值问题

$$Ux - \omega^2 Ax = 0 \tag{1}$$

使用Cholesky分解:$A = L^T L$, 得到:

$$L^{-1}UL^{-1^T}x - \omega^2 x = 0. \tag{2}$$

由于$A$是对角阵, 所以可以很方便的得到

$$L^{-1}{}^T_{ij} = L^{-1}_{ij} = \frac{\delta_{ij}}{\sqrt{A_{ii}}}. \tag{3}$$

这样问题就变成了求对称矩阵的本征值问题, 可以先Householder三对角化再使用带位移的隐式对称QR算法.

# 参考文献


[1] Halima Lakhbab, Souad El Bernoussi, and Abderrahmane El Harif. Energy minimization of point charges on a sphere with a hybrid approach. *American Journal of Clinical Nutrition*, 95(4):916–924, 2014.

[2] Thomson problem. `https://en.wikipedia.org/wiki/Thomson_problem` Accessed May 30, 2020.

[3] William W. Hager and Hongchao Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *Siam Journal on Optimization*, 16(1):170–192, 2005.