Barnes–Wall 格的沃罗诺伊区域 Λ_{16}

Daniel Pook-Kolb、Erik Agrell，*IEEE* 会士，以及 Bruce Allen，*IEEE* 会员

arXiv:2301.06092v1 [cs.IT] 15 Jan 2023

摘要—本文详细描述了 Barnes–Wall 格 Λ_{16} 的沃罗诺伊区域，包括其顶点、相关向量和对称群。我们计算了其量化器常数的精确值，而此前该值仅为近似值。为验证这一结果，我们对同一常数进行数值估计，并提出一种非常简单的新方法来量化此类估计的方差，其准确度远高于常用的刀切估计量。

索引词—**Barnes–Wall 格、晶格量化器、归一化二阶矩、量化器常数、沃罗诺伊区域**

I. 引言

IN 1959，E. S. Barnes 和 G. E. Wall 基于阿贝尔群[1], 提出了一类维数为 4、8、16、.....的格。在维数为 4 和 8 时，这一构造重现了已知的格，如今分别记为 D_4 和E_8; 而在维数为 16 及更高时，则得到了此前未知的格。 [2]–[4]研究了 *Barnes–Wall*（*BW*）格的其他构造方式及其更多性质。

BW 格在格的三项标准优劣指标上表现出色：球堆积、接吻数和量化。事实上，在维数为 $n = 4$、8 和 16 [5, Ch. 1]中，已知或推测它们在这三项指标上均为最优。因此，它们已应用于多种领域，包括数字通信 [2],数据压缩 [6], 密码学 [7], 量子计算[8], 和代数几何 [9]。

D_4 和 E_8 的 Voronoi 区域已得到完整确定。因此，它们的球堆积密度、接吻数，以及量化器常数均已精确获知 [10], [5, 第 4],章，我们将不再进一步讨论这些格。本文确定了16维 BW 格Λ_{16}的沃罗诺伊区域。其相关向量、顶点和量化器常数均为首次精确报告。此外，我们还刻画了它的完整对称群；已知该群的阶为89 181 388 800 [5, 第4节。10],，并使用两个变换矩阵。

II. 面层次结构

本节将自下而上地描述 Λ_{16} 的沃罗诺伊区域，从0维面（顶点）开始，沿着维数层次逐步向上，直至单个 16维面，即沃罗诺伊区域本身。我们在由下块三角生成矩阵定义的坐标系中描述这些面

$$\left[\begin{array}{cccccccc:cccccccc}
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{array}\right]. \quad (1)$$

与 [5,Fig. 4.10] 中同一格的生成矩阵相比，该生成矩阵按线性因子 $\sqrt{2}$ 缩小（或等价地，体积因子为 256）。一些格参数取决于格的缩放。

A. 0维面

沃罗诺伊区域有 201 343 200 个顶点，分属六个等价类，列于 $\boldsymbol{v}_1$，$\boldsymbol{v}_2$，.....，$\boldsymbol{v}_6$ 表 I 中。等价性由旋转 Aut(Λ_{16}) 定义，这些旋转将Λ_{16} 映射到 Λ_{16}。若将格向量平移视为另一种等价操作，则 $\boldsymbol{v}_2$ 与$\boldsymbol{v}_4$ 等价，且 $\boldsymbol{v}_3$ 与 $\boldsymbol{v}_5$等价，从而将六个等价类减为四个。顶点到原点的距离平方为 3/2、 10/9 或 1。因此，覆盖半径为 $\sqrt{3/2}$, 这在 [3], [5, 第 4 节10]中已有所知。

B. 1维面

这些顶点总共由约 $3 \cdot 10^{10}$条棱相连，分属 23 个等价类。[1] 它们的长度分别为 $\sqrt{3/2}$、1、 $\sqrt{17/18}$、$\sqrt{7/3}$、$\sqrt{11/18}$、 2/3、 $\sqrt{5/18}$ 和1/3。在分别等价于 $\boldsymbol{v}_1$、 $\boldsymbol{v}_2$、$\boldsymbol{v}_3$、 $\boldsymbol{v}_4$、 $\boldsymbol{v}_5$ 或 $\boldsymbol{v}_6$ 的各顶点处，相交的棱数分别为 32 768、144、403、179、220 或 398824。

[1]我们明确构造了每个等价类至少一个代表元。然而，我们没有生成这些代表元的完整轨道。面数则通过使用 [11]估算轨道大小来获得。

本文已提交 IEEE，供其考虑发表。版权可能在未通知的情况下转让，届时此版本可能不再可访问。

D. Pook-Kolb 和 B. Allen 任职于德国汉诺威 30167 的马克斯·普朗克引力物理研究所（阿尔伯特·爱因斯坦研究所）（电子邮箱：daniel.pook.kolb@aei.mpg.de 和 bruce.allen@aei.mpg.de）。

E. Agrell 任职于瑞典哥德堡 41296 的查尔姆斯理工大学电气工程系（电子邮箱：agrell@chalmers.se）。

表 I: $\boldsymbol{n}_i$ 的相关向量代表元（前两行）和 $\boldsymbol{v}_i$ 的顶点（其余六行），按长度递增长度排列。所示内容包括 Λ_{16} 的分量、长度平方、Aut(Λ_{16})作用下的轨道大小，以及Aut(Λ_{16})相应稳定子子群的大小。

向量	分量	$\|\cdot\|^2$	轨道大小	稳定子大小
$\boldsymbol{n}_1$	$(1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0)$	2	4 320	20 643 840
$\boldsymbol{n}_2$	$1/2\,(2\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ -1\ \ 0)$	3	61 440	1 451 520
$\boldsymbol{v}_1$	$1/2\,(1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1)$	1	4 320	20 643 840
$\boldsymbol{v}_2$	$1/12\,(9\ \ 3\ \ 3\ \ 3\ \ 1\ \ -1\ \ 1\ \ 1\ \ 1\ \ -1\ \ 3\ \ -1\ \ -3\ \ 3\ \ -3\ \ -3)$	$10/9$	66 355 200	1 344
$\boldsymbol{v}_3$	$1/6\,(5\ \ 1\ \ -1\ \ -1\ \ -1\ \ 1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ 1\ \ 1\ \ 1)$	$10/9$	2 211 840	40 320
$\boldsymbol{v}_4$	$1/6\,(5\ \ 1\ \ 1\ \ 1\ \ -1\ \ 1\ \ -1\ \ -1\ \ 1\ \ -1\ \ -1\ \ -1\ \ -1\ \ -1\ \ 1\ \ -1)$	$10/9$	66 355 200	1 344
$\boldsymbol{v}_5$	$1/6\,(5\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ -1\ \ -1\ \ 1\ \ 1\ \ 1\ \ -1\ \ 1\ \ -1\ \ -1\ \ 1)$	$10/9$	66 355 200	1 344
$\boldsymbol{v}_6$	$1/4\,(3\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ -1\ \ 1\ \ 1\ \ 1\ \ -1\ \ 1\ \ -1\ \ -1\ \ -1)$	$3/2$	61 440	1 451 520

C. 二维面

在 58 个等价类中，约有 $5\cdot10^{11}$ 2 个二维面。其中包括 3 类面积约为 $1/9$ 的正方形，共约 $4\cdot10^{9}$ 个；以及分属 55 类的约 $5\cdot10^{11}$ 个三角形，其中 21 个在几何上彼此不同，面积介于 $\sqrt{15}/72$ 和 $\sqrt{5}/4$ 之间。图1展示了 22 个几何上彼此不同的二维面。

D. 三维至十四维面

在维数为 3 至 14 的情况下，共有 6 052 个面类，这里不再详细描述。表II 总结了其中的一些性质，其中列出了 Aut(Λ_{16}) 下的面类数量、子面数量（即维数为 $d-1$ 的子面）以及所有维数为 $d=0$，1，.....，16 的面的顶点数。更多信息请参见补充材料 [12]。

E. 十五维面

这类十五维面，或称面片，都位于原点与另一个格向量之间的中点处，并且垂直于两者之间的连线。总共有 65 760 个这样的定义刻面的非零向量，或称相关向量。它们属于两个与原点距离不同的等价类（见表 I）。离原点最近的是最短向量，其长度平方为 2，这些向量已在 [1] 中找到。填充半径是它们长度的一半，即 $\sqrt{2}/2$。此类向量共有 4 320 个，这就是该格的接吻数。此外还有 61 440 个其他相关向量，其长度平方为 3。

与 4 320 个最短向量对应的面片各有 7 704 个子面，以及 1 046430 个属于全部六类的顶点；其余 61 440 个面片各有 828 个子面和 26 160 个顶点，这些顶点分别等价于 $\boldsymbol{v}_2$、 $\boldsymbol{v}_4$、 $\boldsymbol{v}_5$ 或 $\boldsymbol{v}_6$。

*F. 16*维面

枚举出 d =0、1、.....、15 的所有不等价 d维面，并利用 [13, Sec. 3],中的递归关系计算它们的体积和二阶矩后，就得到了对 16维面的完整刻画。利用 [11],，我们估计沃罗诺伊区域在所有维数中共有 $1\cdot10^{14}$ 至 $3\cdot10^{14}$ 个面。

接下来，计算协方差矩阵或二阶矩张量：

$$\boldsymbol{U}=\frac{U}{16}\boldsymbol{I}_{16}\,, \tag{2}$$

其中（未归一化的）二阶矩

$$U=\operatorname{tr}\boldsymbol{U}=\frac{207\,049\,815\,983}{4\,287\,303\,820\,800} \tag{3}$$

以及 $\boldsymbol{I}_{16}$ 单位矩阵。经过适当归一化后，可得量化器常数

$$G=\frac{1}{n}\frac{U}{V^{1+2/n}}\,, \tag{4}$$

其中 $n=16$ 是该格的维数， $V=1/16$ 是其沃罗诺伊区域的体积，由此得到

$$G=U\sqrt{2}\approx0.068\,297\,622\,489\,318\,7\,. \tag{5}$$

为验证我们对面类的枚举，我们使用 [13,第 3]节中的递归关系计算沃罗诺伊区域的体积，结果与预期值$1/16$一致。我们还在第IV节中通过数值方法验证结果 (5)。

III. Λ16 的对称群

Λ_{16} 的对称变换由符号改变、置换以及矩阵的乘积生成

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{H}_4&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\\ \boldsymbol{0}&\boldsymbol{H}_4&\boldsymbol{0}&\boldsymbol{0}\\ \boldsymbol{0}&\boldsymbol{0}&\boldsymbol{H}_4&\boldsymbol{0}\\ \boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{H}_4\end{bmatrix}\,, \tag{6}$$

其中

$$\boldsymbol{H}_4=\frac{1}{2}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix} \tag{7}$$

是一个Hadamard矩阵。

共有 2 048 种符号改变，可描述为三个子群 $\mathcal{S}_1$、 $\mathcal{S}_2$ 和 $\mathcal{S}_3$的乘积。第一个子群$\mathcal{S}_1$ 包含对分量对 $(\boldsymbol{x}_i,\boldsymbol{x}_{i+1})$ 中偶数个分量进行符号改变的所有情况，其中 $i=1,3,\ldots15$，阶为 128。 $\mathcal{S}_2$对前 4 个和后 4 个奇数分量（$\boldsymbol{x}_i,\boldsymbol{x}_{16-i}$）中的偶数个分量改变符号，其中 $i=1,3,5,7$。此子群的阶为 8。最后，$\mathcal{S}_3$ 的阶为 2，它改变分量（$\boldsymbol{x}_1,\boldsymbol{x}_3,\boldsymbol{x}_5,\boldsymbol{x}_7$）的符号。

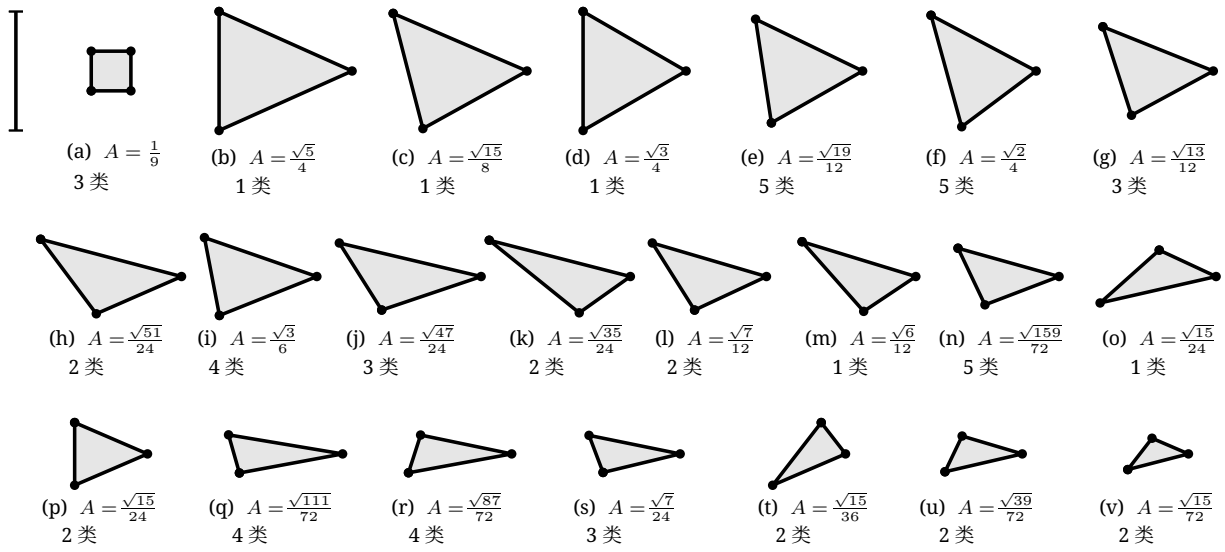


图1： Λ_{16} 的沃罗诺伊区域的 22 种几何上不同的二维面类型。我们展示其面积 A，以及这些面类型在 Aut(Λ_{16}) 作用下所属的类数。作为参考，左上角的线段长度为单位长度。(a) 是边长为 $1/3$ 的正方形。三角形 (d) 是等边三角形，而 (b)、(c)、(e)、(g)、(h)、(l)、(p)、(t) 和 (v) 是等腰三角形。所有三角形的内角都严格小于 $90°$。最大值 $\theta_{\max}$ 仅出现在 (n) 中，其中 $\cos\theta_{\max}=\sqrt{10}/40 (\theta_{\max}\approx 85.47°)$。最小角满足 $\cos\theta_{\min}=29\sqrt{238}/476$ ($\theta_{\min}\approx 19.97°$)，且仅出现在三角形 (q) 中。

保持 Λ_{16} 不变的向量分量置换 $\mathcal{P}\subset$ Aut(Λ_{16}) 在 [1, 引理 3.2]中有所描述。该引理利用伽罗瓦域 GF(2) 上的 4 维向量空间来表示格向量分量的索引。有关这一构造的详细说明，请参阅 [1] 。利用 [14, 第 13]章的公式 (19), $\mathcal{P}$ 的阶为

$$|\mathcal{P}| = 16\prod_{l=0}^{3}(16-2^l) = 322\,560\,, \tag{8}$$

这些正是使长度为 2^4 的首阶二元 Reed–Muller 码保持不变的置换 [14,第 13]章的定理24。

$\mathcal{P}$ 中的置换示例如下:

$$\begin{aligned}
p_1 &= (1\ 2\ 3\ 4)(5\ 6\ 7\ 8)(9\ 10\ 11\ 12)(13\ 14\ 15\ 16)\,,\\
p_2 &= (1\ 2)(5\ 6)(9\ 10)(13\ 14)\,,\\
p_3 &= (1\ 6\ 13)(2\ 8)(3\ 9\ 12\ 5\ 15\ 14)(4\ 11\ 7)\,,\\
p_4 &= (1\ 9\ 16\ 15\ 5\ 7\ 4\ 8\ 10\ 6\ 13\ 2\ 3\ 14\ 12)\,,
\end{aligned} \tag{9}$$

为简洁起见，这里用循环表示法表示。可以使用这些置换的不同子集生成完整的子群 $\mathcal{P}$，例如 $\{p_1,p_2,p_3\}$、$\{p_1,p_4\}$或 $\{p_3,p_4\}$。

完整的自同构群 Aut(Λ_{16}) 可由 $\boldsymbol{H}$ 与 $\mathcal{S}_1$、$\mathcal{S}_2$、$\mathcal{S}_3$ 的生成元以及 $\mathcal{P}$ 的一组生成元组合生成。值得注意的是，它也可以仅由两个矩阵生成。第一个是 16×16 与 p_3 对应的置换矩阵 $\boldsymbol{M}_1$。第二个是一个矩阵

$$\boldsymbol{M}_2 = \begin{bmatrix} \bar{\boldsymbol{H}}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{H}}_4 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \bar{\boldsymbol{H}}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \bar{\boldsymbol{H}}_4 \end{bmatrix}, \tag{10}$$

它由（7）构造而成，只是最后一行的符号改变了，也就是说，使用 Hadamard矩阵

$$\bar{\boldsymbol{H}}_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix}. \tag{11}$$

IV. 数值验证与误差估计

为验证（3），我们通过对沃罗诺伊区域进行蒙特卡罗积分来估计 U。我们还估计 U估计值的方差；为此，我们采用不同于 [15]中“刀切估计量”的方法。本节首先介绍 U 及其方差的估计方法，然后说明为何我们更倾向于使用方差估计量而非刀切法，最后将 Λ_{16} 的 G 数值估计值与（5）中的真实值进行比较。

U 的蒙特卡洛估计为

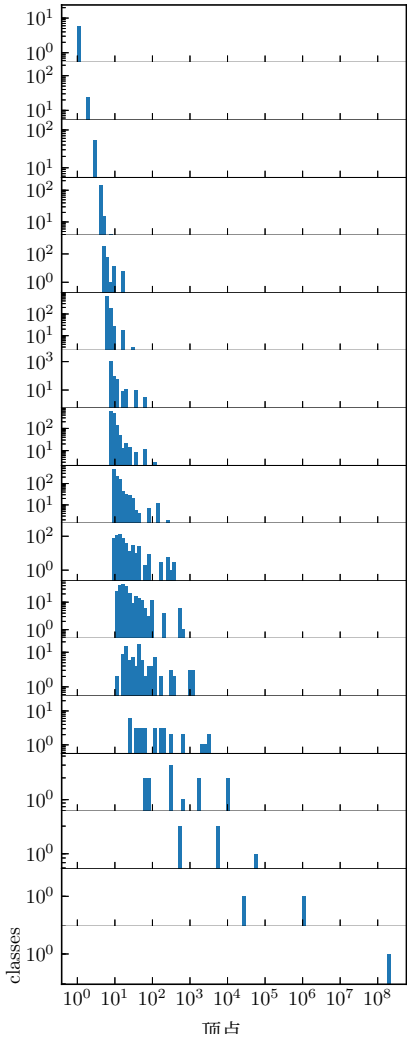
$$\hat{U} = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i\|^2\,, \tag{12}$$

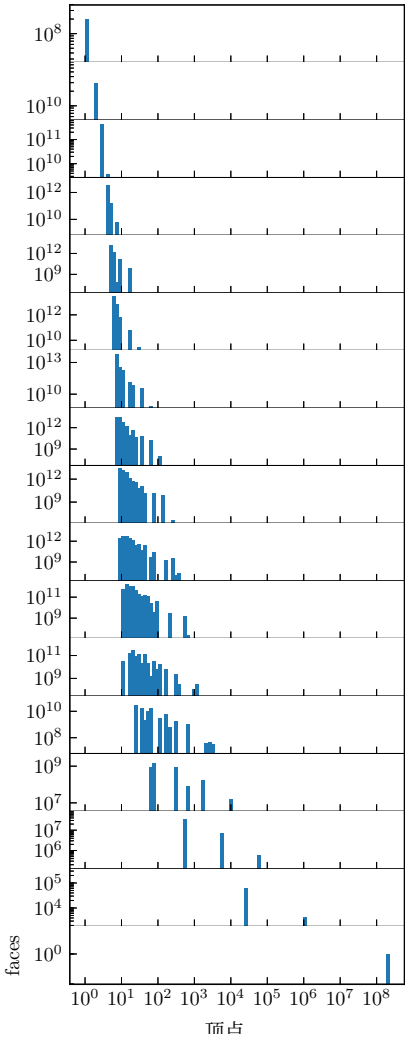
其中，$\boldsymbol{x}_1$、……、$\boldsymbol{x}_N$是 N在 Λ的沃罗诺伊区域内均匀分布的独立随机向量。

为了估计 var$\hat{U}$，我们首先注意到，由于向量$\boldsymbol{x}_i$ 相互独立且服从相同分布，var$U$ =

表 II： Λ_{16}的沃罗诺伊区域的面汇总信息。第一列列出面 d 的维数，第二列列出 Aut(Λ_{16}) 下 d-面类的数量。第三列显示每个 d-面的子面数量范围，第四列显示每个 d-面的顶点数量范围。第五列通过可视化展示了包含特定数量顶点（x-轴）的面类数量（y-轴）。最后一列显示相同的信息，但统计对象是面而非面类；这些数量使用 [11]进行了近似。

dim	类	子面	顶点	面类的顶点数	面的顶点数（约）
0	6	0	1		
1	23	2	2		
2	58	3, 4	3, 4		
3	168	4–6	4–8		
4	441	5–16	5–16		
5	867	6–21	6–32		
6	1 257	7–30	7–64		
7	1 329	8–51	8–128		
8	1 023	9–128	9–256		
9	566	10–194	10–400		
10	253	11–258	11–641		
11	96	12–620	12–1 281		
12	35	16–862	24–2 945		
13	12	42–1 312	64–11 138		
14	5	144–2 763	520–59 907		
15	2	828, 7 704	26 160 , 1 046 430		
16	1	65 760	201 343 200		





$(1/N)$ var$\|\boldsymbol{x}\|^2$，其中 $\boldsymbol{x}$ 是一个与 $\boldsymbol{x}_i$服从相同分布的单个随机向量。因此，我们对 var U的估计量记为 $\widehat{\text{var}}\,U$，定义如下：

$$\widehat{\text{var}}\,\hat{U} = (1/N)\,\widehat{\text{var}}\|\boldsymbol{x}\|^2\,. \tag{13}$$

应用 var$\|\boldsymbol{x}\|^2$ 的标准无偏方差估计量

$$\widehat{\text{var}}\|\boldsymbol{x}\|^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(\|\boldsymbol{x}_i\|^2 - \hat{U}\right)^2 \tag{14}$$

代入（13）得到

$$\begin{aligned}\widehat{\text{var}}\,\hat{U} &= \frac{1}{N(N-1)}\sum_{i=1}^{N}\left(\|\boldsymbol{x}_i\|^2 - \hat{U}\right)^2 \\ &= \frac{1}{N-1}\left(\frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i\|^4 - \hat{U}^2\right)\end{aligned} \tag{15}$$

或者像（4）中那样进行归一化后

$$\hat{G} = \frac{\hat{U}}{nV^{1+2/n}}, \tag{16}$$

$$\widehat{\text{var}}\,\hat{G} = \frac{\widehat{\text{var}}\,\hat{U}}{(nV^{1+2/n})^2}\,. \tag{17}$$

方差估计量 (15) 直接来自概率的基本定律。令人惊讶的是，在以往大多数（甚至可能是全部）涉及格点二阶矩数值估计的研究中，人们一直使用另一种估计量，却从未受到质疑 [15]–[17]。为了纠正这一持续了 39 年的误解，我们现在详细说明为何 (15) 更为准确。

刀切法通过将独立随机选取的向量 $\boldsymbol{x}_1$、……、 $\boldsymbol{x}_N$划分为 g组，计算每组内长度平方的平均值，最后计算这些 g 平均值 [15, 的样本方差，见式 (3)–

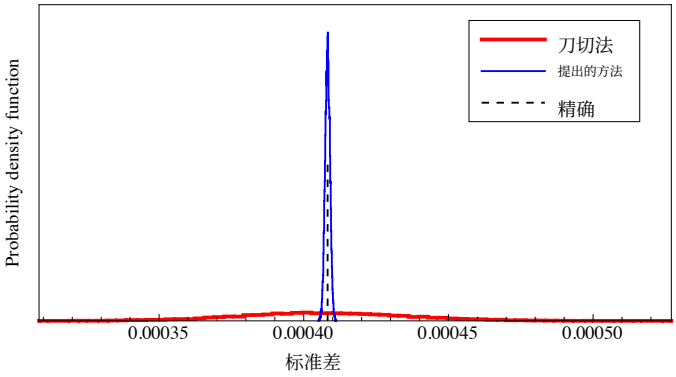


图2：立方格二阶矩 U 估计值标准差 ˆ 的两个估计量的直方图。立方格的精确标准差 (var $\hat{U}$)$^{1/2}$ 可通过解析方法计算，由此可见，所提出的估计量（12）比采用 100 组的刀切法准确得多。

(4)]。这种方法至少有两项缺点：首先，估计的方差取决于列表 $\boldsymbol{x}_1$、……、$\boldsymbol{x}_N$ 的排列顺序；重新排列列表会得到不同的方差估计，尽管估计的二阶矩 (12) 保持不变。其次，它忽略了组内向量的方差。所提出的估计量 (15) 则不存在这两项缺点。

为了量化两种方差估计量的准确性，我们对立方格 $\mathbb{Z}^n$ 在 $n = 3$ 下的二阶矩进行了数值估计。$\mathbb{Z}^n$ 的二阶矩为 $U = \mathbb{E}[\|\boldsymbol{x}\|^2] = n/12$，而 U 的方差可精确计算为 var $U = (1/N)\,\mathrm{var}\|\boldsymbol{x}\|^2 = (1/N)(\mathbb{E}[\|\boldsymbol{x}\|^4] - \mathbb{E}[\|\boldsymbol{x}\|^2]^2) = n/(180N)$。我们在 $\mathbb{Z}^3$ 的沃罗诺伊区域（即单位立方体）中均匀生成了 $N = 100\,000$ 个向量，使用式（12）计算了 U，并用两种方法估计了 U 的方差。对于刀切法，我们采用了与 [15] 中相同的群大小 $g = 100$ 。两种估计量各运行了 10000 次，每次使用 N 个新的随机向量。图 2 展示了所得标准差估计值的直方图，并给出了精确值。可以看出，在这个例子中，式（12）的准确度比使用 $g = 100$ 的刀切法高出一个数量级以上。

刀切法的准确性会随着 g 增大而提高；当每个群只包含一个样本时，即 $g = N$ 时，其准确性最高。在这一极端情况下，刀切法简化为 (15)——但以往的研究并非如此应用刀切法 [15]–[17]。

在确认新的方差估计量的实用性后，我们继续高精度地估计 Λ_{16} 的量化器常数 G。使用 $N = 4 \cdot 10^{12}$ 个随机的 16 维向量，对均值数值计算 (12) 和 (16)，对标准差数值计算 (15) 和 (17)，得到

$$\hat{G} = 0.068297616\,, \tag{18}$$

$$\sqrt{\widehat{\mathrm{var}}\,\hat{G}} = 0.000000009\,. \tag{19}$$

ˆ G 与式 (5) 中精确的 G 之间的差仅为 0.7 个标准差，这可作为面层次结构的数值验证。结果也与先前对同一常数较不精确的估计一致，见 [15, 式 (13)]。

V. 算法

我们的算法 [2] 在 [13]，中有详细描述，该方法基于先前用于寻找所有相关向量 [21] 和面 [22] 的方法。本节简要概述其核心概念，并介绍对方法 [13] 的少量修改。

基本方法保持不变：我们首先找到所有相关向量，即面片的法向量，以及沃罗诺伊区域的所有顶点。随后，通过递归地求交父面的顶点集合，构建面片的子面层级。为降低计算成本，我们先找出在 Aut(Λ_{16}) 作用下等价的面类，然后仅为每一类中任选的一个代表面构建其子面。总共只显式构建了 159143 个面。

面分类按迭代方式进行，如 [13, 第 2.4 节4]所述。在此方法中，我们首先使用适当的子群 $\mathcal{U} \subset$ Aut(Λ_{16}) 来识别等价的面，从而得到在 $\mathcal{U}$ 下的面类。随后，使用另一个子群 $\mathcal{U}'$ 对每个面类中任选一个代表元组成的集合进行分类。我们可以用不同的子群重复这一过程，直至最终使用全群 Aut(Λ_{16})。对于Λ_{16}，我们发现一个不错的方案是只使用单个子群 $\mathcal{U}$，将其选为相关向量$\boldsymbol{n}_2$的稳定子，其稳定子大小为 1 451520（见表 I）。

我们对 [13],中的方法作了三处修改，这些修改影响了两个面等价性的检验方式，以及如何构造单个向量的轨道和稳定子。接下来我们将依次介绍这些修改，简要回顾相应的旧方法，然后介绍我们的新算法。

A. 测试面等价性

我们先前用于检验面 F 在群 $\mathcal{G}$ 作用下是否等价于另一个面 F' 的方法基于以下思路。[3] 对于每个面，我们取一组能够唯一标识该面的向量。我们可以取与包含该面的面片相关联的相关向量集合（即该面的“法向量”），也可以取该面的顶点。具体选择取决于这两个集合中向量的数量，以及它们在 $\mathcal{G}$ 下的分类。令$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$为 F的向量，$\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$为$F'$的向量。我们对这些向量排序，使得对于所有 i，$\boldsymbol{x}_i$ 都等价于 $\boldsymbol{y}_i$（若无法做到，则这两个面不等价）。然后，我们构造所有 i对应的向量对（$\boldsymbol{x}_i, \boldsymbol{y}_i$）之间的变换所组成的集合。如果这些集合的交集非空，那么其中的变换会将 F 映射到 F'。然而，如果交集为空，我们就对其中一个集合进行置换，然后重试。面

[2]算法使用*Python* 实现；代码清单中使用的数据类型“List”和“Dictionary”旨在表现得与相应的 Python 类型相同。群论方面使用 *GAP* [18], [19], , 并通过 Python 调用*gappy* [20]。[3]此处，$\mathcal{G}$ 为 Aut(Λ_{16}) 或 $\mathcal{U}$。

当且仅当所有置换所得的变换集合的交集均为空时，这两个面才不等价。

原则上，任意两个等价向量之间的全部变换都可以按如下方式构造。设 $\boldsymbol{x} = g_x \boldsymbol{x}^{\text{rep}}$ 和 $\boldsymbol{y} = g_y \boldsymbol{x}^{\text{rep}}$ 是两个等价向量，g_x、$g_y \in \mathcal{G}$ 和 $\boldsymbol{x}^{\text{rep}}$ 表示它们的等价类。那么，将 $\boldsymbol{x}$ 映射到 $\boldsymbol{y}$ 的 $\mathcal{G}$ 中的全部变换为[13]

$$\mathcal{T}_{xy} = g_y \,\mathrm{Stab}_{\mathcal{G}}(\boldsymbol{x}^{\text{rep}}) g_x^{-1}\,, \tag{20}$$

其中，$\mathrm{Stab}_{\mathcal{G}}(\boldsymbol{x}^{\text{rep}})$ 是 $\mathcal{G}$中 $\boldsymbol{x}^{\text{rep}}$ 的稳定子。

由表 I 可知，对于 Λ_{16}，集合 (20) 的元素数量介于 1 344 和 20 643840 之间。使用 *GAP* 求交集时，这些集合会保存在内存中；当需要计算多个交集时，这便会成为问题。

现在我们介绍一种内存高效的替代方案，如算法 1 所示。与 [13], 中一样，该方法在确认F 和 F' 具有相同数量的顶点和法向量，并且相应的向量集合可以按某种顺序排列，使得对于所有 i，$\boldsymbol{x}_i \sim \boldsymbol{y}_i$ 之后使用。

主要思路是固定 F 中的一个向量 $\boldsymbol{x}$，然后构造所有变换

$$\mathcal{T}_x = \bigcup_{\boldsymbol{y}\in\mathcal{Y}} \mathcal{T}_{xy} \tag{21}$$

将 $\boldsymbol{x}$ 映射到 $\boldsymbol{y} \in \mathcal{Y}$ 中的任意一个向量，其中 $\mathcal{Y}$ 表示 F'的向量。显然，若 F 和 F' 等价，即对于某个 $g \in \mathcal{G}$，有 $gF = F'$，那么 g 会将 $\boldsymbol{x}$ 映射到 F'的一个向量 $\boldsymbol{y}$，因此 $g \in \mathcal{T}_x$。选择 F' 中稳定子群最小且等价向量最少的向量 $\boldsymbol{x}$ 后，$\mathcal{T}_x$ 通常会很小，可以逐一检查。然而，即使最小稳定子群很大，也可以在不将整个集合保存在内存中的情况下枚举 $\mathcal{T}_x$ 的元素。

算法 1 按如下方式执行此测试。在第 6 行和第 7 行，选择 F' 中稳定子群最小的向量 $\boldsymbol{x}$；如果有多个选择，则选取等价向量数量最少的那个。在第 10 行，我们将这些等价向量的集合存储为 $\mathcal{Y}_x$。无论如何选择 $\boldsymbol{x}$，取 F 的顶点集合与法向量集合中较小的一个作为 $\mathcal{D}$（第 12–17 行）。对于 F'，我们以类似方式选择 $\mathcal{D}'$。由于稳定子群是一个群，我们可以在第 18 行使用*GAP* 中的方法遍历其所有元素，同时每次只在内存中保留一个元素。对于每个元素 $g_s \in \mathrm{Stab}_{\mathcal{G}}(\boldsymbol{x}^{\text{rep}})$ 以及每个 $\boldsymbol{y} \in \mathcal{Y}_x$，我们构造变换（第 21 行）

$$g = g_y g_s g_x^{-1} \tag{22}$$

并判断两个集合 $g\mathcal{D}$ 和 $\mathcal{D}'$ 是否相等。如果相等，那么 F 等价于 F' 和 $gF = F'$。如果对于所有 $g_s \in \mathrm{Stab}_{\mathcal{G}}(\boldsymbol{x}^{\text{rep}})$ 和所有 $\boldsymbol{y} \in \mathcal{Y}_x$，它们都不相等，那么这两个面在 $\mathcal{G}$ 作用下不等价。

B. 构造向量的轨道

我们采用标准轨道枚举技术的一种变体，例如 [18] 中所实现的技术。算法 2 构造群 $\mathcal{G}$ 对向量 $\boldsymbol{x}$ 的轨道，并存储将 $\boldsymbol{x}$ 映射到其轨道中各元素的群元素。这些群元素是

算法 1 判断两个面 F 和 F' 在 $\mathcal{G}$ 作用下是否等价。若等价，则返回将 F 映射到 F' 的变换 $g \in \mathcal{G}$；否则返回 NULL。我们将 $|\mathcal{X}|$ 定义为集合 $\mathcal{X}$ 中元素的个数，将 $\arg\min_{\boldsymbol{x}\in\mathcal{X}} f$ 定义为返回 $\mathcal{X}$ 的子集的函数，该子集中的 $f(\boldsymbol{x})$ 取值最小。

1: **procedure** FINDTRANSFORMATION(F, F', $\mathcal{G}$)
2: $\mathcal{V} \leftarrow$ vertices of F
3: $\mathcal{V}' \leftarrow$ vertices of F'
4: $\mathcal{N} \leftarrow$ normal vectors of F
5: $\mathcal{N}' \leftarrow$ normal vectors of F'
There may be multiple minima, which are all captured:
6: $\mathcal{M} \leftarrow \arg\min_{\boldsymbol{x}\in\mathcal{V}\cup\mathcal{N}} |\mathrm{Stab}_{\mathcal{G}}(\text{REPOF}(\boldsymbol{x}, \mathcal{G}))|$
Any of the elements with the smallest number of equivalent vectors of F' is selected:
7: $\boldsymbol{x} \leftarrow \text{ANY}(\arg\min_{\boldsymbol{x}\in\mathcal{M}} |\{\boldsymbol{y} \in \mathcal{V}' \cup \mathcal{N}' : \boldsymbol{y} \sim \boldsymbol{x}\}|)$
8: $\boldsymbol{x}^{\text{rep}} \leftarrow \text{REPOF}(\boldsymbol{x}, \mathcal{G})$
9: $g_x^{-1} \leftarrow$ inverse of $\text{TRANSFORMOF}(\boldsymbol{x}, \mathcal{G})$
10: $\mathcal{Y}_x \leftarrow \{\boldsymbol{y} \in \mathcal{V}' \cup \mathcal{N}' : \boldsymbol{y} \sim \boldsymbol{x}\}$
11: $\mathcal{T}_y \leftarrow \{\text{TRANSFORMOF}(\boldsymbol{y}, \mathcal{G}) : \boldsymbol{y} \in \mathcal{Y}_x\}$
12: **if** $|\mathcal{V}| < |\mathcal{N}|$ **then**
13: $\mathcal{D} \leftarrow \mathcal{V}$
14: $\mathcal{D}' \leftarrow \mathcal{V}'$
15: **else**
16: $\mathcal{D} \leftarrow \mathcal{N}$
17: $\mathcal{D}' \leftarrow \mathcal{N}'$
18: **for all** $g_s \in \mathrm{Stab}_{\mathcal{G}}(\boldsymbol{x}^{\text{rep}})$ **do**
19: $\bar{g} \leftarrow g_s g_x^{-1}$
20: **for all** $g_y \in \mathcal{T}_y$ **do**
21: $g \leftarrow g_y \bar{g}$
22: **if** $g\mathcal{D} = \mathcal{D}'$ **then**
23: **return** g
24: **return** NULL

实用函数:

- ANY($\mathcal{X}$) 返回 $\mathcal{X}$ 的任意一个元素
- REPOF($\boldsymbol{x}$, $\mathcal{G}$) 返回 $\boldsymbol{x}$ 的一个代表元，假设所有向量都已按 $\mathcal{G}$ 分类，并且已选定一组任意但固定的类代表元
- TRANSFORMOF($\boldsymbol{x}$, $\mathcal{G}$) 返回满足 $\boldsymbol{x} = g_x$REPOF($\boldsymbol{x}, \mathcal{G}$) 的 $g_x \in \mathcal{G}$；同样假设向量已按 $\mathcal{G}$ 分类，并且已知（至少）一个可将其代表元映射到 $\boldsymbol{x}$ 的群元素

在算法 1 的 TRANSFORMOF 过程中需要用到。结果以字典形式存储，其中每个键值对都由轨道中的一个元素 $\boldsymbol{y}$ 作为键，以及一个将 $\boldsymbol{x}$ 变换为 $\boldsymbol{y}$ 的任意变换矩阵作为值。我们将这样的字典称为轨道映射。

对于顶点，算法 1 并不需要大多数群元素，事实上，也不需要大多数顶点本身。由于只为固定的代表父面构造子面，因此只有代表面上的顶点可能出现。因此，我们的轨道算法只选择性地存储部分群元素，具体由算法 2 中使用的条件函数决定。这显著减少了顶点大轨道的内存使用量。对于 Λ_{16}，只需要所有 201 343200 个群元素中对应于 1 067 070 个群元素的

算法 2 构造向量 $\boldsymbol{x}$ 在群$\mathcal{G}$下的轨道。该群由一组生成元给出，这些生成元是生成全群的变换矩阵。对于 Λ_{16}，我们使用生成元 $=\{\boldsymbol{M}_1, \boldsymbol{M}_2\}$，其中 $\boldsymbol{M}_1, \boldsymbol{M}_2$ 见第三节。条件 是一个关于单个向量的布尔函数，用于指定是否应存储给定向量对应的变换矩阵。此过程返回轨道中的所有向量组成的集合以及一个轨道映射。详情见正文。

```
1: 过程 轨道(x,生成元,条件)
2:     轨道 ← {x}
3:     轨道映射 ← 新的空字典
4:     轨道映射[x] ←单位矩阵
5:     池 ←轨道映射的副本
6:     当池不为空时 执行
7:         新池 ← 新的空字典
8:         对于所有 y ∈ 池的键池 执行
9:             h ← 池[y]
10:            对于所有 g ∈ 生成元 执行
11:                y′ ← gy
12:                如果 y′ ∉ 轨道那么
13:                    轨道 ← 轨道 ∪ {y′}
14:                    h′ ← gh
15:                    新池[y′] ← h′
                       此 if 语句请参见正文:
16:                    如果 条件（y′）那么
17:                        轨道映射[y′] ← h′
18:        池 ←新池 _
19:    返回 轨道，轨道映射
```

顶点。[4]

标准轨道算法的思路是反复将群的生成元作用于初始向量和新构造出的向量，直到不再出现新向量。算法 2 使用了这一方法，其中池 和新池 变量用于记录在上一次迭代中出现的新向量。在第 16–17 行中，我们有条件地将向量及其变换存储在轨道映射中。如果所有向量都已知，新池便保持为空，从而满足 while 循环的终止条件。构造顶点的轨道时，选择条件 使其仅在向量属于某个代表面时为真 。对于相关向量，条件则设置为始终为真。

C. 构造向量的稳定子

对 [13]中的方法所做的第三处改动，是一种用于构造向量在群 $\mathcal{G}$ 作用下的稳定子的算法。我们的方法同样受到标准轨道-稳定子算法的启发，例如 [18] 中实现的算法。第 18 行的算法 1 需要用到稳定子：我们会遍历其中一个代表向量的稳定子中的所有元素。对于 $\mathcal{G}$ =Aut(Λ_{16})，表 I 中共列出了 8 个代表向量。此前，我们让 *GAP* 求出向量的稳定子。不过，利用每个向量轨道大小的信息，我们可以实现一种效率更高的方法。

[4]由于某些顶点同时出现在多个面片中，因此这个数量会因所选代表面片而异。

算法 3 构造 $\mathcal{G}$ 中 $\boldsymbol{x}$ 的稳定子群，其轨道大小已知。与算法 2 一样，群 $\mathcal{G}$ 由一组集合生成元 生成矩阵给出。详情请参阅正文。

```
1: 过程 稳定子群(x, 生成元,轨道大小) _
2:     G ← GAP由生成元得到的群
3:     稳定子群大小 ← |G|/轨道大小
4:     稳定子生成元 ←新的空列表
5:     稳定子群 ← GAP 仅包含 I_16 的群
6:     轨道映射 ← 新的空字典
7:     轨道映射[x] ← 单位矩阵
8:     对所有 g ∈ G 执行
9:         x′ ← gx
10:        如果 x′ ∈ 轨道映射的键 则
11:            g′ ←轨道映射[x′]
12:            g_s ← g^{-1}g′
13:            如果 g_s ∉ 稳定子群 则
14:                将 g_s 追加到 稳定子生成元
15:                稳定子群 ← GAP 由.....生成的群 稳定子生成元
16:                如果 |稳定子群| = 稳定子群大小 则
17:                    返回 稳定子群
18:        else
19:            轨道映射[x] ← g
```

在算法 3 中，我们将不同的群元素作用于向量 $\boldsymbol{x}$ （第 9 行），从而构造轨道中的元素。以这种方式访问的任意向量$\boldsymbol{x}'$ 都会与相应的群元素一起存储在轨道映射中（第 19 行）。每当遇到先前发现的向量 $\boldsymbol{x}'$ 时，我们就取回已存储的群元素 g' （第 11 行）。由于 $g\boldsymbol{x} = g'\boldsymbol{x}$，我们有 $g^{-1}g'\boldsymbol{x} = \boldsymbol{x}$ ，因此 $g_s = g^{-1}g'$ 是 $\boldsymbol{x}$的稳定子中的一个元素。如果它尚未属于迄今找到的子群稳定子群 $\subseteq$ Stab$_{\mathcal{G}}(\boldsymbol{x})$，则在第 14 行将其加入群生成元列表。在第 15 行更新稳定子群 后，我们通过将其大小与已知的稳定子群大小进行比较，检查它是否完整。

这得益于两个事实，使计算变得高效。首先，由于“生日悖论” [23, 第 3], 节， 第 10 行中的首次碰撞平均会在经过 $1 + \sum_{n=1}^{N} \prod_{i=1}^{n-1} (1 - i/N)$ 个群元素后出现（见下方第二个未编号公式[23,式 (12)]），其中 N 是 $\boldsymbol{x}$ 在 $\mathcal{G}$ 作用下的轨道大小。对于 Aut(Λ_{16})，这意味着表 I 中各向量的稳定子中，首次找到一个元素所需的迭代次数约为 83（对于 $\boldsymbol{n}_1$ 和 $\boldsymbol{v}_1$）到 10 210（对于 $\boldsymbol{v}_2$， $\boldsymbol{v}_4$， $\boldsymbol{v}_5$）。其次，稳定子通常只需很少的群元素即可生成。对于 Λ_{16}，由于每个稳定子都只需两个生成元即可生成，因此在单个处理器核心上，只需几分钟便可找到全部 8 个稳定子。

VI. 结论

本文完整描述了 Barnes–Wall 格 Λ_{16}的沃罗诺伊区域的相关向量、顶点和面类。我们利用这些结果计算了 Λ_{16}的精确二阶矩。为获得这些结果，我们改进了算法 [13], , 使其能够处理比以往更大的对称群。我们认为，该算法可用于分析

许多对称群已知的格，甚至可能适用于维数高于 16 的情况。

通过蒙特卡罗积分，对二阶矩的精确值进行了数值验证。此外，结果表明，数值结果的方差可以比传统刀切估计量得到的结果更准确地近似。这或许能显著改进未来二阶矩的数值估计。

参考文献

[1] E. S. Barnes 和 G. E. Wall，“以阿贝尔群定义的某些极值形式”，澳大利亚数学会杂志，第 1 卷，第 1 期，第 47–63 页，1959 年 8 月。[2] G. D. Forney, Jr.，“陪集码——第一部分：引言与几何分类”，*IEEE* 信息论汇刊，第 34 卷，第 5 期，第 1123–1151 页，1988 年 9 月。[3] ——，“陪集码——第二部分：二元格及相关码”，*IEEE* 信息论汇刊，第 34 卷，第 5 期，第 1152–1187 页，1988 年 9 月。[4] A. J. Hahn，“E. S. Barnes 和 G. E. Wall 的陪集格”，澳大利亚数学会杂志。*A*辑，第 49 卷，第 3 期，第 418–433 页，1990 年 12 月。[5] J. H. Conway 和 N. J. A. Sloane，《球堆积、格与群》，第 3 版。纽约州纽约市：施普林格，1999 年。 [在线]。网址：https://doi.org/10 1007 978 1 4757 6568 7 。 / - - - - J.-P. Adoul，“用于高效[6]语音编码的格编码与网格编码量化”，载于语音识别与编码，A. J. R. Ayuso 和 J. M. L. Soler 编。德国柏林：施普林格，1995 年，第 57 章，第 405–422 页。[7] S. Lyu、L. Liu、J. Lai、C. Ling 和 H. Chen，“基于格的公钥加密的格码”，2022 [ ] 2208 13325 。在线。[8]网址：https://arxiv.org/abs/。A. R. Calderbank、E. M. Rains、P. Shor 和 N. J. A. Sloane，“通过 GF(4) 上的码实现量子纠错”，*IEEE* 信息论汇刊，第 44 卷，第 4 期，第 1369–1387 页，1998 年。[9] P. Buser 和 P. Sarnak，“关于大亏格黎曼曲面的周期矩阵（附录由 J.H. Conway 和 N.J.A. Sloane 撰写）”，数学发明，第 117 卷，第 1 期，第 27–56 页，1994 年。

[10] J. H. Conway 和 N. J. A. Sloane，“格的沃罗诺伊区域、多胞体的二阶矩与量化”，*IEEE* 信息论汇刊，第 IT-28 卷，第 2 期，第 211–226 页，1982年3月。 [在线]。网址：https://doi.org/10 1109 1982 1056483 。 /[11]TIT。J. Mueller、M. Neunhoffer、F. Noeske 和 M. Horn，“orb，轨道枚举方法，4.8.3 版”，2019年9月，GAP 软件包。 [在线]。网址：https://gap-packages.github.io/orb[12] 本文的 arXiv 版本包含附属文件，其中收录了 Λ_{16}的面类目录。[13] D. Pook-Kolb、B. Allen 和 E. Agrell，“任意格的量化器常数的精确计算”，2022年。 [在线]。网址：https://arxiv.org/abs/2211 01987 。F. J. MacWilliams 和 N. J. A. Sloane，[14]《纠错码理论》。Elsevier，1977 年，第 16 卷。[15] J. H. Conway 和 N. J. A. Sloane，“关于某些格的沃罗诺伊区域”，*SIAM* 代数与离散方法杂志，第 5 卷，第 3 期，第 294–305 页，9 月 1984 [ ] 10 1137 0605031 。在线。网址：[16]https://doi.org/。/ E. Agrell 和 T. Eriksson，“用于量化的格优化”，*IEEE* 信息论汇刊，第 44 卷，第 5 期，第 1814–1828 页，1998 年 9 月。[17] S. Lyu、Z. Wang、C. Ling 和 H. Chen，“由复整数构造的更优格量化器”，2022年。 [在线]。网址：https://arxiv.org/abs/2204.01105[18] GAP 小组，“GAP——群、算法与编程，4.12dev 版。” [在线]。网址：https://www.gap-system.org[19] ——，“GAP 主开发仓库——群、算法与编程”，2021年10月，提交版本 401c797476b787e748a3890be4ce95ae4e5d52ae。 [在线]。网址：https://github.com/gap-system/gap [20] E. M. Bray，“gappy——GAP 的 Python 接口，0.1.0a4 版。”[在线]。网址：https://github.com/embray/gappy [21] E. Agrell、T. Eriksson、A. Vardy 和 K. Zeger，“格中的最近点搜索”，*IEEE* 信息论汇刊，第 48 卷，第 8 期，第 2201–2214 页，2002 年。[22] B. Allen 和 E. Agrell，“九维空间中的最优格量化器”，《物理学年鉴》，第 533 卷，第 12 期，文章编号 2100259，12 月 2021 [ ] 10 1002 202100259 。在线。网[23]址：https://doi.org/。/andp。P. Flajolet、D. Gardy 和 L. Thimonier，“生日悖论、优惠券收集问题、缓存算法与自组织搜索”，《离散应用数学》，第 39 卷，第 3 期，第 207–229 页，1992 年。