arXiv:2302.06230v1 [physics.comp-ph] 13 Feb 2023

高效生成四维随机旋转矩阵

雅各布·T´omas Bullerjahn[¶*] 和 Bal´azsF´abi´an[¶†]*Department of Theoretical Biophysics, Max Planck Institute of Biophysics, 60438 Frankfurt am Main, Germany*

Gerhard Hummer*Department of Theoretical Biophysics, Max Planck Institute of Biophysics, 60438 Frankfurt am Main, Germany andInstitute of Biophysics, Goethe University Frankfurt, 60438 Frankfurt am Main, Germany*（日期：2023年2月14日）

马尔可夫链蒙特卡罗算法依赖于试探移动，这些移动会根据特定准则被拒绝或接受。本文给出一种高效算法，可生成覆盖任意预设旋转角范围的四维（4D）随机旋转矩阵。这些矩阵可与蒙特卡罗方法结合，高效采样四维旋转的 SO(4) 群。这些矩阵无偏，并经过专门构造，使得重复旋转能够在 SO(4) 上实现均匀采样。四维旋转可用于优化粗粒化分子动力学模拟中稳定时间积分所需的质量划分，并有望进一步应用于机器人学和计算机视觉领域。

I. 引言

四维（4D）旋转出现在从理论物理到机器人学的广泛问题中。它们定义了受反平方力作用的二体运动的对称性，例如开普勒问题和氢原子 [1]。四维旋转已用于为相干光通信中的电磁波传播建模 [2]，、信号处理算法 [3]，并为刚体构造质点的等动量系统 [4]，，即保持总质量、质心和惯性张量不变的质点系统。它们还可用于表述机器人学和计算机视觉中的各种问题，例如手眼标定问题 [5] 和点集配准 [6]。四维旋转群 SO(4) 可用行列式为 1 的 4×4正交矩阵表示。四维旋转可视为一对正交平面内的两个不同旋转 [7]，或者等价地，视为两个正交的三维旋转，因为旋转角和一个平面的法向量构成 SO(3) 的一个元素。这反映了 SO(4) 是 SO(3) 的双覆盖这一事实 × SO(3) [7]。大多数四维旋转都是所谓的双旋转，其两个旋转角 α 和 β的值各不相同；但也可能出现等倾旋转（$\alpha=\beta$）和简单旋转（ $\alpha=0$或 $\beta=0$）。

生成随机四维旋转矩阵的方法有多种。例如，凯莱变换

$$\mathbf{R}_4=(\mathbf{I}_4+\mathbf{S}_4)(\mathbf{I}_4+\mathbf{S}_4)^{-1}$$

将任意反对称 4×4 矩阵 $\mathbf{S}_4=-\mathbf{S}_4^\top$变换为 SO(4) 的一个元素，其中 $\top$ 表示转置，而$\mathbf{I}_4$ 是单位矩阵。然而，要生成均匀分布的旋转矩阵，也就是将某个对象均匀映射到三维球面上，$\mathbf{S}_4$的矩阵元素必须按照多元 t-分布 [8]的一种推广来分布。或者，也可以利用 QR 分解 [9] 或豪斯霍尔德变换 [10] ，为 $n\geq2$ 生成 SO(n) 的均匀分布元素。

虽然上述方法可用于高效生成均匀分布的 SO(4) 元素，但其主要缺点在于无法让使用者充分控制矩阵生成过程。具体而言，这些方法都不能专门生成在随机正交二维子空间内的小角度旋转。相反，必须采用计算成本高昂的分解方案，明确检查旋转角的大小，再决定接受或拒绝候选矩阵。然而，要在四维旋转空间中以小步长进行随机游走，从而高效开展蒙特卡罗采样，就必须使用小角度矩阵；例如，在分子动力学模拟中，为稳定数值计算，需要寻找某些分子结构的最优等动量系统 [11]。一种可能的出发点是四元数，即用角度和旋转轴来描述三维旋转的四元组，因为两个不同四元数的乘积会得到一个四维旋转矩阵 [7, 12]。类似地，人们可能会考虑将四维旋转凯莱分解为一对左、右等斜四维旋转矩阵 [13, 14]。然而，要使所得旋转幅度较小，所述四元数对或旋转矩阵对不能任意选取，而必须满足额外约束。

本文介绍了一种在四维空间中构造简单旋转、双旋转和等斜旋转矩阵的几何直观方法，旋转角度可任意取值。该方法利用了这样一个事实：任意四维旋转矩阵都可以写成某个反对称矩阵的矩阵指数，而该反对称矩阵又可以分解为两个正交反对称矩阵的和，分别由旋转角度 α 和 β [15]加权。

* jakob.bullerjahn@biophys.mpg.de
† balazs.fabian@biophys.mpg.de
¶ 这些作者对本研究的贡献相同。

本文结构如下：第II节简要回顾上述在三维和四维中的分解方案，随后求解生成矩阵 **A** 和**B**的逆问题。我们还探讨了四维旋转角的非平凡分布，并指出在小角度极限下， α 和 β 本质上可以从任意分布中采样。第III节给出一种高效算法（即所需的伪随机数数量极少），用于构造适合马尔可夫链蒙特卡洛采样方案的随机小角度 4×4 旋转矩阵。第IV 节通过数值测试验证该算法。第V节总结本文结果，随后附录给出各种技术细节和推导。

II. 理论

原则上，可以将一个双重旋转矩阵与

$$\mathbf{R}'_4(\alpha,\beta)=\begin{pmatrix}\cos(\alpha)&\sin(\alpha)&0&0\\-\sin(\alpha)&\cos(\alpha)&0&0\\0&0&\cos(\beta)&\sin(\beta)\\0&0&-\sin(\beta)&\cos(\beta)\end{pmatrix}$$

正交群 O(4) 中的任意元素结合起来，从而构造一个随机的小角度四维旋转矩阵，如下所示：

$$\mathbf{R}_4(\alpha,\beta\ll1)=\mathbf{Q}_4\mathbf{R}'_4(\alpha,\beta)\mathbf{Q}_4^\top\,. \tag{1}$$

虽然 $\mathbf{R}'_4$ 的旋转平面定义明确（该矩阵描述了在 xy 和 zw 平面内的旋转），但式（1）中的矩阵乘积确保 $\mathbf{R}_4$ 的旋转平面是随机的。

式（1）的缺点是计算效率低下。在一个计算代价高昂的过程中，可以通过对四个四维向量进行格拉姆-施密特正交化来生成 $\mathbf{Q}_4$ 。更高效地，SciPy 等软件包采用一种算法，只需 9 个伪随机数即可生成四维旋转矩阵；而式（1）还需要为旋转角额外提供 2 个自由度。然而，SO(4) 的维数为 6 [7], ，因此显然仍有改进空间；值得注意的是，生成高质量随机数的计算代价很高。

本节余下部分将讨论并建立理论依据，以便开发一种高效算法来生成小角度四维旋转。在第 IVA 节中，我们将证明，与公式（1）相比，我们的新算法运行时性能更优，因此更适用于涉及蒙特卡罗方法的应用。

A. 反对称矩阵的矩阵分解

一个 3×3 表示绕单位向量 $\vec{u}=(-s_{23},\ s_{13},\ -s_{12})^\top$ $|\vec{u}|=1$ 进行三维旋转的矩阵 $\mathbf{R}_3$, ，可以表示为一个反对称矩阵的指数形式

$$\mathbf{S}_3=\begin{pmatrix}0&s_{12}&s_{13}\\-s_{12}&0&s_{23}\\-s_{13}&-s_{23}&0\end{pmatrix}$$

通过罗德里格旋转公式

$$\mathbf{R}_3(\gamma)=\mathrm{e}^{\gamma\mathbf{S}_3}=\mathbf{I}_3+\sin(\gamma)\mathbf{S}_3+[1-\cos(\gamma)]\mathbf{S}_3^2\,.$$

这里， γ 表示旋转角度，$\mathbf{I}_3$ 是单位矩阵。根据凯莱-哈密顿定理以及向量 $\vec{u}$的范数约束，矩阵 $\mathbf{S}_3$满足性质 $\mathbf{S}_3^3=-\mathbf{S}_3$。

反对称矩阵的特征值可以是成对的共轭复数，也可以为零，因此其秩始终为偶数。所以在三维空间中，旋转的无穷小生成元$\mathbf{S}_3$ 的秩为 2，且是奇异矩阵，即 $\det(\mathbf{S}_3)=0$。设 $\lambda_{1,2}=\pm i\theta$ 和 $\lambda_3=0$ 是$\mathbf{S}_3$的特征值，相应的特征向量为 $\vec{s}_1,\vec{s}_2\in\mathbb{C}^3$ 和 $\vec{s}_3\in\mathbb{R}^3$，其中 $\vec{s}_1$ 和 $\vec{s}_2$ 构成一对共轭复向量。于是，矩阵$\mathbf{R}_3$ 与$\mathbf{S}_3$具有相同的特征向量，其特征值为 e^{λ_n}，其中 $n=1,2,3$。$\mathbf{S}_3$ 的秩亏体现了这样一个事实：三维旋转涉及一个平面和一个唯一的正交方向，沿该方向不会发生旋转，这个方向就是旋转轴。$\mathbf{S}_3$ 的非零元素表示相应平面内的旋转，例如，元素 s_{12} 会在 xy- 平面内产生旋转。旋转平面可以直接表示为以下向量张成的空间：

$$\vec{v}_1=\frac{1}{\sqrt{2}}\,\mathrm{Re}(\vec{s}_1+\vec{s}_2)\,,\qquad \vec{v}_2=\frac{1}{\sqrt{2}}\,\mathrm{Im}(\vec{s}_1-\vec{s}_2)\,,$$

其中 Re(·) 和 Im(·) 分别表示其参数的实部和虚部。令 $\vec{v}_3=\vec{s}_3$，即可在 $\mathbb{R}^3$ 中得到一组满足 $\vec{v}_i\cdot\vec{v}_j=\delta_{ij}$ 的正交归一向量组，其中 i，$j=1$，2，3。

矩阵表示与 $\mathbf{S}_3$ 和 $\mathbf{R}_3$ 的几何意义之间的这些联系同样适用于四维旋转，其中任意旋转矩阵都满足$\mathbf{R}_4=\mathrm{e}^{\mathbf{S}_4}$，其中的矩阵为反对称矩阵

$$\mathbf{S}_4=\begin{pmatrix}0&s_{12}&s_{13}&s_{14}\\-s_{12}&0&s_{23}&s_{24}\\-s_{13}&-s_{23}&0&s_{34}\\-s_{14}&-s_{24}&-s_{34}&0\end{pmatrix}\,.$$

矩阵 $\mathbf{S}_4$ 可以唯一分解为 [15]

$$\mathbf{S}_4=\alpha\mathbf{A}+\beta\mathbf{B} \tag{2}$$

对于 α，$\beta\geq0$, $\alpha\neq\beta$，其中有两个秩为2的反对称矩阵 $\mathbf{A}$ 和 $\mathbf{B}$，满足

$$\mathbf{A}^3=-\mathbf{A}\,,\qquad \mathbf{B}^3=-\mathbf{B}\,,\qquad \mathbf{AB}=\mathbf{BA}=0\,. \tag{3}$$

对于简单旋转，当其中一个角度 α，β 为零时，$\mathbf{S}_4$ 的秩也为 2，但一般情况下它是满秩的。相应的罗德里格旋转公式为 [15]

$$\begin{aligned}\mathbf{R}_4(\alpha,\beta)=\mathrm{e}^{\mathbf{S}_4}=\mathbf{I}_4&+\sin(\alpha)\mathbf{A}+[1-\cos(\alpha)]\mathbf{A}^2\\&+\sin(\beta)\mathbf{B}+[1-\cos(\beta)]\mathbf{B}^2\,.\end{aligned} \tag{4}$$

参考文献 15 介绍了一种算法，可根据给定的反对称矩阵 α 和 β，计算角度以及矩阵 $\mathbf{A}$ 和 $\mathbf{B}$。下文将讨论逆问题：构造一对矩阵 $\mathbf{S}_4$，使其满足式（$\mathbf{A}$ 和 $\mathbf{B}$），从而可以利用式（3）为小角度旋转生成随机四维旋转矩阵。我们的结果可用于在四维旋转空间中高效地执行随机游走，例如生成粒子的等惯量系统时就需要这样做 [4, 11]。

B. 构造矩阵 A

受四维旋转可表示为一对三维旋转这一事实的启发，我们考虑两个三维向量，$\vec{a}_1$ 和 $\vec{a}_2$，它们应满足关系

$$(\vec{a}_1 \cdot \vec{a}_2)^2 = \det(\mathbf{A})$$

对于反对称矩阵

$$\mathbf{A} = \begin{pmatrix} 0 & a_{12} & a_{13} & a_{14} \\ -a_{12} & 0 & a_{23} & a_{24} \\ -a_{13} & -a_{23} & 0 & a_{34} \\ -a_{14} & -a_{24} & -a_{34} & 0 \end{pmatrix}.$$

将 $\vec{a}_1$ 和 $\vec{a}_2$ 的元素与 a_{ij}相关联，有多种等价的方式。这里，我们选择 $\vec{a}_1$，与三维情形中的向量 $\vec{u}$ 类似；它唯一确定 $\vec{a}_2$，于是得到

$$\vec{a}_1 = \begin{pmatrix} -a_{23} \\ a_{13} \\ -a_{12} \end{pmatrix}, \qquad \vec{a}_2 = \begin{pmatrix} a_{14} \\ a_{24} \\ a_{34} \end{pmatrix}. \tag{5}$$

与三维情形类似，a_{ij} 表示在ij平面内的旋转，而 $\vec{a}_1$ 和 $\vec{a}_2$ 的点积将表示正交平面的元素组合起来。此外，可以证明，只要 $\vec{a}_1$ 和 $\vec{a}_2$满足以下条件，$\mathbf{A}^3 = -\mathbf{A}$ 就成立

$$\vec{a}_1 \cdot \vec{a}_2 = 0\,, \qquad \vec{a}_1 \cdot \vec{a}_1 + \vec{a}_2 \cdot \vec{a}_2 = 1\,. \tag{6}$$

要求 $\vec{a}_1$ 和 $\vec{a}_2$ 互相垂直，可确保$\mathbf{A}$ 是一个秩二矩阵，即涉及单个旋转平面的简单旋转。第二个要求是 $\vec{a}_1$ 和 $\vec{a}_2$ 构成一个六维单位向量（$\vec{a}_1^{\top}$，$\vec{a}_2^{\top}$）$^{\top}$ 可确保$\mathbf{S}_4$ 按照式（2）进行分解，正如我们将在第IIC节看到的那样。

乍一看，可以认为构造$\mathbf{A}$需要 6 个数，即对角线上方的每个元素各需要一个，但事实证明只需要 4 个。如附录 A中所述，这是因为我们总能从二球面的元素中任意选择一个辅助向量 $\vec{a}_1^*$，而这些元素完全由极角和方位角确定。这对应于 2 个自由度。随后，我们可以确定两个与 $\vec{a}_1^*$ 垂直的向量，它们张成一个平面，在该平面中，第二个辅助向量 $\vec{a}_2^*$ 随后通过增加一个自由度构造而成。最后，第四个自由度 R 充当 $\vec{a}_1^*$和 $\vec{a}_2^*$之间的“混合比”，从而得到两个向量 $\vec{a}_1 = \vec{a}_1^*\sqrt{R}$和 $\vec{a}_2 = \vec{a}_2^*\sqrt{1-R}$，它们满足公式（6）中的约束，并根据公式（5）定义矩阵 $\mathbf{A}$。

C. 根据已知矩阵 A 构造矩阵 B

一般来说，生成一个四维旋转需要 6 条信息 [7]。我们已经证明，构造简单旋转的秩二矩阵$\mathbf{A}$ 需要 4 个数，而式（2）的分解包含 2 个独立角，因此矩阵$\mathbf{B}$ 必定已编码在$\mathbf{A}$中。事实上，这两个矩阵共享相同的 4 个不同特征向量；与三维情形类似，这些特征向量可用于构造一个正交归一向量组，其张成两个正交旋转平面。矩阵 $\mathbf{A}$ 描述一个平面内的旋转，而$\mathbf{B}$ 描述另一个平面内的旋转；$\mathbf{A}$ 和$\mathbf{B}$ 的特征值决定哪个平面与哪个矩阵相关联。

接下来，我们将说明如何利用上述两个矩阵之间的关系，从给定的 $\mathbf{B}$ 构造$\mathbf{A}$，使它们的线性组合 [式（2）] 按照式（4）生成一个旋转矩阵。为此，我们考虑一个反对称矩阵

$$\mathbf{B} = \begin{pmatrix} 0 & b_{12} & b_{13} & b_{14} \\ -b_{12} & 0 & b_{23} & b_{24} \\ -b_{13} & -b_{23} & 0 & b_{34} \\ -b_{14} & -b_{24} & -b_{34} & 0 \end{pmatrix}$$

满足性质 $\mathbf{B}^3 = -\mathbf{B}$。根据第 IIB 节，存在两个向量

$$\vec{b}_1 = \begin{pmatrix} -b_{23} \\ b_{13} \\ -b_{12} \end{pmatrix}, \qquad \vec{b}_2 = \begin{pmatrix} b_{14} \\ b_{24} \\ b_{34} \end{pmatrix},$$

使得

$$\vec{b}_1 \cdot \vec{b}_2 = 0\,, \qquad \vec{b}_1 \cdot \vec{b}_1 + \vec{b}_2 \cdot \vec{b}_2 = 1\,, \tag{7}$$

$\vec{\ }$ $\vec{\ }$必须成立，才能使 $\mathbf{B}$的秩为 2。然而，b_1 和 b_2不能独立于 $\vec{a}_1$ 和 $\vec{a}_2$来选取，因为式（3）中的第三个条件。根据式（6）和（7）的约束，可求解所得方程，得到

$$\begin{aligned} b_{12} &= \mp a_{34}\,, & b_{13} &= \pm a_{24}\,, & b_{14} &= \mp a_{23}\,, \\ b_{23} &= \mp a_{14}\,, & b_{24} &= \pm a_{13}\,, & b_{34} &= \mp a_{12}\,, \end{aligned}$$

或者，等价地，

$$\vec{b}_1 = \pm\vec{a}_2\,, \qquad \vec{b}_2 = \pm\vec{a}_1\,.$$

请注意，选择上方或下方的符号，相当于在$\mathbf{B}$ 和$\mathbf{B}^\top = -\mathbf{B}$之间进行选择，归结为角度 β上的正向或负向旋转。因此，我们可以不失一般性地固定$\mathbf{B}$ 各元素的符号，并允许 β 取正值或负值。

D. 对角度 α 和 β进行采样

对于三维中的随机旋转 $\mathbf{R}_3(\gamma)$，旋转角 $\gamma \in [0, 2\pi)$ 并非均匀分布。相反，有 $\gamma \sim [1 - \cos(\gamma)]/2\pi$ [17]，其中记号 $x \sim p(x)$ 表示 x 服从$p(x)$。在四维情况下，进入方程（2）的角度 α，$\beta \in [0, 2\pi)$ 服从联合概率密度函数（概率密度函数） [17]

$$p(\alpha, \beta) = \frac{[\cos(\alpha) - \cos(\beta)]^2}{4\pi^2} \tag{8}$$

因此二者并不独立，即 $p(\alpha, \beta) \neq p(\alpha)\, p(\beta)$。然而，借助辅助变量 u和 v，满足 $\alpha = u + v$以及 $\beta = v - u$，即可将旋转角解耦。该变量替换将 $p(\alpha, \beta)$ 分解为关于 u，$v \in [0, 2\pi]$ 的独立同分布乘积，其函数形式为（见附录 B）

$$p(z) = \frac{\sin(z)^2}{\pi}\,. \tag{9}$$

原则上，可以在 $z \ll 1$的极限下展开相应的累积分布函数（CDF），并使用逆变换抽样为 α 和 β抽取小角度值，如附录 B中所示。然而，事实证明，由多个连续的小角度旋转生成的大尺度旋转的性质基本不受 α 和 β的小角度分布影响。实际上，要使大尺度旋转在 SO(4) 上服从均匀分布，只需确保：i）使用足够多的小角度旋转，以覆盖整个三维球面；ii）小角度旋转是可逆的。当向量 $\vec{a}_1$和 $\vec{a}_2$ 从二球面上的均匀分布中抽样时，后一项要求会自动满足，因为此时每个四维旋转（由与 $\vec{a}_1$ 和 $\vec{a}_2$相关联的两个正交平面刻画）与其反向旋转（由 $-\vec{a}_1$ 和 $-\vec{a}_2$ 给出）出现的概率相同。因此，我们可以为 α 和 β的小角度值选择任意分布进行抽样。

III. 高效生成小角度四维旋转矩阵的算法

我们现在可以着手制定一种生成随机小角度四维旋转矩阵的算法，该算法可用于马尔可夫链蒙特卡罗采样方案，以探索四维旋转空间。在第 IID 节中，我们已经提到，抽取小角度值 α 和 β 的分布可以任意选择。因此，这里我们将使用均匀分布的伪随机数，以尽量降低计算成本。构造矩阵 $\mathbf{A}$ 和 $\mathbf{B}$所需的随机向量 $\vec{a}^*_{i=1,\,2}$，可以通过将一个随机 SO(3) 元素作用于一对正交向量（例如，$(1,\,0,\,0)^\top$ 和 $(0,\,1,\,0)^\top$）来生成。不过，这种构造方案的效率在很大程度上取决于生成 SO(3) 元素所采用的方法。

因此，我们采用以下算法。该算法之所以高效，是因为它总共只需要 6 个随机数 [，对应于 SO(4)] 的维数：

1. 生成两个服从均匀分布的随机数，$R_1 \sim \mathcal{U}_{[-1,1]}$ 和 $R_2 \sim \mathcal{U}_{[0,2\pi]}$，其中 $\mathcal{U}_{[s,t]}$表示区间 $[s, t]$上的独立均匀分布。辅助单位向量 $\vec{a}^*_1$ 随后按如下方式给出（参见附录A）

$$\vec{a}^*_1 = \begin{pmatrix} \sqrt{1 - R_1^2}\cos(R_2) \\ \sqrt{1 - R_1^2}\sin(R_2) \\ R_1 \end{pmatrix}.$$

2. 再生成一个随机数 $R_3 \sim \mathcal{U}_{[0,2\pi]}$。辅助单位向量

$$\vec{a}^*_2 = \begin{pmatrix} R_1\cos(R_2)\cos(R_3) + \sin(R_2)\sin(R_3) \\ R_1\sin(R_2)\cos(R_3) - \cos(R_2)\sin(R_3) \\ -\sqrt{1 - R_1^2}\cos(R_3) \end{pmatrix}$$

与 $\vec{a}^*_1$ 垂直（参见附录A）。请注意，若随机数生成速度快于三角函数的计算速度，则可以改为在单位圆上取两个点，从而得到 R_2 和 R_3 的正弦和余弦。

3. 生成第四个随机数 $R_4 \sim \mathcal{U}_{[0,1]}$，它定义了辅助向量之间的“混合比”，即

$$\vec{a}_1 = \vec{a}^*_1\sqrt{R_4}\,, \qquad \vec{a}_2 = \vec{a}^*_2\sqrt{1 - R_4}\,.$$

4. 生成另外两个随机数 R_5，$R_6 \sim \mathcal{U}_{[0,1]}$，并设

$$\alpha = \varepsilon R_5\,, \qquad \beta = \varepsilon R_6\,,$$

对于 $\varepsilon \ll 1$。

5. 计算旋转矩阵

$$\mathbf{R}_4 = \mathrm{e}^{\mathbf{S}_4} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{34} & r_{44} \end{pmatrix}$$

对应于 $\mathbf{S}_4 = \alpha\mathbf{A} + \beta\mathbf{B}$，使用罗德里格旋转公式 [式（4）]。这对应于以下逐元素计算：

$$
\begin{aligned}
r_{11} &= (a_{1,y}^2 + a_{1,z}^2 + a_{2,x}^2)\Delta + \cos(\beta)\,,\\
r_{12} &= h_{12}\Delta - a_{1,z}\sin(\alpha) - a_{2,z}\sin(\beta)\,,\\
r_{21} &= h_{21}\Delta + a_{1,z}\sin(\alpha) + a_{2,z}\sin(\beta)\,,\\
r_{22} &= \cos(\alpha) - (a_{1,y}^2 + a_{2,x}^2 + a_{2,z}^2)\Delta\\
r_{13} &= h_{13}\Delta + a_{1,y}\sin(\alpha) + a_{2,y}\sin(\beta)\,,\\
r_{31} &= h_{31}\Delta - a_{1,y}\sin(\alpha) - a_{2,y}\sin(\beta)\,,\\
r_{23} &= h_{23}\Delta - a_{1,x}\sin(\alpha) - a_{2,x}\sin(\beta)\,,\\
r_{32} &= h_{32}\Delta + a_{1,x}\sin(\alpha) + a_{2,x}\sin(\beta)\,,\\
r_{33} &= \cos(\alpha) - (a_{1,z}^2 + a_{2,x}^2 + a_{2,y}^2)\Delta\\
r_{14} &= h_{14}\Delta + a_{2,x}\sin(\alpha) + a_{1,x}\sin(\beta)\,,\\
r_{41} &= h_{41}\Delta - a_{2,x}\sin(\alpha) - a_{1,x}\sin(\beta)\,,\\
r_{24} &= h_{24}\Delta + a_{2,y}\sin(\alpha) + a_{1,y}\sin(\beta)\,,\\
r_{42} &= h_{42}\Delta - a_{2,y}\sin(\alpha) - a_{1,y}\sin(\beta)\,,\\
r_{34} &= h_{34}\Delta + a_{2,z}\sin(\alpha) + a_{1,z}\sin(\beta)\,,\\
r_{43} &= h_{43}\Delta - a_{2,z}\sin(\alpha) - a_{1,z}\sin(\beta)\,,\\
r_{44} &= (a_{2,x}^2 + a_{2,y}^2 + a_{2,z}^2)\Delta + \cos(\beta)\,,
\end{aligned}
$$

其中 $\vec{a}_i = (a_{i,x}, a_{i,y}, a_{i,z})^\top$，$\Delta = \cos(\alpha) - \cos(\beta)$，并且

$$
\begin{aligned}
h_{12} &= h_{21} = a_{2,x}a_{2,y} - a_{1,x}a_{1,y}\,,\\
h_{13} &= h_{31} = a_{2,x}a_{2,z} - a_{1,x}a_{1,z}\,,\\
h_{23} &= h_{32} = a_{2,y}a_{2,z} - a_{1,y}a_{1,z}\,,\\
h_{14} &= h_{41} = a_{1,z}a_{2,y} - a_{1,y}a_{2,z}\,,\\
h_{24} &= h_{42} = a_{1,x}a_{2,z} - a_{1,z}a_{2,x}\,,\\
h_{34} &= h_{43} = a_{1,y}a_{2,x} - a_{1,x}a_{2,y}\,.
\end{aligned}
$$

加快算法速度的一种可能方法是仅依赖简单旋转。给定 $\beta = 0$、$\sin(\beta) = 0$ 和 $\cos(\beta) = 1$，第 4 步所需的随机数可减少一个，而第 5 步中 r_{ij} 的表达式也会得到简化。

通过替换第四步（见附录 B），也可以使用该算法生成均匀分布的 SO(4) 元素。

4. 生成另外两个随机数 $R_5, R_6 \sim \mathcal{U}_{[0,1]}$，并求出下式的根

$$f(z_i) = 2z_i - \sin(2z_i) - 8\pi R_i$$

对于 $z_5 = u$ 和 $z_6 = v$。旋转角随后由下式给出

$$\alpha = u + v\,, \qquad \beta = v - u\,.$$

IV. 算法测试

可以直接进行数值验证，算法在第III 满足

$$\mathbf{R}_4\mathbf{R}_4^\top = \mathbf{I}_4\,, \qquad \det(\mathbf{R}_4) = 1\,,$$

因此也是 SO(4) 的元素。在本节中，我们将其运行时性能与公式（1）进行比较，并验证随后多次应用小角度旋转会产生均匀分布的大尺度旋转。

A. 运行时性能

公式（1）和第 III 节中的算法均使用 Julia [18] 实现，并使用 BenchmarkTools.jl 软件包提供的 `@benchmark`宏进行计时 [19]。为生成进入公式（$\mathbf{Q}_4$ （1）的随机 O(4) 元素，我们将用于生成 SO(4)元素的算法的 Python 实现 [20] 转换为高效的 Julia 代码 [10] 。我们测试了小角度采样和联合角分布采样 p（α，β）[公式（8）]。每种算法的运行时测量均使用 Julia v.1.8.0，在配备 1.2 GHz 四核 Intel Core i7 处理器的计算机上完成。

表 I. 数值测试中生成 SO(4) 旋转矩阵的不同算法的运行时间。

算法	角度	运行时间中位数
第 III	$\alpha, \beta \sim p(\alpha, \beta)$	2.332 微秒
第 III	$\alpha, \beta \sim \mathcal{U}_{[0,\varepsilon]}$	207.980 纳秒
第 III	$\alpha \sim \mathcal{U}_{[0,\varepsilon]}$, $\beta = 0$	182.310 纳秒
式（1）	$\alpha, \beta \sim \mathcal{U}_{[0,\varepsilon]}$	1。618 微秒

我们的结果汇总于表I。使用第III 节的算法生成小角度双旋转，比式（1）快一个数量级，并且仅比将自己限制于具有$\beta = 0$的简单旋转略慢。

B. 渐近均匀性

如果旋转角 α 和 β 是从其联合概率密度函数 [式（8）], 中随机抽取的，我们便可以验证由此得到的随机四维旋转矩阵是否均匀分布。例如，可以使用第 III节中的算法（修改第四步）生成一组随机矩阵 $\{\mathbf{R}_4^{(m)}\}_{m=1}^M$，并检查随机点的分布

$$\vec{q}'_m = \mathbf{R}_4^{(m)}\vec{q} \tag{10}$$

当用这些矩阵旋转一个固定的单位四维向量时所产生的 $\vec{q}$。对于较小的旋转角，需要进行多次旋转才能合成大尺度旋转，因此必须将式（10）替换为

$$\vec{q}'_m = \mathbf{R}_4^{(1,m)}\mathbf{R}_4^{(2,m)}\ldots\mathbf{R}_4^{(N,m)}\vec{q}\,. \tag{11}$$

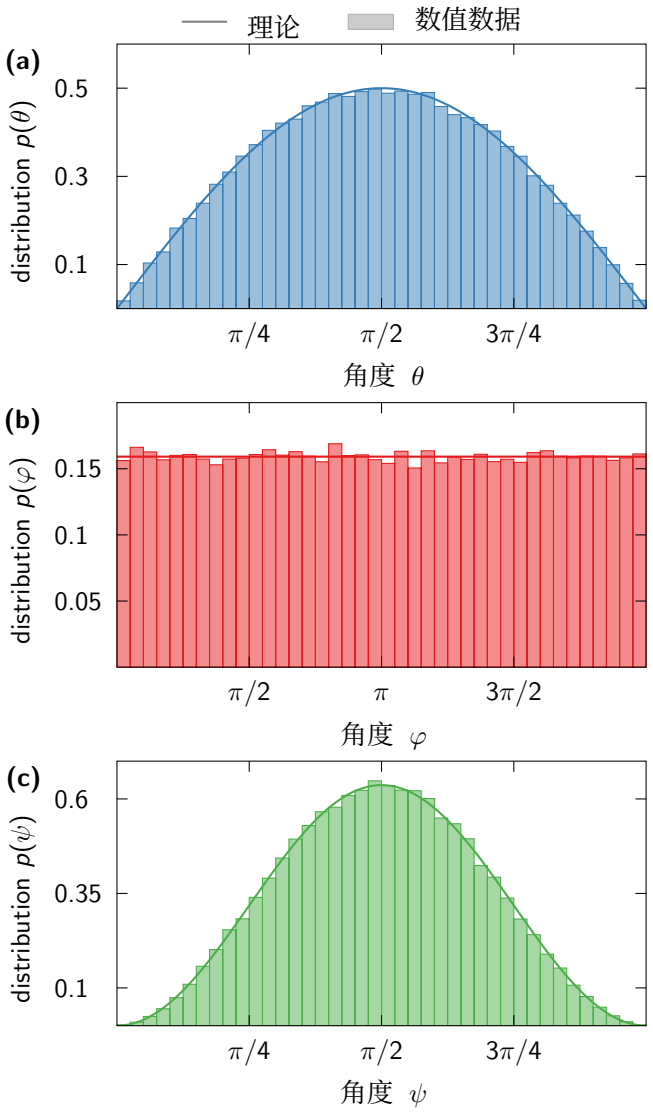


图1. 单位三维球面 S^3 上均匀分布对应的超球面角分布。通过反复将 $N = 10^5$ 随机的小角度（$\varepsilon = 0.05$）旋转矩阵作用于固定的四维向量 $\vec{q} = (0,\ 0,\ 0,\ 1)^\top$，生成了单位三维球面上的一组 $\{\vec{q}'_m\}_{m=1}^M$ $M = 10^5$ 个随机点。数值数据（直方图）很好地再现了角度分布：(a) $p(\theta)$ =sin(θ)/2，(b) $p(\varphi)$ = $1/2\pi$，以及 (c) $p(\psi)$ =2 sin(ψ)^2/2。如果这些点 $\vec{q}'_m$ 均匀地覆盖三维球面的表面，预期就会得到这些分布。

在超球坐标中，我们有（参见附录 C）

$$\vec{q}'_m = \begin{pmatrix} \sin(\psi)\sin(\theta)\cos(\varphi) \\ \sin(\psi)\sin(\theta)\sin(\varphi) \\ \sin(\psi)\cos(\theta) \\ \cos(\psi) \end{pmatrix}$$

由于 $|\vec{q}| = 1$，因此验证点 $\vec{q}'_m$ 是否均匀分布在 S^3 上，就等同于比较

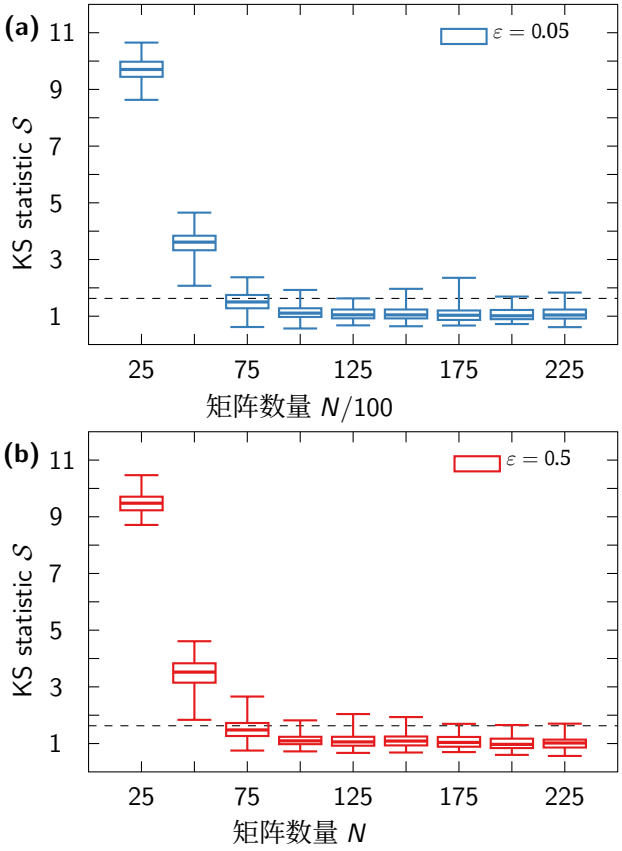


图2。KS统计量 $\mathcal{S} = \max(\mathcal{S}_\theta,\ \mathcal{S}_\varphi,\ \mathcal{S}_\psi)$ 的分布取决于累积旋转之后的矩阵数量 N。对于每个 $\mathcal{S}$，都可以计算相应的 p 值，而虚线标示 $p = 0.01$。因此，如果相应的 $\mathcal{S}$ 分布大多位于虚线下方，我们就可以有把握地认为，N个小角度旋转矩阵的集体旋转会生成均匀分布的 SO(4) 元素。(a) 对于 $\varepsilon = 0.05$，需要数万个小角度旋转矩阵；而 (b) $\varepsilon = 0.5$ 只需要 $N \approx 100$。

$\{\theta_m\}$、$\{\varphi_m\}$ 和 $\{\psi_m\}$ 的分布，以

$$p(\theta) = \frac{\sin(\theta)}{2}, \quad p(\varphi) = \frac{1}{2\pi}, \quad p(\psi) = \frac{2\sin(\psi)^2}{\pi}. \quad (12)$$

在图1中，通过式（11）生成的 $M = 10^5$个随机点的超球坐标被整理成直方图，并与式（12）的理论预测并列展示，其中 $N = 10^5$ 和$\varepsilon = 0.05$。从视觉上看，数值数据与解析概率密度函数吻合得很好。借助柯尔莫哥洛夫–斯米尔诺夫（KS）检验，还可以进行定量比较；该检验考察差异

$$\delta_+^{(m)} = m/M - P(z_m)\,,$$
$$\delta_-^{(m)} = P(z_m) - (m-1)/M\,,$$

经验分布函数与累积分布函数之间

$P(z)$ 变量 $z \in \{\theta, \varphi, \psi\}$。KS统计量

$$\mathcal{S}_z = \sqrt{M} \max_{1 \le m \le M} \left(\delta_+^{(m)}, \delta_-^{(m)}\right) \tag{13}$$

给出了一个衡量样本$\{z_m\}_{m=1}^M$ 取自概率密度函数 $p(z)$ 的可能性的指标。对于图1中的分布，我们有

$$P(\theta) = \sin(\theta/2)^2\,, \quad P(\varphi) = \varphi/2\pi\,,$$
$$P(\psi) = [\psi - \sin(\psi)\cos(\psi)]/\pi\,,$$

相应的 KS 统计量分别为 $\mathcal{S}_\theta \approx 0.90$、$\mathcal{S}_\varphi \approx 0.91$ 和 $\mathcal{S}_\psi \approx 1.04$。最大的 KS 统计量对应的 p 值为 0.23，因此我们可以相当确信这样的假设：$N = 10^5$ 小角度旋转矩阵的累积旋转会产生均匀分布的四维旋转。

图2探讨了构造均匀分布的 SO(4) 元素实际上需要多少个小角度旋转矩阵 N。为此，我们考察了 100 组点 $\{\vec{q}_m\}_{m=1}^M$，其中 $M = 1000$，并针对每组计算球面坐标 [式（13）] 的 KS 统计量 θ，φ 和 ψ。对于 $\varepsilon = 0.05$ [图2(a)]，(该图也用于生成图1中的数据)，我们发现，大约一万个小角度矩阵即可保证得到均匀分布；而对于 $\varepsilon = 0.5$[图2(b)]，只需要大约 100 个矩阵。用于蒙特卡罗方法时，参数 ε 因此可视为步长的大小，可对其进行调整，例如优化接受的移动所占的比例。

V. 结论

本文提出了一种几何上颇具吸引力的方法，可利用两个正交三维向量和两个旋转角构造四维旋转矩阵（第II节）。对角度变量的显式控制使我们在生成 SO(4) 元素时优于现有方法。利用这一优势，我们设计了一种算法，专门用于生成四维空间中的小角度旋转矩阵（第III节）。我们还证明（第IVB节），反复应用由所提出的算法生成的多个连续小角度旋转，可以实现四维旋转的均匀采样，这正是无偏算法所要求的。

该算法的高效性体现在：它仅需最少的 6 个伪随机数，其速度比替代方案快一个数量级（第 IVA节），并且可以与蒙特卡罗方法结合，用于解决物理学 [4] 和工程学中涉及四维旋转的优化问题。一个实际例子是对粗粒粒度胆固醇模型的受约束支架进行优化 [11]，该模型与 MARTINI 力场结合使用 [21]，用于生物系统的分子动力学模拟。

致谢

感谢 Andrew J. Hanson 教授（荣休）与我们进行讨论。本研究得到了马克斯·普朗克学会的支持。B.F. 感谢亚历山大·冯·洪堡基金会提供资助。

附录 A：在三维球面上生成均匀分布的点

在球坐标中，只需两个角度 $\theta \in [0, \pi]$和 $\varphi \in [0, 2\pi)$ 即可确定单位球面 $S^2 = \{\vec{r} \in \mathbb{R}^3 \mid \|\vec{r}\| = 1\}$ 上的一个点。该点的位置由下式给出

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sin(\theta)\cos(\varphi) \\ \sin(\theta)\sin(\varphi) \\ \cos(\theta) \end{pmatrix} \tag{A1}$$

在笛卡尔坐标中。已有许多算法可用于生成在 S^2 上均匀分布的点。这里，我们实际上是从相应的分布中抽取 θ 和 φ 的随机值。

S^2 上的均匀分布用球坐标表示为 $p(\theta,\ \phi) = \sin(\theta)/4\pi$，这意味着方位角 φ 在区间 $[0, 2\pi]$ 上均匀分布，即 $\varphi \sim \mathcal{U}_{[0,2\pi]}$。极角 θ 的分布为 $p(\theta) = \sin(\theta)/2$。因此，为了从该分布中抽样，我们采用逆变换抽样，并根据均匀分布的随机数 $R \in [0, 1]$ 构造 $\theta \sim p(\theta)$，如下所示：

$$\theta = \arccos(1 - 2R)\,.$$

注意，对于 $R \sim \mathcal{U}_{[0,1]}$，我们有 $1 - 2R \sim \mathcal{U}_{[-1,1]}$，因此

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sqrt{1 - R_1^2}\cos(R_2) \\ \sqrt{1 - R_1^2}\sin(R_2) \\ R_1 \end{pmatrix}\,, \tag{A2}$$

其中 $R_1 \sim \mathcal{U}_{[-1,1]}$，且 $R_2 \sim \mathcal{U}_{[0,2\pi]}$ 在 S^2 上均匀分布。这里，我们利用了对于 $-1 \le x \le 1$，必有 $\sin\left(\arccos(x)\right) = \sqrt{1 - x^2}$ 这一事实。

如果将方程（A2）用作辅助向量的替代项$\vec{a}_1^*$，那么 $\vec{a}_2^*$ 在由正交归一向量张成的平面内有无穷多个候选向量

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos(\theta)\cos(\varphi) \\ \cos(\theta)\sin(\varphi) \\ -\sin(\theta) \end{pmatrix}\,, \quad \begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} = \begin{pmatrix} \sin(\varphi) \\ -\cos(\varphi) \\ 0 \end{pmatrix}\,.$$

我们可以通过随机旋转，从该平面中随机选取一个单位长度的候选向量，得到

$$\vec{a}_2^* = \cos(R_3)\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \sin(R_3)\begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix}$$

对于 $R_3 \sim \mathcal{U}_{[0,2\pi]}$。

附录 B：小旋转角采样

正文中的方程（8）可改写如下 [17]:

$$p(\alpha,\beta)=\frac{1}{\pi^2}\sin\left(\frac{\alpha+\beta}{2}\right)^2\sin\left(\frac{\alpha-\beta}{2}\right)^2,$$

这提示我们进行坐标代换 $u=(\alpha-\beta)/2$和 $v=(\alpha+\beta)/2$，以解耦变量。随后，u和 v 均服从正文公式 (9) 所描述的分布。对于 $\alpha,\beta\in[0,2\pi]$，$(u,v)=\big((\alpha-\beta)/2,(\alpha+\beta)/2\big)$所形成的菱形区域，可以扩展为正方形 $[-\pi,\pi]\times[0,2\pi]$，或等价地，$[0,2\pi]\times[0,2\pi]$。

公式 (9) 的累积分布函数为

$$P(z)=\frac{2z-\sin(2z)}{4\pi}$$

对于 $z\in[0,2\pi]$，可通过逆变换抽样生成 u 和 v 的随机值，具体而言，需要求解

$$P(z)=R_z \tag{B1}$$

对于 $z\in\{u,v\}$ 以及一组均匀分布的随机数 $R_z=R_u$，$R_v\sim\mathcal{U}_{[0,1]}$。在小角度极限下，式（B1）化简为

$$\varepsilon^3R_z=\frac{z^3}{3\pi}+\mathcal{O}(z^4)$$

对于 $\varepsilon\ll1$，由此得到以下随机旋转角：

$$\alpha=\varepsilon(R_u^{1/3}+R_v^{1/3}),\qquad \beta=\varepsilon(R_v^{1/3}-R_u^{1/3}).$$

然而，正如第 IID 节所述，并在第III节的具体示例中所展示的，α 和 β 背后的分布不会影响多个连续小角度旋转所产生的大尺度旋转的性质。

附录 C：超球坐标

球坐标非常适合具有 $\mathbb{R}^3$旋转对称性的系统，也可以推广到高维空间中所谓的超球坐标。在四维空间中，在球坐标 r、θ 和 φ中增加一个角坐标 $\psi\in[0,2\pi]$，从而四维向量的笛卡尔坐标可表示为

$$\begin{pmatrix}x\\y\\z\\w\end{pmatrix}=\begin{pmatrix}r\sin(\psi)\sin(\theta)\cos(\varphi)\\r\sin(\psi)\sin(\theta)\sin(\varphi)\\r\sin(\psi)\cos(\theta)\\r\cos(\psi)\end{pmatrix}.$$

反过来，对于给定的四维向量（x,y,z,w），可以按如下方式计算 r、θ、φ 和 ψ：

$$\begin{aligned}r&=\sqrt{x^2+y^2+z^2+w^2},\\ \psi&=\arccos(w/r),\\ \theta&=\arccos(z/\sqrt{r^2-w^2}),\\ \varphi&=\begin{cases}\arccos(x/\sqrt{r^2-w^2-z^2}), & y\ge0\\ 2\pi-\arccos(x/\sqrt{r^2-w^2-z^2}), & y<0\end{cases}.\end{aligned}$$

请注意，本文仅考虑单位长度的四维向量，因此 $r\equiv1$。

[1] D. Oliver, *The Shaggy Steed of Physics*, （Springer-Verlag，纽约，2004）。[2] M. Karlsson，“相干光通信中的四维旋转”，《光波技术杂志》**32**，1246- 1257（2014）。[3] A. Borowicz，“关于使用四元数旋转进行独立分量分析”，载于 *Signal Processing: Algorithms Architectures Arrangements and Applications ,,,, -(SPA 2018)* 114 119 2018（IEEE，）。[4] L. P. Laus 和 J. M. Selig，“使用质点等惯量系统的刚体动力学”，《力学学报》**231**，221 236 2020-（）。[5] J. Wu、Y. Sun、M. Wang 和 M. Liu，“手眼标定：四维普罗克鲁斯分析方法”，《IEEE 仪器与测量汇刊》**69**，2966-2981（2020）。[6] S. Sarabandi 和 F. Thomas，“通过四维旋转近似 $\mathbb{R}^3$ 中的位移及其在点云配准中的应用”，《IEEE 机器人汇刊》**38**，2652- 2664（2022）。

[7] P. Lounesto, *Clifford Algebras and Spinors*（剑桥大学出版社，剑桥，2001年）。[8] C. A.Le´on、J.-C. Mass´e 和 L.-P. Rivest，“随机旋转的统计模型”，《多元分析杂志》**97**，412–430（2006年）。[9] F. Mezzadri，“如何从经典紧群生成随机矩阵”，《美国数学学会通讯》**54**，592–604（2007年）。[10] G. W. Stewart，“随机正交矩阵的高效生成及其在条件估计中的应用”，《SIAM数值分析杂志》**17**，403–409（1980年）。[11] B. F´abi´an、S. Thallmair 和 G. Hummer，“胆固醇分子动力学模拟的最优键约束拓扑”，《化学理论与计算杂志》，https://doi. org/10.1021/acs.jctc.2c01032。[12] A. J. Hanson, *Visualizing quaternions*（Morgan Kaufmann出版社，马萨诸塞州伯灵顿，2006年）。[13] A. Perez-Gracia 和 F. Thomas，“关于Cayley四维旋转分解及其应用”，《应用进展》

Clifford代数 **27**，523–538（2017年）。[14] S. Sarabandi、A. Perez-Gracia 和 F. Thomas，“关于Cayley分解及其在含噪旋转矩阵正交归一化中的应用”，《应用进展》Clifford代数 **29**，49（2019年）。[15] M. Erdo˘gdu 和 M. ¨Ozdemir，“简单旋转、双旋转和等倾旋转及一种可行算法”，《数学科学应用电子笔记》**8**，11–24（2020年）。[16] P. Virtanen、R. Gommers、T. E. Oliphant、M. Haberland、T. Reddy、D. Cournapeau、E. Burovski、P. Peterson、W. Weckesser、J. Bright、S. J. van der Walt、M. Brett、J. Wilson、K. J. Millman、N. Mayorov、A. R. J. Nelson、E. Jones、R. Kern、E. Larson、C. J. Carey、I. Polat、Y. Feng、E. W. Moore、J. VanderPlas、D. Laxalde、J. Perktold、R. Cimrman、I. Henriksen、E. A. Quintero、C. R. Harris、A. M. Archibald、A. H. Ribeiro、F. Pedregosa、P. van Mulbregt 以及SciPy 1.0 贡献者，“SciPy 1.0：有趣的 Python 科学计算基础算法”，Nat. Methods **17**，261-272（2020）。[17] H. Rummler，“关于旋转角的分布：随机旋转的平均旋转角有多大？”，Math. Intell. **24**，6-11（2002）。[18] J. Bezanson、A. Edelman、S. Karpinski 和 V. B. Shah，“Julia：一种数值计算的新方法”，SIAM Rev. 59, 65（2017）。[19] https://github.com/JuliaCI/BenchmarkTools.jl[20] 参见 https://github.com/scipy/scipy/blob/v1.4.1/scipy/stats/_multivariate.py 查看原始代码。[21] S. J. Marrink、H. J. Risselada、S. Yefimov、D. P. Tieleman、A. H. de Vries，“MARTINI 力场：用于生物分子模拟的粗粒化模型”，J. Phys. Chem. B **111**，7812-7824（2007年）。