N-维刚体动力学[1]

MARC TEN BOSCH, mtb design works, Inc., USA

译: 中国人民大学 信息学院

崔冠宇、李泽轩、罗迪、邵宁录、王玺、张晨阳[2]

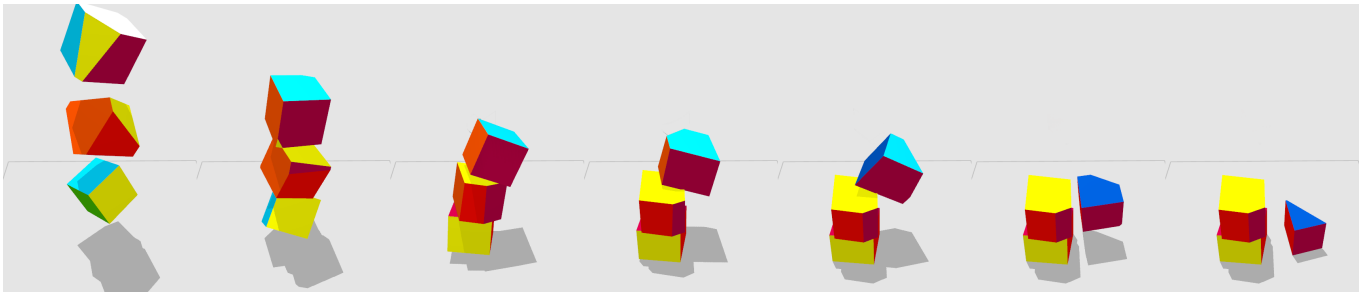


图 1: 三个 4D 超立方体的堆叠.

我提出了一个与空间维数无关的刚体动力学公式，用几何代数 (geometric algebra) 描述刚体的运动状态和运动方程。同时把碰撞检测算法扩展到 n 维就能解决刚体的碰撞和接触问题。在这里的具体实现是在四维情况下的，但这里描述的技术适用于任何维度。我通过四维刚体的三维切片展示它们，同时也允许用户实时操作这些物体。

CCS 概念: • **计算方法 → 物理模拟; 碰撞检测;** *虚拟现实*。

附加的关键字和短语: 刚体, *N* 维, 物理, 几何代数, 第四维度

ACM 引用格式:

Marc ten Bosch. 2020. *N*-Dimensional Rigid Body Dynamics. *ACM Trans. Graph.* 39, 4, Article 55 (July 2020), 6 pages. `https://doi.org/10.1145/3386569.3392483`

1 简介

我们对物理空间的体验是三维的。因此，基于物理的模拟 (物理引擎) 到目前为止只适用于二维和三维的情况。然而，使用所需方程的适当形式就有可能将它们扩展到更高的维度。几何代数提供了一个简单的无量纲公式。这允许实时操纵 n 维形状，使它们就好像是真实的对象一样相互碰撞，这也使得它们变得不那么抽象，与大多数人对它们的直观印象形成了鲜明的对比。虽然人们已经注意到对高维空间和其他抽象数学概念的理解和可视化，但这些被可视化的概念大多数情况下仍有局限，由于它们没有物理上的，或物与物之间的关系。

贡献 本文的贡献包括:

(1) 将经典的三维情况下的刚体几何动力学规则推广到 n 维。通过把几何代数中的运算符表示为矩阵，可以通过一

[1] 原文: Marc ten Bosch. 2020. *N*-Dimensional Rigid Body Dynamics. *ACM Trans. Graph.* 39, 4, Article 55 (July 2020), 6 pages. `https://doi.org/10.1145/3386569.3392483`

[2] 共同翻译，排名不分先后。

种简单的方式对角化并转换惯性张量到任意 n 维的空间中来构造。这使得我们可以构造 n 维情况下的欧拉方程，例如它允许我们学习在四维无力矩条件下的欧拉方程。

(2) 计算在 n 维下的碰撞和接触过程，包括静摩擦和动摩擦。我基于几何代数的碰撞检测方法给出了一个 n 维形式的闵可夫斯基差和分离轴定理 (Minkowski difference and separating axis theorem)。

(3) 一种与四维物体互动的方法，类似于我们对现实的三维体验。

2 相关工作

三维刚体动力学的交互式仿真是一个广阔的研究领域。Bender 等人 [2014] 提供了一项关于这个方面的调查。

[Cameron 1990] 通过考虑每个物体随时间的挤压，将三维连续碰撞检测问题表示为离散的 4D 碰撞问题。

可视化四维对象是一个有趣和具有挑战性的问题，其本身具有悠久的历史 [Abbott 1884; Banchoff 1990; Chu et al. 2009; Hilbert and Cohn-Vossen 1952]。许多操纵四维形状的方法已经被提出 [Yan et al. 2012; Zhang and Hanson 2006]。

几何代数作用于一个多维向量的空间，其中的向量是一个子空间。向量可以看作是有方向的线段，而其他元素可以表示有方向的平面 (双向量) 、空间 (三向量) 等等。几何代数还定义了与这些元素的平移和旋转相对应的操作，Macdonald 的书 [2011] 和 Gunn、De Keninck 的课程 [2019] 提供了介绍。Dorst 等人 [2009] 介绍了如何在程序中应用实现这些几何代数。Doran 和 Lasenby [2003] 将几何代数应用于三维刚体的情况。

凯莱 [1846] 首次提出推广至 n 维的欧拉方程。这个问题主要是通过矩阵分析来研究的，然而矩阵分析很快就变得复杂起来 (例如 [Sinclair and Hurtado 2005])。

由于其简单性、通用性，我选择使用几何代数，这可以使得运动方程在推广到 n 维的时候能够保持和三维情况下一样的形式。

射影几何代数和共形几何代数 (projective and conformal geometric algebras) 允许将平移分量和旋转分量组合成一个方程，其方式类似于齐次坐标中允许矩阵将这两个分量表示为单个变换。Gunn [2011] 使用射影几何代数以一种度量中立的方式 (metric-neutral way) 来表示二维和三维刚体运动，这也适用于非欧几里德空间。为了简单起见，我决定不使用这个公式。

3 背景

本部分简要回顾几何代数和由 Doran 和 Lasenby 提出的 [2003] 在三维刚体动力学中的应用，这些方程在 n 维中保持不变。

利用几何代数可以将刚体在 $n > 1$ 维度的公式表述为:

$$x_t = v \qquad R_t = -\frac{1}{2}\omega R$$

$$v_t = F/m \qquad L_t = \tau$$

其中 x 和 R 分别表示位置 (向量) 和方向 (转子，rotor)，v 表示速度 (向量)，ω 表示角速度 (双向量)，F 是净力 (向量)，m 是质量 (标量)，τ 表示净转矩 (双向量)，L 表示角动量 (双向量)。t 下脚标表示时间的导数。角速度双向量和方向转子的乘积是几何乘积，后面会讲到。

角速度用双向量表示。一个双向量由两个向量的外积组成:

$$B = a \wedge b$$

有 $k = \binom{n}{2}$ 个坐标表示每一对不同的正交基向量。这些分量可以像向量那样存储在内存中，形式是 k 个数的连续数组。

如果一个质点从某个起点开始有动量 p 和位矢 x，那么质点在此处的角动量可以被定义为双向量:

$$L = x \wedge p$$

通常这一表达式包含叉乘，但只在三维空间中这样定义。在三维空间中，向量“对偶”到双向量 (vectors are “dual”to bivectors)。通俗来讲，在三维空间中与一条直线 (向量) 正交的空间是一个平面 (双向量)。这意味着在三维空间中向量和双向量可以交换使用。这导致了角速度被表示为向量的广泛应用。这些“轴”向量的变换和普通向量的不同。相比之下，双向量是更为自然的表示，因为它们可以利用几何代数直接转换到不同的坐标系中，并使方程在任何维度中都有意义。

N 维空间的转子提供了一个旋转表达式，它是四元数 (三维空间) 和复数 (二维空间) 的替代品，并且在任何维度都适用。原本通过向量 a 和 b 之间的双倍夹角定义的平面内旋转的转子是由向量 a 和 b 的几何乘积来定义的。

$$R = ab = a \cdot b + a \wedge b$$

它有一个标量项和一个双向量项，类似于复数和四元数的实项和虚项。这样的和叫做多向量 (multivector)。几何积可以拓展到多向量，因此也可以拓展到转子本身。就像在四元数中，两个转子的乘积可以产生一个新的转子，新转子的编码是按照两个旋转一个接一个来的。若使用一个转子旋转一个向量，可以用以下公式:

$$x' = Rx\tilde{R}$$

其中 $\tilde{R}$ 表示 R 的逆，这类似于四元数和复数的共轭。转子的乘积产生了多向量，它是 m 维向量的和，其中 m 为偶数且小于等于 n (这形成了一个代数分支：偶次代数)。在三维空间中一个转子仍然可以只用一个标量部分和一个双向量部分 (有三个分量) 来表示，即两个转子的乘积仍然得出一个转子，这个转子可以用一个单独的旋转平面表示。但在四维空间中，一个物体可以同时绕两个独立的旋转平面旋转：一个普通的转子可以表示为一个标量部分，一个双向量部分 (有六个分量) 和一个四维向量 (只有一个分量)。因此，我将一个四维转子作为一个连续的 8 位数数组存储在内存中。

刚体上一点 r 的瞬时速度，在这一刚体的参考系中，是：$v + r \cdot \omega$ 这里使用的点积是一个泛化的多矢点积，有时也被称为左收缩 (left contraction)(⌋)

角动量通过双向量到双向量间的线性映射

$$L(\omega) = \int_V r \wedge (r \cdot \omega) \mathrm{d}V$$

与角速度相关。Doran 和 Lasenby [2003] 用此对物体的三维物体的体积进行了积分。

如果物体被转子 R 旋转，那么它的角动量可以表示为在物体的初始局部坐标系中计算一次的与时间无关的线性映射 $\mathcal{I}$ 的形式：

$$L(\omega) = R\mathcal{I}(\omega)\tilde{R} \tag{1}$$

在三维空间中，向量是对偶到双向量的，所以 I 从双向量到双向量的映射可以转化成一个从向量到向量的映射，具体过程是这些向量先对偶到双向量，应用这个映射，然后再次对偶返回到向量。这个映射可以用一个矩阵来表示：惯性张量。

将方程 (1) 对时间求微分，注意只有转子是与依赖于时间的，那么就可以给出熟悉的欧拉方程的形式：

$$L_t(\omega) = \mathcal{I}(\omega_t) - \omega \times \mathcal{I}(\omega) = \tau \tag{2}$$

其中, $\times$ 不是普通的叉乘，而是交换子乘积（commutator product）$A \times B = \frac{1}{2}(AB - BA)$，这里使用了两个双向量 A 和 B。这个方程中的量以物体的参照系 (即与物体一起旋转的参照系) 来表示。

4 n 维的推广

将惯性张量表示为从向量到向量的映射仅适用于 3D，因此我不采用对偶。

为了表示这个双向量映射，我首先定义一个 $k \times n$ 的矩阵 $[r]_\bigstar$，例如这样

$$r \wedge a = [r]_\bigstar a \tag{3}$$

对两个向量 r 和向量 a 成立。它是叉积矩阵 $[r]_\times$ 的推广，使得 $r \times a = [r]_{\times a}$，但我在这里没有采用对偶：此变换的值域是一个视为 k 维向量的双向量。另外注意到

$$r \cdot \omega = [r]_\bigstar^T \omega$$

这里这个矩阵变换的域也是一个双向量，并且这个双向量仍视为一个 k 维向量。

使用这些定义，我定义了一个 $k \times k$ 矩阵 I，该矩阵推广了惯性张量：

$$\begin{aligned}
\mathcal{I}(\omega) &= \int_V [r]_\bigstar [r]_\bigstar^T \mathrm{d}V \omega \\
&= \int_V \Delta I \mathrm{d}V \omega \\
&= I\omega
\end{aligned}$$

积分也取自 n 维空间，而不是通常的三维空间。

我在附录 A 中给出了 2 维、3 维和 4 维中的 $[r]_\star$ 和 ΔI。请注意，它们是按照字典序给出的的双向量基础元素 (例如：e_{xy}，e_{xz}，e_{yz})，但这个顺序选择是任意的。

惯性张量的范围和域是双向量，并按原样进行变换。因此，我定义一个 $k \times k$ 矩阵 $[R]_2$，使得对于被视为向量的双向量，有 $[R]_2 = RB\tilde{R}$。然后，

$$I' = [R]_2 I [R]_2^T \tag{4}$$

或者就映射本身而言：$R\mathcal{I}(\tilde{R}\omega R)\tilde{R} = I\omega$。这使我可以在全局惯性参考系中表达方程式 (2)：

$$L_t(\omega) = I'\omega_t - \omega \times I'\omega = \tau \tag{5}$$

转子 ($R_t = -\frac{1}{2}\omega R$) 对时间的导数的公式类似于并且因此可以替代将在 3D 系统中使用的四元数的公式。我的实现使用了基于 Guendelman 等人的 (半隐式欧拉) 时间步进方案 [2003]，但是可以使用其他时间步长方案。我将等式 (5) 的陀螺项积分分别使用隐式欧拉方法 [Catto 2015]。

随着角速度的积分，代表物体轴向的小误差会累积在转子 R 中。在三个维度上，当使用四元数代替转子时，通过在每个时间步之后对四元数进行归一化来校正效果。但是，这在 4 维及更高版本中是不够的：虽然由两个单位向量的几何积构成的简单转子确实位于 $\mathbb{R}^n$ 内的 $(k+1)$-维球面上，但对应于两次旋转（或更多旋转）的转子却不是这种情况。为了校正小误差，我使用 Perwass [2009] 的算法对转子的几何乘积进行因子分解，以获得一组向量，这些向量的乘积给出了原始转子。该算法返回归一化的向量，从而确保从其乘积重新形成的转子将代表正确的旋转。特别地，陀螺仪项经常产生两次旋转，这不能简单地通过归一化来校正。

5 任意 n 维单纯形网格的惯性张量

一个 n 维单纯形网格是由多个 (n−1) 维的单纯形网格构建的，例如：一个 3 维单纯形网格是由多个三角形构成的，一个 4 维单纯形网格是由多个四面体构成的，依此类推。

由于惯性张量的线性性，要计算任意维网格的惯性张量，只要求出由每个 $(n-1)$-单纯形和参考系原点构成的 n-单纯形的惯性张量之和就足够了。

通过求惯性积分的乘积，可以直接从顶点计算任意单纯形的惯性张量：

$$P_{jk} = \int_V jk\rho \mathrm{d}V$$

其中 j 和 k 是坐标轴。这些积分可以用高斯定理来简化 [Dobrovolskis 1996]。然而，将 Blow 和 Binstock [2004] 的方法进行推广被证明是更简单的。他们首先计算物体的协方差矩阵，然后将协方差矩阵转换为惯性张量。给定一个将标准单纯形的顶点转化为任意单纯形顶点的矩阵 M，任意单纯形的协方差矩阵 C' 与标准单纯形的协方差矩阵 C 相关，使得 $C' = det(M)MCM^T$。标准单纯形的协方差矩阵有两个值得注意的项 C_{xx} 和 C_{xy}。它们是与坐标轴对齐的

单纯形超体积上的 x^2 和 xy 的积分。

$$C_{xx} = \frac{2}{(n+2)!},$$

$$C_{xy} = \frac{1}{(n+2)!}.$$

为了将协方差矩阵转化为 n 维惯性张量，我从 Trenkler 的研究 [2001] 中得到启发，并在坐标系中展开了 $[r]_\star$:

$$[r]_\star = \sum_{i=0}^{n-1} r_i [e_i]_\star$$

惯性张量可以由协方差矩阵 C 导出:

$$\begin{aligned}[r]_\star [r]_\star^T &= (\sum_{i=0}^{n-1} r_i [e_i]_\star)(\sum_{j=0}^{n-1} r_j [e_j]_\star^T) \\ &= \sum_{i,j}^{n-1} r_i r_j [e_i]_\star [e_j]_\star^T\end{aligned}$$

还需要计算 n 维单纯形的质量，可以通过它的密度和体积 $\frac{1}{n!} det\,(M)$ 计算得出。

选择物体的局部坐标系使得惯性张量 I 是对角的是一种普遍而有意义的做法。传统的矩阵对角化算法会产生一个旋转矩阵，但是这里的 $(k \times k)$ 矩阵仅适用于二重向量 (像等式 4 中所述) 而不是普通向量，因此传统方法对此没有帮助。但是我们注意到如果 $n \times n$ 的协方差矩阵如果是对角矩阵，那么惯性张量也会是对角的。因此在计算惯性张量之前，我首先使用能让协方差矩阵对角化的旋转矩阵来旋转物体。

6 碰撞解决

利用公式 3 中所述的矩阵，我推广了 Guendelman 等人 [2003] 提出的碰撞问题处理方法，包括静摩擦和动摩擦，但该方案的通用性使其足以应用于其它方法。我也使用了他们的接触图和激波传播方案。

碰撞后，我对每个物体施加相等或相反的脉冲 j，以获得速度 $v_0 = v \pm j/m$ 和角速度 $\omega_0 = \omega \pm I^{-1}(r \wedge j) = \omega \pm I^{-1}[r]_\star j$，其中 r 分别从它们的质心指向碰撞位置。碰撞点处的新速度为 $u_0 = u \pm Kj$，其中 $K = \delta/m + [r]_\star^T I^{-1} [r]_\star$ 且 δ 为 $n \times n$ 的单位矩阵。

使用此公式，静摩擦和动摩擦可以用与三维时相同的方式表示。

7 碰撞检测

7.1 多面体或圆柱的超球面

为了确定 $n-1$ 维超球面是否与 n 维多面体或者 n 维圆柱发生碰撞，我推广了基于 Minkowski 差分的球体/凸面物体碰撞检测算法 (例如：[Ericson 2004])。该算法在物体表面上找到距离超球体中心最近的点，并检查它是否小于

一单位的超球体半径。注意最近点可能位于 $n-1$ 维的单元上，或位于与一个或多个单元共享的单元边界上——边界的大小取决于共享边界的单元数 (例如在 4D 中，两个三维单元共享一个面，三个单元共享一个边，等等)。利用向量 a 到 n 维向量 B 所表示的子空间上的投影 $a_{||} = (a \cdot B)/B$ 来寻找这个最近的点。

7.2 两个任意的凸多面体

对于两个任意凸多面体的情形，我将分离轴定理碰撞检测方法 [Gottschalk 1996] 推广到了 n 维。一个 n 维多面体 a 包含一定数量的 m 维单元 ($m < n$)。而每个 m 维单元 i 的空间又由 m 个向量张成，这 m 个向量的外积记为 $V_a^m(i)$。给定两个 n 维多面体 a 和 b，对于各自的单元满足 $m_a + m_b = n-1$，产生的所有外积：

$$V_a^{m_a}(i) \wedge V_b^{m_b}(j) \qquad \forall i, j.$$

这个 $n-1$ 维向量的对偶向量可能是分离轴。注意这也包括了外积为空的情形，比如 $m_a = n-1$ 而 $m_b = 0$。这些对偶向量就是每个多面体 $n-1$ 维单元 (三维空间中的面) 的法线。

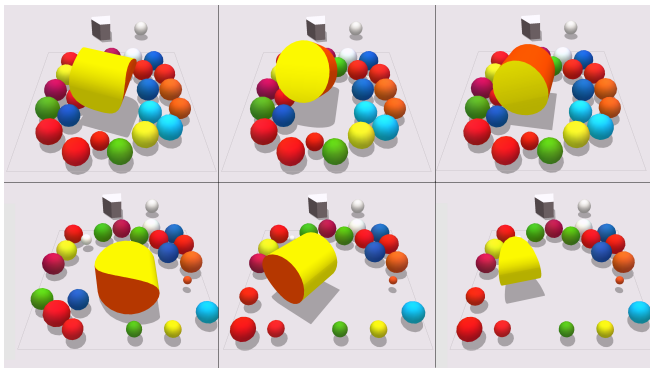


图 2: 用户在超球面的一层上滚动一个圆柱体。其中一些部分看起来是悬浮的，因为它们在第四维空间中而超出了我们的视野。

在三维空间中这个公式等价于将边的方向向量的叉积 (即两个三维向量的外积的对偶) 作为潜在的分离轴，这对应于边和边的碰撞。在四维空间中，则需要取多面体 a 的所有边的方向向量和多面体 b 的面的二重向量的外积的对偶，反之也是如此。这对应于一维边和二维面碰撞的情形。一维边和边的情形也被这种方法包含，类似于三维空间的边与边碰撞的方法也包含了顶点的情形一样 [Ericson 2004]。

我通过将 Eberly [2002] 的方法拓展到 n 维的情形，实现了定向盒 (Oriented Boxes) 的具体情况。所有的标量三重积都被外积所代替。一个四维的超立方体有 8 个单元，24 个面，32 条边和 16 个顶点，但是由于它是对称的，它只有 4 个唯一的向量 (法线或边的方向)，只能形成 6 个唯一的双向量。因此，需要检查来自单元法线的 4×2 个向量和来自外积的 $4 \times 6 \times 2 = 48$ 个向量。

8 交互式模拟与显示

我展示了一个交互式四维对象的三维切片。这些四维对象具有三维的表面，由四维网格来表示 (在本例中为四面体)。在任何时候都只能看到单个三维表面和四维对象之间的交集。这是通过对每个对象的三维表面进行切片以获得

要显示的二维平面来完成的，其方式类似于 Chu 等人 [2009] 的方法，但本文只取了一个切面。对每个四面体进行切片会生成一个二维多边形 (三角形或四边形)，所有这些多边形一起构成了要显示的二维平面。

用户可以使用鼠标、触摸屏或虚拟现实控制器与对象进行交互。它们只能在当前三维切片的范围内点击、拖动和旋转对象，但显然四维对象并不局限于该空间，因此可能会消失在视图之外。在这种情况下，用户可以拖动滑块沿第四维的轴移动三维切片以再次找到它们。我决定使用这种显示和交互方法是因为它可以让用户在他们熟悉的三维环境中观察和使用四维对象，同时还允许他们在对象消失时以非常简单的方式找到它们。

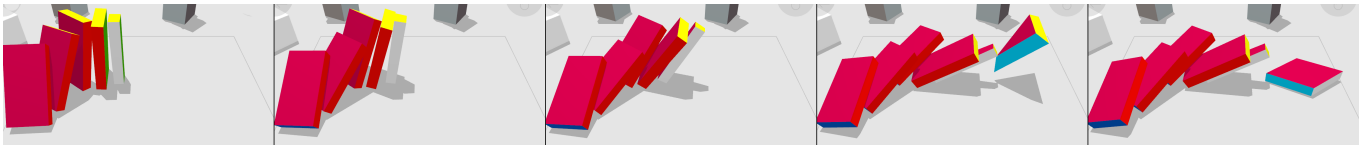


图 3: 一名用户推倒了一排超立方多米诺骨牌。最后面的多米诺骨牌一开始不在视野内，在第四维上有偏移。

为了达到模拟器的目的，重力被平凡地推广到了 n 维：它沿法向量指向地面。空气摩擦的形式与 3D 时的情形相同。

在手机之类的触摸屏设备上，通常配备有加速度传感器 (accelerometer)。该加速度传感器用于控制重力方向。初始状态时，重力方向仅限于可见的 3D 切片。可以按下按钮以使重力向量沿第四维度略微旋转。

一些 VR 控制器支持触觉反馈。当用户持有的 4D 对象撞击另一个 4D 对象时，程序会使 VR 控制器轻微振动。

用户可以看到的任何位置都对应于 4D 中的位置，因此模拟器可以在这些位置上施加力或脉冲，以移动对象，使对象跟随用户的运动。要旋转对象，只需找到 2D 旋转平面即可形成双向量 (bivector) 以获得扭矩 (torque)。在将一个四维帧 (frame) 旋转到另一个帧的最一般情况下，我首先找到一个执行此转换的转子 (rotor)，然后提取其双向量 (bivector) 部分。为了找到该转子，用连续的转子，将转子相对应的坐标轴对相互转换。

在随附的视频以及图 1、2 和 3 中可以看到 4D 情形和用户交互的示例。还可以使用商业软件 [ten Bosch 2017]。

9 4D 贾尼别科夫 (DZHANIBEKOV) 效应

在本节中，我将简要介绍无扭矩条件下的 4D 欧拉方程 (5)。在 3D 中，此方程式的结果是贾尼别科夫 (Dzhanibekov) 效应，该效应最早由 Poinsot [1851] 描述。给定一个具有三个惯性矩的物体，每个惯性矩对应于一个特定的旋转平面。具有最小和最大惯性矩的平面中的旋转对于角动量的较小扰动是稳定的。对于中间平面中的旋转，较小的角动量扰动会放大为较大的周期性旋转。

在 4D 中，方程的 ω_{ij} 分量为：

$$I_{ij}\dot{\omega}_{ij} = (I_{jk}-I_{ik})\omega_{ik}\omega_{jk} + (I_{jm}-I_{im})\omega_{im}\omega_{jm}$$

其中 k 和 m 是除 i 和 j 之外的其他两个索引。请注意，此等式中存在一个加和——这说明围绕每个平面的 4D 旋转与 3D 的情况不同，不能仅仅基于具有不同主惯性矩的物体去声明旋转的稳定性。

平面 (ij) 的方程式取决于除与其垂直的平面 (km) 以外的所有其他旋转平面 (ik, jk, im, jm)。这是可以预料

到的，因为 4D 对象可以在两个垂直平面中独立旋转。

如果初始角速度位于 3D 子空间内 (例如，对某个特定的 c，$\omega_{ic} = \omega_{ci} = 0$ 成立)，则方程式简化为 3D 情况。图 5c 显示了绕中间平面旋转时通常的不稳定旋转。长时间内角动量的小幅下降是因为隐式积分法。

双重旋转似乎是稳定的：图 5a 显示了围绕 (xy) 平面的旋转和围绕其他平面的小扰动是如何引入双重旋转 (zw) 的。惯性矩似乎对行为没有影响。对于大小为 $(\frac{1}{2}, 1, \frac{3}{2}, 2)$，$(\frac{1}{2}, \frac{1}{2}, 3, \frac{1}{2})$ 等的长方体，会生成类似的图形。即使围绕双平面的旋转是独立的 (仅扰动 (zw) 平面不会影响 (xy) 平面)，这些方程也会通过其他平面耦合，从而产生了效果。

图 5b 显示出了从图 5a 的初始状态消除扰动的效果。即使有其他震荡，但系统仍会处于稳定状态。

图 5d 和 4 显示了向图 5c 的初始状态添加附加扰动的效果。效果与之前的情况相似。

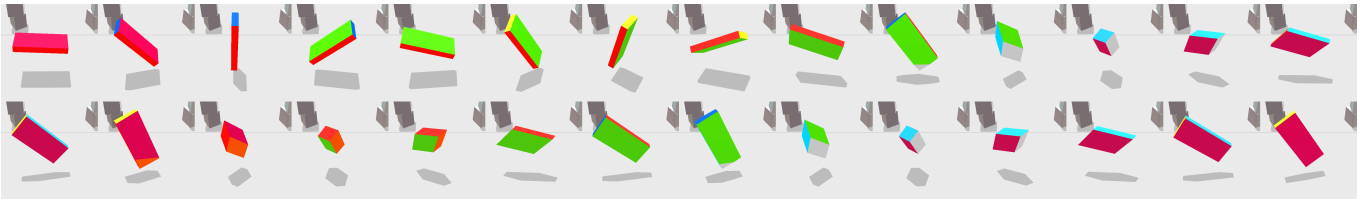
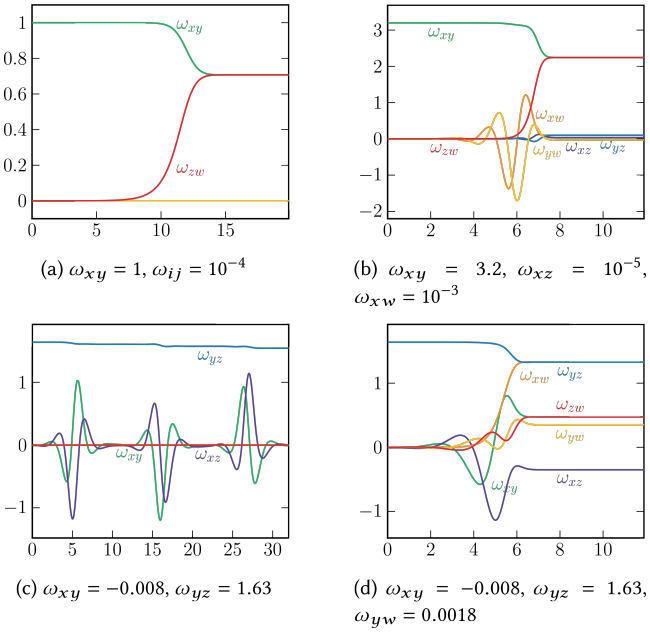


图 4: 在陀螺力作用下旋转的一个超长方体。



(a) $\omega_{xy} = 1, \omega_{ij} = 10^{-4}$

(b) $\omega_{xy} = 3.2, \omega_{xz} = 10^{-5}, \omega_{xw} = 10^{-3}$

(c) $\omega_{xy} = -0.008, \omega_{yz} = 1.63$

(d) $\omega_{xy} = -0.008, \omega_{yz} = 1.63, \omega_{yw} = 0.0018$

图 5: 一个尺寸为 (1.1, 0.3, 2.5, 0.7) 的 4D 长方体在不同初始条件下角速度随时间 (以秒计) 的变化。

10 未来的工作

刚体动力学的公式可以被扩展到非欧几里得空间，比如双曲空间。其他有关碰撞检测、非刚体以及其他力和约束也可以被推广至更高维度。关于交互方法，让用户在他们的 3D 可见的切片之外操作 4D 物体也会很有趣。

参考文献

[1] E.A. Abbott. 1884. *Flatland: A Romance of Many Dimensions.* Seeley & Co.

[2] T. Bancho. 1990. *Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions.* Scientific American Library. https://books.google.com/books?id=8n55QgAACAAJ

[3] Jan Bender, Kenny Erleben, and Je Trinkle. 2014. Interactive simulation of rigid body dynamics in computer graphics. *Computer Graphics Forum (Print)* 33, 1 (2014), 246–270. https://doi.org/10.1111/cgf.12272

[4] Jonathan Blow and Atman J Binstock. 2004. How to find the inertia tensor (or other mass properties) of a 3D solid body represented by a triangle mesh.(2004). http://number-none.com/blow/inertia/

[5] Stephen Cameron. 1990. Collision detection by four-dimensional intersection testing. *Robotics and Automation, IEEE Transactions* on 6 (07 1990),291–302. https://doi.org/10.1109/70.56661

[6] Erin Catto. 2015. Physics for Game Programmers: Numerical Methods. (2015). https://www.gdcvault.com/play/1022197/Physics-for-Game-Programmers-Numerical Presentation at Game Developers Conference 2015.

[7] Arthur Cayley. 1846. Sur quelques propriétés des déterminants gauches. *Journal für die reine und angewandte Mathematik* 1846, 32 (1846), 119–123.

[8] Alan Chu, Chi-Wing Fu, Andrew J. Hanson, and Pheng-Ann Heng. 2009. GL4D: A GPU-based Architecture for Interactive 4D Visualization. *IEEE Transactions on Visualization and Computer Graphics (IEEE Visualization 2009)* 15, 6 (Nov.-Dec. 2009), 1587–1594.

[9] Anthony R Dobrovolskis. 1996. Inertia of any polyhedron. *Icarus* 124, 2 (1996), 698 – 704. Chris Doran and Anthony Lasenby. 2003. Geometric algebra for physicists. Cambridge University Press.

[10] L. Dorst, D. Fontijne, and S. Mann. 2009. *Geometric Algebra for Computer Science: An Object-oriented Approach to Geometry.* Elsevier. https://books.google.com/books?id=yaEqlAEACAAJ

[11] David Eberly. 2002. Dynamic collision detection using oriented bounding boxes. *Geometric Tools, Inc* (2002).

[12] Christer Ericson. 2004. *Real-Time Collision Detection.* CRC Press, Inc., USA.

[13] Stefan Gottschalk. 1996. Separating axis theorem, Technical Report TR96-024, Department of Computer Science, UNC Chapel Hill. (1996).

[14] Eran Guendelman, Robert Bridson, and Ronald Fedkiw. 2003. Nonconvex rigid bodies with stacking. In *ACM SIGGRAPH 2003 Papers* (San Diego, California) *(SIGGRAPH'03)*. ACM, ACM, New York, NY, USA, 871–878. https://doi.org/10.1145/1201775.882358

[15] Charles Gunn. 2011. *Geometry, Kinematics, and Rigid Body Mechanics in Cayley-Klein Geometries.* Doctoral Thesis. Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften, Berlin. https://doi.org/10.14279/depositonce-3058

[16] Charles G. Gunn and Steven De Keninck. 2019. Geometric Algebra and Computer Graphics. In ACM SIGGRAPH 2019 Courses (Los Angeles, California) *(SIGGRAPH'19)*. Association for Computing Machinery, New York, NY, USA, Article 12, 140 pages. https://doi.org/10.1145/3305366.3328099

[17] D. Hilbert and S. Cohn-Vossen. 1952. *Anschauliche Geometrie.* Chelsea Publishing Company. https://books.google.com/books?id=d6sBd9h1HbMC

[18] A. Macdonald. 2011. *Linear and Geometric Algebra.* Alan Macdonald. https://books.google.com/books?id=oxhJYgEACAAJ

[19] Christian Perwass. 2009. *Geometric Algebra with Applications in Engineering* (1st ed.). Springer Publishing Company, Incorporated.

[20] Louis Poinsot. 1851. *Théorie nouvelle de la rotation des corps.* Bachelier.

[21] Andrew J Sinclair and John E Hurtado. 2005. Cayley kinematics and the Cayley form of dynamic equations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 461, 2055 (2005), 761–781.

[22] Marc ten Bosch. 2017. 4D Toys. https://4dtoys.com

[23] Götz Trenkler. 2001. The vector cross product from an algebraic point of view. *Discuss.Math. - General Algebra and Applications* 21, 1 (2001), 67–82.

[24] Xiaoqi Yan, Chi-Wing Fu, and Andrew J. Hanson. 2012. Multitouching the Fourth Dimension. *Computer* 45, 9 (2012), 80–88. https://doi.org/10.1109/MC.2012.77

[25] Hui Zhang and Andrew J. Hanson. 2006. *Physically Interacting with Four Dimensions.* Springer Berlin Heidelberg, Berlin, Heidelberg, 232–242. https://doi.org/10.1007/11919476_24

A 双向量算子及惯性张量矩阵

2D 情况:

$$[r]_\star = (-y \quad x)$$

$$\Delta I = (x^2 + y^2)$$

3D 情况:

$$[r]_\star = \begin{pmatrix} -y & x & 0 \\ -z & 0 & x \\ 0 & -z & y \end{pmatrix}$$

$$\Delta I = \begin{pmatrix} x^2+y^2 & yz & -xz \\ yz & x^2+z^2 & xy \\ -xz & xy & y^2+z^2 \end{pmatrix}$$

4D 情况:

$$[r]_\star = \begin{pmatrix} -y & x & 0 & 0 \\ -z & 0 & x & 0 \\ -w & 0 & 0 & x \\ 0 & -z & y & 0 \\ 0 & -w & 0 & y \\ 0 & 0 & -w & z \end{pmatrix}$$

$$\Delta I = \begin{pmatrix} x^2+y^2 & yz & yw & -xz & -xw & 0 \\ yz & x^2+z^2 & zw & xy & 0 & -xw \\ yw & zw & w^2+x^2 & 0 & xy & xz \\ -xz & xy & 0 & y^2+z^2 & zw & -yw \\ -xw & 0 & xy & zw & w^2+y^2 & yz \\ 0 & -wx & xz & -yw & yz & w^2+z^2 \end{pmatrix}$$