

第 4 讲

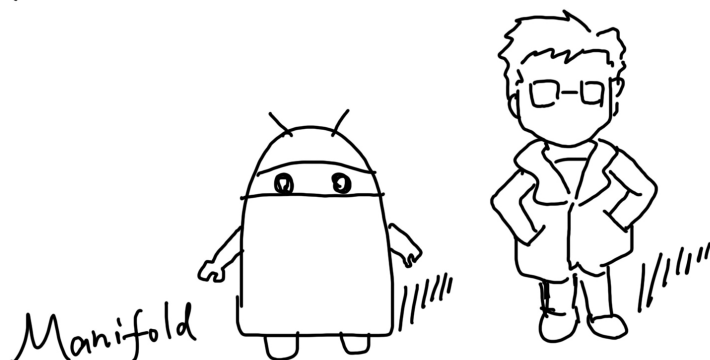
李群与李代数

本节目标

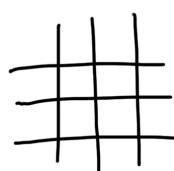
1. 理解李群与李代数的概念，掌握 $SO(3)$, $SE(3)$ 与对应李代数的表示方式。
2. 理解 BCH 近似的意义。
3. 学会在李代数上的扰动模型。
4. 使用 Sophus 对李代数进行运算。

上一讲，我们介绍了三维世界中刚体运动的描述方式，包括旋转矩阵、旋转向量、欧拉角、四元数等若干种方式。我们重点介绍了旋转的表示，但是在 SLAM 中，除了表示之外，我们还要对它们进行估计和优化。因为在 SLAM 中位姿是未知的，而我们需要解决什么样的相机位姿最符合当前观测数据这样的问题。一种典型的方式是把它构建成一个优化问题，求解最优的 \mathbf{R}, \mathbf{t} ，使得误差最小化。

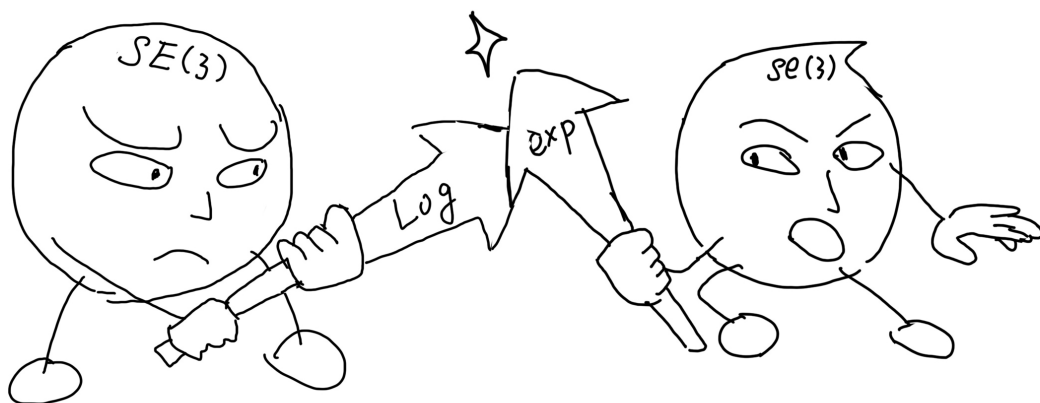
如前所言，旋转矩阵自身是带有约束的（正交且行列式为 1）。它们作为优化变量时，会引入额外的约束，使优化变得困难。通过李群——李代数间的转换关系，我们希望把位姿估计变成无约束的优化问题，简化求解方式。由于读者可能还没有李群李代数的基本知识，我们将从最基本的开始讲起。



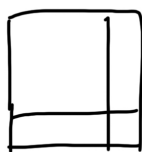
Manifold



Tangent Space



$$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \in SE(3)$$

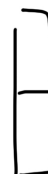


4x4

$$\log(T)^\vee = \zeta$$

$$\exp(\zeta^\wedge) = T$$

$$\zeta = [p, \phi]^T \in se(3)$$



6x1



4.1 李群李代数基础

上一讲，我们介绍了旋转矩阵和变换矩阵的定义。当时，我们说三维旋转矩阵构成了特殊正交群 $SO(3)$ ，而变换矩阵构成了特殊欧氏群 $SE(3)$ ：

$$SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} | \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1\}. \quad (4.1)$$

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} | \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}. \quad (4.2)$$

不过，当时我们并未详细解释群的含义。细心的读者会注意到，旋转矩阵也好，变换矩阵也好，它们对加法是不封闭的。换句话说，对于任意两个旋转矩阵 $\mathbf{R}_1, \mathbf{R}_2$ ，它们按照矩阵加法的定义，和不再是一个旋转矩阵：

$$\mathbf{R}_1 + \mathbf{R}_2 \notin SO(3). \quad (4.3)$$

对于变换矩阵亦是如此。我们发现，这两种矩阵并没有良好定义的加法，相对的，它们只有一种较好的运算：乘法。 $SO(3)$ 和 $SE(3)$ 关于乘法是封闭的：

$$\mathbf{R}_1 \mathbf{R}_2 \in SO(3), \quad \mathbf{T}_1 \mathbf{T}_2 \in SE(3). \quad (4.4)$$

我们知道乘法对应着旋转或变换的复合——两个旋转矩阵相乘表示做了两次旋转。对于这种只有一个运算的集合，我们把它叫做群。

4.1.1 群

群 (Group) 是一种集合加上一种运算的代数结构。我们把集合记作 A ，运算记作 \cdot ，那么群可以记作 $G = (A, \cdot)$ 。群要求这个运算满足以下几个条件：

1. 封闭性: $\forall a_1, a_2 \in A, \quad a_1 \cdot a_2 \in A.$
2. 结合律: $\forall a_1, a_2, a_3 \in A, \quad (a_1 \cdot a_2) \cdot a_3 = a_1 \cdot (a_2 \cdot a_3).$
3. 幺元: $\exists a_0 \in A, \quad s.t. \quad \forall a \in A, \quad a_0 \cdot a = a \cdot a_0 = a.$
4. 逆: $\forall a \in A, \quad \exists a^{-1} \in A, \quad s.t. \quad a \cdot a^{-1} = a_0.$

读者可以记作“封结么逆”^①。我们可以验证，旋转矩阵集合和矩阵乘法构成群，同样

^①谐音风姐咬你。

变换矩阵和矩阵乘法也构成群（因此才能称它们为旋转矩阵群和变换矩阵群）。其他常见的群包括整数的加法 $(\mathbb{Z}, +)$ ，去掉 0 后的有理数的乘法（幺元为 1） $(\mathbb{Q} \setminus 0, \cdot)$ 等等。矩阵中常见的群有：

一般线性群 $GL(n)$ 指 $n \times n$ 的可逆矩阵，它们对矩阵乘法成群。

特殊正交群 $SO(n)$ 也就是所谓的旋转矩阵群，其中 $SO(2)$ 和 $SO(3)$ 最为常见。

特殊欧氏群 $SE(n)$ 也就是前面提到的 n 维欧氏变换，如 $SE(2)$ 和 $SE(3)$ 。

群结构保证了在群上的运算具有良好的性质，而群论则是研究群的各种结构和性质的理论，但我们在此不多加介绍。感兴趣的读者可以参考任意一本近世代数教材。

李群是指具有连续（光滑）性质的群。像整数群 \mathbb{Z} 那样离散的群没有连续性质，所以不是李群。而 $SO(n)$ 和 $SE(n)$ ，它们在实数空间上是连续的。我们能够直观地想象一个刚体能够连续地在空间中运动，所以它们都是李群。由于 $SO(3)$ 和 $SE(3)$ 对于相机姿态估计尤其重要，我们主要讨论这两个李群。如果读者对李群的理论性质感兴趣，请参照 [20]。

下面，我们先从较简单的 $SO(3)$ 开始讨论，我们将发现每个李群都有对应的李代数。我们首先引出 $SO(3)$ 上面的李代数 $\mathfrak{so}(3)$ 。

4.1.2 李代数的引出

考虑任意旋转矩阵 \mathbf{R} ，我们知道它满足：

$$\mathbf{R}\mathbf{R}^T = \mathbf{I}. \quad (4.5)$$

现在，我们说， \mathbf{R} 是某个相机的旋转，它会随时间连续地变化，即为时间的函数： $\mathbf{R}(t)$ 。由于它仍是旋转矩阵，有

$$\mathbf{R}(t)\mathbf{R}(t)^T = \mathbf{I}.$$

在等式两边对时间求导，得到：

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T + \mathbf{R}(t)\dot{\mathbf{R}}(t)^T = 0.$$

整理得：

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T = -\left(\dot{\mathbf{R}}(t)\mathbf{R}(t)^T\right)^T. \quad (4.6)$$

可以看出 $\dot{\mathbf{R}}(t)\mathbf{R}(t)^T$ 是一个反对称矩阵。回忆之前，我们在式 (3.3) 介绍叉积时，引入了 \wedge 符号，将一个向量变成了反对称矩阵。同理，对于任意反对称矩阵，我们亦能找到一个与之对应的向量。把这个运算用符号 \vee 表示：

$$\mathbf{a}^\wedge = \mathbf{A} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \quad \mathbf{A}^\vee = \mathbf{a}. \quad (4.7)$$

于是, 由于 $\dot{\mathbf{R}}(t)\mathbf{R}(t)^T$ 是一个反对称矩阵, 我们可以找到一个三维向量 $\phi(t) \in \mathbb{R}^3$ 与之对应。于是有:

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T = \phi(t)^\wedge.$$

等式两边右乘 $\mathbf{R}(t)$, 由于 \mathbf{R} 为正交阵, 有:

$$\dot{\mathbf{R}}(t) = \phi(t)^\wedge \mathbf{R}(t) = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \mathbf{R}(t). \quad (4.8)$$

可以看到, 每对旋转矩阵求一次导数, 只需左乘一个 $\phi^\wedge(t)$ 矩阵即可。为方便讨论, 我们设 $t_0 = 0$, 并设此时旋转矩阵为 $\mathbf{R}(0) = \mathbf{I}$ 。按照导数定义, 可以把 $\mathbf{R}(t)$ 在 0 附近进行一阶泰勒展开:

$$\begin{aligned} \mathbf{R}(t) &\approx \mathbf{R}(t_0) + \dot{\mathbf{R}}(t_0)(t - t_0) \\ &= \mathbf{I} + \phi(t_0)^\wedge(t). \end{aligned} \quad (4.9)$$

我们看到 ϕ 反映了 \mathbf{R} 的导数性质, 故称它在 $SO(3)$ 原点附近的正切空间 (Tangent Space) 上。同时在 t_0 附近, 设 ϕ 保持为常数 $\phi(t_0) = \phi_0$ 。那么根据式 (4.8), 有

$$\dot{\mathbf{R}}(t) = \phi(t_0)^\wedge \mathbf{R}(t) = \phi_0^\wedge \mathbf{R}(t).$$

上式是一个关于 \mathbf{R} 的微分方程, 而且我们知道初始值 $\mathbf{R}(0) = \mathbf{I}$, 解之, 得:

$$\mathbf{R}(t) = \exp(\phi_0^\wedge t). \quad (4.10)$$

读者可以验证上式对微分方程和初始值均成立。不过, 由于做了一定的假设, 所以它只在 $t = 0$ 附近有效。我们看到, 旋转矩阵 \mathbf{R} 与另一个反对称矩阵 ϕ_0 通过指数关系发生了联系。也就是说, 当我们知道某个时刻的 \mathbf{R} 时, 存在一个向量 ϕ , 它们满足这个矩阵指数关系。但是矩阵的指数是什么呢? 这里我们有两个问题需要澄清:

1. 如果上式成立, 那么给定某时刻的 \mathbf{R} , 我们就能求得一个 ϕ , 它描述了 \mathbf{R} 在局部的导数关系。与 \mathbf{R} 对应的 ϕ 有什么含义呢? 后面会看到, ϕ 正是对应到 $SO(3)$ 上的李代数 $\mathfrak{so}(3)$;
2. 其次, 矩阵指数 $\exp(\phi^\wedge)$ 如何计算?——事实上, 这正是李群与李代数间的指数/对数映射。

下面我们一一加以介绍。

4.1.3 李代数的定义

每个李群都有与之对应的李代数。李代数描述了李群的局部性质。通用的李代数的定义如下:

李代数由一个集合 \mathbb{V} , 一个数域 \mathbb{F} 和一个二元运算 $[\cdot, \cdot]$ 组成。如果它们满足以下几条性质, 称 $(\mathbb{V}, \mathbb{F}, [\cdot, \cdot])$ 为一个李代数, 记作 \mathfrak{g} 。

1. 封闭性 $\forall \mathbf{X}, \mathbf{Y} \in \mathbb{V}, [\mathbf{X}, \mathbf{Y}] \in \mathbb{V}$.
2. 双线性 $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{V}, a, b \in \mathbb{F}$, 有:

$$[a\mathbf{X} + b\mathbf{Y}, \mathbf{Z}] = a[\mathbf{X}, \mathbf{Z}] + b[\mathbf{Y}, \mathbf{Z}], \quad [\mathbf{Z}, a\mathbf{X} + b\mathbf{Y}] = a[\mathbf{Z}, \mathbf{X}] + b[\mathbf{Z}, \mathbf{Y}].$$

3. 自反性^① $\forall \mathbf{X} \in \mathbb{V}, [\mathbf{X}, \mathbf{X}] = \mathbf{0}$.
4. 雅可比等价 $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{V}, [\mathbf{X}, [\mathbf{Y}, \mathbf{Z}]] + [\mathbf{Z}, [\mathbf{Y}, \mathbf{X}]] + [\mathbf{Y}, [\mathbf{Z}, \mathbf{X}]] = \mathbf{0}$.

其中二元运算被称为**李括号**。从表面上来看, 李代数所需要的性质还是挺多的。相比于群中的较为简单的二元运算, 李括号表达了两个元素的差异。它不要求结合律, 而要求元素和自己做李括号之后为零的性质。作为例子, 三维向量 \mathbb{R}^3 上定义的叉积 \times 是一种李括号, 因此 $\mathfrak{g} = (\mathbb{R}^3, \mathbb{R}, \times)$ 构成了一个李代数。读者可以尝试将叉积的性质代入到上面四条性质中。

4.1.4 李代数 $\mathfrak{so}(3)$

下面我们说, 之前提到的 ϕ , 事实上是一种李代数。 $SO(3)$ 对应的李代数是定义在 \mathbb{R}^3 上的向量, 我们记作 ϕ 。根据前面的推导, 每个 ϕ 都可以生成一个反对称矩阵:

^①自反性是指自己与自己的运算为零。

$$\Phi = \phi^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 3}. \quad (4.11)$$

在此定义下，两个向量 ϕ_1, ϕ_2 的李括号为：

$$[\phi_1, \phi_2] = (\Phi_1 \Phi_2 - \Phi_2 \Phi_1)^\vee. \quad (4.12)$$

读者可以去验证该定义下的李括号满足上面的几条性质。由于 ϕ 与反对称矩阵关系很紧密，在不引起歧义的情况下，就说 $\mathfrak{so}(3)$ 的元素是 3 维向量或者 3 维反对称矩阵，不加区别：

$$\mathfrak{so}(3) = \{\phi \in \mathbb{R}^3, \Phi = \phi^\wedge \in \mathbb{R}^{3 \times 3}\}. \quad (4.13)$$

至此，我们已清楚了 $\mathfrak{so}(3)$ 的内容。它们是一个由三维向量组成的集合，每个向量对应到一个反对称矩阵，可以表达旋转矩阵的导数。它与 $SO(3)$ 的关系由指数映射给定：

$$R = \exp(\phi^\wedge). \quad (4.14)$$

指数映射会在稍后介绍。由于已经介绍了 $\mathfrak{so}(3)$ ，我们顺带先来看 $SE(3)$ 上对应的李代数。

4.1.5 李代数 $\mathfrak{se}(3)$

对于 $SE(3)$ ，它也有对应的李代数 $\mathfrak{se}(3)$ 。为省略篇幅，我们就不描述如何引出 $\mathfrak{se}(3)$ 了。与 $\mathfrak{so}(3)$ 相似， $\mathfrak{se}(3)$ 位于 \mathbb{R}^6 空间中：

$$\mathfrak{se}(3) = \left\{ \xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6, \rho \in \mathbb{R}^3, \phi \in \mathfrak{so}(3), \xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \right\}. \quad (4.15)$$

我们把每个 $\mathfrak{se}(3)$ 元素记作 ξ ，它是一个六维向量。前三维为平移，记作 ρ ；后三维为旋转，记作 ϕ ，实质上是 $\mathfrak{so}(3)$ 元素^①。同时，我们拓展了 $^\wedge$ 符号的含义。在 $\mathfrak{se}(3)$ 中，同样使用 $^\wedge$ 符号，将一个六维向量转换成四维矩阵，但这里不再表示反对称：

^①请注意有些地方把旋转放前面，平移放后面，也是可行的。

$$\xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}. \quad (4.16)$$

我们仍使用 \wedge 和 \vee 符号来指代“从向量到矩阵”和“从矩阵到向量”的关系，以保持和 $\mathfrak{so}(3)$ 上的一致性。读者可以简单地把 $\mathfrak{se}(3)$ 理解成“由一个平移加上一个 $\mathfrak{so}(3)$ 元素构成的向量”（尽管这里的 ρ 还不直接是平移）。同样，李代数 $\mathfrak{se}(3)$ 亦有类似于 $\mathfrak{so}(3)$ 的李括号：

$$[\xi_1, \xi_2] = (\xi_1^\wedge \xi_2^\wedge - \xi_2^\wedge \xi_1^\wedge)^\vee. \quad (4.17)$$

读者可以验证它满足李代数的定义（留作习题）。至此我们已经见过两种重要的李代数 $\mathfrak{so}(3)$ 和 $\mathfrak{se}(3)$ 了。

4.2 指数与对数映射

4.2.1 $SO(3)$ 上的指数映射

现在来考虑第二个问题： $\exp(\phi^\wedge)$ 是如何计算的？它是一个矩阵的指数，在李群和李代数中，称为指数映射（Exponential Map）。同样，我们会先讨论 $\mathfrak{so}(3)$ 的指数映射，再讨论 $\mathfrak{se}(3)$ 的情形。

任意矩阵的指数映射可以写成一个泰勒展开，但是只有在收敛的情况下才会有结果，其结果仍是一个矩阵。

$$\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{A}^n. \quad (4.18)$$

同样地，对 $\mathfrak{so}(3)$ 中任意一元素 ϕ ，我们亦可按此方式定义它的指数映射：

$$\exp(\phi^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\phi^\wedge)^n. \quad (4.19)$$

我们来仔细推导一下这个定义。由于 ϕ 是三维向量，我们可以定义它的模长和它的方向，分别记作 θ 和 \mathbf{a} ，于是有 $\phi = \theta \mathbf{a}$ 。这里 \mathbf{a} 是一个长度为 1 的方向向量。首先，对于 \mathbf{a}^\wedge ，有以下两条性质：

$$\mathbf{a}^\wedge \mathbf{a}^\wedge = \mathbf{a} \mathbf{a}^T - \mathbf{I}, \quad (4.20)$$

以及

$$\mathbf{a}^\wedge \mathbf{a}^\wedge \mathbf{a}^\wedge = -\mathbf{a}^\wedge. \quad (4.21)$$

读者可以自行验证上述性质。它们提供了处理 \mathbf{a}^\wedge 高阶项的方法。利用这两个性质，我

们可以把指数映射写成:

$$\begin{aligned}
 \exp(\phi^\wedge) &= \exp(\theta \mathbf{a}^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\theta \mathbf{a}^\wedge)^n \\
 &= \mathbf{I} + \theta \mathbf{a}^\wedge + \frac{1}{2!} \theta^2 \mathbf{a}^\wedge \mathbf{a}^\wedge + \frac{1}{3!} \theta^3 \mathbf{a}^\wedge \mathbf{a}^\wedge \mathbf{a}^\wedge + \frac{1}{4!} \theta^4 (\mathbf{a}^\wedge)^4 + \dots \\
 &= \mathbf{a} \mathbf{a}^T - \mathbf{a}^\wedge \mathbf{a}^\wedge + \theta \mathbf{a}^\wedge + \frac{1}{2!} \theta^2 \mathbf{a}^\wedge \mathbf{a}^\wedge - \frac{1}{3!} \theta^3 \mathbf{a}^\wedge - \frac{1}{4!} \theta^4 (\mathbf{a}^\wedge)^2 + \dots \\
 &= \mathbf{a} \mathbf{a}^T + \left(\theta - \frac{1}{3!} \theta^3 + \frac{1}{5!} \theta^5 - \dots \right) \mathbf{a}^\wedge - \left(1 - \frac{1}{2!} \theta^2 + \frac{1}{4!} \theta^4 - \dots \right) \mathbf{a}^\wedge \mathbf{a}^\wedge \\
 &= \mathbf{a}^\wedge \mathbf{a}^\wedge + \mathbf{I} + \sin \theta \mathbf{a}^\wedge - \cos \theta \mathbf{a}^\wedge \mathbf{a}^\wedge \\
 &= (1 - \cos \theta) \mathbf{a}^\wedge \mathbf{a}^\wedge + \mathbf{I} + \sin \theta \mathbf{a}^\wedge \\
 &= \cos \theta \mathbf{I} + (1 - \cos \theta) \mathbf{a} \mathbf{a}^T + \sin \theta \mathbf{a}^\wedge.
 \end{aligned}$$

最后我们得到了一个似曾相识的式子:

$$\exp(\theta \mathbf{a}^\wedge) = \cos \theta \mathbf{I} + (1 - \cos \theta) \mathbf{a} \mathbf{a}^T + \sin \theta \mathbf{a}^\wedge. \quad (4.22)$$

回忆前一讲内容, 它和罗德里格斯公式, 即式 (3.14) 如出一辙。这表明, $\mathfrak{so}(3)$ 实际上就是由所谓的**旋转向量**组成的空间, 而指数映射即罗德里格斯公式。通过它们, 我们把 $\mathfrak{so}(3)$ 中任意一个向量对应到了一个位于 $SO(3)$ 中的旋转矩阵。反之, 如果定义对数映射, 我们也能把 $SO(3)$ 中的元素对应到 $\mathfrak{so}(3)$ 中:

$$\phi = \ln(\mathbf{R})^\vee = \left(\sum_{n=0}^{\infty} \frac{(-1)^n}{n+1} (\mathbf{R} - \mathbf{I})^{n+1} \right)^\vee. \quad (4.23)$$

不过我们通常不按照泰勒展开去计算对数映射。在第 3 讲中, 我们已经介绍过如何根据旋转矩阵计算对应的李代数, 即使用式 (3.16), 利用迹的性质分别求解转角和转轴, 采用那种方式更加省事一些。

现在, 我们介绍了指数映射的计算方法。读者可能会问, 指数映射性质如何呢? 是否对于任意的 \mathbf{R} 都能找到一个唯一的 ϕ ? 很遗憾, 指数映射只是一个满射。这意味着每个 $SO(3)$ 中的元素, 都可以找到一个 $\mathfrak{so}(3)$ 元素与之对应; 但是可能存在多个 $\mathfrak{so}(3)$ 中的元素, 对应到同一个 $SO(3)$ 。至少对于旋转角 θ , 我们知道多转 360° 度和没有转是一样的——它具有周期性。但是, 如果我们把旋转角度固定在 $\pm\pi$ 之间, 那么李群和李代数元素是一一对应的。

$SO(3)$ 与 $\mathfrak{so}(3)$ 的结论似乎在我们意料之中。它和我们前面讲的旋转向量与旋转矩阵很相似，而指数映射即是罗德里格斯公式。旋转矩阵的导数可以由旋转向量指定，指导着如何在旋转矩阵中进行微积分运算。

4.2.2 $SE(3)$ 上的指数映射

下面我们来介绍 $\mathfrak{se}(3)$ 上的指数映射。为了节省篇幅，我们不再像 $\mathfrak{so}(3)$ 那样详细推导指数映射。 $\mathfrak{se}(3)$ 上的指数映射形式如下：

$$\exp(\xi^\wedge) = \begin{bmatrix} \sum_{n=0}^{\infty} \frac{1}{n!} (\phi^\wedge)^n & \sum_{n=0}^{\infty} \frac{1}{(n+1)!} (\phi^\wedge)^n \rho \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (4.24)$$

$$\triangleq \begin{bmatrix} \mathbf{R} & \mathbf{J}\rho \\ \mathbf{0}^T & 1 \end{bmatrix} = \mathbf{T}. \quad (4.25)$$

如果你有耐心，可以照着 $\mathfrak{so}(3)$ 上的做法推导，把 \exp 进行泰勒展开推导此式。从结果上看， ξ 的指数映射左上角的 \mathbf{R} 是我们熟知的 $SO(3)$ 中的元素，与 $\mathfrak{se}(3)$ 当中的旋转部分 ϕ 对应。而右上角的 \mathbf{J} 则可整理为（设 $\phi = \theta \mathbf{a}$ ）：

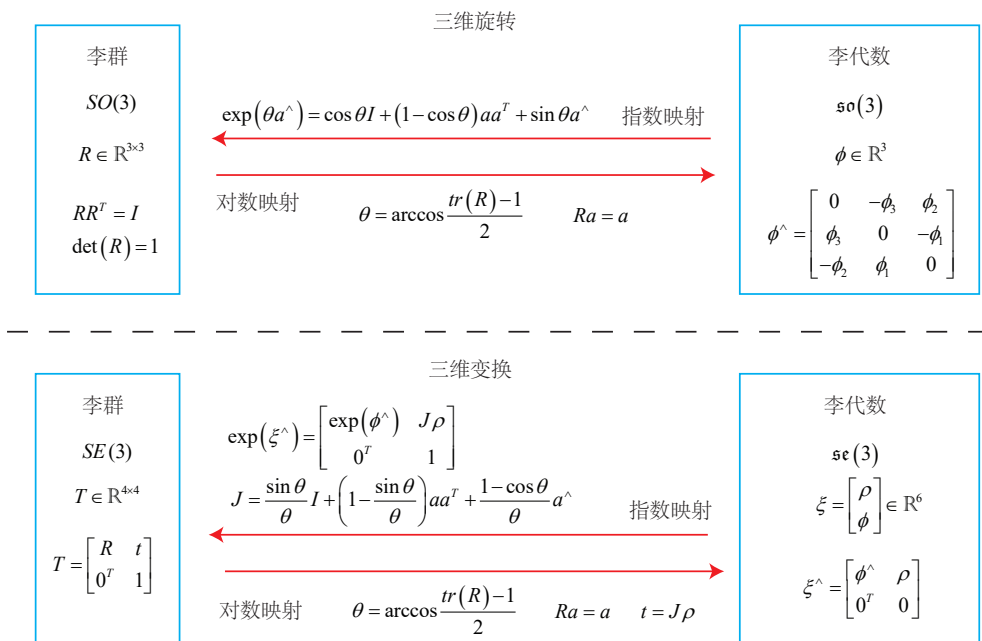
$$\mathbf{J} = \frac{\sin \theta}{\theta} \mathbf{I} + \left(1 - \frac{\sin \theta}{\theta}\right) \mathbf{a} \mathbf{a}^T + \frac{1 - \cos \theta}{\theta} \mathbf{a}^\wedge. \quad (4.26)$$

该式与罗德里格斯有些相似，但不完全一样。我们看到，平移部分经过指数映射之后，发生了一次以 \mathbf{J} 为系数矩阵的线性变换。请读者重视这里的 \mathbf{J} ，因为我们后面还要用到它。

同样的，虽然我们也可以类比推得对数映射，不过根据变换矩阵 \mathbf{T} 求 $\mathfrak{se}(3)$ 上的对应向量也有更省事的方式：从左上的 \mathbf{R} 计算旋转向量，而右上的 \mathbf{t} 满足：

$$\mathbf{t} = \mathbf{J}\rho. \quad (4.27)$$

由于 \mathbf{J} 可以由 ϕ 得到，所以这里的 ρ 亦可由此线性方程解得。现在，我们已经理清了李群、李代数的定义与相互的转换关系，总结如图 4-1 所示。如果读者有哪里不明白，可以翻回去几页看看公式推导。

图 4-1 $SO(3), SE(3), \mathfrak{so}(3), \mathfrak{se}(3)$ 的对应关系。

4.3 李代数求导与扰动模型

4.3.1 BCH 公式与近似形式

使用李代数的一大动机是为了进行优化，而在优化过程中导数是非常必要的信息（我们会在第六讲详细介绍）。下面我们来考虑一个问题。虽然我们已经清楚了 $SO(3)$ 和 $SE(3)$ 上的李群与李代数关系，但是，当我们在 $SO(3)$ 中完成两个矩阵乘法时，李代数中 $\mathfrak{so}(3)$ 上发生了什么改变呢？反过来说，当 $\mathfrak{so}(3)$ 上做两个李代数的加法时， $SO(3)$ 上是否对应着两个矩阵的乘积？如果成立的话，相当于：

$$\exp(\phi_1^\wedge) \exp(\phi_2^\wedge) = \exp((\phi_1 + \phi_2)^\wedge).$$

如果 ϕ_1, ϕ_2 为标量，那显然该式成立；但此处我们计算的是矩阵的指数函数，而非标量的指数。换言之，我们在研究下式是否成立：

$$\ln(\exp(\mathbf{A})\exp(\mathbf{B})) = \mathbf{A} + \mathbf{B} ?$$

很遗憾，该式在矩阵时并不成立。

两个李代数指数映射乘积的完整形式，由 Baker-Campbell-Hausdorff 公式（BCH 公式）^①给出。由于它完整的形式较复杂，我们给出它展开式的前几项：

$$\ln(\exp(\mathbf{A})\exp(\mathbf{B})) = \mathbf{A} + \mathbf{B} + \frac{1}{2}[\mathbf{A}, \mathbf{B}] + \frac{1}{12}[\mathbf{A}, [\mathbf{A}, \mathbf{B}]] - \frac{1}{12}[\mathbf{B}, [\mathbf{A}, \mathbf{B}]] + \cdots \quad (4.28)$$

其中 $[\]$ 为李括号。BCH 公式告诉我们，当处理两个矩阵指数之积时，它们会产生一些由李括号组成的余项。特别地，考虑 $SO(3)$ 上的李代数 $\ln(\exp(\phi_1^\wedge)\exp(\phi_2^\wedge))^\vee$ ，当 ϕ_1 或 ϕ_2 为小量时，小量二次以上的项都可以被忽略掉。此时，BCH 拥有线性近似表达：

$$\ln(\exp(\phi_1^\wedge)\exp(\phi_2^\wedge))^\vee \approx \begin{cases} \mathbf{J}_l(\phi_2)^{-1}\phi_1 + \phi_2 & \text{if } \phi_1 \text{ is small,} \\ \mathbf{J}_r(\phi_1)^{-1}\phi_2 + \phi_1 & \text{if } \phi_2 \text{ is small.} \end{cases} \quad (4.29)$$

以第一个近似为例。该式告诉我们，当对一个旋转矩阵 \mathbf{R}_2 （李代数为 ϕ_2 ）左乘一个微小旋转矩阵 \mathbf{R}_1 （李代数为 ϕ_1 ）时，可以近似地看作，在原有的李代数 ϕ_2 上，加上了一项 $\mathbf{J}_l(\phi_2)^{-1}\phi_1$ 。同理，第二个近似描述了右乘一个微小位移的情况。于是，李代数在 BCH 近似下，分成了左乘近似和右乘近似两种，在使用时我们须加注意，使用的是左乘模型还是右乘模型。

本书以左乘为例。左乘 BCH 近似雅可比 \mathbf{J}_l 事实上就是式（4.26）的内容：

$$\mathbf{J}_l = \mathbf{J} = \frac{\sin \theta}{\theta} \mathbf{I} + \left(1 - \frac{\sin \theta}{\theta}\right) \mathbf{a}\mathbf{a}^T + \frac{1 - \cos \theta}{\theta} \mathbf{a}^\wedge. \quad (4.30)$$

它的逆为：

$$\mathbf{J}_l^{-1} = \frac{\theta}{2} \cot \frac{\theta}{2} \mathbf{I} + \left(1 - \frac{\theta}{2} \cot \frac{\theta}{2}\right) \mathbf{a}\mathbf{a}^T - \frac{\theta}{2} \mathbf{a}^\wedge. \quad (4.31)$$

而右乘雅可比仅需要对自变量取负号即可：

$$\mathbf{J}_r(\phi) = \mathbf{J}_l(-\phi). \quad (4.32)$$

这样，我们就可以谈论李群乘法与李代数加法的关系了。为了方便读者理解，我们重新叙述一下 BCH 近似的意义。

^① 参见 https://en.wikipedia.org/wiki/Baker-Campbell-Hausdorff_formula。

假定对某个旋转 \mathbf{R} ，对应的李代数为 ϕ 。我们给它左乘一个微小旋转，记作 $\Delta\mathbf{R}$ ，对应的李代数为 $\Delta\phi$ 。那么，在李群上，得到的结果就是 $\Delta\mathbf{R} \cdot \mathbf{R}$ ，而在李代数上，根据 BCH 近似，为： $\mathbf{J}_l^{-1}(\phi)\Delta\phi + \phi$ 。合并起来，可以简单地写成：

$$\exp(\Delta\phi^\wedge) \exp(\phi^\wedge) = \exp\left((\phi + \mathbf{J}_l^{-1}(\phi)\Delta\phi)^\wedge\right). \quad (4.33)$$

反之，如果我们在李代数上进行加法，让一个 ϕ 加上 $\Delta\phi$ ，那么可以近似为李群上带左右雅可比的乘法：

$$\exp((\phi + \Delta\phi)^\wedge) = \exp((\mathbf{J}_l\Delta\phi)^\wedge) \exp(\phi^\wedge) = \exp(\phi^\wedge) \exp((\mathbf{J}_r\Delta\phi)^\wedge). \quad (4.34)$$

这将为之后李代数上的做微积分提供了理论基础。同样的，对于 $SE(3)$ ，亦有类似的 BCH 近似公式：

$$\exp(\Delta\xi^\wedge) \exp(\xi^\wedge) \approx \exp\left((\mathcal{J}_l^{-1}\Delta\xi + \xi)^\wedge\right), \quad (4.35)$$

$$\exp(\xi^\wedge) \exp(\Delta\xi^\wedge) \approx \exp\left((\mathcal{J}_r^{-1}\Delta\xi + \xi)^\wedge\right). \quad (4.36)$$

这里 \mathcal{J}_l 形式比较复杂，它是一个 6×6 的矩阵，读者可以参考 [6] 中式 (7.82) 和 (7.83) 内容。由于我们在计算中不用到该雅可比，故这里略去它的实际形式。

4.3.2 $SO(3)$ 李代数上的求导

下面我们来讨论一个带有李代数的函数，如何关于该李代数求导的问题。该问题有很强的实际背景。在 SLAM 中，我们要估计一个相机的位置和姿态，该位姿是由 $SO(3)$ 上的旋转矩阵或 $SE(3)$ 上的变换矩阵描述的。不妨设某个时刻小萝卜的位姿为 \mathbf{T} 。它观察到了一个世界坐标位于 \mathbf{p} 的点，产生了一个观测数据 \mathbf{z} 。那么，由坐标变换关系知：

$$\mathbf{z} = \mathbf{T}\mathbf{p} + \mathbf{w}. \quad (4.37)$$

然而，由于观测噪声 \mathbf{w} 的存在， \mathbf{z} 往往不可能精确地满足 $\mathbf{z} = \mathbf{T}\mathbf{p}$ 的关系。所以，我们通常会计算理想的观测与实际数据的误差：

$$\mathbf{e} = \mathbf{z} - \mathbf{T}\mathbf{p}. \quad (4.38)$$

假设一共有 N 个这样的路标点和观测，于是就有 N 个上式。那么，对小萝卜的位姿

估计，相当于是寻找一个最优的 \mathbf{T} ，使得整体误差最小化：

$$\min_{\mathbf{T}} J(\mathbf{T}) = \sum_{i=1}^N \|\mathbf{z}_i - \mathbf{T}\mathbf{p}_i\|_2^2. \quad (4.39)$$

求解此问题，需要计算目标函数 J 关于变换矩阵 \mathbf{T} 的导数。我们把具体的算法留到后面再讲。这里重点要说的是，我们经常会构建与位姿有关的函数，然后讨论该函数关于位姿的导数，以调整当前的估计值。然而， $SO(3), SE(3)$ 上并没有良好定义的加法，它们只是群。如果我们把 \mathbf{T} 当成一个普通矩阵来处理优化，那就必须对它加以约束。而从李代数角度来说，由于李代数由向量组成，具有良好的加法运算。因此，使用李代数解决求导问题的思路分为两种：

1. 用李代数表示姿态，然后对根据李代数加法来对李代数求导。
2. 对李群左乘或右乘微小扰动，然后对该扰动求导，称为左扰动和右扰动模型。

第一种方式对应到李代数的求导模型，而第二种则对应到扰动模型。下面我们来讨论这两种思路的异同。

4.3.3 李代数求导

首先，考虑 $SO(3)$ 上的情况。假设我们对一个空间点 \mathbf{p} 进行了旋转，得到了 $\mathbf{R}\mathbf{p}$ 。现在，要计算旋转之后点的坐标相对于旋转的导数，我们不严谨地记为^①：

$$\frac{\partial (\mathbf{R}\mathbf{p})}{\partial \mathbf{R}}.$$

由于 $SO(3)$ 没有加法，所以该导数无法按照导数的定义进行计算。设 \mathbf{R} 对应的李代数为 ϕ ，我们转而计算：

$$\frac{\partial (\exp(\phi^\wedge) \mathbf{p})}{\partial \phi}.$$

^①请注意这里并不能按照矩阵微分来定义导数，这只是一个记号。

按照导数的定义，有：

$$\begin{aligned}
 \frac{\partial (\exp(\phi^\wedge) \mathbf{p})}{\partial \phi} &= \lim_{\delta \phi \rightarrow 0} \frac{\exp((\phi + \delta \phi)^\wedge) \mathbf{p} - \exp(\phi^\wedge) \mathbf{p}}{\delta \phi} \\
 &= \lim_{\delta \phi \rightarrow 0} \frac{\exp((\mathbf{J}_l \delta \phi)^\wedge) \exp(\phi^\wedge) \mathbf{p} - \exp(\phi^\wedge) \mathbf{p}}{\delta \phi} \\
 &\approx \lim_{\delta \phi \rightarrow 0} \frac{(\mathbf{I} + (\mathbf{J}_l \delta \phi)^\wedge) \exp(\phi^\wedge) \mathbf{p} - \exp(\phi^\wedge) \mathbf{p}}{\delta \phi} \\
 &= \lim_{\delta \phi \rightarrow 0} \frac{(\mathbf{J}_l \delta \phi)^\wedge \exp(\phi^\wedge) \mathbf{p}}{\delta \phi} \\
 &= \lim_{\delta \phi \rightarrow 0} \frac{-(\exp(\phi^\wedge) \mathbf{p})^\wedge \mathbf{J}_l \delta \phi}{\delta \phi} = -(\mathbf{R} \mathbf{p})^\wedge \mathbf{J}_l.
 \end{aligned}$$

第二行的近似为 BCH 线性近似，第三行为泰勒展开舍去高阶项后近似，第四行至第五行将反对称符号看作叉积，交换之后变号。于是，我们推导了旋转后的点相对于李代数的导数：

$$\frac{\partial (\mathbf{R} \mathbf{p})}{\partial \phi} = (-\mathbf{R} \mathbf{p})^\wedge \mathbf{J}_l. \quad (4.40)$$

不过，由于这里仍然含有形式比较复杂的 \mathbf{J}_l ，我们不太希望计算它。而下面要讲的扰动模型则提供了更简单的导数计算方式。

4.3.4 扰动模型（左乘）

另一种求导方式，是对 \mathbf{R} 进行一次扰动 $\Delta \mathbf{R}$ 。这个扰动可以乘在左边也可以乘在右边，最后结果会有点儿微小的差异，我们以左扰动为例。设左扰动 $\Delta \mathbf{R}$ 对应的李代数为 φ 。然后，对 φ 求导，即：

$$\frac{\partial (\mathbf{R} \mathbf{p})}{\partial \varphi} = \lim_{\varphi \rightarrow 0} \frac{\exp(\varphi^\wedge) \exp(\phi^\wedge) \mathbf{p} - \exp(\phi^\wedge) \mathbf{p}}{\varphi}. \quad (4.41)$$

该式的求导比上面更为简单：

$$\begin{aligned}
\frac{\partial(Rp)}{\partial\varphi} &= \lim_{\varphi \rightarrow 0} \frac{\exp(\varphi^\wedge) \exp(\phi^\wedge) p - \exp(\phi^\wedge) p}{\varphi} \\
&\approx \lim_{\varphi \rightarrow 0} \frac{(1 + \varphi^\wedge) \exp(\phi^\wedge) p - \exp(\phi^\wedge) p}{\varphi} \\
&= \lim_{\varphi \rightarrow 0} \frac{\varphi^\wedge Rp}{\varphi} = \lim_{\varphi \rightarrow 0} \frac{-(Rp)^\wedge \varphi}{\varphi} = -(Rp)^\wedge.
\end{aligned}$$

可见, 扰动模型相比于直接对李代数求导, 省去了一个雅可比 J_l 的计算。这使得扰动模型更为实用。请读者务必理解这里的求导运算, 这在位姿估计当中具有重要的意义。

4.3.5 $SE(3)$ 上的李代数求导

最后, 我们给出 $SE(3)$ 上的扰动模型, 而直接李代数上的求导就不再介绍了。假设某空间点 p 经过一次变换 T (对应李代数为 ξ), 得到 Tp ^①。现在, 给 T 左乘一个扰动 $\Delta T = \exp(\delta\xi^\wedge)$, 我们设扰动项的李代数为 $\delta\xi = [\delta\rho, \delta\phi]^T$, 那么:

$$\begin{aligned}
\frac{\partial(Tp)}{\partial\delta\xi} &= \lim_{\delta\xi \rightarrow 0} \frac{\exp(\delta\xi^\wedge) \exp(\xi^\wedge) p - \exp(\xi^\wedge) p}{\delta\xi} \\
&\approx \lim_{\delta\xi \rightarrow 0} \frac{(I + \delta\xi^\wedge) \exp(\xi^\wedge) p - \exp(\xi^\wedge) p}{\delta\xi} \\
&= \lim_{\delta\xi \rightarrow 0} \frac{\delta\xi^\wedge \exp(\xi^\wedge) p}{\delta\xi} \\
&= \lim_{\delta\xi \rightarrow 0} \frac{\begin{bmatrix} \delta\phi^\wedge & \delta\rho \\ \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} Rp + t \\ 1 \end{bmatrix}}{\delta\xi} \\
&= \lim_{\delta\xi \rightarrow 0} \frac{\begin{bmatrix} \delta\phi^\wedge (Rp + t) + \delta\rho \\ 0 \end{bmatrix}}{\delta\xi} = \begin{bmatrix} I & -(Rp + t)^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \triangleq (Tp)^\odot.
\end{aligned}$$

我们把最后的结果定义成一个算符 $^\odot$ ^②, 它把一个齐次坐标的空间点变换成一个 4×6 的矩阵。

^①请注意为了使乘法成立, p 必须使用齐次坐标。

^②我会读作“咚”, 像一个石子掉在井里。

至此，我们已经介绍了李群李代数上的微分运算。之后的章节中，我们将应用这些知识去解决实际问题。关于李群李代数的某些重要数学性质，我们作为习题留给读者。

4.4 实践：Sophus

我们已经介绍了李代数的入门知识，现在是通过实践演练巩固一下所学知识的机会了。我们来讨论如何在程序中操作李代数。在第三讲中，我们看到 Eigen 提供了几何模块，但没有提供李代数的支持。一个较好的李代数库是 Strasdat 维护的 Sophus 库^①。Sophus 库支持本章主要讨论的 $SO(3)$ 和 $SE(3)$ ，此外还含有二维运动 $SO(2)$, $SE(2)$ 以及相似变换 $Sim(3)$ 的内容。它是直接在 Eigen 基础上开发的，我们不需要安装额外的依赖库。读者可以直接从 github 上获取 Sophus^②，或者，在本书的代码目录 slambook/3rdparty 下也提供了 Sophus 源代码。由于历史原因，Sophus 早期版本只提供了双精度的李群/李代数类。后续版本改写成了模板类。模板类的 Sophus 中可以使用不同精度的李群/李代数，但同时增加了使用难度。本书使用非模板的 Sophus 库。如果读者准备使用 github 上的 Sophus，请确保使用的是非模板的版本。你可以输入以下命令获得非模板类的 Sophus：

```
1 git clone https://github.com/strasdat/Sophus.git
2 cd Sophus
3 git checkout a621ff
```

本书的 3rdparty 中提供的 Sophus 也是非模板版本。Sophus 本身亦是一个 cmake 工程。想必你已经了解如何编译 cmake 工程了，我们就不再赘述。Sophus 库只须编译即可，无须安装。

下面我们来演示一下 Sophus 库中的 $SO(3)$ 和 $SE(3)$ 运算：

slambook/ch4/useSophus/useSophus.cpp:

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 #include <Eigen/Core>
6 #include <Eigen/Geometry>
7
8 #include "sophus/so3.h"
9 #include "sophus/se3.h"
10
11 int main( int argc, char** argv )
12 {
13     // 沿 Z 轴转 90 度的旋转矩阵
14     Eigen::Matrix3d R = Eigen::AngleAxisd(M_PI/2, Eigen::Vector3d(0,0,1)).toRotationMatrix();
```

^①最早提出李代数的是 Sophus Lie，这个库就以他的名字命名了。

^②<https://github.com/strasdat/Sophus>

```

15
16 Sophus::S03 S03_R(R); // Sophus::SO(3)可以直接从旋转矩阵构造
17 Sophus::S03 S03_v( 0, 0, M_PI/2 ); // 亦可从旋转向量构造
18 Eigen::Quaterniond q(R); // 或者四元数
19 Sophus::S03 S03_q( q );
20 // 上述表达方式都是等价的
21 // 输出 SO(3) 时, 以 so(3) 形式输出
22 cout<<"SO(3) from matrix: "<<S03_R<<endl;
23 cout<<"SO(3) from vector: "<<S03_v<<endl;
24 cout<<"SO(3) from quaternion :"<<S03_q<<endl;
25
26 // 使用对数映射获得它的李代数
27 Eigen::Vector3d so3 = S03_R.log();
28 cout<<"so3 = "<<so3.transpose()<<endl;
29 // hat 为向量到反对称矩阵
30 cout<<"so3 hat="<<Sophus::S03::hat(so3)<<endl;
31 // 相对的, vee 为反对称到向量
32 cout<<"so3 hat vee= "<<Sophus::S03::vee( Sophus::S03::hat(so3) ).transpose()<<endl;
33
34 // 增量扰动模型的更新
35 Eigen::Vector3d update_so3(1e-4, 0, 0); // 假设更新量有这么多
36 Sophus::S03 S03_updated = Sophus::S03::exp(update_so3)*S03_R; // 左乘更新
37 cout<<"S03 updated = "<<S03_updated<<endl;
38
39 /***** 萌萌的分割线 *****/
40 cout<<"***** 我是分割线 *****"<<endl;
41 // 对 SE(3) 操作大同小异
42 Eigen::Vector3d t(1,0,0); // 沿 X 轴平移1
43 Sophus::SE3 SE3_Rt(R, t); // 从 R, t 构造SE(3)
44 Sophus::SE3 SE3_qt(q,t); // 从 q, t 构造SE(3)
45 cout<<"SE3 from R,t= "<<endl<<SE3_Rt<<endl;
46 cout<<"SE3 from q,t= "<<endl<<SE3_qt<<endl;
47 // 李代数 se(3) 是一个六维向量, 方便起见先 typedef 一下
48 typedef Eigen::Matrix<double,6,1> Vector6d;
49 Vector6d se3 = SE3_Rt.log();
50 cout<<"se3 = "<<se3.transpose()<<endl;
51 // 观察输出, 会发现在 Sophus 中, se(3) 平移在前, 旋转在后。与我们的书是一致的。
52 // 同样的, 有 hat 和 vee 两个算符
53 cout<<"se3 hat = "<<endl<<Sophus::SE3::hat(se3)<<endl;
54 cout<<"se3 hat vee = "<<Sophus::SE3::vee( Sophus::SE3::hat(se3) ).transpose()<<endl;
55
56 // 最后, 演示一下更新
57 Vector6d update_se3; //更新量
58 update_se3.setZero();
59 update_se3(0,0) = 1e-4d;
60 Sophus::SE3 SE3_updated = Sophus::SE3::exp(update_se3)*SE3_Rt;
61 cout<<"SE3 updated = "<<endl<<SE3_updated.matrix()<<endl;
62
63 return 0;
64 }

```

该演示程序为分两部分。前半部分介绍 $SO(3)$ 上的操作，后半部分则为 $SE(3)$ 。我们演示了如何构造 $SO(3)$, $SE(3)$ 对象，对它们进行指数、对数映射，以及当知道更新量后，如何对李群元素进行更新。如果读者切实理解了本章内容，那么这个程序对你来说应该没有什么难度。为了编译它，请在 CMakeLists.txt 里添加以下几行：

slambook/ch4/useSophus/CMakeLists.txt

```
1 # 为使用 sophus，需要使用 find_package 命令找到它
2 find_package( Sophus REQUIRED )
3 include_directories( ${Sophus_INCLUDE_DIRS} )
4
5 add_executable( useSophus useSophus.cpp )
6 target_link_libraries( useSophus ${Sophus_LIBRARIES} )
```

find_package 命令是 cmake 提供的寻找某个库的头文件与库文件的指令。如果 cmake 能够找到它，就会提供头文件和库文件所在的目录的变量。在 Sophus 这个例子中，就是 Sophus_INCLUDE_DIRS 和 Sophus_LIBRARIES 这两个变量。根据它们，我们就能将 Sophus 库引入自己的 cmake 工程了。请读者自行查看此程序的输出信息，它与我们之前的推导是一致的。

4.5 * 相似变换群与李代数

最后，我们要提一下在单目视觉中使用的相似变换群 $Sim(3)$ ，以及对应的李代数 $\mathfrak{sim}(3)$ 。如果你只对双目 SLAM 或 RGBD SLAM 感兴趣，可以跳过本节。

我们已经介绍过单目的尺度不确定性。如果在单目 SLAM 中使用 $SE(3)$ 表示位姿，那么由于尺度不确定性与尺度漂移，整个 SLAM 过程中的尺度会发生变化，这在 $SE(3)$ 中未能体现出来。因此，在单目情况下我们一般会显式地把尺度因子表达出来。用数学语言来说，对于位于空间的点 \mathbf{p} ，在相机坐标系下要经过一个相似变换，而非欧氏变换：

$$\mathbf{p}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p} = s\mathbf{R}\mathbf{p} + \mathbf{t}. \quad (4.42)$$

在相似变换中，我们把尺度 s 表达了出来。它同时作用在 \mathbf{p} 的三个坐标之上，对 \mathbf{p} 进行了一次缩放。与 $SO(3)$ 、 $SE(3)$ 相似，相似变换亦对矩阵乘法构成群，称为相似变换群 $Sim(3)$ ：

$$Sim(3) = \left\{ \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \right\}. \quad (4.43)$$

同样地， $Sim(3)$ 也有对应的李代数、指数映射、对数映射等等。李代数 $\mathfrak{sim}(3)$ 元素

是一个七维向量 ζ 。它的前六维与 $\mathfrak{se}(3)$ 相同，最后多了一项 σ 。

$$\mathfrak{sim}(3) = \left\{ \zeta | \zeta = \begin{bmatrix} \rho \\ \phi \\ \sigma \end{bmatrix} \in \mathbb{R}^7, \zeta^\wedge = \begin{bmatrix} \sigma \mathbf{I} + \phi^\wedge & \rho \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \right\}. \quad (4.44)$$

它比 $\mathfrak{se}(3)$ 多了一项 σ 。关联 $Sim(3)$ 和 $\mathfrak{sim}(3)$ 的仍是指数映射和对数映射。指数映射为：

$$\exp(\zeta^\wedge) = \begin{bmatrix} e^\sigma \exp(\phi^\wedge) & \mathbf{J}_s \rho \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (4.45)$$

其中 \mathbf{J}_s 形式为：

$$\begin{aligned} \mathbf{J}_s = & \frac{e^\sigma - 1}{\sigma} \mathbf{I} + \frac{\sigma e^\sigma \sin \theta + (1 - e^\sigma \cos \theta) \theta}{\sigma^2 + \theta^2} \mathbf{a}^\wedge \\ & + \left(\frac{e^\sigma - 1}{\sigma} - \frac{(e^\sigma \cos \theta - 1) \sigma + (e^\sigma \sin \theta) \theta}{\sigma^2 + \theta^2} \right) \mathbf{a}^\wedge \mathbf{a}^\wedge. \end{aligned}$$

通过指数映射，我们能够找到李代数与李群的关系。对于李代数元素 ζ ，它与李群的对应关系为：

$$s = e^\sigma, \mathbf{R} = \exp(\phi^\wedge), \mathbf{t} = \mathbf{J}_s \rho. \quad (4.46)$$

旋转部分和 $SO(3)$ 是一致的。平移部分，在 $\mathfrak{se}(3)$ 中需要乘一个雅可比 \mathbf{J} ，而相似变换的雅可比更复杂一些。对于尺度因子，可以看到李群中的 s 即为李代数中 σ 的指数函数。

$Sim(3)$ 的 BCH 近似与 $SE(3)$ 是类似的。我们可以讨论一个点 \mathbf{p} 经过相似变换 \mathbf{Sp} 后，相对于 \mathbf{S} 的导数。同样的，存在微分模型和扰动模型两种方式，而扰动模型较为简单。我们省略推导过程，直接给出扰动模型的结果。设给予 \mathbf{Sp} 左侧一个小扰动 $\exp(\zeta^\wedge)$ ，并求 \mathbf{Sp} 对于扰动的导数。因为 \mathbf{Sp} 四维的齐次坐标， ζ 是七维向量，该导数应该是 4×7 的雅可比。为了方便起见，记 \mathbf{Sp} 的前三维组成向量 \mathbf{q} ，那么：

$$\frac{\partial \mathbf{Sp}}{\partial \zeta} = \begin{bmatrix} \mathbf{I} & -\mathbf{q}^\wedge & \mathbf{q} \\ \mathbf{0}^T & \mathbf{0}^T & 0 \end{bmatrix}. \quad (4.47)$$

关于 $Sim(3)$ ，我们就介绍到这里。更详细关于 $Sim(3)$ 的资料，建议读者参见 [21]。

4.6 小结

本讲引入了李群 $SO(3)$ 和 $SE(3)$ ，以及它们对应的李代数 $\mathfrak{so}(3)$ 和 $\mathfrak{se}(3)$ 。我们介绍了位姿在它们上面的表达和转换，然后通过 BCH 的线性近似，我们可以对位姿进行求导和扰动了。这给之后讲解位姿的优化打下了理论基础，因为我们需要经常的对某一个位姿的估计值进行调整，使它对应的误差减小。只有在弄清楚如何对位姿进行调整和更新之后，我们才能继续下一步的内容。

可能本讲的内容比较的偏理论化，毕竟它不像计算机视觉那样经常有好看的图片可以展示。相比于讲解李群李代数的数学书，由于我们只关心实用的内容，所以讲的内容非常精简，速度相对快了一些。请读者务必理解本章内容，它是解决后续许多问题的基础，特别是位姿估计部分。

习题

1. 验证 $SO(3)$ 、 $SE(3)$ 和 $Sim(3)$ 关于乘法成群。
2. 验证 $(\mathbb{R}^3, \mathbb{R}, \times)$ 构成李代数。
3. 验证 $\mathfrak{so}(3)$ 和 $\mathfrak{se}(3)$ 满足李代数要求的性质。
4. 验证性质 (4.20) 和 (4.21)。
5. 证明：

$$Rp^\wedge R^T = (Rp)^\wedge.$$

6. 证明：

$$R \exp(p^\wedge) R^T = \exp((Rp)^\wedge).$$

该式称为 $SO(3)$ 上的伴随性质。同样的，在 $SE(3)$ 上亦有伴随性质：

$$T \exp(\xi^\wedge) T^{-1} = \exp((\text{Ad}(T)\xi)^\wedge), \quad (4.48)$$

其中：

$$\text{Ad}(T) = \begin{bmatrix} R & t^\wedge R \\ 0 & R \end{bmatrix}. \quad (4.49)$$

7. 仿照左扰动的推导，推导 $SO(3)$ 和 $SE(3)$ 在右扰动下的导数。
8. 搜索 cmake 的 find_package 指令是如何运作的。它有哪些可选的参数？为了让 cmake 找到某个库，需要哪些先决条件？