Exercises 06 — Solutions

CMU 15-462/662

1 An Exponentially Better Representation of Rotations

How can you smoothly interpolate rotations? And what does it mean to "average" a bunch of rotations? Many problems in visual computing require you to answer these kinds of questions. For instance, suppose you have many noisy estimates of a camera pose, and want to average these out to get a more reliable estimate. Or suppose you want to interpolate between known coordinate frames at several points on a surface, to get a smoothly-varying frame. These tasks are hard to perform directly because rotation matrices don't form a vector space—for instance, adding two rotation matrices doesn't give you another rotation matrix!

However, we can convert rotation matrices into an axis-angle form where rotations are represented by ordinary vectors. The direction of the vector gives the axis of rotation, and the magnitude of the vector gives the angle of the rotation. These vectors can be added, scaled, and averaged just like normal vectors—and then we can convert back to matrices in order to perform rotations.

$$\underset{\textbf{angle \& axis}}{\boxed{\begin{array}{c}\theta \in \mathbb{R}\\ \mathbf{u} \in \mathbb{R}^n\end{array}}} \;\underset{\log}{\overset{\exp}{\rightleftarrows}}\; \underset{\textbf{rotation matrix}}{\boxed{R \in \mathbb{R}^{n\times n}}}$$

More specifically,

- the **exponential map** can be used to turn an angle θ and unit-length axis u into a rotation matrix R, and
- the **logarithmic map** can be used to turn a rotation matrix R into an axis and angle.

As we'll see, these maps generalize the usual exp and log functions that you know and love[1]. Beyond being very useful for manipulating rotations, they help connect the dots on some of the rotation representations we've already seen in lecture (Euler angles, complex numbers, quaternions, ...).

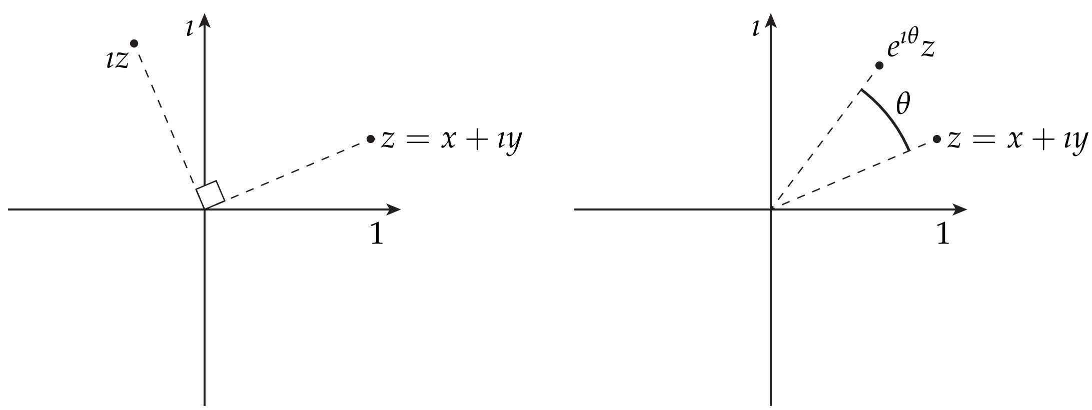
[1] By the way, the super fancy terminology people often use to talk about all this stuff is that the rotation matrices form a *Lie group* $\mathrm{SO}(n)$, and the associated vector space is the *Lie algebra* $\mathfrak{so}(3)$. The exponential map is then a map $\exp : \mathfrak{so}(3) \to \mathrm{SO}(3)$, and the logarithmic map is $\log : \mathrm{SO}(3) \to \mathfrak{so}(3)$. You **definitely won't need to know these terms for this class**—but if you start working a lot with rotations, it's a perspective well worth understanding!

2 Rotations in 2D

Let's start out in 2D, where we already started to see how rotations are connected to the exponential map. In particular, we said that we can rotate a point $z \in \mathbb{C}$ by an angle θ via the map

$$z \mapsto e^{\imath\theta} z.$$

(If you like, you can think of the "axis" in this case as the direction pointing *out of the plane*.) Let's try to understand this idea in a bit more depth.



1. First of all, suppose we have a complex number $z = x + \imath y$. If $\imath$ denotes the imaginary unit, how can we represent the product $\imath z$ as a matrix-vector product between a real 2×2 matrix $J \in \mathbb{R}^{2\times 2}$ (representing $\imath$) and a column vector $[x\ y]^\mathsf{T}$ (respresenting z)?

   Solution.

   In complex arithmetic, we have

   $$\imath z = \imath(x + \imath y) = \imath x + \imath^2 y = -y + \imath x.$$

   In other words, $\imath$ negates y and swaps the two components (corresponding to a quarter-rotation in the counter-clockwise direction). We can encode this operation in matrix form as

   $$\underbrace{\left[\begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array}\right]}_{J} \left[\begin{array}{c} x \\ y \end{array}\right] = \left[\begin{array}{c} -y \\ x \end{array}\right].$$

2. As a sanity check, let's now compute J^2 by taking a matrix-matrix product. What do we get?

   Solution.

   We get

   $$\left[\begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array}\right] \left[\begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array}\right] = \left[\begin{array}{cc} -1 & 0 \\ 0 & -1 \end{array}\right].$$

   In other words, we get $J^2 = -I$, as expected.

3. Now suppose we want to represent the complex product

$$(\cos(\theta) + \imath \sin(\theta))z$$

as a matrix-vector product. Which real 2×2 matrix should we multiply the column vector $[x\ y]^\mathsf{T}$ by? (Hint: start by writing everything in terms of the matrices $I, J \in \mathbb{R}^{2\times 2}$.)

> **Solution.**
>
> We can represent this product as
> $$(\cos(\theta)I + \sin(\theta)J)z,$$
> giving us a matrix
> $$\begin{bmatrix} \cos(\theta) & 0 \\ 0 & \cos(\theta) \end{bmatrix} + \begin{bmatrix} 0 & -\sin(\theta) \\ \sin(\theta) & 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$
> In other words, just the usual 2×2 matrix representation of a rotation by θ.

Taylor series expansions

$\cos(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k}$	$= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$	$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}$	$= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$
$\exp(x) = \sum_{k=0}^{\infty} \frac{1}{k!} x^k$	$= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$	$\log(1-x) = -\sum_{k=1}^{\infty} \frac{1}{k} x^k$	$= -x - \frac{x^2}{2} - \frac{x^2}{3} - \cdots$

From here, we'll finally make sense of the relationship to exponentiation. To do so we'll make use of the *Taylor series*, which expresses a given function $f(x)$ around a fixed point x_0 as

$$f(x) = \sum_{k=0}^{\infty} \frac{1}{k!} f^{(k)}(x_0)(x - x_0)^k,$$

where $f^{(k)}$ denotes the kth derivative of f. In the special case where $x_0 = 0$, this approximation is called a *Mclaurin series*. Several common examples are given above (though in principle you could just derive these yourself by applying the general formula!).

One way to generalize common functions to matrices is to apply the same Taylor series to a matrix $A \in \mathbb{R}^{n\times n}$ rather than a scalar variable $x \in \mathbb{R}$. In this case, the power A^k just means we multiply together k copies of the matrix A.

4. Let $A = \theta J$ for some angle θ and the matrix J derived above[2]. Write out the Taylor series for $\exp(A)$, truncated to the first four terms.

[2]Side note: A is called an *antisymmetric matrix* or *skew-symmetric matrix*, since $A^\mathsf{T} = -A$.

Solution.

The Taylor series for the matrix exponential looks just like the series for the ordinary exponential:

$$\exp(A) = \sum_{k=0}^{\infty} \frac{1}{k!} A^k.$$

The first four terms are therefore

$$I + A + \tfrac{1}{2!}A^2 + \tfrac{1}{3!}A^3.$$

5. Re-write this same truncated series using *Horner's rule*, i.e., factor out multiples of A so that you get a nested expression in terms of just A. For example, if you just had an ordinary polynomial $a + bx + cx^2 + dx^3$, Horner's rule would re-write this polynomial as $a + x(b + x(c + dx))$.

Solution.

Rearranging the expression from the previous solution, we have

$$I + A(I + \tfrac{1}{2}A(I + \tfrac{1}{3}A))$$

6. Now work out the entries of this matrix by performing the additions and multiplications from the "inside out." (Notice that, even for a computer, doing it this way is a *lot* more efficient than repeating all those powers!)

Solution.

The innermost term is

$$I + \tfrac{1}{3}A = \begin{bmatrix} 1 & -\theta/3 \\ \theta/3 & 1 \end{bmatrix},$$

and multiplying this matrix with $A/2$ yields

$$\tfrac{1}{2}A(I + \tfrac{1}{3}A) = \tfrac{1}{2}\begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix}\begin{bmatrix} 1 & -\theta/3 \\ \theta/3 & 1 \end{bmatrix} = \begin{bmatrix} -\theta^2/6 & -\theta/2 \\ \theta/2 & -\theta^2/6 \end{bmatrix}.$$

If we add the identity to this matrix we get

$$I + \tfrac{1}{2}A(I + \tfrac{1}{3}A) = \begin{bmatrix} 1 - \theta^2/6 & -\theta/2 \\ \theta/2 & 1 - \theta^2/6 \end{bmatrix}.$$

Multiplying by A again and adding I yields our final expression

$$I + A(I + \tfrac{1}{2}A(I + \tfrac{1}{3}A)) =$$
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix}\begin{bmatrix} 1 - \theta^2/6 & -\theta/2 \\ \theta/2 & 1 - \theta^2/6 \end{bmatrix} = \begin{bmatrix} 1 - \theta^2/2 & -\theta + \theta^3/6 \\ \theta - \theta^3/6 & 1 - \theta^2/2 \end{bmatrix}.$$

7. Inspect the entries of your solution from the previous question. What does this matrix remind you of? What do you think will happen if you use more terms of the Taylor series?

Solution.

The entries of the final matrix strongly resemble the Taylor series for $\cos(\theta)$ and $\sin(\theta)$. Indeed, if we carry out this calculation further, we'll discover that

$$\exp(\theta J) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

In other words, we recover a real representation of Euler's formula

$$e^{\imath\theta} = \cos(\theta) + \imath\sin(\theta).$$

via 2×2 matrices.

8. For ordinary numbers, the logarithm is the inverse of the exponential and vice-versa. I.e.,

$$\log(e^x) = x \qquad \text{and} \qquad e^{\log(x)} = x.$$

Suppose you are given a matrix

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

and are told to compute $log(R)$, i.e., you are told to find some matrix $A \in \mathbb{R}^{2\times 2}$ such that $\exp(A) = R$. What matrix A would you use? How could you compute the entries of this matrix, given only the four entries of R?

Solution.

From our analysis using the Taylor series, we know that $\exp(\theta J)$ yields the 2×2 rotation matrix for a rotation by θ. So, the logarithm of R must be a matrix of the form θJ, where θ is the rotation described by R. To obtain this angle, we could (for instance) evaluate

$$\theta = \operatorname{atan2}(R_{21}, R_{11}),$$

since R_{21} and R_{11} give the sine and cosine of the rotation angle, respectively. We then have

$$\log(R) = \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix}.$$

9. Is the matrix exponential $\exp(A)$ an injective (i.e., 1-to-1) map? Why or why not? What does this fact tell you about representing rotations as angles θ versus rotation matrices R?

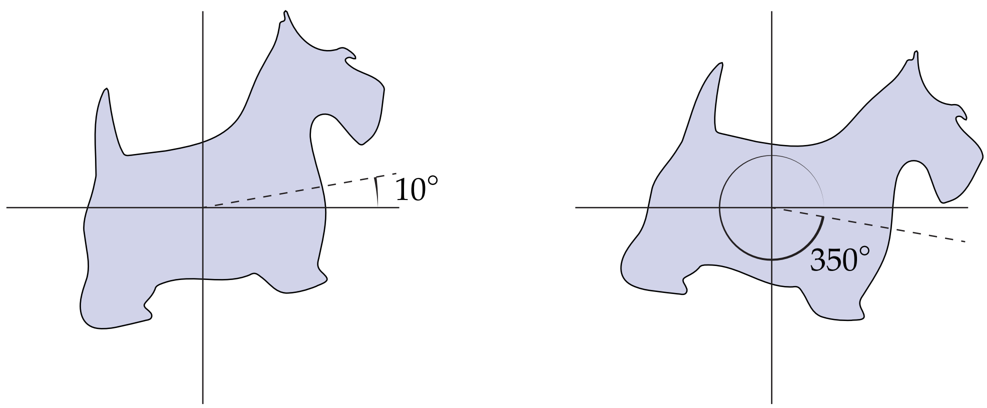
Solution.

No: for instance, $\exp(\theta J)$ and $\exp((\theta + 2\pi)J)$ will produce the same rotation matrix. In the latter case, you can imagine that we make one full "additional turn," but the actual entries of the final matrix will be no different.
In this sense, an angle θ encodes strictly more information than a 2×2 rotation matrix: it tells you not only how where you end up after performing the rotation, but also how many full turns you took to get there.

10. In light of the previous question, is the solution you gave for $\log(R)$ unique? Or are there other matrices A such that $\exp(A) = R$? If the solution is not unique, what is special about the matrix you computed in the previous part?

> **Solution.**
>
> No: as noted in the previous solution, since sin and cos are both 2π-periodic functions, any matrix of the form
> $$A = (\theta + 2k\pi)J, \quad k \in \mathbb{Z}$$
> will satisfy the relationship $\exp(A) = R$. For this reason one always adopts the convention that $\log(R)$ yields the *smallest* angle θ describing the given rotation matrix (or more properly: the 2×2 skew-symmetric matrix containing the smallest angle θ). Computing $\log(R)$ as in the previous solution will give exactly this smallest rotation, since atan2 yields values in $[-\pi/2, \pi/2]$.

At this point we start to see how exponentiation is related to rotation. But why is this approach *useful*? Let's start by playing around with interpolation of rotations.



11. Suppose that in 2D we start out with Scotty at an angle $\theta_0 = 10^\circ$ to the horizontal, and want to rotate him to a pose where he's making an angle 350° to the horizontal over a time interval $t \in [0, 1]$. What's a very simple way to write down a family of rotation matrices $R(t)$ that performs this animation? Do *not* use the exp/log map for this problem.

> **Solution.**
>
> An easy way to do this is to first interpolate the angle, via
> $$\theta(t) = (1-t)\theta_0 + t\theta_1,$$
> then plug this angle in to the usual expression for a 2D rotation matrix to get
> $$R(t) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

12. Wouldn't it be much easier to just average the rotation matrices themselves? Give one example where interpolating the matrices directly turns out *very badly*.

Solution.

Consider the rotation matrices $R_0 = I$ and $R_1 = -I$. Linear interpolation of these matrices would then look like

$$R(t) = (1-t)I + t(-I) = I - 2tI,$$

which means that, for instance $R(1/2) = 0$, i.e., we get a transformation that squashes all of space to a point. In general the issue is that the sum of two rotation matrices is not a rotation matrix, and motion interpolated this way will squash and stretch figures in an undesirable way.

13. If you think of each rotation R_θ as a point $(\cos\theta, \sin\theta)$ on the unit circle, does the family of matrices $R(t)$ from your previous solution describe the shortest path of rotations between the inital and final poses? If not, how might we get a shorter path? This time you can and should use the exp/log maps! (Hint: which rotation takes you directly from the initial pose to the final pose?)

Solution.

No, $R(t)$ defined this way will not give the smallest rotation, since it starts at 10° and goes counter-clockwise almost all the way around the circle to reach 350°. Instead, suppose we let $R_0 \in \mathbb{R}^{2\times 2}$ be the rotation by θ_0, and $R_1 \in \mathbb{R}^{2\times 2}$ be the rotation by θ_1. Then the rotation taking us directly from the initial pose to the final pose is

$$R_1 R_0^{-1},$$

i.e., first reverse the rotation R_0, then apply the rotation R_1. The smallest rotation from R_0 to R_1 is obtained via the log map:

$$A = \log(R_1 R_0^{-1}).$$

The family of rotations used for our animation is then

$$R(t) = \exp(tA)R_0.$$

At time $t = 0$ we have $\exp(tA) = I$ (no rotation), and so we just start at R_0. At time $t = 1$ we have $\exp(A) = \exp(\log(R_1 R_0^{-1}))R_0 = R_1 R_0^{-1} R_0$, and so we end up at R_1.

3 Rotations in 3D

In 3D[3], we have a similar setup:

- Rotations are represented by 3×3 matrices R that are orthogonal ($R^\mathsf{T} R = RR^\mathsf{T} = I$) and have determinant $+1$.
- A rotation by θ around a unit-length axis $u \in \mathbb{R}^3$ is encoded by a 3×3 skew-symmetric matrix $A = \theta \hat{u}$, where
$$\hat{u} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}.$$
- To go from axis-angle form to rotation matrix form, we can evaluate the exponential map $R = \exp(A)$.
- To go from a rotation matrix to axis-angle form, we can evaluate the log map $A = \log(R)$.

But don't take my word for it! Let's make sure this machinery still works as expected in 3D. First, a couple little warm-ups to get us more comfortable with this setup.

14. In 3D, do rotation matrices form a vector space? Do skew-symmetric matrices form a vector space? Give a little argument for why/why not in each case.

Solution.

Just as in 2D, 3D rotation matrices do not form a vector space. For instance, take the identity matrix I, and the matrix
$$R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
which represents a rotation by 180° around the z-axis. The sum of I and R is then
$$R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$
which is not a rotation (it squashes flat the $x-$ and $y-$ components). Hence, the set of 3D rotation matrices is not a vector space, since it is not closed under addition.
However, if $A_1^\mathsf{T} = -A_1$ and $A_2^\mathsf{T} = -A_2$ are any two skew-symmetric matrices, then
$$(A_1 + A_2)^\mathsf{T} = A_1^\mathsf{T} + A_2^\mathsf{T} = -A_1 - A_2 = -(A_1 + A_2),$$
i.e., the sum of any two skew-symmetric matrices is skew-symmetric. Moreover, scaling by any constant $c \in \mathbb{R}$ preserves skew-symmetry, i.e., if $A^\mathsf{T} = -A$ then
$$(cA)^\mathsf{T} = c(A^\mathsf{T}) = -cA.$$
All other properties follow from properties of addition for general matrices (associativity, etc.).

[3] In fact, this same setup applies for n-dimensional rotations—just replace "3" with "n".

Ok, so rotations don't form a vector space. But what does the set of all rotations look like? Just like it helps to visualize vectors as "little arrows," visualizing rotations geometrically is super helpful whenever we face problems like how to interpolate or average rotations. In general, we will call the set of all n-dimensional rotations the *special orthogonal group* $\mathrm{SO}(n)$. "Orthogonal" because rotation matrices are orthogonal; "special" because they are not just any orthogonal matrices—they must also have positive determinant. In other words,

$$\mathrm{SO}(n) := \{R \in \mathbb{R}^{n\times n} \mid R^T R = I,\ \det(R) > 0\}.$$

We call this set a "group" because composition of rotations (via matrix multiplication) satisfies some very natural properties: the product of two rotations is a rotation; every rotation R can be reversed by some inverse rotation R^{-1}, there is an "identity" rotation that does nothing ($R = I$), and the way we group rotations doesn't matter: $(R_1 R_2)R_3 = R_1(R_2 R_3)$. But what is the *shape* of the rotation group?

15. In 2D, we said that all rotations could be expressed as complex numbers $e^{\imath\theta} = \cos(\theta) + \imath\sin(\theta)$, for any angle $\theta \in [-\pi, \pi)$. Equivalently, we could say that rotations are just complex numbers $z \in \mathbb{C}$ with unit norm: $|z| = 1$. What shape does this set describe?

    Solution.

    The set of all points with unit norm in 2D is the unit circle S^1. So, we can think of each 2D rotation as a point on the circle, and each point on the circle as a 2D rotation—there is a 1-to-1 relationship.

16. In general, for any function f we can consider the set of points x such that $f(x) = 0$. For instance, in the previous question the function $f(x) = |x|^2 - 1$ describes the circle. In general, if we start out in n dimensions, then each scalar condition reduces the dimension of the set by one. For instance, consider the functions $f(x) = |x|^2 - 1$ and $g(x) = x_1$ (where x_1 is the first coordinate of x). Describe the set

    $$S := \{x \in \mathbb{R}^2 | f(x) = 0, g(x) = 0\}.$$

    What points does it contain? What dimension is it?

    Solution.

    The set S contains just the two points $(0, +1)$ and $(0, -1)$, and is hence zero-dimensional. We could also get the dimension by considering that we start out in 2D, and have two scalar conditions. Hence, $2 - 2 = 0$. Note that, in general, this set could be empty—or, if we have repeated conditions, it could be higher-dimensional. For instance, if $f = g$ then effectively we just have one condition, and S would still be one-dimensional. Just like in linear algebra, we have to think about whether our conditions are independent at each point.

17. Another way of getting our head around the shape of the rotation group is to consider the relationship $A^T A = I$ characterizing orthogonal matrices, or rather, the function $f(A) = A^T A - I$. Importantly, the equation $f(A) = 0$ is not a scalar equation: it relates all n^2 entries of the matrix $A^T A$ to the n^2 entries of the identity matrix I. If we let

    $$\mathrm{O}(n) := \{A \in \mathbb{R}^{n\times n} | A^T A - I = 0\}$$

    be the set of orthogonal matrices, what dimension does this set have for $n = 2$? For $n = 3$? For general n?

Solution.

Since the matrix $A^T A$ is symmetric, equations above and below the diagonal will be repeated. For instance, in 2D we have

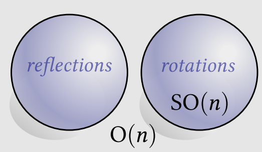
$$A^T A = \begin{bmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11}^2 + A_{21}^2 & A_{11}A_{12} + A_{21}A_{22} \\ A_{11}A_{12} + A_{21}A_{22} & A_{12}^2 + A_{22}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Notice, then, that we have only three independent equations, since the top-right and bottom-left entries of $A^T A$ are the same. Since we have four entries, but only three conditions, we see that the dimension of the set of 2D orthogonal matrices is $4 - 3 = 1$, just like the 2D rotations.
Likewise, for $n = 3$ the matrix A will have $3 \times 3 = 9$ entries, but only six independent equations: we have three entries along the diagonal, and three more above the diagonal. Hence, the dimension of 3D orthogonal matrices is $9 - 6 = 3$.
In general, we will have n^2 entries, but the $(n^2 - n)/2$ equations below the diagonal repeat the corresponding equations above the diagonal. Hence, the dimension of $O(n)$ is $(n^2 - n)/2$.

18. Remember that rotations are not just orthogonal matrices—they're orthogonal matrices with positive determinant. However, this additional *inequality* doesn't reduce the dimension of the set any further: instead, it splits the orthogonal matrices $\mathrm{O}(n)$ into two pieces: those with positive determinant (rotations) and those with negative determinant. What do the negative-determinant orthogonal matrices correspond to?

Solution.

Since the determinant is negative, these matrices reverse orientation; since they are orthogonal, they still preserve distances and angles. Hence, these matrices describe reflections. A good "cartoon" for the orthogonal group is hence two components that are identical in shape and size:



19. Let's get back to the conversion between rotation matrices and the axis-angle encoding. In particular, given a 3×3 skew-symmetric matrix $A^\mathsf{T} = -A$ encoding a rotation in axis-angle form, how can we recover the axis u and angle θ of rotation?

Solution.

Assume $A = \theta\hat{u}$, as above. We want to write A as $\theta\hat{u}$ for some vector $u \in \mathbb{R}^3$. Since each component of u appears twice in $\hat{u}$, we know that the (Frobenius) norm of the matrix A is twice the norm of θu, i.e.,

$$\|A\|^2 := \sum_{i=1}^{3}\sum_{j=1}^{3} A_{ij}^2 = 2|\theta u|^2.$$

But since $|u|^2 = 1$, we have just

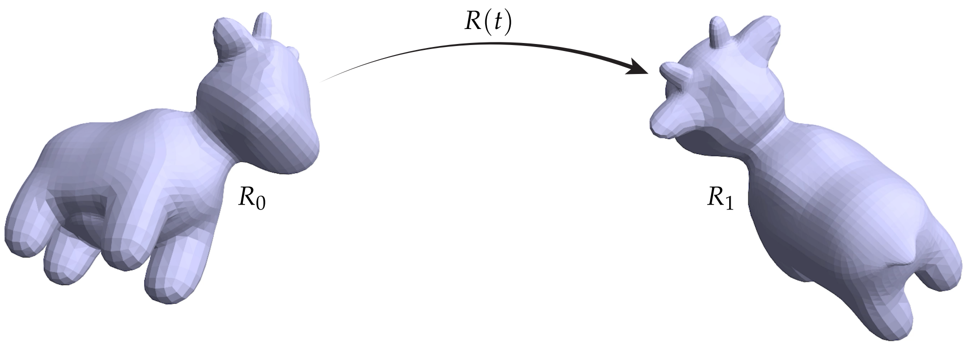
$$\|A\|^2 = 2|\theta|^2,$$

i.e.,

$$\theta = \frac{1}{\sqrt{2}}\|A\|.$$

We can then divide A by θ to recover $\hat{u}$. The entries of u itself are given by (for instance)

$$u = \frac{1}{\theta}\begin{bmatrix} A_{32} \\ A_{13} \\ A_{21} \end{bmatrix}.$$



20. Suppose we have two rotation matrices given in rotation angles:

$$\begin{aligned} R_0 &= R^x(\alpha_0)R^y(\beta_0)R^z(\gamma_0), \\ R_1 &= R^x(\alpha_1)R^y(\beta_1)R^z(\gamma_1), \end{aligned}$$

where $R^x(\theta)$ denotes a rotation by θ around the x-axis (and similarly for y, z). What's a simple way to interpolate between these two rotation matrices using Euler angles? Give an expression for a family of rotation matrices $R(t)$ with $t \in [0, 1]$ such that $R(0) = R_0$ and $R(1) = R_1$. For now, do *not* use the exp/log map.

Solution.

Similar to the 2D case, we can just interpolate the individual Euler angles:

$$\begin{aligned}\alpha(t) &= (1-t)\alpha_0 + t\alpha_1,\\ \beta(t) &= (1-t)\beta_0 + t\beta_1,\\ \gamma(t) &= (1-t)\gamma_0 + t\gamma_1.\end{aligned}$$

The interpolating family of rotations is then given by

$$R(t) = R^x(\alpha(t))R^y(\beta(t))R^z(\gamma(t)).$$

21. Based on your intuition from the 2D case, how do you think you might interpolate the rotations using the exp/log map for 3D rotations?

Solution.

We can just do the same thing: given two rotations R_0 and R_1, the smallest rotation between them (in axis-angle form) is given by

$$A = \log(R_1 R_0^{-1}).$$

Hence, we can interpolate via

$$R(t) = \exp(tA)R_0.$$

As before, this family of rotations starts at $t = 0$ with R_0, and interpolates to R_1 at $t = 1$ with the minimal amount of additional "twisting" in between.

22. If you were paying close attention in our lecture on vector calculus, you may recall that $\hat{u}$ represents a cross product with the vector u, i.e., $\hat{u}x = u \times x$ for any vector $x \in \mathbb{R}^3$. Using your knowledge of the cross product, make a *geometric* argument that $\hat{u}^3 = -\hat{u}$, and hence $\hat{u}^{k+2} = -\hat{u}^k$ for $k \geq 1$.

Solution.

Recall that taking a cross product with the vector u is equivalent to performing a counter-clockwise quarter-turn in the (oriented) plane with normal $+u$. Hence, for any vector $x \in \mathbb{R}^3$,

$$u \times (u \times (u \times x)) = -u \times x,$$

and since this identity holds for all vectors x, we must have

$$\hat{u}^3 = -\hat{u}.$$

23. Derive Rodrigues' formula $\exp(\theta\hat{u}) = I + \sin(\theta)\hat{u} + (1-\cos(\theta))\hat{u}^2$ by writing out the left-hand side via the Taylor series for the matrix exponential.

Solution.

Recall that the matrix exponential has the Taylor series

$$\exp(A) = \sum_{k=0}^{\infty} \frac{1}{k!} A^k.$$

Expanding A, we get

$$I + \theta\hat{u} + \tfrac{1}{2!}\theta^2\hat{u}^2 + \tfrac{1}{3!}\theta^3\hat{u}^3 + \tfrac{1}{4!}\theta^4\hat{u}^4 + \tfrac{1}{5!}\theta^5\hat{u}^5 + \tfrac{1}{6!}\theta^6\hat{u}^6 + \cdots$$

Applying the identity $\hat{u}^{k+2} = -\hat{u}^k$ $(k \geq 1)$ from the previous question yields

$$I + \theta\hat{u} + \tfrac{1}{2!}\theta^2\hat{u}^2 - \tfrac{1}{3!}\theta^3\hat{u} - \tfrac{1}{4!}\theta^4\hat{u}^2 + \tfrac{1}{5!}\theta^5\hat{u} + \tfrac{1}{6}\theta^6\hat{u}^2 + \cdots,$$

and we can group these terms as

$$I + (\theta - \tfrac{1}{3!}\theta^3 + \tfrac{1}{5!}\theta^5 + \cdots)\hat{u} + (\tfrac{1}{2!}\theta^2 - \tfrac{1}{4!}\theta^4 + \tfrac{1}{6}\theta^6 + \cdots)\hat{u}^2.$$

Here we notice that the coefficient for $\hat{u}$ is the Taylor series for $\sin(\theta)$, and likewise, the coefficient for $\hat{u}^2$ converges to $1 - \cos(\theta)$. Hence, we have

$$\exp(\theta\hat{u}) = I + \sin(\theta)\hat{u} + (1 - \cos(\theta))\hat{u}^2,$$

as desired.

24. Is the matrix $\exp(\theta u)$ provided by Rodrigues' formula a rotation matrix for all vectors u and angles θ? How could you check?

Solution.

A matrix is a rotation matrix if and only if it is orthogonal and has positive determinant. We can first check if this matrix is orthogonal by multiplying it with its transpose. Remembering that $\hat{u}^\mathsf{T} = -\hat{u}$, we have

$$\begin{aligned}
&(I + \sin(\theta)\hat{u} + (1-\cos(\theta))\hat{u}^2)^\mathsf{T}(I + \sin(\theta)\hat{u} + (1-\cos(\theta))\hat{u}^2) = \\
&(I^\mathsf{T} + \sin(\theta)\hat{u}^\mathsf{T} + (1-\cos(\theta))(\hat{u}^\mathsf{T})^2)(I + \sin(\theta)\hat{u} + (1-\cos(\theta))\hat{u}^2) = \\
&(I - \sin(\theta)\hat{u} + (1-\cos(\theta))\hat{u}^2)(I + \sin(\theta)\hat{u} + (1-\cos(\theta))\hat{u}^2) = \\
&I + \sin(\theta)\hat{u} + (1-\cos(\theta))\hat{u}^2 \\
&-\sin(\theta)\hat{u} - \sin^2(\theta)\hat{u}^2 - \sin(\theta)(1-\cos(\theta))\hat{u}^3 \\
&+(1-\cos(\theta))\hat{u}^2 + \sin(\theta)(1-\cos(\theta))\hat{u}^3 + (1-\cos(\theta))^2\hat{u}^4.
\end{aligned}$$

If we now apply the identity $\hat{u}^{k+2} = -\hat{u}^k$ for $k \geq 3$, we get

$$\begin{aligned}
&I + \sin(\theta)\hat{u} + (1-\cos(\theta))\hat{u}^2 \\
&-\sin(\theta)\hat{u} - \sin^2(\theta)\hat{u}^2 + \sin(\theta)(1-\cos(\theta))\hat{u} \\
&+(1-\cos(\theta))\hat{u}^2 - \sin(\theta)(1-\cos(\theta))\hat{u} - (1-\cos(\theta))^2\hat{u}^2.
\end{aligned}$$

Canceling equal and opposite terms yields

$$I + (1-\cos(\theta))\hat{u}^2 - \sin^2(\theta)\hat{u}^2 + (1-\cos(\theta))\hat{u}^2 - (1-\cos(\theta))^2\hat{u}^2,$$

and if we collect coefficients for $\hat{u}$ we get

$$\begin{aligned}
&I + \left(1 - \cos(\theta) - \sin^2(\theta) + 1 - \cos(\theta) - 1 + 2\cos(\theta) - \cos(\theta)^2\right)\hat{u}^2 = \\
&I + \left(1 - (\sin^2(\theta) + \cos(\theta)^2)\right)\hat{u}^2 = \\
&I + (1-1)\,\hat{u}^2 = \\
&I.
\end{aligned}$$

Hence, the matrix $\exp(\theta u$ is always orthogonal. We don't yet know if it is a rotation (positive determinant) or reflection (negative determinant). However, the determinant of the identity is positive, and $\exp(\theta u)$ (hence its determinant) changes continuously with respect to continuous changes in u and θ. Since this expression cannot be zero for any values of u, θ, the determinant cannot pass through zero, i.e., it is always positive.

25. In 3D, is the exponential map injective (1-to-1), or not? At a high level, then, what do you think the 3D log map should give us?

Solution.

No: just like in 2D, if we add multiples of 2π to the rotation angle θ we still get the same rotation matrix R. Also as in 2D, it is therefore reasonable to think of log as giving the *smallest* angle of rotation, i.e., for a given rotation matrix R, $\log(R)$ gives us a matrix $A = \theta\hat{u}$ such that (i) $\exp(A) = R$, and (ii) θ is the smallest (in magnitude) angle for which this relationship is satisfied.

26. Let's define the 3D log map explicitly. In particular, given a 3×3 rotation matrix R, the log map should produce a matrix $A = \log(R)$ such that $\exp(A) = R$. Give an explicit formula for $\log(R)$. (Hint: what happens when you take the trace of Rodrigues' formula for R? What happens when you antisymmetrize this formula, i.e., evaluate $R - R^\mathsf{T}$?)

Solution.

Recall Rodrigues formula, which says that a rotation R by θ around a unit axis u is given by

$$R = \exp(\theta u) = I + \sin(\theta)\hat{u} + (1 - \cos(\theta))\hat{u}^2.$$

Taking the trace of the given rotation matrix therefore yields

$$\mathrm{tr}(R) = \mathrm{tr}(I) + \sin\theta\,\mathrm{tr}(\hat{u}) + (1 - \cos\theta)\,\mathrm{tr}(\hat{u}^2).$$

In 3D, $\mathrm{tr}(I) = 1 + 1 + 1 = 3$, and since $\hat{u}$ is a skew-symmetric matrix it has zeros along its diagonal, hence $\mathrm{tr}(\hat{u}) = 0$. Writing out $\hat{u}^2$ yields

$$\begin{bmatrix} -u_2^2 - u_3^2 & u_1 u_2 & u_1 u_3 \\ u_1 u_2 & -u_1^2 - u_3^2 & u_2 u_3 \\ u_1 u_3 & u_2 u_3 & -u_1^2 - u_2^2 \end{bmatrix},$$

which means $\mathrm{tr}(\hat{u}^2) = -2|u|^2$. But since u is a unit vector, we get just $\mathrm{tr}(\hat{u}^2) = -2$. Hence, our overall trace becomes

$$\mathrm{tr}(R) = 3 + \sin(\theta) \cdot 0 + (1 - cos\theta) \cdot (-2) = 1 + 2\cos\theta,$$

which means the angle of rotation is

$$\boxed{\theta = \arccos\left(\frac{\mathrm{tr}(R) - 1}{2}\right).}$$

Next, if we antisymmetrize R, we get

$$R - R^\mathsf{T} = I - I^\mathsf{T} + \sin(\theta)(\hat{u} - \hat{u}^\mathsf{T}) + (1 - \cos\theta)(\hat{u}^2 - (\hat{u}^2)^\mathsf{T}).$$

The first term vanishes, and since $\hat{u}$ is skew-symmetric we have $\hat{u} - \hat{u}^\mathsf{T} = 2\hat{u}$. Also note that

$$\hat{u}^2 = \hat{u}\hat{u} = -\hat{u}^\mathsf{T}\hat{u},$$

i.e., $\hat{u}^2$ is a symmetric matrix. Hence, the third term also vanishes and we get just

$$R - R^\mathsf{T} = 2\sin(\theta)\hat{u},$$

or equivalently,

$$\hat{u} = \frac{1}{2\sin\theta}(R - R^\mathsf{T}).$$

Hence, the axis of rotation is

$$\boxed{u = \frac{1}{2\sin\theta}\begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}.}$$

Together, the two boxed expressions determine the log map $\log(R) = \theta\hat{u}$.

27. For ordinary numbers, we have the useful identity

$$e^{a+b} = e^a e^b.$$

Do you think this same identity holds for matrices, i.e., that

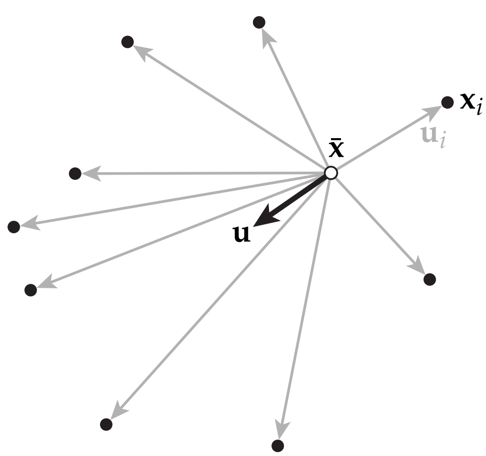
$$\exp(A_1 + A_2) = \exp(A_1)\exp(A_2)$$

for any two skew-symmetric matrices A_1, A_2? In 2D? In 3D? Why or why not? Give an argument that makes an appeal to the way rotations behave *geometrically*—not an algebraic argument.

Solution.

Yes in 2D; no in 3D. The reason this identity can't possibly hold in 3D is that, in general, rotations R_1, R_2 are not commutative, i.e., $R_1R_2 \neq R_2R_1$. Since each of the matrices R_i can be represented by a logarithm $A_i = \log(R_i)$, this identity cannot hold in general. In 2D, however, rotations do commute.

28. Finally, if all we wanted to do with the log/exp map was interpolate rotations, this might feel like much ado about nothing! But now that we can treat rotations like vectors, there's other useful stuff we can do. For instance, given a collection of rotation matrices $R_1, \ldots, R_n \in \mathbb{R}^{3\times 3}$, we can define a meaningful notion of the *average rotation*. As discussed above, we can't just take an average of the matrices $\frac{1}{n}\sum_{i=1}^{n} R_i$, since the average of rotation matrices will not in general be a rotation matrix.



Instead, let's first think about a "funny" way to average a bunch of points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in the plane. Instead of just directly taking the average (which we can't do with rotations), we'll pick some initial guess $\bar{\mathbf{x}}$ for the average. We'll then compute, for each point, the smallest vector $\mathbf{u}_i$ that takes us from $\bar{\mathbf{x}}$ to $\mathbf{x}_i$. In the case of points in the plane, this vector is just $\mathbf{u}_i = \mathbf{x}_i - \mathbf{x}$. We'll then compute the average vector $\mathbf{u} := \frac{1}{n}\sum_{i=1}^n \mathbf{u}_i$ that tries to pull us toward all the points, and take a little step in this direction of size $\tau \in [0,1]$. If we reach a point where $\mathbf{u} = 0$—or at least, below some very small $\epsilon > 0$—then the algorithm stops and we know we've reached the average. This algorithm can be summarized as follows:

- Pick an initial guess $\bar{\mathbf{x}} \in \mathbb{R}^2$.
- Do
  - $\mathbf{u}_i \leftarrow \mathbf{x_i} - \mathbf{x}$
  - $\mathbf{u} \leftarrow \frac{1}{n}\sum_{i=1}^n \mathbf{u}_i$
  - $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} + \tau\mathbf{u}$
- While $|\mathbf{u}| > \epsilon$

Your task is to re-write this same algorithm for rotations. For instance, given a current guess $\bar{R}$ for the average rotation, how do you find the smallest rotation from $\bar{R}$ to each of the R_i? Where and how do you compute the average vector? How do you take a small step in this direction? How do you pick a valid initial guess?

Solution.

The algorithm for averaging rotations is nearly identical:

- Pick an initial guess $\bar{R} \in \mathbb{R}^{3\times3}$.
- Do
  - $A_i \leftarrow \log(R_i\bar{R}^{-1})$
  - $A \leftarrow \frac{1}{n}\sum_{i=1}^n A_i$
  - $\bar{R} \leftarrow \exp(\tau A)\bar{R}$
- While $|A| > \epsilon$

The initial guess could be any rotation matrix—say, the identity $\bar{R} = I$. Rather than computing differences of rotation matrices, we compute the smallest rotation from $\bar{R}$ to each of the R_i via the log map, yielding a bunch of skew-symmetric matrices A_i. Since these matries belong to a common vector space, we can average them to produce another skew-symmetric matrix, i.e., another axis-angle representation of a rotation. Applying the exponential map moves $\bar{R}$ a little bit toward the average, and we repeat until convergence [a].

[a]This algorithm is essentially known as the *Weiszfeld algorithm*, and the notion of average we get in the end is referred to as the *Karcher mean* of the rotations.

29. Implement some code (in your favorite language) that computes the average of a given list of rotations. You should now have all the tools and formulas you need to really do this! (Note: you may need to be a bit more careful about corner cases when you implement the log map...)