Derivatives and Differentials

Frank Dellaert

December 17, 2013

Part I
Theory

1 Optimization

We will be concerned with minimizing a non-linear least squares objective of the form

$$x^* = \arg\min_x \|h(x) - z\|^2_\Sigma \tag{1.1}$$

where $x \in \mathcal{M}$ is a point on an n-dimensional manifold (which could be $\mathbb{R}^n$, an n-dimensional Lie group G, or a general manifold $\mathcal{M}$), $z \in \mathbb{R}^m$ is an observed measurement, $h : \mathcal{M} \to \mathbb{R}^m$ is a measurement function that predicts z from x, and $\|e\|^2_\Sigma \overset{\Delta}{=} e^T \Sigma^{-1} e$ is the squared Mahalanobis distance with covariance Σ.

To minimize (1.1) we need a notion of how the non-linear measurement function $h(x)$ behaves in the neighborhood of a linearization point a. Loosely speaking, we would like to define an $m \times n$ Jacobian matrix H_a such that

$$h(a \oplus \xi) \approx h(a) + H_a \xi \tag{1.2}$$

with $\xi \in \mathbb{R}^n$, and the operation $\oplus$ "increments" $a \in \mathcal{M}$. Below we more formally develop this notion, first for functions from $\mathbb{R}^n \to \mathbb{R}^m$, then for Lie groups, and finally for manifolds.

Once equipped with the approximation (1.2), we can minimize the objective function (1.1) with respect to δx instead:

$$\xi^* = \arg\min_\xi \|h(a) + H_a \xi - z\|^2_\Sigma \tag{1.3}$$

This can be done by setting the derivative of (1.3) to zero, yielding the **normal equations**,

$$H_a^T H_a \xi = H_a^T (z - h(a))$$

which can be solved using Cholesky factorization. Of course, we might have to iterate this multiple times, and use a trust-region method to bound ξ when the approximation (1.2) is not good.

2 Multivariate Differentiation

2.1 Derivatives

For a vector space $\mathbb{R}^n$, the notion of an increment is just done by vector addition

$$a \oplus \xi \overset{\Delta}{=} a + \xi$$

and for the approximation 1.2 we will use a Taylor expansion using multivariate differentiation. However, loosely following [5], we use a perhaps unfamiliar way to define derivatives:

Definition 1. We define a function $f : \mathbb{R}^n \to \mathbb{R}^m$ to be **differentiable** at a if there exists a matrix $f'(a) \in \mathbb{R}^{m \times n}$ such that

$$\lim_{\delta x \to 0} \frac{|f(a) + f'(a)\xi - f(a+\xi)|}{|\xi|} = 0$$

where $|e| \overset{\Delta}{=} \sqrt{e^T e}$ is the usual norm. If f is differentiable, then the matrix $f'(a)$ is called the **Jacobian matrix** of f at a, and the linear map $Df_a : \xi \mapsto f'(a)\xi$ is called the **derivative** of f at a. When no confusion is likely, we use the notation $F_a \overset{\Delta}{=} f'(a)$ to stress that $f'(a)$ is a matrix.

The benefit of using this definition is that it generalizes the notion of a scalar derivative $f'(a) : \mathbb{R} \to \mathbb{R}$ to multivariate functions from $\mathbb{R}^n \to \mathbb{R}^m$. In particular, the derivative Df_a maps vector increments ξ on a to increments $f'(a)\xi$ on $f(a)$, such that this linear map locally approximates f:

$$f(a+\xi) \approx f(a) + f'(a)\xi$$

Example 1. The function $\pi : (x,y,z) \mapsto (x/z, y/z)$ projects a 3D point (x,y,z) to the image plane, and has the Jacobian matrix

$$\pi'(x,y,z) = \frac{1}{z}\begin{bmatrix} 1 & 0 & -x/z \\ 0 & 1 & -y/z \end{bmatrix}$$

2.2 Properties of Derivatives

This notion of a multivariate derivative obeys the usual rules:

Theorem 1. *(Chain rule) If $f : \mathbb{R}^n \to \mathbb{R}^p$ is differentiable at a and $g : \mathbb{R}^p \to \mathbb{R}^m$ is differentiable at $f(a)$, then the Jacobian matrix H_a of $h = g \circ f$ at a is the $m \times n$ matrix product*

$$H_a = G_{f(a)} F_a$$

where $G_{f(a)}$ is the $m \times p$ Jacobian matrix of g evaluated at $f(a)$, and F_a is the $p \times n$ Jacobian matrix of f evaluated at a.

Proof. See [5] □

Example 2. If we follow the projection π by a calibration step $\gamma : (x,y) \mapsto (u_0 + fx, u_0 + fy)$, with

$$\gamma'(x,y) = \begin{bmatrix} f & 0 \\ 0 & f \end{bmatrix}$$

then the combined function $\gamma \circ \pi$ has the Jacobian matrix

$$(\gamma \circ \pi)'(x,y) = \frac{f}{z} \begin{bmatrix} 1 & 0 & -x/z \\ 0 & 1 & -y/z \end{bmatrix}$$

Theorem 2. *(Inverse) If $f : \mathbb{R}^n \to \mathbb{R}^n$ is differentiable and has a differentiable inverse $g \triangleq f^{-1}$, then its Jacobian matrix G_a at a is just the inverse of that of f, evaluated at $g(a)$:*

$$G_a = \left[F_{g(a)}\right]^{-1}$$

Proof. See [5] □

Example 3. The function $f : (x,y) \mapsto (x^2, xy)$ has the Jacobian matrix

$$F_{(x,y)} = \begin{bmatrix} 2x & 0 \\ y & x \end{bmatrix}$$

and, for $x \geq 0$, its inverse is the function $g : (x,y) \mapsto (x^{1/2}, x^{-1/2}y)$ with the Jacobian matrix

$$G_{(x,y)} = \frac{1}{2} \begin{bmatrix} x^{-1/2} & 0 \\ -x^{-3/2}y & 2x^{-1/2} \end{bmatrix}$$

It is easily verified that

$$g'(a,b)f'(a^{1/2}, a^{-1/2}b) = \frac{1}{2} \begin{bmatrix} a^{-1/2} & 0 \\ -a^{-3/2}b & 2a^{-1/2} \end{bmatrix} \begin{bmatrix} 2a^{1/2} & 0 \\ a^{-1/2}b & a^{1/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Problem 1. Verify the above for $(a,b) = (4,6)$. Sketch the situation graphically to get insight.

2.3 Computing Multivariate Derivatives

Computing derivatives is made easy by defining the concept of a partial derivative:

Definition 2. For $f : \mathbb{R}^n \to \mathbb{R}$, the **partial derivative** of f at a,

$$D_j f(a) \triangleq \lim_{h \to 0} \frac{f\left(a^1, \ldots, a^j + h, \ldots, a^n\right) - f\left(a^1, \ldots, a^n\right)}{h}$$

which is the ordinary derivative of the scalar function $g(x) \triangleq f\left(a^1, \ldots, x, \ldots, a^n\right)$.

Using this definition, one can show that the Jacobian matrix F_a of a differentiable *multivariate* function $f : \mathbb{R}^n \to \mathbb{R}^m$ consists simply of the $m \times n$ partial derivatives $D_j f^i(a)$, evaluated at $a \in \mathbb{R}^n$:

$$F_a = \begin{bmatrix} D_1 f^1(a) & \cdots & D_n f^1(a) \\ \vdots & \ddots & \vdots \\ D_1 f^m(a) & \ldots & D_n f^m(a) \end{bmatrix}$$

Problem 2. Verify the derivatives in Examples 1 to 3.

3 Multivariate Functions on Lie Groups

3.1 Lie Groups

Lie groups are not as easy to treat as the vector space $\mathbb{R}^n$ but nevertheless have a lot of structure. To generalize the concept of the total derivative above we just need to replace $a \oplus \xi$ in (1.3) with a suitable operation in the Lie group G. In particular, the notion of an exponential map allows us to define an incremental transformation as tracing out a geodesic curve on the group manifold along a certain **tangent vector** ξ,

$$a \oplus \xi \overset{\Delta}{=} a \exp\left(\hat{\xi}\right)$$

with $\xi \in \mathbb{R}^n$ for an n-dimensional Lie group, $\hat{\xi} \in \mathfrak{g}$ the Lie algebra element corresponding to the vector ξ, and $\exp \hat{\xi}$ the exponential map. Note that if G is equal to $\mathbb{R}^n$ then composing with the exponential map $ae^{\hat{\xi}}$ is just vector addition $a + \xi$.

Example 4. For the Lie group $SO(3)$ of 3D rotations the vector ξ is denoted as ω and represents an angular displacement. The Lie algebra element $\hat{\xi}$ is a skew symmetric matrix denoted as $[\omega]_\times \in \mathfrak{so}(3)$, and is given by

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Finally, the increment $a \oplus \xi = ae^{\hat{\xi}}$ corresponds to an incremental rotation $R \oplus \omega = Re^{[\omega]_\times}$.

3.2 Derivatives

We can generalize Definition 1 to map exponential coordinates ξ to increments $f'(a)\xi$ on $f(a)$, such that the linear map Df_a locally approximates a function f from G to $\mathbb{R}^m$:

$$f(ae^{\hat{\xi}}) \approx f(a) + f'(a)\xi$$

Definition 3. We define a function $f : G \to \mathbb{R}^m$ to be **differentiable** at $a \in G$ if there exists a matrix $f'(a) \in \mathbb{R}^{m \times n}$ such that

$$\lim_{\xi \to 0} \frac{\left| f(a) + f'(a)\xi - f(ae^{\hat{\xi}}) \right|}{|\xi|} = 0$$

If f is differentiable, then the matrix $f'(a)$ is called the **Jacobian matrix** of f at a, and the linear map $Df_a : \xi \mapsto f'(a)\xi$ is called the **derivative** of f at a.

Note that the vectors ξ can be viewed as lying in the tangent space to G at a, but defining this rigorously would take us on a longer tour of differential geometry. Informally, ξ is simply the direction, in a local coordinate frame, that is locally tangent at a to a geodesic curve $\gamma : t \mapsto ae^{\widehat{t\xi}}$ traced out by the exponential map, with $\gamma(0) = a$.

3.3 Derivative of an Action

The (usual) action of an n-dimensional matrix group G is matrix-vector multiplication on $\mathbb{R}^n$, i.e., $f : G \times \mathbb{R}^n \to \mathbb{R}^n$ with

$$f(T,p) = Tp$$

Since this is a function defined on the product $G \times \mathbb{R}^n$ the derivative is a linear transformation $Df : \mathbb{R}^{2n} \to \mathbb{R}^n$ with

$$Df_{(T,p)}(\xi, \delta p) = D_1 f_{(T,p)}(\xi) + D_2 f_{(T,p)}(\delta p)$$

Theorem 3. *The Jacobian matrix of the group action$f(T,p) = Tp$ at (T,p) is given by*

$$F_{(T,p)} = \left[\begin{array}{cc} TH(p) & T \end{array} \right] = T \left[\begin{array}{cc} H(p) & I_n \end{array} \right]$$

with $H : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ a linear mapping that depends on p, and I_n the $n \times n$ identity matrix.

Proof. First, the derivative $D_2 f$ with respect to p is easy, as its matrix is simply T:

$$f(T, p + \delta p) = T(p + \delta p) = Tp + T\delta p = f(T,p) + D_2 f(\delta p)$$

For the derivative $D_1 f$ with respect to a change in the first argument T, we want to find the linear map $D_1 f$ such that

$$Tp + D_1 f(\xi) \approx f(Te^{\hat{\xi}}, p) = Te^{\hat{\xi}} p$$

Since the matrix exponential is given by the series $e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$ we have, to first order

$$Te^{\hat{\xi}} p \approx T(I + \hat{\xi})p = Tp + T\hat{\xi}p$$

and $D_1 f(\xi) = T\hat{\xi}p$. Hence, to complete the proof, we need to show that

$$\hat{\xi}p = H(p)\xi \tag{3.1}$$

with $H(p)$ an $n \times n$ matrix that depends on p. Expressing the map $\xi \to \hat{\xi}$ in terms of the Lie algebra generators G^i, using tensors and Einstein summation, we have $\hat{\xi}^i_j = G^i_{jk}\xi^k$ allowing us to calculate $\hat{\xi}p$ as

$$\left(\hat{\xi}p\right)^i = \hat{\xi}^i_j p^j = G^i_{jk}\xi^k p^j = \left(G^i_{jk}p^j\right)\xi^k = H^i_k(p)\xi^k$$

□

Example 5. For 3D rotations $R \in SO(3)$, we have $\hat{\omega} = [\omega]_\times$ and

$$G_{k=1} : \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} G_{k=2} : \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} G_{k=3} : \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The matrices $\left(G^i_k\right)_j$ are obtained by assembling the j^{th} columns of the generators above, yielding $H(p)$ equal to:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} p^1 + \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} p^2 + \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} p^3 = \begin{pmatrix} 0 & p^3 & -p^2 \\ -p^3 & 0 & p^1 \\ p^2 & -p^1 & 0 \end{pmatrix} = [-p]_\times$$

Hence, the Jacobian matrix of $f(R,p) = Rp$ is given by

$$F_{(R,p)} = R \left(\begin{array}{cc} [-p]_\times & I_3 \end{array} \right)$$

3.4 Derivative of an Inverse Action

Applying the action by the inverse of $T \in G$ yields a function $g : G \times \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$g(T,p) = T^{-1}p$$

Theorem 4. *The Jacobian matrix of the inverse group action $g(T,p) = T^{-1}p$ is given by*

$$G_{(T,p)} = \left[\begin{array}{cc} -H(T^{-1}p) & T^{-1} \end{array} \right]$$

where $H : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the same mapping as before.

Proof. Again, the derivative D_2g with respect to in p is easy, the matrix of which is simply T^{-1}:

$$g(T,p+\delta p) = T^{-1}(p+\delta p) = T^{-1}p + T^{-1}\delta p = g(T,p) + D_2g(\delta p)$$

Conversely, a change in T yields

$$g(Te^{\hat{\xi}},p) = \left(Te^{\hat{\xi}}\right)^{-1} p = e^{-\hat{\xi}}T^{-1}p$$

Similar to before, if we expand the matrix exponential we get

$$e^{-A} = I - A + \frac{A^2}{2!} - \frac{A^3}{3!} + \dots$$

so

$$e^{-\hat{\xi}}T^{-1}p \approx (I - \hat{\xi})T^{-1}p = g(T,p) - \hat{\xi}\left(T^{-1}p\right)$$

□

Example 6. For 3D rotations $R \in SO(3)$ we have $R^{-1} = R^T$, $H(p) = -[p]_\times$, and hence the Jacobian matrix of $g(R,p) = R^Tp$ is given by

$$G_{(R,p)} = \left(\begin{array}{cc} [R^Tp]_\times & R^T \end{array} \right)$$

4 Instantaneous Velocity

For matrix Lie groups, if we have a matrix $T^n_b(t)$ that depends on a parameter t, i.e., $T^n_b(t)$ follows a curve on the manifold, then it would be of interest to find the velocity of a point $q^n(t) = T^n_b(t)p^b$ acted upon by $T^n_b(t)$. We can express the velocity of $q(t)$ in both the n-frame and b-frame:

$$\dot{q}^n = \dot{T}^n_b p^b = \dot{T}^n_b \left(T^n_b\right)^{-1} p^n \quad \text{and} \quad \dot{q}^b = \left(T^n_b\right)^{-1}\dot{q}^n = \left(T^n_b\right)^{-1}\dot{T}^n_b p^b$$

Both the matrices $\hat{\xi}^n_{nb} \overset{\Delta}{=} \dot{T}^n_b \left(T^n_b\right)^{-1}$ and $\hat{\xi}^b_{nb} \overset{\Delta}{=} \left(T^n_b\right)^{-1}\dot{T}^n_b$ are skew-symmetric Lie algebra elements that describe the **instantaneous velocity** [4, page 51 for rotations, page 419 for SE(3)]. We will revisit this for both rotations and rigid 3D transformations.

5 Differentials: Smooth Mapping between Lie Groups

5.1 Motivation and Definition

The above shows how to compute the derivative of a function $f : G \to \mathbb{R}^m$. However, what if the argument to f is itself the result of a mapping between Lie groups? In other words, $f = g \circ \varphi$, with $g : G \to \mathbb{R}^m$ and where $\varphi : H \to G$ is a smooth mapping from the n-dimensional Lie group H to the p-dimensional Lie group G. In this case, one would expect that we can arrive at Df_a by composing linear maps, as follows:

$$f'(a) = (g \circ \varphi)'(a) = G_{\varphi(a)} \varphi'(a)$$

where $\varphi'(a)$ is an $n \times p$ matrix that is the best linear approximation to the map $\varphi : H \to G$. The corresponding linear map $D\varphi_a$ is called the **differential** or **pushforward** of the mapping φ at a.

Because a rigorous definition will lead us too far astray, here we only informally define the pushforward of φ at a as the linear map $D\varphi_a : \mathbb{R}^n \to \mathbb{R}^p$ such that $D\varphi_a(\xi) \overset{\Delta}{=} \varphi'(a)\xi$ and

$$\varphi\left(ae^{\hat{\xi}}\right) \approx \varphi(a) \exp\left(\widehat{\varphi'(a)\xi}\right) \tag{5.1}$$

with equality for $\xi \to 0$. We call $\varphi'(a)$ the **Jacobian matrix** of the map φ at a. Below we show that even with this informal definition we can deduce the pushforward in a number of useful cases.

5.2 Left Multiplication with a Constant

Theorem 5. *Suppose G is an n-dimensional Lie group, and $\varphi : G \to G$ is defined as $\varphi(g) = hg$, with $h \in G$ a constant. Then $D\varphi_a$ is the identity mapping and*

$$\varphi'(a) = I_n$$

Proof. Defining $y = D\varphi_a x$ as in (5.1), we have

$$\begin{aligned} \varphi(a)e^{\hat{y}} &= \varphi(ae^{\hat{x}}) \\ hae^{\hat{y}} &= hae^{\hat{x}} \\ y &= x \end{aligned}$$

□

5.3 Pushforward of the Inverse Mapping

A well known property of Lie groups is the the fact that applying an incremental change $\hat{\xi}$ in a different frame g can be applied in a single step by applying the change $Ad_g\hat{\xi}$ in the original frame,

$$ge^{\hat{\xi}}g^{-1} = \exp\left(Ad_g\hat{\xi}\right) \tag{5.2}$$

where $Ad_g : \mathfrak{g} \to \mathfrak{g}$ is the **adjoint representation**. This comes in handy in the following:

Theorem 6. *Suppose that $\varphi : G \to G$ is defined as the mapping from an element g to its* ***inverse*** *g^{-1}, i.e., $\varphi(g) = g^{-1}$, then the pushforward $D\varphi_a$ satisfies*

$$(D\varphi_a x)\hat{}= -Ad_a\hat{x} \tag{5.3}$$

In other words, and this is intuitive in hindsight, approximating the inverse is accomplished by negation of $\hat{\xi}$, along with an adjoint to make sure it is applied in the right frame. Note, however, that (5.3) does not immediately yield a useful expression for the Jacobian matrix $\varphi'(a)$, but in many important cases this will turn out to be easy.

Proof. Defining $y = D\varphi_a x$ as in (5.1), we have

$$\begin{aligned} \varphi(a)e^{\hat{y}} &= \varphi(ae^{\hat{x}}) \\ a^{-1}e^{\hat{y}} &= \left(ae^{\hat{x}}\right)^{-1} \\ e^{\hat{y}} &= -ae^{\hat{x}}a^{-1} \\ \hat{y} &= -Ad_a\hat{x} \end{aligned}$$

□

Example 7. For 3D rotations $R \in SO(3)$ we have

$$Ad_g(\hat{\omega}) = R\hat{\omega}R^T = [R\omega]_\times$$

and hence the pushforward for the inverse mapping $\varphi(R) = R^T$ has the matrix $\varphi'(R) = -R$.

5.4 Right Multiplication with a Constant

Theorem 7. *Suppose $\varphi : G \to G$ is defined as $\varphi(g) = gh$, with $h \in G$ a constant. Then $D\varphi_a$ satisfies*

$$(D\varphi_a x)\hat{}= Ad_{h^{-1}}\hat{x}$$

Proof. Defining $y = D\varphi_a x$ as in (5.1), we have

$$\begin{aligned} \varphi(a)e^{\hat{y}} &= \varphi(ae^{\hat{x}}) \\ ahe &= ae^{\hat{x}}h \\ e^{\hat{y}} &= h^{-1}e^{\hat{x}}h = \exp\left(Ad_{h^{-1}}\hat{x}\right) \\ \hat{y} &= Ad_{h^{-1}}\hat{x} \end{aligned}$$

□

Example 8. In the case of 3D rotations, right multiplication with a constant rotation R is done through the mapping $\varphi(A) = AR$, and satisfies

$$[D\varphi_A x]_\times = Ad_{R^T}[x]_\times$$

For 3D rotations $R \in SO(3)$ we have

$$Ad_{R^T}(\hat{\omega}) = R^T\hat{\omega}R = [R^T\omega]_\times$$

and hence the Jacobian matrix of φ at A is $\varphi'(A) = R^T$.

5.5 Pushforward of Compose

Theorem 8. *If we define the mapping $\varphi : G \times G \rightarrow G$ as the product of two group elements $g,h \in G$, i.e., $\varphi(g,h) = gh$, then the pushforward will satisfy*

$$D\varphi_{(a,b)}(x,y) = D_1\varphi_{(a,b)}x + D_2\varphi_{(a,b)}y$$

with

$$\left(D_1\varphi_{(a,b)}x\right)\hat{} = Ad_{b^{-1}}\hat{x} \;\; \textit{and} \;\; D_2\varphi_{(a,b)}y = y$$

Proof. Looking at the first argument, the proof is very similar to right multiplication with a constant b. Indeed, defining $y = D\varphi_a x$ as in (5.1), we have

$$\begin{aligned} \varphi(a,b)e^{\hat{y}} &= \varphi(ae^{\hat{x}},b) \\ abe^{\hat{y}} &= ae^{\hat{x}}b \\ e^{\hat{y}} &= b^{-1}e^{\hat{x}}b = \exp\left(Ad_{b^{-1}}\hat{x}\right) \\ \hat{y} &= Ad_{b^{-1}}\hat{x} \end{aligned} \tag{5.4}$$

In other words, to apply an incremental change $\hat{x}$ to a we first need to undo b, then apply $\hat{x}$, and then apply b again. Using (5.2) this can be done in one step by simply applying $Ad_{b^{-1}}\hat{x}$.

The second argument is quite a bit easier and simply yields the identity mapping:

$$\begin{aligned} \varphi(a,b)e^{\hat{y}} &= \varphi(a,be^{\hat{x}}) \\ abe^{\hat{y}} &= abe^{\hat{x}} \\ y &= x \end{aligned} \tag{5.5}$$

□

Example 9. For 3D rotations $A,B \in SO(3)$ we have $\varphi(A,B) = AB$, and $Ad_{B^T}[\omega]_\times = [B^T\omega]_\times$, hence the Jacobian matrix $\varphi'(A,B)$ of composing two rotations is given by

$$\varphi'(A,B) = \left[\begin{array}{cc} B^T & I_3 \end{array}\right]$$

5.6 Pushforward of Between

Finally, let us find the pushforward of **between**, defined as $\varphi(g,h) = g^{-1}h$. For the first argument we reason as:

$$\begin{aligned} \varphi(g,h)e^{\hat{y}} &= \varphi(ge^{\hat{x}},h) \\ g^{-1}he^{\hat{y}} &= \left(ge^{\hat{x}}\right)^{-1}h = -e^{\hat{x}}g^{-1}h \\ e^{\hat{y}} &= -\left(h^{-1}g\right)e^{\hat{x}}\left(h^{-1}g\right)^{-1} = -\exp Ad_{\left(h^{-1}g\right)}\hat{x} \\ \hat{y} &= -Ad_{\left(h^{-1}g\right)}\hat{x} = -Ad_{\varphi(h,g)}\hat{x} \end{aligned} \tag{5.6}$$

The second argument yields the identity mapping.

Example 10. For 3D rotations $A,B \in SO(3)$ we have $\varphi(A,B) = A^TB$, and $Ad_{B^TA}[-\omega]_\times = [-B^TA\omega]_\times$, hence the Jacobian matrix $\varphi'(A,B)$ of between is given by

$$\varphi'(A,B) = \left[\begin{array}{cc} \left(-B^TA\right) & I_3 \end{array}\right]$$

5.7 Numerical PushForward

Let's examine

$$f(g)\,e^{\hat{y}} = f\left(ge^{\hat{x}}\right)$$

and multiply with $f(g)^{-1}$ on both sides:

$$e^{\hat{y}} = f(g)^{-1} f\left(ge^{\hat{x}}\right)$$

We then take the log (which in our case returns y, not $\hat{y}$):

$$y(x) = \log\left[f(g)^{-1} f\left(ge^{\hat{x}}\right)\right]$$

Let us look at $x = 0$, and perturb in direction i, $e_i = [0,0,1,0,0]$. Then take derivative,

$$\frac{\partial y(d)}{\partial d} \triangleq \lim_{d\to 0} \frac{y(d)-y(0)}{d} = \lim_{d\to 0} \frac{1}{d} \log\left[f(g)^{-1} f\left(ge^{\widehat{de_i}}\right)\right]$$

which is the basis for a numerical derivative scheme.

5.8 Derivative of the Exponential and Logarithm Map

Theorem 9. *The derivative of the function $f : \mathbb{R}^n \to G$ that applies the wedge operator followed by the exponential map, i.e., $f(\xi) = \exp\hat{\xi}$, is the identity map for $\xi = 0$.*

Proof. For $\xi = 0$, we have

$$\begin{aligned} f(\xi)e^{\hat{y}} &= f(\xi + x) \\ f(0)e^{\hat{y}} &= f(0+x) \\ e^{\hat{y}} &= e^{\hat{x}} \end{aligned}$$

□

Corollary 1. *The derivative of the inverse f^{-1} is the identity as well, i.e., for $T = e$, the identity element in G.*

For $\xi \neq 0$, things are not simple, see . `http://deltaepsilons.wordpress.com/2009/11/06/helgasons-formula-for-the-differential-of-the-exponential/`.

6 General Manifolds

6.1 Retractions

General manifolds that are not Lie groups do not have an exponential map, but can still be handled by defining a **retraction** $\mathscr{R}:\mathscr{M}\times\mathbb{R}^n\rightarrow\mathscr{M}$, such that

$$a\oplus\xi\overset{\Delta}{=}\mathscr{R}_a(\xi)$$

A retraction [1] is required to be tangent to geodesics on the manifold $\mathscr{M}$ at a. We can define many retractions for a manifold $\mathscr{M}$, even for those with more structure. For the vector space $\mathbb{R}^n$ the retraction is just vector addition, and for Lie groups the obvious retraction is simply the exponential map, i.e., $\mathscr{R}_a(\xi)=a\cdot\exp\hat{\xi}$. However, one can choose other, possibly computationally attractive retractions, as long as around a they agree with the geodesic induced by the exponential map, i.e.,

$$\lim_{\xi\rightarrow0}\frac{\left|a\cdot\exp\hat{\xi}-\mathscr{R}_a(\xi)\right|}{|\xi|}=0$$

Example 11. For $SE(3)$, instead of using the true exponential map it is computationally more efficient to define the retraction, which uses a first order approximation of the translation update

$$\mathscr{R}_T\left(\begin{bmatrix}\omega\\v\end{bmatrix}\right)=\begin{bmatrix}R&t\\0&1\end{bmatrix}\begin{bmatrix}e^{[\omega]_\times}&v\\0&1\end{bmatrix}=\begin{bmatrix}Re^{[\omega]_\times}&t+Rv\\0&1\end{bmatrix}$$

6.2 Derivatives

Equipped with a retraction, then, we can generalize the notion of a derivative for functions f from general a manifold $\mathscr{M}$ to $\mathbb{R}^m$:

Definition 4. We define a function $f:\mathscr{M}\rightarrow\mathbb{R}^m$ to be **differentiable** at $a\in\mathscr{M}$ if there exists a matrix $f'(a)$ such that

$$\lim_{\xi\rightarrow0}\frac{|f(a)+f'(a)\xi-f(\mathscr{R}_a(\xi))|}{|\xi|}=0$$

with $\xi\in\mathbb{R}^n$ for an n-dimensional manifold, and $\mathscr{R}_a:\mathbb{R}^n\rightarrow\mathscr{M}$ a retraction $\mathscr{R}$ at a. If f is differentiable, then $f'(a)$ is called the **Jacobian matrix** of f at a, and the linear transformation $Df_a:\xi\mapsto f'(a)\xi$ is called the **derivative** of f at a.

For manifolds that are also Lie groups, the derivative of any function $f:G\rightarrow\mathbb{R}^m$ will agree no matter what retraction $\mathscr{R}$ is used.

Part II
Practice

Below we apply the results derived in the theory part to the geometric objects we use in GTSAM. Above we preferred the modern notation $D_1 f$ for the partial derivative. Below (because this was written earlier) we use the more classical notation

$$\frac{\partial f(x,y)}{\partial x}$$

In addition, for Lie groups we will abuse the notation and take

$$\left.\frac{\partial \varphi(g)}{\partial \xi}\right|_a$$

to be the Jacobian matrix φ'(a) of the mapping φ at $a \in G$, associated with the pushforward $D\varphi_a$.

7 SLAM Example

Let us examine a visual SLAM example. We have 2D measurements z_{ij}, where each measurement is predicted by

$$z_{ij} = h(T_i, p_j) = \pi(T_i^{-1} p_j)$$

where T_i is the 3D pose of the i^{th} camera, p_j is the location of the j^{th} point, and $\pi : (x,y,z) \mapsto (x/z, y/z)$ is the camera projection function from Example 1.

8 BetweenFactor

BetweenFactor is a factor in GTSAM that is used ubiquitously to process measurements indicating the relative pose between two unknown poses T_1 and T_2. Let us assume the measured relative pose is Z, then the code that calculates the error in ***BetweenFactor*** first calculates the predicted relative pose T_{12}, and then evaluates the error between the measured and predicted relative pose:

```
T12 = between(T1, T2);
return localCoordinates(Z, T12);
```

where we recall that the function ***between*** is given in group theoretic notation as

$$\varphi(g,h) = g^{-1}h$$

The function ***localCoordinates*** itself also calls ***between***, and converts to canonical coordinates:

```
localCoordinates(Z,T12) = Logmap(between(Z, T12));
```

Hence, given two elements T_1 and T_2, ***BetweenFactor*** evaluates $g : G \times G \to \mathbb{R}^n$,

$$g(T_1, T_2; Z) = f^{-1}\left(\varphi(Z, \varphi(T_1, T_2)\right) = f^{-1}\left(Z^{-1}\left(T_1^{-1}T_2\right)\right)$$

where f^{-1} is the inverse of the map $f : \xi \mapsto \exp \hat{\xi}$. If we assume that the measurement has only small error, then $Z \approx T_1^{-1}T_2$, and hence we have $Z^{-1}T_1^{-1}T_2 \approx e$, and we can invoke Theorem 9, which says that the derivative of the exponential map $f : \xi \mapsto \exp \hat{\xi}$ is identity at $\xi = 0$, as well as its inverse.

Finally, because the derivative of ***between*** is identity in its second argument, the derivative of the ***BetweenFactor*** error is identical to the derivative of pushforward of $\varphi(T_1, T_2)$, derived in Section 5.6.

9 Point3

A cross product $a \times b$ can be written as a matrix multiplication

$$a \times b = [a]_\times b$$

where $[a]_\times$ is a skew-symmetric matrix defined as

$$[x, y, z]_\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

We also have

$$a^T [b]_\times = -([b]_\times a)^T = -(a \times b)^T$$

The derivative of a cross product

$$\frac{\partial(a \times b)}{\partial a} = [-b]_\times \tag{9.1}$$

$$\frac{\partial(a \times b)}{\partial b} = [a]_\times \tag{9.2}$$

10 2D Rotations

10.1 Rot2 in GTSAM

A rotation is stored as $(\cos\theta, \sin\theta)$. An incremental rotation is applied using the trigonometric sum rule:

$$\cos\theta' = \cos\theta\cos\delta - \sin\theta\sin\delta$$

$$\sin\theta' = \sin\theta\cos\delta + \cos\theta\sin\delta$$

where δ is an incremental rotation angle.

10.2 Derivatives of Actions

In the case of $SO(2)$ the vector space is $\mathbb{R}^2$, and the group action $f(R,p)$ corresponds to rotating the 2D point p

$$f(R,p) = Rp$$

According to Theorem 3, the Jacobian matrix of f is given by

$$f'(R,p) = \left[\begin{array}{cc} RH(p) & R \end{array}\right]$$

with $H : \mathbb{R}^2 \to \mathbb{R}^{2\times2}$ a linear mapping that depends on p. In the case of $SO(2)$, we can find $H(p)$ by equating (as in Equation 3.1):

$$[w]_+ p = \left[\begin{array}{cc} 0 & -\omega \\ \omega & 0 \end{array}\right]\left[\begin{array}{c} x \\ y \end{array}\right] = \left[\begin{array}{c} -y \\ x \end{array}\right]\omega = H(p)\omega$$

Note that

$$H(p) = \left[\begin{array}{c} -y \\ x \end{array}\right] = \left[\begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array}\right]\left[\begin{array}{c} x \\ y \end{array}\right] = R_{\pi/2}p$$

and since 2D rotations commute, we also have, with $q = Rp$:

$$f'(R,p) = \left[\begin{array}{cc} R\left(R_{\pi/2}p\right) & R \end{array}\right] = \left[\begin{array}{cc} R_{\pi/2}q & R \end{array}\right]$$

10.3 Pushforwards of Mappings

Since $Ad_R[\omega]_+ = [\omega]_+$, we have the derivative of **inverse**,

$$\frac{\partial R^T}{\partial\omega} = -Ad_R = -1$$

compose,

$$\frac{\partial\left(R_1R_2\right)}{\partial\omega_1} = Ad_{R_2^T} = 1 \text{ and } \frac{\partial\left(R_1R_2\right)}{\partial\omega_2} = 1$$

and **between:**

$$\frac{\partial\left(R_1^TR_2\right)}{\partial\omega_1} = -Ad_{R_2^TR_1} = -1 \text{ and } \frac{\partial\left(R_1^TR_2\right)}{\partial\omega_2} = 1$$

11 2D Rigid Transformations

11.1 The derivatives of Actions

The action of $SE(2)$ on 2D points is done by embedding the points in $\mathbb{R}^3$ by using homogeneous coordinates

$$f(T,p)=\hat{q}=\left[\begin{array}{c} q \\ 1 \end{array}\right]=\left[\begin{array}{cc} R & t \\ 0 & 1 \end{array}\right]\left[\begin{array}{c} p \\ 1 \end{array}\right]=T\hat{p}$$

To find the derivative, we write the quantity $\hat{\xi}\hat{p}$ as the product of the 3×3 matrix $H(p)$ with ξ:

$$\hat{\xi}\hat{p}=\left[\begin{array}{cc} [\omega]_+ & v \\ 0 & 0 \end{array}\right]\left[\begin{array}{c} p \\ 1 \end{array}\right]=\left[\begin{array}{c} [\omega]_+p+v \\ 0 \end{array}\right]=\left[\begin{array}{cc} I_2 & R_{\pi/2}p \\ 0 & 0 \end{array}\right]\left[\begin{array}{c} v \\ \omega \end{array}\right]=H(p)\xi \tag{11.1}$$

Hence, by Theorem 3 we have

$$\frac{\partial\left(T\hat{p}\right)}{\partial\xi}=TH(p)=\left[\begin{array}{cc} R & t \\ 0 & 1 \end{array}\right]\left[\begin{array}{cc} I_2 & R_{\pi/2}p \\ 0 & 0 \end{array}\right]=\left[\begin{array}{cc} R & RR_{\pi/2}p \\ 0 & 0 \end{array}\right]=\left[\begin{array}{cc} R & R_{\pi/2}q \\ 0 & 0 \end{array}\right] \tag{11.2}$$

Note that, looking only at the top rows of (11.1) and (11.2), we can recognize the quantity $[\omega]_+p+v=v+\omega\left(R_{\pi/2}p\right)$ as the velocity of p in $\mathbb{R}^2$, and $\left[\begin{array}{cc} R & R_{\pi/2}q \end{array}\right]$ is the derivative of the action on $\mathbb{R}^2$.

The derivative of the inverse action $g(T,p)=T^{-1}\hat{p}$ is given by Theorem 4 specialized to $SE(2)$:

$$\frac{\partial\left(T^{-1}\hat{p}\right)}{\partial\xi}=-H(T^{-1}p)=\left[\begin{array}{cc} -I_2 & -R_{\pi/2}\left(T^{-1}p\right) \\ 0 & 0 \end{array}\right]$$

11.2 Pushforwards of Mappings

We can just define all derivatives in terms of the adjoint map, which in the case of $SE(2)$, in twist coordinates, is the linear mapping

$$Ad_T\xi=\left[\begin{array}{cc} R & -R_{\pi/2}t \\ 0 & 1 \end{array}\right]\left[\begin{array}{c} v \\ \omega \end{array}\right]$$

and we have

$$\frac{\partial T^{^{-1}}}{\partial\xi} = -Ad_T$$

$$\frac{\partial\left(T_1T_2\right)}{\partial\xi_1} = Ad_{T_2^{-1}} \text{ and } \frac{\partial\left(T_1T_2\right)}{\partial\xi_2}=I_3$$

$$\frac{\partial\left(T_1^{-1}T_2\right)}{\partial\xi_1} = -Ad_{T_2^{-1}T_1}=-Ad_{between(T_2,T_1)} \text{ and } \frac{\partial\left(T_1^{-1}T_2\right)}{\partial\xi_2}=I_3$$

12 3D Rotations

12.1 Derivatives of Actions

In the case of $SO(3)$ the vector space is $\mathbb{R}^3$, and the group action $f(R,p)$ corresponds to rotating a point

$$q = f(R,p) = Rp$$

To calculate $H(p)$ for use in Theorem (3) we make use of

$$[\omega]_\times p = \omega \times p = -p \times \omega = [-p]_\times \omega$$

so $H(p) \stackrel{\Delta}{=} [-p]_\times$. Hence, the final derivative of an action in its first argument is

$$\frac{\partial (Rp)}{\partial \omega} = RH(p) = -R[p]_\times \tag{12.1}$$

Likewise, according to Theorem 4, the derivative of the inverse action is given by

$$\frac{\partial \left(R^T p\right)}{\partial \omega} = -H(R^T p) = [R^T p]_\times$$

12.2 Instantaneous Velocity

For 3D rotations R_b^n from a body frame b to a navigation frame n we have the spatial angular velocity ω_{nb}^n measured in the navigation frame,

$$[\omega_{nb}^n]_\times \stackrel{\Delta}{=} \dot{R}_b^n (R_b^n)^T = \dot{R}_b^n R_n^b$$

and the body angular velocity ω_{nb}^b measured in the body frame:

$$[\omega_{nb}^b]_\times \stackrel{\Delta}{=} (R_b^n)^T \dot{R}_b^n = R_n^b \dot{R}_b^n$$

These quantities can be used to derive the velocity of a point p, and we choose between spatial or body angular velocity depending on the frame in which we choose to represent p:

$$v^n = [\omega_{nb}^n]_\times p^n = \omega_{nb}^n \times p^n$$

$$v^b = [\omega_{nb}^b]_\times p^b = \omega_{nb}^b \times p^b$$

We can transform these skew-symmetric matrices from navigation to body frame by conjugating,

$$[\omega_{nb}^b]_\times = R_n^b [\omega_{nb}^n]_\times R_b^n$$

but because the adjoint representation satisfies

$$Ad_R [\omega]_\times \stackrel{\Delta}{=} R[\omega]_\times R^T = [R\omega]_\times$$

we can even more easily transform between spatial and body angular velocities as 3-vectors:

$$\omega_{nb}^b = R_n^b \omega_{nb}^n$$

12.3 Pushforwards of Mappings

For $SO(3)$ we have $Ad_R[\omega]_\times = [R\omega]_\times$ and, in terms of angular velocities: $Ad_R\omega = R\omega$. Hence, the Jacobian matrix of the **inverse** mapping is (see Equation 5.3)

$$\frac{\partial R^T}{\partial \omega} = -Ad_R = -R$$

for **compose** we have (Equations 5.4 and 5.5):

$$\frac{\partial (R_1 R_2)}{\partial \omega_1} = R_2^T \text{ and } \frac{\partial (R_1 R_2)}{\partial \omega_2} = I_3$$

and **between** (Equation 5.6):

$$\frac{\partial \left(R_1^T R_2\right)}{\partial \omega_1} = -R_2^T R_1 = -between(R_2, R_1) \text{ and } \frac{\partial (R_1 R_2)}{\partial \omega_2} = I_3$$

12.4 Retractions

Absil [1, page 58] discusses two possible retractions for $SO(3)$ based on the QR decomposition or the polar decomposition of the matrix $R[\omega]_\times$, but they are expensive. Another retraction is based on the Cayley transform $\mathscr{C} : \mathfrak{so}(3) \to SO(3)$, a mapping from the skew-symmetric matrices to rotation matrices:

$$Q = \mathscr{C}(\Omega) = (I - \Omega)(I + \Omega)^{-1}$$

Interestingly, the inverse Cayley transform $\mathscr{C}^{-1} : SO(3) \to \mathfrak{so}(3)$ has the same form:

$$\Omega = \mathscr{C}^{-1}(Q) = (I - Q)(I + Q)^{-1}$$

The retraction needs a factor $-\frac{1}{2}$ however, to make it locally align with a geodesic:

$$R' = \mathscr{R}_R(\omega) = R\mathscr{C}(-\frac{1}{2}[\omega]_\times)$$

Note that given $\omega = (x, y, z)$ this has the closed-form expression below

$$\frac{1}{4 + x^2 + y^2 + z^2} \left[\begin{array}{ccc} 4 + x^2 - y^2 - z^2 & 2xy - 4z & 2xz + 4y \\ 2xy + 4z & 4 - x^2 + y^2 - z^2 & 2yz - 4x \\ 2xz - 4y & 2yz + 4x & 4 - x^2 - y^2 + z^2 \end{array} \right]$$

$$= \frac{1}{4 + x^2 + y^2 + z^2} \left\{ 4(I + [\omega]_\times) + \left[\begin{array}{ccc} x^2 - y^2 - z^2 & 2xy & 2xz \\ 2xy & -x^2 + y^2 - z^2 & 2yz \\ 2xz & 2yz & -x^2 - y^2 + z^2 \end{array} \right] \right\}$$

so it can be seen to be a second-order correction on $(I + [\omega]_\times)$. The corresponding approximation to the logarithmic map is:

$$[\omega]_\times = \mathscr{R}_R^{-1}(R') = -2\mathscr{C}^{-1}\left(R^T R'\right)$$

13 3D Rigid Transformations

13.1 The derivatives of Actions

The action of $SE(3)$ on 3D points is done by embedding the points in $\mathbb{R}^4$ by using homogeneous coordinates

$$\hat{q} = \begin{bmatrix} q \\ 1 \end{bmatrix} = f(T,p) = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = T\hat{p}$$

The quantity $\hat{\xi}\hat{p}$ corresponds to a velocity in $\mathbb{R}^4$ (in the local T frame), and equating it to $H(p)\xi$ as in Equation 3.1 yields the 4×6 matrix $H(p)$[1]:

$$\hat{\xi}\hat{p} = \begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} \omega \times p + v \\ 0 \end{bmatrix} = \begin{bmatrix} [-p]_\times & I_3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix} = H(p)\xi$$

Note how velocities are analogous to points at infinity in projective geometry: they correspond to free vectors indicating a direction and magnitude of change. According to Theorem 3, the derivative of the group action is then

$$\frac{\partial (T\hat{p})}{\partial \xi} = TH(p) = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} [-p]_\times & I_3 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R[-p]_\times & R \\ 0 & 0 \end{bmatrix}$$

$$\frac{\partial (T\hat{p})}{\partial \hat{p}} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

in homogenous coordinates. In $\mathbb{R}^3$ this becomes $R \begin{bmatrix} -[p]_\times & I_3 \end{bmatrix}$.

The derivative of the inverse action $T^{-1}p$ is given by Theorem 4:

$$\frac{\partial \left(T^{-1}\hat{p}\right)}{\partial \xi} = -H\left(T^{-1}\hat{p}\right) = \begin{bmatrix} [T^{-1}\hat{p}]_\times & -I_3 \end{bmatrix}$$

$$\frac{\partial \left(T^{-1}\hat{p}\right)}{\partial \hat{p}} = \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix}$$

Example 12. Let us examine a visual SLAM example. We have 2D measurements z_{ij}, where each measurement is predicted by

$$z_{ij} = h(T_i, p_j) = \pi(T_i^{-1} p_j) = \pi(q)$$

where T_i is the 3D pose of the i^{th} camera, p_j is the location of the j^{th} point, $q = (x', y', z') = T^{-1}p$ is the point in camera coordinates, and $\pi : (x,y,z) \mapsto (x/z, y/z)$ is the camera projection function from Example 1. By the chain rule, we then have

$$\frac{\partial h(T,p)}{\partial \xi} = \frac{\partial \pi(q)}{\partial q} \frac{\partial (T^{-1}p)}{\partial \xi} = \frac{1}{z'} \begin{bmatrix} 1 & 0 & -x'/z' \\ 0 & 1 & -y'/z' \end{bmatrix} \begin{bmatrix} [q]_\times & -I_3 \end{bmatrix} = \begin{bmatrix} \pi'(q)[q]_\times & -\pi'(q) \end{bmatrix}$$

$$\frac{\partial h(T,p)}{\partial p} = \pi'(q) R^T$$

[1] $H(p)$ can also be obtained by taking the j^{th} column of each of the 6 generators to multiply with components of $\hat{p}$

13.2 Instantaneous Velocity

For rigid 3D transformations T_b^n from a body frame b to a navigation frame n we have the instantaneous spatial twist ξ_{nb}^n measured in the navigation frame,

$$\hat{\xi}_{nb}^n \triangleq \dot{T}_b^n \left(T_b^n\right)^{-1}$$

and the instantaneous body twist ξ_{nb}^b measured in the body frame:

$$\hat{\xi}_{nb}^b \triangleq \left(T_b^n\right)^T \dot{T}_b^n$$

13.3 Pushforwards of Mappings

As we can express the Adjoint representation in terms of twist coordinates, we have

$$\begin{bmatrix} \omega' \\ v' \end{bmatrix} = \begin{bmatrix} R & 0 \\ [t]_\times R & R \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix}$$

Hence, as with $SO(3)$, we are now in a position to simply posit the derivative of **inverse**,

$$\frac{\partial T^{-1}}{\partial \xi} = -Ad_T = -\begin{bmatrix} R & 0 \\ [t]_\times R & R \end{bmatrix}$$

compose in its first argument,

$$\frac{\partial \left(T_1 T_2\right)}{\partial \xi_1} = Ad_{T_2^{-1}}$$

in its second argument,

$$\frac{\partial \left(T_1 T_2\right)}{\partial \xi_2} = I_6$$

between in its first argument,

$$\frac{\partial \left(T_1^{-1} T_2\right)}{\partial \xi_1} = -Ad_{T_2^{-1} T_1} = \begin{bmatrix} -R_2^T R_1 & 0 \\ R_2^T \left[t_2 - t_1\right]_\times R_1 & -R_2^T R_1 \end{bmatrix}$$

and in its second argument,

$$\frac{\partial \left(T_1^{-1} T_2\right)}{\partial \xi_1} = I_6$$

13.4 Retractions

For $SE(3)$, instead of using the true exponential map it is computationally more efficient to design other retractions. A first-order approximation to the exponential map does not quite cut it, as it yields a 4×4 matrix which is not in $SE(3)$:

$$\begin{aligned}
T\exp\hat{\xi} &\approx T(I+\hat{\xi}) \\
&= T\left(I_4 + \begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix}\right) \\
&= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} I_3+[\omega]_\times & v \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} R(I_3+[\omega]_\times) & t+Rv \\ 0 & 1 \end{bmatrix}
\end{aligned}$$

However, we can make it into a retraction by using any retraction defined for $SO(3)$, including, as below, using the exponential map $Re^{[\omega]_\times}$:

$$\mathscr{R}_T\left(\begin{bmatrix} \omega \\ v \end{bmatrix}\right) = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} e^{[\omega]_\times} & v \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Re^{[\omega]_\times} & t+Rv \\ 0 & 1 \end{bmatrix}$$

Similarly, for a second order approximation we have

$$\begin{aligned}
T\exp\hat{\xi} &\approx T(I+\hat{\xi}+\frac{\hat{\xi}^2}{2}) \\
&= T\left(I_4 + \begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix}\begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix}\right) \\
&= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}\left(\begin{bmatrix} I_3+[\omega]_\times+\frac{1}{2}[\omega]_\times^2 & v+\frac{1}{2}[\omega]_\times v \\ 0 & 1 \end{bmatrix}\right) \\
&= \begin{bmatrix} R\left(I_3+[\omega]_\times+\frac{1}{2}[\omega]_\times^2\right) & t+R[v+(\omega\times v)/2] \\ 0 & 1 \end{bmatrix}
\end{aligned}$$

inspiring the retraction

$$\mathscr{R}_T\left(\begin{bmatrix} \omega \\ v \end{bmatrix}\right) = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} e^{[\omega]_\times} & v+(\omega\times v)/2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Re^{[\omega]_\times} & t+R[v+(\omega\times v)/2] \\ 0 & 1 \end{bmatrix}$$

14 The Sphere S^2

14.1 Definitions

The sphere S^2 is the set of all unit vectors in $\mathbb{R}^3$, i.e., all directions in three-space:

$$S^2 = \{p \in \mathbb{R}^3 | \, \|p\| = 1\}$$

The tangent space T_pS^2 at a point$p$ consists of three-vectors $\hat{\xi}$ such that $\hat{\xi}$ is tangent to S^2 at p, i.e.,

$$T_pS^2 \stackrel{\Delta}{=} \left\{ \hat{\xi} \in \mathbb{R}^3 | p^T \hat{\xi} = 0 \right\}$$

While not a Lie group, we can define an exponential map, which is given in Ma et. al [3], as well as in this CVPR tutorial by Anuj Srivastava: http://stat.fsu.edu/∼anuj/CVPR_Tutorial/Part2.pdf.

$$\exp_p \hat{\xi} = \cos\left(\left\|\hat{\xi}\right\|\right) p + \sin\left(\left\|\hat{\xi}\right\|\right) \frac{\hat{\xi}}{\left\|\hat{\xi}\right\|}$$

The latter also gives the inverse, i.e., get the tangent vector z to go from p to q:

$$z = \log_p q = \frac{\theta}{\sin\theta} (q - p\cos\theta) p$$

with $\theta = \cos^{-1}\left(p^T q\right)$.

14.2 Local Coordinates

We can find a basis B_p for the tangent space T_pS^2, with $B_p = [b_1 | b_2]$ a 3×2 matrix, by either

1. Decompose $p = QR$, with Q orthonormal and R of the form $[1\,0\,0]^T$, and hence $p = Q_1$. The basis $B_p = [Q_2 | Q_3]$, i.e., the last two columns of Q.
2. Form $b_1 = p \times a$, with a (consistently) chosen to be non-parallel to p, and $b_2 = p \times b_1$.

Now we can write $\hat{\xi} = B_p \xi$ with $\xi \in R^2$ the 2D coordinate in the tangent plane basis B_p.

14.3 Retraction

The exponential map uses cos and sin, and is more than we need for optimization. Suppose we have a point $p \in S^2$ and a 3-vector $\hat{\xi} \in T_pS^2$, Absil [1] tells us we can simply add $\hat{\xi}$ to p and renormalize to get a new point q on the sphere. This is what he calls a **retraction** $\mathscr{R}_p(\hat{\xi})$,

$$q = \mathscr{R}_p(\hat{\xi}) = \frac{p + \hat{\xi}}{\|p + z\|} = \frac{p + \hat{\xi}}{\alpha}$$

with α the norm of $p + \hat{\xi}$.

We can also define a retraction from local coordinates $\xi \in \mathbb{R}^2$:

$$q = \mathscr{R}_p(\xi) = \frac{p + B_p\xi}{\left\|p + B_p\xi\right\|} = \frac{p + B_p\xi}{\alpha}$$

Since $\hat{\xi}$ is in the tangent space T_pS^2 at p, we have $p^T \hat{\xi} = 0$.

Inverse Retraction

If $\hat{\xi} = B_p \xi$ with $\xi \in R^2$ the 2D coordinate in the tangent plane basis B_p, we have

$$\xi = \frac{B^T q}{p^T q}$$

Proof. We seek

$$\alpha q = p + B_p \xi$$

If we multiply both sides with B_p^T (project on the basis B_p) we obtain

$$\alpha B_p^T q = B_p^T p + B_p^T B_p \xi$$

and because $B_p^T p = 0$ and $B_p^T B_p = I$ we trivially obtain ξ as the scaled projection $B^T q$:

$$\xi = \alpha B^T q$$

To recover the scale factor α we multiply with p^T on both sides we have

$$\alpha p^T q = p^T p + p^T B_p \xi$$

and (since $p^T p = 1$ and $p^T B_p \xi = 0$) we have $\alpha = 1/(p^T q)$. □

15 The Essential Matrix Manifold

We parameterize essential matrices as a pair (R, t), where $R \in SO(3)$ and $t \in S^2$, the unit sphere. The epipolar matrix is then given by

$$E = [t]_\times R$$

and the epipolar error given two corresponding points a and b is

$$e(R, t; a, b) = a^T E b$$

We are of course interested in the derivative with respect to orientation (using 12.1)

$$\frac{\partial (a^T [t]_\times R b)}{\partial \omega} = a^T [t]_\times \frac{\partial (Rb)}{\partial \omega} = -a^T [t]_\times R [b]_\times = -a^T E [b]_\times$$

and with respect to change in the direction t

$$\frac{\partial e(a^T [t]_\times R b)}{\partial \xi} = a^T \frac{\partial (B\xi \times Rb)}{\partial v} = -a^T [Rb]_\times B$$

where we made use of the fact that the retraction can be written as $t + B\xi$, with B a local basis, and we made use of (9.1):

$$\frac{\partial (a \times b)}{\partial a} = [-b]_\times$$

16 2D Line Segments (Ocaml)

The error between an infinite line (a,b,c) and a 2D line segment $((x1,y1),(x2,y2))$ is defined in Line3.ml.

17 Line3vd (Ocaml)

One representation of a line is through 2 vectors (v,d), where v is the direction and the vector d points from the orgin to the closest point on the line.

In this representation, transforming a 3D line from a world coordinate frame to a camera at (R_w^c, t^w) is done by

$$v^c = R_w^c v^w$$

$$d^c = R_w^c \left(d^w + (t^w v^w) v^w - t^w\right)$$

18 Line3 (Ocaml)

For 3D lines, we use a parameterization due to C.J. Taylor, using a rotation matrix R and 2 scalars a and b. The line direction v is simply the Z-axis of the rotated frame, i.e., $v = R_3$, while the vector d is given by $d = aR_1 + bR_2$.

Now, we will *not* use the incremental rotation scheme we used for rotations: because the matrix R translates from the line coordinate frame to the world frame, we need to apply the incremental rotation on the right-side:

$$R' = R(I + \Omega)$$

Projecting a line to 2D can be done easily, as both v and d are also the 2D homogenous coordinates of two points on the projected line, and hence we have

$$\begin{aligned} l &= v \times d \\ &= R_3 \times (aR_1 + bR_2) \\ &= a(R_3 \times R_1) + b(R_3 \times R_2) \\ &= aR_2 - bR_1 \end{aligned}$$

This can be written as a rotation of a point,

$$l = R \begin{pmatrix} -b \\ a \\ 0 \end{pmatrix}$$

but because the incremental rotation is now done on the right, we need to figure out the derivatives again:

$$\frac{\partial (R(I+\Omega)x)}{\partial \omega} = \frac{\partial (R\Omega x)}{\partial \omega} = R\frac{\partial (\Omega x)}{\partial \omega} = R[-x]_\times \tag{18.1}$$

and hence the derivative of the projection l with respect to the rotation matrix $R$of the 3D line is

$$\frac{\partial(l)}{\partial\omega} = R[\left(\begin{array}{c} b \\ -a \\ 0 \end{array}\right)]_{\times} = \left[\begin{array}{ccc} aR_3 & bR_3 & -(aR_1 + bR_2) \end{array}\right] \tag{18.2}$$

or the a, b scalars:

$$\frac{\partial(l)}{\partial a} = R_2$$
$$\frac{\partial(l)}{\partial b} = -R_1$$

Transforming a 3D line $(R, (a, b))$ from a world coordinate frame to a camera frame (R_w^c, t^w) is done by

$$R' = R_w^c R$$
$$a' = a - R_1^T t^w$$
$$b' = b - R_2^T t^w$$

Again, we need to redo the derivatives, as R is incremented from the right. The first argument is incremented from the left, but the result is incremented on the right:

$$\begin{array}{rcl} R'(I+\Omega') = (AB)(I+\Omega') & = & (I+[S\omega]_{\times})AB \\ I+\Omega' & = & (AB)^T(I+[S\omega]_{\times})(AB) \\ \Omega' & = & R'^T[S\omega]_{\times}R' \\ \Omega' & = & [R'^T S\omega]_{\times} \\ \omega' & = & R'^T S\omega \end{array}$$

For the second argument R we now simply have:

$$\begin{array}{rcl} AB(I+\Omega') & = & AB(I+\Omega) \\ \Omega' & = & \Omega \\ \omega' & = & \omega \end{array}$$

The scalar derivatives can be found by realizing that

$$\left(\begin{array}{c} a' \\ b' \\ \ldots \end{array}\right) = \left(\begin{array}{c} a \\ b \\ 0 \end{array}\right) - R^T t^w$$

where we don't care about the third row. Hence

$$\frac{\partial((R(I+\Omega_2))^T t^w)}{\partial\omega} = -\frac{\partial(\Omega_2 R^T t^w)}{\partial\omega} = -[R^T t^w]_{\times} = \left[\begin{array}{ccc} 0 & R_3^T t^w & -R_2^T t^w \\ -R_3^T t^w & 0 & R_1^T t^w \\ \ldots & \ldots & 0 \end{array}\right]$$

19 Aligning 3D Scans

Below is the explanation underlying Pose3.align, i.e. aligning two point clouds using SVD. Inspired but modified from CVOnline...

Our model is

$$p^c = R\left(p^w - t\right)$$

i.e., R is from camera to world, and t is the camera location in world coordinates. The objective function is

$$\frac{1}{2}\sum\left(p^c - R(p^w - t)\right)^2 = \frac{1}{2}\sum\left(p^c - Rp^w + Rt\right)^2 = \frac{1}{2}\sum\left(p^c - Rp^w - t'\right)^2 \tag{19.1}$$

where $t' = -Rt$ is the location of the origin in the camera frame. Taking the derivative with respect to t' and setting to zero we have

$$\sum\left(p^c - Rp^w - t'\right) = 0$$

or

$$t' = \frac{1}{n}\sum\left(p^c - Rp^w\right) = \bar{p}^c - R\bar{p}^w \tag{19.2}$$

here $\bar{p}^c$ and $\bar{p}^w$ are the point cloud centroids. Substituting back into (19.1), we get

$$\frac{1}{2}\sum\left(p^c - R(p^w - t)\right)^2 = \frac{1}{2}\sum\left((p^c - \bar{p}^c) - R\left(p^w - \bar{p}^w\right)\right)^2 = \frac{1}{2}\sum\left(\hat{p}^c - R\hat{p}^w\right)^2$$

Now, to minimize the above it suffices to maximize (see CVOnline)

$$trace\left(R^T C\right)$$

where $C = \sum \hat{p}^c \left(\hat{p}^w\right)^T$ is the correlation matrix. Intuitively, the cloud of points is rotated to align with the principal axes. This can be achieved by SVD decomposition on C

$$C = USV^T$$

and setting

$$R = UV^T$$

Clearly, from (19.2) we then also recover the optimal t as

$$t = \bar{p}^w - R^T \bar{p}^c$$

Appendix

Differentiation Rules

Spivak [5] also notes some multivariate derivative rules defined component-wise, but they are not that useful in practice:

- Since $f : \mathbb{R}^n \to \mathbb{R}^m$ is defined in terms of m component functions f^i, then f is differentiable at a iff each f^i is, and the Jacobian matrix F_a is the $m \times n$ matrix whose i^{th} row is $\left(f^i\right)'(a)$:

$$F_a \triangleq f'(a) = \begin{bmatrix} \left(f^1\right)'(a) \\ \vdots \\ \left(f^m\right)'(a) \end{bmatrix}$$

- Scalar differentiation rules: if $f, g : \mathbb{R}^n \to \mathbb{R}$ are differentiable at a, then

$$(f+g)'(a) = F_a + G_a$$
$$(f \cdot g)'(a) = g(a)F_a + f(a)G_a$$
$$(f/g)'(a) = \frac{1}{g(a)^2}\left[g(a)F_a - f(a)G_a\right]$$

Tangent Spaces and the Tangent Bundle

The following is adapted from Appendix A in [4].

The **tangent space** T_pM of a manifold M at a point $p \in M$ is the vector space of **tangent vectors** at p. The **tangent bundle** TM is the set of all tangent vectors

$$TM \triangleq \bigcup_{p \in M} T_pM$$

A **vector field** $X : M \to TM$ assigns a single tangent vector $x \in T_pM$ to each point p.

If $F : M \to N$ is a smooth map from a manifold M to a manifold N, then we can define the **tangent map** of F at p as the linear map $F_{*p} : T_pM \to T_{F(p)}N$ that maps tangent vectors in T_pM at p to tangent vectors in $T_{F(p)}N$ at the image $F(p)$.

Homomorphisms

The following *might be* relevant [2, page 45]: suppose that $\Phi : G \to H$ is a mapping (Lie group homomorphism). Then there exists a unique linear map $\phi : \mathfrak{g} \to \mathfrak{h}$

$$\phi(\hat{x}) \triangleq \lim_{t \to 0} \frac{d}{dt} \Phi\left(e^{t\hat{x}}\right)$$

such that

1. $\Phi\left(e^{\hat{x}}\right) = e^{\phi(\hat{x})}$
2. $\phi\left(T\hat{x}T^{-1}\right) = \Phi(T)\phi(\hat{x})\Phi(T^{-1})$
3. $\phi\left([\hat{x}, \hat{y}]\right) = [\phi(\hat{x}), \phi(\hat{y})]$

In other words, the map ϕ is the derivative of Φ at the identity. As an example, suppose $\Phi(g) = g^{-1}$, then the corresponding derivative *at the identity* is

$$\phi(\hat{x}) \triangleq \lim_{t \to 0} \frac{d}{dt} \left(e^{t\hat{x}}\right)^{-1} = \lim_{t \to 0} \frac{d}{dt} e^{-t\hat{x}} = -\hat{x} \lim_{t \to 0} e^{-t\hat{x}} = -\hat{x}$$

In general it suffices to compute ϕ for a basis of $\mathfrak{g}$.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2007.

[2] B.C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer, 2000.

[3] Yi Ma, Jana Košecká, and Shankar Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *Intl. J. of Computer Vision*, 44(3):219–249, 2001.

[4] R.M. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[5] M. Spivak. *Calculus on manifolds*. Benjamin New York;, 1965.