LITAMIN2 derivatives of Cost function

Xiaoguo Du (bzdfzfer)

January 2022

1 Basic principle of Matrix differential

First, two types of matrix differential definitions are introduced. Then, some useful rules are discussed for later derivation of Jacobian and Hessian matrix. These operations are essential for calculating the partial derivatives (i.e. Jacobian Matrix) and the second-order derivatives (i.e. Hessian Matrix).

1.1 Matrix differential definition

This definition is adopted from Jan R. Magnus's "On the concept of matrix derivative". To define matrix differential, we need to introduce two operators first.

1.1.1 Tensor product (Kronecker product)

If A is an $m \times n$ matrix and B is a pq matrix, then the Kronecker product $A \otimes B$ is the $pm \times qn$ block matrix:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \tag{1}$$

more explicitly:

$$\mathbf{A} \otimes \mathbf{B} =$$
$$\begin{bmatrix}
a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\
a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\
\vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\
a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\
\vdots & \vdots & & \vdots & \ddots & \vdots & \vdots & & \vdots \\
a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\
a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\
\vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\
a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq}
\end{bmatrix} \tag{2}$$

1.1.2 "vec" Matrix operator

This operator turns a Matrix M with size of $m \times n$ (m rows and n columns) into a $mn \times 1$ column vector, by concatenating each $m \times 1$ column vector into one column, from the first column to the n-th column. That is:

$$vec(M_{m\times n}) = \begin{pmatrix} M_{[:,1]} \\ M_{[:,2]} \\ \vdots \\ M_{[:,n]} \end{pmatrix}_{mn\times 1} \tag{3}$$

1.1.3 ω-derivative

Consider a matrix function F of a matrix of variables X, $F \in R^{m\times p}$ and $X \in R^{n\times q}$, any $r \times c$ matrix satisfying $rc = mnpq$ and containing all partial derivatives of $\partial F_{st}/\partial X_{ij}$ is called a derivative of F with respect to X. This is called ω-derivative. This definition does not consider the organization of the partial derivatives. For example,

$$\frac{\partial F(X)}{\partial X} = \frac{\partial}{\partial X} \otimes F \tag{4}$$

More explicitly,

$$\begin{aligned}
\frac{\partial F(X)}{\partial X} &= \begin{bmatrix} \frac{\partial}{\partial X_{11}} & \frac{\partial}{\partial X_{12}} & \cdots & \frac{\partial}{\partial X_{1q}} \\ \frac{\partial}{\partial X_{21}} & \frac{\partial}{\partial X_{22}} & \cdots & \frac{\partial}{\partial X_{2q}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial X_{n1}} & \frac{\partial}{\partial X_{n2}} & \cdots & \frac{\partial}{\partial X_{nq}} \end{bmatrix} \otimes F \\
&= \begin{bmatrix} \frac{\partial F}{\partial X_{11}} & \frac{\partial F}{\partial X_{12}} & \cdots & \frac{\partial F}{\partial X_{1q}} \\ \frac{\partial F}{\partial X_{21}} & \frac{\partial F}{\partial X_{22}} & \cdots & \frac{\partial F}{\partial X_{2q}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F}{\partial X_{n1}} & \frac{\partial F}{\partial X_{n2}} & \cdots & \frac{\partial F}{\partial X_{nq}} \end{bmatrix} \\
&= \begin{bmatrix}
\frac{\partial F_{11}}{\partial X_{11}} & \frac{\partial F_{12}}{\partial X_{11}} & \cdots & \frac{\partial F_{1p}}{\partial X_{11}} & \cdots & \frac{\partial F_{11}}{\partial X_{1q}} & \frac{\partial F_{12}}{\partial X_{1q}} & \cdots & \frac{\partial F_{1p}}{\partial X_{1q}} \\
\frac{\partial F_{21}}{\partial X_{11}} & \frac{\partial F_{22}}{\partial X_{11}} & \cdots & \frac{\partial F_{2p}}{\partial X_{11}} & \cdots & \frac{\partial F_{21}}{\partial X_{1q}} & \frac{\partial F_{22}}{\partial X_{1q}} & \cdots & \frac{\partial F_{2p}}{\partial X_{1q}} \\
\vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial F_{m1}}{\partial X_{11}} & \frac{\partial F_{m2}}{\partial X_{11}} & \cdots & \frac{\partial F_{mp}}{\partial X_{11}} & \cdots & \frac{\partial F_{m1}}{\partial X_{1q}} & \frac{\partial F_{m2}}{\partial X_{1q}} & \cdots & \frac{\partial F_{mp}}{\partial X_{1q}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial F_{11}}{\partial X_{n1}} & \frac{\partial F_{12}}{\partial X_{n1}} & \cdots & \frac{\partial F_{1p}}{\partial X_{n1}} & \cdots & \frac{\partial F_{11}}{\partial X_{nq}} & \frac{\partial F_{12}}{\partial X_{nq}} & \cdots & \frac{\partial F_{1p}}{\partial X_{nq}} \\
\frac{\partial F_{21}}{\partial X_{n1}} & \frac{\partial F_{22}}{\partial X_{n1}} & \cdots & \frac{\partial F_{2p}}{\partial X_{n1}} & \cdots & \frac{\partial F_{21}}{\partial X_{nq}} & \frac{\partial F_{22}}{\partial X_{nq}} & \cdots & \frac{\partial F_{2p}}{\partial X_{nq}} \\
\vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial F_{m1}}{\partial X_{n1}} & \frac{\partial F_{m2}}{\partial X_{n1}} & \cdots & \frac{\partial F_{mp}}{\partial X_{n1}} & \cdots & \frac{\partial F_{m1}}{\partial X_{nq}} & \frac{\partial F_{m2}}{\partial X_{nq}} & \cdots & \frac{\partial F_{mp}}{\partial X_{nq}}
\end{bmatrix}
\end{aligned} \tag{5}$$

1.1.4 α-derivative

Above definition is not convenient to use for example, the Hessian matrix computation and Newton iteration. To ease the derivation, *vec* operations are used to make the expression well organized, which is called α-derivative. Here two common types are frequently used in the research.

Type 1: Numerator-layout notation. This definition keeps the resulting derivative's width the same as numerator's size, then the height keeps the same as denominator's size. That is:

$$D_\alpha F(X) = \frac{\partial vecF(X)}{\partial (vecX)^T} \tag{6}$$

Type 2: Denominator-layout notation.

$$D_\omega F(X) = \frac{\partial vecF^T(X)}{\partial (vecX)^T} \tag{7}$$

1.1.5 Correlation between two α-derivatives

There is no essential difference between those two definitions. The results can be easily converted to the other by matrix transpose operation.

In this report, **I use type 1 α-derivative to calculate Jacobian and Hessian, which can unify the chain rule expression for all dimensions**.

1.2 Useful Rules

1.2.1 chain rule of matrix differential

The chain rule is an essential ingredient without which matrix calculus cannot exist. For $H(X) = G(F(X))$, $Y = F(X)$

$$DH(X) = (DG(Y))(DF(X)) \tag{8}$$

1.2.2 product rule of matrix differential

Let$F(mp)$ and $G(pr)$ be functions of $X(nq)$. Then the product rule for differentials is simply

$$d(FG) = (dF)G + F(dG) \tag{9}$$

Applying the vec-operator gives

$$dvec(FG) = (G^T \otimes I_m)dvec(F) + (I_r \otimes F)dvec(G) \tag{10}$$

The product rule for α-derivatives is therefore

$$D(FG) = (G^T \otimes I_m)DF + (I_r \otimes F)DG \tag{11}$$

1.2.3 trace derivatives

(This part is about to be summarized.)

2 Energy function Definition

LiTAMIN2 models similarity distance of two Gaussian distributions with KL-Divergence, which is described as:

$$\begin{aligned} D_{KL}(p||q) = \int p(x) log \frac{p(x)}{q(x)} dx \propto \\ (\mu_p - \mu_q)^T C_q^{-1} (\mu_p - \mu_q) + Tr(C_q^{-1} C_p) - d + log \frac{C_q}{C_p} \end{aligned} \tag{12}$$

In (1), two distribution are represented with numerical characteristics of random variables, i.e. $P(X) \sim N(\mu_p, C_p)$, $Q(X) \sim N(\mu_q, C_q)$. Operator Tr means trace of matrix, and d is dimension of X. The introduction of KL-Divergence aims to take into account the shape of the distribution (i.e. co-variance).

However, above divergence is not symmetric, i.e. $D_{KL}(p||q) \neq D_{KL}(q||p)$, which means it is not a distance. Therefore, a symmetric KL-Divergence is used as:

$$\begin{aligned} & D_{symKL}(p||q) = \\ & (\mu_p - \mu_q)^T (C_p + C_q)^{-1} (\mu_p - \mu_q) + Tr(C_q^{-1} C_p) + Tr(C_p^{-1} C_q) - 2d \end{aligned} \tag{13}$$

This metric is suitable for the distance minimization problem of ICP. Observing (2), the first term is the Mahalanobis distance between the target point cloud distribution and the reference distribution. The rest terms are co-variances similarities: when two distributions are indistinguishable, i.e. $C_q = C_p$, these product results are Identiy matrices I_3, whose trace equals to d and thus the rest terms are 0; when two distributions are dissimilar, those terms are larger than 0???? why???

The ICP cost is designed as:

$$E_{ICP} = (q - (Rp + t))^T \frac{(C_q + RC_pR^T + \lambda I)^{-1}}{||(C_q + RC_pR^T + \lambda I)^{-1}||_F} (q - (Rp + t)) \tag{14}$$

The covariance cost is set as:

$$E_{COV} = Tr(RC_p^{-1}R^TC_q) + Tr(C_q^{-1}RC_pR^T) - 6 \tag{15}$$

So, the total cost function is:

$$E = \Sigma(w_{ICP,i}E_{ICP,i} + w_{COV,i}E_{COV,i}) \tag{16}$$

The weight is obtained according to each error:

$$w_{ICP} = 1 - \frac{E_{ICP}}{E_{ICP} + \sigma_{ICP}^2} \tag{17}$$

$$w_{COV} = 1 - \frac{E_{COV}}{E_{COV} + \sigma_{COV}^2} \tag{18}$$

According to paper, Newton method was used to optimize the above cost function. This is because the second cost function is not quadratic, whose Hessian matrix cannot be approximated with Jacobian matrix. Thus, it needs to be found up the Hessian of the second term, i.e. covariance cost function E_{COV}.

3 Jacobian and Hessian of ICP Cost function

3.1 Rewritting cost function

This part is relatively easy. Here we assume the covariance matrix of C_{qp} is constant, and the Jacobian matrix of E_{ICP} with respect to transformation variables $se(3)$ is to be solved. The weight matrix is defined as:

$$W = C_{qp}^{-1} = \frac{(C_q + RC_pR^T + \lambda I)^{-1}}{||(C_q + RC_pR^T + \lambda I)^{-1}||_F} \tag{19}$$

Then, the ICP cost function is:

$$f(\xi) = E_{ICP}(\xi) = e_T We \tag{20}$$

where

$$e(\xi) = q - T(\xi)p \tag{21}$$

and

$$T(\xi) = \begin{pmatrix} R & t \\ \mathbf{0}^T & 1 \end{pmatrix}, \xi = (\phi, \rho)^T \in se(3) \tag{22}$$

3.2 Solving Jacobian

The Jacobian matrix is:

$$J_f(\xi) = \frac{\partial f}{\partial \xi} = \frac{\partial f}{\partial e}\frac{\partial e}{\partial \xi} \tag{23}$$

where

$$\frac{\partial f}{\partial e} = e^T(W + W^T) \tag{24}$$

Since the information matrix is symmetric, i.e. $W = W^T$, this derivative is $\frac{\partial f}{\partial e} = 2e^TW$ Note that $\frac{\partial f}{\partial e} \in R^{1\times 3}$.

The second part, when applying left perturbation on $se(3)$, this derivative can be obtained as:

$$\frac{\partial e}{\partial \xi} = \frac{T(\xi)p}{\partial \xi} = \begin{pmatrix} [Rp + t]_\times & -I_3 \end{pmatrix} \tag{25}$$

. Note that $\frac{\partial e}{\partial \xi} \in R^{3\times 6}$. Let $J_e(\xi) = \frac{\partial e}{\partial \xi}$.

So, the Jacobian matrix can be computed as:

$$J_f(\xi) = 2e^TWJ_e(\xi) = 2e^TW \begin{pmatrix} [Rp + t]_\times & -I_3 \end{pmatrix} \tag{26}$$

3.3 Solving Hessian

The Hessian matrix:

$$H_f(\xi) = \frac{\partial J_f(\xi)}{\partial \xi} = \frac{\partial(\frac{\partial f}{\partial e}\frac{\partial e}{\partial \xi})}{\partial \xi} \tag{27}$$

Thus we apply rule of the product of two functions to derive the Hessian:

$$H_f(\xi) = (\frac{\partial e}{\partial \xi}^T \otimes I_1)\frac{\partial^2 f}{\partial e^2}\frac{\partial e}{\partial \xi} + (I_6 \otimes \frac{\partial f}{\partial e})\frac{\partial^2 e}{\partial \xi^2} \tag{28}$$

3.3.1 Solving Hessian of $f(e)$

The Hessian matrix of $f(e)$ with respect to e is $2W$, that is:

$$\frac{\partial^2 f}{\partial e^2} = 2W \tag{29}$$

When ignoring the second term of $H_f(\xi)$, it turns as Gauss-Newton Hessian:

$$H_f(\xi) = \frac{\partial e}{\partial \xi}^T \frac{\partial^2 f}{\partial e^2}\frac{\partial e}{\partial \xi} \tag{30}$$

Inject $J_e(\xi)$:

$$H_f(\xi) = 2J_e^T(\xi)WJ_e(\xi) \tag{31}$$

The Gauss-Newton Iteration step is:

$$H\delta\xi = -J^T \tag{32}$$

$$J_e^T(\xi)WJ_e(\xi) * \delta\xi = -J_e^T(\xi)We \tag{33}$$

3.3.2 Solving Hessian of $e(\xi)$

If we do not ignore the second term of $H_f(\xi)$, we need to compute the Hessian of $e(\xi)$. The Hessian matrix of $e(\xi)$ with respect to ξ is denoted by $H_e(\xi)$. That is:

$$H_e(\xi) = \frac{\partial J_e(\xi)}{\partial \xi} \tag{34}$$

Recall that:

$$J_e(\xi) = \begin{pmatrix} [Rp+t]_\times & -I_3 \end{pmatrix} \tag{35}$$

and $J_e(\xi) \in R^{3\times 6}$. Therefore, $H_e(\xi) = M \in R^{18\times 6}$.

Observing right three columns of $J_e(\xi)$, we find that these elements are constant values and thus their derivatives are all zeros. So, $M_{[10:18,1:6]} = \mathbf{0}_{9\times 6}$.

Let $s(\xi) = Rp + t = (s_1, s_2, s_3)^T$, and the left three columns of $J_e(\xi)$ are denoted as:

$$\begin{pmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{pmatrix} \tag{36}$$

We can calculate derivative of vector $s(\xi)$ firstly, and then assign rows of $H_e(\xi)$ accordingly.

$$\frac{\partial s(\xi)}{\partial \xi} = -J_e(\xi) = \begin{pmatrix} -[Rp+t]_\times & I_3 \end{pmatrix} \tag{37}$$

This part is denoted as $S = \frac{\partial s(\xi)}{\partial \xi}$ and $S \in R^{3\times 6}$.

Therefore,

$$\begin{aligned} M_{[2,1:6]} &= S_{[3,1:6]} \\ M_{[3,1:6]} &= -S_{[2,1:6]} \\ M_{[4,1:6]} &= -S_{[3,1:6]} \\ M_{[6,1:6]} &= S_{[1,1:6]} \\ M_{[7,1:6]} &= S_{[2,1:6]} \\ M_{[8,1:6]} &= -S_{[1,1:6]} \end{aligned} \tag{38}$$

The rest rows of M are zero vectors. That is:

$$M = \begin{pmatrix} \mathbf{0}_{1\times 6} \\ -J_{e,[3,1:6]} \\ J_{e,[2,1:6]} \\ J_{e,[3,1:6]} \\ \mathbf{0}_{1\times 6} \\ -J_{e,[1,1:6]} \\ -J_{e,[2,1:6]} \\ J_{e,[1,1:6]} \\ \mathbf{0}_{1\times 6} \\ \mathbf{0}_{9\times 6} \end{pmatrix} \tag{39}$$

To calculate the second part of hessian matrix $H_f(\xi)$, we rewrite it as:

$$\begin{aligned}
{}_2H_f(\xi) &= \{I_6 \otimes \frac{\partial f}{\partial e}\}H_e(\xi) \\
&= \{I_6 \otimes e^T W\}M \\
&= \begin{pmatrix} e^TW & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & e^TW & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & e^TW & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & e^TW & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & e^TW & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & e^TW \end{pmatrix} \begin{pmatrix} \mathbf{0}_{1\times6} \\ -J_{e,[3,1:6]} \\ J_{e,[2,1:6]} \\ J_{e,[3,1:6]} \\ \mathbf{0}_{1\times6} \\ -J_{e,[1,1:6]} \\ -J_{e,[2,1:6]} \\ J_{e,[1,1:6]} \\ \mathbf{0}_{1\times6} \\ \mathbf{0}_{9\times6} \end{pmatrix} \\
&= \begin{pmatrix} e^TW & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & e^TW & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & e^TW \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \end{pmatrix} \begin{pmatrix} \mathbf{0}_{1\times6} \\ -J_{e,[3,1:6]} \\ J_{e,[2,1:6]} \\ J_{e,[3,1:6]} \\ \mathbf{0}_{1\times6} \\ -J_{e,[1,1:6]} \\ -J_{e,[2,1:6]} \\ J_{e,[1,1:6]} \\ \mathbf{0}_{1\times6} \end{pmatrix} = \begin{pmatrix} N_{3\times6} \\ \mathbf{0}_{3\times6} \end{pmatrix}
\end{aligned} \tag{40}$$

where:

$$\begin{aligned}
\begin{pmatrix} N_{3\times6} \end{pmatrix} &= \begin{pmatrix} e^TW & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & e^TW & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & e^TW \end{pmatrix} M_{[1:9,1:6]} \\
&= \begin{pmatrix} e^TW \begin{pmatrix} \mathbf{0}_{1\times6} \\ -J_{e,[3,:]} \\ J_{e,[2,:]} \end{pmatrix} \\ e^TW \begin{pmatrix} J_{e,[3,:]} \\ \mathbf{0}_{1\times6} \\ -J_{e,[1,:]} \end{pmatrix} \\ e^TW \begin{pmatrix} -J_{e,[2,:]} \\ J_{e,[1,:]} \\ \mathbf{0}_{1\times6} \end{pmatrix} \end{pmatrix}
\end{aligned} \tag{41}$$

This results in 3×6 upper half of Hessian matrix ${}_2H_f(\xi)$. This means no extra derivatives are needed here, but only a re-organization of known Jacobian matrices of $\frac{\partial f}{\partial e}$ and $\frac{\partial e}{\partial \xi}$ is enough for the hessian computation.

However, this part's second derivatives leads to asymmetric Hessian matrix of $H_f(\xi)$. I doubt whether this hessian matrix is correct or not.

3.3.3 Hessian considering both $\frac{\partial^2 f}{\partial e^2}$ and $H_e(\xi) = \frac{\partial^2 e}{\partial \xi^2}$

From section 3.3.1, we know that if we ignore the second part, the Hessian matrix degenerates to Gauss-Newton approximation of Hessian matrix as $J^T W J$. If the second part is reserved, we can collect above two Hessian results into chain rule to obtain a more accurate Hessian matrix:

$$
\begin{aligned}
H_f(\xi) &= (\frac{\partial e}{\partial \xi}^T \otimes I_1)\frac{\partial^2 f}{\partial e^2}\frac{\partial e}{\partial \xi} + (I_6 \otimes \frac{\partial f}{\partial e})\frac{\partial^2 e}{\partial \xi^2} \\
&= J_e^T(\xi) W J_e(\xi) + \{I_6 \otimes e^T W\} H_e(\xi) \\
&= J_e^T W J_e + \begin{pmatrix} e^T W \begin{pmatrix} \mathbf{0}_{1\times 6} \\ -J_{e,[3,:]} \\ J_{e,[2,:]} \end{pmatrix} \\ e^T W \begin{pmatrix} J_{e,[3,:]} \\ \mathbf{0}_{1\times 6} \\ -J_{e,[1,:]} \end{pmatrix} \\ e^T W \begin{pmatrix} -J_{e,[2,:]} \\ J_{e,[1,:]} \\ \mathbf{0}_{1\times 6} \end{pmatrix} \\ \mathbf{0}_{3\times 6} \end{pmatrix}
\end{aligned}
\tag{42}
$$

From experiments, we have found that the second term is useless, which demonstrates that the jacobian approximation is good enough for the hessian matrix computation.

4 Jacobian and Hessian of Cov. Cost function

$$
g(\xi) = E_{COV}(\xi) = Tr(RC_p^{-1}R^T C_q) + Tr(C_q^{-1} R C_p R^T) - 6 \tag{43}
$$

This part remains a challenge.

4.1 Lie algebra so(3) of Lie Group SO(3)

The Lie algebra so(3) of SO(3) is the real vector space consisting of all 3×3 real skew symmetric matrices. Every matrix is of the form

$$
A = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} \tag{44}
$$

where $a, b, c \in R$. The Lie bracket $[A, B]$ in $so(3)$ is also given by the usual commutator, $[A, B] = AB - BA$.

We can define an isomorphism of Lie algebras: $\psi : (R^3, \times) \longrightarrow so(3)$ by the formula

$$
\psi(\mathbf{u}) = \psi((a, b, c)^T) = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} \tag{45}
$$

To verify it is a isomorphism, we need to prove that $\psi(\mathbf{u} \times \mathbf{v}) = [\psi(\mathbf{u}), \psi(\mathbf{v})]$. For $\mathbf{u} = (a, b, c)^T$ and $\mathbf{v} = (d, e, f)^T$, the cross product is

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} \times \begin{pmatrix} d \\ e \\ f \end{pmatrix} = \begin{pmatrix} bf - ec \\ -(af - dc) \\ ae - db \end{pmatrix} \tag{46}$$

Thus,

$$\psi(\mathbf{u} \times \mathbf{v}) = \begin{pmatrix} 0 & -(ae - db) & -(af - dc) \\ (ae - db) & 0 & -(bf - ec) \\ (af - dc) & (bf - ec) & 0 \end{pmatrix} \tag{47}$$

As for lie bracket,

$$\begin{aligned} &[\psi(\mathbf{u}), \psi(\mathbf{v})] \\ &= \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} \begin{pmatrix} 0 & -f & e \\ f & 0 & -d \\ -e & d & 0 \end{pmatrix} - \begin{pmatrix} 0 & -f & e \\ f & 0 & -d \\ -e & d & 0 \end{pmatrix} \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} \end{aligned} \tag{48}$$

The result is

$$\begin{aligned} &[\psi(\mathbf{u}), \psi(\mathbf{v})] \\ &= \begin{pmatrix} -fc - be & bd & cd \\ ae & -fc - ad & ce \\ af & bf & -be - ad \end{pmatrix} - \begin{pmatrix} -fc - be & ae & af \\ bd & -fc - ad & bf \\ cd & ce & -be - ad \end{pmatrix} \\ &= \begin{pmatrix} 0 & (bd - ae) & (cd - af) \\ (ae - bd) & 0 & (ce - bf) \\ (af - cd) & (bf - ce) & 0 \end{pmatrix} \end{aligned} \tag{49}$$

Therefore,

$$\psi(\mathbf{u} \times \mathbf{v}) = [\psi(\mathbf{u}), \psi(\mathbf{v})] \tag{50}$$

Isomorphism: is a one-to-one correspondence (mapping) between two sets that preserves binary relationships between elements of the sets.

Based on above lie algebra definition, we obtain exponential map $exp : so(3) \longrightarrow SO(3)$ by Rodrigues's formula:

$$\begin{aligned} e^A &= I_3 + \frac{sin\theta}{\theta} A + \frac{1 - cos\theta}{\theta^2} A^2 \\ &= cos\theta I_3 + \frac{sin\theta}{\theta} A + \frac{1 - cos\theta}{\theta^2} B^2 \end{aligned} \tag{51}$$

where $\theta \neq 0$ and

$$A = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix}, \tag{52}$$

$$A^2 = \begin{pmatrix} -b^2-c^2 & ab & ac \\ ab & -c^2-a^2 & bc \\ ac & bc & -b^2-a^2 \end{pmatrix}, \tag{53}$$

$\theta = \sqrt{a^2+b^2+c^2}$, $B = A^2 + \theta^2 I_3$, and with $e^{0_3} = I_3$. Besides, $AB = BA = 0_{3\times 3}$ and $A^3 = -\theta^2 A$.

Since $R = e^A$, the derivative of R with respect to $\phi = (a,b,c)^T$ can be computed by the derivative of Rodrigues's formula.

It is easy to find derivatives two matrix involved with A that

$$\frac{\partial A}{\partial \phi} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{54}$$

and

$$\frac{\partial A^2}{\partial \phi} = \begin{pmatrix} 0 & -2b & -2c \\ b & a & 0 \\ c & 0 & a \\ b & a & 0 \\ -2a & 0 & -2c \\ 0 & c & b \\ c & 0 & a \\ 0 & c & b \\ -2a & -2b & 0 \end{pmatrix} \tag{55}$$

Let $\Sigma_1 = diag\{\frac{sin\theta}{\theta}, \frac{sin\theta}{\theta}, \frac{sin\theta}{\theta}\}$ and $\Sigma_2 = diag\{\frac{1-cos\theta}{\theta^2}, \frac{1-cos\theta}{\theta^2}, \frac{1-cos\theta}{\theta^2}\}$. Applying product rule to derivative of $\frac{\partial R}{\partial \phi}$ by

$$\frac{\partial R}{\partial \phi} = \frac{\partial e^A}{\partial \phi} = \frac{\partial(\Sigma_1 A)}{\partial \phi} + \frac{\partial(\Sigma_2 A^2)}{\partial \phi} \tag{56}$$

where

$$\frac{\partial(\Sigma_1 A)}{\partial \phi} = (A^T \otimes I_3)\frac{\partial \Sigma_1}{\partial \phi} + (I_3 \otimes \Sigma_1)\frac{\partial A}{\partial \phi} \tag{57}$$

and

$$\frac{\partial(\Sigma_2 A^2)}{\partial \phi} = (A^2 \otimes I_3)\frac{\partial \Sigma_2}{\partial \phi} + (I_3 \otimes \Sigma_2)\frac{\partial A^2}{\partial \phi} \tag{58}$$

Here we notice that $A^T = -A$ and $(A^2)^T = A^2$, i.e. A is skew symmetric while A^2 is symmetric.

The rest two derivatives are easy to calculate:

$$\frac{\partial \Sigma_1}{\partial \phi}_{[row:1,5,9]} = \frac{\theta cos\theta - sin\theta}{\theta^3} \begin{bmatrix} a & b & c \end{bmatrix} \tag{59}$$

And

$$\frac{\partial \Sigma_2}{\partial \phi}_{[row:1,5,9]} = \frac{\theta sin\theta - 2(1 - cos\theta)}{\theta^3} \begin{bmatrix} a & b & c \end{bmatrix} \tag{60}$$

4.2 Derivative of matrix trace

Recall that

$$g(\xi) = E_{COV}(\xi) = Tr(RC_p^{-1}R^T C_q) + Tr(C_q^{-1} RC_p R^T) - 6 \tag{61}$$

Besides, the derivative of second order matrix product is [adopted from matrix cook book]:

$$\frac{\partial}{\partial X} Tr(AXBX^T C) = (vec(B^T X^T A^T C^T + BX^T CA))^T \tag{62}$$

Let $A = I_3$, $B = C_p^{-1}$, $C = C_q$ for first term, $A = C_q^{-1}$, $B = C_p$, $C = I_3$ for the second term, then

$$\frac{\partial g}{\partial R} = (vec(C_p^{-T} R^T C_q^T + C_p^{-1} R^T C_q))^T + (vec(C_p^T R^T C_q^{-T} + C_p R^T C_q^{-1}))^T \tag{63}$$

So,

$$J_g(\xi) = \begin{bmatrix} \frac{\partial g}{\partial R} \frac{\partial R}{\partial \phi} & 0_{1\times 3} \end{bmatrix} \tag{64}$$

Apparently, above Jacobian of $\frac{\partial R}{\partial \delta\phi}$ is too complicated to calculate. Here we adopt a strategy by finding derivatives at the zero point (i.e. identity matrix) by $\frac{\partial R}{\partial \delta\phi}|_{\delta\phi = \mathbf{0}_{3\times 1}}$.

Based on lie algebra expression of $\phi = \mathbf{a} = (a_1, a_2, a_3)^T$ as $A = \mathbf{a}_\times$, A can be generated with three generators as

$$A = \Sigma_{i=1}^3 a_i G_i \tag{65}$$

where three generators are

$$G_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, G_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, G_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{66}$$

Then we apply left perturbation on rotation matrix to calculate the derivatives:

Applying left perturbation to R, we otain

$$\begin{aligned} \frac{\partial R}{\partial \phi} &= \lim_{\delta\phi \to 0} \frac{exp(\delta\phi_\times) exp(\phi_\times) - \exp(\phi_\times)}{\delta\phi} \\ &\approx \lim_{\delta\phi \to 0} \frac{(I + \delta\phi_\times) exp(\phi_\times) - \exp(\phi_\times)}{\delta\phi} \\ &= \lim_{\delta\phi \to 0} \frac{\delta\phi_\times R}{\delta\phi} \\ &= \begin{bmatrix} vec(G_1 R) & vec(G_2 R) & vec(G_3 R) \end{bmatrix} \end{aligned} \tag{67}$$

So

$$\begin{aligned}\frac{\partial g}{\partial \phi} =&(vec(C_p^{-T} R^T C_q^T + C_p^T R^T C_q^{-T} + C_p^{-1} R^T C_q + C_p R^T C_q^{-1}))^T \cdot \\ &\begin{bmatrix} vec(G_1 R) & vec(G_2 R) & vec(G_3 R) \end{bmatrix}\end{aligned} \tag{68}$$

4.3 Hessian Of Matrix trace

This part is to compute

$$\begin{aligned}\frac{\partial^2 g}{\partial \phi^2} &= \frac{\partial (J_{gR} * J_{R\phi})}{\partial \phi} \\ &= J_{R\phi}^T \frac{\partial J_{gR}}{\partial R} J_{R\phi} + (I_3 \otimes J_{R\phi}) \frac{\partial^2 R}{\partial \phi^2}\end{aligned} \tag{69}$$

Similar to hessian of ICP cost, the second term is also neglected. So we only need to find the hessian of g with respect to R.

$$\begin{aligned}H_{gR} &= \frac{\partial J_{gR}}{\partial R} \\ &= (C_q \otimes C_p^{-T} + C_q^{-1} \otimes C_p^T + C_q^T \otimes C_p^{-1} + C_q^{-T} \otimes C_p) K_{33}\end{aligned} \tag{70}$$

Here we utilize the Kronecker product rule to derive below matrix differential.

$$\frac{\partial}{\partial X} AXB = B^T \otimes A \tag{71}$$

and

$$\frac{\partial}{\partial X} AX^T B = (B^T \otimes A) K_{mn} \tag{72}$$

where m, n are dimensions of X.