**视觉定位与感知-HW3**

**516021910700 郭远帆**

**1.**

设：

首先验证封闭性:

可以看到结果也是反对称矩阵，封闭性得以验证。

接下来验证双线性:

双线性得以验证。

接下来验证自反性。

自反性得以验证。

最后验证雅可比等价性。

由之前的性质，该运算在向量形式上可以看作叉积(观察结果可知复原的向量是原向量叉积的结果)

而在向量叉积中，有如下性质:

该性质与雅可比等价性一致，由此一致性可以得到雅可比等价性的验证，因此不再展开。

以上，验证了该李代数。实际上，该李代数即是三维向量上定义的叉积。

**2.**

对三维向量x，我们假设其四元数表示为，其中x,y,z为其三个坐标轴的坐标。

根据四元数的运算法则，对于单位四元数**，**有:

有

由四元数的乘法计算公式，有:

要证明该变换是旋转变换，需要证明变换后的四元数的实部为0。根据四元数的乘法计算公式,有:

最终所有项均可两两消去，有,故该变换可以表示为旋转变换。利用乘法公式对虚部进行同样的操作，有:

解得最终结果为:

这实际上对应旋转矩阵**R**的第一行。对其它两个虚数部分进行同样的计算，可以得到最终旋转矩阵:

**3.**

**实验原理:**

设旋转矩阵 **R** 以及其扰动**,** 设扰动的李代数为**,** 有:

为的李代数表示，将表示为旋转向量的形式:，其中为旋转轴（单位向量），为旋转角。由此定义参数更新 如下:

其中符号^为向量的反对称矩阵表示(或者表示为)。严格上来说，上式为近似公式，参数更新也可用下式表达:

如果采用四元数来代表旋转，那么参数更新的定义**:**

当迭代到足够小时，采用近似公式表示:

这里的乘符号是四元数的乘法。

根据高斯牛顿法的思想，每一轮迭代的近似优化目标为:

其中即扰动的李代数表示。根据扰动模型，近似为求解此方程:

定义, ，上式关于 导数为0可得:

求解此方程，得:

根据之前的参数更新定义，更新**R**（或者**q**）即可。最后对于更新后的**R**需要保证其正交性，对于**q**需要保证其单位性。采用SVD分解保证**R**的正交性，采用归一化保证**q**的单位性即可。

除此之外，参考《视觉SLAM十四讲》，3D位姿估计问题还有一种SVD解法，其原理如下:

定义两组点的质心如下:

然后得到这两组点的去质心坐标:

定义矩阵**W:**

对**W**进行SVD分解，得到:

然后可以解得**R:**

该方法是一种几何的方法，将多点的变换“浓缩”为质心之间的变换，在样本数大的情况下其效果不一定比高斯牛顿法好，但由于本次实验中样本点只有3个，我们采用这个方法来与高斯牛顿法进行对比。

**实现关键思路:**

代码实现通过Matlab语言，以下为实现的关键思路:

**基础模块实现:**

1. 为了计算向量的反对称矩阵，实现 X = skew(x) 函数，接受3\*1向量x作为输入，输出其反对称矩阵X
2. 实现R = q2r(q) 以及 q = r2q（R）函数，实现四元数到旋转矩阵的转换
3. 误差计算函数 target(R,x,y) 计算目标函数
4. 实现 q = Qmult(q1,q2) 实现四元数的乘法

基础模块实现后，根据高斯牛顿法的迭代思想持续更新参数并记录误差即可。Matlab对矩阵运算的支持相当完善，代码语言与数学语言近似，在此不再进行冗余的描述。

在本实验中分别实现了四元数表达方式、旋转矩阵表达方式下的高斯牛顿法，其功能接近，可用于验证计算是否正确。

**实验结果与分析:**

采用SVD方法得到的旋转矩阵为:

给出的误差约为，在此基础上加入随机扰动项，然后在不同扰动的情况下开始迭代，观察扰动大小不同情况下的收敛速度如表1(迭代x次项中为得到的误差)。

表1 实验数据表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 扰动大小 | 初始误差 | 迭代2次 | 迭代5次 | 迭代10次 | 迭代20次 | 迭代50次 | 迭代100次 |
| 0.01 | 1.21e-4 | 7.82e-6 | 7.78e-8 | 1.42e-15 | 2.32e-20 | 2.27e-31 | 8.51e-32 |
| 0.1 | 6.52 | 0.093 | 2.61e-4 | 1.1e-6 | 2.17e-12 | 2.27e-29 | 2.54e-32 |
| 1 | 7.13 | 0.014 | 2.42e-4 | 3.02e-7 | 5.52e-13 | 6.45e-30 | 4.22e-31 |
| 10 | 7.66 | 0.031 | 1.27e-4 | 1.81e-7 | 3.22e-13 | 1.51e-30 | 2.50e-32 |
| 随机正交基 | 10.31 | 0.85 | 0.013 | 8.81e-5 | 1.43e-10 | 1.15e-27 | 1.91e-31 |

随机正交基下的误差变换图:

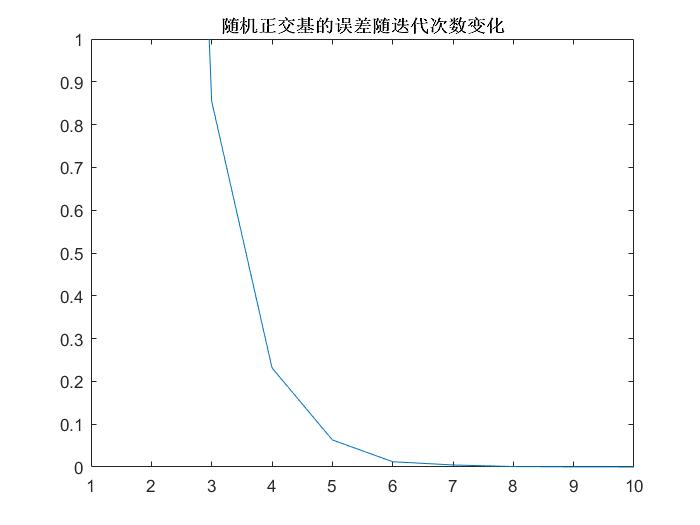


图1

可以看到，初始误差越大，收敛的速度也就较慢，然而无论如何，10代之后已经有了相当可观的结果，迭代到50代已经接近最优解。

**总结:**

本实验中对高斯牛顿法在3D-3D位姿估计问题中的应用进行了实现，对比了不同初值情况下的高斯牛顿法收敛结果，并比较了SVD方法与高斯牛顿法的误差结果。在样本点较小的情况下，SVD方法可以较快地得到精确的结果，但相比高斯牛顿法，计算开销较大。

本实验的代码由Matlab编写，实现了基于旋转矩阵和四元数表达的高斯牛顿法模型以及SVD求解方法。

**感想和体会:**

第三题是踩了不少的坑啊！由于理论知识不够扎实，对于李代数的扰动模型不熟悉，对于参数更新的不理解，一开始得到的结果千奇百怪。但也正因为如此，去查阅了各种资料，将高翔博士的《视觉SLAM十四讲》看了一遍又一遍，将老师的PPT温习了一遍又一遍，最后终于成功地分别实现了四元数表达的高斯牛顿法、旋转矩阵表达的高斯牛顿法，还把SVD方法（基于几何学）也实现了一遍，历时约两个星期，终于将这个问题解了出来。

感谢老师的指导和支持！SLAM真的很有趣，本次作业相当精彩