2 群的性质（2 分，约 1 小时）

课上我们讲解了什么是群。请根据群定义，求解以下问题：

说明理由。 其中 Z 为整数集，N 为自然数集

1. {Z,+} 是否为群？若是，验证其满足群定义；若不是，说明理由。

{Z,+}是群，因为它满足以下性质：
封闭性：对于任意整数 a 和 b,易得 a+b 也是整数。
结合律：整数加法满足结合律。
幺元：存在整数 0 使得任意整数 a 加上 0 等于 a。
逆：对于任意整数 a，易知存在-a 使得 a+（-a）=0。

2. {N,+} 是否为群？若是，验证其满足群定义；若不是，说明理由。

{Z,+}不是群，因为它虽然满足"幺元"但不满足"逆"性质。
幺元：存在自然数 0 使得任意自然数 a 加上 0 等于 a。
逆：对于任意自然数 a，找不到对应的a^{-1}使得$a + a^{-1} = 0$。

3 验证向量叉乘的李代数性质（2 分，约 1 小时）

请验证 g = (R3,R,×) 构成李代数。

假设有向量$X = [x_1, y_1, z_1]^T; Y = [x_2, y_2, z_2]^T; Z = [x_3, y_3, z_3]^T$

1. 封闭性

   X 向量为 3 行 1 列向量，X^为 3 行 3 列，Y 向量 3 行 1 列，可知$X^\wedge Y$结果为 3 行 1 列向量。属于集合v，

2. 双线性

$$[aX + bY, Z] = \begin{bmatrix} ax_1 + bx_2 \\ ay_1 + by_2 \\ az_1 + bz_2 \end{bmatrix}^\wedge \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0 & -az_1 - bz_2 & ay_1 + by_2 \\ az_1 + bz_2 & 0 & -ax_1 - bx_2 \\ -ay_1 - by_2 & ax_1 + bx_2 & 0 \end{bmatrix} * \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}$$

$$= \begin{bmatrix} -az_1y_3 - bz_2y_3 + ay_1z_3 + by_2z_3 \\ az_1x_3 + bz_2x_3 - ax_1z_3 - bx_2z_3 \\ -ay_1x_3 - by_2x_3 + ax_1y_3 + bx_2y_3 \end{bmatrix}$$

$$a[X, Z] + b[Y, Z] = a \begin{bmatrix} 0 & -z_1 & y_1 \\ z_1 & 0 & -x_1 \\ -y_1 & x_1 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} + b \begin{bmatrix} 0 & -z_2 & y_2 \\ z_2 & 0 & -x_2 \\ -y_2 & x_2 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}$$

$$= \begin{bmatrix} -az_1y_3 + ay_1z_3 \\ az_1x_3 - ax_1z_3 \\ -ay_1x_3 + ax_1y_3 \end{bmatrix} + \begin{bmatrix} -bz_2y_3 + by_2z_3 \\ bz_2x_3 - bx_2z_3 \\ -by_2x_3 + bx_2y_3 \end{bmatrix}$$

第一行相同部分用颜色标出，第二三行略，$[Z, aX + bY] = a[Z, X] + b[Z, Y]$同理，略。

3. 自反性

$$X^{\wedge}X=\begin{bmatrix}0 & -z_1 & y_1\\ z_1 & 0 & -x_1\\ -y_1 & x_1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ y_1\\ z_1\end{bmatrix}=\begin{bmatrix}-z_1y_1+y_1z_1\\ z_1x_1-x_1z_1\\ -y_1x_1+x_1y_1\end{bmatrix}=\begin{matrix}0\\0\\0\end{matrix}$$

4. 雅可比等价

$$[x,[y,z]]+[y,[z,x]]+[z,[x,y]]=\begin{bmatrix}0 & -z_1 & y_1\\ z_1 & 0 & -x_1\\ -y_1 & x_1 & 0\end{bmatrix}*\left(\begin{bmatrix}0 & -z_2 & y_2\\ z_2 & 0 & -x_2\\ -y_2 & x_2 & 0\end{bmatrix}*\begin{bmatrix}x_3\\ y_3\\ z_3\end{bmatrix}\right)$$

$$+\begin{bmatrix}0 & -z_2 & y_2\\ z_2 & 0 & -x_2\\ -y_2 & x_2 & 0\end{bmatrix}*\left(\begin{bmatrix}0 & -z_3 & y_3\\ z_3 & 0 & -x_3\\ -y_3 & x_3 & 0\end{bmatrix}*\begin{bmatrix}x_1\\ y_1\\ z_1\end{bmatrix}\right)$$

$$+\begin{bmatrix}0 & -z_3 & y_3\\ z_3 & 0 & -x_3\\ -y_3 & x_3 & 0\end{bmatrix}*\left(\begin{bmatrix}0 & -z_1 & y_1\\ z_1 & 0 & -x_1\\ -y_1 & x_1 & 0\end{bmatrix}*\begin{bmatrix}x_2\\ y_2\\ z_2\end{bmatrix}\right)$$

$$=\begin{bmatrix}0 & -z_1 & y_1\\ z_1 & 0 & -x_1\\ -y_1 & x_1 & 0\end{bmatrix}*\begin{bmatrix}-z_2y_3+y_2z_3\\ z_2x_3-x_2z_3\\ -y_2x_3+x_2y_3\end{bmatrix}$$

$$+\begin{bmatrix}0 & -z_2 & y_2\\ z_2 & 0 & -x_2\\ -y_2 & x_2 & 0\end{bmatrix}*\begin{bmatrix}-z_3y_1+y_3z_z\\ z_3x_1-x_3z_1\\ -y_3x_1+x_3y_1\end{bmatrix}$$

$$+\begin{bmatrix}0 & -z_3 & y_3\\ z_3 & 0 & -x_3\\ -y_3 & x_3 & 0\end{bmatrix}*\begin{bmatrix}-z_1y_2+y_1z_2\\ z_1x_2-x_1z_2\\ -y_1x_2+x_1y_2\end{bmatrix}$$

$$=\begin{bmatrix}-z_1z_2x_3+z_1x_2z_3-y_1y_2x_3+y_1x_2y_3\\ -z_1z_2y_3+z_1y_2z_3+x_1y_2x_3-x_1x_2y_3\\ y_1z_2y_3-y_1y_2z_3+x_1z_2x_3-x_1x_2z_3\end{bmatrix}$$

$$+\begin{bmatrix}-z_2z_3x_1+z_2x_3z_1-y_2y_3x_1+y_2x_3y_1\\ -z_2z_3y_1+z_2y_3z_1+x_2y_3x_1-x_2x_3y_1\\ y_2z_3y_1-y_2y_3z_1+x_2z_3x_1-x_2x_3z_1\end{bmatrix}$$

$$+\begin{bmatrix}-z_3z_1x_2+z_3x_1z_2-y_3y_1x_2+y_3x_1y_2\\ -z_3z_1y_2+z_3y_1z_2+x_3y_1x_2-x_3x_1y_2\\ y_3z_1y_2-y_3y_1z_2+x_3z_1x_2-x_3x_1z_2\end{bmatrix}$$

$$=0$$

第一行中可以抵消的元素用颜色标出，第二第二行同理，略。

4 推导 SE(3) 的指数映射（2 分，约 1 小时）

SE(3)指数映射

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

$$=\mathrm{I}+\sum_{n=1}^{\infty}\frac{1}{n!}\left(\begin{bmatrix}\phi^\wedge & \rho\\ 0^T & 0\end{bmatrix}\right)^n = I+\sum_{n=1}^{\infty}\frac{1}{n!}\left(\begin{bmatrix}\phi^{\wedge n} & \phi^{\wedge n-1}\rho\\ 0^T & 0\end{bmatrix}\right)$$

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

左雅可比 J

$$\text{定义}\phi=\theta a$$

$$\sum_{n=0}^{\infty}\frac{1}{(n+1)!}\phi^{\wedge n}=\mathrm{I}+\frac{1}{2!}(\theta a^\wedge)+\frac{1}{3!}(\theta a^\wedge)^2+\frac{1}{4!}(\theta a^\wedge)^3+\frac{1}{5!}(\theta a^\wedge)^4+\frac{1}{6!}(\theta a^\wedge)^5+\cdots$$

$$=\mathrm{I}+\frac{1}{2!}\theta a^\wedge+\frac{1}{3!}\theta^2 a^\wedge a^\wedge+\frac{1}{4!}\theta^3 a^\wedge a^\wedge a^\wedge+\frac{1}{5!}\theta^4 a^\wedge(a^\wedge a^\wedge a^\wedge)+\frac{1}{6!}\theta^5(a^\wedge a^\wedge)(a^\wedge a^\wedge a^\wedge)+\cdots$$

$$\text{有}a^\wedge a^\wedge=aa^T-I\text{和}a^\wedge a^\wedge a^\wedge=-a^\wedge,\text{代入至上式得到}$$

$$=\mathrm{I}+\frac{1}{2!}\theta a^\wedge+\frac{1}{3!}\theta^2 a^\wedge a^\wedge-\frac{1}{4!}\theta^3 a^\wedge-\frac{1}{5!}\theta^4 a^\wedge a^\wedge+\frac{1}{6!}\theta^5 a^\wedge+\cdots$$

$$=\mathrm{I}+\left(\frac{1}{2!}\theta-\frac{1}{4!}\theta^3+\frac{1}{6!}\theta^5-\cdots\right)a^\wedge+\left(\frac{1}{3!}\theta^2-\frac{1}{5!}\theta^4+\cdots\right)a^\wedge a^\wedge$$

$$=\mathrm{I}+\frac{1}{\theta}\left(\frac{1}{2!}\theta^2-\frac{1}{4!}\theta^4+\frac{1}{6!}\theta^6-\cdots\right)a^\wedge+\frac{1}{\theta}\left(\frac{1}{3!}\theta^3-\frac{1}{5!}\theta^5+\cdots\right)a^\wedge a^\wedge$$

$$\text{又有 }\cos\theta=1-\frac{1}{2!}\theta^2+\frac{1}{4!}\theta^4-\frac{1}{6!}\theta^6+\cdots\text{和 }\sin\theta=\theta-\frac{1}{3!}\theta^3+\frac{1}{5!}\theta^5-\cdots,\text{代入上式得}$$

$$=\mathrm{I}+\frac{1}{\theta}(1-cos\theta)a^\wedge+\frac{1}{\theta}(\theta-sin\theta)a^\wedge a^\wedge$$

$$=I+\frac{1}{\theta}(1-cos\theta)a^\wedge+\frac{1}{\theta}(\theta-sin\theta)(aa^T-I)$$

$$=\mathrm{I}+\frac{1}{\theta}(1-cos\theta)a^\wedge+\frac{1}{\theta}(\theta aa^T-\theta I-sin\theta aa^T+sin\theta I)$$

$$=\ \mathrm{I}+\frac{1}{\theta}(1-cos\theta)a^\wedge+aa^T-I-\frac{sin\theta aa^T}{\theta}+\frac{sin\theta I}{\theta}$$

$$=\frac{sin\theta I}{\theta}+\left(1-\frac{\sin\theta}{\theta}\right)aa^{\mathrm{T}}+\frac{1}{\theta}(1-cos\theta)a^\wedge$$

5 伴随（2 分，约 1 小时）

证明伴随矩阵性质:$\mathrm{Ad(R)} = \mathrm{R}$

首先证明$\mathrm{Ra}\hat{}R^T = (Ra)\hat{}$，假设有向量v，则：

$$(Ra)\hat{}v = (Ra)\hat{}RR^Tv$$

又有矩阵 U 和向量 a，b：$(\mathrm{Ua})\hat{}(Ub) = U(a\hat{}b)$. *代入上式得到*

$$(R*a)\hat{}(R*R^Tv) = R(a\hat{}R^Tv)$$
$$(Ra)\hat{} = Ra\hat{}R^T$$

再证明$\mathrm{Ad(R)} = \mathrm{R}$

$$\mathrm{Rexp(p\hat{})R}^T = \exp((Ad(R)p)\hat{})$$
$$\log(\mathrm{Rexp(p\hat{})R}^T) = \log(\exp((Ad(R)p)\hat{}))$$

因为有$\log(VAV^{-1}) = Vlog(A)V^{-1}$, *代入上式左边*

$\mathrm{Rlog(exp(P\hat{}))R^T} = \log(\exp((Ad(R)p)\hat{}))$，两边的 log 和 exp 各自抵消

$$\mathrm{Rp\hat{}R}^T = (Ad(R)p)\hat{}$$

因为有上面证明的$\mathrm{Ra}\hat{}R^T = (Ra)\hat{}$，*代入上式*

$$(\mathrm{Rp})\hat{} = (Ad(R)p)\hat{}$$

得到 $\mathrm{R} = \mathrm{Ad(R)}$

6 轨迹的描绘（2 分，约 1 小时）

1. 事实上，TWC 的平移部分即构成了机器人的轨迹。它的物理意义是什么？为何画出 TWC 的平移部分就得到了机器人的轨迹

$\mathrm{T}_c^{\mathrm{W}} = \begin{bmatrix} R^{3x3} & T^{3x1} \\ 0 & 1 \end{bmatrix}$平移部分的$T^{3x1}$表示的是机器人相对于 world 的位移。把每个时间的相对位移画出来就是机器人在 world 系下的位置轨迹。（当然需要的话画上旋转部分还能获得每个时刻的朝向）

2. 请你完成数据读取部分的代码

```cpp
/// implement pose reading code
// start your code here (5~10 lines)
ifstream fin(trajectory_file);
if(!fin)
{
    cout<<"trajectory_file not found : "<<trajectory_file<<endl;
    return -1;
}
while (!fin.eof())
{
    double time, tx, ty, tz, qx, qy, qz, qw;
    fin >> time >> tx >> ty >> tz >> qx >> qy >> qz >> qw;
    Sophus::SE3 pose_tmp(Eigen::Quaterniond(qx, qy, qz, qw), Eigen::Vector3d(tx, ty, tz));
    poses.push_back(pose_tmp);
}
cout<<"read "<<poses.size()<<" poses"<<endl;
fin.close();
// end your code here
```

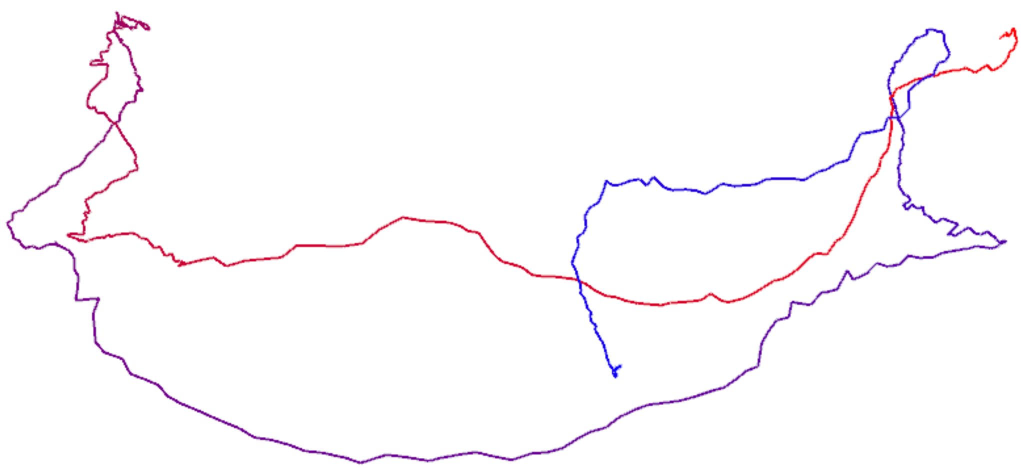
CMakeLists.txt

```cmake
cmake_minimum_required( VERSION 2.8 )
project( code_homework )

set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++11")

find_package( Sophus REQUIRED )
find_package( Pangolin REQUIRED )
find_package( Eigen3 REQUIRED)
include_directories( ${Sophus_INCLUDE_DIRS} )
include_directories( ${Pangolin_INCLUDE_DIRS} )
include_directories( ${EIGEN3_INCLUDE_DIRS} )

add_executable( code_homework src/draw_trajectory.cpp)
target_link_libraries( code_homework ${Sophus_LIBRARIES} ${Pangolin_LIBRARIES})
```



7* 轨迹的误差 (2 分，约 1 小时)

按照上一题读取轨迹的方法读取了 groundtruth.txt 和 estimated.txt，分别存储在 gt_pose 和 est_pose 中。上述代码略。以下为计算误差的代码。

```
// calculated error
double rmse = 0;
for(int i = 0 ; i < est_pose.size();i++)
{
    Sophus::SE3 gt = gt_pose[i];
    Sophus::SE3 est = est_pose[i];
    double errortmp = (gt.inverse()*est).log().norm();
    //cout<<"i "<<i<<" "<<errortmp<<endl;
    rmse = rmse + errortmp*errortmp;
}
cout <<"rmse "<<rmse <<endl;
rmse = sqrt(rmse / est_pose.size());
cout <<"rmse "<<rmse <<endl;
DrawTrajectory(gt_pose,est_pose);
```

运行结果 rmse2.20727

```
xin@ubuntu16:~/VSLAM-course/vSLAM-course/Ch3/homework/L3/build$ ./code_homework
read 620 poses
read 613 gt poses
read 613 est poses
rmse 2986.57
rmse 2.20727
```