习题 1

1. 当 $rank(A) = rank(b)$ 时有唯一解；当 $rank(A) > rank(b)$ 时有多解；当 $rank(A) < rank(b)$ 时无解。
2. 高斯分布：$f(x) = (\sqrt{2\pi}\sigma)^{-1} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$；$f_{\mathbf{X}}(x_1, \cdots, x_k) = \frac{exp(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu))}{\sqrt{|2\pi\Sigma|}}$
3. 初步了解 C++ 的类和 **STL**，但还未用过
4. 以前用过的编辑器有 VC6.0、Xcode、Sublime Text 和 Vim
5. 从 [?] 了解到 C++11 标准，新特性包括：**正则表达式、智能指针等**，新标准还有 C++14, C++17
6. 熟悉 Linux(Ubuntu)，本科期间用过 Ubuntu 来学习和调试基于 ROS 的移动机器人
7. 查看 Linux 目录结构 **$ls /**。常用命令：**ls, cat, pwd, cd, make, mv, tar...**
8. Ubuntu 中常用 apt 安装软件，macOS 常用 brew 安装软件。前者默认安装路径：PATH=/home/brigh
后者 Homebrew 会将软件包安装到独立目录，并将其文件软链接至 /usr/local。以安装 **Eigen** 库为例，源码安装 [1] 或 **brew install eigen**
9. **Vim**[2] **$ vimtutor**。执行外部命令 **:!command**

    切记在使用中学习，而不是在记忆中学习

习题 2

1. [?, ?]
2. [?],[?],[?],[?],[?]。这些文献关于 SLAM 的看法与 [?] 的异同：
    - ...ToDo
    - ...ToDo
3. **g++** 命令参数举例：

    ```
    $ g++ -o main main.cpp -I /头文件路径 -L /库文件路径 -l 库文件名称
    ```

4. **Qt Creator** 配置 build 文件夹

    ```
    ./%{JS: Util.asciify("build")}
    ```

[1] Eigen 只有头文件故不用编译

[2] 不要在它的插件上浪费时间，不要想着把 VIM 用成 IDE，我们只用它做文本编辑的工作

图 1: 两个可执行程序时，在 **Debug** 前提下选 helloSLAM 或 useHello

5. 语法错误的情况下，**cmake** 不能检测出来[3]，**make** 才会提示语法错误。

6. **cmake** 通过，但是 **make** 是报错:

```
Undefined symbols for architecture x86_64:
"printHello()", referenced from:
_main in useHello.cpp.o
ld: symbol(s) not found for architecture x86_64
```

7. DONE

8. **find_package**
   把头文件和库文件安装在自定义路径时，**动态库**[4]**链接出错**提示：[5][6]

```
dyld: Library not loaded: libmyhello.6.dylib
Referenced from: /Users/zhangjixiang/Documents/Programs/myhello/build/./main
Reason: image not found
```

   @rpath
   Q：**一定要安装在/usr/local 路径吗???**

```
"otool -D <file>" to view the install name of a dylib
"otool -L <file>" to view the dependencies
"otool -l <file> | grep LC_RPATH -A2" to view the RPATHs
"install_name_tool -id ..." to change an install name
"install_name_tool -change ..." to change the dependencies
"install_name_tool -rpath ... -add_rpath ... -delete_rpath ..." to change RPATHs
```

   [7]

   C 和 C++ 相互调用出错时的解决方案：**Linking C and CXX files in CMake**

[3] cmake 的功能是根据库的依赖关系生成 **Makefile** 文件，却不检查依赖关系是否完善?

[4] 小技巧：用 **otool -L main** 代替 **ldd main**

[5] 解决方案，考虑链接到**静态库** TARGET_LINK_LIBRARIES(main /Users/zhangjixiang/slambook/lib/libmyhello.a)

[6] g++ -o main main.cpp -I /Users/zhangjixiang/slambook/include/myhello -L /Users/zhangjixiang/slambook/lib -l myhello

[7] install_name_tool -change libmyhello.6.dylib /Users/zhangjixiang/slambook/lib/libmyhello.6.dylib ./a.out

```
#ifndef F_H
#define F_H

#ifdef __cplusplus
extern "C" {
#endif

void f();

#ifdef __cplusplus
}
#endif

#endif
```

The standard locations for dynamic libraries are /lib, /usr/local/lib, and /usr/lib.

总结：macOS locates dependent libraries using FULLPATH to each dylib. 解决方法有三种：

(a) **安装到系统目录/usr/local**

(b) 修改共享库的 FULLPATH **install name**

```
install_name_tool -id "new_install_name" libdummy.dylib
```

(c) 修改生成共享库的 CMakeLists.txt

```
IF(APPLE)
SET(CMAKE_INSTALL_NAME_DIR /Users/zhangjixiang/slambook/lib)
SET(CMAKE_BUILD_WITH_INSTALL_RPATH ON)
ENDIF(APPLE)
```

参考资料：

- Install name on OS X
- How do folks work with rpath on OS X using cmake?

FindHELLO.cmake：Search the paths specified by the **HINTS** option

```
find_path(HELLO_INCLUDE_DIR libHelloSLAM.h /Users/zhangjixiang/slambook/include/myh
find_library(HELLO_LIBRARY NAMES myhello HINTS /Users/zhangjixiang/slambook/lib)
```

图 2:

```
#set(HELLO_LIBRARY /Users/zhangjixiang/slambook/lib/libmyhello.dylib)

if(HELLO_INCLUDE_DIR AND HELLO_LIBRARY)
set(HELLO_FOUND TRUE)
endif(HELLO_INCLUDE_DIR AND HELLO_LIBRARY)

if(HELLO_FOUND)
if(NOT HELLO_FIND_QUIETRY)
message(STATUS "Found hello: ${HELLO_LIBRARY}")
endif(NOT HELLO_FIND_QUIETRY)
else(HELLO_FOUND)
if(HELLO_FIND_REQUIRED)
message(FATAL_ERROR "Could not find myhello library")
endif(HELLO_FIND_REQUIRED)
endif(HELLO_FOUND)
```

参考：cmake - find_library - custom library location
find_library

9. DONE

10. 熟悉灵活使用自己的工具 **Qt Creator**，了解其它特性

11. **Qt Creator** 的 **vim** 编辑功能

习题 3

1.
$$RR^T = \begin{bmatrix} e_1^T e_1' & e_1^T e_2' & e_1^T e_3' \\ e_2^T e_1' & e_2^T e_2' & e_2^T e_3' \\ e_3^T e_1' & e_3^T e_2' & e_3^T e_3' \end{bmatrix} \begin{bmatrix} e_1^T e_1' & e_2^T e_1' & e_3^T e_1' \\ e_1^T e_2' & e_2^T e_2' & e_3^T e_2' \\ e_1^T e_3' & e_2^T e_3' & e_3^T e_3' \end{bmatrix} = \mathbf{I} \tag{1}$$

   从投影[8]的角度理解旋转矩阵，并利用单位和正交条件[9]

2. Rodrigues' rotation formula

$$R(a,\theta) = I_{3\times3}\cos\theta + aa^T(1-\cos\theta) + a^{\wedge}\sin\theta \tag{2}$$

[8]内积

[9]一共有 6 个约束方程

3. $$p' = [\cos\frac{\theta}{2}, \mathbf{n}\sin\frac{\theta}{2}][0, \mathbf{v}][\cos\frac{\theta}{2}, \mathbf{n}\sin\frac{\theta}{2}]^{-1} \tag{3}$$

   其中 $s = [-\mathbf{n}^T\mathbf{v}\sin\frac{\theta}{2}\cos\frac{\theta}{2} + \mathbf{v}^T\mathbf{n}\sin\frac{\theta}{2}\cos\frac{\theta}{2} + (\mathbf{n}\times\mathbf{v})^T\mathbf{n}\sin^2\frac{\theta}{2}, ...] = [0, ...]$

4. 见图表

5.

```
#include <iostream>
using namespace std;

#include <Eigen/Core>
#define MATRIX_SIZE 4

int main(int argc, char** argv)
{
        Eigen::Matrix<double, MATRIX_SIZE, MATRIX_SIZE> matrix_NN;
        matrix_NN = Eigen::MatrixXd::Random(MATRIX_SIZE, MATRIX_SIZE);
        cout << "Original Matrix = \n" << matrix_NN << endl;

        Eigen::Matrix3d matrix_I = Eigen::Matrix3d::Identity();
        cout << "TopLeft33 of the Matrix = \n" << matrix_NN.topLeftCorn

        matrix_NN.topLeftCorner(3, 3) = matrix_I;
        cout << "Final Matrix = \n" << matrix_NN << endl;

        return 0;
}
```

6. **Ax=b** 求解方法:
   - Direct Methods
   - Iteration Methods

7. 程序[10]:

```
#include <iostream>
using namespace std;
```

[10]**pretranslate()**: Applies on the **left** the translation matrix represented by the vector other to *this and returns a reference to *this. **translate()**:Applies on the **right** the translation matrix represented by the vector other to *this and returns a reference to *this.

```cpp
#include <Eigen/Core>
#include <Eigen/Geometry>

#define MATRIX_SIZE 4

int main(int argc, char** argv)
{
        Eigen::Vector3d p_c1(0.5,0,0.2);
        Eigen::Vector3d p_c2;

        Eigen::Quaterniond q1(0.35, 0.2, 0.3, 0.1);
        q1.normalize();
        Eigen::Vector3d t1(0.3,0.1,0.1);
        Eigen::Isometry3d T_c1w = Eigen::Isometry3d::Identity();
        T_c1w.rotate(q1);
        T_c1w.pretranslate(t1);


        Eigen::Quaterniond q2(-0.5, 0.4, -0.1, 0.2);
        q2.normalize();
        Eigen::Vector3d t2(-0.1,0.5,0.3);
        Eigen::Isometry3d T_c2w = Eigen::Isometry3d::Identity();
        T_c2w.rotate(q2);
        T_c2w.pretranslate(t2);

        p_c2 = T_c2w * (T_c1w.inverse()) * p_c1;

        // cout << q2.toRotationMatrix() << endl;
        // cout << T_c2w.matrix() << endl;
        cout << "p_c1 = \n" << p_c1 << endl;
        cout << "p_c2 = \n" << p_c2 << endl;

        return 0;
}
```

习题 4

1. 群的四个条件：**封闭性、结合律、幺元、逆**，封闭性、结合律、幺元容易验证，只需验

证逆：

$$Sim(3) = \left\{ S = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \right\} \tag{4}$$

则

$$S^{-1} = \begin{bmatrix} \frac{R^T}{s} & -\frac{R^T}{s}t \\ 0 & 1 \end{bmatrix} \in Sim(3) \tag{5}$$

2. 李代数的四个条件：**封闭性、双线性、自反性、雅可比等价**。证明：记 $\mathbf{g}$=$(R^3, R, \times)$

   - $\forall a, b \in R^3, a \times b \in R^3$, 故满足封闭性；
   - $\forall$**a,b,c** $\in R^3, m, n \in R$, 有：
     $(m\mathbf{a} + n\mathbf{b}) \times \mathbf{c} = m(\mathbf{a} \times \mathbf{c}) + n(\mathbf{b} \times \mathbf{c}), \quad \mathbf{c} \times (m\mathbf{a} + n\mathbf{b}) = m(\mathbf{c} \times \mathbf{a}) + n(\mathbf{c} \times \mathbf{b})$
     故满足双线性；
   - $\forall a \in R^3, a \times a = \mathbf{0}$, 故满足自反性；
   - $\forall a, b, c \in R^3$, 有
     $a \times (b \times c) + b \times (c \times a) + c \times (a \times b) =$
     $b(a \cdot c) - c(a \cdot b) + c(b \cdot a) - a(b \cdot c) + a(c \cdot b) - b(c \cdot a) = \mathbf{0}$, 故满足雅可比等价. 综上有 $\mathbf{g}$=$(R^3, R, \times)$ 为李代数。

3. $\mathfrak{so}(3)$ 和 $\mathfrak{se}(3)$ 满足李代数的四个条件：**封闭性、双线性、自反性、雅可比等价**? TODO

4. TODO

   - $$a^\wedge a^\wedge = ... = aa^T - I$$
   - $$a^\wedge a^\wedge a^\wedge = ... = -a^\wedge$$

5. 因为

$$\forall v \in R^3, \quad (Ra)^\wedge v = (Ra) \times v = (Ra) \times (RR^{-1}v) = R[a \times (R^{-1}v)] = Ra^\wedge R^{-1}v$$

, 且 $R^{-1} = R^T$, 故成立。$\blacksquare$

6. 记 $p = \theta a$, 原式左边：

$$= R[\cos\theta\mathbf{I} + (1 - \cos\theta)aa^T + \sin\theta a^\wedge]R^T = \cos\theta\mathbf{I} + (1 - \cos\theta)R(aa^T)R^T + \sin\theta(Ra^\wedge R^T)$$

$$= \cos\theta\mathbf{I} + (1 - \cos\theta)R(aa^T)R^T + \sin\theta(Ra)^\wedge$$

原式右边：

$$= \cos\theta\mathbf{I} + (1 - \cos\theta)(Ra)(Ra)^T + \sin\theta(Ra)^\wedge = \cos\theta\mathbf{I} + (1 - \cos\theta)R(aa^T)R^T + \sin\theta(Ra)^\wedge$$

. 故等式左边 = 右边, 即:

$$R\exp(p^\wedge)R^T = \exp((Rp)^\wedge) \tag{6}$$

故原式成立。■

SE(3) 伴随性质证明:

原式左边:

$$= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sum_1 & \sum_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R\sum_1 R^T & R\sum_1(-R^T t) + R\sum_2 + t \\ 0 & 1 \end{bmatrix}$$

, 式中 $\sum_1 = \sum_{n=0}^{\infty} \frac{1}{n!}(\phi^\wedge)^n$, $\quad \sum_2 = \sum_{n=0}^{\infty} \frac{1}{(n+1)!}(\phi^\wedge)^n \rho$

原式右边:

$$= \exp(\begin{bmatrix} R\rho + (t^\wedge R)\phi \\ R\phi \end{bmatrix}^\wedge) = \begin{bmatrix} \sum_3 & \sum_4 \\ 0 & 1 \end{bmatrix}$$

, 式中 $\sum_3 = \sum_{n=0}^{\infty} \frac{1}{n!}((R\phi)^\wedge)^n$, $\quad \sum_4 = \sum_{n=0}^{\infty} \frac{1}{(n+1)!}((R\phi)^\wedge)^n [R\rho + t^\wedge R\phi]$.

故只需要证明:

$$\sum_3 = R\sum_1 R^T, \quad \sum_4 = R\sum_1(-R^T t) + R\sum_2 + t$$

, 其中 $\sum_3 = R\sum_1 R^T$ 显然成立。下面证明 $\sum_4 = R\sum_1(-R^T t) + R\sum_2 + t$:

原式等价于

$$\Leftrightarrow \sum_{n=0}^{\infty} \frac{1}{(n+1)!}((R\phi)^\wedge)^n [R\rho + t^\wedge R\phi] = -\sum_{n=0}^{\infty} \frac{1}{n!}((R\phi)^\wedge)^n t + R\sum_{n=0}^{\infty} \frac{1}{(n+1)!}(\phi^\wedge)^n \rho + t$$

$$\Leftrightarrow \sum_{n=0}^{\infty} \frac{1}{(n+1)!}((R\phi)^\wedge)^n [t^\wedge R\phi] = -\sum_{n=0}^{\infty} \frac{1}{n!}((R\phi)^\wedge)^n t + t$$

$$\Leftrightarrow \sum_{n=0}^{\infty} \frac{1}{(n+1)!}((R\phi)^\wedge)^n [-(R\phi)^\wedge t] = -\sum_{n=0}^{\infty} \frac{1}{n!}((R\phi)^\wedge)^n t + t$$

$$\Leftrightarrow -\sum_{n=0}^{\infty} \frac{1}{(n+1)!}((R\phi)^\wedge)^{n+1} t = -\sum_{n=0}^{\infty} \frac{1}{n!}((R\phi)^\wedge)^n t + t$$

, 故等式成立。■

7. - **SO(3) 右扰动模型** (图3)
   - **SE(3) 右扰动模型** TODO

8. `find_package(<package> [version] [EXACT] [QUIET] [MODULE] [REQUIRED] [[COMPONENTS] [components...]] [OPTIONAL_COMPONENTS components...] [NO_POLICY_SCOPE])`

   Command **find_package** has two modes: **Module mode** and **Config mode**.

$$
\begin{aligned}
\frac{{}^{\mathbf{R}}\partial \mathbf{R}\mathbf{p}}{\partial \mathbf{R}} &= \lim_{\boldsymbol{\theta}\to 0} \frac{(\mathbf{R}\oplus\boldsymbol{\theta})\mathbf{p}\ominus\mathbf{R}\mathbf{p}}{\boldsymbol{\theta}} = \lim_{\boldsymbol{\theta}\to 0} \frac{\mathbf{R}\,\mathrm{Exp}(\boldsymbol{\theta})\mathbf{p}-\mathbf{R}\mathbf{p}}{\boldsymbol{\theta}} \\
&= \lim_{\boldsymbol{\theta}\to 0} \frac{\mathbf{R}(\mathbf{I}+[\boldsymbol{\theta}]_{\times})\mathbf{p}-\mathbf{R}\mathbf{p}}{\boldsymbol{\theta}} = \lim_{\boldsymbol{\theta}\to 0} \frac{\mathbf{R}\,[\boldsymbol{\theta}]_{\times}\,\mathbf{p}}{\boldsymbol{\theta}} \\
&= \lim_{\boldsymbol{\theta}\to 0} \frac{-\mathbf{R}\,[\mathbf{p}]_{\times}\,\boldsymbol{\theta}}{\boldsymbol{\theta}} = -\mathbf{R}\,[\mathbf{p}]_{\times} \qquad \in \mathbb{R}^{3\times 3}\ .
\end{aligned}
$$

图 3: SO(3) 右扰动模型