

## 第二节课习题

高翔

2019 年 5 月 29 日

### 1 习题说明

- 第  $i$  节课习题所有材料打包在  $L_i.zip$  中,  $\forall i = 1 \dots 8$ 。
- 习题分为若干种: **计算类**习题, 需要读者编程计算一个实际问题, 我们会附有参考答案以供自测。**操作类**习题, 会指导读者做一个具体的实验, 给出中间步骤截图或结果。**简述类**习题则提供阅读材料, 需要读者阅读材料后, 回答若干问题。
- 每个习题会有一些的分值。每次习题分值加和为 10 分。你需要获得 8 分以上才能得到“通过”的评价。带 \* 的习题为附加题, 会在总分之外再提供一定的分值, 所以总和可能超过 10 分。换句话说, 你也可以选择一道附加题, 跳过一道正常题。
- 每道习题的给分由助教评判, 简述类习题可能存在一定开放性, 所以评分也存在主观因素。
- 请利用深蓝学院系统提交习题。每次习题我们会记通过与否。提交形式为 word 或 pdf 格式报告, 如有编程习题请提交可编译的源码。
- 为方便读者, 我通常会准备一些阅读材料, 放在 books/或 papers/目录下。请读者按个人需求使用这些材料。它们多数是从网络下载的, 如果侵犯到你的权利, 请及时告诉我。
- 每个习题会标注大致用时, 但视同学个人水平可能会有出入。
- 习题的完成情况会影响你对本课程内容的掌握程度, 请认真、独立完成。**习题总得分较高的同学将获得推荐资格。**

## 2 熟悉 Eigen 矩阵运算 (3 分, 约 2 小时)

Eigen (<http://eigen.tuxfamily.org>) 是常用的 C++ 矩阵运算库, 具有很高的运算效率。大部分需要在 C++ 中使用矩阵运算的库, 都会选用 Eigen 作为基本代数库, 例如 Google Tensorflow, Google Ceres, GTSAM 等。本次习题, 你需要使用 Eigen 库, 编写程序, 求解一个线性方程组。为此, 你需要先了解一些有关线性方程组数值解法的原理。

设线性方程  $\mathbf{Ax} = \mathbf{b}$ , 在  $\mathbf{A}$  为方阵的前提下, 请回答以下问题:

1. 在什么条件下,  $\mathbf{x}$  有解且唯一?
2. 高斯消元法的原理是什么?
3. QR 分解的原理是什么?
4. Cholesky 分解的原理是什么?
5. 编程实现  $\mathbf{A}$  为  $100 \times 100$  随机矩阵时, 用 QR 和 Cholesky 分解求  $\mathbf{x}$  的程序。你可以参考本次课用到的 useEigen 例程。

提示: 你可能需要参考相关的数学书籍或文章。请善用搜索引擎。Eigen 固定大小矩阵最大支持到 50, 所以你会用到动态大小的矩阵。

### 3 几何运算练习 (2 分, 约 1 小时)

下面我们来练习如何使用 Eigen/Geometry 计算一个具体的例子。

设有小萝卜<sup>1</sup>一号和小萝卜二号位于世界坐标系中。小萝卜一号的位姿为:  $\mathbf{q}_1 = [0.55, 0.3, 0.2, 0.2]$ ,  $\mathbf{t}_1 = [0.7, 1.1, 0.2]^T$  ( $\mathbf{q}$  的第一项为实部)。这里的  $\mathbf{q}$  和  $\mathbf{t}$  表达的是  $\mathbf{T}_{cw}$ , 也就是世界到相机的变换关系。小萝卜二号的位姿为  $\mathbf{q}_2 = [-0.1, 0.3, -0.7, 0.2]$ ,  $\mathbf{t}_2 = [-0.1, 0.4, 0.8]^T$ 。现在, 小萝卜一号看到某个点在自身的坐标系下, 坐标为  $\mathbf{p}_1 = [0.5, -0.1, 0.2]^T$ , 求该向量在小萝卜二号坐标系下的坐标。请编程实现此事, 并提交你的程序。

提示:

1. 四元数在使用前需要归一化。
2. 请注意 Eigen 在使用四元数时的虚部和实部顺序。
3. 参考答案为  $\mathbf{p}_2 = [1.08228, 0.663509, 0.686957]^T$ 。你可以用它验证程序是否正确。

---

<sup>1</sup> 此处小萝卜指代机器人。

## 4 旋转的表达 (2 分, 约 1 小时)

课程中提到了旋转可以用旋转矩阵、旋转向量与四元数表达, 其中旋转矩阵与四元数是日常应用中常见的表达方式。请根据课件知识, 完成下述内容的证明。

1. 设有旋转矩阵  $\mathbf{R}$ , 证明  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$  且  $\det \mathbf{R} = +1^2$ 。
2. 设有四元数  $\mathbf{q}$ , 我们把虚部记为  $\boldsymbol{\epsilon}$ , 实部记为  $\eta$ , 那么  $\mathbf{q} = (\boldsymbol{\epsilon}, \eta)$ 。请说明  $\boldsymbol{\epsilon}$  和  $\eta$  的维度。
3. 定义运算  $+$  和  $\oplus$  为:

$$\mathbf{q}^+ = \begin{bmatrix} \eta \mathbf{1} + \boldsymbol{\epsilon}^\times & \boldsymbol{\epsilon} \\ -\boldsymbol{\epsilon}^T & \eta \end{bmatrix}, \quad \mathbf{q}^\oplus = \begin{bmatrix} \eta \mathbf{1} - \boldsymbol{\epsilon}^\times & \boldsymbol{\epsilon} \\ -\boldsymbol{\epsilon}^T & \eta \end{bmatrix}, \quad (1)$$

其中运算  $\times$  含义与  $\wedge$  相同, 即取  $\boldsymbol{\epsilon}$  的反对称矩阵 (它们都成叉积的矩阵运算形式),  $\mathbf{1}$  为单位矩阵。请证明对任意单位四元数  $\mathbf{q}_1, \mathbf{q}_2$ , 四元数乘法可写成矩阵乘法:

$$\mathbf{q}_1 \mathbf{q}_2 = \mathbf{q}_1^+ \mathbf{q}_2 \quad (2)$$

或者

$$\mathbf{q}_1 \mathbf{q}_2 = \mathbf{q}_2^\oplus \mathbf{q}_1. \quad (3)$$

---

<sup>2</sup> 若行列式为-1, 通常称为瑕旋转 (improper rotation, 对应物理当中旋转 + 镜像)。 $\det \mathbf{R} = +1$  主要由定义给出。

## 5 罗德里格斯公式的证明 (2 分, 约 1 小时)

罗德里格斯公式描述了从旋转向量到旋转矩阵的转换关系。设旋转向量长度为  $\theta$ , 方向为  $\mathbf{n}$ , 那么旋转矩阵  $\mathbf{R}$  为:

$$\mathbf{R} = \cos \theta \mathbf{I} + (1 - \cos \theta) \mathbf{n} \mathbf{n}^T + \sin \theta \mathbf{n}^\wedge. \quad (4)$$

我们在课程中仅指出了该式成立, 但没有给出证明。请你证明此式。

提示: 参考[https://en.wikipedia.org/wiki/Rodrigues%27\\_rotation\\_formula](https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula)。

## 6 四元数运算性质的验证 (1 分, 约 1 小时)

课程中介绍了单位四元数可以表达旋转。其中, 在谈论用四元数  $q$  旋转点  $p$  时, 结果为:

$$p' = qpq^{-1}. \quad (5)$$

我们说, 此时  $p'$  必定为虚四元数 (实部为零)。请你验证上述说法。

此外, 上式亦可写成矩阵运算:  $p' = Qp$ 。请根据你的推导, 给出矩阵  $Q$ 。注意此时  $p$  和  $p'$  都是四元数形式的变量, 所以  $Q$  为  $4 \times 4$  的矩阵。

提示: 如果使用第 4 题结果, 那么有:

$$\begin{aligned} p' &= qpq^{-1} = q^+p^+q^{-1} \\ &= q^+q^{-1\oplus}p. \end{aligned} \quad (6)$$

从而可以导出四元数至旋转矩阵的转换方式:

$$R = \text{Im}(q^+q^{-1\oplus}). \quad (7)$$

其中  $\text{Im}$  指取出虚部的内容。

## 7 \* 熟悉 C++11 (2 分, 约 1 小时)

请注意本题为附加题。

C++ 是一门古老的语言，但它的标准至今仍在不断发展。在 2011 年、2014 年和 2017 年，C++ 的标准又进行了更新，被称为 C++11，C++14，C++17。其中，C++11 标准是最重要的一次更新，让 C++ 发生了重要的改变，也使得近年来的 C++ 程序与你在课本上（比如谭浩强）学到的 C++ 程序有很大的不同。你甚至会惊叹这是一种全新的语言。C++14 和 C++17 则是对 11 标准的完善与扩充。

越来越多的程序开始使用 11 标准，它也会让你在写程序时更加得心应手。本题中，你将学习一些 11 标准下的新语法。请参考本次作业 books/目录下的两个 pdf，并回答下面的问题。

设有类 A，并有 A 类的一组对象，组成了一个 vector。现在希望对这个 vector 进行排序，但排序的方式由 A.index 成员大小定义。那么，在 C++11 的语法下，程序写成：

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 class A {
8 public:
9     A(const int& i) : index(i) {}
10    int index = 0;
11 };
12
13 int main() {
14     A a1(3), a2(5), a3(9);
15     vector<A> avec{a1, a2, a3};
16     std::sort(avec.begin(), avec.end(), [](const A&a1, const A&a2) {return a1.index<a2.index;});
17     for ( auto& a: avec ) cout<<a.index<<" ";
18     cout<<endl;
19     return 0;
20 }
```

请说明该程序中哪些地方用到了 C++11 标准的内容。提示：请关注范围 for 循环、自动类型推导、lambda 表达式等内容。