第二节课习题

高翔

2021 年 9 月 8 日

1 习题说明

- 第 i 节课习题所有材料打包在 Li.zip 中，$\forall i = 1 \ldots 8$。
- 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。
- 每个习题会有一定的分值。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**。

备注：

- 本习题内容更新于 2021 年 9 月。考虑到大家的知识水平增加，本次更新增加了一些作业内容和难度。

2 熟悉 Eigen 矩阵运算（2 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 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$，在 $\boldsymbol{A}$ 为方阵的前提下，请回答以下问题：

1. 在什么条件下，$\boldsymbol{x}$ 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 $\boldsymbol{A}$ 为 100×100 随机矩阵时，用 QR 和 Cholesky 分解求 $\boldsymbol{x}$ 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 矩阵论基础 （2 分，约 2 小时）

除了我们本科学过的基础线性代数之外，大部分研究生课程还会开设矩阵论课程，以作为对本科阶段知识的扩充。对于很多线性问题（SLAM 里也会碰到许多线性问题），了解一些矩阵论基础知识是很有好处的。我们在附件中为大家提供了张贤达老师的《矩阵分析与应用》。请参考该书内容（或者你能找到的其他书籍），回答以下问题。

1. 什么是正定矩阵和半正定矩阵？
2. 对于方阵 $\boldsymbol{A}$，它的特征值是什么？特征向量是什么？特征值一定是实数吗？如何计算一个矩阵的特征值？
3. 什么是矩阵的相似性？相似性反映了什么几何意义？
4. 矩阵一定能对角化吗？什么样的矩阵能保证对角化？不能对角化的矩阵能够形成什么样的形式（Jordan 标准形）？
5. 奇异值分解（SVD）是什么意思？
6. 矩阵的伪逆是什么意思（Pseudo inverse）？莫尔——彭多斯逆是如何定义的？怎么计算一个矩阵的伪逆？
7. 对于超定方程：$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 且 $\boldsymbol{A}$ 不可逆时，我们通常计算最小二乘解：$\boldsymbol{x} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|$。线性方程的最小二乘解在代数意义上是可以解析地写出来的。请回答以下小问题：
   (a)在 $\boldsymbol{b} \neq \boldsymbol{0}$ 时，$\boldsymbol{x}$ 的解是什么形式？事实上，我们可以对 $\boldsymbol{A}$ 求奇异值或对于 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 求特征值。请阐述两者之间的关系。
   (b)当 $\boldsymbol{b} = \boldsymbol{0}$ 时，我们希望求 $\boldsymbol{x}$ 的非零解。请说明如何求解 $\boldsymbol{x}$。
   (c)请谈谈你对上述解法在几何意义上的理解。该问题为开放问题。

4 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。

一个机器人上通常会安装许多不同的传感器，而且这些传感器之间还存在固连关系。我们举一个典型的例子。

在世界系 W 下，存在一个运动的机器人 R。按照固定的或者某些开发人员或者领导的特殊喜好，R 系定义在机器人脚部的位置。但是机器人在设计的时候，又定义了 B 系（Body 系，或本体系），位于机器人头部的位置。由于沟通不畅，标定人员把一台激光传感器和一台视觉传感器标定在了 B 系下。我们称激光传感器为 L 系，视觉传感器为 C 系。现在请你完成以下工作：

旋转矩阵再听一下，不太理解

1. 说明一个激光传感器下的看到的点应该如何计算它的世界坐标。
2. 取：$\boldsymbol{q}_{WR} = [0.55, 0.3, 0.2, 0.2], \boldsymbol{t}_{WR} = [0.1, 0.2, 0.3]^T$，$\boldsymbol{q}_{RB} = [0.99, 0, 0, 0.01], \boldsymbol{t}_{RB} = [0.05, 0, 0.5]^T$，$\boldsymbol{q}_{BL} = [0.3, 0.5, 0, 20.1], \boldsymbol{t}_{BL} = [0.4, 0, 0.5]^T$，$\boldsymbol{q}_{BC} = [0.8, 0.2, 0.1, 0.1], \boldsymbol{t}_{BC} = [0.5, 0.1, 0.5]^T$。现在假设相机传感器观察到自身坐标系下的点 $[0.3, 0.2, 1.2]$，请计算：

   (a)这个点在激光系下的坐标；

   (b)这个点在世界系下的坐标。

提示：

1. 本题的数据是随意取的，没有真实意义，产生什么结果都不要奇怪。
2. 四元数在使用前需要归一化。
3. 请注意 Eigen 在使用四元数时的虚部和实部顺序。

这个记住旋转矩阵是正交矩阵，det=1即可。

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

$$\boldsymbol{q}^+ = \begin{bmatrix} \eta\mathbf{1} + \boldsymbol{\varepsilon}^\times & \boldsymbol{\varepsilon} \\ -\boldsymbol{\varepsilon}^\mathrm{T} & \eta \end{bmatrix}, \quad \boldsymbol{q}^\oplus = \begin{bmatrix} \eta\mathbf{1} - \boldsymbol{\varepsilon}^\times & \boldsymbol{\varepsilon} \\ -\boldsymbol{\varepsilon}^\mathrm{T} & \eta \end{bmatrix}, \tag{1}$$

其中运算 $^\times$ 含义与 $^\wedge$ 相同，即取 $\boldsymbol{\varepsilon}$ 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 $\boldsymbol{q}_1, \boldsymbol{q}_2$，四元数乘法可写成矩阵乘法：

$$\boldsymbol{q}_1\boldsymbol{q}_2 = \boldsymbol{q}_1^+\boldsymbol{q}_2 \tag{2}$$

或者

$$\boldsymbol{q}_1\boldsymbol{q}_2 = \boldsymbol{q}_2^\oplus\boldsymbol{q}_1. \tag{3}$$

[1]若行列式为-1，通常称为瑕旋转（inproper rotation，对应物理当中旋转 + 镜像）。$\det \boldsymbol{R} = +1$ 主要由定义给出。

6 罗德里格斯公式的证明（1 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 θ，方向为 $\boldsymbol{n}$，那么旋转矩阵 $\boldsymbol{R}$ 为：

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

1. 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。
2. 请使用此式请明 $\boldsymbol{R}^{-1} = \boldsymbol{R}^T$。

   这里的运算我算的不对，不知道为什么是正交

7 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 $\boldsymbol{q}$ 旋转点 $\boldsymbol{p}$ 时，结果为：

$$\boldsymbol{p}' = \boldsymbol{q}\boldsymbol{p}\boldsymbol{q}^{-1}. \tag{5}$$

我们说，此时 $\boldsymbol{p}'$ 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：$\boldsymbol{p}' = \boldsymbol{Q}\boldsymbol{p}$。请根据你的推导，给出矩阵 $\boldsymbol{Q}$。注意此时 $\boldsymbol{p}$ 和 $\boldsymbol{p}'$ 都是四元数形式的变量，所以 $\boldsymbol{Q}$ 为 4×4 的矩阵。

提示：如果使用第 4 题结果，那么有：

$$\begin{aligned} \boldsymbol{p}' &= \boldsymbol{q}\boldsymbol{p}\boldsymbol{q}^{-1} = \boldsymbol{q}^{+}\boldsymbol{p}^{+}\boldsymbol{q}^{-1} \\ &= \boldsymbol{q}^{+}{\boldsymbol{q}^{-1}}^{\oplus}\boldsymbol{p}. \end{aligned} \tag{6}$$

从而可以导出四元数至旋转矩阵的转换方式：

$$\boldsymbol{R} = \mathrm{Im}(\boldsymbol{q}^{+}{\boldsymbol{q}^{-1}}^{\oplus}). \tag{7}$$

其中 Im 指取出虚部的内容。

8 * 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：

```cpp
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

class A {
public:
    A(const int& i ) : index(i) {}
    int index = 0;
};

int main() {
    A a1(3), a2(5), a3(9);
    vector<A> avec{a1, a2, a3};
    std::sort(avec.begin(), avec.end(), [](const A&a1, const A&a2) {return a1.index<a2.index;});
    for ( auto& a: avec ) cout<<a.index<<" ";
    cout<<endl;
    return 0;
}
```

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。