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，GoogleCeres，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 $Ax = b$，在 A 为方阵的前提下，请回答以下问题：

1. 在什么条件下，x 有解且唯一？

答：当A为方阵且A可逆（即A满秩），x有解且唯一。

2. 高斯消元法的原理是什么？

答：高斯消元法(Gaussian elimination)是求解线性方程组的一种算法，它也可用来求矩阵的秩，以及求可逆方阵的逆矩阵。它通过逐步消除未知数来将原始线性系统转化为另一个更简单的等价的系统。

原理：通过初等行变化将线性方程组的增广矩阵转化为行阶梯矩阵（上三角或下三角矩阵）。通过初等行变换把A编程阶梯矩阵求解

求解过程：

1.构造增广矩阵，即系数矩阵A增加上常数向量b(A|b).

2.通过以交换行、某行乘以非负常数和两行相加这三种初等变化将原系统转化为更简单的三角形式(**triangular form**)

注:这里的初等变化可以通过系数矩阵A乘上初等矩阵E来实现。

从而得到简化的三角方阵组，这样就容易解了

3.最后再使用向后替换算法(Algorithm for Back Substitution)求解得。

3. QR 分解的原理是什么？

答： 将A分解成正交矩阵和上三角矩阵的乘积A=QR，易解Ra=Q^Tb

原理：利用 2 个正交矩阵的乘积一定是正交矩阵，正交矩阵的逆矩阵也是正交矩阵的特点，对正交矩阵进行多次分解，相乘。QR分解的原理主要是把矩阵分解成一个列向量正交矩阵与一个上三角矩阵的迹，原理是将矩阵每个列作为一个基本单元，将其化为正交的基向量，与在这个基向量上的投影长度的QR分解，经常用来解决最小回归问题，也是特征值算法QR算法的基础。

求解过程：

1.对需要求解的特征值的矩阵A进行QR分解；

2.对分解出来的结果进行逆向相乘；

3.将相乘得到的矩阵进行QR分解；

4.对分解出来的结果进行逆向相乘；

4. Cholesky 分解的原理是什么？

答：

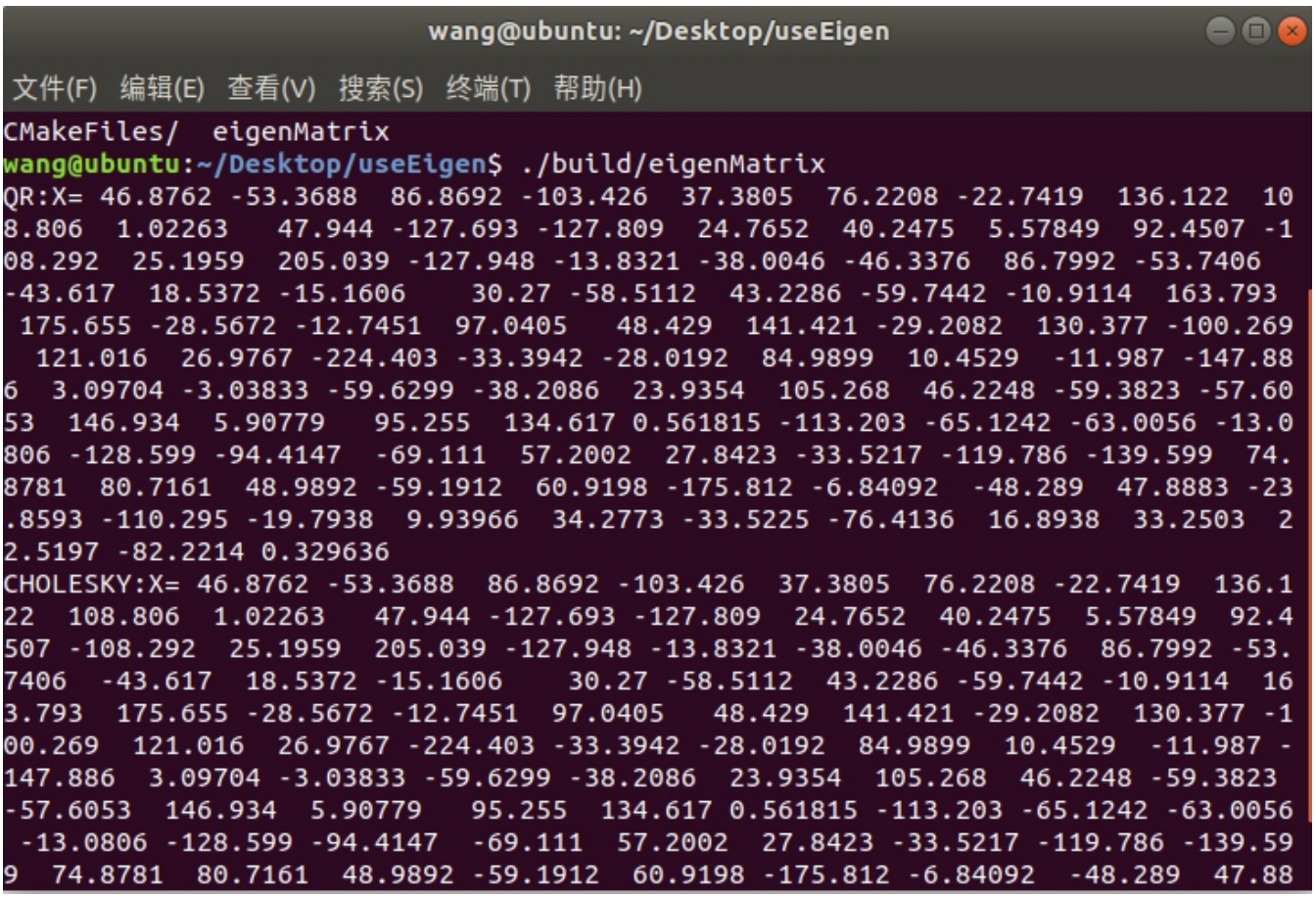
定义：如果矩阵A为n阶对称正定矩阵，则存在一个对角元素为正数的下三角实矩阵L，使得：\(A=LL^T\)，当限定L的对角元素为正时，这种分解是唯一的，称为Cholesky分解。

cholesky分解是把一个对称正定的矩阵表示成一个下三角矩阵L和其转置的乘积的分解。它要求矩阵的所有特征值必须大

于零，故分解的下三角的对角元也是大于零的。Cholesky分解法又称平方根法，是当A为实对称正定矩阵时，LU三角分解法的变形

5. 编程实现 A 为 100×100 随机矩阵时，用 QR 和 Cholesky 分解求 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？

答：给定一个n乘以n的实对称矩阵A，若对于任意长度为 n 的非零向量 x，有 x的转置乘以A再乘以x恒大于0，则矩阵 A 是一个正定矩阵。

给定一个n乘以n的实对称矩阵A，若对于任意长度为 n 的非零向量 x，有 x的转置乘以A再乘以x恒大于等于0，则矩阵 A 是一个正定矩阵。

2. 对于方阵 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？

答：设A为n阶实方阵，如果存在某个数 λ_0 及某个n维非零列向量 η ，使得 $A\eta = \lambda_0\eta$ ，则称 λ_0 是方阵A的一个特征值，η 是方阵A的属于特征值 λ_0 的一个特征向量。

设A为n阶实方阵， λ 为一个参数，称n阶方阵 $\lambda E - A$ 为A的特征方阵，它的行列式 $|\lambda E - A|$ 称为A的特征多项式，把 $|\lambda E - A| = 0$ 称为A的特征方程，把特征方程或特征多项式的根称为A的特征根。

特征值不一定是实数，也可能是虚数。

3. 什么是矩阵的相似性？相似性反映了什么几何意义？

答：在线性代数中，相似矩阵（英语：similar matrix）是指存在相似关系的矩阵。相似关系是两个矩阵之间的一种等价关系。两个n×n矩阵A与B为相似矩阵当且仅当存在一个n×n的可逆矩阵P，使得：{P^{-1}AP=B}

P被称为矩阵A与B之间的相似变换矩阵。

相似的矩阵是同一个线性变换在不同基/坐标系下的的不同描述。

线性变换若粗略看成一个刚体的特定运动。刚体的特定运动是同一个，但坐标系改变的话这个运动的描述函数就会不一样，如果这个函数可用矩阵等价替代的话，一个坐标系就对应着一个矩阵，因此这些矩阵就不同，但这些矩阵必有关系，这个关系就是相似。

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（Jor-dan 标准形）?

答：矩阵不一定能对角化，n阶方阵可对角化的充分必要条件是它有n个线性无关的特征向量。不能对角化的矩阵一定具有多重特征值，对于不能对角化的矩阵也希望找到某种标准形式，使之尽量接近对角化的形式——Jordan标准形。

5. 奇异值分解（SVD）是什么意思？

答：奇异值分解（SVD）是一种用于将矩阵归约成其组成部分的矩阵分解方法，以使后面的某些矩阵计算更简单。

X=U∑V＊,X=U∑V＊，

其中U是m×m阶酉矩阵；∑是m×n阶非负实数对角矩阵；而V*，即V的共轭转置，是n×n阶酉矩阵。这样的分解就称作X的奇异值分解。

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？

答：对于矩阵A，如果存在一个矩阵B，使得AB=BA=I，其中I为与A,B同维数的单位阵，就称A为可逆矩阵（或者称A可逆），并称B是A的逆矩阵，简称逆阵。（此时的逆称为凯利逆）矩阵A可逆的充分必要条件是|A|≠0。奇异矩阵阵或非方阵的矩阵不存在逆矩阵，但可以用函数pinv(A)求其伪逆矩阵。基本语法为X=pinv(A),X=pinv(A,tol),其中tol为误差：max(size(A))*norm(A)*eps。函数返回一个与A的转置矩阵A'同型的矩阵X，并且满足：AXA=A,XAX=X.此时，称矩阵X为矩阵A的伪逆，也称为广义逆矩阵。pinv(A)具有inv(A)的部分特性，但不与inv(A)完全等同。

设 A∈Cm*n,若存在矩阵G∈Cn*m（C为复数域）,使得

(1) AGA = A;

(2) GAG = G;

(3) (AG)H = AG;

(4) (GA)H = GA;

则称 G 为 A 的Moore-Penrose广义逆或加号广义逆，简称为A的M-P逆。A的任意M-P逆记为A+。

7. 对于超定方程：$Ax = b$ 且 A 不可逆时，我们通常计算最小二乘解：$x = \arg\min_x \|Ax - b\|$。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 $b = 0$ 时，x 的解是什么形式？事实上，我们可以对 A 求奇异值或对于 A^TA 求特征值。请阐述两者之间的关系。

答：对于超定方程Ax = 0的解就是 A^TA 最小特征值对应的特征向量。

简单的讲　所有的矩阵都可以进行奇异值分解，不管其是否是方阵以及对称矩阵。当所给的矩阵是对称的方阵，A(T)=A，二者的结果是相同的。也就是说对称矩阵的特征值分解是所有奇异值分解的一个特例。

但是二者还是存在一些小的差异，奇异值分解需要对奇异值从大到小的排序，而且全部是大于等于零。

特征值用来描述方阵，可看做是从一个空间到自身的映射，这也表现在了名字eigenvalue中。奇异值可以描述长方阵或奇异矩阵，可看做是从一个空间到另一个空间的映射。

(b) 当 $b = 0$ 时，我们希望求 x 的非零解。请说明如何求解 x。

答：对于超定方程Ax = 0的解就是 A^TA 最小特征值对应的特征向量。

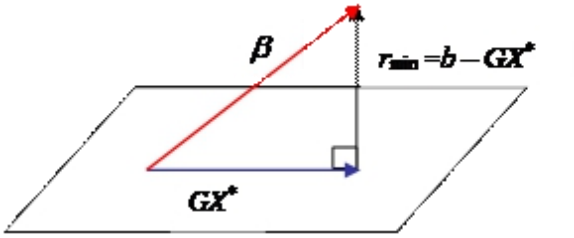
(c) 请谈谈你对上述解法在几何意义上的理解。该问题为开放问题。

答：首先考虑一个简单的超定方程组

$$\begin{bmatrix} 2 & 4 \\ 3 & -5 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 11 \\ 3 \\ 6 \end{bmatrix}$$

该方程组的右端向量是三维向量，系数矩阵的每一列也是三维向量，但待求的未知向量却是二维向量。将系数矩阵按列分块，G =[a1 ，a2]，记右端向量为b。则方程组求解问题可表示为求组合系数x和y使

xa1 + ya2 = b



的向量的线性组合问题。由于两个向量a1 ，a2不构成三维空间的一组基，所以一般情况下这一问题无解。而由向量a1 ，a2张成的子空间span{a1 ，a2}是一张平面，记为p。则超定方程组的最小二乘解实际上是求X* ，使GX* 恰好等于b 在平面p上的投影。而最小二乘解所对应的残差向量则垂直于向量GX*。事实上，由正规方程组

GTGX=GTb

得

GT (b -GX *) = 0

上式的几何意义可解释为：最小二乘解的残差向量与超定方程组的系数矩阵G的所有列向量正交。从而

(X*)T GT (b -GX *) = 0

所以

(GX * , b -GX *) = 0

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：

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：

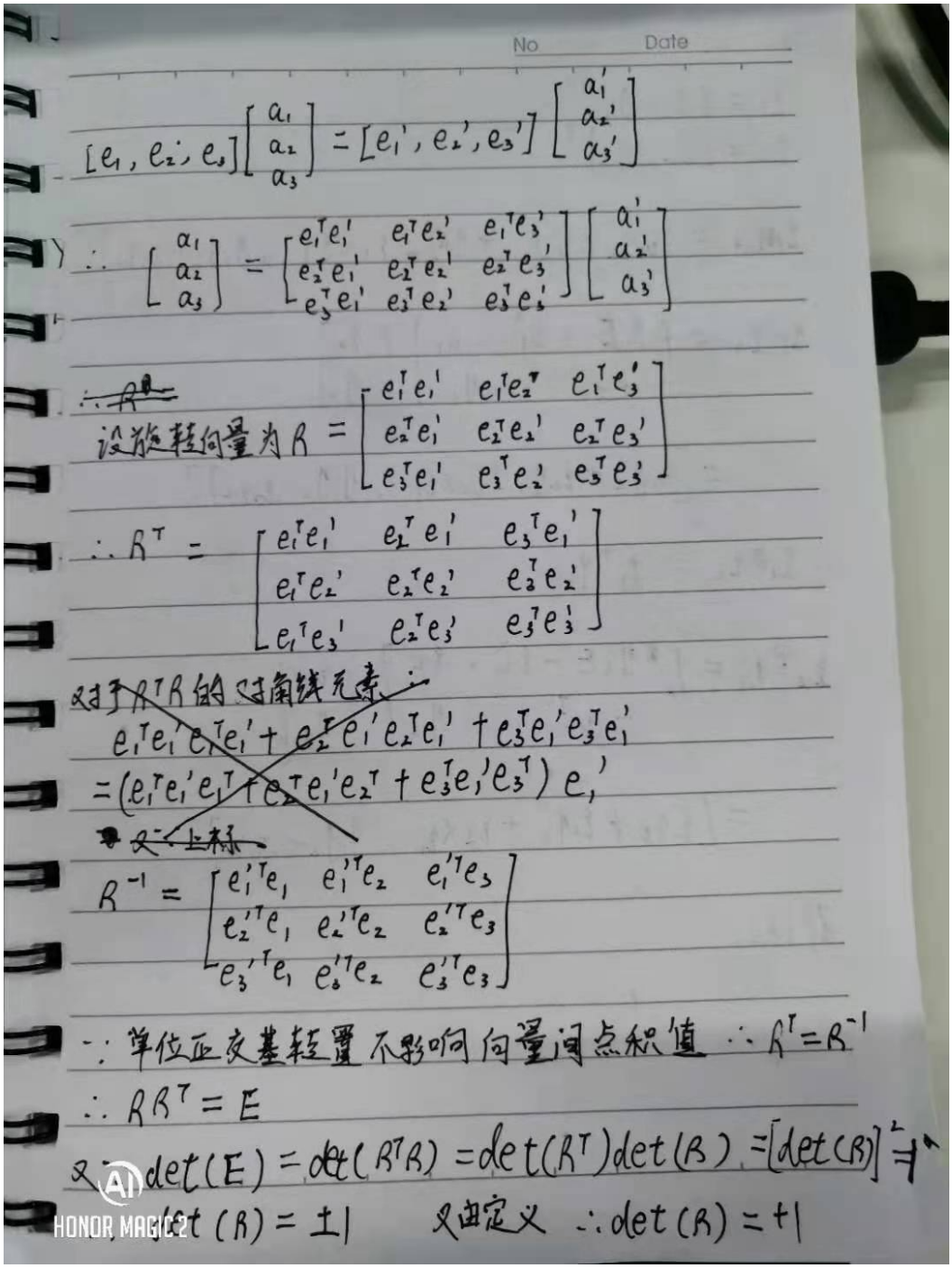
1. 本题的数据是随意取的，没有真实意义，产生什么结果都不要奇怪。
2. 四元数在使用前需要归一化。
3. 请注意 Eigen 在使用四元数时的虚部和实部顺序。

```
wang@ubuntu:~/Desktop/作业/wang_第二章作业/little_radish/build$ cmake ..
-- Configuring done
-- Generating done
-- Build files have been written to: /home/wang/Desktop/作业/wang_第二章作业/lit
tle_radish/build
wang@ubuntu:~/Desktop/作业/wang_第二章作业/little_radish/build$ make
Scanning dependencies of target robot
[ 50%] Building CXX object CMakeFiles/robot.dir/robot.cpp.o
[100%] Linking CXX executable robot
[100%] Built target robot
wang@ubuntu:~/Desktop/作业/wang_第二章作业/little_radish/build$ ./
CMakeFiles/ robot
wang@ubuntu:~/Desktop/作业/wang_第二章作业/little_radish/build$ ./robot
这个点在激光系下的坐标  0.86452 -0.325616   0.97537
这个点在世界系下的坐标-0.0231094   0.737298    1.04868
wang@ubuntu:~/Desktop/作业/wang_第二章作业/little_radish/build$
```

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 R，证明 $R^T R = I$ 且 $\det R = +1$[1]。



2. 设有四元数 q，我们把虚部记为 ε，实部记为 η，那么 $q = (\varepsilon, \eta)$。请说明 ε 和 η 的维度。

答：ε 的维度是3，η 的维度是1.

3. 定义运算 $^{+}$ 和 $^{\oplus}$ 为：

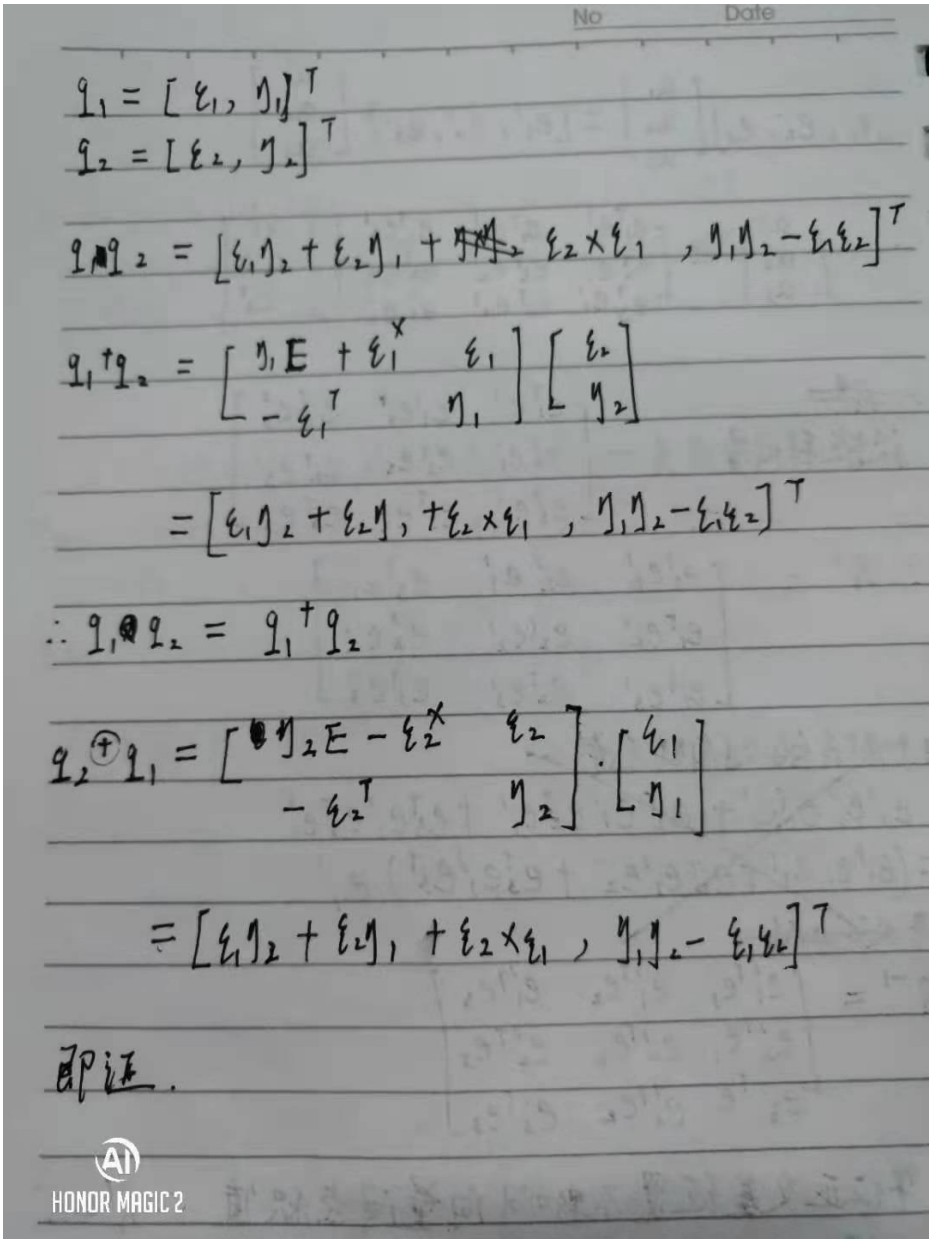
$$q^{+}=\begin{bmatrix}\eta 1+\varepsilon^{\times} & \varepsilon\\ -\varepsilon^{\mathrm{T}} & \eta\end{bmatrix},\quad q^{\oplus}=\begin{bmatrix}\eta 1-\varepsilon^{\times} & \varepsilon\\ -\varepsilon^{\mathrm{T}} & \eta\end{bmatrix},\tag{1}$$

其中运算 $^{\times}$ 含义与 $^{\wedge}$ 相同，即取 ε 的反对称矩阵（它们都成叉积的矩阵运算形式），1 为单位矩阵。请证明对任意单位四元数 q_1，q_2，四元数乘法可写成矩阵乘法：

$$q_1 q_2 = q_1^{+} q_2 \tag{2}$$

或者

$$q_1 q_2 = q_2^{\oplus} q_1 . \tag{3}$$



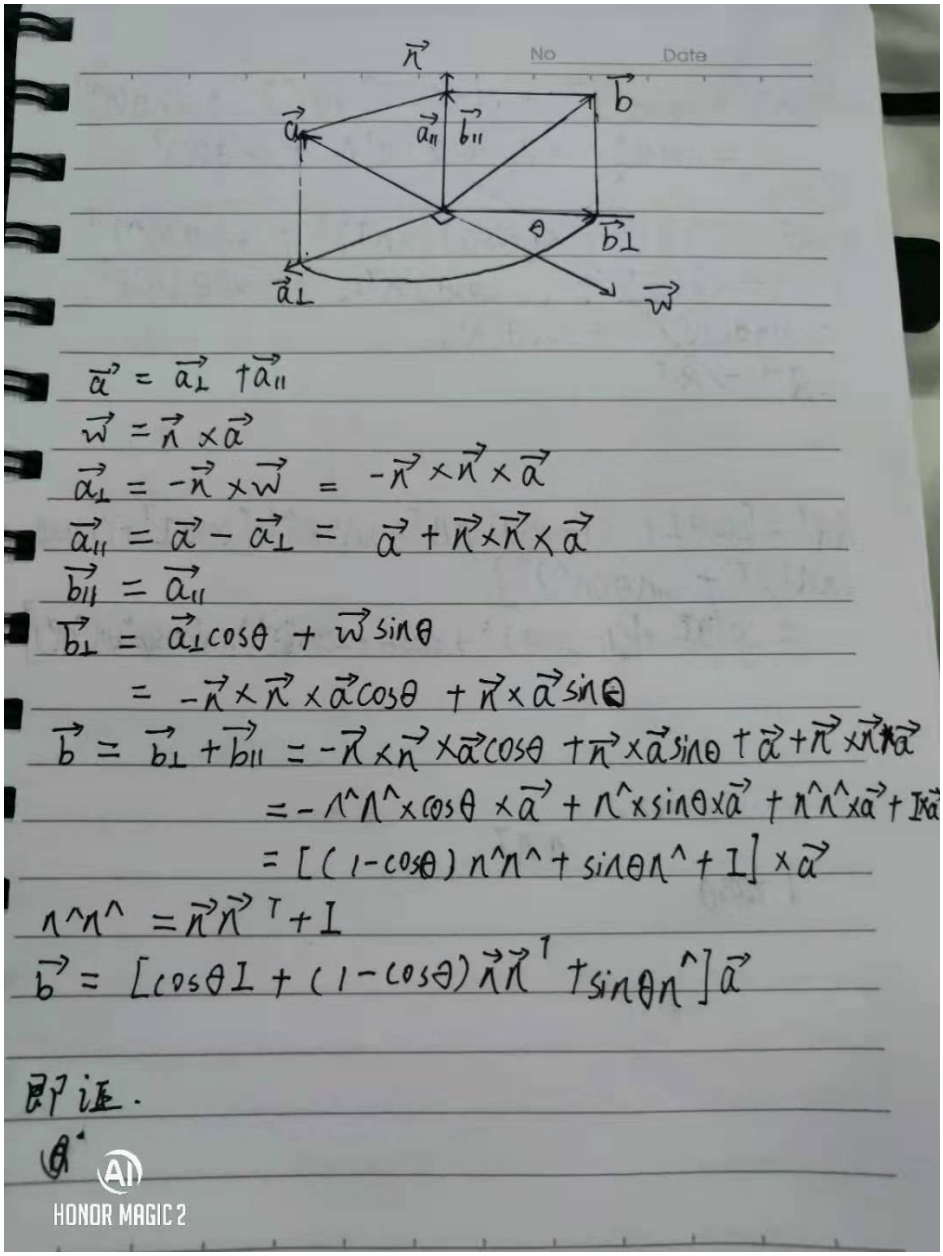
[1]若行列式为-1，通常称为瑕旋转（inproper rotation，对应物理当中旋转 + 镜像）。$\det \boldsymbol{R} = +1$ 主要由定义给出。

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 $\boldsymbol{R}^{-1} = \boldsymbol{R}^T$。

No Date

$RR^T=[\cos\theta I+(1-\cos\theta)nn^T+\sin\theta n^\wedge][(\cos\theta I)^T+(1-\cos\theta)(nn^T)^T+\sin\theta \cdot(n^\wedge)^T]$

$=[\cos\theta I+(1-\cos\theta)nn^T+\sin\theta n^\wedge][\cos\theta I+(1-\cos\theta)nn^T+\sin\theta(n^\wedge)]$

$=\cos^2\theta I+(1-\cos\theta)^2nn^Tnn^T+\sin^2\theta n^\wedge(n^\wedge)^T+2\cos\theta(1-\cos\theta)nn^T$
$+\sin\theta\cos\theta[n^\wedge+(n^\wedge)^T]+(1-\cos\theta)\sin\theta nn^T[(n^\wedge)^T+n^\wedge)]$

$=\cos^2\theta I+\sin^2\theta[nn^T+n^\wedge(n^\wedge)^T]$

$\because n^\wedge$是反对称矩阵，$\therefore (n^\wedge)^T=-n^\wedge$

又$\because n^\wedge n^\wedge=\vec{n}\cdot\vec{n}^T-I$

$\therefore RR^T=\cos^2\theta I+\sin\theta^2 I$

$=I$

$\therefore R^{-1}=R^T$

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 q 旋转点 p 时，结果为：

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

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 4×4 的矩阵。

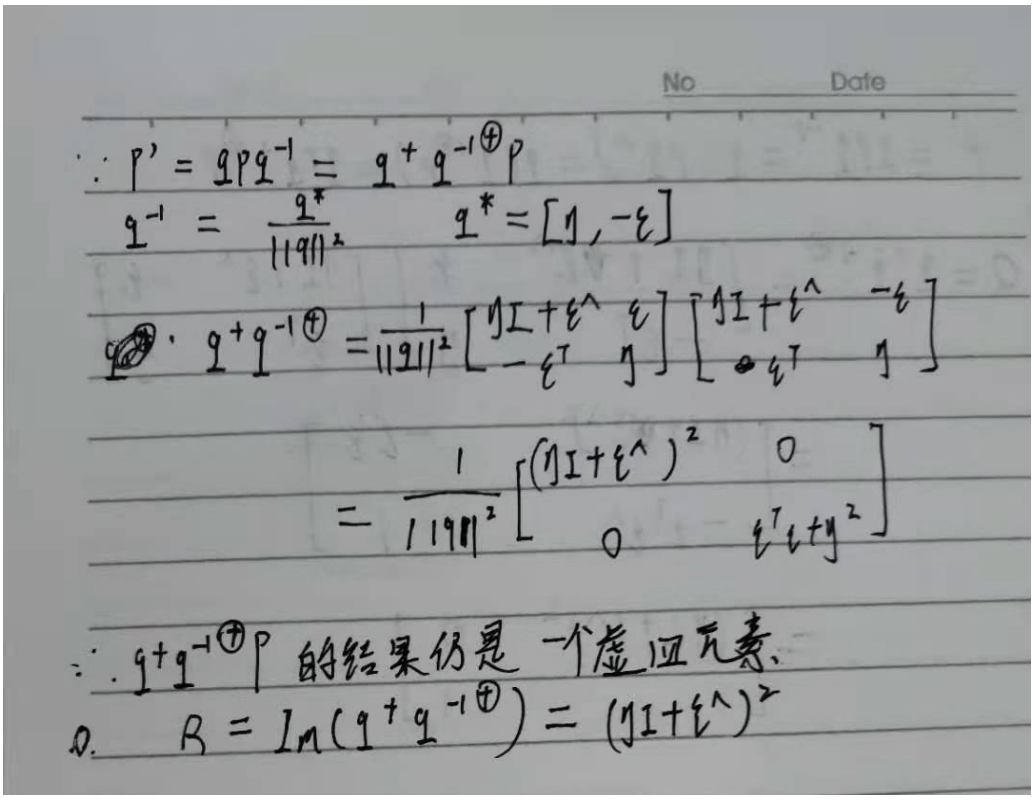
提示：如果使用第 4 题结果，那么有：

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

从而可以导出四元数至旋转矩阵的转换方式：

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

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：

```
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
18
19
```

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表达式内容。

第15行：使用了初始化列表来初始化对象： C++11 把初始化列表的概念绑定到了类型上，并将其称之为 std::initializer_list，允许构造函数或其他函数像参数一样使用初始化列表，这就为类对象的初始化与普通数组和 POD 的初始化方法提供了统一的桥梁。

第16行：使用了lambda表达式来比较元素大小，其中：const A&a1，const A&a2是参数列表，return a1.index<a2.index;是函数体，返回值是布尔型的大小比较结果。

第17行：用auto关键字实现了自动类型推导，让编译器自动设置变量a的类型；

第17行：C++引入了基于范围的for循环，不用下标就能访问元素；