ex02-3D Rigid Body Motion

Eigen Matrix Computation

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?

   A is full rank matrix, x has a solution and is unique

2. 高斯消元法的原理是什么?

   Gaussian elimination is an algorithm in linear algebra that can be used to solve systems of linear equations, as well as to find the rank of a matrix, and to find the inverse of an invertible matrix. The Gaussian elimination method is based on the following principle: by transforming the increasing matrix into a row-order array using the elementary row transform, and then solving the solution of the system of linear equations by backbanding.

3. QR 分解的原理是什么?

   QR decomposition is one of the three ways to decompose a matrix. In this way, the matrix is decomposed into the product of an orthogonal matrix and an upper triangular matrix.QR decomposition is often used to solve linear least squares problems.

4. Cholesky 分解的原理是什么?

   Cholesky decomposition is the representation of a real symmetric positive definite matrix as a product of a lower triangular matrix L and its transpose. The process of solving the equation becomes.

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。

Geometry Computation

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[1]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: $\boldsymbol{q}_1 = [0.55, 0.3, 0.2, 0.2], \boldsymbol{t}_1 = [0.7, 1.1, 0.2]^T$（$\boldsymbol{q}$ 的第一项为实部）。这里的 $\boldsymbol{q}$ 和 $\boldsymbol{t}$ 表达的是 $\boldsymbol{T}_{cw}$，也就是世界到相机的变换关系。小萝卜二号的位姿为 $\boldsymbol{q}_2 = [-0.1, 0.3, -0.7, 0.2], \boldsymbol{t}_2 = [-0.1, 0.4, 0.8]^T$。现在，小萝卜一号看到某个点在自身的坐标系下，坐标为 $\boldsymbol{p}_1 = [0.5, -0.1, 0.2]^T$，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 $\boldsymbol{p}_2 = [1.08228, 0.663509, 0.686957]^T$。你可以用它验证程序是否正确。

```
lwh@lwh:~/Desktop/Learning/shenlanVSLAM/2.3D_Rigid_Body_Motion/PA2/build$ ./eigenMatrix
Qr decomposition time 1.238ms
LLT decomposition time 0.169ms
P2 :  1.08228 0.663509 0.686957
```

Rotation Representation

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 $\det \boldsymbol{R} = +1$[2]。
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

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

$$R = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \begin{bmatrix} e_1^{'} & e_2^{'} & e_3^{'} \end{bmatrix}$$

$$R^T = \begin{bmatrix} e_1^{'T} \\ e_2^{'T} \\ e_3^{'T} \end{bmatrix} \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$$

$$R^TR = \begin{bmatrix} e_1^{'T} \\ e_2^{'T} \\ e_3^{'T} \end{bmatrix} \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix} \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \begin{bmatrix} e_1^{'} & e_2^{'} & e_3^{'} \end{bmatrix} = I$$

quaternion

$$q = (\epsilon, \eta)$$

imagary part 3 dimension

$$\epsilon$$

1 dimension

$$\eta$$

quaternion multiplication

$$q \otimes p = \begin{bmatrix} p_wq_w - p_xq_x - p_yq_y - p_zq_z \\ p_wq_x + p_xq_w + p_yq_z - p_zq_y \\ p_wq_y - p_xq_z + p_yq_w + p_zq_x \\ p_wq_z + p_xq_y - p_yq_x + p_zq_w \end{bmatrix}$$
$$= \begin{bmatrix} p_wq_w - \boldsymbol{p}_v^T\boldsymbol{q}_v \\ p_w\boldsymbol{q}_v + q_w\boldsymbol{p}_v + \boldsymbol{p}_v \times \boldsymbol{q}_v \end{bmatrix}$$

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = [\boldsymbol{q}_1]_L\boldsymbol{q}_2$$
$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = [\boldsymbol{q}_2]_R\boldsymbol{q}_1$$

$$
\begin{aligned}
L &= \begin{bmatrix} q_w & -q_z & qy & q_x \\ q_z & q_w & -q_x & q_y \\ -q_y & q_x & q_w & q_z \\ -q_x & -q_y & -q_z & q_w \end{bmatrix} \\
&= \begin{bmatrix} \eta 1 + \epsilon^{\wedge} & \epsilon \\ -\epsilon & \eta \end{bmatrix}
\end{aligned}
$$

$$[q_2]_R$$

Rodrigues' rotation formula

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：

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

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。

$$RR^T = RR^{-1} = I$$

$$det(R) = det(r_1, r_2, r_3) = r_1 \cdot (r_2 \times r_3) = 1$$

Quaternion Computation

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。

assume that

$$q = \begin{bmatrix} \epsilon \\ \eta \end{bmatrix}, q^{-1} = \begin{bmatrix} -\epsilon \\ \eta \end{bmatrix}$$

$$p = \begin{bmatrix} r_b \\ 0 \end{bmatrix}, p^{-1} = \begin{bmatrix} r_b' \\ 0 \end{bmatrix}$$

then

$$p' = q^{+} q^{-1(+)} p = \begin{bmatrix} \eta I + \epsilon^{\times} & \epsilon \\ -\epsilon^T & \eta \end{bmatrix} \begin{bmatrix} \eta I - \epsilon^{\times} & \epsilon \\ -\epsilon^T & \eta \end{bmatrix} \begin{bmatrix} r_b \\ 0 \end{bmatrix} = \begin{bmatrix} -(\eta I - \epsilon^{\times})^2 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} r_b \\ 0 \end{bmatrix}$$

we get

$$r_b' = (-(\eta I - \epsilon^{\times})^2 - \epsilon\epsilon^T)r_b$$

$$R = -(\eta I - \epsilon^{\times})^2 - \epsilon\epsilon^T$$

C++11

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：

```
#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。

```
#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;
}
```