



深蓝学院
shenlanxueyuan.com

第二次作业讲解



主讲人 王一鸣



1. 熟悉 Eigen 矩阵运算

1. 在什么条件下, x 有解且唯一?

答: 当且仅当 A 满秩

方程组类型	表达式	有解且唯一的条件	无穷解条件
齐次线性方程组	$Ax=0$	$R(A)=n$ 时, 有唯一解, 零解	$R(A)<n$ 时, 方程组有无穷非零解
非齐次线性方程组	$Ax=b$	$R(A)=R(B)=n$	$R(A)=R(B)<n$

2. 高斯消元法的原理是什么?

答: 通过初等变换将 A 变成阶梯形求解。

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 $Ra=Q^Tb$

1. 熟悉 Eigen 矩阵运算

4. Cholesky 分解的原理是什么?

答：将 A 分解成 $A = LL^T$, 其中 L 是下三角矩阵。然后, 先求解 $Lc = b$, c 为中间结果；再求解 $L^T a = c$, 得到最终结果。此过程要求 A 实对称且正定。

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。

2. 几何运算练习

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： $\mathbf{q}_1 = [0.55, 0.3, 0.2, 0.2]$, $\mathbf{t}_1 = [0.7, 1.1, 0.2]^T$ (\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。

3. 旋转的表达

1. 设有旋转矩阵 R , 证明 $R^T R = I$ 且 $\det R = +1$

答:

$$R = \begin{bmatrix} e_1^T e'_1 & e_1^T e'_2 & e_1^T e'_3 \\ e_2^T e'_1 & e_2^T e'_2 & e_2^T e'_3 \\ e_3^T e'_1 & e_3^T e'_2 & e_3^T e'_3 \end{bmatrix}, R^T R = \begin{bmatrix} e_1^T e'_1 & e_2^T e'_1 & e_3^T e'_1 \\ e_1^T e'_2 & e_2^T e'_2 & e_3^T e'_2 \\ e_1^T e'_3 & e_2^T e'_3 & e_3^T e'_3 \end{bmatrix} \begin{bmatrix} e_1^T e'_1 & e_1^T e'_2 & e_1^T e'_3 \\ e_2^T e'_1 & e_2^T e'_2 & e_2^T e'_3 \\ e_3^T e'_1 & e_3^T e'_2 & e_3^T e'_3 \end{bmatrix} = I$$

$\det R = 1$ 主要根据定义得到。由于 R 是正交阵, 它的列向量是三个互为正交的向量, 故 $\det R$ 为 $+1$ 或 -1 。 -1 的称为反射旋转, $+1$ 的则是通常意义上的旋转。

2. 设有四元数 q , 请说明虚部和实部的维度。

答: 虚部3维, 实部1维

3. 旋转的表达

3. 定义运算 $+$ 和 \oplus 为:

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

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 $\mathbf{q}_1, \mathbf{q}_2$, 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

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

4. 罗德里格斯公式的证明

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 \mathbf{n} ，那么旋转矩阵 \mathbf{R} 为：

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

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. 请使用此式证明 $\mathbf{R}^{-1} = \mathbf{R}^T$ 。

5. 四元数运算性质的验证

使用第三题结果：

设用 \mathbf{q} 对 \mathbf{p} 进行旋转，那么旋转之后的点变为：

$$\mathbf{q}\mathbf{p}\mathbf{q}^{-1} = \mathbf{q}^+ (\mathbf{q}^{-1})^{\oplus} \mathbf{p}$$

因此

$$\mathbf{q}^+ (\mathbf{q}^{-1})^{\oplus} = \begin{bmatrix} \eta 1 + \boldsymbol{\varepsilon}^{\times} & \boldsymbol{\varepsilon} \\ -\boldsymbol{\varepsilon}^T & \eta \end{bmatrix} \begin{bmatrix} \eta 1 + \boldsymbol{\varepsilon}^{\times} & -\boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon}^T & \eta \end{bmatrix} = \begin{bmatrix} \eta^2 1 + 2\eta \boldsymbol{\varepsilon}^{\times} + (\boldsymbol{\varepsilon}^{\times})^2 & 0 \\ 0 & 1 \end{bmatrix}$$

根据矩阵维度，取左上角块即可。此式亦说明旋转之后必得纯虚四元数。

6. 熟悉 C++11

使用第三题结果：

设用 q 对 p 进行旋转，那么旋转之后的点变为：

$$qpq^{-1} = q^+(q^{-1})^{\oplus} p$$

因此

$$q^+(q^{-1})^{\oplus} = \begin{bmatrix} \eta 1 + \epsilon^{\times} & \epsilon \\ -\epsilon^T & \eta \end{bmatrix} \begin{bmatrix} \eta 1 + \epsilon^{\times} & -\epsilon \\ \epsilon^T & \eta \end{bmatrix} = \begin{bmatrix} \eta^2 1 + 2\eta \epsilon^{\times} + (\epsilon^{\times})^2 & 0 \\ 0 & 1 \end{bmatrix}$$

根据矩阵维度，取左上角块即可。此式亦说明旋转之后必得纯虚四元数。

6. 熟悉 C++11

1. lamda 表达式

```
std::sort(avec.begin(), avec.end(), [](const A&a1, const A&a2) {return a1.index<a2.index;});
```

上边代码使用了 C++11 中的,有函数体,没有函数名,其定义方式如下;

[外部变量访问方式说明符] (参数表) -> 返回值类型

```
{  
语句块  
}
```

2.自动类型推导

```
auto& a: avec
```

从初始化表达式中推断出变量的数据类型, 使用 `auto` 时,编译器在编译时会根据上下文情况,确定

`auto` 变量的真正类型,不会影响程序运行效率,且 `auto` 不能用来声明函数的返回值.

6. 熟悉 C++11

3. 序列 for 循环

```
for ( auto& a: avec )
```

序列 for 循环语句允许重复遍历一组序列,而这组序列可以是任何可以重复遍历的序列,如由 `begin()` 和 `end()` 函数定义的 STL 序列。所有的标准容器都可用作这种序列,同时它也同样可以是 `std::string`, 初始化列表(list),数组,以及任何由 `begin()` 和 `end()` 函数定义的序列。

5. 初始化列表

```
vector<A> avec{a1, a2, a3};
```

C++11 把初始化列表的概念绑定到了类型上,称之为 `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,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

6. 熟悉 C++11

5. 非静态成员在类定义时初始化

```
int index = 0;
```

类型	是否可以在类定义时初始化	
	C++03	C++11
static const	是	是
const 非 static	否	是
static 非 const	否	否
非 static 非 const	否	是

感谢各位聆听 !

Thanks for Listening

