2. 熟悉Eigen矩阵运算

1. 当A的秩等于增广矩阵$[A|b]$的秩时, $Ax=b$有解且唯一.
2. 高斯消元法的原理是, 通过用初等行变换将增广矩阵化为行阶梯阵, 然后通过回带求解线性方程组的解.
3. 任意实数方阵A, 都能被分解为$A=QR$. 这里的Q为正交单位阵, 即$Q^TQ=I$. R是一个上三角矩阵. 这种分解被称为QR分解.
4. 设A是一个n阶厄米特正定矩阵(Hermitian positive-definite matrix). Cholesky分解的目标是把A变成: $A=LL^T$, L是下三角矩阵.
5. 见附件.

3. 几何运算

见附件, 运算结果与答案不同, 感觉答案应该有问题, 答案的算法刚好反过来了.

4. 旋转表达

1. 对于旋转矩阵R, 给定任意非零向量n, 其旋转过后, 长度不变. 即:

$$(R\cdot n)^T(R\cdot n)=n^Tn$$

   则有: $n^TR^TRn=n^Tn$, 对任意向量n均成立, 从而有$R^TR=I$. 对于不共面的三个向量a,b,c, 对每一个向量进行R的旋转, 则$a\times b\cdot c$的值不变.
   即$det([a\quad b\quad c])=det(R\cdot[a\quad b\quad c])$, 从而可知$\det(R)=+1$
2. ϵ为3维, η为1维.
3. 定义$\dot{r}=r_0+ir_x+jr_y+kr_z$, 我们有:

$$\begin{aligned}\dot{r}\dot{q}=&(r_0q_0-r_xq_x-r_yq_y-r_zq_z)\\&+i(r_0q_x+r_xq_0+r_yq_z-r_zq_y)\\&+j(r_0q_y-r_xq_z+r_yq_0+r_zq_x)\\&+k(r_0q_z+r_xq_y-r_yq_x+r_zq_0)\end{aligned}$$

也可以通过矩阵相乘来表示:

$$\dot{r}\dot{q}=\begin{bmatrix}r_0&-r_x&-r_y&-r_z\\r_x&r_0&-r_z&r_y\\r_y&r_z&r_0&-r_x\\r_z&-r_y&r_x&r_0\end{bmatrix}\dot{q}=\hat{R}\dot{q}$$

或者有:

$$\dot{q}\dot{r}=\begin{bmatrix}r_0&-r_x&-r_y&-r_z\\r_x&r_0&r_z&-r_y\\r_y&-r_z&r_0&r_x\\r_z&r_y&-r_x&r_0\end{bmatrix}\dot{q}=\bar{R}\dot{q}$$

可以看到$\hat{R}$与$\bar{R}$的区别是其左下三角矩阵被转置了.

5. 罗德里格斯公式

① 罗德里格斯公式的证明

向量$\mathbf{v}$绕着$\mathbf{n}$旋转θ角度, $\mathbf{v}$可以分解两部分, 为垂直于$\mathbf{n}$的部分$\mathbf{v}_\parallel$和平行于$\mathbf{n}$的部分$\mathbf{v}_\perp$, 即

$$\mathbf{v}=\mathbf{v}_\parallel+\mathbf{v}_\perp$$

我们有:

$$\begin{aligned}\mathbf{v}_\parallel&=(\mathbf{n}\cdot\mathbf{v})\mathbf{n}\\\mathbf{v}_\perp&=\mathbf{v}-\mathbf{v}_\parallel=\mathbf{v}-(\mathbf{n}\cdot\mathbf{v})\mathbf{n}=-\mathbf{n}\times(\mathbf{n}\times\mathbf{v})\end{aligned}$$

显然旋转过后, $\mathbf{v}_{\parallel}$不变, 因此, 只需计算$\mathbf{v}_{\perp}$. 以$\mathbf{v}_{\perp}$方向为$\mathbf{x}$轴方向, 以$\mathbf{n}$方向为$\mathbf{z}$轴方向, 构建局部坐标系, 可知$\mathbf{v}_{\parallel}$, 旋转过后为

$$\mathbf{v}_{\perp rot} = \cos\theta\mathbf{v}_{\perp} + \sin\theta\mathbf{n}\times\mathbf{v}_{\perp} = \cos\theta\mathbf{v}_{\perp} + \sin\theta\mathbf{n}\times\mathbf{v}$$

从而有

$$\begin{aligned}
\mathbf{v}_{rot} &= \mathbf{v}_{\parallel} + \mathbf{v}_{\perp rot} \\
&= \mathbf{v}_{\parallel} + \cos\theta\mathbf{v}_{\perp} + \sin\theta\mathbf{n}\times\mathbf{v}_{\perp} \\
&= \mathbf{v}_{\parallel} + \cos\theta(\mathbf{v} - \mathbf{v}_{\parallel}) + \sin\theta\mathbf{n}\times\mathbf{v} \\
&= \cos\theta\mathbf{v} + (1-\cos\theta)\mathbf{v}_{\parallel} + \sin\theta\mathbf{n}\times\mathbf{v} \\
&= \cos\theta\mathbf{v} + (1-\cos\theta)\mathbf{n}\mathbf{n}^T\mathbf{v} + \sin\theta\mathbf{n}^{\wedge}\mathbf{v} \\
&= (\cos\theta + (1-\cos\theta)\mathbf{n}\mathbf{n}^T + \sin\theta\mathbf{n}^{\wedge})\mathbf{v}
\end{aligned}$$

因此, 有$R = \cos\theta\mathbf{I} + (1-\cos\theta)\mathbf{n}\mathbf{n}^T + \sin\theta\mathbf{n}^{\wedge}$

② $R^{-1} = R^T$

旋转轴不变, θ取反, 有:

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

显然, $\cos\theta\mathbf{I} + (1-\cos\theta)\mathbf{n}\mathbf{n}^T$是对称矩阵, 而$-n^{\wedge} = (n^{\wedge})^T$, 因此, $R^{-1} = R^T$.

6. 四元数运算性质的验证

虽然任意的单位四元素保持点积，但无法将纯虚四元素映射为纯虚四元素, 所以简单的四元素乘积无法表示旋转. 因此, 四元素的旋转定义为如下形式:

$$p' = qpq^{-1} = (\mathcal{Q}p)q^{-1} = \bar{\mathcal{Q}}^*\mathcal{Q}p = \bar{\mathcal{Q}}^T\mathcal{Q}p$$

这里p是纯虚四元素, 表示一个3d空间中的向量. 通过计算我们可以得到

$$\begin{aligned}
Q = \bar{\mathcal{Q}}^T\mathcal{Q} &= \begin{bmatrix} q_0 & q_x & q_y & q_z \\ -q_x & q_0 & -q_z & q_y \\ -q_y & q_z & q_0 & -q_x \\ -q_z & -q_y & q_x & q_0 \end{bmatrix} \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & -q_z & q_y \\ q_y & q_z & q_0 & -q_x \\ q_z & -q_y & q_x & q_0 \end{bmatrix} \\
&= \begin{bmatrix} q\cdot q & 0 & 0 & 0 \\ 0 & (q_0^2+q_x^2-q_y^2-q_z^2) & 2(q_xq_y-q_0q_z) & 2(q_xq_z+q_0q_y) \\ 0 & 2(q_yq_x+q_0q_z) & (q_0^2-q_x^2+q_y^2-q_z^2) & 2(q_yq_z-q_0q_x) \\ 0 & 2(q_zq_x-q_0q_y) & 2(q_xq_y+q_0q_x) & (q_0^2-q_x^2-q_y^2+q_z^2) \end{bmatrix}
\end{aligned}$$

从而有

$$R = \begin{bmatrix} (q_0^2+q_x^2-q_y^2-q_z^2) & 2(q_xq_y-q_0q_z) & 2(q_xq_z+q_0q_y) \\ 2(q_yq_x+q_0q_z) & (q_0^2-q_x^2+q_y^2-q_z^2) & 2(q_yq_z-q_0q_x) \\ 2(q_zq_x-q_0q_y) & 2(q_xq_y+q_0q_x) & (q_0^2-q_x^2-q_y^2+q_z^2) \end{bmatrix}$$

7. c++11

见注释

```
#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}; //c++11 构造函数
    std::sort(avec.begin(), avec.end(),
        [](const A&a1, const A&a2) {return a1.index<a2.index;}); // c++11 lambda 表达式
    for ( auto& a: avec ) cout<<a.index<<" "; // c++11 for循环, 以及auto
    cout<<endl;
    return 0;
}
```