0. 说明

本PDF文档为自动生成，如有遗漏的格式错误请及时告知！

---

1. 熟悉 Eigen 矩阵运算

设线性方程 Ax = b,在 A 为方阵的前提下,请回答以下问题:

问1-1：在什么条件下,x 有解且唯一?

A为满秩矩阵，即r(A)=n. (n为方阵的阶数)。

问1-2：高斯消元法的原理是什么?

- 构造增广矩阵$(A|b)$ ；
- 通过初等行变换，将矩阵化为阶梯阵（或三角式）；
- 最后通过回代求解；
- 其核心在于减少方程中的未知数个数，算法复杂度一般为$O(n^3)$。

问1-3：QR 分解的原理是什么?

- 将矩阵分解为Q和R两部分，其中Q是标准正交矩阵，R是一个上三角矩阵；
- 通过将矩阵A提前分解，可以降低解算线性方程的复杂度；
- 求解时一般先求出标准正交矩阵Q，然后就可以快速求出矩阵R；

$$A = QR \Rightarrow R = Q^{-1}A = Q^TA$$

- 常用方法有：Householder 、Gram-Schmidt 、Givens

问1-4：Cholesky 分解的原理是什么?

- 把矩阵分解为一个下三角矩阵以及它的共轭转置矩阵的乘积；
- 矩阵A为实对称正定矩阵；
- 通过将矩阵A提前分解，可以降低解算线性方程的复杂度；

问1-5：你需要使用 Eigen 库,编写程序实现 A 为 100 × 100 随机矩阵时,用 QR 和 Cholesky 分解求 x 的程序。

- 进行Cholesky分解需要矩阵A为实对称正定矩阵，代码中将生成的随机矩阵M，然后令$A = M^TM$

  下面进行非严格证明：

  - 对称

  $$A^T = (M^TM)^T = M^TM = A$$

  - 正定

如果M^TM存在负特征值，则二次多项式$f(x)=x^TM^TMx$存在非零向量x使得$f(x)<0$

但是$f(x)=||Mx||^2\geqslant 0$,所以矩阵A的特征值一定大于等于0.

- 代码：
  - useEigen.cpp

```
//求解Ax=b这个线性方程，其中A为100×100的随机方阵，b为向量
#include<iostream>
#include<ctime>

#include<eigen3/Eigen/Eigen>

using namespace std;
using namespace Eigen;

#define MATRIX_SIZE 100

int main(int argc, char** argv){
    srand((unsigned)time(NULL));
    MatrixXd M = MatrixXd::Random(MATRIX_SIZE, MATRIX_SIZE);
    MatrixXd b = VectorXd::Random(MATRIX_SIZE);
    //Cholesky分解要求矩阵A必须为实对称正交矩阵
    MatrixXd A = M.transpose()*M;
    //QR(Eigen提供了许多QR分解函数，这里选择colPivHouseholderQr)
    VectorXd x_qr = A.colPivHouseholderQr().solve(b);
    assert(b.isApprox(A*x_qr));
    cout<<"Here is a  solution(QR decomposition) x to the equation
Ax=b:"<<endl<<x_qr.transpose()<<endl;
    //Cholesky
    VectorXd x_cholesky = A.llt().solve(b);
    assert(b.isApprox(A*x_cholesky));
    cout<<"Here is a  solution(Cholesky decomposition) x to the equation
Ax=b:"<<endl<<x_cholesky.transpose()<<endl;

    return 0;

}
```

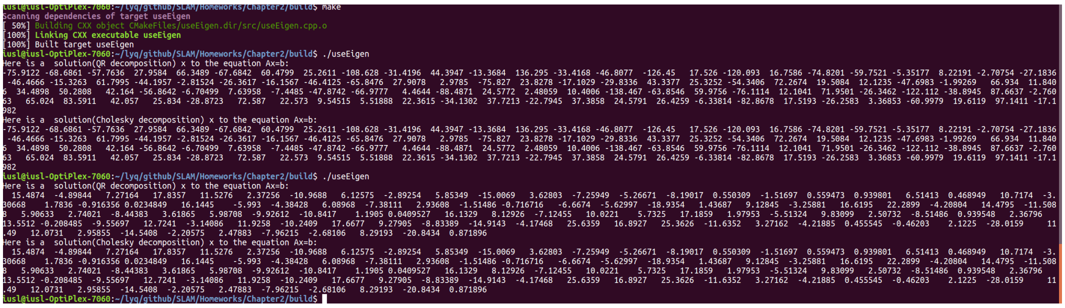
  - CMakeLists.txt

```
cmake_minimum_required(VERSION 2.8.3)
SET(CMAKE_BUILD_TYPE "Release")
PROJECT (Chapters2)

INCLUDE_DIRECTORIES(${PROJECT_SOURCE_DIR}/include)
INCLUDE_DIRECTORIES("/usr/include/eigens3")


SET(SRC_LIST ${PROJECT_SOURCE_DIR}/src/useEigen.cpp)
ADD_EXECUTABLE(useEigen ${SRC_LIST})
```

  - 运行结果



2. 几何运算练习

- useGeometry.cpp

```
#include<iostream>

#include<eigen3/Eigen/Eigen>

using namespace std;
using namespace Eigen;

int main(int argc, char** argv){
    Quaterniond q1={0.55,0.3,0.2,0.2},q2={-0.1,0.3,-0.7,0.2};
    Vector3d t1={0.7,1.1,0.2},t2={-0.1,0.4,0.8},p1={0.5,-0.1,0.2},p2;

    q1=q1.normalized();
    q2=q2.normalized();

    Isometry3d Tcw1(q1),Tcw2(q2);
    Tcw1.pretranslate(t1);
    Tcw2.pretranslate(t2);

    p2=Tcw2* Tcw1.inverse()*p1;
    cout<<"p2 = "<<p2.transpose()<<endl;
    return 0;
}
```

- CMakeLists.txt

```
cmake_minimum_required(VERSION 2.8.3)
SET(CMAKE_BUILD_TYPE "Release")
PROJECT (Chapters2)

add_compile_options(-std=c++11)

INCLUDE_DIRECTORIES(${PROJECT_SOURCE_DIR}/include)
INCLUDE_DIRECTORIES("/usr/include/eigens3")


SET(SRC_LIST ${PROJECT_SOURCE_DIR}/src/useEigen.cpp)
ADD_EXECUTABLE(useEigen ${SRC_LIST})

ADD_EXECUTABLE(useGeometry ${PROJECT_SOURCE_DIR}/src/useGeometry.cpp)
```

- 运行结果

```
iusl@iusl-OptiPlex-7060:~/lyq/github/SLAM/Homeworks/Chapter2/build$ cmake ..
-- Configuring done
-- Generating done
-- Build files have been written to: /home/iusl/lyq/github/SLAM/Homeworks/Chapter2/build
iusl@iusl-OptiPlex-7060:~/lyq/github/SLAM/Homeworks/Chapter2/build$ make
[ 25%] Building CXX object CMakeFiles/useGeometry.dir/src/useGeometry.cpp.o
[ 50%] Linking CXX executable useGeometry
[ 50%] Built target useGeometry
[ 75%] Building CXX object CMakeFiles/useEigen.dir/src/useEigen.cpp.o
[100%] Linking CXX executable useEigen
[100%] Built target useEigen
iusl@iusl-OptiPlex-7060:~/lyq/github/SLAM/Homeworks/Chapter2/build$ ./useGeometry
p2 =  1.08228 0.663509 0.686957
iusl@iusl-OptiPlex-7060:~/lyq/github/SLAM/Homeworks/Chapter2/build$
```

---

3. 旋转的表达

证3-1：设有旋转矩阵 R,证明 $\mathbf{R}^{\mathbf{T}}\mathbf{R} = \mathbf{I}$且 $det\mathbf{R} = +1$

旋转矩阵$\mathbf{R}_{\mathbf{B}}^{\mathbf{A}}$可以表示为：

$$R_B^A = [\hat{X}_B^A \quad \hat{Y}_B^A \quad \hat{Z}_B^A] = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} \cdots\cdots\cdots\cdots\cdots\cdots (3-1)$$

观察式子（3-1）可得：

$$R_B^A = [\hat{X}_B^A \quad \hat{Y}_B^A \quad \hat{Z}_B^A] = \begin{bmatrix} \hat{X}_A^{B^T} \\ \hat{Y}_A^{B^T} \\ \hat{Z}_A^{B^T} \end{bmatrix} = R_A^{B^T} \cdots\cdots\cdots\cdots\cdots\cdots (3-2)$$

从式子（3-2）可得：

$$R_B^A R_A^B = R_A^{B^T} R_A^B = I \Rightarrow 旋转矩阵为正交矩阵$$

因为旋转矩阵是正交矩阵，所以其行列式等于1或者-1，行列式的符号反映了基底的定向变化，因为我们一直使用右手法则，所以基底方向不变，取正号。

如果从几何上来考虑，旋转变换未改变其各向量之间的夹角和模的大小，从而向量构成的“体积”也没变，所以$detR = +1$。

证3-2：设有四元数 q,我们把虚部记为 ε,实部记为 η,那么 q = (ε, η)。请说明 ε 和 η 的维度

　四元数都是由实数加上三个虚数单位 i、j、k 组成，一般可表示为$q = a + bi + cj + dk$，即$\eta = a, \epsilon = [b, c, d]^T$。所以实部的维度为1,虚部的维度为3。

证3-3：定义运算$^+$和$^\oplus$为：

$$q^+ = \begin{bmatrix} \eta\mathbf{I} + \epsilon^\times & \epsilon \\ -\epsilon^T & \eta \end{bmatrix}, q^\oplus = \begin{bmatrix} \eta\mathbf{I} - \epsilon^\times & \epsilon \\ -\epsilon^T & \eta \end{bmatrix}$$

其中运算$^\times$含义与^相同，即取ϵ的反对称矩阵，$\mathbf{I}$为单位矩阵，请证明对任意单位四元数q_1, q_2，四元数乘法可以写成矩阵乘法：

$$q_1 q_2 = q_1^+ q_2 \quad 或\, q_1 q_2 = q_2^{\oplus} q_1$$

令$q_1 = bi + cj + dk + a$，$q_2 = fi + gj + hk + e$，则$\eta_1 = a, \eta_2 = e; \epsilon_1 = [b, c, d]^T, \epsilon_2 = [f, g, h]^T$：

$$\begin{gathered} q_1 q_2 = (bi + cj + dk + a)(fi + gj + hk + e) = \\ (ae - bf - cg - dh) + \\ (be + af - dg + ch)i \\ (ce + df + ag - bh)j \\ (de - cf + bg + ah)k \\ = \\ \begin{bmatrix} a & -d & c & b \\ d & a & -b & c \\ -c & b & a & d \\ -b & -c & -d & a \end{bmatrix} \begin{bmatrix} f \\ g \\ h \\ e \end{bmatrix} = \begin{bmatrix} \eta \mathbf{I} + \epsilon^{\times} & \epsilon \\ -\epsilon^T & \eta \end{bmatrix} q_2 = q_1^+ q_2 \end{gathered}$$

同理可证：$q_1 q_2 = q_2^{\oplus} q_1$

---

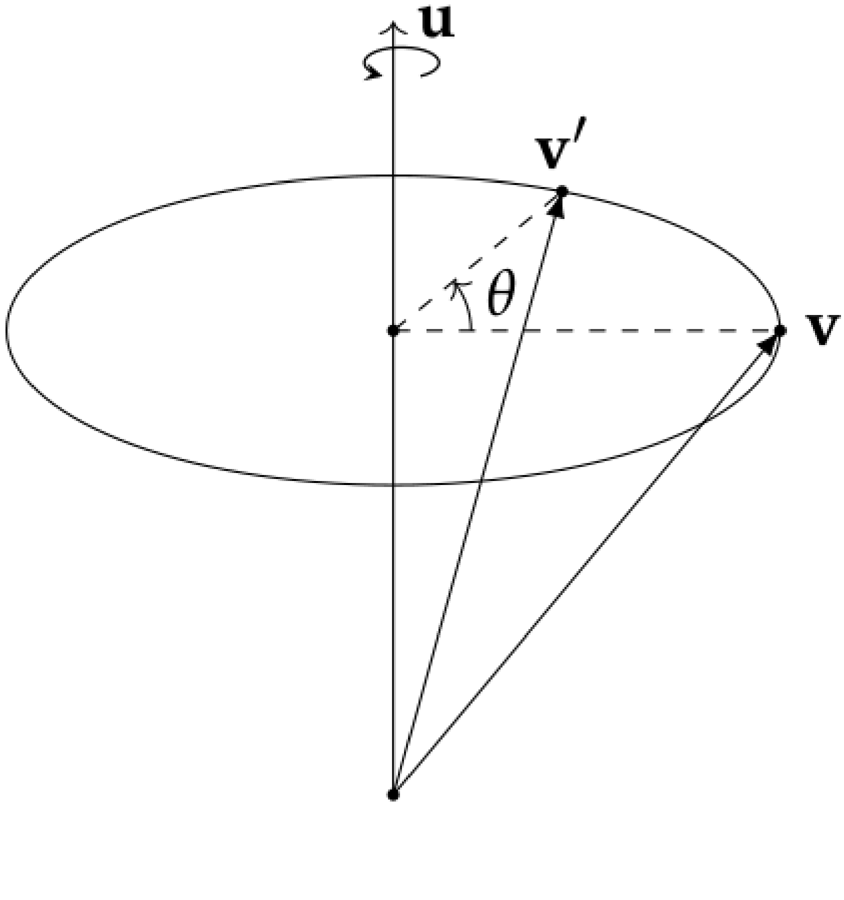
4. 罗德里格斯公式的证明

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 n,那么旋转矩阵 R 为:

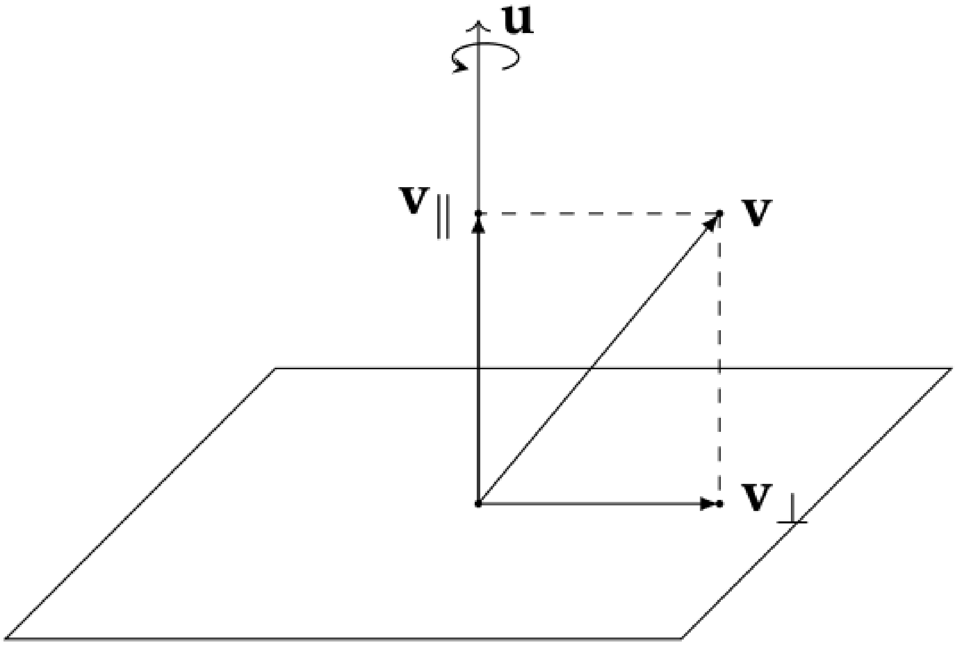
$$R = cos\theta \mathbf{I} + (1 - cos\theta)\mathbf{n}\mathbf{n}^T + sin\theta \mathbf{n}^{\wedge}.$$

证4-1：证明罗德里格斯公式

(这里证明参考了https://github.com/Krasjet/quaternion)



上图表示的意思是向量$\mathbf{v}$绕旋转轴$\mathbf{u}$（单位向量）旋转θ角度后变为$\mathbf{v}'$ 。



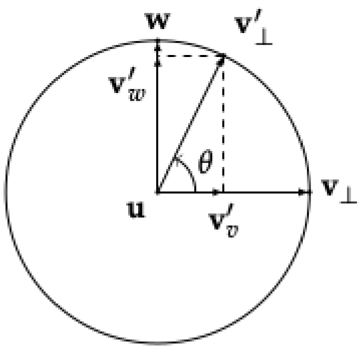
将向量$\mathbf{v}$ 分解为分别平行和垂直旋转轴的两个向量，即$\mathbf{v} = \mathbf{v}_{||} + \mathbf{v}_{\perp}$；从而$\mathbf{v}' = \mathbf{v}'_{||} + \mathbf{v}'_{\perp}$ 。

其中$\mathbf{v}_{||} = (\mathbf{u} \cdot \mathbf{v})\mathbf{u}$，$\mathbf{v}_{\perp} = \mathbf{v} - \mathbf{v}_{||} = \mathbf{v} - (\mathbf{u} \cdot \mathbf{v})\mathbf{u}$.

- $\mathbf{v}_{||}$的旋转

  因为其平行于且重合于旋转轴，所以旋转前后不变，即$\mathbf{v'}_{||} = \mathbf{v}_{||}$。

- $\mathbf{v}_{\perp}$的旋转

  $\mathbf{v}_{\perp}$的旋转位于一个平面内，同时为了方便从数学上描述这个旋转过程，构造一个向量$\mathbf{w} = \mathbf{u} \times \mathbf{v}_{\perp}$，因为$||\mathbf{u}|| = 1$，可得$||\mathbf{w}|| = ||\mathbf{v}_{\perp}||$，从而：

  $$\mathbf{v'}_{\perp} = \mathbf{v'}_{v} + \mathbf{v'}_{w} = cos(\theta)\mathbf{v}_{\perp} + sin(\theta)\mathbf{w} = cos(\theta)\mathbf{v}_{\perp} + sin(\theta)(\mathbf{u} \times \mathbf{v}_{\perp})$$

  综上可得：

  $$\mathbf{v}' = \mathbf{v}'_{||} + \mathbf{v}'_{\perp} = \mathbf{v}_{||} + cos(\theta)\mathbf{v}_{\perp} + sin(\theta)(\mathbf{u} \times \mathbf{v}_{\perp})$$

  又因为$\mathbf{u} \times \mathbf{v}_{\perp} = \mathbf{u} \times (\mathbf{v} - \mathbf{v}_{||}) = \mathbf{u} \times \mathbf{v}$，最后可得：

$$
\begin{aligned}
\mathbf{v}' &= \mathbf{v}_{||} + cos(\theta)\mathbf{v}_{\perp} + sin(\theta)(\mathbf{u}\times\mathbf{v}_{\perp}) \\
&= (\mathbf{u}\cdot\mathbf{v})\mathbf{u} + cos(\theta)(\mathbf{v}-(\mathbf{u}\cdot\mathbf{v})\mathbf{u}) + sin(\theta)(\mathbf{u}\times\mathbf{v}) \\
&= cos(\theta)\mathbf{v} + (1-cos(\theta))(\mathbf{u}\cdot\mathbf{v})\mathbf{u}) + sin(\theta)(\mathbf{u}\times\mathbf{v}) \\
&= [cos(\theta)\mathbf{I} + (1-cos(\theta))\mathbf{u}\mathbf{u}^T + sin(\theta)\mathbf{u}^{\wedge}]\mathbf{v}
\end{aligned}
$$

（这里用到了正交投影公式和叉乘的矩阵运算形式）

在本题中，旋转轴$\mathbf{u}=\mathbf{n}$，故得到$R = cos\theta\mathbf{I} + (1-cos\theta)\mathbf{n}\mathbf{n}^T + sin\theta\mathbf{n}^{\wedge}$

证4-2：利用罗德里格斯公式证明$R^{-1} = R^T$

通过直接计算$RR^T = I$来证明：

```
syms x y z theta real;

n = [x,y,z]';
R = cos(theta)*eye(3)+(1-cos(theta))*n*n'+sin(theta)*skew(n);
```

通过计算可得：

$$
RR^T = \begin{bmatrix} ((1-c\theta)x^2+c\theta)^2+(zs\theta+xy(c\theta-1))^2+(ys\theta-xz(c\theta-1))^2 & xy(c\theta-1)^2(x^2+y^2+z^2-1) & xz(c\theta-1)^2(x^2+y^2+z^2-1) \\ xy(c\theta-1)^2(x^2+y^2+z^2-1) & ((1-c\theta)y^2+c\theta)^2+(xs\theta+yz(c\theta-1))^2+(zs\theta-xy(c\theta-1))^2 & yz(c\theta-1)^2(x^2+y^2+z^2-1) \\ xz(c\theta-1)^2(x^2+y^2+z^2-1) & yz(c\theta-1)^2(x^2+y^2+z^2-1) & ((1-c\theta)z^2+c\theta)^2+(ys\theta+xz(c\theta-1))^2+(xs\theta-yz(c\theta-1))^2 \end{bmatrix}
$$

代入$x^2+y^2+z^2=1$可得$RR^T = I \Rightarrow R^{-1} = R^T$。

---

5. 四元数运算性质的验证

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:

$$p' = qpq^{-1}$$

证5-1：p' 必定为虚四元数(实部为零)

这里的点p写成四元数形式分别为[0,x,y,z]，即[0,v]

$$p' = qpq^{-1} = q[0,\mathbf{v}]q^{-1}$$

令$q = [cos(\theta/2), sin(\theta/2)\mathbf{u}]$，则：

$$p' = [cos(\theta/2), sin(\theta/2)\mathbf{u}][0,\mathbf{v}][cos(\theta/2), -sin(\theta/2)\mathbf{u}]$$

根据Graßmann 积的计算公式可得实部为：

$$
\begin{aligned}
&-sin(\theta/2)cos(\theta/2)\mathbf{u}\cdot\mathbf{v} - (cos(\theta/2)\mathbf{v} + sin(\theta/2)\mathbf{u}\times\mathbf{v})\cdot(-sin(\theta/2)\mathbf{u}) \\
&= -sin(\theta/2)cos(\theta/2)\mathbf{u}\cdot\mathbf{v} + sin(\theta/2)cos(\theta/2)\mathbf{u}\cdot\mathbf{v} + sin^2(\theta/2)(\mathbf{u}\times\mathbf{v})\cdot\mathbf{u} = 0
\end{aligned}
$$

故可证p'必定为虚四元数。

扩：四元数与罗德里格斯公式的关系如下：

$$\mathbf{v}' = q\mathbf{v}q^* = q\mathbf{v}q^{-1} = qq^*\mathbf{v}_{||} + qq\mathbf{v}_{\perp} = \mathbf{v}_{||} + q^2\mathbf{v}_{\perp}$$

从而可得：

$$
\begin{aligned}
\mathbf{v}' &= \mathbf{v}_{||} + q^2\mathbf{v}_{\perp} = [0, (\mathbf{u}\cdot\mathbf{v})\mathbf{u}] + [0, cos\theta\mathbf{v}_{\perp} + sin\theta\mathbf{u}\times\mathbf{v}_{\perp}] \\
&= [0, cos(\theta)\mathbf{v} + (1-cos(\theta))(\mathbf{u}\cdot\mathbf{v})\mathbf{u}) + sin(\theta)(\mathbf{u}\times\mathbf{v})]
\end{aligned}
$$

问5-2：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。

令四元数$q = a + bi + cj + dk$，（**注意这里与第四题不同，实部在前，最后推出来的结果也不一样**）。

- 左乘这个四元数等价于左乘下面这个矩阵：

$$L(q) = \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix}$$

- 右乘这个四元数等价于左乘下面这个矩阵：

$$R(q) = \begin{bmatrix} a & -b & -c & -d \\ b & a & d & -c \\ c & -d & a & b \\ d & c & -b & a \end{bmatrix}$$

则$p' = qpq^{-1} = qpq^* = L(q)R(q)p = L(q)R(q)^T p$，将$a^2 + b^2 + c^2 + d^2 = 1$代入计算结果并化简可得：

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 - 2c^2 - 2d^2 & 2bc - 2ad & 2ac + 2bd \\ 0 & 2bc + 2ad & 1 - 2b^2 - 2d^2 & 2cd - 2ab \\ 0 & 2bd - 2ac & 2ab + 2cd & 1 - 2b^2 - 2c^2 \end{bmatrix}$$

---

6. 熟悉 C++11

```
#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。

- 用到了新的关键字auto；
- 用到了序列for循环；

- Lambda表达式；
- 初始化列表；