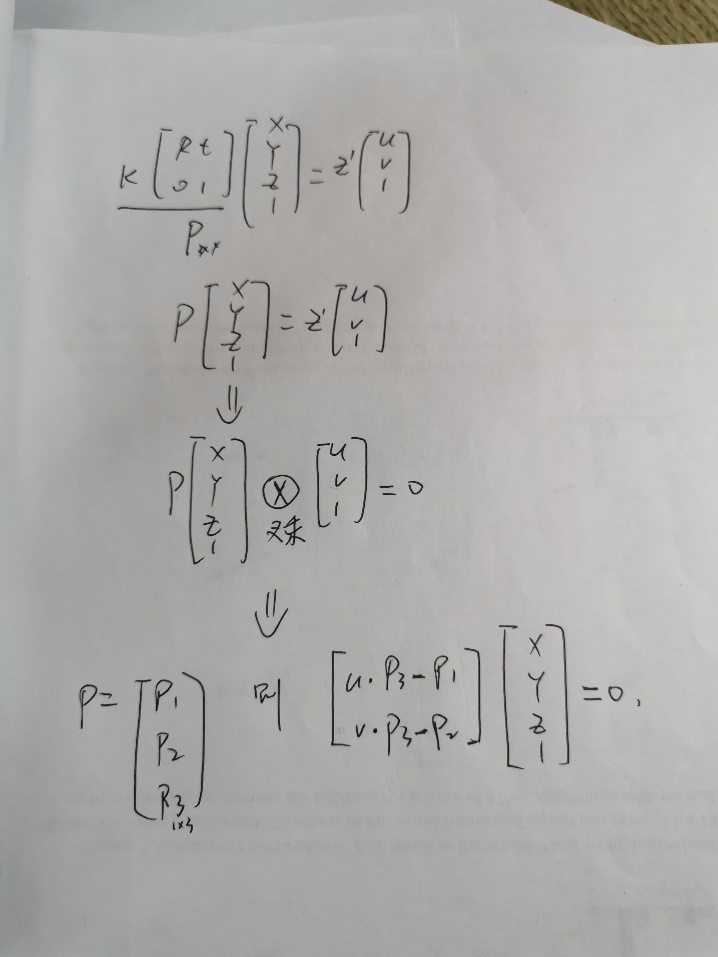
## 1.imu topic订阅topic函数

通过上一个时刻的位姿和imu测量值推算得到当前的位姿(fastPredictIMU函数实现)：

## 2. compensatedParallax2

## 3. triangulatePoint

已知两帧在世界坐标系下的位姿，已知一个地图点在这两帧坐标系下的像素值，求这个地图点在世界坐标系下的坐标



## 4.1 ThreadsConstructA

遍历这个线程的cost function然后对JTJ和-JTb矩阵进行赋值。

流程如下：遍历所有的cost function：

遍历这个cost function的所有优化参数：

得到这个cost function对参数i的雅克比；

得到第i个优化参数的维度，和在雅克比中的位置

遍历这个cost function从第i个参数开始的参数：

得到这个cost function对参数j的雅克比；

得到第j个优化参数的维度，和在雅克比中的位置;

根据i和j的雅克比和位置更新JTJ和-JTb雅克比。

## 4.2 marginalize

a. ThreadsConstructA：根据边缘化之后的残差构造JTJ和JTb矩阵

现在需要求解JTJ\* = -JTf=，从而得到即状态的变化量。其中令JTJ=，其中U矩阵对应保留下的状态，整个的维度为m\*m，V对应需要被边缘化的状态，整个的维度为n\*n。

在求解JTJ的逆矩阵时作者为了保证求逆的可靠性使用了如下的技巧：

## 5. CalibrationExRotation

此函数用于计算imu到相机的旋转矩阵

输入的参数是当前帧和上一帧相机观测到的像素坐标，imu的相对位姿变化

# 6. 残差函数

6.1 FourDOFError 用于四自由度的回环残差函数

6.2 RelativeRTError 用于六个自由度优化的回环残差函数

6.3 IMUFactor imu残差函数

6.4FourDOFWeightError用于四自由度的回环残差函数

6.4 ProjectionTwoFrameOneCamFactor 重投影误差函数

6.5ProjectionTwoFrameTwoCamFactor重投影误差函数

## 7. 参数化方法

7.1 ceres::LocalParameterization \*local\_parameterization = new PoseLocalParameterization();//位姿的参数化方法

7.2 ceres::LocalParameterization\* local\_parameterization = new ceres::QuaternionParameterization();

四元数的参数化方法

7.3 ceres::LocalParameterization\* angle\_local\_parameterization = AngleLocalParameterization::Create();//角度的参数化方法