准备工作

构造 BA 问题实例；

输入 N 个相机 pose 节点；

输入 M 个地标 position 节点；

输入 K 个相机 pose 节点与地标 position 节点约束的二元边；

为 M 个地标节点分别构造 landmark block 实例；

初始化 landmark block

for 每一个 landmark block 实例：

基于当前相机 pose 和地标 position 计算误差 r；

计算观测误差对相机 pose 和地标 position 的雅可比矩阵 J_p 和 J_l；

对 J_l 进行 QR 分解，得到 Q_1、Q_2、R_1、R_2（R_2 为零矩阵）；

计算对应位置元素，构造 landmark block 矩阵如下：

fig 2，即执行NM

$$A = \begin{bmatrix} Q_1^T J_p & R & Q_1^T r \\ Q_2^T J_p & 0 & Q_2^T r \end{bmatrix}$$

end for

初始化 LM 求解器

这个最好参考Ceres跟Jacobian进行耦合，而不是像g2o一样取一样的值(max(diag(H)))，数值稳定性更好

初始化 LM 算法相关参数，主要确定阻尼因子 λ 的初值；

初始化迭代失败计数器 cnt = 0；

开始迭代

for 循环至最大迭代次数：

计算整体增量

for 每一个 landmark block 实例：

为 landmark block 矩阵 A 添加 LM 算法的阻尼因子 λ，得如下矩阵：

$$\hat{A} = \begin{bmatrix} Q_1^T J_p & R & Q_1^T r \\ Q_2^T J_p & 0 & Q_2^T r \\ 0 & \sqrt{\lambda} D & 0 \end{bmatrix}$$

一个3*3对角阵

采用吉文斯矩阵 Q_λ 消除 $\hat{A}$ 中的 $\sqrt{\lambda} D$ 部分，使其形式上类似于 A 矩阵：

关注Givens矩阵Qλ如何构建的，是哪6个Givens矩阵的积？

Givens旋转Qλ消掉landmark的damp，得到B，其中“第二行”只与pose有关，
如要叠加λp则进行叠加，然后GN或者PCG求解，得△xp

$$B = Q_\lambda \hat{A} = \begin{bmatrix} \hat{Q}_1^T J_p & \hat{R} & \hat{Q}_1^T r \\ \hat{Q}_2^T J_p & 0 & \hat{Q}_2^T r \end{bmatrix}$$

基于以下待求解问题，构造每个 landmark block 的增量方程，即为每个 landmark block 计算 $J_p^T \hat{Q}_2 \hat{Q}_2^T J_p$ 和 $J_p^T \hat{Q}_2 \hat{Q}_2^T r$：

$$\Delta x_p = \min_{\Delta x_p} \left\| \hat{Q}_2^T r + \hat{Q}_2^T J_p \Delta x_p \right\|^2 = solve\left\{ \left[J_p^T \hat{Q}_2 \hat{Q}_2^T J_p \right] \Delta x_p = -J_p^T \hat{Q}_2 \hat{Q}_2^T r \right\}$$

end for

累加所有的landmark的增量方程，求整体的△xp

\# 叠加所有 landmark block 的增量方程，得到整体 LM 增量方程

for 每一个 landmark block 实例：

$$b = \sum -J_p^T \hat{Q}_2 \hat{Q}_2^T r$$

$$H = \sum J_p^T \hat{Q}_2 \hat{Q}_2^T J_p$$

end for

采用 PCG 求解器求解以下问题，得到所有相机 pose 的增量 Δx_p：

或者GN

$$H \Delta x_p = b$$

for 每一个 landmark block 实例：

将对应于当前 landmark 的相机 pose 的增量 Δx_p 代入如下公式，得到当前地标 position 的增量 Δx_l：

$$\Delta x_l = -\hat{R}_1^{-1} \left(\hat{Q}_1^T r + \hat{Q}_1^T J_p \Delta x_p \right)$$

由△xp*得到△xl*，
这个△x由△xp*得到△xl*l使得式(15)第一项为0

end for

\# 更新状态向量并计算新的误差

叠加所有 landmark block 计算所得增量，更新相机 pose 和地标 position 的节点存储数据，并为之前的做备份；

for 每一个 landmark block 实例：

基于当前相机 pose 和地标 position 计算误差 r；

end for

计算总误差与平均误差；

\# 衡量是否满足提前终止条件

if 平均误差足够小 or 平均误差改变量足够小 or 增量足够小 or 达到最大迭代次数：

终止迭代；

\# 评价本次迭代结果，并作对应选择

if 基于总误差和线性化的效果，认为此步可通过：

迭代失败的计数器清空计数；

for 每一个 landmark block 实例：迭代成功则构造新的A矩阵

计算观测误差对相机 pose 和地标 position 的雅可比矩阵 J_p 和 J_l；

对 J_l 进行 QR 分解，得到 Q_1、Q_2、R_1、R_2（R_2 为零矩阵）；

计算对应位置元素，构造 landmark block 矩阵如下：

$$A=\begin{bmatrix} Q_1^T J_p & R & Q_1^T r \\ Q_2^T J_p & 0 & Q_2^T r \end{bmatrix}$$

end for

else：

迭代失败的计数器自加一；迭代失败则回滚A

读取相机 pose 和地标 position 的节点的备份数据；

利用保存的吉文斯矩阵 Q_λ 对 landmark block 矩阵进行回退，并去除其中的 $\sqrt{\lambda}D$ 部分；（若对 A 已有保存，则可不用此步）

由于B=Qλ*A，则A=Qλ^T*B
（若A有backup则可以直接赋值回滚）

$$\hat{A}=Q_\lambda^T\begin{bmatrix} \hat{Q}_1^T J_p & \hat{R} & \hat{Q}_1^T r \\ \hat{Q}_2^T J_p & 0 & \hat{Q}_2^T r \end{bmatrix}=\begin{bmatrix} Q_1^T J_p & R & Q_1^T r \\ Q_2^T J_p & 0 & Q_2^T r \\ 0 & \sqrt{\lambda}D & 0 \end{bmatrix}$$

调整 LM 算法阻尼因子 λ 的数值；

end if

if 迭代连续失败次数 > 阈值：

求解失败，终止迭代；

end for

\# 完成 LM 迭代计算

核心问题就是求出JL的QR分解，加上阻尼因子，然后LM求解，回滚等。