Gradient\_problem.h

GradientProblem的例子代表了一般的非线性优化问题，要求必须知道目标函数的值和对应的梯度。Problem类只可用来构建非线性最小二乘问题，GradientProblem对目标函数的形式没有限制。

GradientProblem的结构由FirstOrderFunction和LocalParameterization组成，后者不是必须包含的。

FirstOrderFunction用来评估（evaluate）目标函数的代价cost和梯度gradient；

LocalParameterization在ambient空间和局部正切空间之间做选择，当没有提供LocalParameterization时，假定了正且空间与梯度向量所在的ambient欧式空间一致。

例子：

Rosenbrock方程：

还看不懂这个类写的啥

sized\_cost\_function.h

数据上固定的代价函数类，与数据动态固定的基类相比，减少了boilerplate。

kNumResiduals模板参数可以是2或者5,或ceres::DYNAMIC,若为最后一种，运行时子类需要调用set\_num\_residuals()函数。

Ceres运行报错：

估计residualblock时发生错误：

有两种可能的原因：1.CostFunction没有evaluate和填满所需要的residual及jacobian；2.计算jacobian时出现了无穷值，nan/infinite

Residual block大小1；参数block x1 residuals

对每个参数快，其值打印在第一列，jacobian值打印在对应的residual下。若参数块为常数，其jacobian打印为”not computed”；若jacobian/residual某列要求给出，但用户代码中没有写出，表示未初始化(uninitialized)，视为错误；residual和jacobian被evaluate为inf/nan也视为错误。

CostFunction

输入：

输入参数是指针数组，指向含有多个参数块的数组，参数的元素相同，都为parameter block\_sizes，参数块的顺序和parameter\_block\_size一致

输出：

Residuals，大小为num\_residuals的数组

Jacobians，是大小为parameter\_block\_size的数组，数组元素为指针，指向每个参数块的雅可比矩阵，雅可比的顺序与parameter\_block\_size相同，即第i个元素是num\_residuals\_\*parameter\_block\_size[i]，每个雅可比矩阵块都以row-major顺序存储，即：？

如果jacobians为空，则不返回导数，当只计算cost时是这样。如果jacobians[i]为空，则第i个参数的雅可比就不可返回。返回值表示residuals和/或jacobians的计算是否成功。

可用于交流数值failure