Alternating Direction Method (ADM)

代码实现了求解下面模型的交替方向乘子法（ADM）算法：

$$\min_A \{\tau\|A\|_1 + \|C-A\|_*\}$$

其中 C 是待分解的数据矩阵（data），将其分解为一个稀疏矩阵 A 和一个低秩矩阵 $C-A$。

其中，$|\cdot|*1$为L_1范数，$|\cdot|$*为核范数或称为矩阵秩。该函数主要解决了很多实际问题，如图像分割、压缩和恢复等。

输入参数包括：数据矩阵 C，小正数 τ，以及一个包含选项参数的结构体 opts。选项参数包括：初始矩阵 A_0、初始矩阵 B_0、初始矩阵 Λ_0、参数 β、误差容忍度 tol、最大迭代次数 maxit、是否记录迭代过程中**稀疏矩阵**的误差 errsSP、是否记录迭代过程中**低秩矩阵**的误差 errsLR、是否记录目标函数值 obj、是否记录残差 res。

算法的实现过程中，先对选项参数进行解析，并初始化矩阵 A、B、Λ。然后通过交替优化求解 A 和 B 两个子问题，利用更新的 A 和 B 更新 Λ。在每次迭代后，可以选择记录稀疏矩阵误差、低秩矩阵误差、目标函数值或残差，并在满足一定条件时停止迭代。最终输出分解得到的稀疏矩阵和低秩矩阵，以及迭代次数和停止原因等信息。

输入参数：

- C：待分解的矩阵；
- τ：小正数参数；
- opts：一个结构体变量，包含了一些选项参数。具体包括：
- beta：正则化参数，默认为0.25除以C的绝对值的平均值；
- tol：停止迭代的相对误差，默认值为1e-6；
- maxit：最大迭代次数，默认为1000；
- print：是否输出迭代信息，默认为0；
- A0、B0、Lam0：分别为A、B、Lambda的初始值；
- Sparse、LowRank、record_obj、record_res：这几个变量用来记录中间过程信息，分别表示待分解矩阵的稀疏部分、低秩部分、目标函数值和残差，缺省值均为0。

输出参数：

- out：一个结构体变量，包含了分解后的稀疏矩阵A、低秩矩阵B、迭代次数iter、停止原因exit等信息。

在迭代计算中，主要分为两个步骤：

- 更新A：通过计算梯度来更新A；
- 更新B：通过奇异值分解来更新B。

每个迭代步骤中，还需要判断是否满足停止条件，以及是否需要记录中间信息。

该算法首先对变量进行初始化，然后进行主要的迭代过程。主要的迭代过程分为三个步骤：

1. 求解稀疏矩阵 A 的子问题。这个子问题是一个 L1 正则化问题，可以使用一个硬阈值函数来求解。
2. 求解低秩矩阵 $C - A$ 的子问题。这个子问题可以使用一个矩阵奇异值分解 (SVD) 来求解。
3. 更新 ADMM 的拉格朗日乘子。

在主要的迭代过程中，还会记录并输出误差和目标函数值，以便于调试和性能评估。最后，该算法将稀疏矩阵和低秩矩阵作为输出。