```m
function out = ADM(C,tau,opts)
%% ADM 基于交替方向乘子法（ADMM）的矩阵分解函数
%% parameter setting
beta = .25/mean(abs(C(:))); %作为一个参数beta。这个参数后面会用于计算迭代步长。这个过程相当于对C进行了归一化。
tol = 1.e-6; %这个变量定义了一个迭代终止的容忍误差，当每次迭代中变化的值小于这个容差时，迭代将停止。
maxit = 1000;%这个变量定义了最大迭代次数，如果达到了这个次数还没有收敛，迭代也会停止。
print = 0; %print变量默认值为0，如果用户输入了print字段，则print变量将被更新为opts.print的值。
if isfield(opts,'beta'); beta = opts.beta; end %检查输入参数中是否包含beta、tol、maxit和print的字段，并将它们的值赋给相应的变量。
if isfield(opts,'tol'); tol = opts.tol; end%opts是一个包含所有输入参数的结构体，可以使用isfield函数来检查是否包含特定的字段
if isfield(opts,'maxit'); maxit = opts.maxit; end
if isfield(opts,'print'); print = opts.print; end

%% initialization
[m,n] = size(C); %size() 函数获取 C 的行数和列数
A = zeros(m,n); %然后初始化三个矩阵 A、B 和 Lambda，并将它们全部初始化为 0。
B = zeros(m,n);
Lambda = zeros(m,n);
if isfield(opts,'A0');  A = opts.A0; end %通过 isfield()函数判断是否传入了初始值 A0、B0 和 Lam0，如果有，则将其赋值给对应的矩阵，否则保持为 0。
if isfield(opts,'B0');  B = opts.B0; end
if isfield(opts,'Lam0'); Lambda = opts.Lam0; end

%% keep record 为了设置记录输出的参数。 如果在输入的opts结构体中指定了这些参数，就会将这些参数对应的值设为1
RECORD_ERRSP = 0; RECORD_ERRLR = 0; RECORD_OBJ = 0; RECORD_RES = 0;
if isfield(opts,'Sparse'); SP = opts.Sparse; nrmSP = norm(SP,'fro'); out.errsSP = 1; RECORD_ERRSP = 1; end % opts有Sparse记录稀疏矩阵的误差，将RECORD_ERRSP设为1，表示记录误差;且稀疏矩阵的Frobenius范数nrmSP，并将out.errsSP初始化为1，稀疏矩阵的误差。
if isfield(opts,'LowRank'); LR = opts.LowRank; nrmLR = norm(LR,'fro'); out.errsLR = 1; RECORD_ERRLR = 1; end%有LowRank这个字段，说明需要记录低秩矩阵的误差，那么就将RECORD_ERRLR设为1，需要记录误差；低秩Frobenius范数nrmLR，并将out.errsLR初始化为1。低秩的误差。
if isfield(opts,'record_obj'); RECORD_OBJ = 1; out.obj = []; end %opts结构体中有record_obj这个字段，说明需要记录每次迭代的目标函数值，那么将RECORD_OBJ设为1，并将out.obj初始化为空向量。
if isfield(opts,'record_res'); RECORD_RES = 1; out.res = []; end%如果opts结构体中有record_res这个字段，说明需要记录每次迭代的残差，那么将RECORD_RES设为1，并将out.res初始化为空向量。


% main
for iter = 1:maxit

    nrmAB = norm([A,B],'fro'); % % 计算矩阵 A 和 B 的 Frobenius 范数，Frobenius 范数是矩阵所有元素的平方和的平方根

    %% A - subproblem
    X = Lambda / beta + C; %计算子问题中用到的矩阵$X$，其中$\Lambda$是已知矩阵，$\beta$是参数，$C$是已知矩阵。
    Y = X - B; %计算子问题中用到的矩阵Y其中X是已知矩阵，B是未知矩阵。
    dA = A;%记录未更新前的矩阵$A$。
    A = sign(Y) .* max(abs(Y) - tau/beta, 0); %更新矩阵$A$，其中$\tau$是参数。这个式子可以看成是对$Y$做软阈值。
    dA = A - dA; %记录矩阵$A$更新后的变化量。
```

```
41
42      %% B - subprolbme
43      Y = X - A; %重新计算子问题中用到的矩阵$Y$。
44      dB = B; %保存B的当前值以便于计算改变
45      [U,D,VT] = svd(Y); % 对Y进行奇异值分解
46      D = diag(D);% 把对角线元素保存为列向量
47      ind = find(D > 1/beta);% 找到大于阈值的元素
48      D = diag(D(ind) - 1/beta);   % 对大于阈值的元素进行修正
49      B = U(:,ind) * D * VT(ind,:);% 更新B的值
50      dB = B - dB;% 计算B的改变量
51
52      %% keep record
53      %记录一些输出结果、计算相对变化，判断算法是否收敛，以及更新拉格朗日乘子Lambda。
54      %% 如果 RECORD_ERRSP 为真，计算当前 A 和预设的稀疏矩阵 SP 的 Frobenius 范数之间的相对误差，并将误差添加到 out.errsSP 数组中。
55      %% 如果 RECORD_ERRLR 为真，计算当前 B 和预设的低秩矩阵 LR 的 Frobenius 范数之间的相对误差，并将误差添加到 out.errsLR 数组中。
56      %% 如果 RECORD_OBJ 为真，计算当前目标函数的值，并将值添加到 out.obj 数组中。
57      %% 如果 RECORD_RES 为真，计算当前重构误差并将其添加到 out.res 数组中。
58      %% 计算相对变化 RelChg，并检查它是否小于预设的容差 tol。如果小于，跳出迭代循环。
59      %% 如果 print 为真，打印当前迭代轮数、相对变化等信息。
60      %% 更新拉格朗日乘子 Lambda。
61      if RECORD_ERRSP; errSP = norm(A - SP,'fro') / (1 + nrmSP); out.errsSP = [out.errsSP; errSP]; end
62      if RECORD_ERRLR; errLR = norm(B - LR,'fro') / (1 + nrmLR); out.errsLR = [out.errsLR; errLR]; end
63      if RECORD_OBJ;   obj = tau*norm(A(:),1) + sum(diag(D));    out.obj = [out.obj; obj];         end
64      if RECORD_RES;   res = norm(A + B - C, 'fro');             out.res = [out.res; res];         end
65
66      %% stopping criterion
67      RelChg = norm([dA,dB],'fro') / (1 + nrmAB);
68      if print, fprintf('Iter %d, RelChg %4.2e',iter,RelChg); end
69      if print && RECORD_ERRSP && RECORD_ERRLR, fprintf(', errSP %4.2e, errLR %4.2e',errSP,errLR); end
70      if print, fprintf('\n'); end
71      if RelChg < tol, break; end
72
73      %% Update Lambda
74      Lambda = Lambda - beta * (A + B - C);
75  end
76
77
78  % output
79  out.Sparse = A;
80  out.LowRank = B;
81  out.iter = iter;
82  out.exit = 'Stopped by RelChg < tol';
83  if iter == maxit, out.exit = 'Maximum iteration reached'; end
84  end
```