第二题

matlab代码：

```matlab
clear;clc;

m = 100;n = 500;s = 50;
A = randn(m,n);
xs = zeros(n,1);picks=randperm(n);xs(picks(1:s)) = randn(s,1);
b = A*xs;tau=0.001;

cvx_begin
    variable x(n)
    minimize 1/2*square_pos(norm(A*x-b))+tau*norm(x,1)
cvx_end

f = @(x) 1/2*square_pos(norm(A*x-b))+tau*norm(x,1);

% constant step size
x = zeros(n,1); alpha = 0.0001;
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_const_step_size_1(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-alpha*subgd;
end

x = zeros(n,1); alpha = 0.0005;
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_const_step_size_2(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-alpha*subgd;
end

x = zeros(n,1); alpha = 0.001;
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_const_step_size_3(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-alpha*subgd;
end

% constant step length
x = zeros(n,1); gamma = 0.0005;
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_const_step_len_1(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-gamma/norm(subgd)*subgd;
end

x = zeros(n,1); gamma = 0.001;
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_const_step_len_2(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-gamma/norm(subgd)*subgd;
end

x = zeros(n,1); gamma = 0.005;
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_const_step_len_3(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-gamma/norm(subgd)*subgd;
end

% diminishing step size
x = zeros(n,1);
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_dimi_step_size_1(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-0.0005/sqrt(k)*subgd;
end

x = zeros(n,1);
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_dimi_step_size_2(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-0.001/sqrt(k)*subgd;
end

x = zeros(n,1);
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_dimi_step_size_3(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-0.001/k*subgd;
end

% Polyak's step size
```

```
x = zeros(n,1);
for k = 1 : 10000
    funcv = f(x);
    funcv_diff_poly_step_size(k) = funcv-cvx_optval;
    % 计算在x点处的一个次梯度
    subgd = A'*(A*x-b)+tau*sign(x);
    % 更新x
    x = x-(funcv-cvx_optval)/sum_square(subgd)*subgd;
end

figure;
semilogy(funcv_diff_const_step_size_1, 'DisplayName', 'alpha(k)=0.0001', 'LineWidth', 1.5);
hold on
semilogy(funcv_diff_const_step_size_2, 'DisplayName', 'alpha(k)=0.0005', 'LineWidth', 1.5);
semilogy(funcv_diff_const_step_size_3, 'DisplayName', 'alpha(k)=0.001', 'LineWidth', 1.5);
xlabel('Iteration');
ylabel('f(xk)-f*');
legend;
hold off;

figure;
semilogy(funcv_diff_const_step_len_1, 'DisplayName', 'alpha(k)=0.0005/norm(subgrad)', 'LineWidth', 1.5);
hold on;
semilogy(funcv_diff_const_step_len_2, 'DisplayName', 'alpha(k)=0.001/norm(subgrad)', 'LineWidth', 1.5);
semilogy(funcv_diff_const_step_len_3, 'DisplayName', 'alpha(k)=0.005/norm(subgrad)', 'LineWidth', 1.5);
xlabel('Iteration');
ylabel('f(xk)-f*');
legend;
hold off;

figure;
semilogy(funcv_diff_dimi_step_size_1, 'DisplayName', 'alpha(k)=0.0005/sqrt(k)', 'LineWidth', 1.5);
hold on;
semilogy(funcv_diff_dimi_step_size_2, 'DisplayName', 'alpha(k)=0.001/sqrt(k)', 'LineWidth', 1.5);
semilogy(funcv_diff_dimi_step_size_3, 'DisplayName', 'alpha(k)=0.001/k', 'LineWidth', 1.5);
xlabel('Iteration');
ylabel('f(xk)-f*');
legend;
hold off;

figure;
semilogy(funcv_diff_poly_step_size, 'DisplayName', 'Polyak', 'LineWidth', 1.5);
xlabel('Iteration');
ylabel('f(xk)-f*');
legend;
```