
```
function [f, g, H] = objective_fun(x, A)
% The objective function from Boyd and Vandenberghe p. 519.
% Returns the function, gradient, and Hessian at a point x.
%
% Author: Terrence Alsup
% Date: March 10, 2020
% File: objective_fun.m

[n, m] = size(A); % Get the dimensions of the problem.

% Check if x is in the domain.
atx = A'*x;
for i=1:m
    % Check if the point is outside the domain.
    if atx(i) >= 1 || abs(x(i)) >= 1
        f = Inf;
        g = nan*ones(n,1);
        %H = nan*ones(n,n);
        return;
    end
end

% If we have reached this point in the program, then x is in the
domain.

f = -sum(log(1 - atx)) - sum(log(1 - x.^2)); % The objective function.

% Compute the gradient.
g = zeros(n,1);
for j=1:n
    % We take the transpose on A(j,:) to change it from a row to a
column
    % vector, making the dimensions match.
    g(j) = sum(A(j,:)'./(1 - atx)) + 2*x(j)/(1 - x(j)^2);
end

% Compute the Hessian.
H = zeros(n, n);
for j=1:n
    for k=1:n
        H(j,k) = sum(A(j,:)'.*A(k,:)'./(1 - atx).^2);
        if j == k
            % Add the extra diagonal term.
            H(j,k) = H(j,k) + 2*(1 + x(j)^2)/((1 - x(j)^2)^2);
        end
    end
end
end
end
```

Published with MATLAB® R2019b