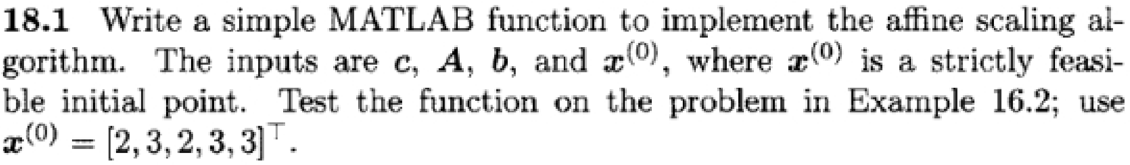
전기전자공학수학

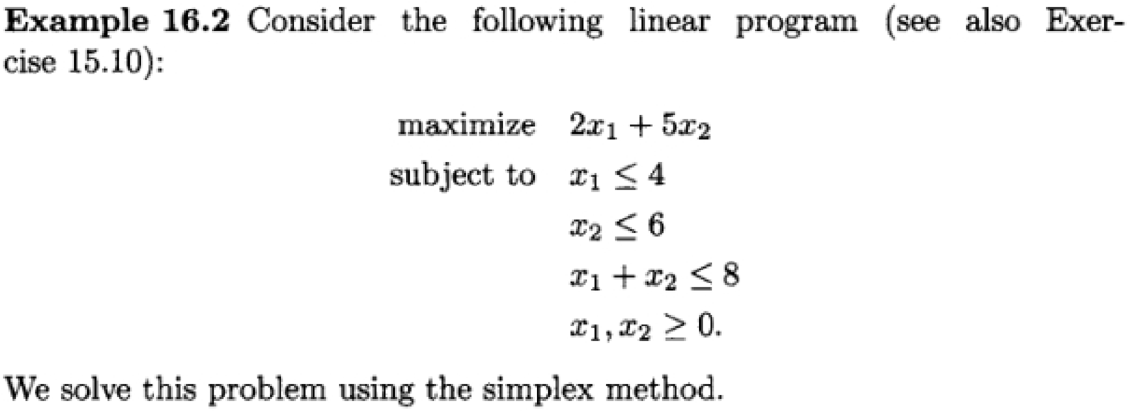
Computer Simulation HW4

2018142023 조성민



18.1 Write a simple MATLAB function to implement the affine scaling algorithm. The inputs are $\boldsymbol{c}$, $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{x}^{(0)}$, where $\boldsymbol{x}^{(0)}$ is a strictly feasible initial point. Test the function on the problem in Example 16.2; use $\boldsymbol{x}^{(0)} = [2, 3, 2, 3, 3]^\top$.

※참고※



Example 16.2 Consider the following linear program (see also Exercise 15.10):

$$\begin{array}{rl} \text{maximize} & 2x_1 + 5x_2 \\ \text{subject to} & x_1 \le 4 \\ & x_2 \le 6 \\ & x_1 + x_2 \le 8 \\ & x_1, x_2 \ge 0. \end{array}$$

We solve this problem using the simplex method.

<Theory>

Affine Scaling algorithm

-mininize $c^T x$

subject to $Ax = b$

$x \geq 0$

-Suppose that a feasible point $x^{(0)}$ is strictly interior

-$x^{(1)}$ by searching in direction $d^{(0)} : x^{(1)} = x^{(0)} + \alpha_0 d^{(0)}$

-Negative gradient: $d^{(0)} = -c$

-$Ax^{(0)} = b$ $Ax^{(1)} = b$ → $A(x^{(1)} - x^{(0)}) = \alpha_0 A d^{(0)} = 0$

-Orthogonal projection of the negative gradient to the null space of A

$d^{(0)} = -Pc, P = I_n - A^T (AA^T)^{-1} A$

<Implementation>

maximize $2x_1 + 5x_2$

subject to $x_1 \leq 4$

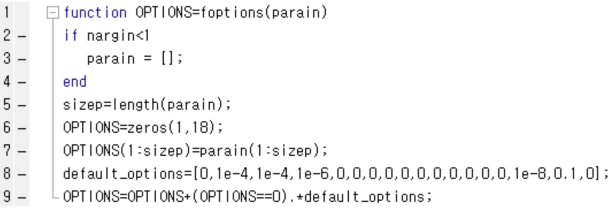
$x_2 \leq 6$

$x_1 + x_2 \leq 8$

$x_1, x_2 \geq 0$

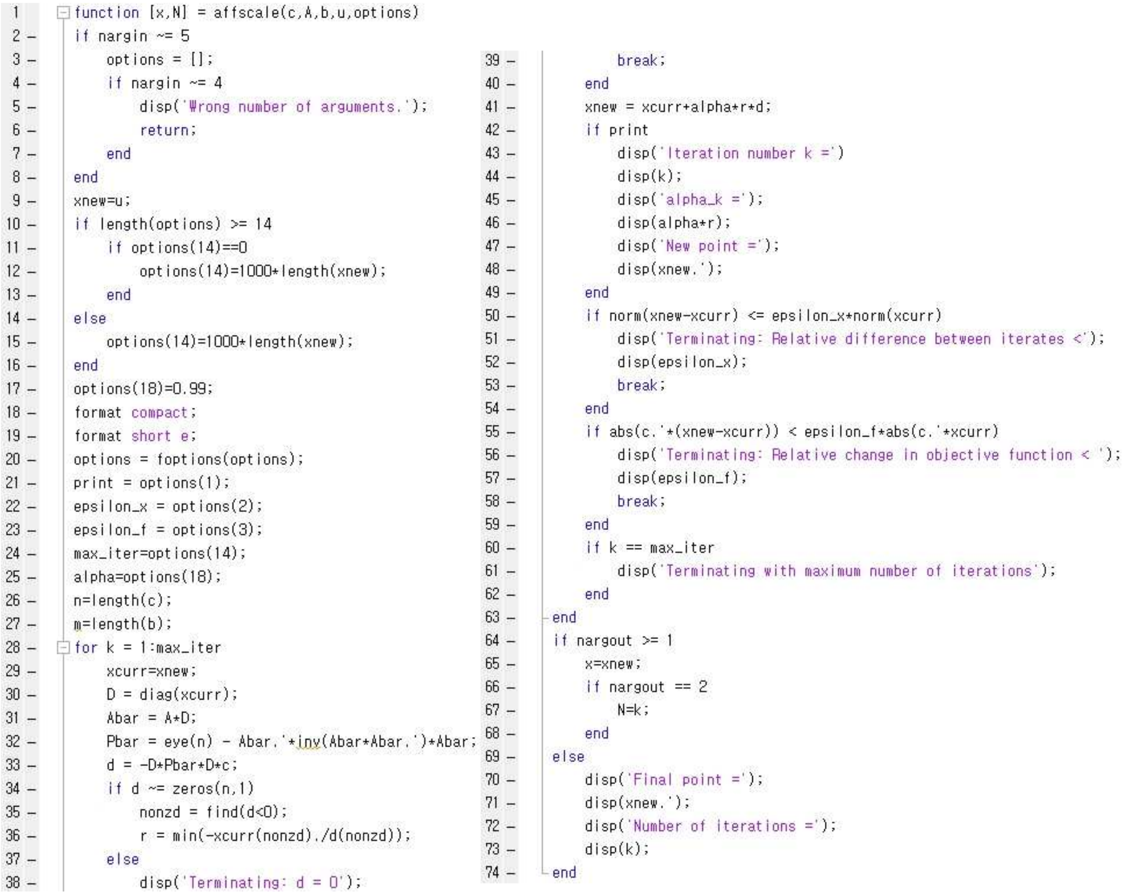
<MATLAB code>

1. foptions



```
function OPTIONS=foptions(parain)
if nargin<1
   parain = [];
end
sizep=length(parain);
OPTIONS=zeros(1,18);
OPTIONS(1:sizep)=parain(1:sizep);
default_options=[0,1e-4,1e-4,1e-6,0,0,0,0,0,0,0,0,0,0,0,0,1e-8,0.1,0];
OPTIONS=OPTIONS+(OPTIONS==0).*default_options;
```

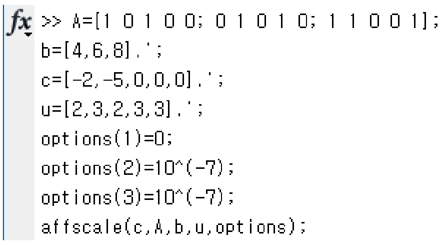
2. affscale



```
function [x,N] = affscale(c,A,b,u,options)
if nargin ~= 5
    options = [];
    if nargin ~= 4
        disp('Wrong number of arguments.');
        return;
    end
end
xnew=u;
if length(options) >= 14
    if options(14)==0
        options(14)=1000*length(xnew);
    end
else
    options(14)=1000*length(xnew);
end
options(18)=0.99;
format compact;
format short e;
options = foptions(options);
print = options(1);
epsilon_x = options(2);
epsilon_f = options(3);
max_iter=options(14);
alpha=options(18);
n=length(c);
m=length(b);
for k = 1:max_iter
    xcurr=xnew;
    D = diag(xcurr);
    Abar = A*D;
    Pbar = eye(n) - Abar.'*inv(Abar*Abar.')*Abar;
    d = -D*Pbar*D*c;
    if d ~= zeros(n,1)
        nonzd = find(d<0);
        r = min(-xcurr(nonzd)./d(nonzd));
    else
        disp('Terminating: d = 0');
        break;
    end
    xnew = xcurr+alpha*r*d;
    if print
        disp('Iteration number k =')
        disp(k);
        disp('alpha_k =');
        disp(alpha*r);
        disp('New point =');
        disp(xnew.');
    end
    if norm(xnew-xcurr) <= epsilon_x*norm(xcurr)
        disp('Terminating: Relative difference between iterates <');
        disp(epsilon_x);
        break;
    end
    if abs(c.'*(xnew-xcurr)) < epsilon_f*abs(c.'*xcurr)
        disp('Terminating: Relative change in objective function < ');
        disp(epsilon_f);
        break;
    end
    if k == max_iter
        disp('Terminating with maximum number of iterations');
    end
end
if nargout >= 1
    x=xnew;
    if nargout == 2
        N=k;
    end
else
    disp('Final point =');
    disp(xnew.');
    disp('Number of iterations =');
    disp(k);
end
```

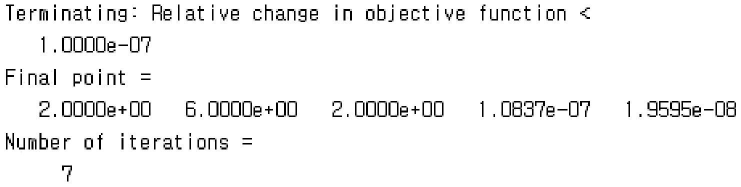
<Result & Analysis>

-입력



```
>> A=[1 0 1 0 0; 0 1 0 1 0; 1 1 0 0 1];
b=[4,6,8].';
c=[-2,-5,0,0,0].';
u=[2,3,2,3,3].';
options(1)=0;
options(2)=10^(-7);
options(3)=10^(-7);
affscale(c,A,b,u,options);
```

-출력



```
Terminating: Relative change in objective function <
   1.0000e-07
Final point =
   2.0000e+00   6.0000e+00   2.0000e+00   1.0837e-07   1.9595e-08
Number of iterations =
     7
```