Inequality constrained minimization: log-barrier method

We wish to solve

$$\min c^T x \qquad \text{subject to} \qquad Ax \leq b$$

with $n = 50$ and $m = 100$.

We use the barrier method with logarithmic barrier function

$$\varphi(x) = -\sum_{i=1}^{m} \log(-(a_i^T x - b_i))$$

and solve a sequence of smooth unconstrained problems

$$x^*(t) = \text{argmin}_{x \in \mathbb{R}^n} \quad tc^T x + \phi(x)$$

Objective functional augmented with the log-barrier

```
function [f,g,H] = objective_barrier(t,x,A,b,c);

[m,n] = size(A);
d = A*x - b; D = diag(1./d);

f = t*c'*x - log(-d')*ones(m,1);
g = t*c - A'*D*ones(m,1);
H = A'*D^2*A;
```

Problem parameters

```
m = 100; n = 50;
ALPHA = .1; BETA = .7;
mu = 50;

A = randn(m,n);
b = 1 + abs(randn(m,1));
c = randn(n,1);
```

Smooth unconstrained minimization

- Start with strictly feasible point $x = 0$
- Terminates when $t = 10^8$ (Duality gap 10^{-6})
- Centering uses Newton method with backtracking

```
x = zeros(n,1);
t = 1;
histobj=[];

NTTOL = 1e-10;     % stop inner iteration if lambda^2/2 < NTTOL
MAXITERS = 500;

while (t <= 1e8), % Outer loop

  niter = 0;

   for k=1:MAXITERS,  % Inner loop

     [val,g,H] = objective_barrier(t,x,A,b,c);

     v = -H\g;         % Newton step
     lambda = g'*v;    % Newton decrement

     % Perform backtracking line search along search direction
     s = 1;
     while (min([b - A*(x+s*v) ]) < 0),
       s = BETA*s;
     end  % first get feasible point  ... then search minimum

     while (objective_barrier_val(t,x+s*v,A,b,c) > val + ALPHA*s*lambda),
       s = BETA*s;
     end

     x = x+s*v;
     niter = niter + 1;

     % Test if optimum achieved
     if (abs(lambda/2) < NTTOL),   % decrement smaller than NTTOL?
        break;
     end

   end

    % Display progress
    obj = c'*x;
    histobj=[[histobj],[obj;niter;m/t]];  % Bookkeeeping
    disp(['obj: ',num2str(obj,'%1.6e'),'; PDGap: ', num2str(m/t,'%1.2e'), ';
               number iterations: ',int2str(niter)]);

   % Update
    t = mu*t;
end
```

Plot results (more Matlab commands to produce the nice plot below)

```
PDGap = histobj(3,:); niter = histobj(2,:);
total_iter = cumsum(niter);
figure; semilogy(total_iter,PDgap,'*');
```

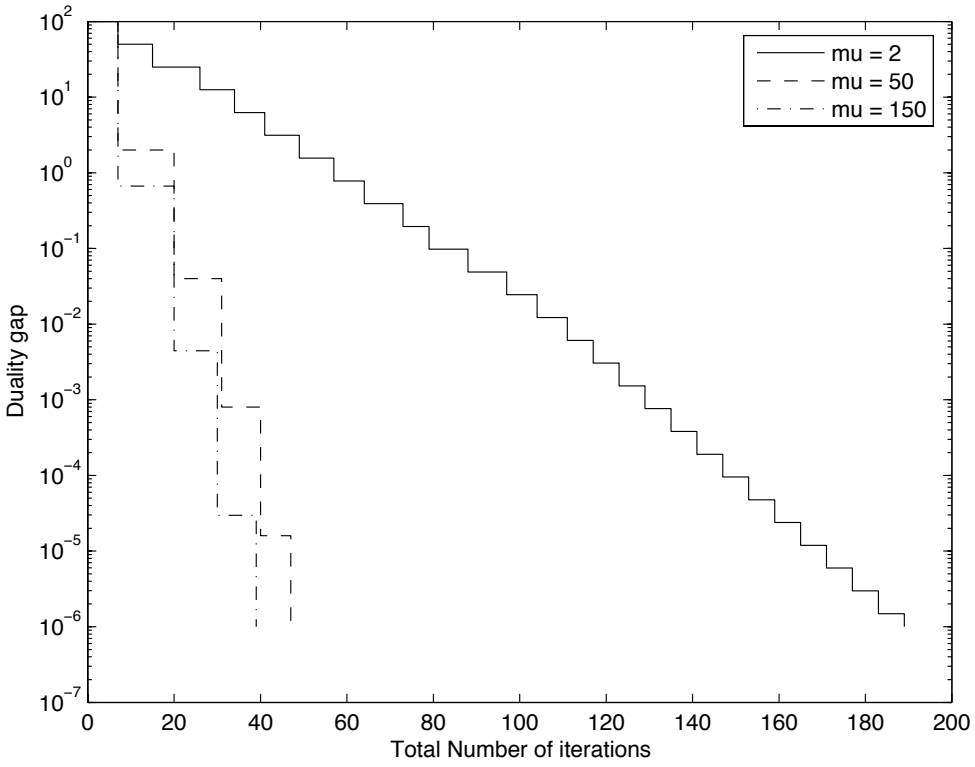


Figure 1: Plot of duality gap vs. total number of Newton iterations for $\mu = 2, 50, 150$

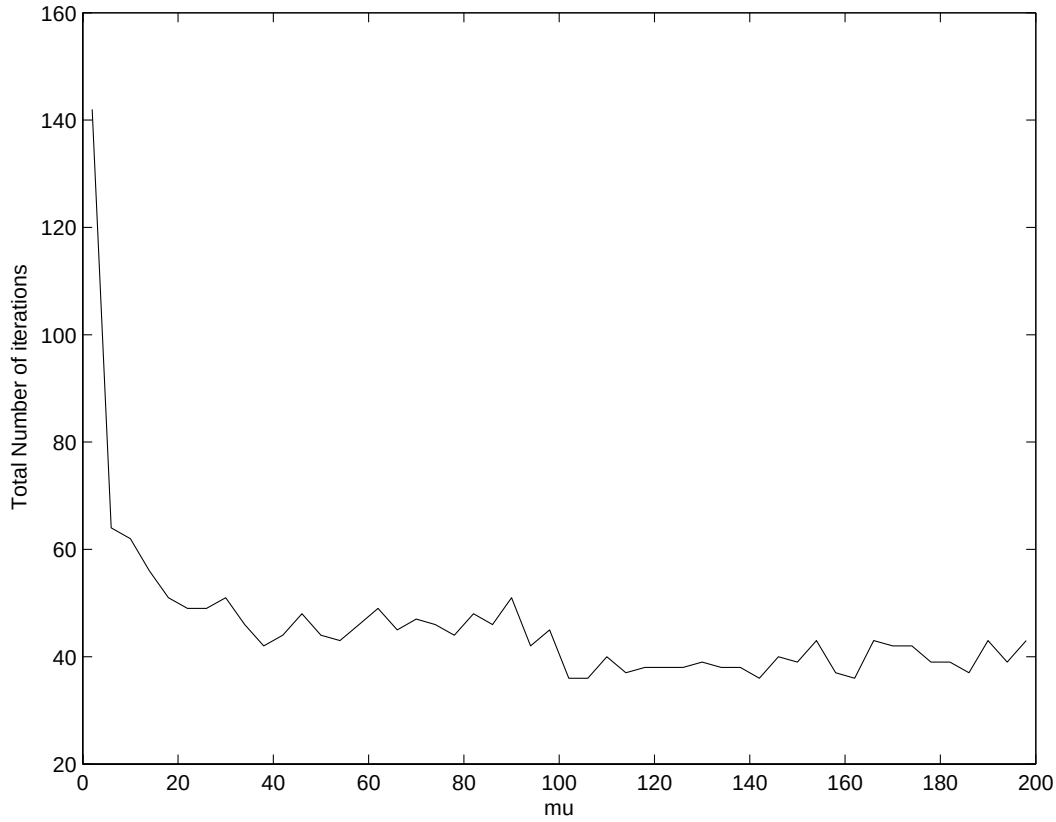


Figure 2: Trade-off between μ and the total number of Newton iterations needed to reduce the duality gap from 100 to 10^{-4}. The optimization problem is a 'moderately' small inequality constrained LP, just as before. This shows that the method is not very sensitive to the value of μ provided $\mu \geq 10$

The following figures are taken from our textbook (Boyd and Vandenberghe).

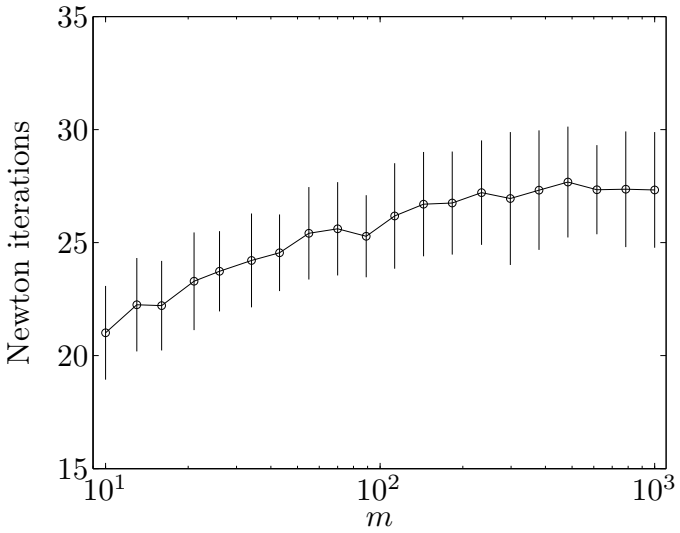


Figure 11.8 Average number of Newton steps required to solve 100 randomly generated LPs of different dimensions, with $n = 2m$. Error bars show standard deviation, around the average value, for each value of m. The growth in the number of Newton steps required, as the problem dimensions range over a 100 : 1 ratio, is very small.

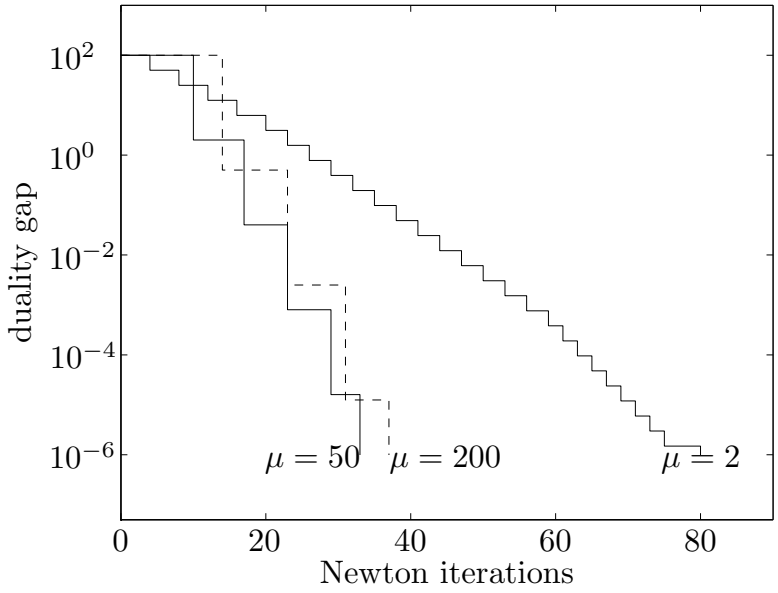


Figure 11.15 Progress of barrier method for an SOCP, showing duality gap versus cumulative number of Newton steps.

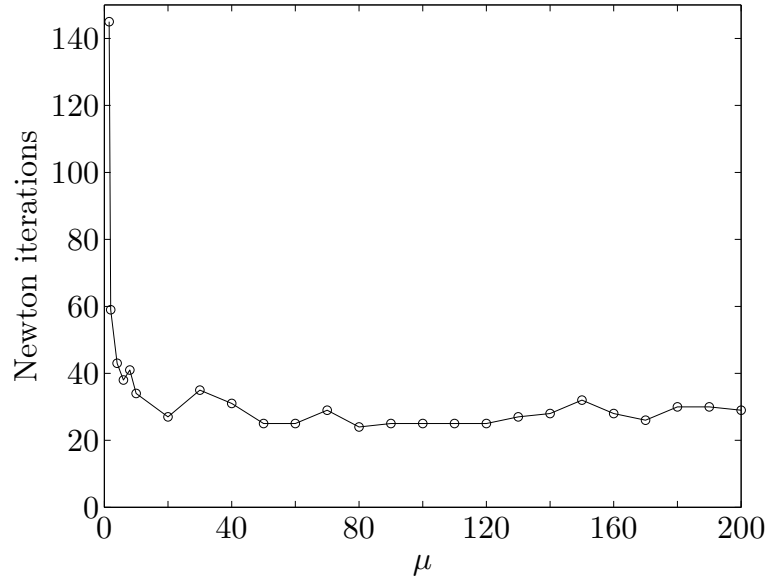


Figure 11.16 Trade-off in the choice of the parameter μ, for a small SOCP. The vertical axis shows the total number of Newton steps required to reduce the duality gap from 100 to 10^{-3}, and the horizontal axis shows μ.