Newton method example

We wish to minimize

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

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

Function to minimize

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

[m,n] = size(A);
d = A*x + b;

f = c'*x - log(d')*ones(m,1);

D = diag(1./d);
g = c - A'*D*ones(m,1);
H = A'*D^2*A;
```

Newton method

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

A = randn(m,n);
b = 5*abs(randn(m,1));
c = randn(n,1);

x0 = zeros(n,1);

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

% Main loop
histobj=[];x=x0;
disp('Begin main loop...');

for k=1:MAXITERS

   % Evaluate function
   [val,g,H] =my_objective(x,A,b,c);

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

```
   histobj=[histobj,val]; % Record the function value at current point.

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

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

   % Display progress
   disp(['lambda: ',num2str(lambda,'%1.4e'),
            'step: ',num2str(s,'%1.4e'),'  obj: ',num2str(val,'%1.6e')]);
end;
```

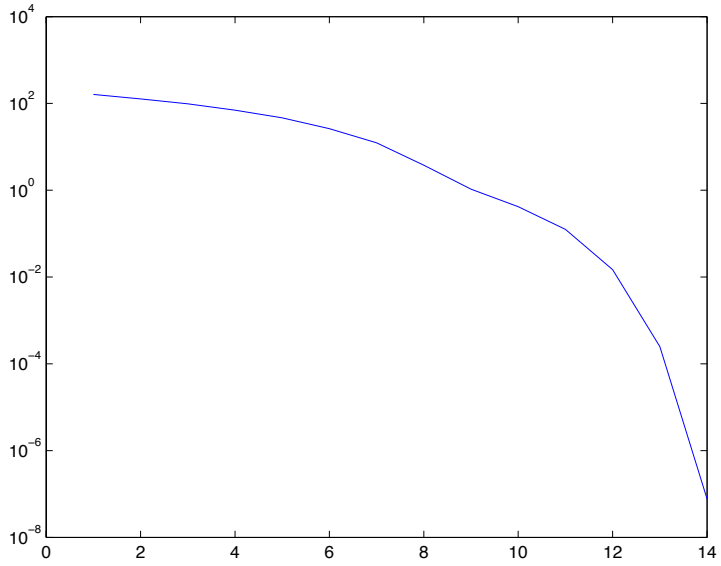


Figure 1: Log plot of $f(x^{(k)}) - p^*$ as a function of k