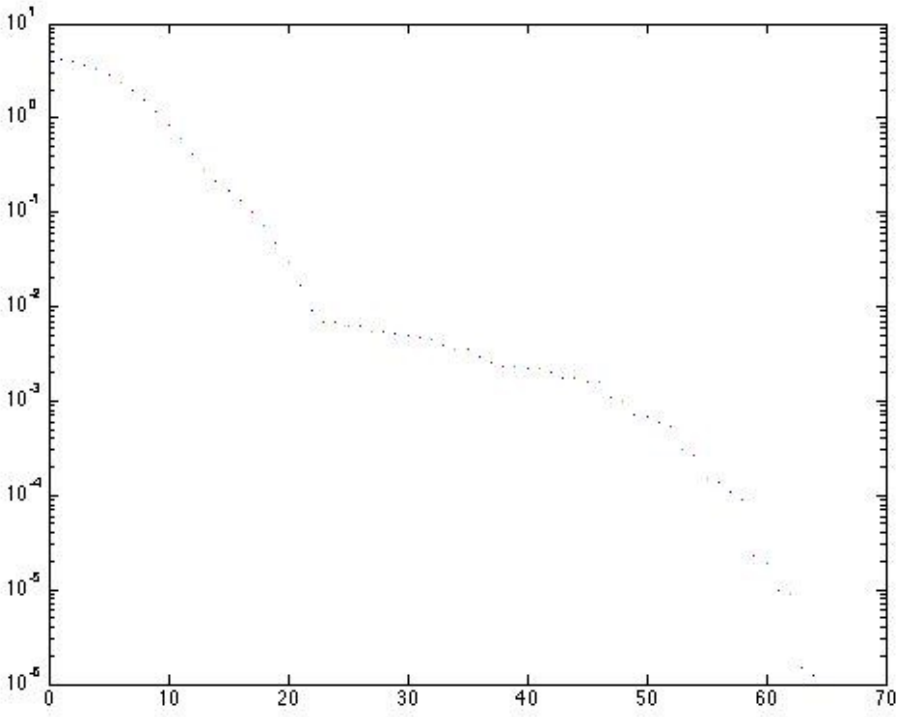
Image Deblurring - Matlab Sections

Project Abstract: A digital imaging system, like a DSLR camera, consists of recording light intensity from an object in a 2D array of picture elements called pixels. Some blurring always arises in the recording of a digital image. This could be due to several factors, such as an out of focus optical system which results in information smeared across the pixels. In this project, we will consider the problem of image deblurring; that is, given a blurred image and a mathematical model of the blurring process, we seek to reconstruct the original sharp scene. In particular, if we use a linear blurring model, then we are solving a so called linear inverse problem. The term inverse is used because we use the observed information (the blurred image) to determine the original scene information.

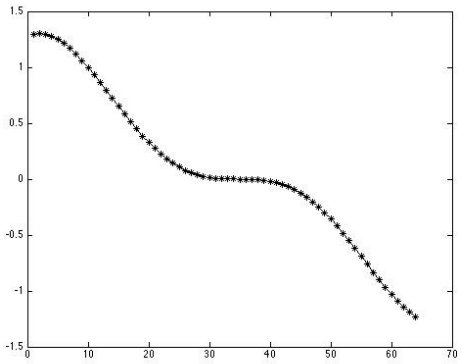
Problem 1 (a) - Plot the singular values of A (nxn image matrix)
A is a sparse matrix so we convert into full representation before using SVD

```
[U S V] = svd(full(A));
format long
diag(S,0)
semilogy(S)
```



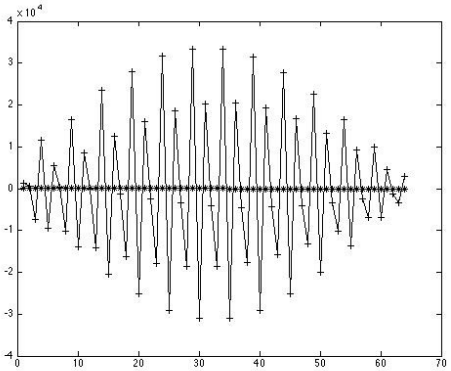
Problem 1(b)

```
x = A\b
n = length(xexact)
plot(1:n, x, 'k-+', 1:n, xexact, 'k-*')
```

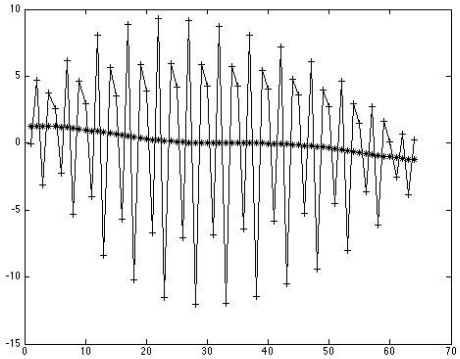


Problem 1(c)

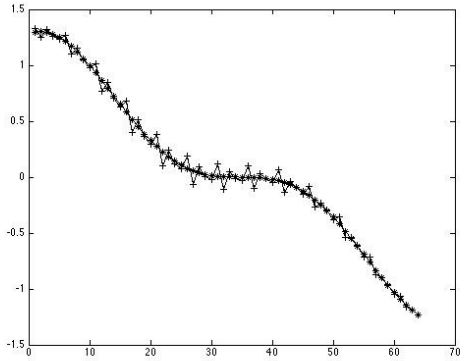
```
k = 10;
e = 10^(-k)*randn(size(x),1);
bnew = b + e;
plot(1:n, b, 'k-+', 1:n, bnew, 'k-*')
x = A\bnew
plot(1:n, x, 'k-+', 1:n, xexact, 'k-*')
```



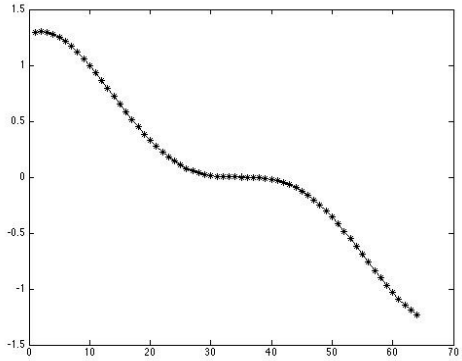
k=1



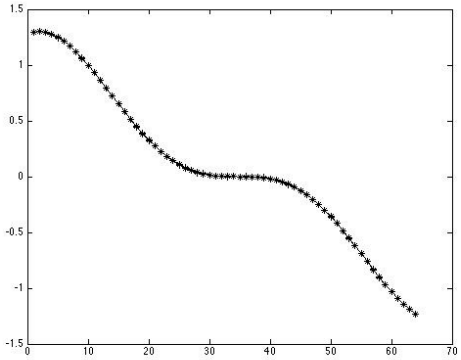
k=4



k=6



k=8

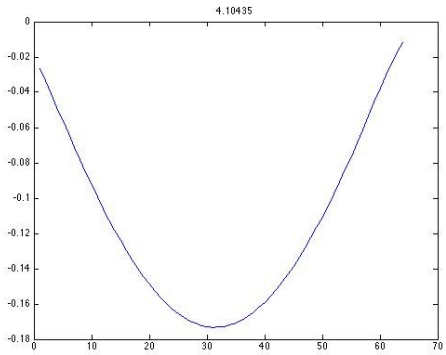


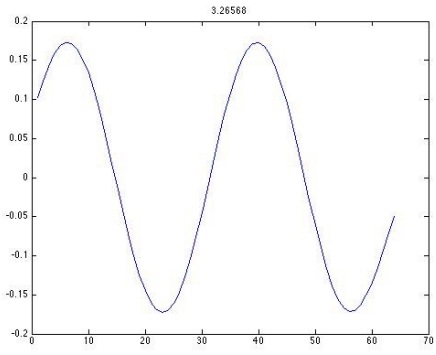
k=10

Problem 2(b)

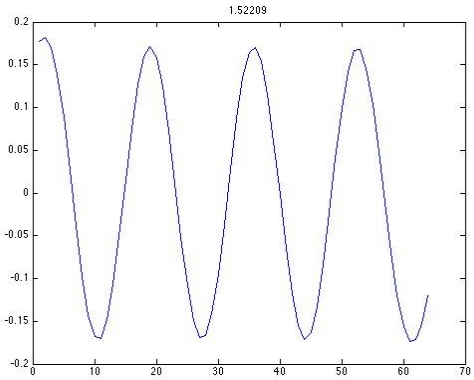
```
[U S V] = svd(full(A));
singVec = diag(S,0);

v1 = V(:,1);
sig1 = singVec(1);
plot(1:n, v1)
title(sig1)
```

```
v4 = V(:,4);
sig4 = singVec(4);
plot(1:n, v4)
title(sig4)
```

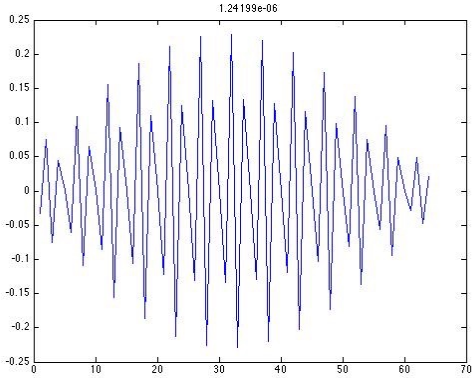




```
v8 = V(:,8);
sig8 = singVec(8);
plot(1:n, v8)
title(sig8)
```

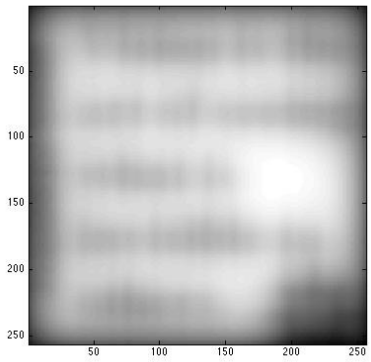



```
sig64 = singVec(64);
plot(1:n, v64)
title(sig64)
```

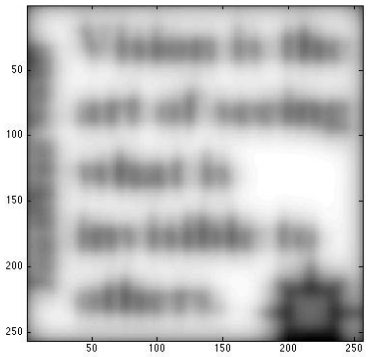



```
v64 = V(:,64);
```

Problem 6

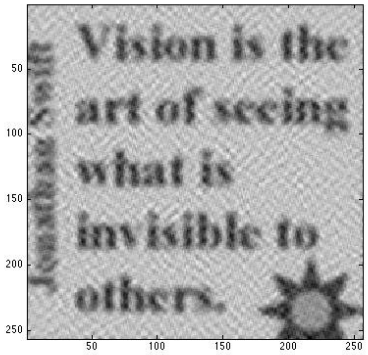


```
imagesc(B);
axis image;
colormap gray;

[Uc, Sc, Vc] = svd(C);
[Ud, Sd, Vd] = svd(D);
alpha = .005;

%% X = Vd*(Sd + alpha*eye(256))^(-2)*Sd*transpose(Ud)*B*Uc*Sc*(Sc +
alpha*eye(256))^(-2)*transpose(Vc)

x = Vd*((Sd + alpha*eye(256))\((Sd + alpha*eye(256))\
(Sd*transpose(Ud)*B*Uc*Sc*((Sc + alpha*eye(256))\(((Sc + alpha*eye(256))
\transpose(Vc)))))));
imagesc(x);
axis image;
colormap gray;
```

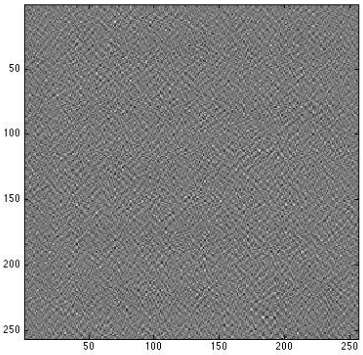


alpha=.1

alpha=.01



alpha=.005



alpha=.001