Digital Signal Processing II
Homework 7

Name: Amitesh Kumar Sah
NYU ID: N19714360

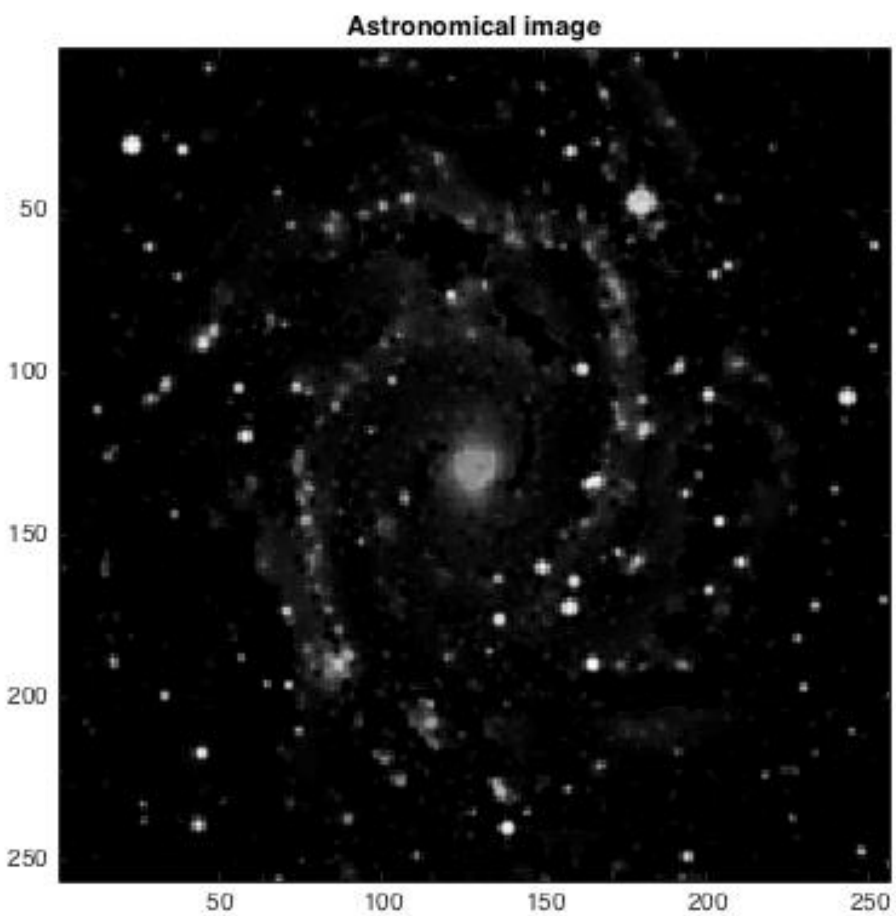
Question 1
1) Least square one-dimensional deconvolution with non-negativity constraints.

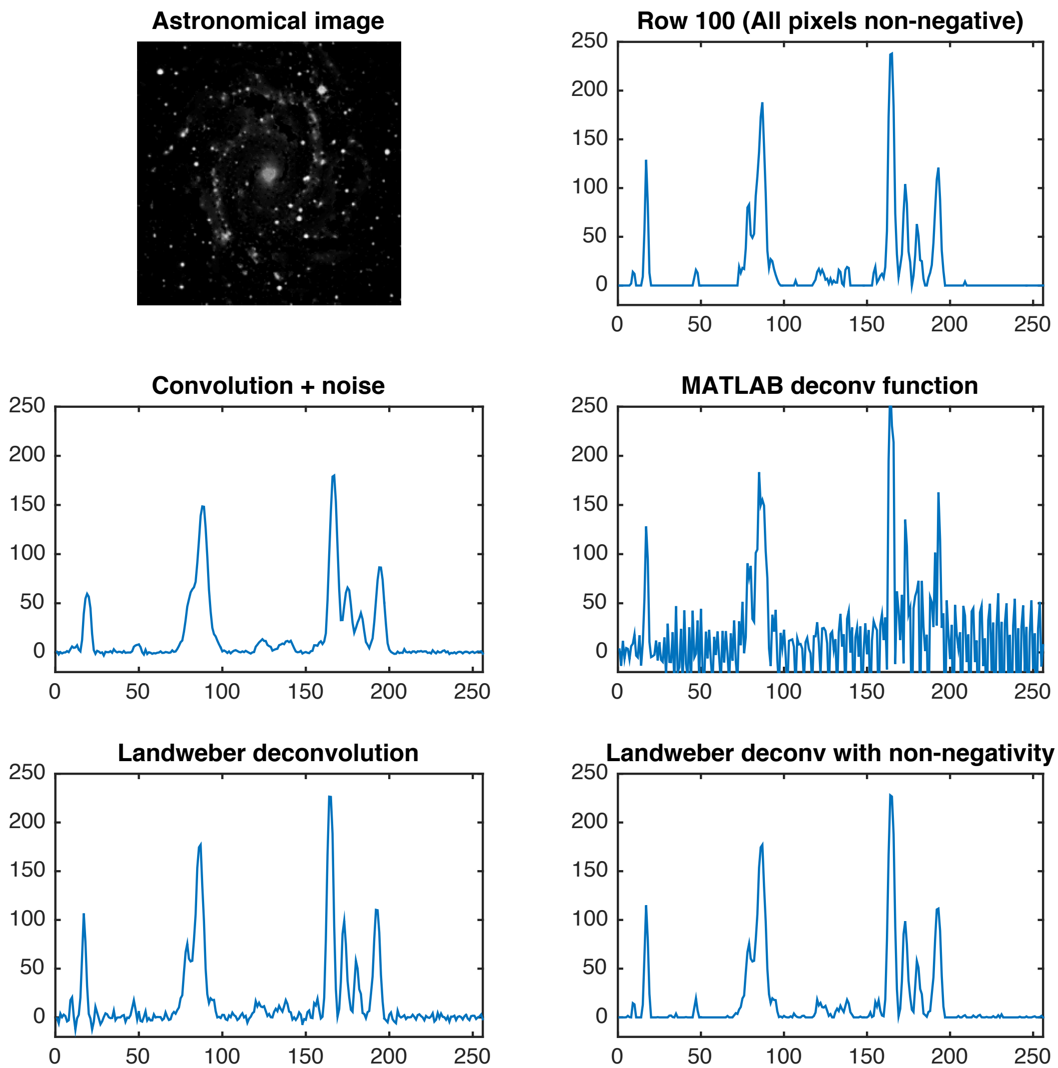
```
clear
v = imread('ngc_atrou.gif');   % Astronomical image from Jean Luc Starck
v = double(v);                 % Convert from 8-bit integer to floating point

figure(1), clf
imagesc(v)
axis image
colormap(gray)

x = v(190, :);
N = length(x);

M = 5;
h = ones(1, M)/M;

y = conv(h, x);
y = y + 1.0 * randn(size(y));

y=y';
h=h';
x_deconv = deconv(y, h);

lam = 0.05;
Nit = 50;
[x_deconv_LW, J1] = deconv_landweber1(y, h, lam, Nit);
[x_deconv_LW_NonNeg, J2] = deconv_landweber_nonneg1(y, h, lam, Nit);


figure(2), clf
subplot(4, 2, 1)
imagesc(v)
axis image off
colormap(gray)
title('Astronomical image')
subplot(4, 2, 2)
plot(x)
xlim([0 N])
ylim([-20 250])
title('Row 100 (All pixels non-negative)')
subplot(4, 2, 3)
plot(y)
title('Convolution + noise')
xlim([0 N])
ylim([-20 250])
subplot(4, 2, 4)
plot(x_deconv)
```

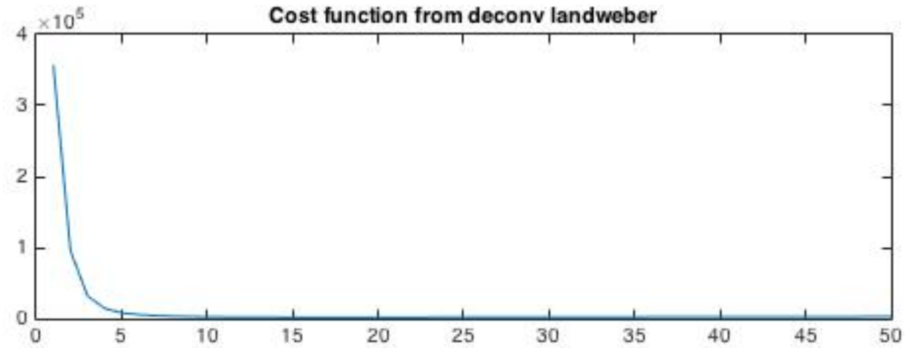
```
xlim([0 N])
ylim([-20 250])
title('MATLAB deconv function')

subplot(4, 2, 5)
plot(x_deconv_LW)
xlim([0 N])
ylim([-20 250])
title('Landweber deconvolution')

subplot(4, 2, 6)
  plot(x_deconv_LW_NonNeg)
xlim([0 N])
ylim([-20 250])
title('Landweber deconv with non-negativity')
%
orient tall
print -dpdf demo_deconv_nonneg1


% Matrix free method
function [x_deconv_LW, J1] = deconv_landweber1(y, h, lambda, Nit)
N = 256;
H = @(x) conv(h,x);
Ht = @(y) convt(h,y);                          % H : convolution matrix
e = ones(N, 1);
D = spdiags([e -2*e e], 0:2, N-2, N);

alpha = 1;
beta=max(eig(D'*D));

J1 = zeros(1, Nit); % Objective function
x_deconv_LW = 0*Ht(y); % Initialize x
T = lambda/(2*(alpha+lambda*beta));
for k = 1:Nit
Hx = H(x_deconv_LW);
Dx= D*x_deconv_LW;
J1(k) = sum(abs(Hx(:)-y(:)).^2) + lambda*sum(abs(Dx(:)).^2);
x_deconv_LW = wthresh(x_deconv_LW*alpha + (Ht(y - Hx))/(alpha+lambda*beta),'s',T);
end


% Matrix free method
function [x_deconv_LW_NonNeg, J2] = deconv_landweber_nonneg1(y, h, lambda, Nit);
N = 256;
H = @(x) conv(h,x);
Ht = @(y) convt(h,y);                          % H : convolution matrix
e = ones(N, 1);
D = spdiags([e -2*e e], 0:2, N-2, N);

alpha = 1;
beta=max(eig(D'*D));

J2 = zeros(1, Nit); % Objective function
x_deconv_LW_NonNeg = 0*Ht(y); % Initialize x
```

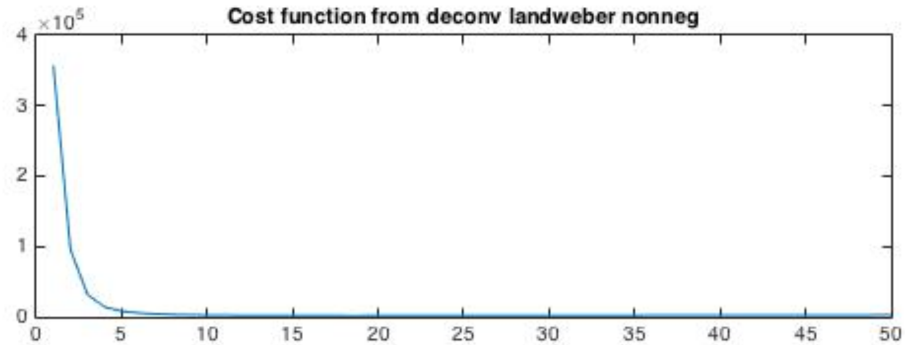
```
T = lambda/(2*(alpha+lambda*beta));
for k = 1:Nit
Hx = H(x_deconv_LW_NonNeg);
Dx= D*x_deconv_LW_NonNeg;
J2(k) = sum(abs(Hx(:)-y(:)).^2) + lambda*sum(abs(Dx(:)).^2);
x_deconv_LW_NonNeg= wthresh(x_deconv_LW_NonNeg*alpha + (Ht(y - Hx))/(alpha+lambda*beta),'s',T);
x_deconv_LW_NonNeg(x_deconv_LW_NonNeg<0)=0;
end


function f = convt(h, g)
% f = convt(h, g);
% Transpose convolution: f = H' g
Nh = length(h);
Ng = length(g);
f = conv(h(Nh:-1:1), g);
f = f(Nh:Ng);
end
```

Output




Astronomical image
Row 100 (All pixels non-negative)
Convolution + noise
MATLAB deconv function
Landweber deconvolution
Landweber deconv with non-negativity
0
50
100
150
200
250





Result:

Here least squares deconvolution with non-negativity constraints is performed. The algorithm is based on ISTA (iterative shrinkrage-thresholding algorithm) which is a variant of the Forward-Backward Splitting algorithm . It can be see from the output that landweber deconvolution with non negativity gives better result then landweber deconvolution. All those negative value is removed as the image cannot have negative value.It is verified that your algorithm converges by plotting the value of the cost function for each iteration.

Question 2
2) Two-dimensional least square deconvolution by matrix-free Landweber iteration.

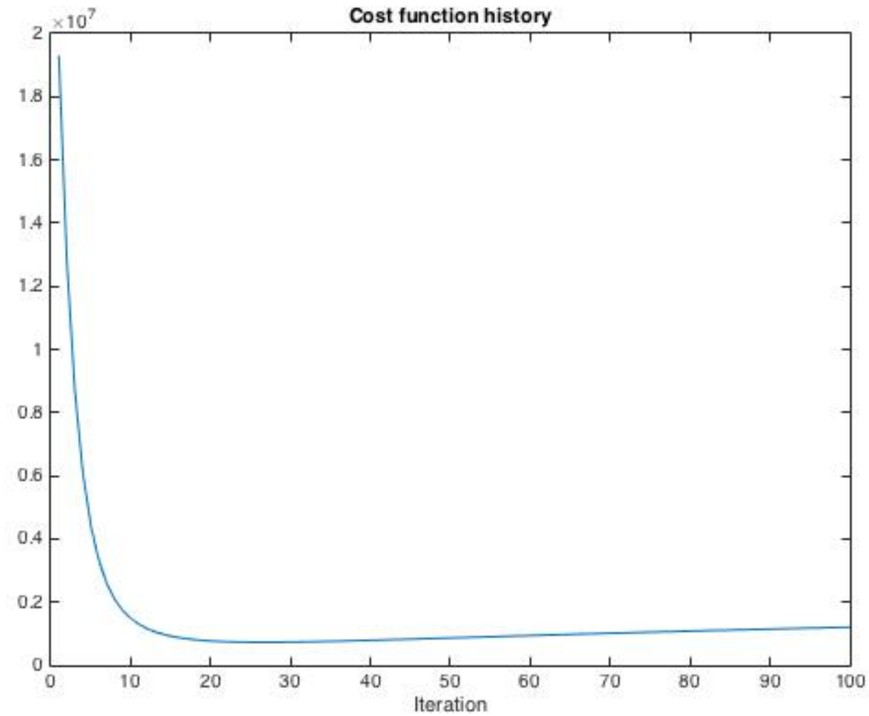
```
%% demo_image_deconv
% Image deconvolution using least squares and the Landweber algorithm
clear

%% Load image

x = imread('ngc_atrou.gif');    % Astronomical image from Jean Luc Starck
x = double(x);              % Convert from 8-bit integer to floating point
[N1, N2] = size(x);

figure(1), clf
imagesc(x)
axis image
colormap(gray)
title('Astronomical image')
```

```
% Note: imagesc scales the image display so that
% black = minimum pixel value
% white = maximum pixel value

%% Convolution
% Create a blurry noisy image

% Two-dimensional impulse response
M = 6;
h = eye(M);
h = [h zeros(M, 3)];
h = h + fliplr(h);
h = h/sum(h(:));                    % The impulse response is not square

y = conv2(h, x);            % Two-dimensional convolution

y = y + 1.0 * randn(size(y));   % Add noise

figure(2), clf
imagesc(y)
axis image
colormap(gray)
title('Convolution + noise')

%% Deconvolution
% Can we estimate the original image x from the blurred noisy image y?

lam = 0.2;
Nit = 100;
[x_deconv_LW, J1] = image_deconv_landweber(y, h, lam, Nit);

figure(1), clf
imagesc(x_deconv_LW);
axis image
colormap(gray)
title('Landweber deconvolution')

% Verify convergence of algorithm
figure(2)
plot(J1)
title('Cost function history')
xlabel('Iteration')

%% Deconvolution with non-negativity constraint
% We know the pixels are non-negative, so let us include that
% constraint in the method.

[x_deconv_LW_NonNeg, J2] = image_deconv_landweber_nonneg(y, h, lam, Nit);

figure(1), clf
imagesc(x_deconv_LW_NonNeg)
axis image
colormap(gray)
title('Landweber deconv with non-negativity')
```

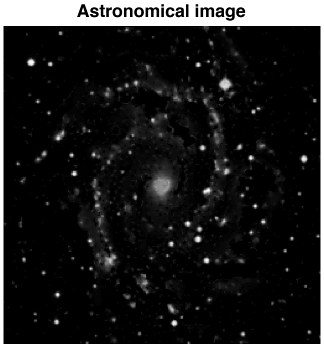
```
% Verify convergence of algorithm
figure(2)
plot(J2)
title('Cost function history')
xlabel('Iteration')

%% Display results
% The algorithm does not perfectly recover the original image,
% but it does significantly correct the blurring.

figure(1), clf
subplot(2, 2, 1)
imagesc(x)
colormap(gray)
axis image off
title('Astronomical image')

subplot(2, 2, 2)
imagesc(y)
colormap(gray)
axis image off
title('Convolution + noise')

subplot(2, 2, 3)
imagesc(x_deconv_LW)
axis image off
title('Landweber deconvolution')

subplot(2, 2, 4)
imagesc(x_deconv_LW_NonNeg)
axis image off
title('Landweber deconv with non-negativity')

orient landscape
print -dpdf demo_image_deconv_fig11

%% Display a single row
% Note that without the non-negativity constraint, some pixels
% negative.

M = 177;

figure(1), clf
subplot(2, 2, 1)
plot(x(M, :))
ylim([-20 250])
title('Astronomical image (row 177)')

subplot(2, 2, 2)
plot(y(M, :))
ylim([-20 250])
title('Convolution + noise (row 177)')

subplot(2, 2, 3)
```

```
plot(x_deconv_LW(M, :))
ylim([-20 250])
title('Landweber deconvolution (row 177)')

subplot(2, 2, 4)
plot(x_deconv_LW_NonNeg(M, :))
ylim([-20 250])
title('Landweber deconv with non-negativity (row 177)')

orient landscape
print -dpdf demo_image_deconv_fig21


%% Frequency response of 2D impulse response h
% It is informative (but not necessary) to display the frequency response

Nfft = 64;
Hf = fft2(h, Nfft, Nfft);
Hf = fftshift(Hf);
f = (0:Nfft-1)/Nfft - 0.5;

figure(1), clf
subplot(2, 1, 1)
mesh(f, f, abs(Hf), 'Edgecolor', 'black')
title('Frequency response of 2D impulse response')
xlabel('f_1')
ylabel('f_2')

%% Frequency response of 2D high-pass difference filter d

d = [1 -1; -1 1];
Df = fft2(d, Nfft, Nfft);
Df = fftshift(Df);

subplot(2, 1, 2)
mesh(f, f, abs(Df), 'Edgecolor', 'black')
title('Frequency response of 2D impulse response')
xlabel('f_1')
ylabel('f_2')

orient landscape
print -dpdf demo_image_deconv_fig31




function [x_deconv_LW, J1] = image_deconv_landweber(y, h, lambda, Nit)

% Matrix free method
N = 256;
H = @(x) conv2(h,x);
Ht = @(y) convt2(h,y);                      % H : convolution matrix
e = ones(N, 1);
D = spdiags([e -2*e e], 0:2, N-2, N);
```

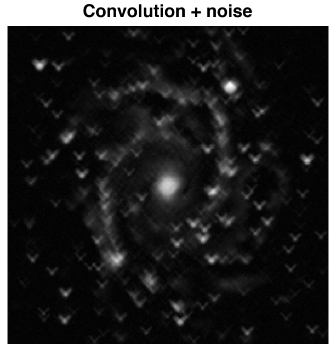
```
alpha = 1;
beta=max(eig(D'*D));

J1 = zeros(1, Nit); % Objective function
x_deconv_LW = 0*Ht(y); % Initialize x
T = lambda/(2*(alpha+lambda*beta));
for k = 1:Nit
Hx = H(x_deconv_LW);
Dx= D*x_deconv_LW;
J1(k) = sum(abs(Hx(:)-y(:)).^2) + lambda*sum(abs(Dx(:)).^2);
x_deconv_LW = wthresh(x_deconv_LW*alpha + (Ht(y - Hx))/(alpha+lambda*beta),'s',T);
end
end

function [x_deconv_LW_NonNeg, J2] = image_deconv_landweber_nonneg(y, h, lambda, Nit)
%matrix free
N = 256;
H = @(x) conv2(h,x);
Ht = @(y) convt2(h,y);                    % H : convolution matrix
e = ones(N, 1);
D = spdiags([e -2*e e], 0:2, N-2, N);

alpha = 1;
beta=max(eig(D'*D));

J2 = zeros(1, Nit); % Objective function
x_deconv_LW_NonNeg = 0*Ht(y); % Initialize x
T = lambda/(2*(alpha+lambda*beta));
for k = 1:Nit
Hx = H(x_deconv_LW_NonNeg);
Dx= D*x_deconv_LW_NonNeg;
J2(k) = sum(abs(Hx(:)-y(:)).^2) + lambda*sum(abs(Dx(:)).^2);
x_deconv_LW_NonNeg= wthresh(x_deconv_LW_NonNeg*alpha + (Ht(y - Hx))/(alpha+lambda*beta),'s',T);
x_deconv_LW_NonNeg(x_deconv_LW_NonNeg<0)=0;
end
end


function f = convt2(h,g)
% f = convt(h,g);
% Transpose convolution: f = H? g
[Nhh,Nhw] = size(h);
[Ngh,Ngw] = size(g);
f = conv2(h(Nhh:-1:1,Nhw:-1:1), g);
f = f(Nhh:Ngh,Nhw:Ngw);
end
```



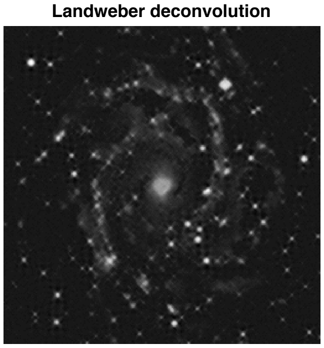
Astronomical image



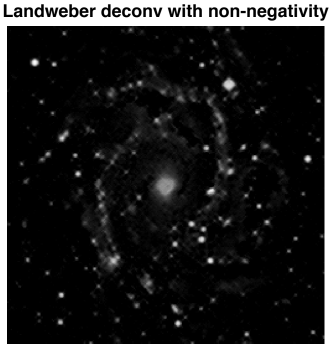
Convolution + noise

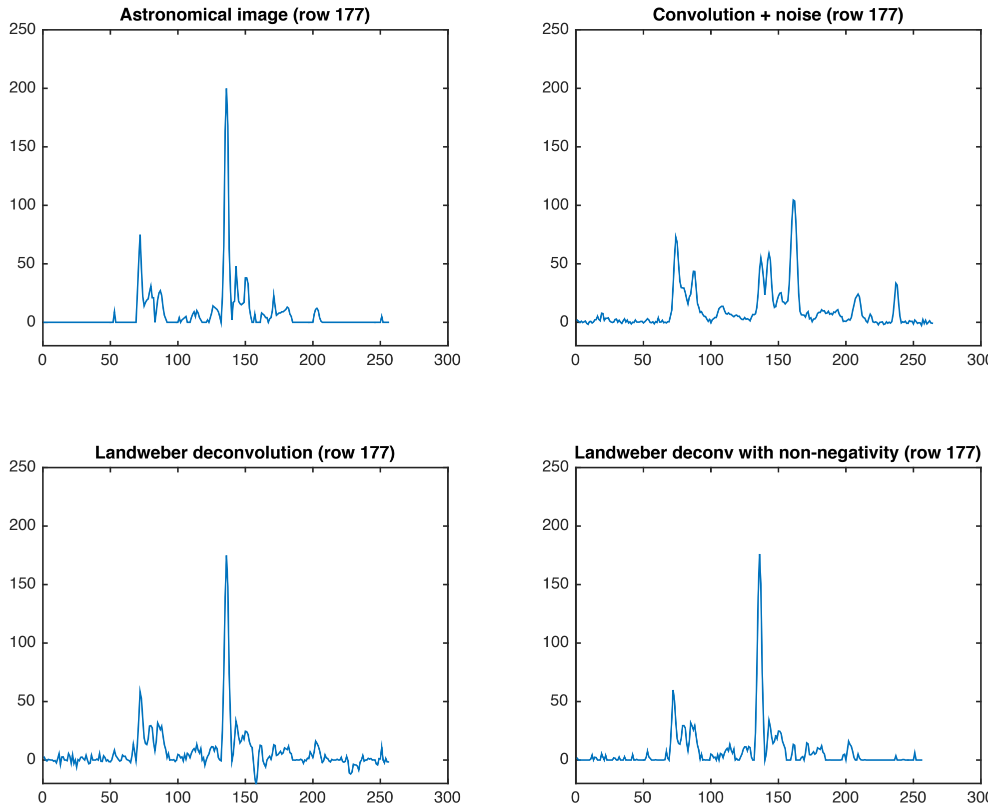


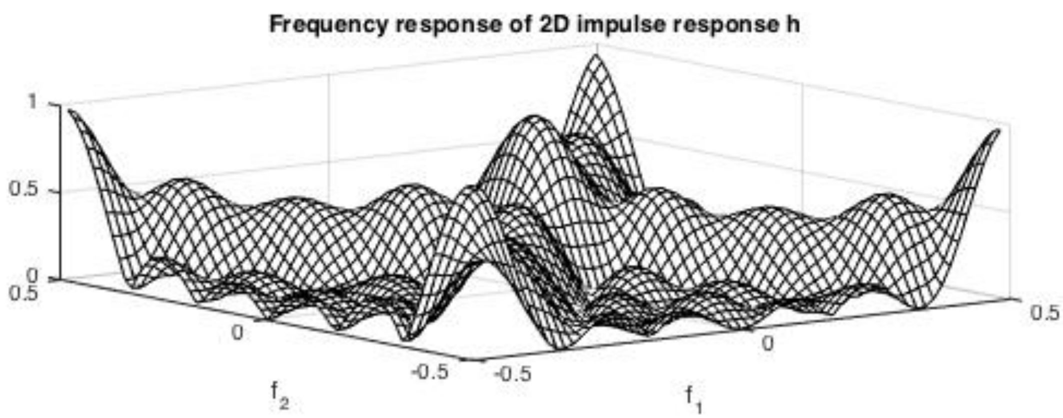
Landweber deconvolution

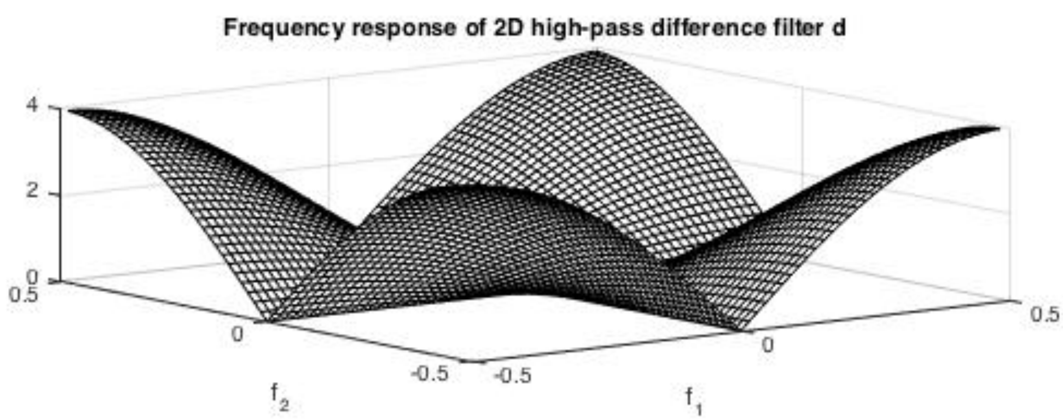


Landweber deconv with non-negativity




Astronomical image (row 177)
Convolution + noise (row 177)
Landweber deconvolution (row 177)
Landweber deconv with non-negativity (row 177)


Frequency response of 2D impulse response h
f
2
f
1


Frequency response of 2D high-pass difference filter d
f
2
f
1

Question 3

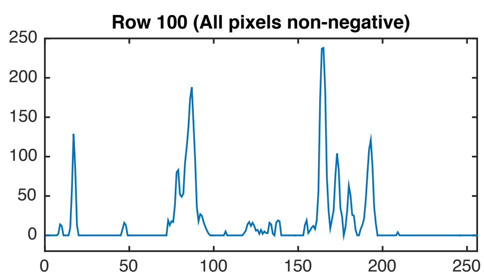
3) One-dimensional deconvolution using Haar wavelet sparsity

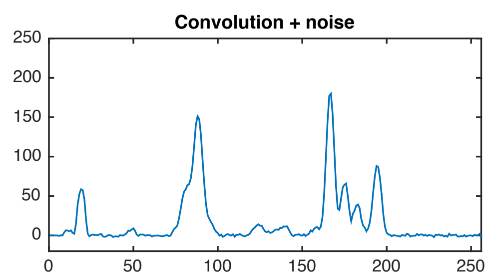
```
clear all;
close all;
v = imread('ngc_atrou.gif');     % Astronomical image from Jean Luc Starck
v = double(v);                   % Convert from 8-bit integer to floating point

figure(1), clf
imagesc(v)
axis image
colormap(gray)

x = v(190, :);
N = length(x);

M = 5;
h = ones(1, M)/M;

y = conv(h, x);
y = y + 1.0 * randn(size(y));

%% Least square deconvolution
% using the Landweber algorithm

lam = 0.04;
Nit = 50;
[x_deconv_LW, cost_landweber] = deconv_landweber(y, h, lam, Nit);

%% Sparse Haar wavelet deconvolution
% Using the iterative shrinkage-thresholding algorithm (ISTA)

lam = 0.2;
J = 4;          % number of wavelet levels
[x_deconv_sparswlet, cost_sparswlet, w] = deconv_sparse_Haar(y, h, J, lam, Nit);

%% Display signals

figure(2), clf
subplot(3, 1, 1)
imagesc(v)
axis image off
colormap(gray)
title('Astronomical image')

subplot(3, 2, 3)
plot(x)
xlim([0 N])
ylim([-20 250])
title('Row 100 (All pixels non-negative)')

subplot(3, 2, 4)
plot(y)
title('Convolution + noise')
xlim([0 N])
ylim([-20 250])

subplot(3, 2, 5)
plot(x_deconv_LW)
xlim([0 N])
```

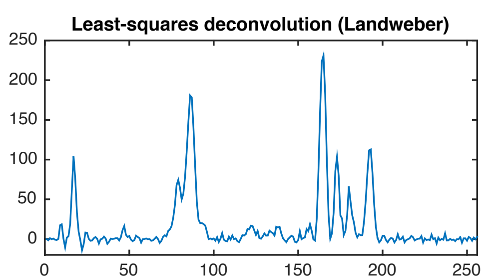
```
ylim([-20 250])
title('Least-squares deconvolution (Landweber)')

subplot(3, 2, 6)
plot(x_deconv_sparswlet);
xlim([0 N])
ylim([-20 250])
title('Sparse Haar deconvolution (ISTA)')

orient landscape
print -dpdf demo_deconv_sparse_Haar1


% Matrix free method
function [x_deconv_LW, J1] = deconv_landweber(y, h, lambda, Nit)
y=y';
h=h';
N = 256;
H = @(x) conv(h,x);
Ht = @(y) convt(h,y);                          % H : convolution matrix
e = ones(N, 1);
D = spdiags([e -2*e e], 0:2, N-2, N);

alpha = 1;
beta=max(eig(D'*D));

J1 = zeros(1, Nit); % Objective function
x_deconv_LW = 0*Ht(y); % Initialize x
% x_deconv_LW=x_deconv_LW';
T = lambda/(2*(alpha+lambda*beta));
for k = 1:Nit
Hx = H(x_deconv_LW);
Dx= D*x_deconv_LW;
J1(k) = sum(abs(Hx(:)-y(:)).^2) + lambda*sum(abs(Dx(:)).^2);
x_deconv_LW = wthresh(x_deconv_LW*alpha + (Ht(y - Hx))/(alpha+lambda*beta),'s',T);
end


% The cost function to minimized is
% F(w) = 0.5 || y - H invW w ||_2^2 + lambda || w ||_1
% where
% y - observed noisy blurry signal
% H - convolution operator
% invW - inverse Haar wavelet transform
% w - Haar wavelet coefficients

function [x_deconv_sparswlet, cost_sparswlet, w] = deconv_sparse_Haar(y, h, J,
lambda, Nit)
N = 256;
H = @(x) conv(h,x);
Ht = @(y) convt(h,y);                          % H : convolution matrix
e = ones(N, 1);
D = spdiags([e -2*e e], 0:2, N-2, N);
alpha = 1;
% Haar  and Inverse Transform
 w = Haar(y, J);           % w: wavelet coefficients
invW = InverseHaar(w, J, N); % invW: reconstructed signal
cost_sparswlet = zeros(1, Nit); % Objective function
x_deconv_sparswlet = 0*Ht(y); % Initialize x
p=zeros(1,length(y));
ck=Haar(p,J);
% x_deconv_sparswlet=H(invW);
```

```
T = lambda/(2*(alpha));
S = [w{:}];
for k = 1:Nit
% Hx = H(invW'*x_deconv_sparswlet);
S = [ck{:}];
invW = InverseHaar(w, J, N);
cost_sparswlet(k) = 0.5*sum(abs(y-H(invW)).^2) + lambda*sum(abs(S));
pk=Haar(Ht(y-H(invW)),J);
    for i=1:4
        ck{1,i}=wthresh(ck{1,i} + pk{1,i}/alpha,'s',T);
    end
x_deconv_sparswlet=InverseHaar(ck, J, N);
ck=Haar(H(x_deconv_sparswlet), J);
w=ck;
end
figure,plot(x_deconv_sparswlet);
figure,plot(cost_sparswlet);
```

Output





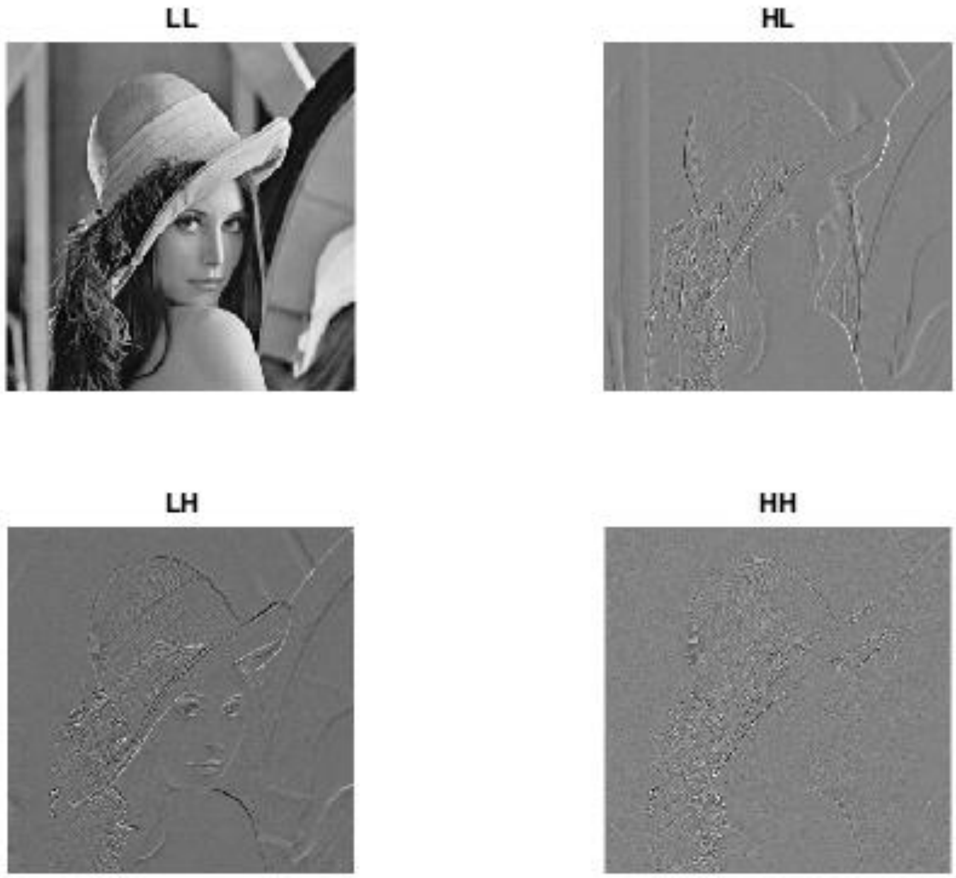


Result:
My estimated input signal after deconvolution by Landweber technique is approximately close to the original input signal. But there are negative values and a bit of noisy. I could not obtain the result for Sparse Haar Deconvolution ISTA. I am getting some error in the code and my objective function was not converging.

Question 4

4) Two-dimensional Haar Wavelet Transform. Implement the 2D Haar transform in Matlab. Verify perfect reconstruction condition and Parseval Energy Identity. Illustrate the 2D Haar transform as applied to an image.

```
clc;
clear all;
close all;
I=imread('lena_gray.bmp');
I=im2double(I);
[r,c]=size(I);

% 2D Forward HAAR Wavelet transform
S=zeros(r,c);
for i=1:r
    L{i,:}=Haar(I(i,:),1);
end

for i=1:r
    S(i,1:c/2)=[L{i,1}{1,2}];
    S(i,c/2+1:end)=[L{i,1}{1,1}];
end

for i=1:c
    L1{:,i}=Haar(S(:,i),1);
end

S1=zeros(r,c);
for i=1:c
    S1(1:r/2,i)=[L1{1,i}{1,2}];
    S1(r/2+1:end,i)=[L1{1,i}{1,1}];
end

LL=S1(1:r/2,1:c/2);
LH=S1(r/2+1:end,1:c/2);
HL=S1(1:r/2,c/2+1:end);
HH=S1(r/2+1:end,c/2+1:end);

figure,
subplot(2,2,1),imshow(LL,[]);
subplot(2,2,2),imshow(HL,[]);
subplot(2,2,3),imshow(LH,[]);
subplot(2,2,4),imshow(HH,[]);
```



Question 5
Project/paper topic:

My Project topic is on Bandelets.

1) "Sparse Geometric Image Representations With Bandelets "Erwan Le Pennec and Stéphane Mallat, Fellow, IEEE
IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 14, NO. 4, APRIL 2005

This paper introduces a new class of bases, called bandelet bases, which decompose the image along multiscale vectors that are elongated in the direction of a geometric flow. This geometric flow indicates directions in which the image gray levels have regular variations. The image decomposition in a bandelet basis is implemented with a fast subband-filtering algorithm. Bandelet bases lead to optimal approximation rates for geometrically regular images. For image compression and noise removal applications, the geometric flow is optimized with fast algorithms so that the resulting bandelet basis produces minimum distortion. Comparisons are made with wavelet image compression and noise-removal algorithms.

2) "Sparse and Redundant Representations , From Theory to Applications in Signal and Image Processing"
Textbook by Michael Elad ,The Technion – Israel Institute of Technology Haifa, Israel

3)" IMAGE COMPRESSION WITH GEOMETRICAL WAVELETS", E. Le Pennec and S. Mallat, "Image compression with geometrical wavelets," presented at the Int. Conf. Image Processing, Vancouver, BC, Canada, Sep. 2000.

We introduce a sparse image representation that takes advantage of the geometrical regularity of edges in images. A new class of one-dimensional wavelet orthonormal bases, called foveal wavelets, are introduced to detect and reconstruct singularities. Foveal wavelets are extended in two dimensions, to follow the geometry of arbitrary curves. The resulting two dimensional "bandelets" define orthonormal families that can restore close approximations of regular edges with few non-zero coefficients. A double layer image coding algorithm is described. Edges are coded with quantized bandelet coefficients, and a smooth residual image is coded in a standard two-dimensional wavelet basis.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.207.1302&rep=rep1&type=pdf

4) S.G. Mallat. "A Wavelet Tour of Signal Processing." 2nd Edition. Academic Press, 1999. ISBN 0-12-466606-X.

These are the links for few of the related paper and textbook.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1407972

file:///Users/amitesh/Downloads/[Michael_Elad]_Sparse_and_redundant_representation(BookZZ.org).pdf

http://www.cmap.polytechnique.fr/~mallat/papiers/07-NumerAlgo-MallatPeyre-BandletsReview.pdf

https://hal.archives-ouvertes.fr/hal-00359744/document

http://www.cmap.polytechnique.fr/~mallat/papiers/CRM-Mallat-Course1.pdf