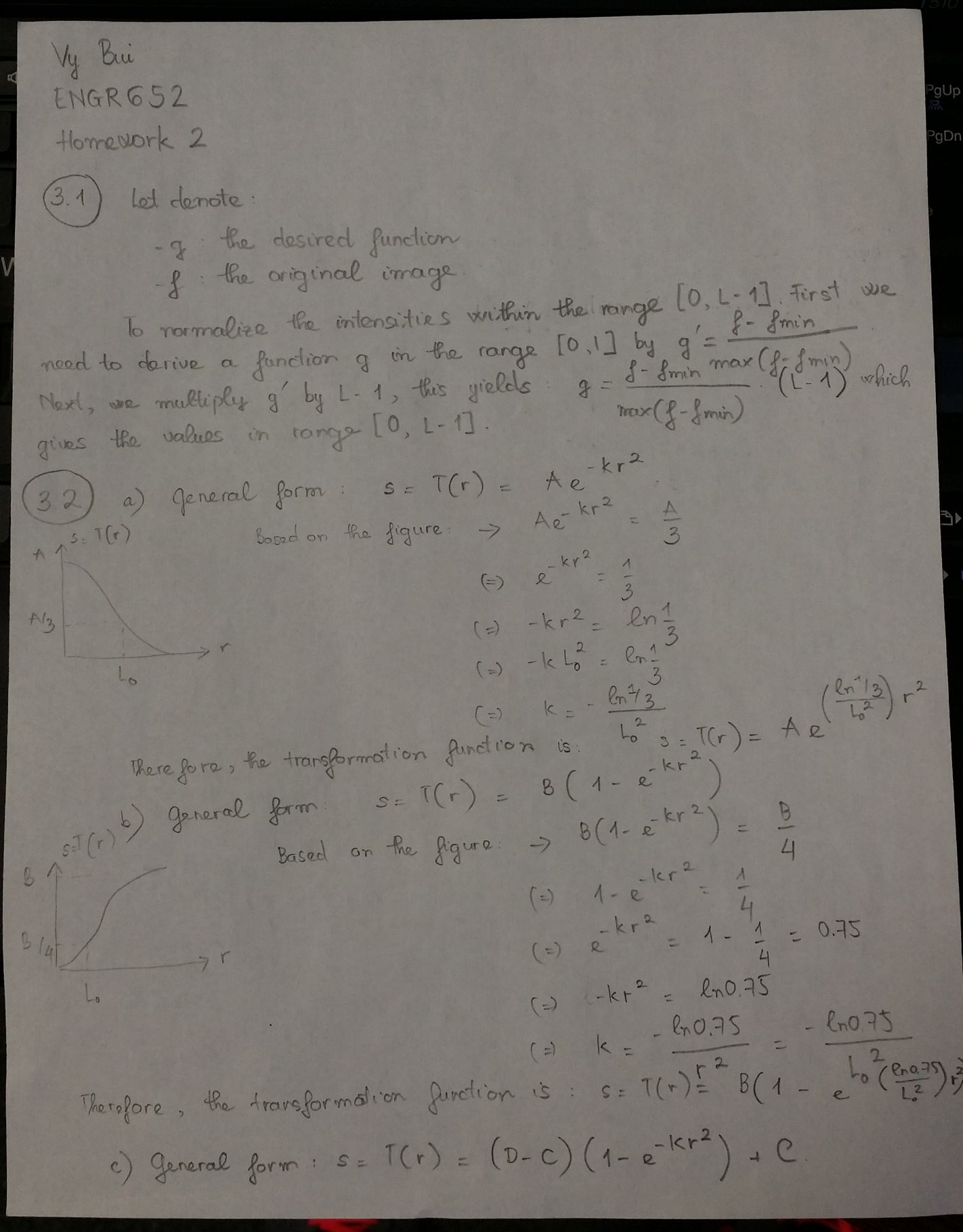
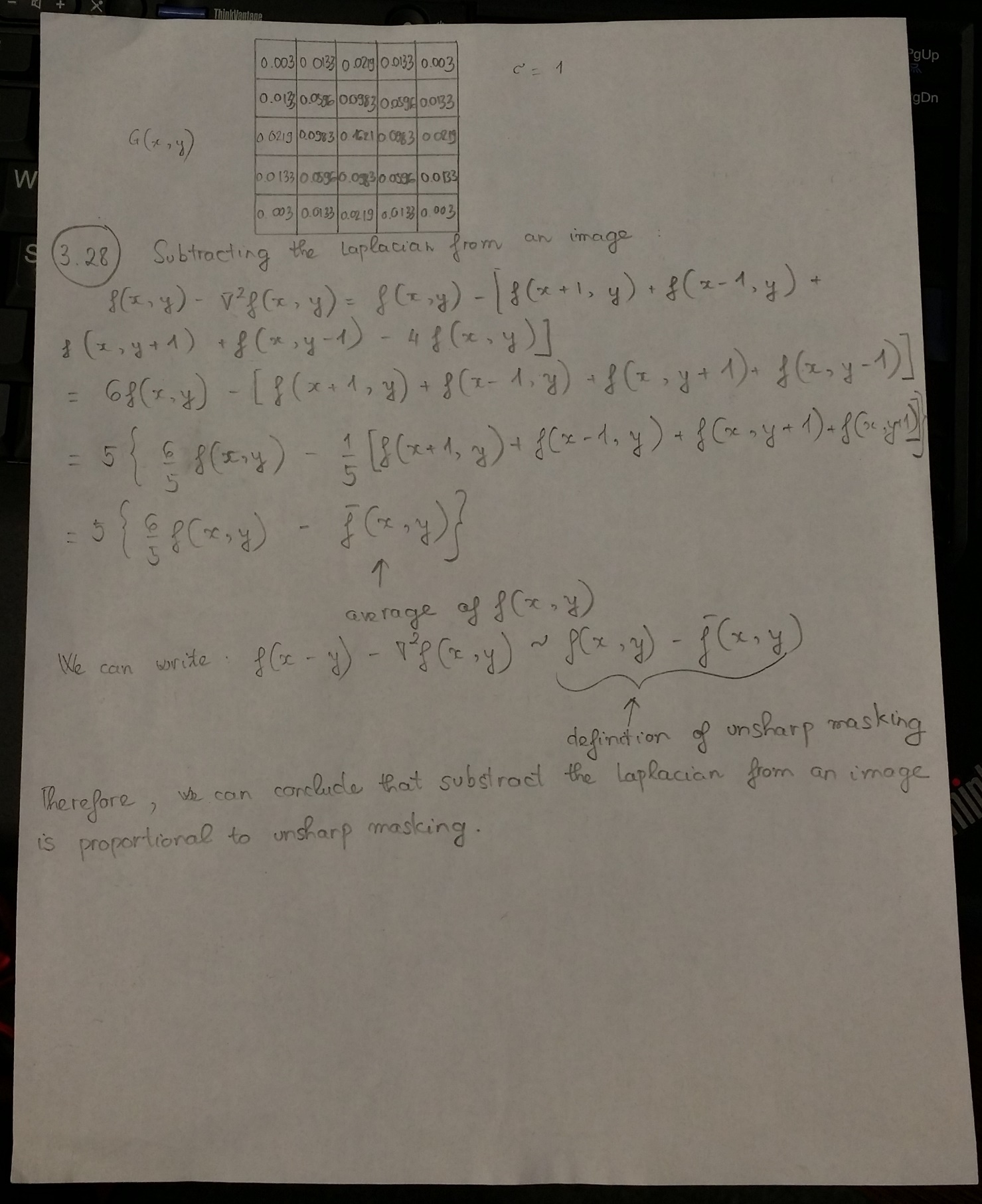
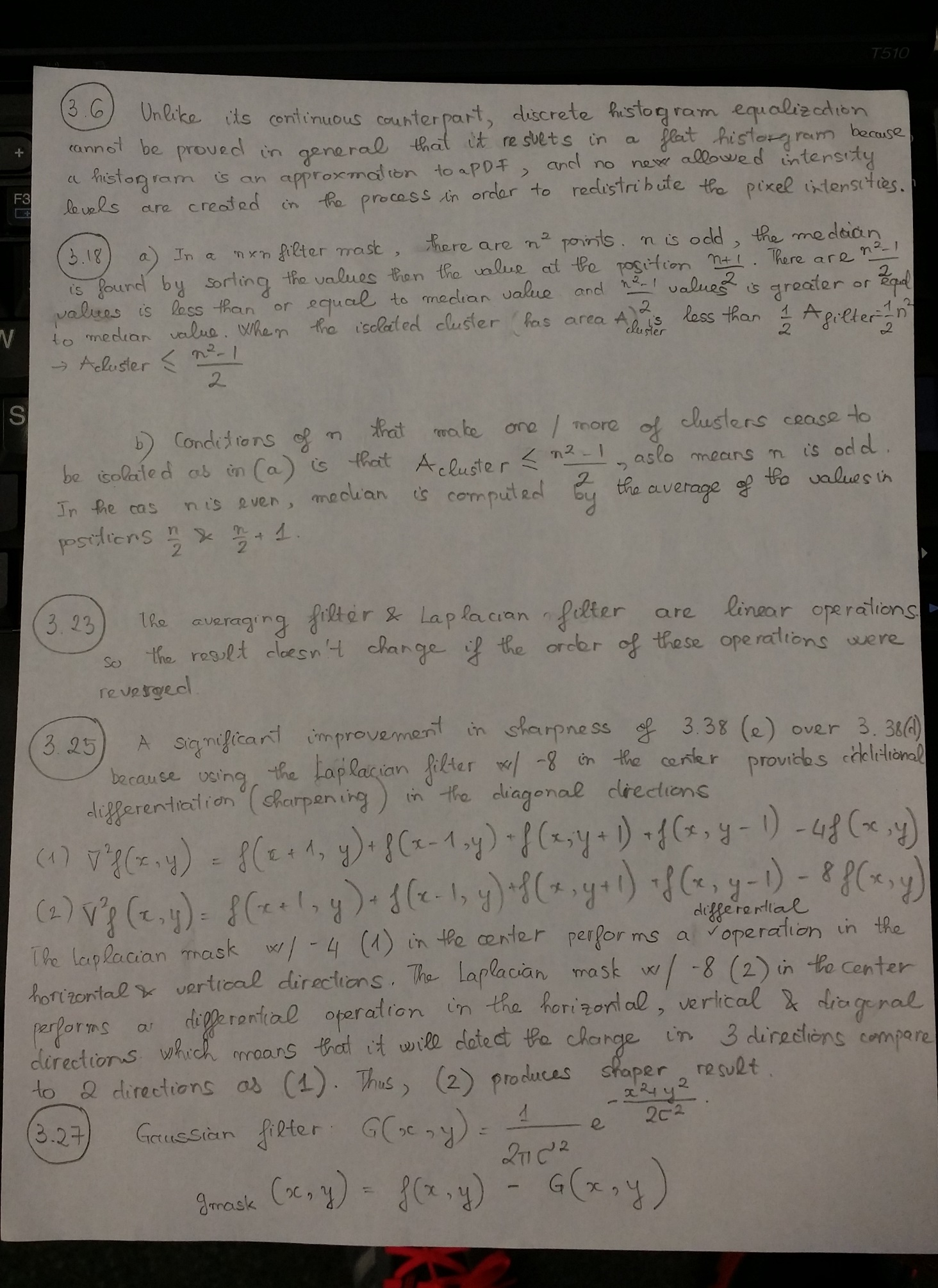
**Vy Bui**

**Homework 2**

**ENGR 652**





**MATLAB Exercise 1**

function [ out ] = imlin( in, G, B)

out = (in - min(in(:)))\*255/(max(in(:)) - min(in(:)));

out = G \* out + B;

end

**MATLAB Exercise 2**

function [ out ] = imlin2( in, a, b)

out = (in - min(in(:)))\*(b - a)/((max(in(:)) - min(in(:))) + a;

end

**MATLAB Exercise 3**

function [out,F]=myhisteq(in,L);

%

% [out,F]=myhisteq(in,L);

%

% My histogram equalization function

% Takes an L level input image 'in' (must be integers)

% and maps it to an L Level output

% with an approximately flat histogram.

%

% out equalized image (L levels)

% F mapping function

% in input image (L levels, integer values)

% L # levels in input image

im\_size = size(in);

row = im\_size(1);

col = im\_size(2);

total\_pixels = row\*col;

j = 1;

s = 0;

out = in;

for rk = 0:L-1;

nk(j) = length(find(in == rk));

pr(j) = nk(j)/total\_pixels;

s = s + (L-1)\*pr(j);

sk(j) = s;

out(find(in == rk)) = sk(j);

j = j + 1;

end

F = sk;

end

load wva

test = wva(:,:,1);

hist(test(:),100); figure;

image(test); colormap(gray(256)); axis image

test2=myhisteq(test,256);

figure; hist(test2(:),100); figure;

image(test2); colormap(gray(256)); axis image









**MATLAB Exercise 4**

clear

clc

%% Convolution image using 2D Gaussion

x = double(imread('cameraman.tif'));

figure

image(x)

colormap(gray(256));

title('Original Image');

% Gauss\_filter = fspecial('gaussian', [5 5], 3);

hx = fspecial('gaussian',[1,5],3) ;

hy = fspecial('gaussian',[5,1],3) ;

hx2=repmat(hx,[5,1]);

hy2=repmat(hy,[1,5]);

h=hx2.\*hy2;

tic

conv2D = conv2(x,h,'same');

toc

figure

image(conv2D);

colormap(gray(256));

title('Convolution image using 2D Gaussion');

%% Convolution image using 1D Gaussion

row\_Gauss = fspecial('gaussian',[1,5],3);

col\_Gauss = fspecial('gaussian',[5,1],3) ;

[row col] = size(x);

im\_row = reshape(x',1,row\*col);

tic

row\_conv = conv(im\_row,row\_Gauss,'same');

toc

row\_conv = reshape(row\_conv, col, row);

row\_conv = row\_conv';

im\_col = (row\_conv(:))';

tic

col\_conv = conv(im\_col,col\_Gauss','same');

toc

col\_conv = reshape(col\_conv, row, col);

figure

image(col\_conv);

colormap(gray(256));

title('Convolution image using seperated Gaussian vector filter');

%% Compare conv 1D and 2D

diff = conv2D - col\_conv;

figure

imshowpair(conv2D,col\_conv,'diff');







|  |  |  |
| --- | --- | --- |
| **Filter Size** | **Time 2D** | **Time 1D** |
| 2x2 | 0.000748 seconds | 0.001262 + 0.001134 seconds |
| 5x5 | 0.001052 seconds | 0.001336 + 0.001302 seconds |
| 9x9 | 0.002443 seconds | 0.000735 + 0.000977 seconds |

The difference between these two techniques is compared in Line 41-44. It’s observed that there’s only different at the edges of the two images.