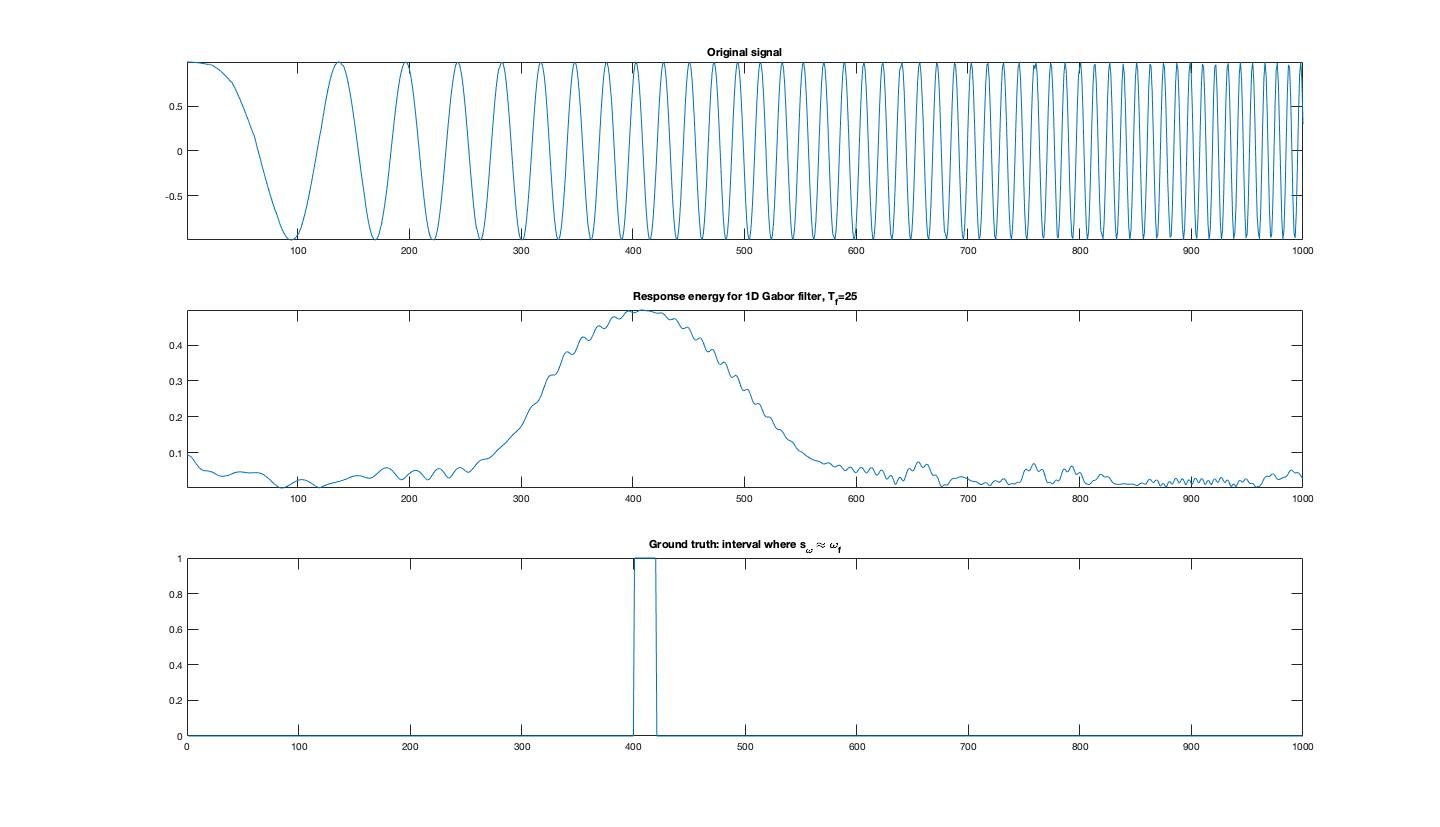
CIS580 Machine Perception

Homework 5

Apr.15.19

Chun Chang

1 Gabor 1d

(c) Gabor script

% Your code goes here %%%%%%%%%%%%%%%%%%%%

r1 = conv(s,g2,'same');

r2 = conv(s,g1,'same');

% energy = abs(r1) + abs(r2);

energy = sqrt(r1.^2 + r2.^2);

% End of your code %%%%%%%%%%%%%%%%%%%%%%%

(b)gabor 1d

n = -(len-1)/2:(len-1)/2;

X = gaussian1d(sigma, len);

filter\_cos = X .\* cos(n\*2\*pi / T\_f);

filter\_sin = X .\* sin(n\*2\*pi / T\_f);

1. Gaussian 1d

% use gaussian1d.m

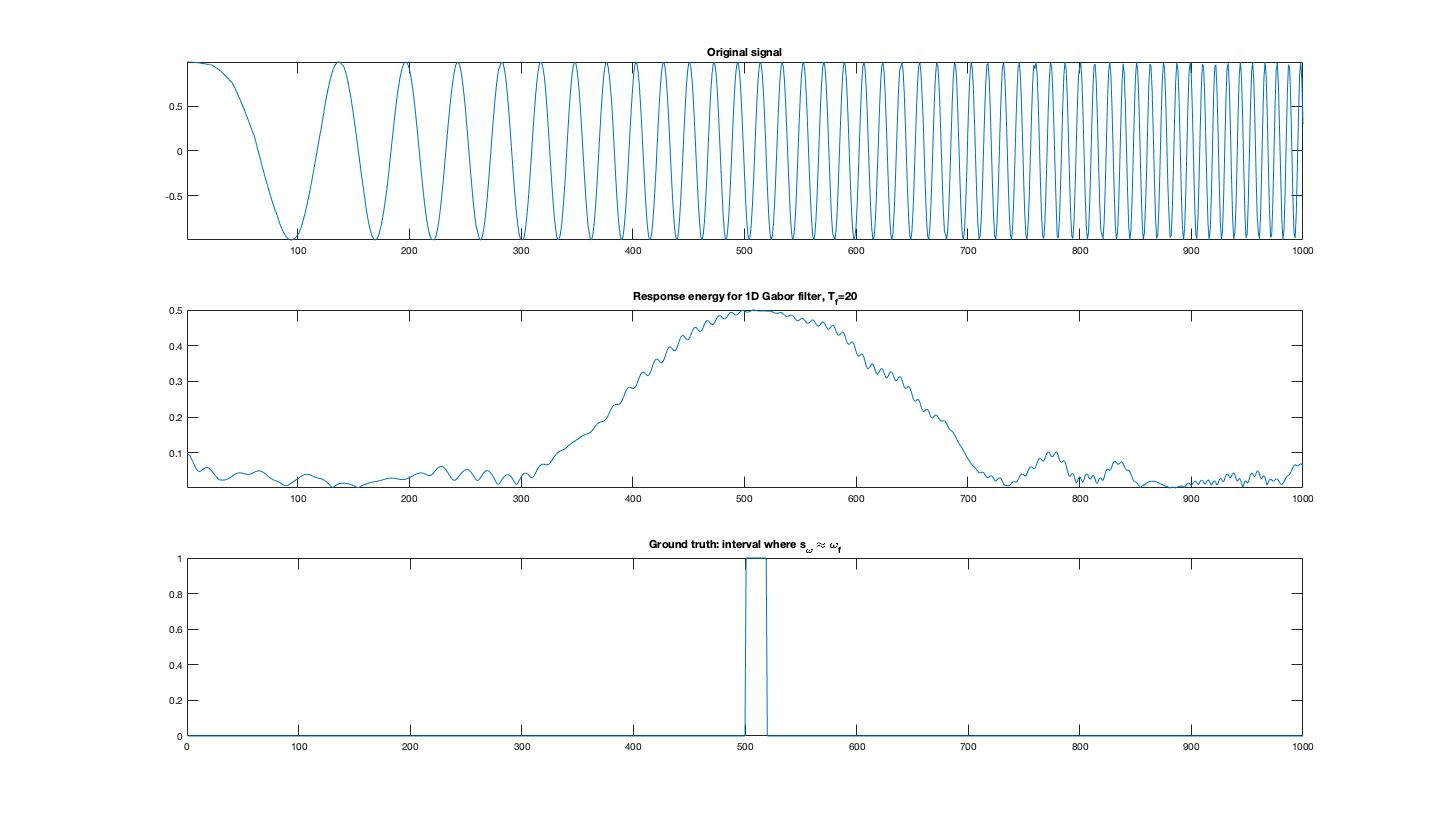
n = -(len-1)/2:(len-1)/2;

gau = exp(-n.^2/(2\*sigma^2))/(sigma \* sqrt(2\*pi));

gau\_norm = gau / sum(gau);

g = gau\_norm;

(d) two distinct time periods (20, 25)



2. Gabor 2D

(a)Gaussian 2d

% Your code goes here %%%%%%%%%%%%%%%%%%%%%%

n = -(len-1)/2: (len-1)/2;

g = zeros(len,len);

for i = 1 : len

for j = 1: len

g(i,j) = exp(-0.5 \* [n(j), n(i)] \* inv(Sigma) \* [n(j);n(i)])...

/ (2\* pi \* sqrt(det(Sigma)));

end

end

g = g / sum(g(:));

end

% End of your code %%%%%%%%%%%%%%%%%%%%%%%%%

(b)gabor 2d

% use gaussian2d.m

phi = theta / 180 \* pi;

g = gaussian2d(Sigma, len);

n = -(len-1)/2 : (len-1)/2;

an = 2\*pi/T\_f \*( n \* cos(phi) - n'\*sin(phi));

filter\_cos = g .\* cos(an);

filter\_sin = g .\* sin(an);

end

% End of your code %%%%%%%%%%%%%%%%%%%%%%%%%

(c) Gabor2d script

% Your code goes here %%%%%%%%%%%%%%%%%%%%

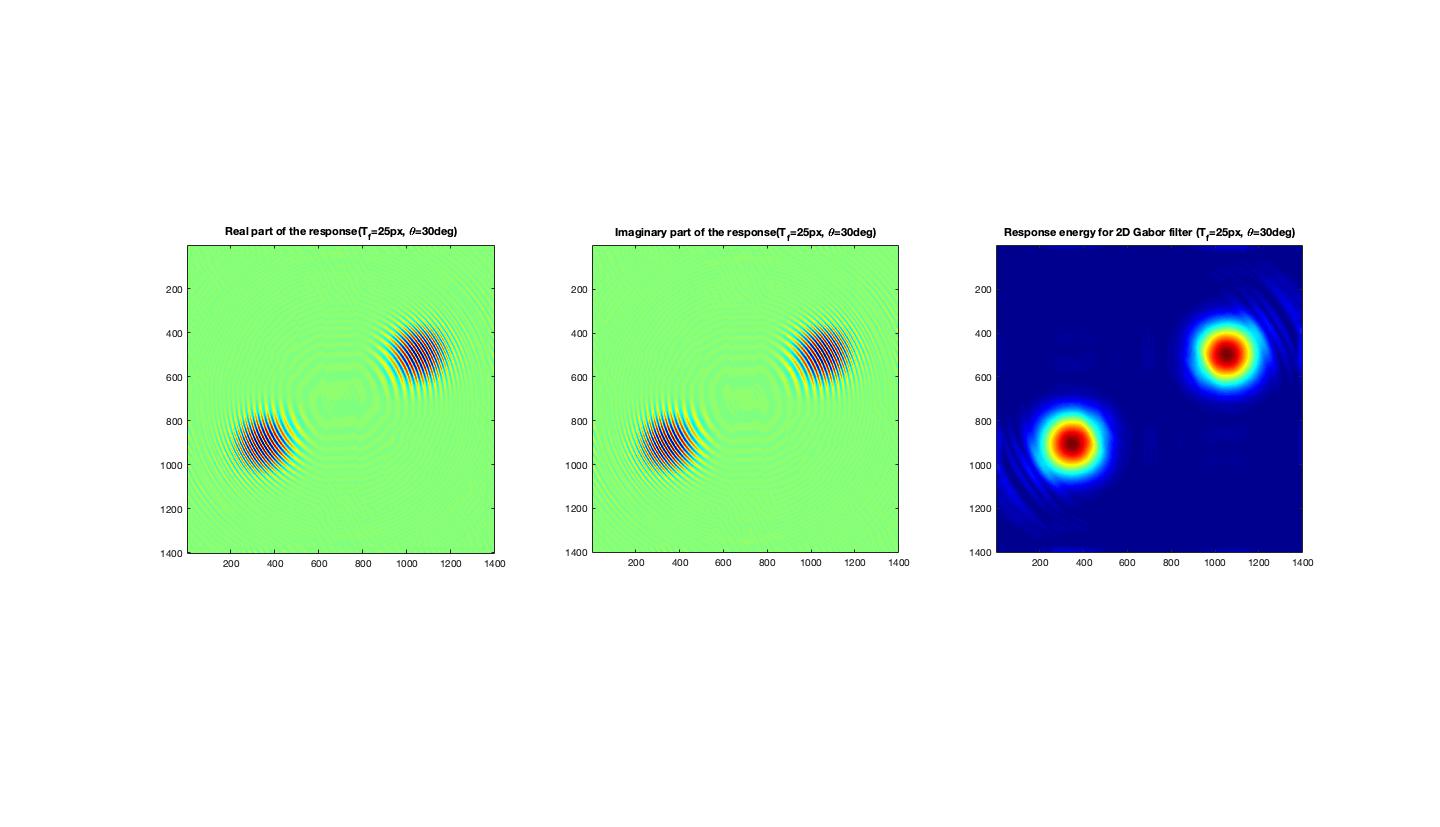
r1 = conv(s,g2,'same');

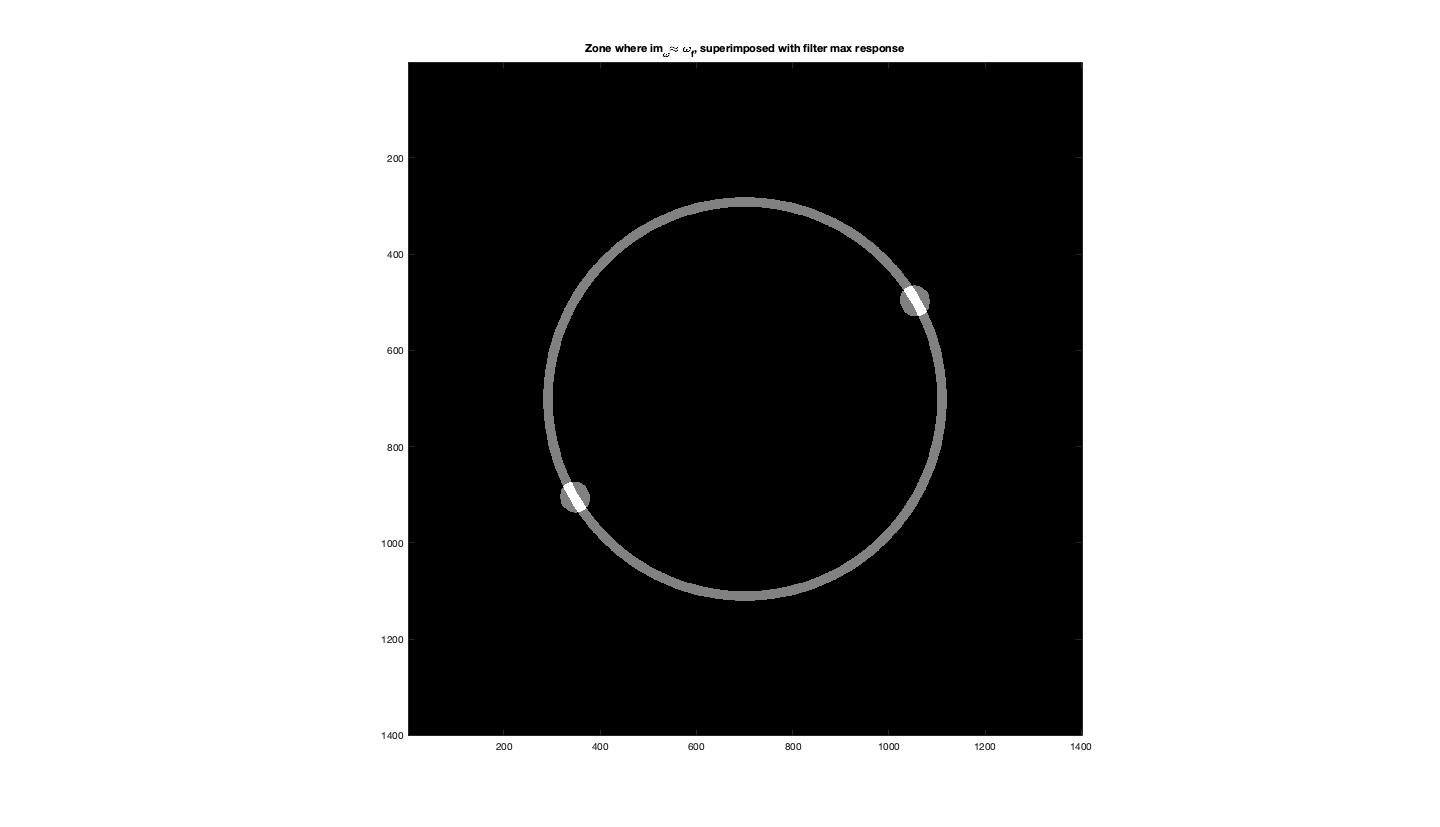
r2 = conv(s,g1,'same');

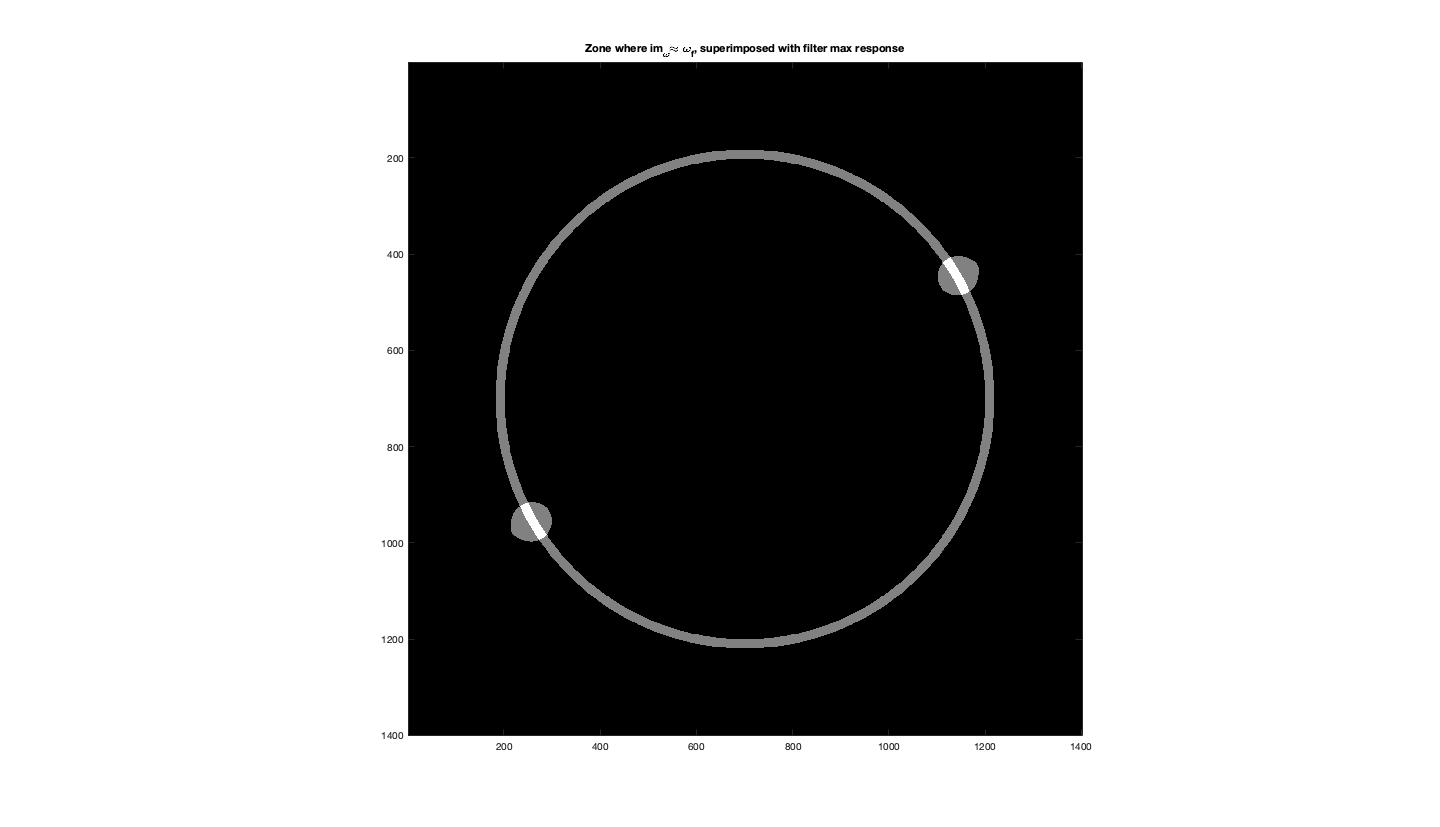
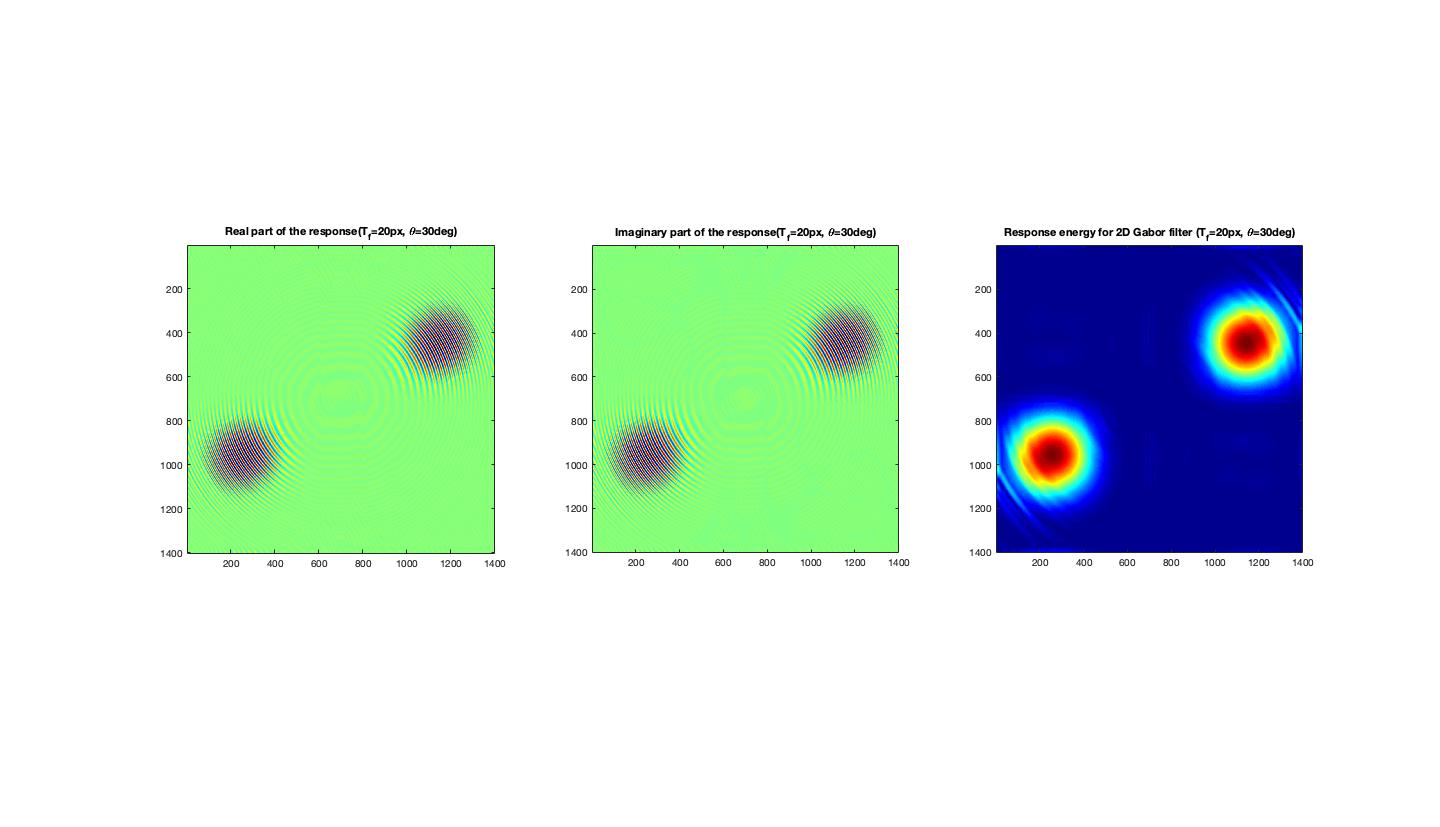
% energy = abs(r1) + abs(r2);

energy = sqrt(r1.^2 + r2.^2);

% End of your code %%%%%%%%%%%%%%%%%%%%%%%

(d) different time periods ( 20 & 25)





2 Scale invariant detection

1 Approximate LoG by DoG

1. DoG

DoG = dog1d(sigma, k, fSize);

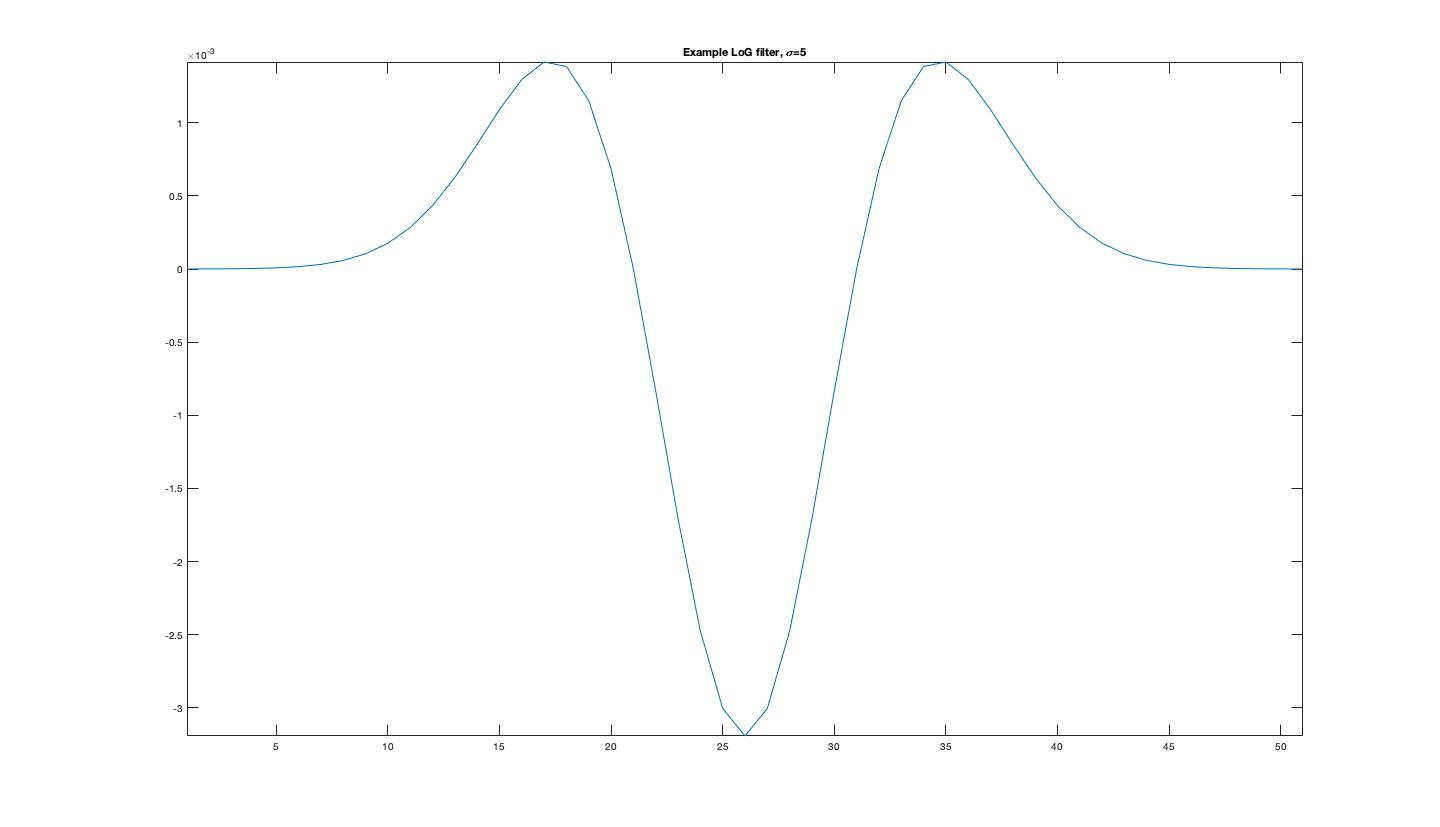
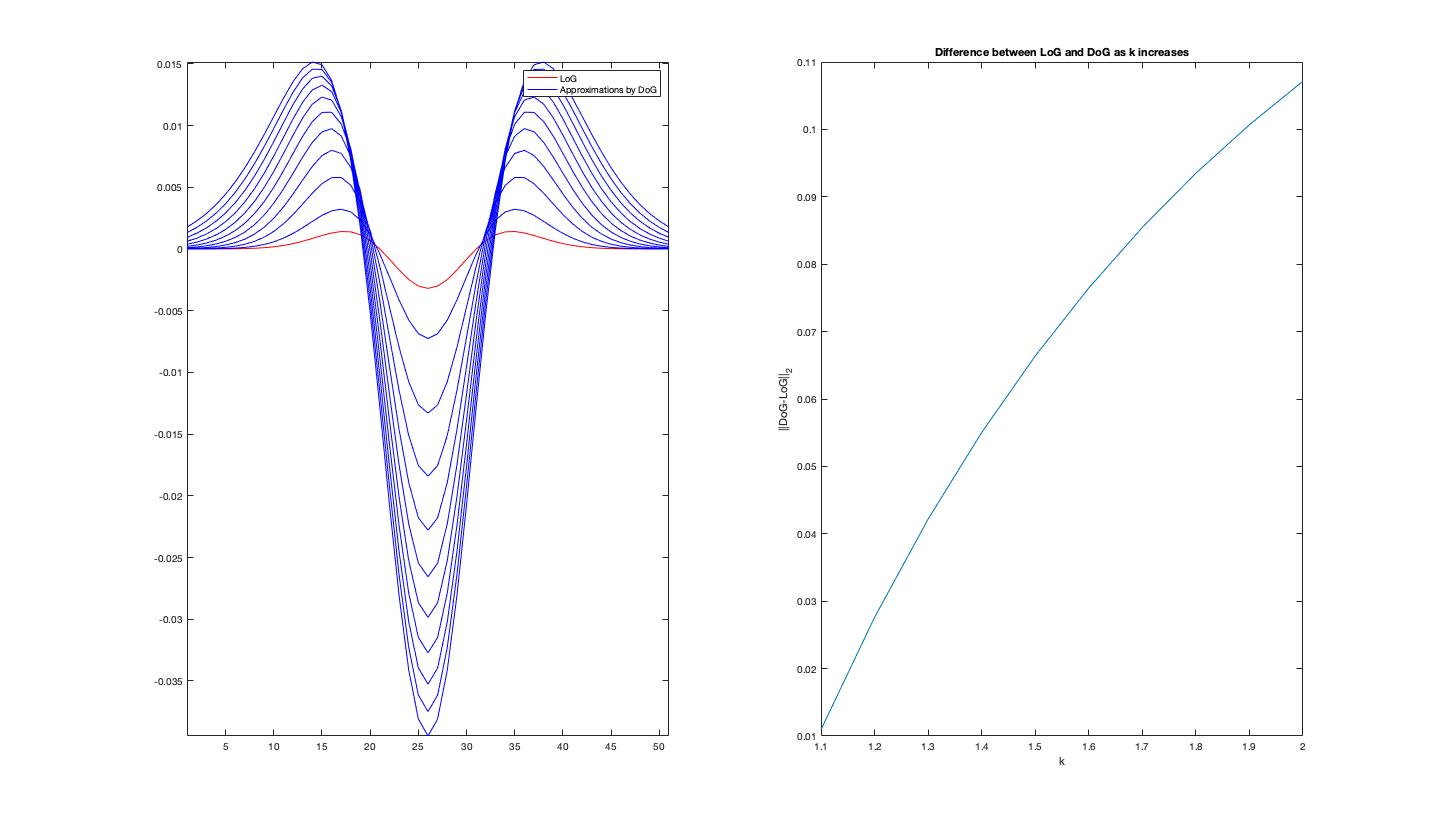
% use dog1d.m

LoG\_approx(i,:) = DoG;

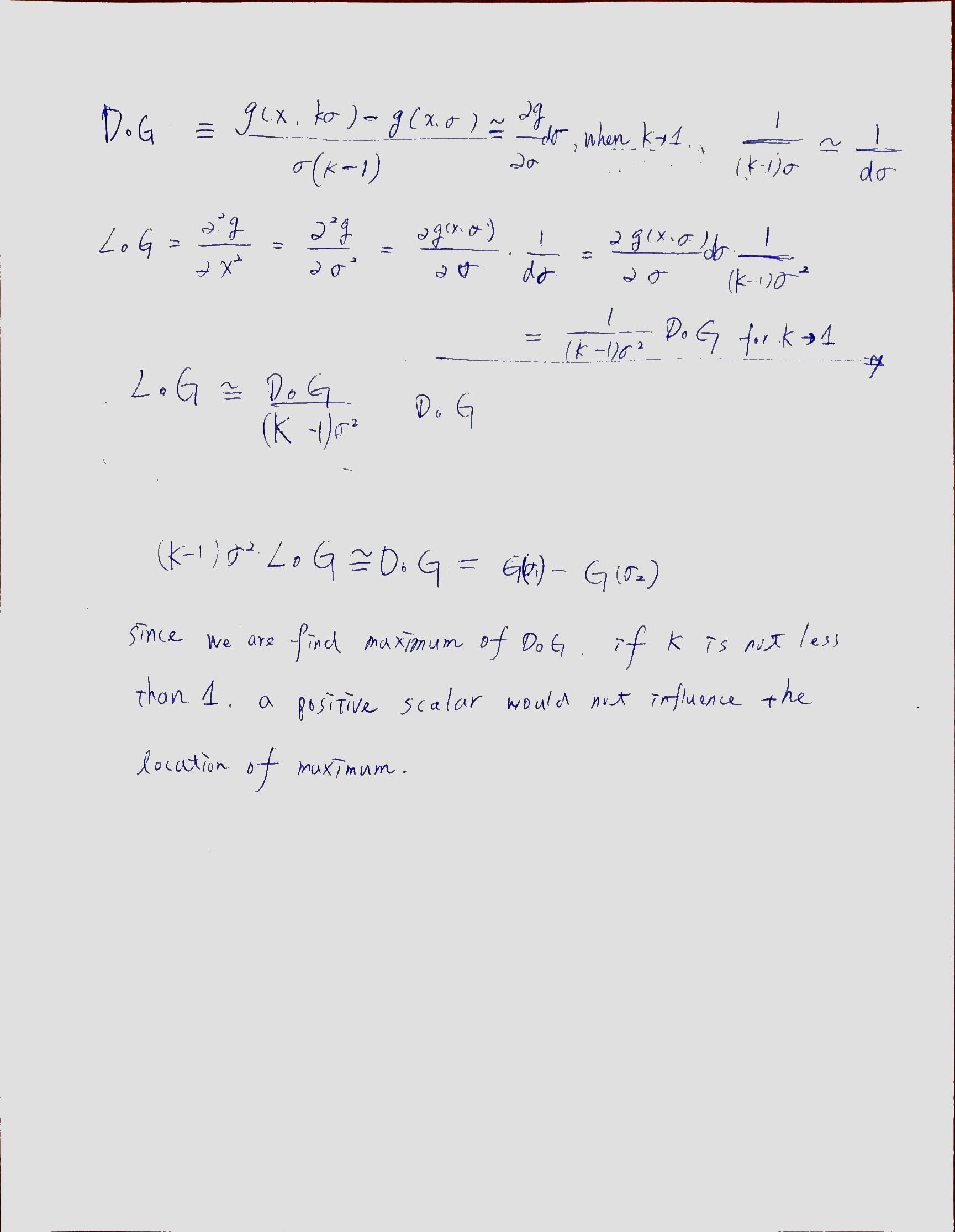
Dog 1d

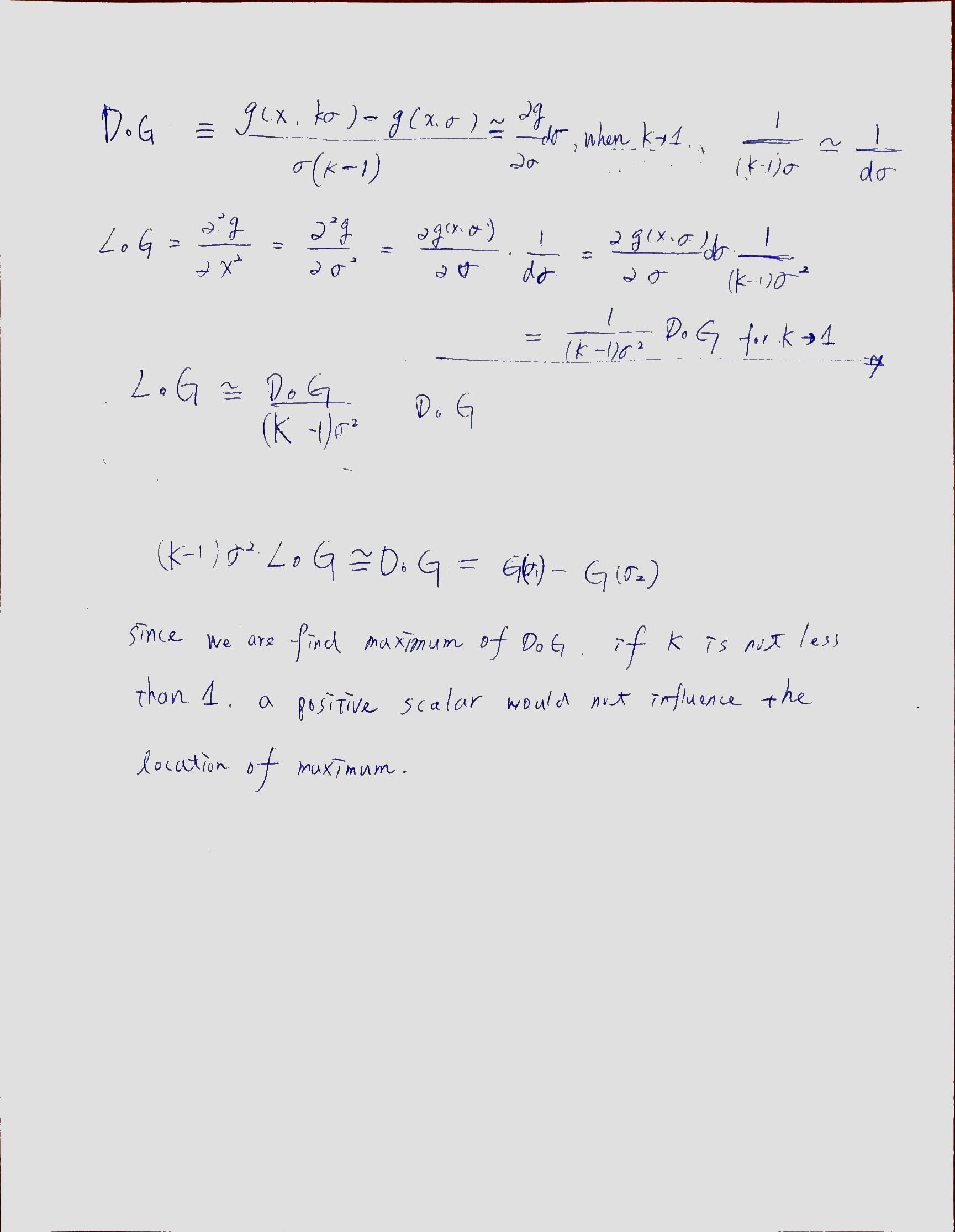
dog = gaussian1d(k\*sigma, len) - gaussian1d(sigma,len);

Quick comment: Difference of Gaussian is closer to laplace of gaussian when the coefficient k approaches to 1



（b）



（c）

2 flower detection

(a)convolution with 1d gaussian

% Your code starts here %%%%%%%%%%%%%%%%%%%%

% convolve with gaussians to build the scale space

gau\_i = gaussian1d(sigma\_i, round(fSize));

scales(:,:,i) = conv2(im, gau\_i, 'same' );

scales(:,:,i) = conv2(scales(:,:,i), gau\_i', 'same' );

% End of your code %%%%%%%%%%%%%%%%%%%%%%%%%

(b)find local maximum

ind = double.empty(0,1);

d\_max = 0.5 \* max(dog\_max(:));

for i = 1 : size(xmax)

if dog(ymax(i), xmax(i), smax(i)) > d\_max

ind = [ind; i];

end

end

ymax = ymax(ind);

xmax = xmax(ind);

smax = smax(ind);

(c) radius

% Your code goes here %%%%%%%%%%%%%%%%%%%%%%

r = sqrt(2)\*sigma\*k^(smax(i));

% End of your code %%%%%%%%%%%%%%%%%%%%%%%%%



