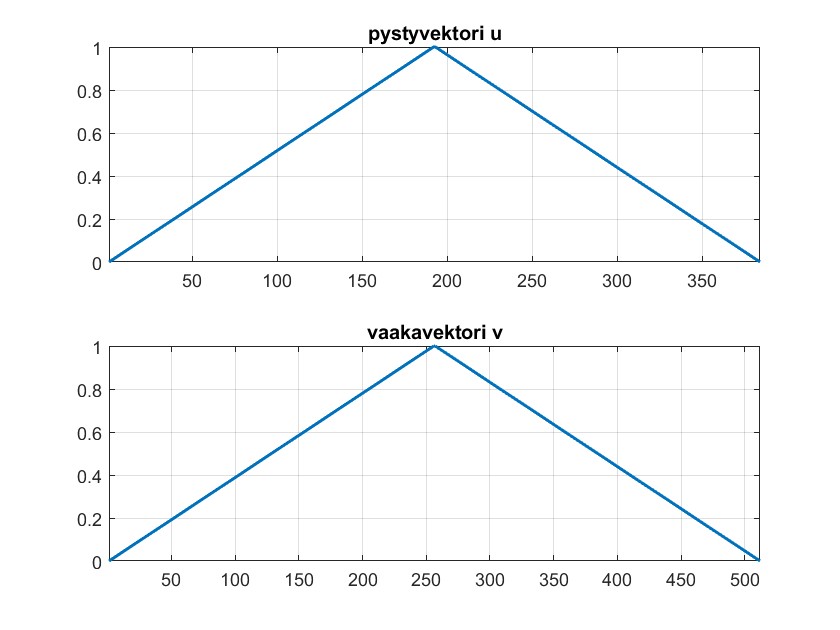
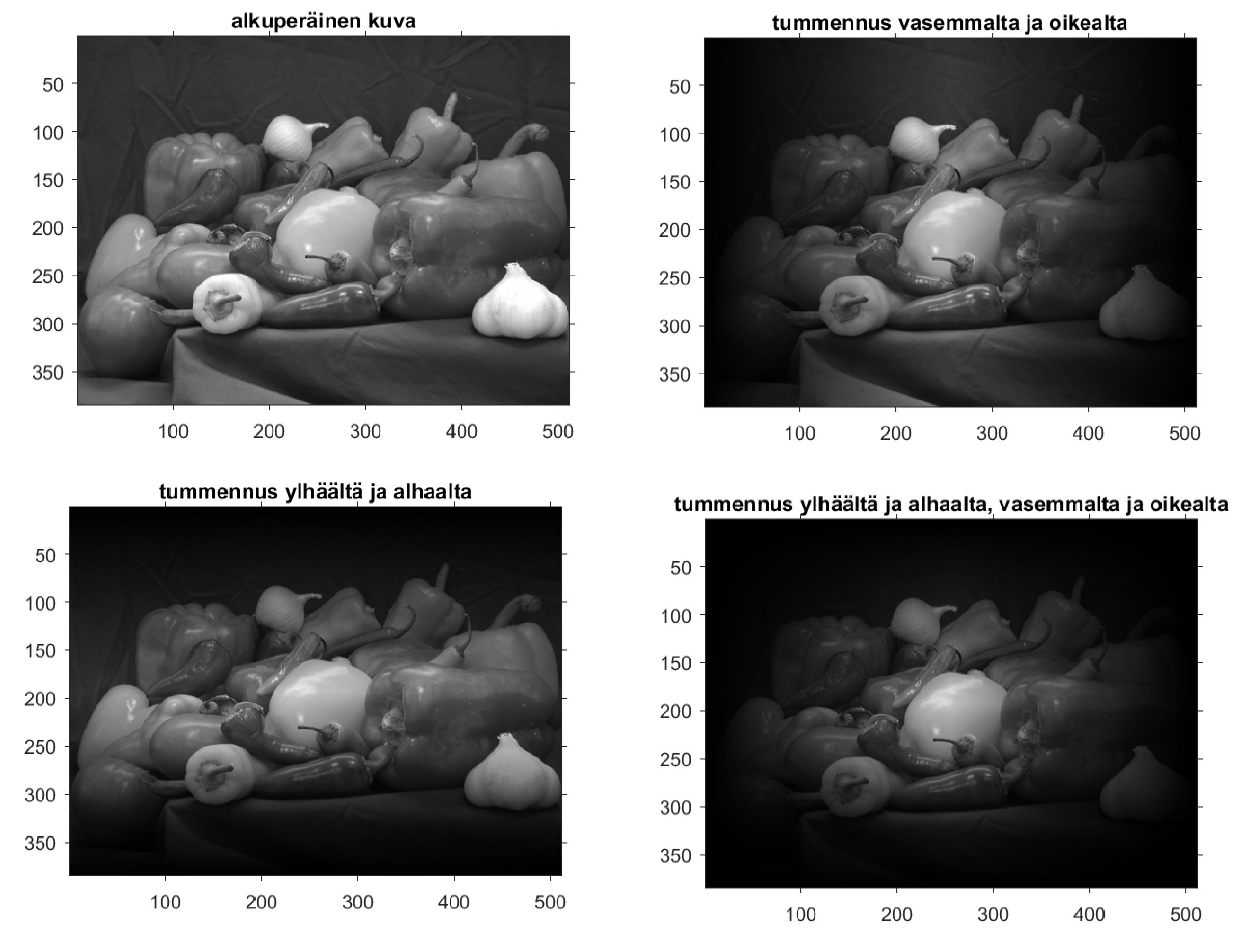
1. Read kuva.png to a matrix, create a column vector **u** and row vector **v** like below



and use them to darken the picture from left and right, from up and down and from left,right,up and down

hint: **u** = [**u**1;**u**2]*,* **v** = [**v**1*,***v**2]

alkuperainen¨ kuva = original picture tummennus vasemmalta ja oikealta = darkened from left and right tummennus ylha¨alt¨ a¨ ja alhaalta = darkened from up and down tummennus ylha¨alt¨ a¨ ja alhaalta, vasemmalta ja oikealta = darkened from up and down, left and right

answer:

% Read image to a matrix

img = imread('kuva.png');

% Create column and row vectors

u = [zeros(size(img,1),round(size(img,2)/2)); ones(size(img,1),round(size(img,2)/2))];

v = [zeros(round(size(img,1)/2),size(img,2)), ones(round(size(img,1)/2),size(img,2))];

% Darken from left and right

img\_dark\_lr = img .\* u;

% Darken from up and down

img\_dark\_ud = img .\* v;

% Darken from up and down, left and right

img\_dark\_all = img .\* (u .\* v);

% Display the original and darkened images

figure;

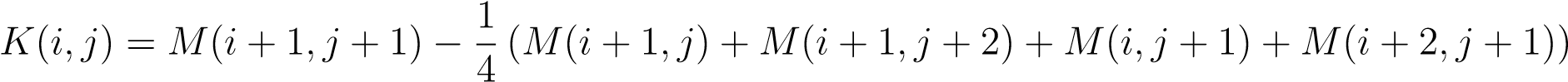
subplot(2,2,1); imshow(img); title('Original');

subplot(2,2,2); imshow(img\_dark\_lr); title('Darkened from left and right');

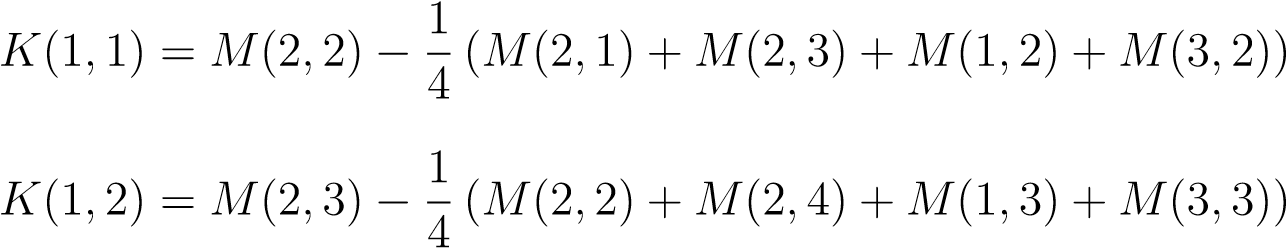
subplot(2,2,3); imshow(img\_dark\_ud); title('Darkened from up and down');

subplot(2,2,4); imshow(img\_dark\_all); title('Darkened from all directions');

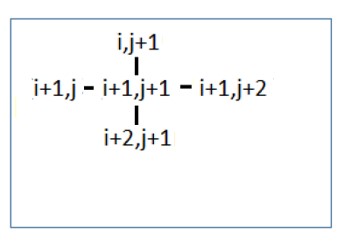
1. Find edges from the picture i.e from *m* × *n*-matrix *M* as follows: create (*m* − 2) × (*n* − 2)-matrix *K*, whose elements are the differences of elements of *M* and the averages of their four neigbours, i.e



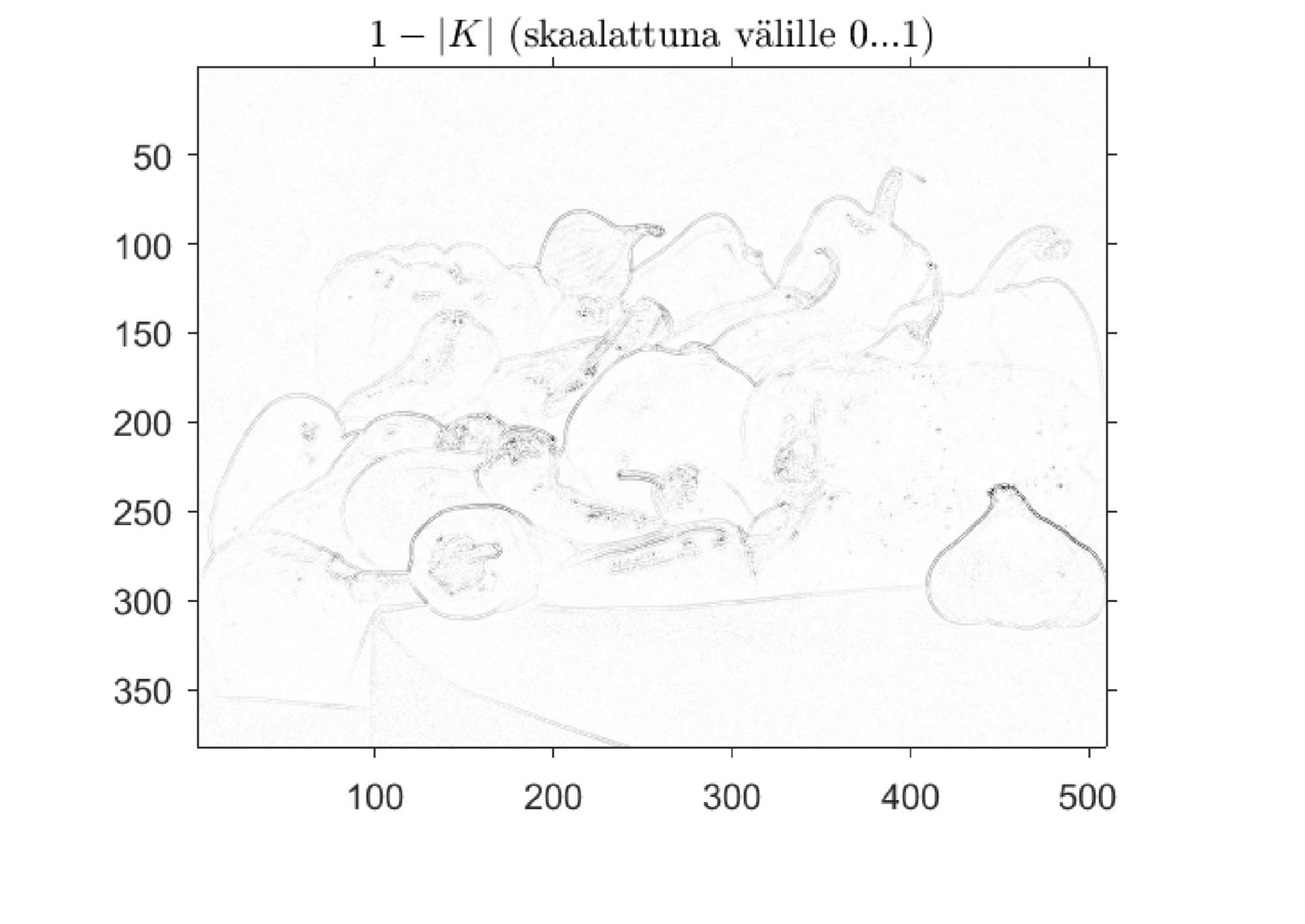
*i* = 1*,*2*,...,m* − 2*, j* = 1*,*2*,...,n* − 2 i.e



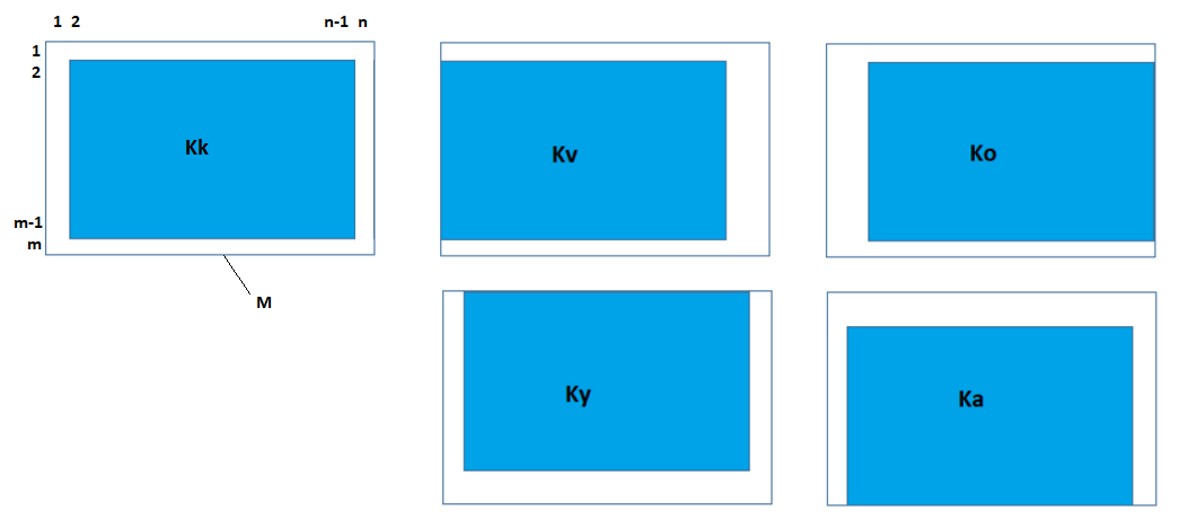
jne



Scale the absolute values |*K*| between 0*...*1 and draw a picture of matrix 1 − |*K*|



hint: use the (*m* − 2) × (*n* − 2)-matrices Kk, Kv, Ko, Ky, Ka



Answer:

% Read image

I = imread('kuva.png');

% Convert to grayscale

I = rgb2gray(I);

% Calculate K matrix

K = zeros(size(I)-2);

for i=1:size(K,1)

for j=1:size(K,2)

K(i,j) = I(i+1,j+1) - mean([I(i+1,j), I(i+1,j+2), I(i,j+1), I(i+2,j+1)]);

end

end

% Scale absolute values of K between 0 and 1

K\_scaled = abs(K) / max(max(abs(K)));

% Invert scaled K to obtain edge detection

edge\_detection = 1 - K\_scaled;

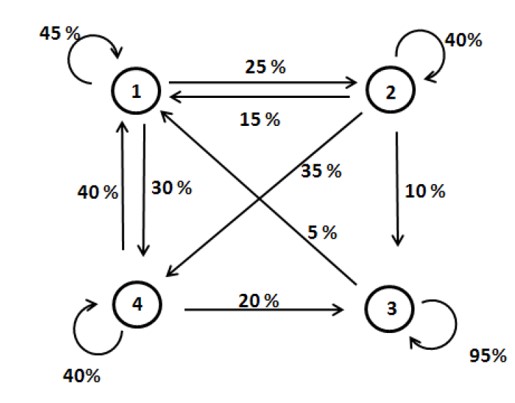
% Display results

subplot(1,2,1), imshow(I), title('Original Image');

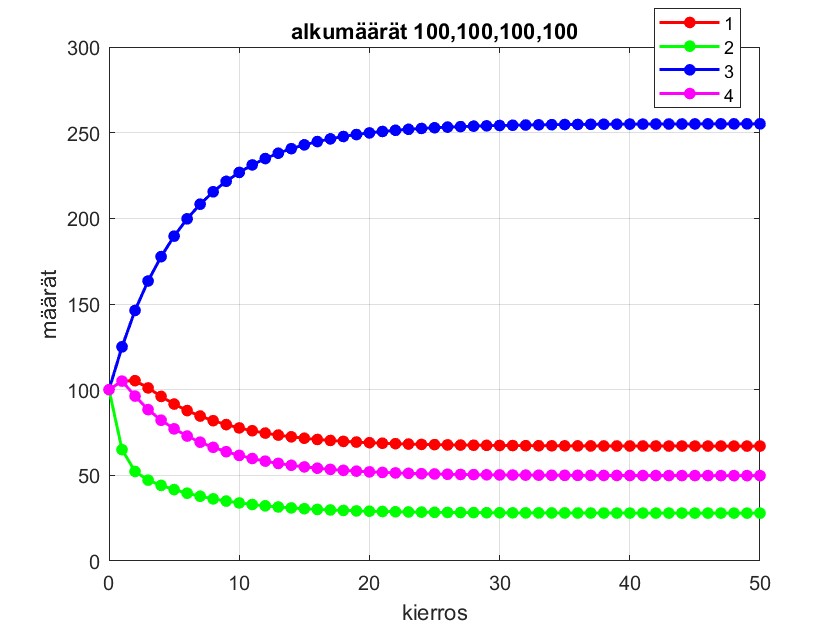
subplot(1,2,2), imshow(edge\_detection), title('Edge Detection');

Make sure the kuva.png image file is in your current working directory.

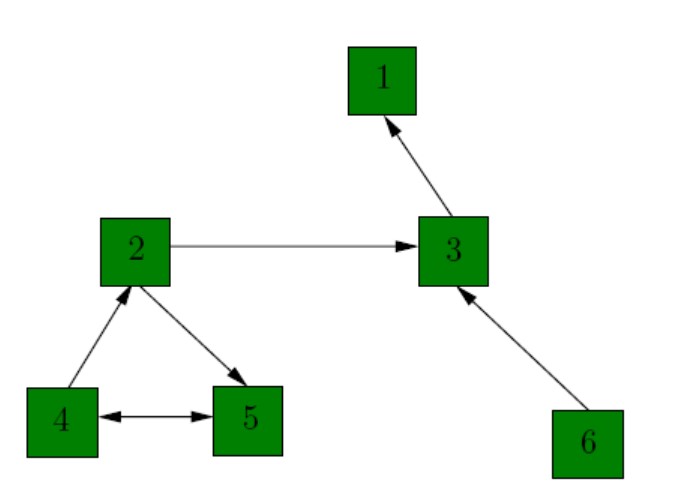
1. Given the initial amounts, calculate the amounts during 50 rounds

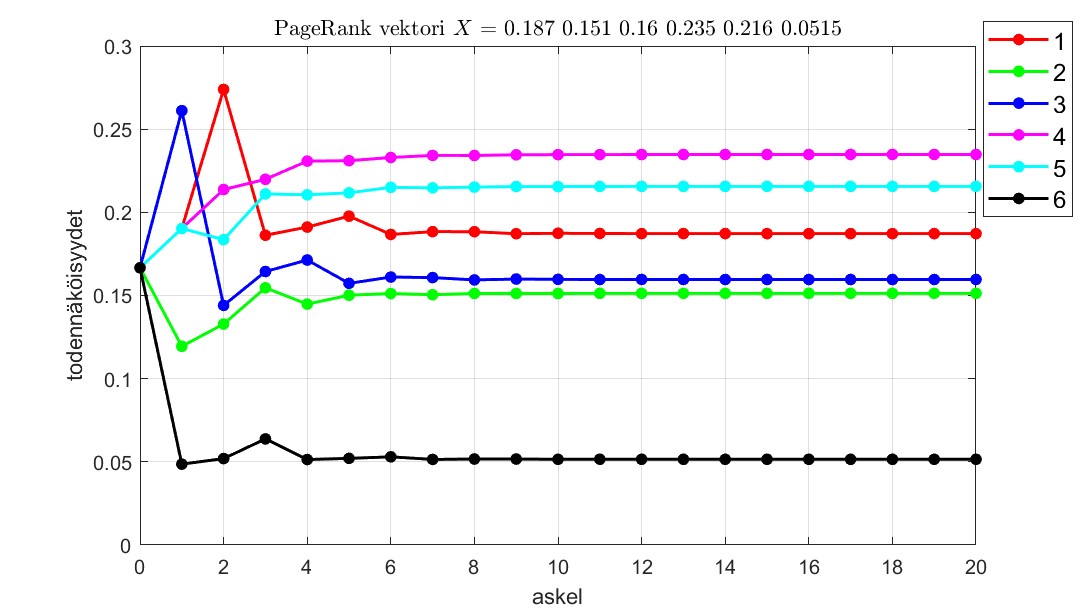


and draw a picture like below



1. Create the PageRank-matrix of the following web and calculate the probabilities for surfer being in pages 1-6 during 20 steps, when initial probabilities are 1 / 6 Draw a picture like below



ja piirra¨ allaolevan nak¨ oinen¨ kuva

1. Given points [*x*1*,y*1], [*x*2*,y*2], [*x*3*,y*3] and [*x*4*,y*4], find the coefficients *a* − *d* of the third degree polynomial

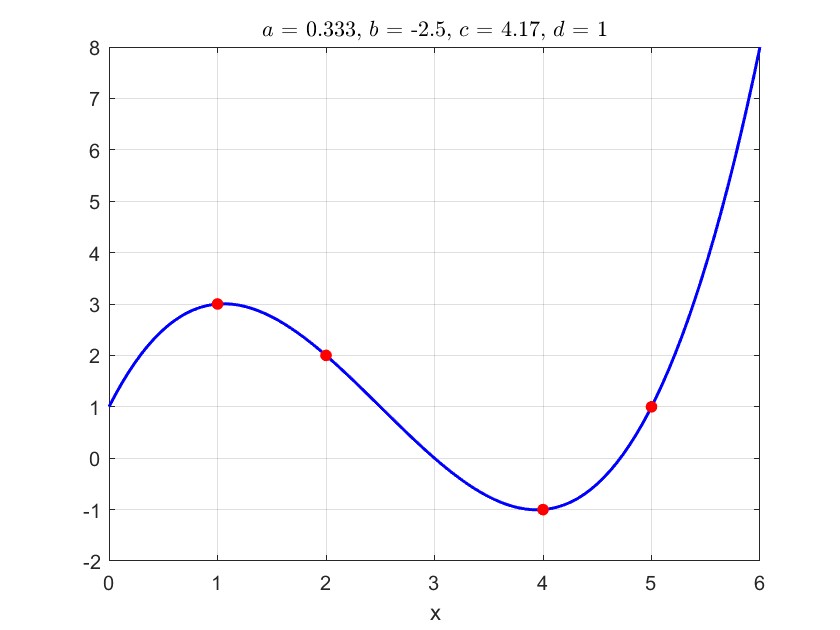
*y* = *ax*3 + *bx*2 + *cx* + *d*

through the points by solving the system of equations

1  *ax ax*332 ++ *bxbx*21 22 ++ *cxcx*12 ++ *dd* == *yy*1 2

*ax*33 + *bx*23 + *cx* 3 + *d* = *y*3  *ax*34 + *bx*24 + *cx*4 + *d* = *y*4

and draw a picture like below



1. Given points *P*1*,P*2*,P*3 and *P*4, find the center *X* of the sphere through them by solving the linear system *AX* = *B*,

where

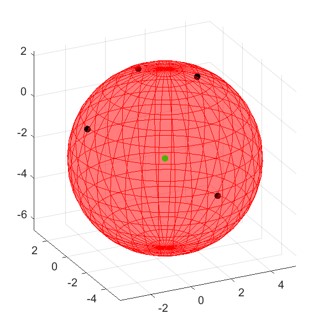
 *P*4 − *P*1 

*A* = 2 *P*4 − *P*2 *, X*

*P*4 − *P*3

 *x*0   ∥*P*4∥2 − ∥*P*1∥2 

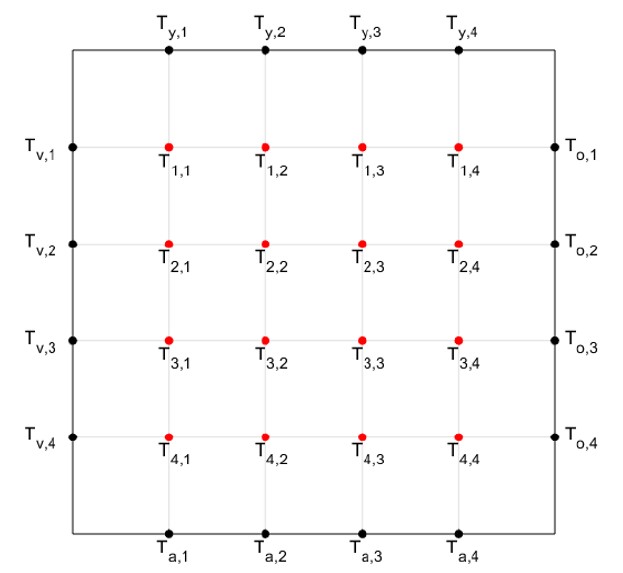
=  *y*0 *, B* =  ∥*P*4∥2 − ∥*P*2∥2  *z*0 ∥*P*4∥2 − ∥*P*3∥2

∥*Pk*∥2 = *x*2*k* + *yk*2 + *zk*2*, k* = 1*,*2*,*3*,*4 and draw a picture like below

1. Given the temperatures

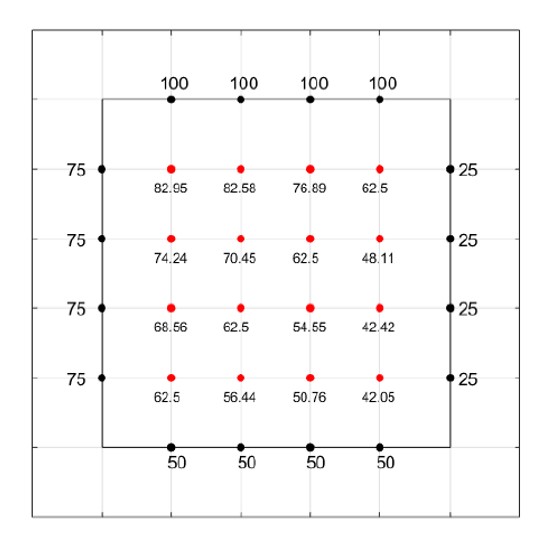
*Ty* = [*Ty*1*,Ty*2*,Ty*3*,Ty*4]*, Tv* = [*Tv*1*,Tv*2*,Tv*3*,Tv*4]

*To* = [*To*1*,To*2*,To*3*,To*4]*, Ta* = [*Ta*1*,Ta*2*,Ta*3*,Ta*4]

of the edges, solve the temperatures *T*11−*T*44 from the system

*AT* = *B*

and draw a picture like below



1. Calculate the temperatures *T*11−*T*44 of ex. 7 directly using the average principle as follows: Create a matrix

 *Tv,y Ty,*1 *Ty,*2 *Ty,*3 *Ty,*4 *To,y* 

 *Tv,*1 0 0 0 0 *To,*1 

  *Tv,*2 0 0 0 0 *To,*2 

*Tka* =  *Tv,*3 0 0 0 0 *To,*3 

 *Tv,*4 0 0 0 0 *To,*4 

*Tv,a Ta,*1 *Ta,*2 *Ta,*3 *Ta,*4 *To,a*

i.e initially the temperatures are = 0, corner-values can be anything

Update the temperatures using the average principle starting from *T*11:

for *r* = 2 : 5 for *s* = 2 : 5

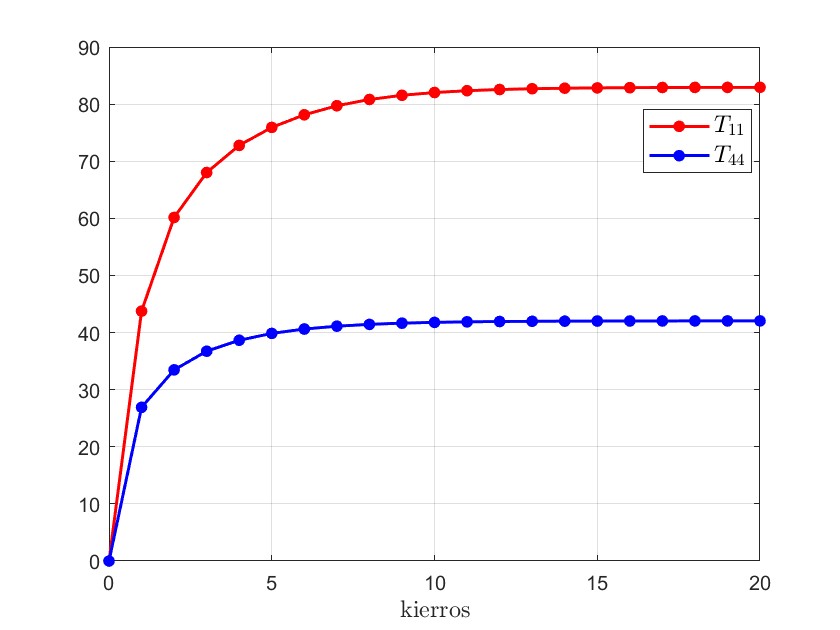
*Tka*(*r,s*)=average of the temperartures of the four neighbours end end

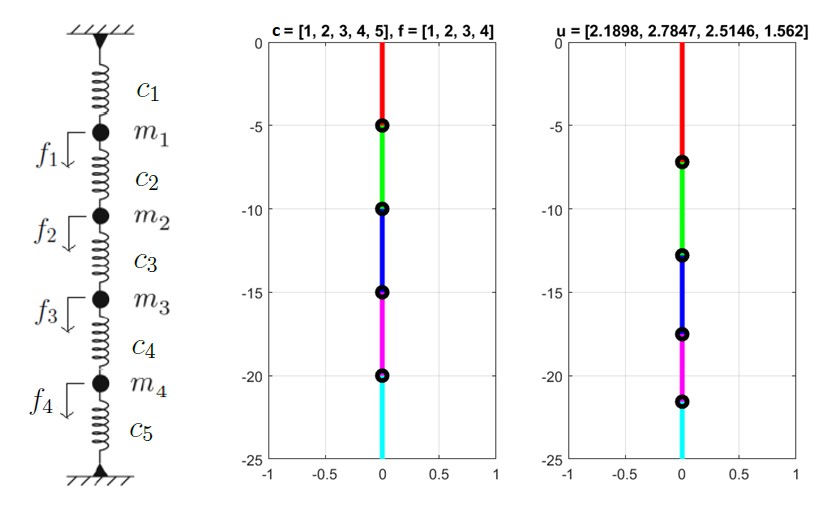
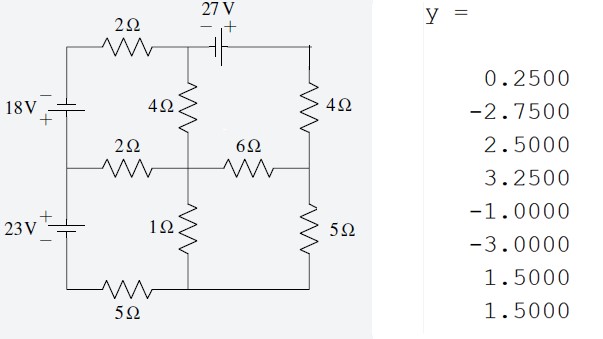
Repeating the updating process several times, values of the temperatures approach the exact values of ex.7

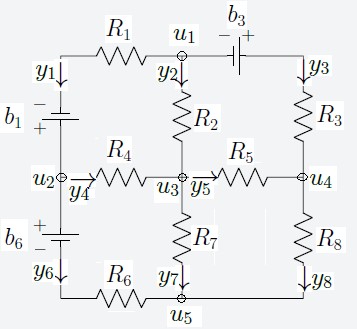
Convince yourself of this by collecting the values of *T*11 and

*T*44

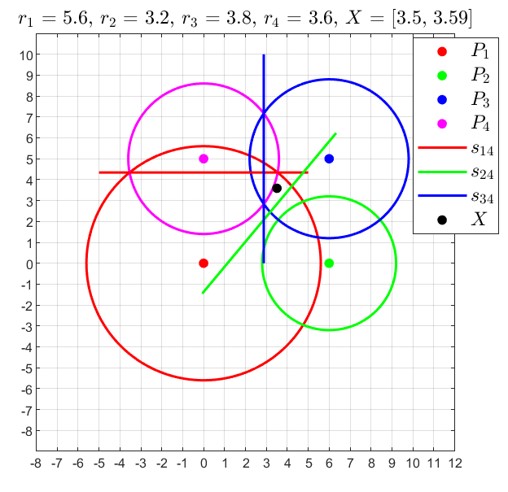
and draw a picture like below



1. Given spring constants *c*1 − *c*5 and external forces *f*1 − *f*4, calculate the displacements *u*1 − *u*4 of the masses and draw a picture like below
2. Given resistances *R*1 − *R*8 and battery voltages *b*1*,b*3*,b*6, calculate the currents *y*1 − *y*8



1. Given 2D-base stations *P*1*,P*2*,P*3*,P*4 and measured distances *r*1*,r*2*,r*3*,r*4 to a point *P*, calculate an estimate *X* for the coordinates of *P* using method 1 and draw a picture like below:



1. Given 2D-base stations *P*1*,P*2*,P*3*,P*4 and measured distances *r*1*,r*2*,r*3*,r*4 to a point *P*, calculate an estimate *X* for the coordinates of *P* using method 2 and draw a picture like below:

