**Numeric – ex2**

**Name: Danielle Tabashi**

**ID: 305619835**

1. Code:

function [L,U,P,Q] = mylu(A,pivot)

[n,n]= size(A);

L=eye(n);

H = zeros(n,n,n);

U = zeros(n,n);

M = zeros(n,n);

P = eye(n);

Q = eye(n);

H(:,:,1)=A(:,:);

% k iteration

for k = 1:1:n

%partial pivot implementation

if(pivot==1)

[h m0]=max(abs(H(k:n,k,k)));

m0=m0+k-1;

temp=H(m0,:,k);

H(m0,:,k)=H(k,:,k);

H(k,:,k)=temp;

temp=P(k,:);

P(k,:)=P(m0,:);

P(m0,:)=temp;

end

%complete pivot

if(pivot==2)

[cv ri]=max(abs(H(k:n,k:n,k)));

[rv ci]=max(cv);

rp=ri(ci)+k-1;

cp=ci+k-1;

i0=rp;

m0=cp;

temp=H(k,:,k);

H(k,:,k)=H(i0,:,k);

H(i0,:,k)=temp;

temp=H(:,m0,k);

H(:,m0,k)=H(:,k,k);

H(:,k,k)=temp;

temp=P(i0,:);

P(i0,:)=P(k,:);

P(k,:)=temp;

temp=Q(:,m0);

Q(:,m0)=Q(:,k);

Q(:,k)=temp;

end

H(:,:,k+1)=H(:,:,k);

%if h(k,k,k)==0 we replace row k with another row...

if(H(k,k,k)==0)

for temp=k+1:1:n

if (H(temp,k,k)>0)||(H(temp,k,k)<0)

%replace row temp and row k

tempA=H(k,:,k);

H(k,:,k)=H(temp,:,k);

H(temp,:,k)=tempA;

end

end

end

for i = k+1:1:n

M(i,k) = H(i,k,k)/H(k,k,k);

L(i,k) = M(i,k);

%update the i'th row according M

for tempColumn = 1:1:n

H(i,tempColumn,k+1)=H(i,tempColumn,k)-(M(i,k)\*H(k,tempColumn,k));

end

end

end

% set U

U(:,:)=H(:,:,n);

end

1. Code:

function [D,V]=myeig(A)

[n,n]=size(A);

V=eye(n);

D=A;

epsilon=(10^-12)\*norm(A,'fro');

while abs(off(D))>epsilon

[p,q]=getMaxIndex(D);

if p>q

temp = q;

q=p;

p=temp;

end

tau=(D(q,q)-D(p,p))/(2\*D(p,q));

if(tau>0)

selectedt=(-1)\*tau+sqrt(tau\*tau+1);

else

selectedt=(-1)\*tau-sqrt(tau\*tau+1);

end

c=sqrt(1/(1+selectedt\*selectedt));

s=selectedt\*c;

Q = eye(n);

Q(p,p)=c;

Q(q,q)=c;

Q(p,q)=s;

Q(q,p)=(-1)\*s;

D=transpose(Q)\*D\*Q;

V=V\*Q;

end

end

function [T] = updateColumnsInD(D,p,q,s,c)

[n,n]=size(D);

T=D;

for i=1:n

T(i,p)=T(i,p)\*c-T(i,q)\*s;

T(i,q)=T(i,p)\*s+T(i,q)\*c;

end

end

function [T] = updateRowsInD(D,p,q,s,c)

[n,n]=size(D);

T=D;

for i=1:n

T(p,i)=T(p,i)\*c-T(q,i)\*s;

T(q,i)=T(p,i)\*s+T(q,i)\*c;

end

end

function [p,q]=getMaxIndex(D)

[n,n]=size(D);

maxVal=0;

for i=1:n

for j=1:n

if(i~=j)

if(abs(D(i,j))>maxVal)

maxVal=abs(D(i,j));

p=i;

q=j;

end

end

end

end

end

function [sum]=off(A)

[n,n]=size(A);

sum=0;

for i=1:n

for j=1:n

if(i~=j)

sum=sum+A(i,j)^2;

end

end

end

end

1. לא הצלחתי...

אני מצליחה להגיע ל5 משוואות, אבל לא הצלחתי להתקדם משם.

4.

האלגוריתם התכנס אחרי 12942 איטרציות לy המצורף. דרושות המון איטרציות בגלל קמירות הפונקציה. תנאי העצירה של האלגוריתם שלי הוא המרחק בין xk לבין xk+1 משום שאם לא התרחקנו הרבה, סימן שאנחנו ממש קרובים למינימום.

Code:

function [y,iter]=myGD(f,gradf,x0,a,tol,maxiter)

currx=x0;

prevx=x0+2\*tol; %in order to get into the while loop.

iter=0;

while (abs(currx-prevx)>tol)

prevx=currx;

currx=prevx-a\*feval(gradf,prevx);

iter=iter+1;

if iter>=maxiter

break;

end

end

y=currx;

end

y =

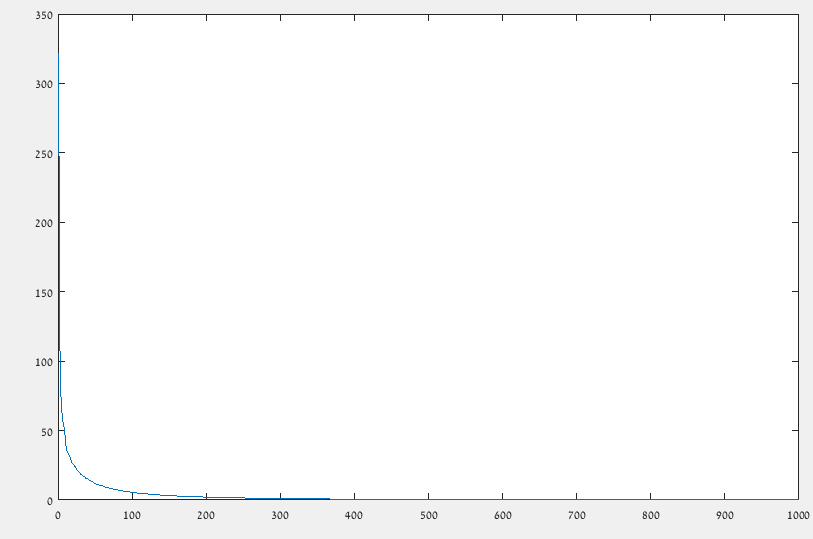
1.0025

1.0050

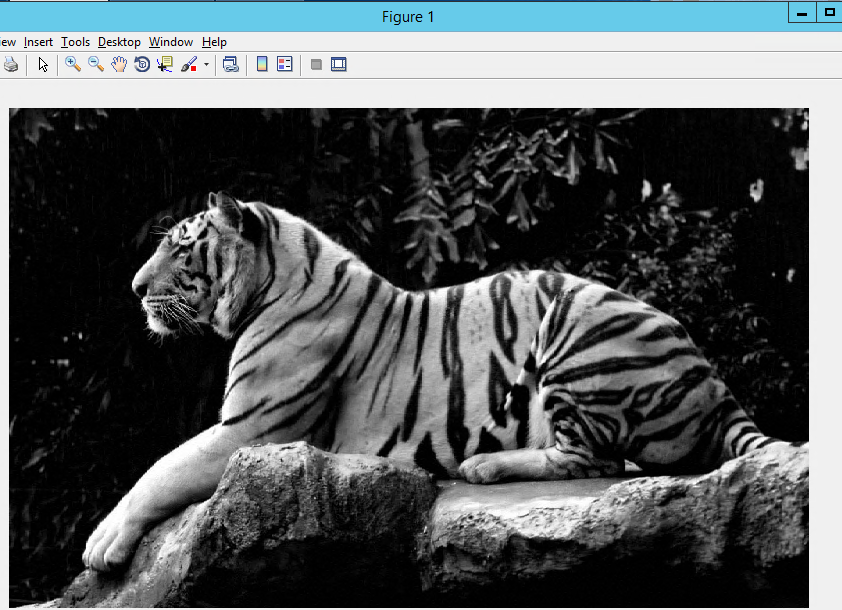
iter =

12942

5.

א. לפי הגרף מספיק לקחת בערך 250 ערכים סינגולרים כי האחרים מאוד קרובים לאפס ולכן בכל מקרה יאופסו בגלל שגיאות עיגול ואיבוד ספרות.

ג. השתמשתי בפונקציה Krank עם k=250. מצורפת התמונה המתקבלת:



מצורף הקוד של השאלה:

tiger=double(imread('Tiger.jpg'));

bwtiger = (tiger(:,:,1)+tiger(:,:,1)+tiger(:,:,1))/3;

bwtiger = bwtiger/max(bwtiger(:));

%imshow(bwtiger);

%plot(svd(bwtiger));

[Uk, Sk, Vk] = Krank(bwtiger,250);

newtiger = Uk\*Sk\*Vk';

imwrite(newtiger,'newTiger.jpg');

function [Uk, Sk, Vk] = Krank(A,k)

[U,S,V]=svd(A, 'econ');

Uk=U(:,1:k);

Vk=V(:,1:k);

Sk=S(1:k,1:k);

end