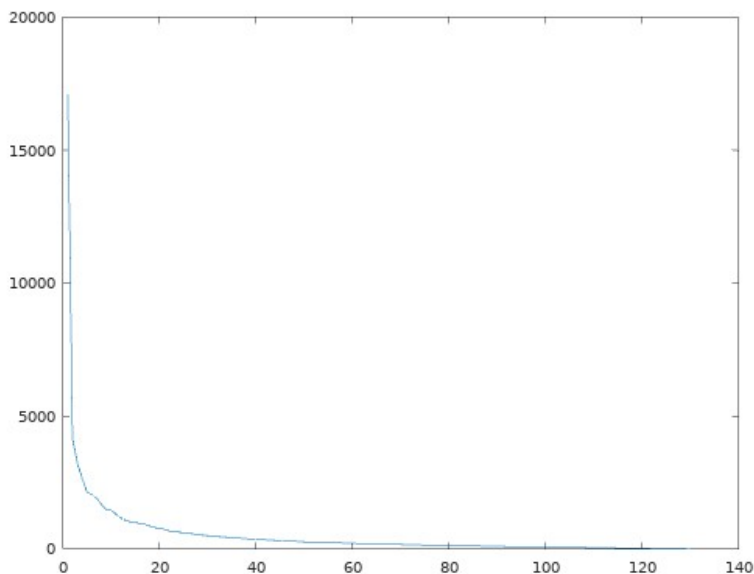


Cerinta 1

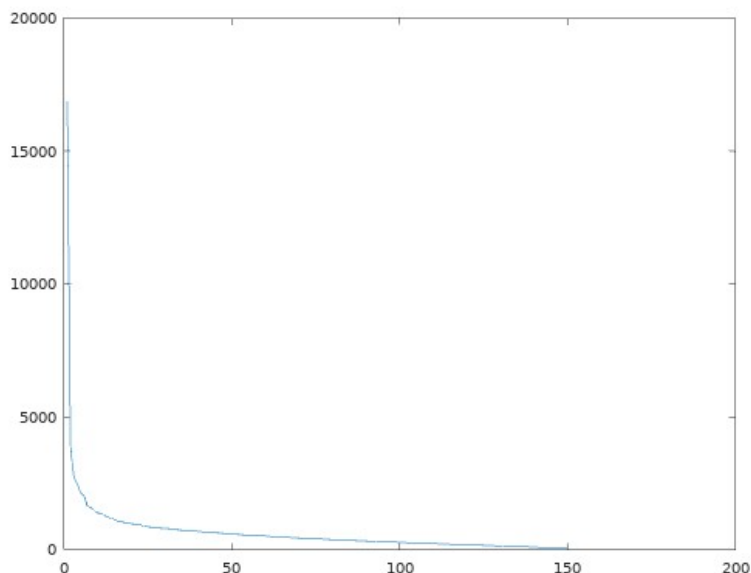
– pentru aceasta cerinta am completat functia „function A_k = task1 (image, k)”, unde image este calea catre imagine si k numarul de valori singulare;
- am citit matricea imaginii, am salvat in 2 variabile dimensiunile matricei, am aplicat SVD si am salvat valorile intoarse de functie in variabilele initiale, apoi am eliminat portiunile din cele 3 matric, in final am construit matricea dupa formula de la compresia imaginilor;

Cerinta 2

- pentru aceasta cerinta am completat functia „task2(image)”, antetul fiind ales de mine, unde image este calea catre imagine;
- k, numărul de valori singulare ia $\min(m,n)$, unde m și n sunt dimensiunile matricei;
-am folosit subplot pentru a afisa cele 4 grafice în același timp;
1. reprezentarea grafica a tuturor valorilor singulare
- am aplicat SVD, determinand doar valorile singulare, apoi folosesc functia „plot” pentru reprezentare;

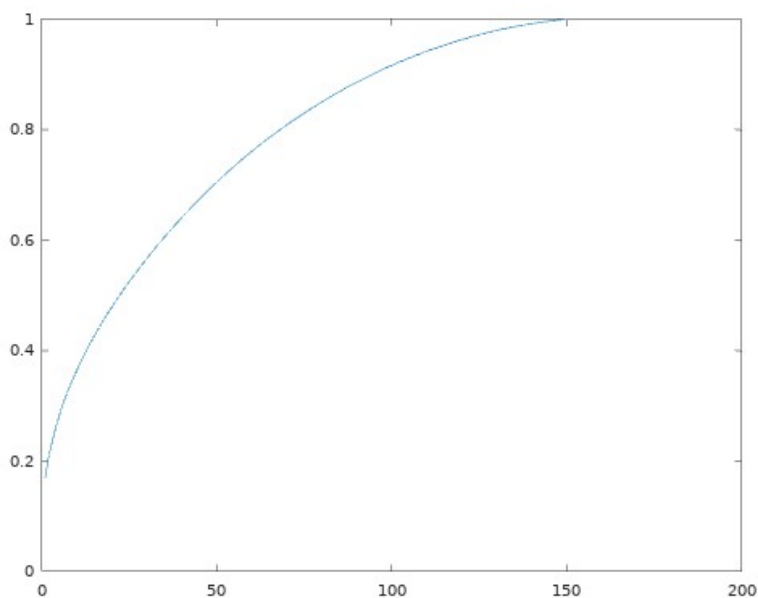


grafic optinut pentru apelarea
„task2("in/images/image3.gif")”

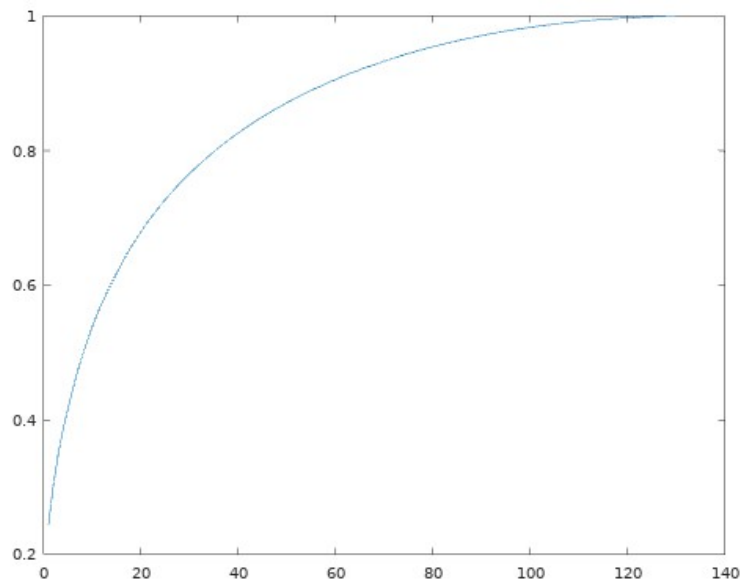


grafic optinut pentru apelarea
„task2("in/images/image2.gif")”

2. reprezentarea grafica a lui k si a informatiei lui A calculata dupa formula data
- aplic SVD, determinand U , S , V , calculez numitorul in functie de formula data si pentru fiecare numarator calculat in cel de al doilea for adaug rezultatul impartirii celor 2 in vectorul „info”.
Folosesc functia „plot” pentru reprezentare;



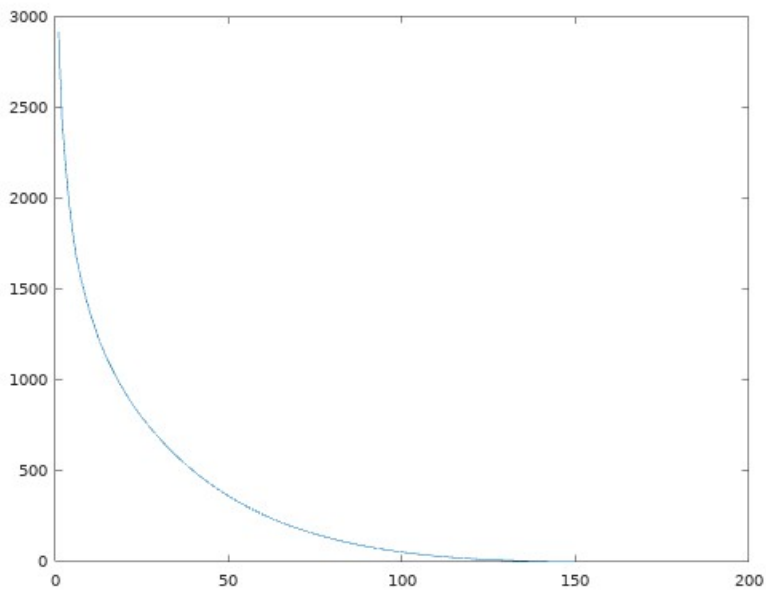
grafic optinut pentru apelarea
„task2("in/images/image2.gif")”



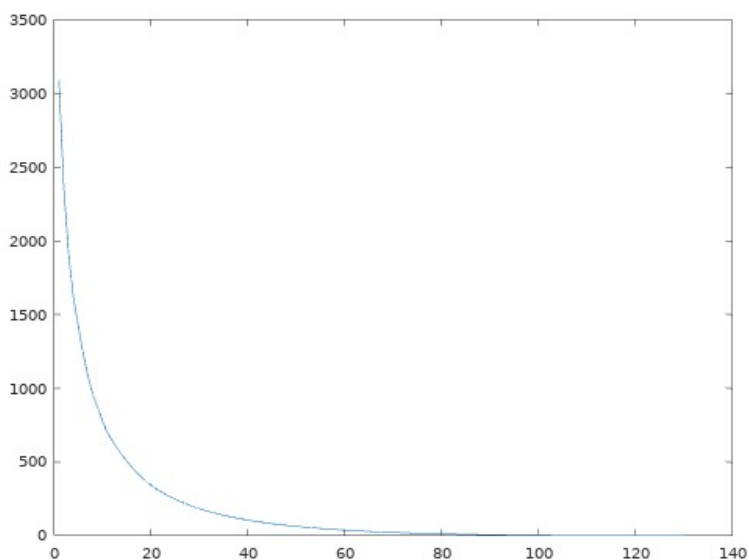
grafic optinut pentru apelarea
„task2("in/images/image3.gif")”

3. reprezentarea grafica a lui k și a erorii aproximative

- precizez ca acest grafic se încarcă mai greu ca restul;
- într-un for cu i de la 1:k, apelez funcția creată la task1 pentru fiecare i pentru a crea matricea A_k , apoi calculez suma de la numitor și pentru fiecare i calculez eroarea aproximativă după formula dată și introduc în vectorul „err”. Folosesc funcția „plot” pentru reprezentare.



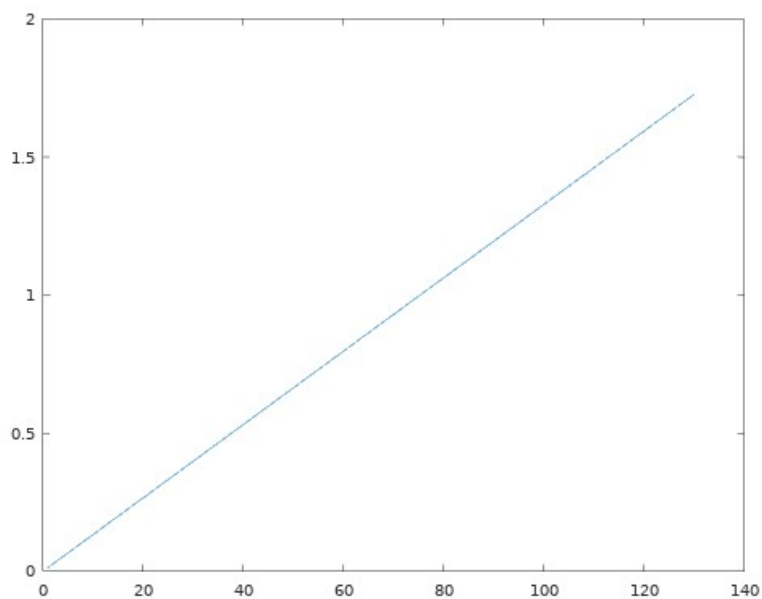
grafic optinut pentru apelarea
„task2("in/images/image2.gif")”



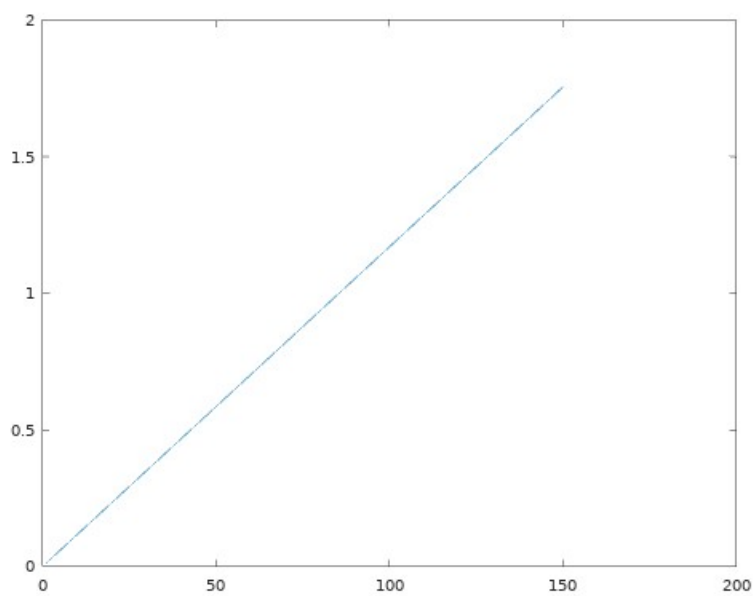
grafic optinut pentru apelarea
„task2("in/images/image3.gif")”

4. reprezentarea grafica a lui k si a ratei de compresie dupa formula data

- intr- un for cu i de la 1:k calculez rata de compresie pentru fiecare i după formula data și folosesc „plot” pentru reprezentare.



grafic optinut pentru apelarea
„task2("in/images/image3.gif")”



grafic optinut pentru apelarea
„task2("in/images/image2.gif")”

Cerinta 3

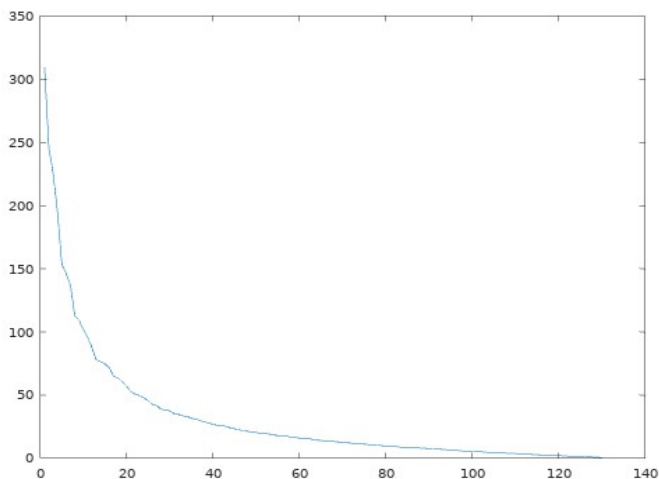
- pentru aceasta cerinta am completat functia „function [A_k S] = task3(image, k)”, unde image este calea catre imagine si k numarul de valori singulare;
- am citit matricea imaginii, am salvat în 2 variabile dimensiunile matricei, am calculat media pe fiecare linie a matricei și am pastrat-o în vectorul „medie”;
- am calculat eroarea și am introdus-o în matrice;
- am calculat SVD pentru Z, Z- ul fiind calculat după formula data;
- am determinat spatiul k-dimensional al componentelor principale;
- am calculat proiectia matricei imagine și am aproximat-o după formulele date.

Cerinta 4

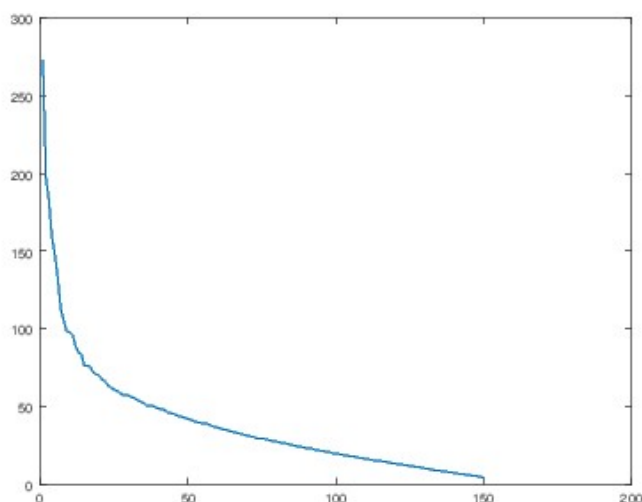
- pentru aceasta cerinta am completat functia „function [A_k S] = task4(image, k)”, unde image este calea catre imagine si k numarul de valori singulare;
- abortarea este similara cu cea de la cerinta 3 singura diferența este ca în loc de aplicarea; algoritmului SVD am folosit functia „eig”, în determinarea spatiul k-dimensional al componentelor principale am ținut cont ca functia „eig” întoarce valorile în ordine descrescatoare.

Cerinta 5

- pentru aceasta cerinta am completat functia „task5(image)”, antetul fiind ales de mine, unde image este calea catre imagine;
 - apelez [A_k S] = task3(image, k);
1. reprezentare grafica a vectorul diag(S);



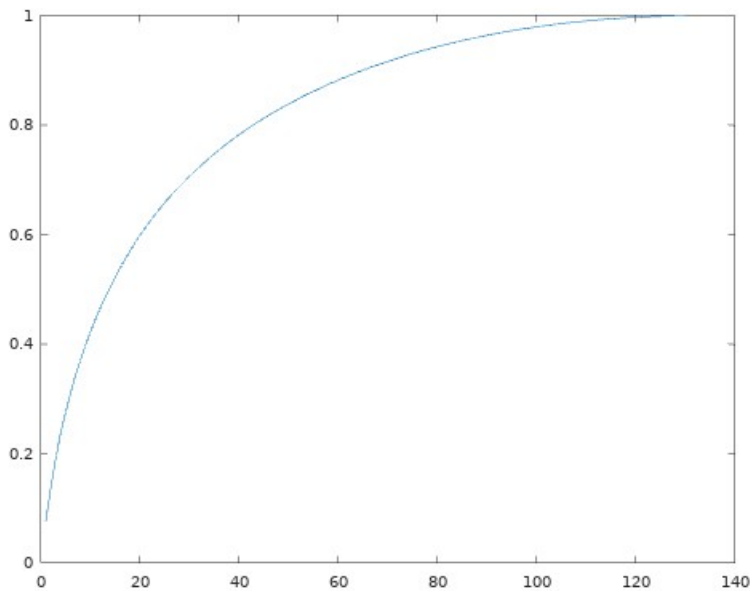
grafic optinut pentru apelarea
„task5("in/images/image3.gif")”



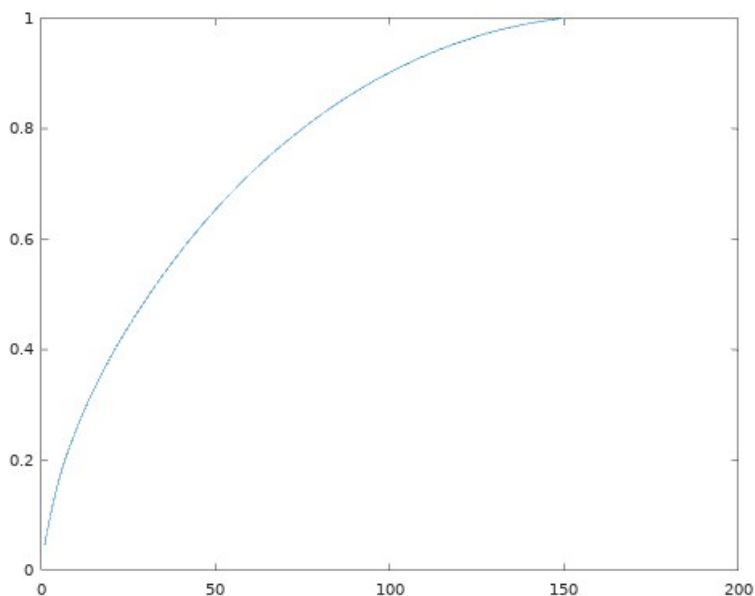
grafic optinut pentru apelarea
„task5("in/images/image2.gif")”

2. reprezentarea grafica a lui k si a informatiei lui Z calculata dupa formula data

- am citit matricea imaginii, am salvat in 2 variabile dimensiunile matricei, am calculat media pe fiecare linie a matricei și am pastrat-o în vectorul „medie”;
- am calculat eroarea și am introdus-o în matrice;
- am calculat SVD pentru Z , Z - ul fiind calculat după formula data;
- calculez numitorul in functie de formula data si pentru fiecare numarator calculat in cel de al doilea for adaug rezultatul impartirii celor 2 in vectorul „info”. Folosesc functia „plot” pentru reprezentare;



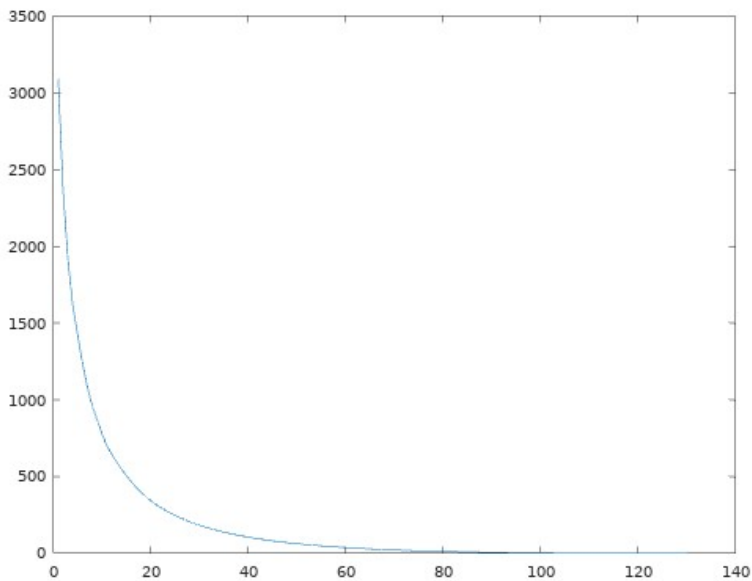
grafic optinut pentru apelarea
„task5("in/images/image3.gif")”



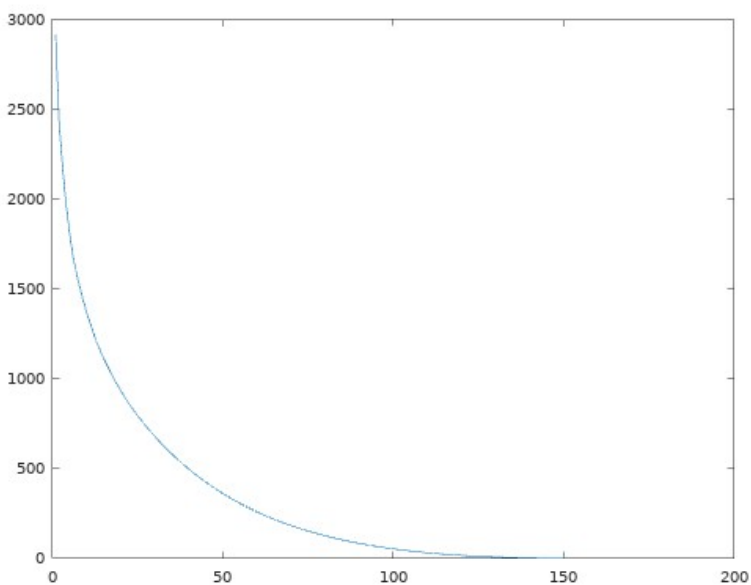
grafic optinut pentru apelarea
„task5("in/images/image2.gif")”

3. reprezentarea grafica a lui k și a erorii aproximative

- precizez ca acest grafic se încarcă mai greu ca restul;
- într-un for cu i de la 1: k , apelez funcția creată la task1 pentru fiecare i pentru a crea matricea A_k , apoi calculez suma de la numitor și pentru fiecare i calculez eroarea aproximativă după formula dată și introduc în vectorul „err”. Folosesc funcția „plot” pentru reprezentare.



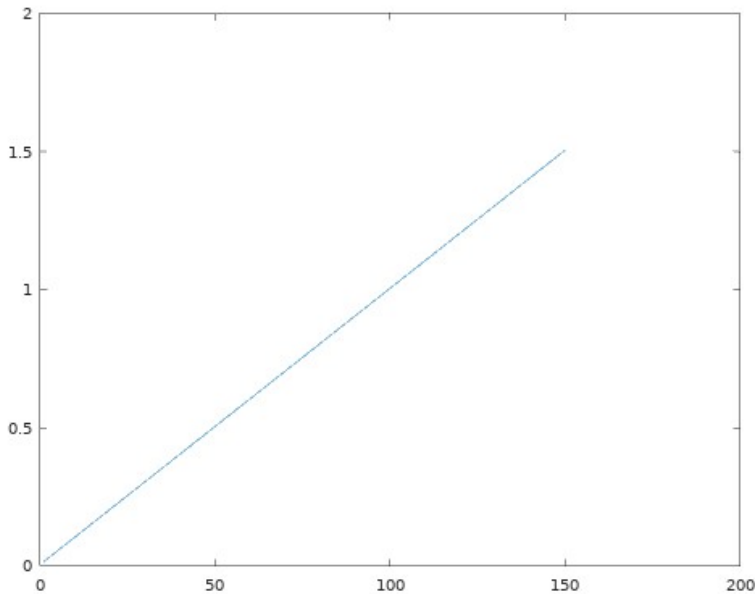
grafic optinut pentru apelarea
„task5("in/images/image3.gif")”



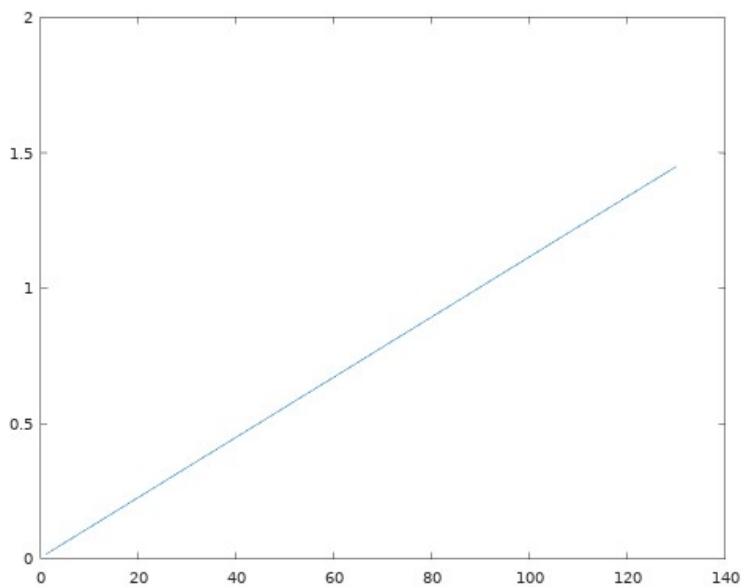
grafic optinut pentru apelarea
„task5("in/images/image2.gif")”

4. reprezentarea grafica a lui k si a ratei de compresie a datelor dupa formula data

- intr- un for cu i de la $1:k$ calculez rata de compresie pentru fiecare i după formula data și folosesc „plot” pentru reprezentare.



grafic optinut pentru apelarea
„task5("in/images/image2.gif")”



grafic optinut pentru apelarea
„task5("in/images/image3.gif")”

Cerinta 6

- pentru prima parte a cerintei am completat functia „function [m A eigenv aces pr_img] = eigenf ace_core(database_path)” ;
- am creat un string folosind sprintf pentru a crea caile către fiecare imagine;
- pentru fiecare j citesc matricea imaginii, apoi o transform într-un vector coloana;
- fiecare vector coloana îl introduc în matricea T pe coloana j ;
- calculez media aritmetica a fiecărei linii din coloana T ;
- calculez matricea folosindu-ma de formula data în cerinta;
- aplic functia „eig” pentru a avea matricea vectorilor proprii și matricea ce conține valorile proprii;

- pentru fiecare valoare proprie mai mica ca 1 găsită în matricea lambda elimin linia corespunzătoare din matricea V;
 - calculez „eigenfaces” după formula data;
 - calculez proiectia imaginii;
-
- pentru a doua parte a cerinței am completat funcția „function [min_dist output_img_index] = face_recognition(image_path, m, A, eigenfaces, pr_img)”;
 - am citit matricea imaginii și am transformat-o într-un vector coloană;
 - am calculat vectorul $V = \text{vector coloană} - \text{media}$;
 - calculez proiectia imaginii;
 - determin distanța minimă folosindu-mă de funcția „norm” ce calculează distanța;
 - atunci când am determinat distanța minimă o salvez în min_dist și indexul ei în output_img_index.