Metode Numerice README - TEMA 2

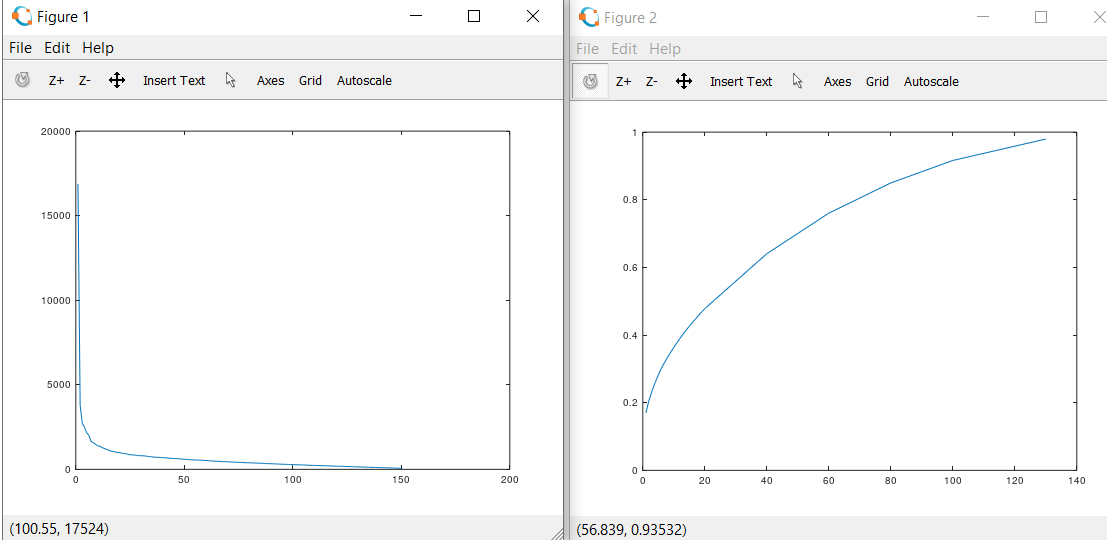
Task 1 - Comprimarea de imagini folosind SVD

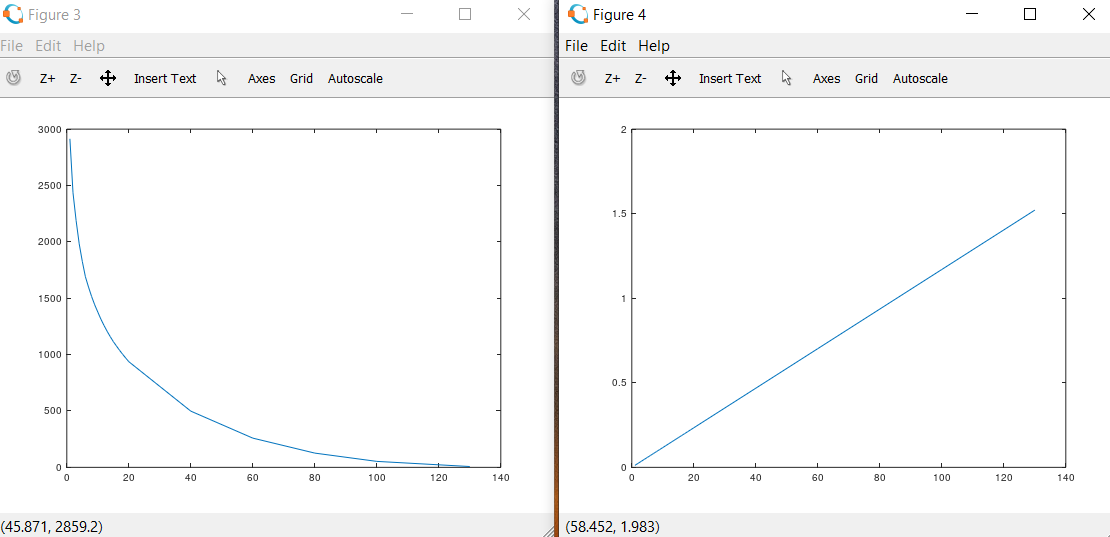
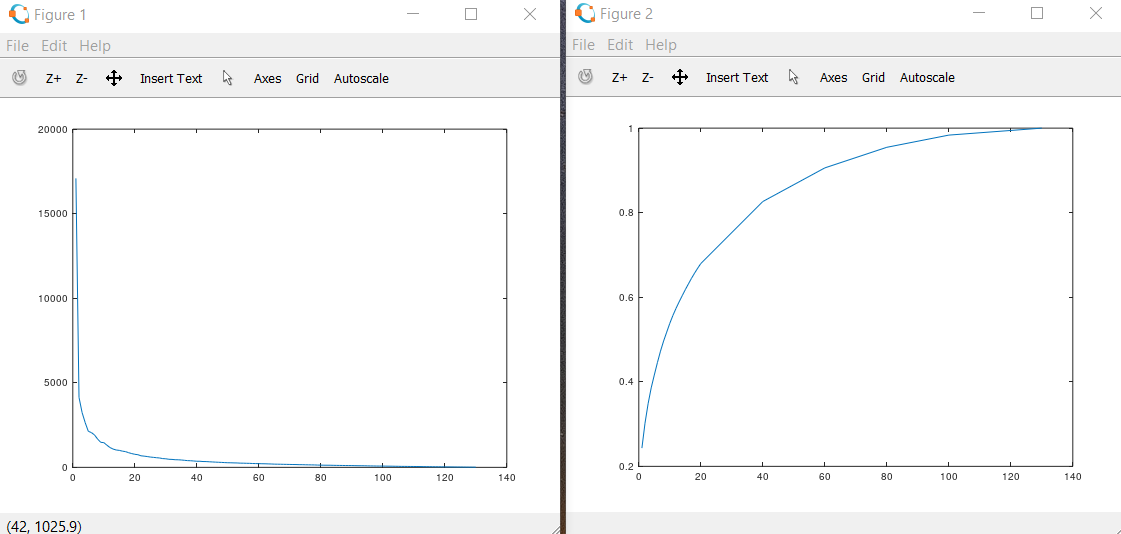
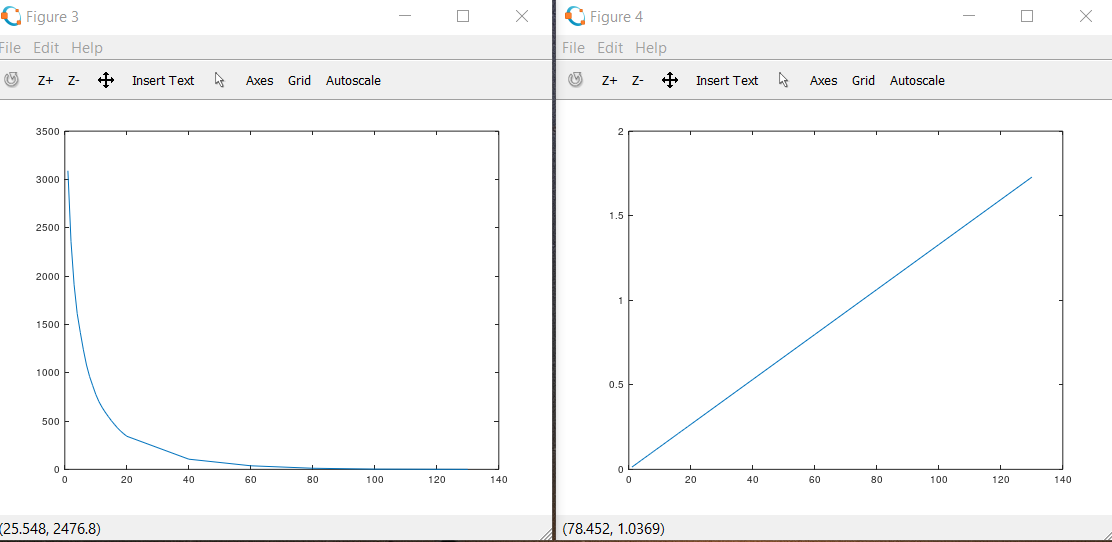
Task-ul 1 presupune compresia imaginilor folosind descumpunerea redusa a valorilor singulare. In implementarea propriu-zisa are loc citirea imaginii ca matrice folosind functia imread. Functia size returneaza dimensiunile matricei, respectiv numarul de linii si de coloane. Folosindu-ma de vectorizari construiesc matricile U\_k, S\_k si V\_k. Produsul lor va avea drept rezultat matricea A\_k care reprezinta o imagine de aceleasi dimensiuni cu matricea initiala.

Task 2

Pentru reprezentarea graficelor de la cerinta 2 m-am folosit de functia svd pentru. Astfel am reprezentat pe rand toate valorile singulare ale matricei citite, informatia data de primele k valori singulare rezultate din descompunerea svd pentru o matrice A, eroarea aproximarii pentru aceeasi matrice si rata de compresie a datelor. Pentru cele 3 cerinte am considerat k luat ca in exemplul dat in cerinta. Aparitia graficelor 3 si 4 dureaza aproximativ 30 de secunde de la apelarea functiei.

Graficele rezultate pentru imaginea 2 sunt urmatoarele:



Graficele rezultate pentru imaginea 3 sunt urmatoarele:

COMPRESIA IMAGINILOR FOLOSIND ANALIZA COMPONENTELOR PRINCIPALE

Task 3

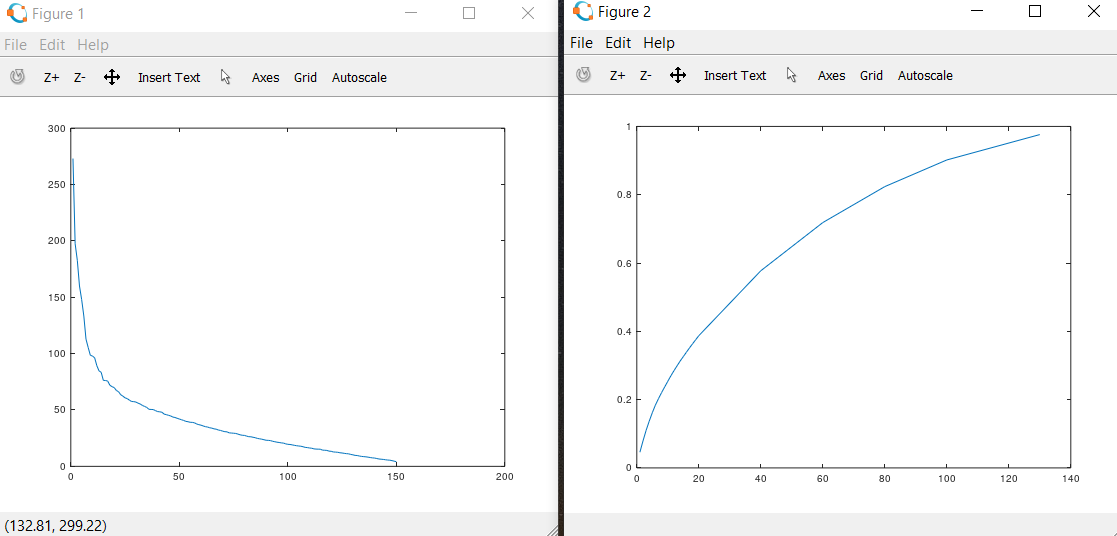
Citesc imaginea folosind functia de la precizari si calculez dimensiunile acestuia cu functia size. Declar vectorul miu in care adaug elemente conform formulei de la punctul 1, apoi actualizez v liniile matricii folosindu-ma de acesta. Calculez matricea Z conform formulei de la punctul 3. Aplic descompunerea valorilor singulare pe matricea Z. Creez matricea W cu toate liniile, respectiv primele k coloane din Z, iar apoi calculez matricea Y conform formulei date. Rezultatul functiei este maatricea A\_k calculata ca produs intre W si Y pe care o actualizez adunand vectorul miu la liniile sale. Precizez ca am modificat antetul functiei task3 pentru a returna si matricea Z intrucat voi avea nevoie de aceasta pentru crearea graficelor de la task5.

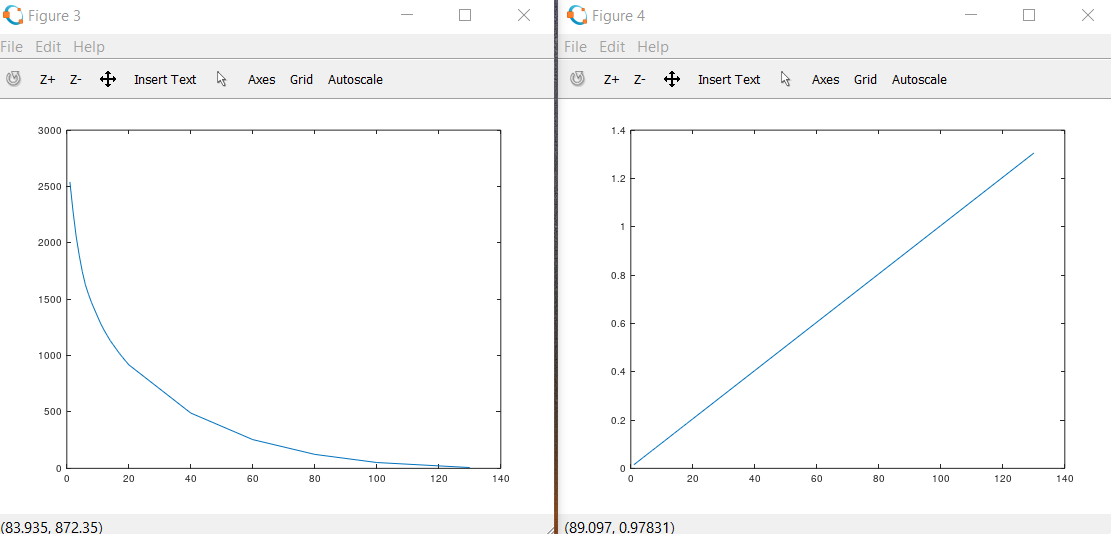
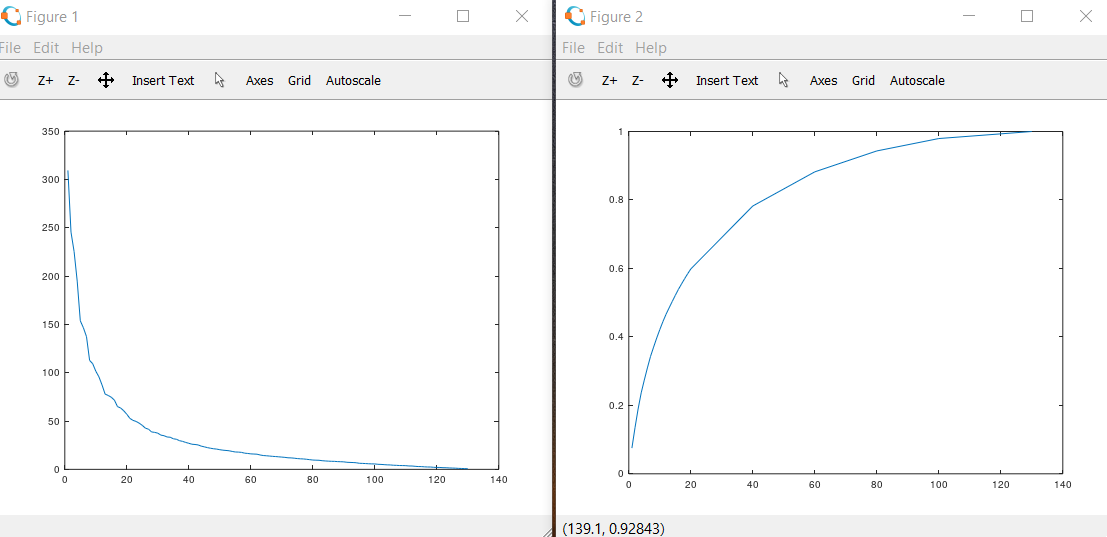
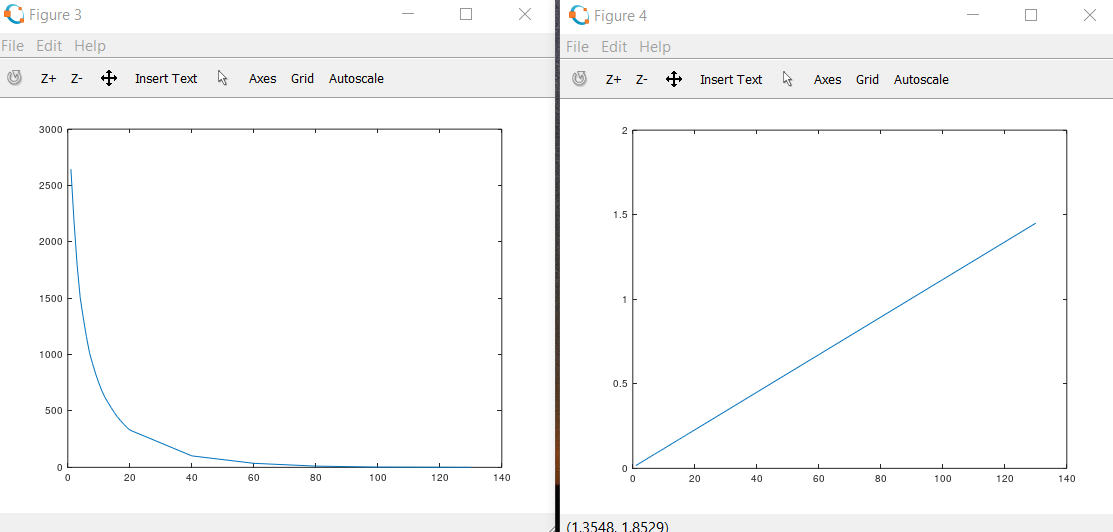
Task 4

Pentru cerinta 4 am urmat primii 2 pasi de la cerinta 3, am calculat matricea Z dupa o formula diferita si anume (A\*A’)/n-1 si am folosit functia eig pentru a calcula valorile si vectorii proprii ai matricei Z. Am definit matricea W folosind liniile matricei V din rezultatul functiei eig si primele k coloane ale acesteia. Ultimii 2 pasi sunt aceeasi ca la cerinta 3 privind calculul matriceai A\_k si actualizarea sa cu elementele vectorului miu.

Task 5

Pentru graficele de la task-ul 5 am apelat functia task3 care returneaza matricile A\_k S si Z necesare in reprezentarea grafica. Astfel am reprezentat pe rand vectorul diag(S), informatia data de primele k valori care rezulta din descompunerea svd a matricei Z, eroarea aproximarii pentru matricea C citita la inceput si rata de compresie a datelor data formula precizata. Am considerat vectorul k luat ca in exemplul din cerinta. Aparitia graficelor 3 si 4 dureaza in jur de 30 de secunde de la apelarea functiei.

Graficele pentru imaginea 2 sunt urmatoarele:

Graficele pentru imaginea 3 sunt urmatoarele:

Task 6 – Recunoasterea faciala

Pentru citirea imaginilor deschid directorul database\_path care la apelare va fi inlocuit cu dataset folosesc functia readdir. Contorul nr\_img reprezinta numarul imaginii citite cu ajutorul functiei date la precizari. Transform apoi matricea in vector coloana parcurgand fiecare linie si adaug acest vector in matricea T. Pentru matricea T calculez media pe fiecare linie si adaug rezultatul in vectorul m. Definesc matricea A ca diferenta dintre matricea T si vectorul, iar inainte de a calcula matricea eigenfaces returnata de functie aflu componentele matricei V. V contine vectorii proprii corespunzatori valorilor proprii mai mari decat 1 ai matricei B = A \* A’. Calzulez proiectia fiecarei imagini cu ajutorul formulei din cerinta. Toate aceastea sunt implementate in functia eigenface\_core.

Pentru cea de-a doua parte citesc matricea M pe care o transform intr-un vector coloana reprezentat de T. Fac din nou diferenta T-m pe care o depun in matricea A0. Vectorul m este luat din antetul functiei fiind acelasi ca la functia anterioara. Calculez proiectia pentru imaginea citita cu formula. Matricea eigenfaces este data ca parametru fiind calculata anterior. Calculez distanta minima dintre proiectia imaginii de test, adica proiectia imaginii citite in functia face\_recognition si prima coloana a matricei pr\_img care contine proiectiilor tuturor imaginilor din fisierul dataset, urmand ca aceasta sa se modifice odata cu calcularea distantei minime dintre pr\_test\_img si celelalte coloane din pr\_img. Functia face\_recognition returneaza indexul proiectiei imaginii pentru care se obtine distanta minima, respectiv distanta cea mai mica.