

Facial Recognition Graph

Vitoga George Patrick

1 Cerinta 2

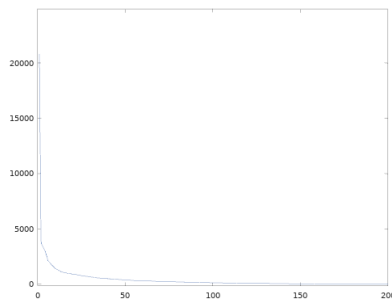
La task-ul 2 am analizat formulele de calcul ale graficelor pentrut a le aduce la cea mai simpla forma .

Pentru toate graficele am folosit array X (axa absciselor) in care se afla numere naturale de la 1 la k .

Pentru fiecare reprezentare grafica am construit un array Y_i in care sa stochez valorile procesate ale functiilor .

Primul grafic :

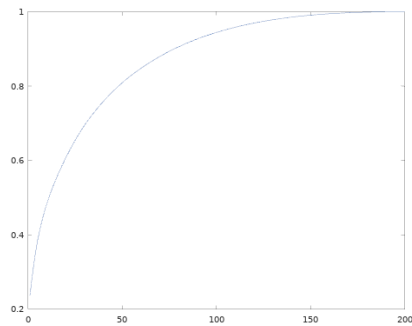
Pentru primul grafic am folosit pur si simplu X si in Y_1 valorile computeate de functia `svd()`, si am calculat si suma elementelor in variabila `sums` pentru taskurile urmatoare .



```
plot(X, Y1);
```

Al doilea grafic :

Pentru al doilea grafic am folosit pe X, iar pe Y_2 l-am populat cu suma calculata recursiv a elementelor S_i obtinute cu funtia `dvs()` si am impartit la variabila `sums` obtinuta anterior.

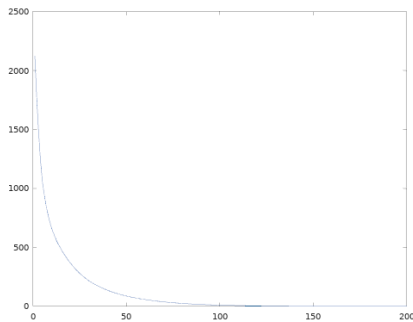


```
plot(X, Y2);
```

Al treilea grafic :

Pentru al treilea grafic am observat ca pot calcula pe Y_3 mai usor, scazand din matricea A_k pe A si ridicand la patrat, urmand ca mai apoi sa fac suma tuturor elementelor din matricea rezultata si sa impart la m si la n .

Procesarea acestui grafic dureaza mai mult din cauza apelurilor succesive a funtiei $A_k = \text{task1}(\text{image}, i)$; necesara popularii lui Y_3 cu "eroarea aproximarii" .

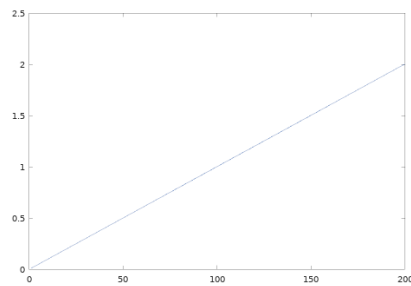


```
plot(X, Y3);
```

Al patrulea grafic :

Pentru ultimul task din cadrul Cerintei 2, am dat factor comun pe k si am observat ca ramane o fractie care depinde numai de m si n (valori statice) .

Asadar, am iterat cu un for valorile lui k si am inmultit fractia cu k .



```
plot(X, Y4);
```

2 Cerinta 5

La task-ul 5, la fel ca la Cerinta 2, am analizat formulele de calcul ale graficelor pentrut a le aduce la cea mai simpla forma .

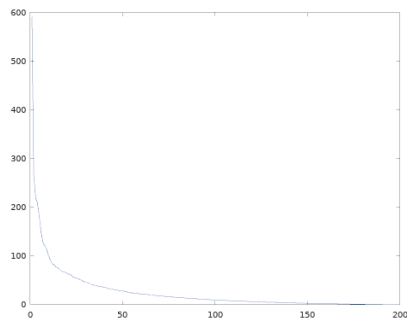
Pentru toate graficele am folosit array X (axa absciselor) in care se afla numere naturale de la 1 la k .

Pentru fiecare reprezentare grafica am construit un array Y_i in care sa stochez

valorile procesate ale functiilor .

Primul grafic :

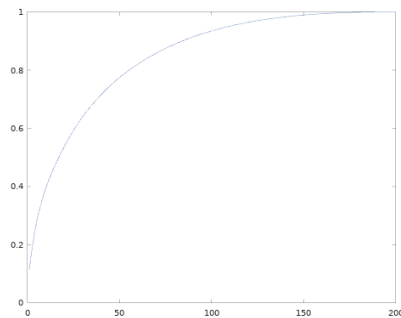
Pentru primul grafic am folosit pur si simplu X si in Y_1 valorile compute de functia $\text{svd}(Z)$, si am calculat si suma elementelor in variabila ssum pentru taskurile urmatoare .



```
plot(X, Y1);
```

Al doilea grafic :

Pentru al doilea grafic am folosit pe X, iar pe Y_2 l-am populat cu suma calculata recursiv a elementelor S_i obtinute cu funtia $\text{dvs}(Z)$ si am impartit la variabila ssum obtinuta anterior.

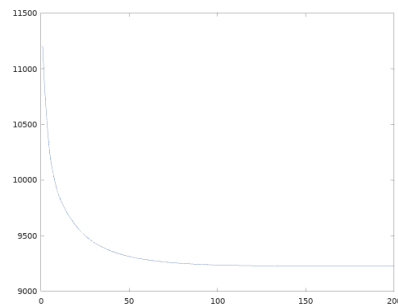


```
plot(X, Y2);
```

Al treilea grafic :

Pentru al treilea grafic am observat ca pot calcula pe Y_3 mai usor, intocmai ca la Cerinta 2, scazand din matricea A_k pe A si ridicand la patrat, urmand ca mai apoi sa fac suma tuturor elementelor din matricea rezultata si sa impart la m si la n .

Procesarea acestui grafic dureaza mai mult, la fel ca la Cerinta 2, din cauza apelurilor succesive a functiei $A_k = \text{task3}(\text{image}, i)$; necesara popularii lui Y_3 cu "eroarea aproximarii" .



```
plot(X, Y3);
```

Al patrulea grafic :

Pentru ultimul task am inlocuit in formula pe k , fiind singura variabila care se modifica . Asadar, am iterat cu un for valorile lui k si am inmultit cu 2, am adunat 1, si am impartit la n .

