Dekompozicije matrica

Vresk, Marko

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Marko Vresk

DEKOMPOZICIJE MATRICA

ZAVRŠNI RAD

Varaždin, 2023.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Vresk

Matični broj: 0016141883

Studij: Informacijski sustavi

DEKOMPOZICIJE MATRICA

ZAVRŠNI RAD

Mentor :

Doc. dr. sc. Bojan Žugec

Varaždin, rujan 2023.

Marko Vresk

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Tema ovog rada će biti upoznavanje s osnovama LU, Cholesky, SVD i QR dekompozicija i njihovih raznih varijanta. Na početku ćemo se kratko upoznati s matricama i najvažnijim pojmovima vezanim uz matrice, a koje ćemo koristiti za razumijevanje ovih metoda. Zatim ćemo se upoznati sa LU dekompozicijom i njezinom varijantom i pokazati implementaciju u programskom jeziku Python. Nakon LU dekompozicije upoznati ćemo se sa Cholesky dekompozicijom i njezinom varijantom. Nakon Cholesky dekompozicije slijedi QR dekompozicija sa svojim varijantama te ćemo na kraju prikazati SVD dekompoziciju. Za svaku dekompoziciju biti će prikazan primjer rješavanja zadatka korištenjem te dekompozicije i njezina implementacija u programskom jeziku Python.

Ključne riječi: matrice, LU dekompozicija, Cholesky dekompozicija, SVD dekompozicija, QR dekompozicija, implementacija dekompozicija u Pythonu

Sadržaj

1. Uvod 1

2. Matrice 2
2.1. Vrste matrica 2
2.2. Determinante matrica 4
2.3. Množenje matrica 5

3. Sustav linearnih jednadžbi 7

4. Gaussov postupak eliminacija 8

5. Python 9

6. LU dekompozicija 10
6.1. Primjer LU dekompozicije 12
6.2. LU dekompozicija s pivotiranjem 13
6.2.1. Primjer LU dekompozicije s pivotiranjem 15
6.3. Implementacija u programskom jeziku Python 18

7. Cholesky dekompozicija 21
7.1. Primjer Cholesky dekompozicije 21
7.2. LDL^T dekompozicija 23
7.2.1. Primjer LDL^T dekompozicije 24
7.3. Implementacija Cholesky dekompozicije u Pythonu 25

8. QR dekompozicija 28
8.1. QR dekompozicija koristeći Gram-Schmidt metodu 28
8.1.1. Primjer QR dekompozicije koristeći Gram-Schmidt metodu 29
8.2. QR dekompozicija koristeći Givensove rotacije 30
8.2.1. Primjer QR dekompozicije koristeći Givensove rotacije 32
8.3. QR dekompozicija koristeći Householderove transformacije 33
8.3.1. Primjer QR dekompozicije koristeći Householderove transformacije 34
8.4. Implementacija QR dekompozicije u Pythonu 36

9. SVD dekompozicija 38
9.1. Primjer SVD dekompozicije 39
9.1.1. Implementacija SVD dekompozicije u Pythonu 42

10. Zaključak 44

11. Navođenje literature 45

Popis literature 46

Popis slika 47

1. Uvod

Rješavanje sustava jednadžbi jedan je od najvažnijih problema sa kojima se susreću matematičari. Problem rješavanja sustava jednadžbi je vrlo složen pa zbog toga postoje posebne grane numeričke analize koje se bave izradom efikasnih algoritama za rješavanje ovog problema. Za rješavanje sustava jednadžbi pomoću matrica najčešće se koristi postupak Gaussove eliminacije. Takav način rješavanja sustava jednadžbi je često spor kod velike količine jednadžbi zbog toga je potrebno transformirati zadanu matricu u produkt dviju ili više jednostavnijih matrica. Postupak transformiranja zadane matrice u produkt dviju ili više jednostavnijih matrica nazivamo matrična dekompozicija odnosno matrična faktorizacija [11] .

Matričnim dekompozicijama pojednostavljuje se postupak dobivanja rješenja. Postoje razne vrste dekompozicija, a mi ćemo se u ovom radu baviti s LU dekompozicijom i njezinom varijacijom, Cholesky dekompozicijom i njezinom varijacijom, QR dekompozicijom i njezinim varijacijama te na kraju sa SVD dekompozicijom. Dekompozicije su složeno područje i nećemo detaljnije ulaziti u njih već će se ovaj rad bazirati na osnovnim informacijama koje su potrebne za potpuno razumijevanje ranije nabrojanih dekompozicija.

U drugom poglavlju bavit ćemo se osnovnom definicijom matrice i najvažnijim pojmovima vezanim uz matrice koji su nam potrebni za razumijavanje dekompozicija. Opisati ćemo osnovne vrste matrica koje su nam potrebne za dekompozicije, upoznat ćemo se sa pojmom determinante i kako se ona dobiva i samim postupkom množenja matrica koje je neophodno za razumijevanje dekompozicija. U trećem poglavlju ćemo kratko opisati sustave linearnih jednadžbi. U četvrtom poglavlju bavimo se Gaussovim postupkom eliminacije koji predstavlja temelj većine metoda dekompozicije. Svaku metodu dekompozicije ćemo implementirati u Pythonu pa se zbog toga u petom poglavlju bavimo programskim jezikom Python, navodimo neke njegove prednosti i nedostatke i kratki opis njegovih mogućnosti i osobina.

U šestom poglavlju bavimo se LU dekompozicijom. Opisujemo sam postupak LU dekompozicije i dajemo praktični primjer rješavanja. Bavimo se i njezinom varijacijom, LU dekompozicijom sa pivotiranjem i na praktičnom primjeru prikazujemu tu varijaciju. Na kraju tog poglavlja imamo implementaciju LU dekompozicije u Pythonu. Sedmo poglavlje bavi se Cholesky dekompozicijom i njezinom varijacijom LDL^T dekompozicijom. Kao i za prethodno poglavlje i tu dekompoziciju prikazujemo na praktičnom primjeru i u Pythonu. Osmo poglavlje je rezervirano za QR dekompoziciju i njezine varijacije. Prvo se upoznajemo s QR dekompozicijom koristeći Gram-Schmidt metodu, zatim koristeći Givensove rotacije i na kraju s varijacijom koja koristi Householderove transformacije. Svaka od varijacija je prikazana na praktičnom primjeru. Deveto poglavlje bavi se SVD dekompozicijom i njezinim osnovnim karakteristikama.

Kroz ovaj rad upoznajemo se s postupkom rješavanja gore navedenih dekompozicija kao i s njihovim osnovnim prednostima i manama. Dobit ćemo kratki uvid u svaku od ovih dekompozicija kao i u moguće probleme sa kojima se susrećemo kod korištenja tih dekompozicija.

2. Matrice

Matrica je pravokutna tablica koja ima određeni broj redaka i stupaca u kojima se nalaze elementi. Elementi matrice su najčešće brojevi, ali mogu biti i drugi objekti (vektori, funkcije). Broj redaka najčešće označavamo s m dok broj stupaca označavamo s n. Za matricu još možemo reći da je to niz elemenata $(a_{11},\ a_{12}, ..., a_{1n},\ a_{21},\ a_{22}, ...,\ a_{2n}, ..., a_{m1},\ a_{m2}, ...,\ a_{mn}) \in \mathbb{R}^{m \times n}$ posloženih u pravokutnu tablicu [1]. Mi ćemo se u ovom radu baviti realnim matricama. Neka je $M = (1,\ 2,\ m)$ i $N = (1,\ 2,\ n)$ tada je **realna matrica** A tipa $(m,\ n)$ funkcija oblika $A : M \times N \Rightarrow \mathbb{R}$ pri čemu se funkcijska vrijednost $A(ij)$ označava s a_{ij} i smješta u i-ti redak i $j-$ti stupac tablice s m redova i n stupaca. Skup svih realnih matrica tipa $m \times n$ označavamo sa $M_{mn}(\mathbb{R})$. Za elemente a_{ij} matrice A koristimo oznaku $[a_{ij}]$. Važno je napomenuti da sve operacije koje vrijede za realne matrice vrijede i za kompleksne matrice [18].

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

Redak je horizontalni niz elemenata u matrici. Za red možemo još reći da je to uređena n-torka $(a_{11},\ a_{12},\ a_{13}, ..., a_{1n}), (a_{21},\ a_{22},\ a_{23}, ..., a_{2n}), (a_{31},\ a_{32},\ a_{33}, ..., a_{3n}), ..., (a_{m1},\ a_{m2},\ a_{m3}, ..., a_{mn})$. Stupac je vertikalni niz elemenata u matrici. Stupac je uređena n-torka koju čine elementi $(a_{11},\ a_{21},\ a_{31}, ..., a_{m1}), (a_{12},\ a_{22},\ a_{32}, ..., a_{m2}), (a_{13},\ a_{23},\ a_{33}, ..., a_{m3}), ..., (a_{1n},\ a_{2n},\ a_{3n}, ..., a_{mn})$. Matrice se označavaju velikim tiskanim slovima (A, B, C...).

2.1. Vrste matrica

Postoji veliki broj različitih vrsta matrica. Mi ćemo se u ovom dijelu upoznati samo s osnovnim vrstama matrica koje su nam potrebne za razumijevanje LU, Cholesky, SVD i QR dekompozicija.

Kvadratna matrica je matrica koja ima isti broj redova i stupaca. Za takvu matricu kažemo da ima red koji odgovara broju redaka (stupaca). Ako matrica ima red n to znači da ta matrica ima n redova i stupaca, odnosno vrijedi $m = n$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Jednostupčasta matrica (vektor-stupac) je matrica koja ima samo jedan stupac. Ova matrica je oblika $(m, 1)$. Najčešće se označava malim slovom b.

$$b = \begin{bmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{m1} \end{bmatrix}$$

Transponirana matrica je matrica koju dobivamo tako da retke zamijenimo stupcima i stupce zamijenimo recima. Vrijedi sljedeća jednakost $A_{ij} = A^T_{ji}$. Transponiranu matricu matrice A označavamo kao A^T .

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{bmatrix}$$

Simetrična matrica je vrsta kvadratne matrice čijom zamjenom redova sa stupcima i stupaca sa redovima dobivamo istu matricu. Da bi u potpunosti mogli razumijeti simetričnu matricu uvodimo pojam **glavne dijagonale**. Glavna dijagonala (označena na doljnjoj matrici slovom g) počinje u gornjem lijevom kutu matrice i završava u donjem desnom kutu matrice. Preciznije glavna dijagonala je uređena n-torka $(a_{11},\ a_{22},\ a_{33}, ..., a_{nn})$. Sad možemo precizirati simetričnu matricu kao kvadratnu matricu čiji su elementi simetrični s obzirom na glavnu dijagonalu. Vrijedi $A = A^T$, tako za taj tip matrice npr. vrijedi $a_{12} = a_{21},\ a_{13} = a_{31}$ (označeno na donjoj matrici istim slovima)...

$$A = \begin{bmatrix} g & s & a \\ s & g & b \\ a & b & g \end{bmatrix}$$

Gornjetrokutasta matrica je kvadratna matrica kojoj su svi elementi ispod dijagonale jednaki nuli, tj. $a_{ij} = 0$ kad je $i > j$. Gornjetrokutasta matrica označava se velikim slovom U. [2]

$$U = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}$$

Donjetrokutasta matrica je kvadratna matrica kojoj su svi elementi iznad dijagonale jednaki nuli, tj. $a_{ij} = 0$ kad je $i < j$. Donjetrokutasta matrica označava se velikim slovom L. [2]

$$L = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Dijagonalna matrica je kvadratna matrica kojoj su svi elementi izvan glavne dijagonale jednaki nuli, tj. $a_{ij} = 0$ kad je $i \neq j$. [2]

$$A = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}$$

Jedinična matrica je kvadratna matrica kojoj su svi elementi izvan glavne dijagonale jednaki nuli, a elementi glavne dijagonale su jednaki 1. Vrijedi $a_{ij} = 0$ kad je $i \neq j$ i $a_{ij} = 1$ kad je $i = j$. Označavamo ju sa slovom I. [2]

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Inverzna matrica je ona matrica kojom moramo pomnožiti kvadratnu matricu da bismo dobili jediničnu matricu. Vrijedi $A^{(-1)}A = I$. $A^{(-1)}$ nazivamo inverznom matricom

Regularna matrica je ona matrica za koju postoji inverz matrice, odnosno njezina determinanta mora biti različita od 0. Ako vrijedi $A^{(-1)}A = I$. A nazivamo regularnom matricom, ako to ne vrijedi A nazivamo **singularnom matricom**.

Unitarna matrica je ona matrica $A \in (\mathbb{C})^{n\times n}$ za koju vrijedi $AA = I$, to jest stupci matrice A su ortonormirani s obzirom na standarni skalarni umnožak [17].

2.2. Determinante matrica

Determinanta je funkcija definirana na skupu svih kvadratnih matrica, a poprima vrijednosti iz skupa skalara. Determinante najčešće označavamo sa det i imenom matrice. Determinante definiramo induktivno. To znači da determinantu n-tog reda definiramo pomoću determinante $n - 1$ reda [5]. Determinante su složena tema, a mi ćemo zbog jednostavnosti pokazati izračun determinante za matrice reda 3. Ovdje uvodimo i pojam **minore**. Minoru možemo definirati kao determinantu podmatrice. Minora je zapravo determinanta kvadratne matrice dobivena brisanjem nekih njezinih redova ili stupaca. Determinanta matrice $A[a]$ koja je prvog reda je broj a (a je vrijednost elementa matrice reda 1). Determinanta matrice A drugog reda

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

računa se prema formuli:

$$det(A) = a_{11}a_{22} - a_{12}a_{21}$$

Determinanta matrice A trećeg reda

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

računa se prema formuli:

$$det(A) = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Iz navedenih formula možemo doći do zaključka da determinanta za matricu A oblika

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

ima sljedeći oblik:

$$det(A) = a_{11}det(A)_{11} - a_{21}det(A)_{21} + ... + (-1)^{n+1}a_{n1}det(A)_{n1} = \sum_{k=1}^{n}(-1)^{k+1}a_{k1}det(A)_{k1}$$

Svojstva determinanti pomoću kojih možemo ubrzati njihovo dobivanje:

1. Ako matrica ima red ili stupac čiji su svi elementi 0 tada je njena determinanta jednaka 0.
2. Ako matrica ima dva jednaka reda ili stupca tada je njena determinanta jednaka 0.
3. Ako matrica ima dva proporcionalna reda ili stupca tada je njena determinanta jednaka 0.
4. Ako je jedan red ili stupac matrice jednak sumi druga dva reda ili stupca tada je njena determinanta jednaka 0.
5. U determinanti možemo faktorizirati neki red ili stupac.
6. Vrijednost determinante se ne mijenja ako zbrajamo ili oduzimamo redove ili stupce sa drugim redovima ili stupcima.
7. U determinanti možemo zbrajati ili oduzimati prethodno pomnožene redove ili stupce.
8. Determinanta matrice jednaka je determinanti njene transponirane matrice [6].

2.3. Množenje matrica

Množenje matrica je moguće samo za one matrice koje su ulančane. Ulančane matrice su one matrice čiji je broj stupaca prve matrice jednak broju redaka druge matrice. Drmač, Marušić, Singer, Hari, Rogina, Singer [3] navode da je umnožak (produkt) matrica A i B matrica $C = A \times B \in \mathbb{R}^{m \times p}$ kojoj su elementi određeni formulom

$$C_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj} \text{ za sve } 1 \leq i \leq m, 1 \leq j \leq p$$

Umnoškom matrice A s m redova i n stupaca i matrice B s n redaka i p stupaca dobivamo matricu C s m redova i p stupaca. Matrice množimo tako da svaki redak prve matrice pomnožimo sa svakim stupcem druge matrice. Prvo množimo prvi element prvog retka sa prvim elementom prvog stupca, zatim množimo drugi element prvog retka sa drugim elementom prvog stupca i taj postupak ponavljamo tako dugo dok ne pomnožimo sve elemente prvog retka sa svim elementima prvog stupca. Zatim zbrojimo dobivene vrijednosti. Dobiveni rezultat biti će element prvog retka i prvog stupca. Nakon toga množimo sve elemente prvog reda sa svim elementima drugog stupca i to ponavljamo dok sa prvim retkom ne pomnožimo sve stupce. Nakon toga prelazimo na drugi redak i ponavljamo isti postupak. Zatim prelazimo na sljedeći redak i tako sve dok ne izmnožimo zadnji redak s zadnjim stupcem.

$$AB = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{bmatrix}$$

$$BA = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = \begin{bmatrix} b_{11}a_{11} + b_{12}a_{21} & b_{11}a_{12} + b_{12}a_{22} & b_{11}a_{13} + b_{12}a_{23} \\ b_{21}a_{11} + b_{22}a_{21} & b_{21}a_{12} + b_{22}a_{22} & b_{21}a_{13} + b_{22}a_{23} \\ b_{31}a_{11} + b_{32}a_{21} & b_{31}a_{12} + b_{32}a_{22} & b_{31}a_{13} + b_{32}a_{23} \end{bmatrix}$$

Neka je $A \in \mathbb{R}^{m \times n}$ i $B \in \mathbb{R}^{n \times p}$. Umnožak matrica A i B je matrica $C = A \times B \in \mathbb{R}^{m \times p}$ dok je s oznaka za skalar. Iz prethodnih umnožaka vidimo da ne vrijedi svojstvo komutativnosti (AB = BA). Važna svojstva koja koristimo tijekom umnoška matrica su:

1. $A(BC) = (AB)C \Rightarrow$ lijeva distributivnost
2. $(A + B)C = AC + BC \Rightarrow$ desna distributivnost
3. $s(AB) = (sA)B = A(sB) \Rightarrow$ kompatibilnost množenja matrica sa skalarom
4. $A(B + C) = AB + AC \Rightarrow$ asocijativnost množenja [3]

3. Sustav linearnih jednadžbi

Linearna jednadžba sa nepoznanicama $x_1,\ x_2, ...x_{n-1},\ x_n$ je jednadžba koja ima oblik:

$$a_1x_1 + a_2x_2 + ... + a_{n-1}x_{n-1} + a_nx_n = b$$

Sada dok znamo što je linearna jednadžba možemo definirati sustav linearnih jednadžbi. Sustav linearnih jednadžbi je zapravo skup od m linearnih jednadžbi s n nepoznanica pri tome vrijedi $m, n \in \mathbb{N}$.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1n-1}x_{n-1} + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n-1}x_{n-1} + a_{2n}x_n = b_2 \\ ::: \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn-1}x_{n-1} + a_{mn}x_n = b_m \end{aligned}$$

Pri tome, skalari a_{ij} , $i = 1,\ 2, ..., m,\ \ j = 1,\ 2, ..., n$ zovu se **koeficijenti sustava**, $b_1,\ b_2, ..., b_m$ **slobodni članovi**, a $x_1,\ x_2, ...x_n$ zovu se **nepoznanicama**. [4]

Linearne jednadžbe možemo zapisati pomoću matrica u obliku $Ax = b$. Matrica A sadrži koeficijente sustava. Matrica x je jednostupčana matrica koja sadrži nepoznanice $x_1,\ x_2, ...x_n$, a matrica b je jednostupčana matrica koja sadrži slobodne članove $b_1,\ b_2, ..., b_m$. Kada transformiramo prethodne jednadžbe u odgovarajuće matrice dobivamo sljedeće matrice.

$$\begin{bmatrix} a_{11} & a_{12} & ... & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n-1} & a_{2n} \\ : & : & ::: & : & : \\ a_{m1} & a_{m2} & ... & a_{mn-1} & a_{mn} \end{bmatrix}$$

Nakon što imamo tako zapisani sustav cilj nam je dobiti rješenje korištenjem elementarnim transformacijama. Rješenje ovakvog sustava je uređena n-torka čijim uvrštvanjem u početne jednadžbe dobijemo odgovarajuće vrijednosti (rješenja koja odgovaraju matrici slobodnih članova). Elementarne transformacije koje se izvode nad matricama su:

1. Zamjena redoslijeda stupaca (redaka)
2. Množenje stupca (retka) sa nekim skalarom
3. Dodavanje stupcu (retku) drugi redak (stupac)

4. Gaussov postupak eliminacija

Da bi kasnije mogli objasniti dekompozicije moramo objasniti što je to Gaussov postupak eliminacije. Gaussov postupak eliminacije je postupak rješavanja sustava linearnih jednadžbi oblika $Ax = b$. Osnovna ideja ovog postupka je svođenje na ekvivalentni sustav kako bi lakše mogli odrediti rješenja sustava jednadžbi. Pretpostavimo da imamo zadani sljedeći sustav jednadžbi:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3
\end{aligned}
$$

Vidimo da je sustav tipa 3×3. Tom sustavu sada pridružujemo matricu sustava koja je tipa 3×4. U lijevi dio pišemo vrijednosti koje se nalaze uz x dok u desni dio (odvojen |) pišemo vrijednosti b_1, b_2 i b_3.

$$
\left(\begin{array}{ccc|c}
a_{11} & a_{12} & a_{13} & b_1 \\
a_{21} & a_{22} & a_{23} & b_2 \\
a_{31} & a_{32} & a_{33} & b_3
\end{array}\right)
$$

Transformiramo navedenu matricu korištenjem elementarnih operacija koje su opisane u prethodnom poglavlju. Transformiranjem matrice dobivamo sljedeću matricu.

$$
\left(\begin{array}{ccc|c}
1 & 0 & 0 & c_1 \\
0 & 1 & 0 & c_2 \\
0 & 0 & 1 & c_3
\end{array}\right)
$$

Stupci koji se nalaze lijevo od | mogu imati i drugačiji redoslijed. Dok imamo ovakav sustav možemo očitati rješenje sustava iz odgovarajućih jednadžbi. Prvi stupac se odnosi na x_1, drugi se stupac odnosi na x_2, a treći se stupac odnosi na x_3. x poprima vrijednost one dobivene vrijednosti c gdje je za taj x postavljena vrijednost 1 u lijevom dijelu matrice. Ukoliko postoji redak koji ima sve nule lijevo od znaka |, a desno od tog znaka je različita vrijednost od nula onda zaključujemo da sustav nema rješenja. Ako u desnom dijelu ima red s nulom i za taj red u lijevom dijelu su sve nule onda takav red zanemarujemo.

5. Python

Python je jedan od najkorištenijih programskih jezika današnjice. Python je **interpreterski** programski jezik. Interpreterske programske jezike ne treba prevesti (kompajlirati) prije izvođenja pa se mogu relativno brzo izvesti. Python je osmislio **Guido van Rossum** 1991. godine. Python se koristi za programiranje web aplikacija, desktop aplikacija, malih skripti... Glavne prednosti ovog programskog jezika su jednostavnost i jednostavna i razumljiva sintaksa te veliki broj različitih biblioteka. Python omogućuje nekoliko različitih stilova programiranja. Objektno-orijentirano programiranje, aspektno programiranje i strukturno programiranje su stilovi koje podržava Python. Kao glavna specifičnost u odnosu na druge programske jezike ističe se uvlačanje koje se koristi kao metoda razlikovanja blokova umjesto vitičastih zagrada i ključnih riječi koje se koriste u drugim programskim jezicima. Python se koristi za obradu podataka, zbog toga je važno da su stručnjaci upoznati s njime jer bez njega ne mogu razvijati napredne tehnologije. Ranije smo spomenuli da je jedna od glavnih prednosti veliki broj ugrađenih biblioteka. Najpoznatije biblioteke koje se koriste su **NumPy, SciPy, Django, Math** [8]. Te biblioteke imaju ugrađene funkcije i klase kako bi se olakšalo pisanje programskog koda. Osnovne karakteristike Pythona:

1. Besplatan i ima dostupan izvorni kod
2. Jednostavno pisanje i razumijevanje koda
3. Objektno-orijentiran, ali podržava i druge stilove programiranja
4. Ima podršku za grafičko korisničko sučelje
5. Lako debugiranje (pronalaženje grešaka i nedostataka)
6. Neovisan o platformi (program napisan na windowsima bez problema možemo pokrenuti na drugoj platformi (Unix ili Mac))
7. Lako se integrira s drugim programskim jezicima
8. Interpreterski jezik
9. Veliki broj različitih biblioteka
10. Dinamičan jezik (nije potrebno specificirati tip varijable)
11. Koristi dinamičku alokaciju memorije [7]

Slika 1: Python logo

Iz svega navedenog možemo zaključiti zašto je Python toliko korišteni programski jezik. Njegova fleksibilnost i svestranost je velika prednost u odnosu na druge programske jezike i uzrok njegovog uspona u korištenju.

6. LU dekompozicija

LU dekompozicija je rastav matrice A na umnožak donjetrokutaste matrice i gornjetrokutaste matrice. Za matricu A mora vrijediti $A \in \mathbb{R}^{n\times n}$. LU dekompozicija vrijedi samo za regularne matrice. Odnosno determinanta matrice za koju izvodimo LU dekompoziciju mora biti različita od 0. LU dekompozicija ima široku primjenu, a danas se najčešće koristi kod izračuna tokova snage kod velikih dimenzija i velikog broja članova zbog uštede vremena [3].

Nakon što smo se ukratko upoznali s LU dekompozicijom i njenom primjenom, možemo objasniti kako se sama dekompozicija izvodi. Zbog jednostavnosti uzet ćemo matricu reda 3 (matrica oblika 3×3). Pretpostavimo da imamo definiranu matricu A oblika:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Gornjetrokutasta matrica za matricu A:

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Donjetrokutasta matrica za matricu A:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$$

Matrica b je oblika:

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Prvo provjerimo da li je matrica A regularna. To provjeravamo izračunom determinante. Determinanta matrice reda 3 se računa prema formuli:

$$det(A) = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Zatim zapišemu matricu A kao umnožak matrica L i U. $A = LU$

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{bmatrix}$$

Sad treba riješiti sustav jednadžbi

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Rješavanjem sustava jednadžbi dobivamo sljedeće vrijednosti

$$\begin{aligned} &u_{11} = a_{11} \\ &u_{12} = a_{12} \\ &u_{13} = a_{13} \\ &l_{21}u_{11} = a_{21} \\ &l_{21}u_{12} + u_{22} = a_{22} \\ &l_{21}u_{13} + u_{23} = a_{23} \\ &l_{31}u_{11} = a_{31} \\ &l_{31}u_{12} + l_{32}u_{22} = a_{32} \\ &l_{31}u_{13} + l_{32}u_{23} + u_{33} = a_{33} \end{aligned}$$

Sad kada smo dobili nepoznate vrijednosti matrica L i U možemo izračunati vrijednost matrice y prema formuli $Ly = b$.

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Možemo uočiti da u prvom redu imamo samo jednu vrijednost koja je različita od 0 pa možemo odmah dobiti vrijednost y_1. U drugom redu imamo dvije vrijednosti u matrici L različite od 0. Uvrštavanje izračunate vrijednosti y_1 ostaje nam samo jedna nepoznanica y_2, a nju jednostavno dobivamo. Uvrštavanjem y_1 i y_2 u treći red trebamo izračunati samo jednu nepoznanicu y_3 pa bez većeg problema dobivamo skup rješenja. Za računanje vrijednosti y koristimo postupak koji se zove **eliminacija ili supstitucija unaprijed**. Tu zapravo vidimo koja je prednost svođenja na gornjetrokutaste i donjetrokutaste matrice.

Nakon što smo dobili y_1, y_2, y_3 možemo krenuti s posljednjim korakom. Uvrštavanjem izračunatih vrijednosti u formulu $Ux = y$ dobivamo skup x-eva koji su rješenje zadanog sustava linearnih jednadžbi.

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Možemo uočiti da u zadnjem redu u matrici L imamo samo jednu vrijednost različitu od nule pa krećemo prvo s izračunom vrijednosti zadnjeg x-a u ovom slučaju x_3. Nakon što dobimo x_3 prelazimo na rješavanje predzadnje jednadžbe pošto smo dobili već x_2. Zatim prelazimo na prethodnu jednadžbu gdje trebamo dobiti samo x_1. Ovakav postupak zove se **eliminacija ili supstitucija unazad**.

6.1. Primjer LU dekompozicije

Uzmimo za primjer sljedeće matrice

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix}, b = \begin{bmatrix} 3 \\ 13 \\ 4 \end{bmatrix}$$

Prvo izračunamo determinantu za matricu A

$$det(A) = 1 \begin{vmatrix} 8 & 14 \\ 6 & 13 \end{vmatrix} - 2 \begin{vmatrix} 14 & 3 \\ 13 & 2 \end{vmatrix} + 4 \begin{vmatrix} 3 & 8 \\ 2 & 6 \end{vmatrix} =$$

$a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{31}(a_{21}a_{32} - a_{22}a_{31}) = (1(8 \cdot 13 - 14 \cdot 6) - 2(3 \cdot 13 - 14 \cdot 2) + 4(3 \cdot 6 - 8 \cdot 2)) = 6$

Dobili smo determinantu različitu od 0 pa možemo dobiti LU dekompoziciju. Sada pomnožimo matricu L i U.

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{bmatrix}$$

Rješavamo dobiveni sustav jednadžbi

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix}$$

$$u_{11} = a_{11} \Rightarrow u_{11} = 1$$
$$u_{12} = a_{12} \Rightarrow u_{12} = 2$$
$$u_{13} = a_{13} \Rightarrow u_{13} = 4$$
$$l_{21}u_{11} = a_{21} \Rightarrow l_{21} = 3$$
$$l_{21}u_{12} + u_{22} = a_{22} \Rightarrow 3 \cdot 2 + u_{22} = 8 \Rightarrow u_{22} = 2$$
$$l_{21}u_{13} + u_{23} = a_{23} \Rightarrow 3 \cdot 4 + u_{23} = 14 \Rightarrow u_{23} = 2$$
$$l_{31}u_{11} = a_{31} \Rightarrow l_{31} \cdot 1 = 2 \Rightarrow l_{31} = 2$$
$$l_{31}u_{12} + l_{32}u_{22} = a_{32} \Rightarrow 2 \cdot 2 + l_{32} \cdot 2 = 6 \Rightarrow l_{32} = 1$$
$$l_{31}u_{13} + l_{32}u_{23} + u_{33} = a_{33} \Rightarrow 2 \cdot 4 + 1 \cdot 2 + u_{33} = 13 \Rightarrow u_{33} = 3$$

Matrice U i L sad izgledaju ovako:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}, U = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix}$$

Slijedi izračun y na temelju formule $Ly = b$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 13 \\ 4 \end{bmatrix}$$

$$y_1 = 3$$
$$3y_1 + y_2 = 13 \Rightarrow 3 \cdot 3 + y_2 = 13 \Rightarrow y_2 = 4$$
$$2y_1 + y_2 + y_3 = 4 \Rightarrow 2 \cdot 3 + 4 \cdot 1 + y_3 = 4 \Rightarrow y_3 = -6$$

Sad kad imamo y možemo dobiti x na temelju formule $Ux = y$

$$\begin{bmatrix} 1 & 2 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ -6 \end{bmatrix}$$

$$3x_3 = -6 \Rightarrow x_3 = -2$$
$$2x_2 + 2x_3 = 4 \Rightarrow 2 \cdot x_2 + 2 \cdot (-2) = 4 \Rightarrow x_2 = 4$$
$$x_1 + 2x_2 + 4x_3 = 3 \Rightarrow x_1 + 2 \cdot 4 + 4 \cdot (-2) = 3 \Rightarrow x_1 = 3$$

Time smo pomoću LU dekompozicije dobili da je rješenje sustava linearnih jednadžbi uređena trojka (3, 4, -2).

6.2. LU dekompozicija s pivotiranjem

Ranije prikazani algoritam ima neke nedostatke. Naime matrica A mora zadovoljavati strukturu i određene uvjete. Sve glavne minore matrice moraja biti različite od 0. Odnosno sve podmatrice moraju biti regularne. Dokaz te tvrdnje navode Drmač, Singer, Marušić, Hari, Singer, Rogina [3, str.157. - 161.] pa mi sada nećemo trošiti vrijeme na dokaz te tvrdnje nego ćemo to prikazati na primjeru. Pretpostavimo da imamo matricu A:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Determinanta matrice A je -1 pa znamo da ova matrica ima rješenje, ali nema LU dekompoziciju jer joj je prva glavna minora $m_{11} = a_{11} = 0$. Prema formuli $A = LU$ matricu A možemo zapisati kao:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$$

To znači da je $u_{11} = 0$, a $l_{21}u_{21} = 1$. Vidimo da je to nemoguće jer $l_{21} \times 0 = 1$ je nemoguće dobiti.

Znamo da matrica A prikazuje sustav

$$0x_1 + x_2 = b_1$$
$$x_1 + x_2 = b_2$$

koji uvijek ima rješenje

$$x_2 = b_1$$
$$x_1 = b_2 - b_1$$

Zapisujemo ju kao:

$$x_1 + x_2 = b_2$$
$$0x_1 + x_2 = b_1$$

Matrica ovog sustava je

$$A' = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Vidimo da sad postoji LU dekompozicija. Odnos između matrica A i A' zapisujemo kao $A' = PA$.

$$A' = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Matricu P zovemo **matrica permutacija**.[3]

Drugi važan pojam za potpuno razumijevanje LU dekompozicije s pivotiranjem je **pivotni element (pivot)**. Pivotni element možemo definirati kao element koji se nalazi na dijagonali prilikom izvođenja Gaussove eliminacije. Pomoću njega se eliminiraju elementi koji se nalaze u istom stupcu ispod njega korištenjem elementarnih transformacija. Sad kad smo se upoznali sa svim važnim pojmovima za razumijevanje ove metode možemo krenuti s općenitim algoritmom.

Prvi korak u LU dekompoziciji na temelju pivotiranja je određivanje pivota za prvi stupac. Pivot za prvi stupac je onaj element koji ima najveću apsolutnu vrijednost za prvi stupac. Ako se ta vrijednost ne nalazi u prvom redu mijenjamo prvi red s redom u kojem se nalazi ta vrijednost. Red s najvećom apsolutnom vrijednošću u prvom stupcu postaje prvi red dok red koji je do sada bio prvi odlazi na poziciju na kojoj se nalazi red s kojim smo ga zamijenili. Promjenu reda označimo u matrici P. Ranije smo spomenuli da je P matrica permutacija. Matrica P je istog reda kao matrica A i označava koje su permutacije vršene nad matricom A. Matrica P ima 1 na glavnoj dijagonali dok su ostali elementi 0. Tako na primjer ako zamijenimo prvi red matrice s trećim redom matrice u matrici P prikazujemo tu promjenu tako da zamijenimo treći red s prvim redom. Dok na primjer zamjenom drugog i trećeg reda promjenu označavamo tako da u matrici P zamijenimo drugi i treći red. Nakon promjene u matrici P možemo izvesti Gaussovu eliminaciju. Ispod pivotnog elementa moramo dobiti 0 pa množimo pivotni element s vrijednošću koja pribrojena vrijednosti koja se nalazi ispod pivota za taj red daje 0. Postupak ponavljamo za sve retke i stupce. Dok smo pomnožili elemente i pridodali odgovarajućim vrijednostima krećemo s traženjem pivotnog elementa za drugi stupac na temelju vrijednosti matrice A dobivenih nakon eliminacije vrijednosti koje se nalaze ispod pivota za prvi stupac. Zanemaruje se prvi red i gledaju se drugi stupci u ostalim redovima. Ponavlja se postupak opisan za prvi red (svi elementi koji se nalaze ispod pivota se množe s vrijednostima kojima se dobiva 0 za taj stupac svakog reda koji se nalazi ispod pivota). Postupak ponavljamo za sljedeći stupac i tako sve dok ne dobimo gornjetrokutastu matricu **matricu** U.

Matricu L možemo dobiti paralelno tijekom pronalaska matrice U ili na kraju. Elemente matrice L za prvi stupac dobivamo tako da za svaki red podijelimo vrijednosti iz matrice A u prvom stupcu tog reda s pivotom za taj stupac. Za drugi stupac tako da elemente matrice dobivene nakon prve permutacije matrice A podijelimo s pivotom za drugi stupac koji se također određuje nakon permutacije. Za treći stupac tako da elemente matrice dobivene nakon druge permutacije matrice A podijelimo s pivotom za treći stupac i taj postupak ponavljamo sve dok ne dobimo sve elemente matrice L. Nakon što su nam poznate matrice P, L, U određujemo **matricu** b'. Matricu b' određujemo permutacijom matrica b i P. Primjerice ako matrica P za prvi red ima 1 u trećem stupcu to znači da u prvi red matrice b' dolazi vrijednost koja se nalazi u trećem redu matrice b. Nakon što odredimo matricu b' iz formule $Ly = b'$ odredimo vrijednosti y rješavanjem sustava linernih jednadžbi eliminacijom unaprijed. Kad su nam poznati svi y, x-eve dobivamo na temelju formule $Ux = y$ rješavanjem sustava jednadžbi eliminacijom unatrag. Dobiveni x-evi pretstavlja rješenje zadanog sustava jednadžbi.

6.2.1. Primjer LU dekompozicije s pivotiranjem

Uzmimo za primjer matrice:

$$A = \begin{bmatrix} 2 & 0 & 4 & 6 \\ -2 & 4 & 4 & -6 \\ 0 & 2 & 2 & 8 \\ 12 & 4 & 4 & 8 \end{bmatrix}, b = \begin{bmatrix} 2 \\ -2 \\ 4 \\ 2 \end{bmatrix}, P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Prvo što radimo je odabir pivotnog elementa za prvi stupac. Vidimo da najveću apsolutnu vrijednost u prvom stupcu ima red 4. Pa zamijenimo prvi i četvrti red. Tu promijenu naznačimo u matrici P.

$$P^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \ P^{(1)}A = \begin{bmatrix} 12 & 4 & 4 & 8 \\ -2 & 4 & 4 & -6 \\ 0 & 2 & 2 & 8 \\ 2 & 0 & 4 & 6 \end{bmatrix}$$

Zatim računamo matricu $A^{(1)}$ korištenjem Gaussovih eliminacija. Vrijednosti matrice L dobivamo tako da vrijednost koja se nalazi ispod pivota podijelimo s pivotom za taj stupac. Za prvi slučaj imamo sljedeću matricu L:

$$\mathsf{L}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{41}} & 1 & 0 & 0 \\ \frac{a_{31}}{a_{41}} & 0 & 1 & 0 \\ \frac{a_{11}}{a_{41}} & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-1}{6} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{6} & 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathsf{A}^{(1)} = L^{(1)}P^{(1)}A = \begin{bmatrix} 12 & 4 & 4 & 8 \\ 0 & \frac{14}{3} & \frac{14}{3} & \frac{-14}{3} \\ 0 & 2 & 2 & 8 \\ 0 & \frac{-2}{3} & \frac{10}{3} & \frac{14}{3} \end{bmatrix}$$

Sljedeći korak je provjera pivot elementa za drugi stupac. Pivot za drugi stupac se određuje na temelju vrijednosti matrice $A^{(1)}$. Vidimo da najveću apsolutnu vrijednost za drugi stupac ima drugi red pa matrica permutacija ostaje identična prethodnoj matrici permutacija jer nema promjena. Računamo matricu $A^{(2)}$ i $L^{(2)}$.

$$\mathsf{L}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{41}} & 1 & 0 & 0 \\ \frac{a_{31}}{a_{41}} & \frac{a_{32}}{a_{22}}(1) & 1 & 0 \\ \frac{a_{11}}{a_{41}} & \frac{a_{42}}{a_{22}}(1) & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-1}{6} & 1 & 0 & 0 \\ 0 & \frac{3}{7} & 1 & 0 \\ \frac{1}{6} & \frac{-1}{7} & 0 & 1 \end{bmatrix} \Rightarrow \mathsf{A}^{(2)} = L^{(2)}P^{(2)}A^{(1)} = \begin{bmatrix} 12 & 4 & 4 & 8 \\ 0 & \frac{14}{3} & \frac{14}{3} & \frac{-14}{3} \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 4 & 4 \end{bmatrix}$$

Zatim tražimo pivot element za treći stupac. Vidimo da je 4 u četvrtom redu veća vrijednost od 0 u trećem redu. Ranije smo spomenuli da se ne gledaju redovi koje smo prethodno koristili za pivotiranje. Moramo promijeniti redoslijed redova 3 i 4. Promjenu označimo u tablici permutacija.

$$P^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P^{(3)}A^{(2)} = \begin{bmatrix} 12 & 4 & 4 & 8 \\ 0 & \frac{14}{3} & \frac{14}{3} & \frac{-14}{3} \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

Sad računamo matricu $L^{(3)}$ i $A^{(3)}$.

$$\mathsf{L}^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{a_{21}}{a_{41}} & 1 & 0 & 0 \\ \frac{a_{11}}{a_{41}} & \frac{a_{42}}{a_{22}}^{(1)} & 1 & 0 \\ \frac{a_{31}}{a_{41}} & \frac{a_{32}}{a_{22}}^{(1)} & \frac{a_{33}}{a_{43}}^{(2)} & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-1}{6} & 1 & 0 & 0 \\ \frac{1}{6} & \frac{-1}{7} & 1 & 0 \\ 0 & \frac{3}{7} & 0 & 1 \end{bmatrix} \Rightarrow \mathsf{A}^{(3)} = L^{(3)}P^{(3)}A^{(2)=} \begin{bmatrix} 12 & 4 & 4 & 8 \\ 0 & \frac{14}{3} & \frac{14}{3} & \frac{-14}{3} \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

Vidimo da smo sad dobili donjetrokutastu i gornjetrokutastu matricu.

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-1}{6} & 1 & 0 & 0 \\ \frac{1}{6} & \frac{-1}{7} & 1 & 0 \\ 0 & \frac{3}{7} & 0 & 1 \end{bmatrix}, U = \begin{bmatrix} 12 & 4 & 4 & 8 \\ 0 & \frac{14}{3} & \frac{14}{3} & \frac{-14}{3} \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

Računamo permutaciju matrice b na temelju matrice permutacija

$$b^{'} = Pb = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 2 \\ 4 \end{bmatrix}$$

Kad imamo matricu b' računamo y uvrštavanjem u formulu $Ly = b'$. y dobivamo supstitucijom unaprijed.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-1}{6} & 1 & 0 & 0 \\ \frac{1}{6} & \frac{-1}{7} & 1 & 0 \\ 0 & \frac{3}{7} & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 2 \\ 4 \end{bmatrix}$$

Rješavanjem sustava dobivamo

$$\begin{aligned} & y_1 = 2 \\ & \tfrac{-1}{6}y_1 + y_2 = -2 \Rightarrow \tfrac{-1}{6} \cdot 2 + y_2 = -2 \Rightarrow y_2 = \tfrac{-5}{3} \\ & \tfrac{1}{6}y_1 + \tfrac{-1}{7}y_2 + y_3 = 2 \Rightarrow \tfrac{1}{6} \cdot 2 - \tfrac{-1}{7} \cdot \tfrac{-5}{3} + y_3 = 2 \Rightarrow y_3 = \tfrac{10}{7} \\ & \tfrac{3}{7}y_2 = 4 \Rightarrow \tfrac{3}{7} \cdot \tfrac{-5}{3} \Rightarrow y_4 = \tfrac{33}{7} \end{aligned}$$

Uvrštavanjem u formulu $Ux = y$ dobivamo rješenje. x dobivamo supstitucijom unatrag.

$$\begin{bmatrix} 12 & 4 & 4 & 8 \\ 0 & \frac{14}{3} & \frac{14}{3} & \frac{-14}{3} \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ \frac{-5}{3} \\ \frac{10}{7} \\ \frac{33}{7} \end{bmatrix}$$

$$10x_4 = \frac{33}{7} \Rightarrow x_4 = \frac{33}{70}$$
$$4x_3 + 4x_4 = \frac{10}{7} \Rightarrow 4x_3 + 4 \cdot \frac{33}{70} = \frac{10}{7} \Rightarrow x_3 = \frac{-4}{35}$$
$$\frac{14}{3}x_2 + \frac{14}{3}x_3 - \frac{14}{3}x_4 = \frac{10}{7} \Rightarrow \frac{14}{3}x_2 + \frac{14}{3} \cdot \frac{-4}{35} - \frac{14}{3} \cdot \frac{33}{70} \Rightarrow x_2 = \frac{8}{35}$$
$$12x_1 + 4x_2 + 4x_3 + 8x_4 = 2 \Rightarrow 12x_1 + 4 \cdot \frac{8}{35} + 4 \cdot \frac{-4}{35} + 8 \cdot \frac{33}{70} = 2 \Rightarrow x_1 = \frac{-13}{70}$$

6.3. Implementacija u programskom jeziku Python

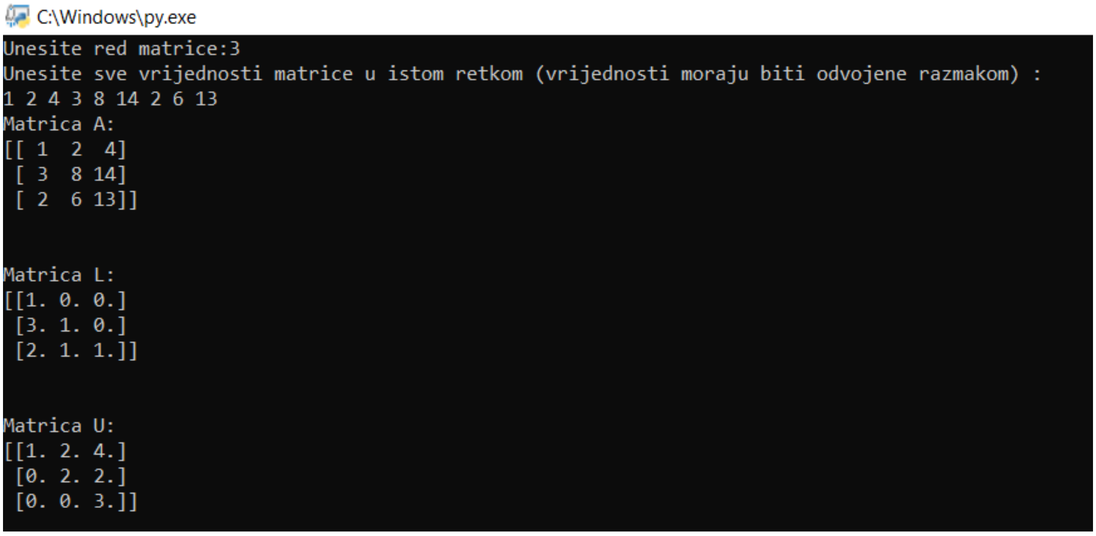
U poglavlju 5 smo ukratko opisali najvažnija svojstva programskog jezika Python, a sad ćemo prikazati implementaciju LU dekompozicije.

```
import numpy as np
import math

np.seterr(divide='ignore', invalid='ignore')
red = int(input("Unesite red matrice:"))
redak=stupac=red

print("Unesite sve vrijednosti matrice u istom retkom (vrijednosti moraju biti
    odvojene razmakom) : ")

unos = list(map(int, input().split()))

A = np.array(unos).reshape(redak, stupac)
print("Matrica A:")
print(A)
print("\n")

def luDekompozicija(A):
    U = np.zeros((len(A),len(A)))
    L = np.eye(redak)
    for i in range(len(A)):
        for j in range(i, len(A)):
            zbroj = 0
            for k in range(i):
                zbroj += L[i][k] * U[k][j]
            U[i][j] = A[i][j] - zbroj

        for j in range(i + 1, len(A)):
            zbroj = 0
            for k in range(i):
                zbroj += L[j][k] * U[k][i]
            L[j][i] = (A[j][i] - zbroj) / U[i][i]
            if(math.isnan(L[j][i])):
                L[j][i]=0
```

```
34     print("Matrica L:")
35     print (L)
36     print("\n")
37     print("Matrica U:")
38     print(U)
39
40 luDekompozicija(A)
```

U prethodnom poglavlju smo spomenuli da je jedna od najvećih prednosti Pythona veliki broj različitih biblioteka. Jedna od takvih biblioteka je i **NumPy**. NumPy biblioteka omogućuje manipulaciju višedimenzionalnim nizovima i matricama. Druga važna biblioteka koju koristimo je **math** koja u sebi sadrži ugrađene matematičke funkcije. Da bi mogli koristiti biblioteke potrebno ih je instalirati preko naredbenog retka (cmd) u mapu gdje je instaliran Python. Da bi instalirali biblioteku NumPy koristimo naredbu **python -m pip install numpy**. Sad kad smo pripremili okruženje možemo krenuti s tumačenjem programskog koda. Prvim dvjema linijama uključujemo biblioteke NumPy i math u programsko rješenje. Naredbom iz linije 4 onemogućujemo ispis greške kada neki broj djelimo sa 0. Taj slučaj će biti kasnije opisan. Naredbe iz linija 5-15 služe za definiranje dimenzija matrice A i za njezin ispis. Naredbom u liniji 5 tražimo od korisnika da unese red matrice. Zatim u sljedećoj liniji pridružujemo vrijednost varijabli $redak$ i $stupac$ koja odgovara vrijednosti varijable red. Linija 8 omogućuje unos vrijednosti od strane korisnika ovisno o redu matrice. Važno je napomenutu da se sve vrijednosti unose u istom redu i odvajaju se pomoću razmaka. Naredbom **map** iz linije 10 kreiramo listu čije su vrijednosti odvojene zarezom na temelju unesenih vrijednosti iz linije 8. Linija 12 služi za kreiranje matrice A dimenzija vrijednosti pridruženih varijablama $redak$ i $stupac$. Linije 13-15 služe za ispis matrice A. Sama LU dekompozicija je definirana između linija 17-38. U liniji 17 definiramo funkciju koja kao parametar prima matricu A i ispisuje matrice U i L nakon što ih popunimo odgovarajućim vrijednostima. Na početku kreiramo varijablu U koja ima samo 0. Za kreiranje matrice koja ima samo 0 koristimo naredbu **Np.zeros**. Pomoću te naredbe kreira se matrica s brojem redaka i stupaca koji odgovaraju broju redaka i stupaca matrice A. Matricu L definiramo kao matricu koja na dijagonalama ima 1, a na ostalim pozicijima ima 0. Za to koristimo naredbu **np.eye**. Sad kad smo kreirali početni izgled matrica L i U možemo krenuti sa zamjenom 0 sa odgovarajućim vrijednostima ovisno o kojoj se matrici radi (L ili U). Prvo ćemo objasniti kako se dobiju vrijednosti za matricu U. Za izračun vrijednosti matrice U koristimo 3 for petlje. Prva petlja ima iterator i koji se povećava tako dugo dok ne dođe do duljine matrice A (u našem slučaju 3). Druga for petlja ima iterator j koji je u rasponu između vrijednosti iteratora i i dimenzije matrice A. Postavljamo varijablu **zbroj** na 0. Varijabla $zbroj$ služi za dobivanje vrijednosti koju oduzimamo od vrijednosti koja se nalazi na toj poziciji u matrici A. Na kraju imamo petlju koja ima iterator k koji se izvodi sve dok ne poprimi vrijednost iteratora i. Ulaskom u petlju k varijabla $zbroj$ se povećava na temelju umnoška matrice $L[i][k]$ i matrice $U[k][j]$. Nakon što izračunamo varijablu $zbroj$ koristimo ju za dobivanje vrijednosti elementa matrice U koji se nalazi na poziciji i, j. Sad prelazimo na izračun matrice L, osim petlje koja ima iterator i, za izračun se koristi petlja j koja je sad u rasponu od $i + 1$ i dimenzije matrice A. Postavljamo $zbroj$ na 0 i iteriramo kroz petlju s iteratorom k koji je u rasponu iteratora i. Izračunavamo $zbroj$ prema formuli $L[j][k] \times U[k][i]$. Sad kad nam je poznata vrijednost varijable $zbroj$ možemo dobiti vrijednost elementa matrice L koji se nalazi na poziciji $L[j][i]$ na temelju formule $(A[j][i] - zbroj)/U[i][i]$. Na kraju provjeravamo da li je izračunata vrijednost jednaka nan-u (zbog dijeljenja s 0 postoji ta mogućnost). Ako imamo takvu vrijednost postavljamo ju na 0. Naredbe u liniji 34 - 38 služe za ispis matrica L i U. U liniji 40 imamo poziv funkcije.

Možemo primijetiti da kod prvog prolaska kroz petlju s iteratorom i ne ulazimo u petlju s iteratorom k jer je iterator $i = 0$. Zbog toga vrijednosti prvog reda matrice U odgovaraju vrijednostima matrice A. Dok se vrijednosti za matricu L ne računaju za prvi red. One ostaju identične početnoj definiranoj matrici L (1 u prvom stupcu, a drugim stupcima 0). U ostalim iteracijama kroz petlju s iteratorom i ulazi se u petlju s iteratorom k i računa se varijabla $zbroj$ prema zadanoj formuli iz linija 24 i 30.



```
C:\Windows\py.exe
Unesite red matrice:3
Unesite sve vrijednosti matrice u istom retkom (vrijednosti moraju biti odvojene razmakom) :
1 2 4 3 8 14 2 6 13
Matrica A:
[[ 1  2  4]
 [ 3  8 14]
 [ 2  6 13]]


Matrica L:
[[1. 0. 0.]
 [3. 1. 0.]
 [2. 1. 1.]]


Matrica U:
[[1. 2. 4.]
 [0. 2. 2.]
 [0. 0. 3.]]
```

Slika 2: Matrice L i U dobivene na temelju Python programa opisanog iznad slike

Na temelju izračuna koji smo napravili u poglavlju 6.1 možemo vidjeti da su matrice L i U dobivene pomoću programa napisanog u Pythonu identične matricama L i U dobivenim u poglavlju 6.1.

7. Cholesky dekompozicija

Cholesky metoda se koristi za rješavanje sustava linearnih jednadžbi pri tome matrica sustava mora biti simetrična. Cholesky dekompozicija je rastav simetrične matrice A na umnožak donjetrokutaste matrice i njezine transponirane gornjetrokutaste matrice pri tome matrica A mora biti **pozitivno definitna**. Pozitivno definitna matrica A je ona matrica kojoj za sve $x \in \mathbb{R}^n$, $x \neq 0$ vrijedi sljedeće

$$x^T A x > 0$$

Ako za primjer uzmemo $x = e_i$ (i-ti stupac jedinične matrice), onda je $a_{ii} = e_i^T A e_i > 0$. Odnosno dijagonalni elementi pozitivno definitne matrice su uvijek pozitivni. Pretpostavimo da je S bilo koja regularna matrica i $x \neq 0$ onda je $y = Sx \neq 0$ i vrijedi

$$x^T(S^T A S)x = (Sx)^T A(Sx) = y^T S y > 0.$$

Iz toga zaključujemo da je i $S^T AS$ pozitivno definitna matrica.[3]

Cholesky dekompozicijom možemo zapisati matricu A kao $A = LL^T$.

$$A = LL^T = \begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots & \vdots \\ l_{1,n-1} & l_{2,n-1} & l_{3,n-1} & \dots & 0 \\ l_{1,n} & l_{2,n} & l_{3,n} & \dots & l_{n,n} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & \dots & l_{1,n-1} & l_{1,n} \\ 0 & l_{22} & \dots & l_{2,n-1} & l_{2,n} \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots \\ 0 & 0 & \dots & l_{n-1,n-1} & l_{n-1,n} \\ 0 & 0 & \dots & 0 & l_{n,n} \end{bmatrix}$$

Za izračun nepoznatih elemenata matrice L i L^T koristimo sljedeće formule:

$$l_{11} = \sqrt{a_{11}}$$

$$l_{jj} = \sqrt{a_{jj} - \sum_{s=1}^{j-i} l_{js}^2}, \qquad j = 2, ..., n$$

$$l_{jl} = \tfrac{a_{jl}}{l_{11}}, \qquad j = 2, ..., n$$

$$l_{jk} = \tfrac{1}{l_{kk}}(a_{jk} - \sum_{s=1}^{k-i} l_{js} l_{ks}), \qquad j = k+1, ..., n, \qquad k \geq 2$$

Rješavamo sustav $Ly = b$, iz tog sustava dobivamo y korištenjem supstitucije unaprijed. Kad dobijemo y možemo dobiti x na temelju formule $L^T x = y$ supstitucijom unatrag. x-evi pretstavljaju rješenje sustava.

7.1. Primjer Cholesky dekompozicije

Uzmimo za primjer matrice:

$$A = \begin{bmatrix} 4 & 10 & 8 \\ 10 & 26 & 26 \\ 8 & 26 & 61 \end{bmatrix}, b = \begin{bmatrix} 44 \\ 126 \\ 214 \end{bmatrix}$$

Matricu A zapisujemo kao umnožak matrice L i L^T.

$$A = LL^T = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

Korištenjem formula koje smo naveli u prethodnoj sekciji dobivamo sljedeće vrijednosti

$$l_{11} = \sqrt{a_{11}} \Rightarrow \sqrt{4} \Rightarrow l_{11} = 2$$

$$l_{21} = \tfrac{a_{21}}{l_{11}} \Rightarrow \tfrac{10}{2} \Rightarrow l_{21} = 5$$

$$l_{22} = \sqrt{a_{22} - l_{21}^2} \Rightarrow \sqrt{26 - 5^2} \Rightarrow l_{22} = 1$$

$$l_{31} = \tfrac{a_{31}}{l_{11}} \Rightarrow \tfrac{8}{2} \Rightarrow l_{31} = 4$$

$$l_{32} = \tfrac{1}{l_{22}}(a_{32} - l_{31}l_{21}) \Rightarrow \tfrac{1}{1}(26 - 4 \cdot 5) \Rightarrow l_{32} = 6$$

$$l_{33} = \sqrt{a_{33} - l_{31}^2 - l_{32}^2} \Rightarrow \sqrt{61 - 4^2 - 6^2} \Rightarrow l_{33} = 3$$

Sad možemo matricu A zapisati kao LL^T s time da su nam sad poznate nepoznate vrijednosti elemenata matrice L.

$$A = LL^T = \begin{bmatrix} 4 & 10 & 8 \\ 10 & 26 & 26 \\ 8 & 26 & 61 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 5 & 1 & 0 \\ 4 & 6 & 3 \end{bmatrix} \begin{bmatrix} 2 & 5 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 3 \end{bmatrix}$$

Trebamo dobiti y iz jednadžbe $Ly = b$

$$\begin{bmatrix} 2 & 0 & 0 \\ 5 & 1 & 0 \\ 4 & 6 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 44 \\ 128 \\ 214 \end{bmatrix}$$

$$2y_1 = 44 \Rightarrow y_1 = 22$$

$$5y_1 + y_2 = 128 \Rightarrow 5 \cdot 22 + y_2 = 128 \Rightarrow y_2 = 18$$

$$4y_1 + 6y_2 + 3y_3 = 214 \Rightarrow 4 \cdot 22 + 6 \cdot 18 + 3y_3 = 214 \Rightarrow y_3 = 6$$

Sad kad su nam poznati y možemo dobiti x-eve iz jednadžbe $L^T x = y$

$$\begin{bmatrix} 2 & 5 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 22 \\ 18 \\ 6 \end{bmatrix}$$

$$3x_3 = 6 \Rightarrow x_3 = 2$$

$$x_2 + 6x_3 = 18 \Rightarrow x_2 + 6 \cdot 2 = 18 \Rightarrow x_2 = 6$$

$$2x_1 + 5x_2 + 4x_3 = 22 \Rightarrow 2x_1 + 5 \cdot 6 + 4 \cdot 2 = 22 \Rightarrow x_1 = -8$$

Time smo dobili da je rješenje ovog sustava $x =$ (-8, 6, 2).

7.2. LDL^T dekompozicija

Korištenjem Cholesky dekompozicije susrećemo se s problemom korjenovanja koji otežava izračun potrebnih vrijednosti. Za izbjegavanje korjenovanja koristi se Cholesky dekompozicija bez korijena poznatija pod nazivom LDL^T dekompozicija. Matricu A možemo zapisati kao LDL^T pri tome matrica L predstavlja donjetrokutastu matricu, L^T predstavlja transponiranu donjetrokutastu matricu dok D predstavlja dijagonalnu matricu. Prednost ove metode je da se može koristiti za neke slučaje kada ne možemo koristiti Cholesky dekompoziciju. Uzmimo za primjer matricu oblika 3×3. Korištenjem ove metode imamo sljedeći oblik matrica $A =$ LDL^T .

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \begin{bmatrix} 1 & l_{12} & l_{13} \\ 0 & 1 & l_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Nepoznate elemente matrice D dobivamo na temelju formule:

$$d_{jj} = a_{jj} - \sum_{k=1}^{j-i} l_{jk}^2 d_{kk}$$

Nepoznate elemente matrice L dobivamo na temelju formule:

$$l_{ij} = (a_{ij} - \sum_{k=1}^{j-i} l_{ik} d_{kk} l_{jk}) \times \frac{1}{d_{jj}}$$

Kad znamo to možemo zapisati jednadžbu

$$Ax = y \text{ kao } LDL^T x = y.$$

Možemo definirati

$$L^T x = y, \qquad x_i = y_i - \sum_{k=i+1}^{n} l_{ki} x_k \qquad za \qquad i = n, n-1, ..., 2, 1$$

Također definiramo

$$Dy = z, \qquad y_i = \frac{z_i}{d_{ii}} \qquad za \qquad i = 1, 2, ..., n$$

Početna jednadžba je

$$Lz = b, \qquad z_i = b_i - \sum_{k=1}^{i-1} l_{ik} z_k \qquad za \qquad i = 1, 2, ..., n$$

Možemo uočiti da prvo trebamo dobiti z iz $Lz = b$ dok nam je poznati z možemo dobiti y iz $Dy = z$ i na kraju dobivamo x iz $L^T x = y$ [12].

7.2.1. Primjer LDL^T dekompozicije

Uzmimo za primjer

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Računamo vrijednosti elemenata D_{ij} i L_{ij} prema formulama koje smo ranije definirali.

$$D_{11} = A_{11} = 2$$

$$L_{21} = \frac{A_{21}}{D_{11}} = \frac{-1}{2}$$

$$L_{31} = \frac{A_{31}}{D_{11}} = \frac{0}{2} = 0$$

$$D_{22} = A_{22} - L_{21}^2 D_{11} = 2 - (\frac{-1}{2})^2(2) = \frac{3}{2}$$

$$L_{32} = \frac{A_{32} - L_{31} D_{11} L_{21}}{D_{22}} = \frac{-1 - (0)(2)(\frac{-1}{2})}{\frac{3}{2}} = \frac{-2}{3}$$

$$D_{33} = A_{33} - L_{31}^2 D_{11} - L_{32}^2 D_{22} = 1 - (0)^2(2) - (\frac{-2}{3})^2(\frac{3}{2}) = \frac{1}{3}$$

Dobili smo sljedeće matrice:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{-1}{2} & 1 & 0 \\ 0 & \frac{-2}{3} & 1 \end{bmatrix}, D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & \frac{3}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$$

Izračunamo nepoznanicu z iz izraza $Lz = b$

$$Lz = b = \begin{bmatrix} 1 & 0 & 0 \\ \frac{-1}{2} & 1 & 0 \\ 0 & \frac{-2}{3} & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$z_1 = 1$$

$$\frac{-1}{2} z_1 + z_2 = 0 \Rightarrow z_2 = \frac{1}{2}$$

$$\frac{-2}{3} z_2 + z_3 = 0 \Rightarrow \frac{-2}{3} \cdot \frac{1}{2} + z_3 = 0 \Rightarrow z_3 = \frac{1}{3}$$

Zatim računamo nepoznanicu y iz izraza $Dy = z$

$$Dy = z = \begin{bmatrix} 2 & 0 & 0 \\ 0 & \frac{3}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \end{bmatrix}$$

$$2y_1 = 1 \Rightarrow y_1 = \frac{1}{2}$$

$$\frac{3}{2} y_2 = \frac{1}{2} \Rightarrow y_2 = \frac{1}{3}$$

$$\frac{1}{3} y_3 = \frac{1}{3} \Rightarrow y_3 = 1$$

Na kraju dobimo nepoznanicu x supstitucijom unatrag iz izraza $L^T x = y$

$$L^T x = y = \begin{bmatrix} 1 & \frac{-1}{2} & 0 \\ 0 & 1 & \frac{-2}{3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{3} \\ 1 \end{bmatrix}$$

$$x_3 = 1$$

$$x_2 - \frac{2}{3} x_3 = \frac{1}{3} \Rightarrow x_2 - \frac{2}{3} \cdot 1 = \frac{1}{3} \Rightarrow x_2 = 1$$

$$x_1 - \frac{1}{2} x_2 = \frac{1}{2} \Rightarrow x_1 - \frac{1}{2} \cdot 1 = \frac{1}{2} \Rightarrow x_1 = 1$$

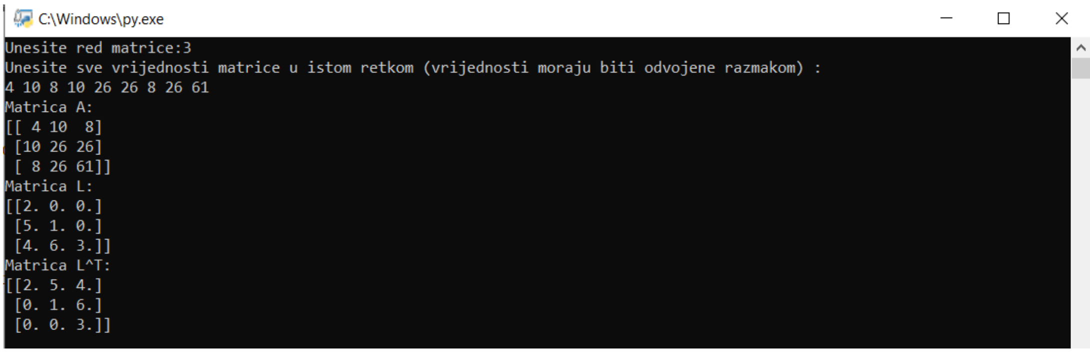
Time smo dobili da je rješenje $x =$ (1, 1, 1).

7.3. Implementacija Cholesky dekompozicije u Pythonu

U ovom dijelu ćemo se upoznati sa implementacijom Cholesky dekompozicije u Pythonu. Neki dijelovi koda su identični dijelovima LU dekompozicije pa nećemo trošiti previše vremena na njihovo objašnjavanje jer su već objašnjenji u poglavlju Implementacija LU dekompozicije u Pythonu.

```
import numpy as np
red = int(input("Unesite red matrice:"))
redak=stupac=red

print("Unesite sve vrijednosti matrice u istom retkom (vrijednosti moraju biti
    odvojene razmakom) : ")

unos = list(map(int, input().split()))

A = np.array(unos).reshape(redak, stupac)
print("Matrica A:")
print(A)

def cholesky(A):
    n = len(A)
    L = np.zeros((n, n))

    for i in range(n):
        for j in range(i + 1):
            if i == j:
                zbroj = sum(L[i][k] ** 2 for k in range(j))
                L[i][j] = np.sqrt(A[i][i] - zbroj)
            else:
                zbroj = sum(L[i][k] * L[j][k] for k in range(j))
                L[i][j] = (A[i][j] - zbroj) / L[j][j]

    return L


def simetricna(A):
    for i in range(len(A)):
        for j in range(len(A)):
            if (A[i][j] != A[j][i]):
                return False
    return True

if (simetricna(A)):
    if(np.all(np.linalg.eigvals(A) > 0)):
        L = cholesky(A)
        Ltransponirana=L.transpose()
        print("Matrica L:")
        print(L)
        print("Matrica L^T:")
        print(Ltransponirana)
    else:
        print("Matrica nije pozitivno definitna")

else:
    print("Matrica nije simetricna")
```

Naredbe u linijama od 1 do 11 služe za definiranje dimenzija matrice, unos vrijednosti matrica i ispis matrice A. U liniji 13 definiramo funkciju za izvođenje cholesky algoritma. U liniji 14 definiramo varijablu n koja sadrži veličinu matrice. U liniji 15 kreiramo matricu L koja ima sve 0 i veličine je varijable n. Zatim imamo dvije for petlje jedna s iteratorom j i jedna s iteratorm i. Kada je uvjet $i == j$ ispunjen (kada imamo vrijednost glavne dijagonale) izvode se linije 20 i 21. U liniji 20 koristimo funkciju **sum** koja zbraja sve vrijednosti kvadrata $L[i][k]$ koji zadovoljavaju petlju. U liniji 21 dobivamo vrijednost elementa $L[i][j]$ korištenjem **np.sqrt** i vrijednosti $A[i][i]$ i varijable $zbroj$ koju smo dobili u prethodnoj liniji. Sqrt vraća drugi korijen. Kada je $i \in j$ tada $zbroj$ dobivamo kao $L[i][k] + L[j][k]$ sve dok je k u rasponu varijable j. Vrijednost $L[i][j]$ dobivamo kao $(A[i][j] - zbroj)/L[j][j]$. Na kraju funkcije vraćamo matricu L. Linija 29-34 služi za provjeru simetričnosti matrice. Ako $A[i][j]! = A[j][i]$ vraćamo false inače vraćamo true. U liniji 36 pozivamo funkciju $simetricna$ za matricu A. Ako matrica A nije simetrična ispišemo poruku da nije simetrična u suprotno provjeravamo da li je pozitivno definitna. U liniji 37 se nalazi naredba za provjeru pozitivne definitnosti (**np.all(np.linalg.eigvals(A) > 0**). Ako matrica A nije pozitivno definitna ispisujemo poruku u liniji 45, suprotno se izvode naredbe u liniji 38-43. U liniji 38 pozivamo funkciju cholesky koja vraća matricu L. U liniji 39 kreiramo matricu $Ltransponirana$ koja je transponirana matrica matrice L. Za kreiranje transponirane matrice koristimo naredbu **L.transpose**. U linijama od 40-43 ispisujemo matricu L i $Ltransponirana$.



Slika 3: Matrice L i L^T dobivene na temelju Python programa opisanog iznad slike (Cholesky dekompozicija)

Vidimo da je rezultat dobiven korištenjem Python programa identičan onome dobivenom izračunom kod primjera cholesky dekompozicije.

8. QR dekompozicija

QR dekompozicija je rastav matrice A na matrice Q i R. Matrica **R** je gornjetrokutasta matrica dok je matrica **Q** ortogonalna matrica. Prednost ove dekompozicije je da se može provoditi i na singularnim matricama. Pretpostavimo da imamo matricu $A \in \mathbb{R}^{m \times n}$ tada imamo ortogonalnu matricu Q oblika $m \times m$ i gornjetrokutastu matricu koja na dijagonalama ima elemente različite od 0. Osnovne metode za izračun QR dekompozicije su Gram-Schmidt metoda, Givensove rotacije i Householderove transformacije [9]. Ovdje je važno da se upoznamo s pojmovima vezanim uz ortogonalne matrice. **Ortogonalna matrica** je matrica koja pomnožena sa svojom transponiranom matricom daje jediničnu matricu. Također vrijedi i da je umnožak transponirane matrice sa zadanom matricom jedinična matrica. To nas dovodi do zaključka da inverz matrice mora odgovarati transponiranoj matrici zadane matrice. Odnosno vrijedi:

$$AA^T = A^T A = I, A^T = A^{-1}$$

Ortogonalne matrice imaju karakteristiku da je njihov broj redaka jednak broju stupaca. Determinanta takve matrice je uvijek +/- 1 [13].

8.1. QR dekompozicija koristeći Gram-Schmidt metodu

Da bismo mogli razumijeti Gram-Schmidtov postupak ortogonalizacije moramo definirati Euklidsku normu. **Euklidsku normu** definiramo kao:

$$||x|| = \sqrt{(x, x)} = \sqrt{\sum_{i=1}^{m} x_i^2}$$

Ako stupce matrice A označimo s a_i gdje je i oznaka stupca dobivamo sljedeći oblik matrice A

$$A = \begin{bmatrix} a_1 | a_2 | ... | a_n \end{bmatrix}$$

Tada vrijedi:

$$u_1 = a_1,\ e_1 = \frac{u_1}{||u_1||}$$
$$u_2 = a_2 - \frac{u_1^T a_2}{u_1^T u_1} u_1,\ e_2 = \frac{u_2}{||u_2||}$$

Općenito vrijedi:

$$u_{k+1} = a_{k+1} - \frac{u_1^T a_{k+1}}{u_1^T u_1} u_1 - ... - \frac{u_k^T a_{k+1}}{u_k^T u_k} u_k$$

Matricu Q dobivamo na sljedeći način:

$$Q = \begin{bmatrix} u_1 | & u_2 | & ... | & u_k \end{bmatrix}$$

Matricu R dobivamo na sljedeći način:

$$R = Q^T A$$

Faktorizacija matrice A tipa $m \times n$, $m \geq n$, oblika $A = QR$ gdje je Q matrica tipa $m \times n$ s ortonormiranim stupcima, a R gornjetrokutasta matrica tipa $n \times n$, naziva se **reducirana QR faktorizacija** matrice A. **Potpuna QR faktorizacija** matrice A je faktorizacija oblika $A = QR$ gdje je Q matrica tipa $m \times m$ s ortonormiranim stupcima, a R gornjetrokutasta matrica tipa $m \times n$. Potpuna QR faktorizacija se dobiva iz reducirane QR faktorizacije tako da se matrici Q doda $m - n$ ortonormiranih stupaca koji s ostalim stupcima čine ortonormiranu bazu u $A \in M_{mn}$, a matrici R se dodaju nul-reci kako bi se dopunila do matrice tipa $m \times n$ [10].

8.1.1. Primjer QR dekompozicije koristeći Gram-Schmidt metodu

Uzmimo za primjer matricu

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, a = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Korištenjem Gram-Schmidt metode dobivamo sljedeće:

$$u_1 = a = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$u_2 = b - \frac{u_1^T b}{u_1^T u_1} u_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \frac{\begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}}{\begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{-1}{2} \\ \frac{1}{2} \end{bmatrix}$$

$$u_3 = c - \frac{u_1^T c}{u_1^T u_1} u_1 - \frac{u_2^T c}{u_2^T u_2} u_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \frac{\begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}}{\begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - \frac{\begin{bmatrix} 1 & \frac{-1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}}{\begin{bmatrix} 1 & \frac{-1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{-1}{2} \\ \frac{1}{2} \end{bmatrix}} \begin{bmatrix} 1 \\ \frac{-1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{2}{3} \\ \frac{-2}{3} \end{bmatrix}$$

Trebamo podijeliti zadane vrijednosti s Euklidskom normom i normalizirati dobivene vektore

$$1.red \Rightarrow \sqrt{1^2 + 1^2} = \sqrt{2},\ 2.red \Rightarrow \sqrt{1^2 + \frac{-1}{2}^2 + \frac{1}{2}^2} = \frac{\sqrt{6}}{2},\ 3.red \Rightarrow \sqrt{\frac{2}{3}^2 + \frac{2}{3}^2 + \frac{-2}{3}^2} = \frac{2\sqrt{3}}{3}$$

$$u_1 = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}, u_2 = \frac{1}{\frac{\sqrt{6}}{2}} \cdot \begin{bmatrix} 1 \\ \frac{-1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{6}}{3} \\ \frac{-\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \end{bmatrix}, u_3 = \frac{1}{\frac{2\sqrt{3}}{3}} \cdot \begin{bmatrix} \frac{2}{3} \\ \frac{2}{3} \\ \frac{-2}{3} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \\ \frac{-\sqrt{3}}{3} \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & \frac{-\sqrt{6}}{6} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} & \frac{-\sqrt{3}}{3} \end{bmatrix}$$

Sada izračunamo R iz formule $R = Q^T A$

$$R = \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{6}}{3} & \frac{-\sqrt{6}}{6} & \frac{\sqrt{6}}{6} \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & \frac{-\sqrt{3}}{3} \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{6} \\ 0 & 0 & \frac{2\sqrt{3}}{3} \end{bmatrix}$$

Time smo dobili da je QR dekompozicija za zadanu matricu

$$A = QR = \begin{bmatrix} 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & \frac{-\sqrt{6}}{6} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} & \frac{-\sqrt{3}}{3} \end{bmatrix} \begin{bmatrix} \sqrt{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{6} \\ 0 & 0 & \frac{2\sqrt{3}}{3} \end{bmatrix}$$

8.2. QR dekompozicija koristeći Givensove rotacije

Važan pojam vezan uz Givensove rotacije je matrica rotacije. **Matrica rotacije** je matrica $A \in M_2$ za koju vrijedi.

$$A = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix}$$

pri tome je φ kut rotacije koji je suprotan smjeru kazaljke na satu. Matrica $A \in M_2$ ima 4 važna pravila:

1. Matrica A je regularna (ima determinantu 0)
2. Matrica A je ortogonalna (vrijedi $A^{-1} = A^T$)
3. Matrica A čuva Euklidsku normu vektora
4. Produkt matrica rotacija je opet matrica rotacija

Givensove rotacije $G \in M_2$ su matrice rotacija koje rotiraju elemente u ravnini tako da drugu komponentu postave na 0. Uzmimo za primjer vektor

$$x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \in M_{21}$$

Matrica G djeluje na njega:

$$Gx = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ 0 \end{bmatrix}$$

Dobivamo da je:

$$x_1 cos\varphi - x_2 sin\varphi = y_1 \; x_1 sin\varphi + x_2 cos\varphi = 0$$

Zaključujemo da je matrica G jednaka jediničnoj matrici I kada je $x_2 = 0$, odnosno kada nije 0 imamo:

$$x_2 cos\varphi = -x_1 sin\varphi$$

Korištenjem formule $sin^2\varphi + cos^2\varphi = 1$ i kvadriranjem dobivenog rezultata dobivamo

$$cos\varphi = \frac{x_1}{\sqrt{x_1^2+x_2^2}}$$

Želimo da y_1 bude pozitivan pa uzimamo samo pozitivno rješenje $cos\varphi$. Tada je $sin\varphi$:

$sin\varphi = \frac{-x_2}{\sqrt{x_1^2+x_2^2}}$, odnosno $y_1 = \frac{x_1^2+x_2^2}{\sqrt{x_1^2+x_2^2}}$ i dobivamo $y = \begin{bmatrix} y_1 & 0 \end{bmatrix}^T$

Konstruiramo matricu G koja je oblika

$$G(i, k\varphi) = \begin{bmatrix} 1 & ... & 0 & ... & 0 & ... & 0 \\ \vdots & \because & \vdots & \because & \vdots & \because & \vdots \\ 0 & ... & cos\varphi & ... & -sin\varphi & ... & 0 \\ \vdots & \because & \vdots & \because & \vdots & \because & \vdots \\ 0 & ... & sin\varphi & ... & cos\varphi & ... & 0 \\ \vdots & \because & \vdots & \because & \vdots & \because & \vdots \\ 0 & ... & 0 & ... & 0 & ... & 1 \end{bmatrix}^T$$

podmatrica

$$\begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix}$$

nalazi se na presjeku i-tog i k-tog retka i stupca, elementi na dijagonali imaju vrijednost 1 dok ostali elementi imaju 0. Vrijednosti $sin\varphi$ i $cos\varphi$ računaju se na sljedeći način:

$$sin\varphi = \frac{-x_k}{\sqrt{x_i^2+x_k^2}},\ cos\varphi = \frac{x_i}{\sqrt{x_i^2+x_k^2}}$$ [14]

Jednostavnije rečeno x_i (poglavlju Primjer QR dekompozicije koristeći Givensove rotacije označena s x) je vrijednost pivot elementa, a x_k (poglavlju Primjer QR dekompozicije koristeći Givensove rotacije označena s y) je vrijednost koju eliminiramo u određenoj iteraciji. Sad kada smo opisali osnovne pojmove možemo ukratko opisati postupak kod matrice 3×3. Uzimamo x kao pivot element koji se nalazi na poziciji 1,1, dok je y element kojeg želimo eliminirati u ovom slučaju element na poziciji 2,1. Računamo $cos\varphi$ (dalje ćemo ga zbog jednostavnosti označavati samo sa slovom c) i $sin\varphi$ (označavati ćemo ga sa s) prema formulama koje smo ranije naveli. Kreiramo matricu G_1 s vrijednostima c i s na pozicijama kombinacija 1 i 2 jer imamo 1,2 redak/stupac. Računamo A_2=$G_1 A$. Time smo postavili 0 na element koji se nalazi na poziciji 2,1. Sada moramo eliminirati element na poziciji 3,1. Vrijednosti elemenata x, y, c , s računamo na temelju matrice A_2. Kreiramo matricu G_2 s vrijednostima c i s na kombinacijama 1 i 3 jer imamo 1,3 redak/stupac, nakon toga računamo matricu $A_3 = G_2 A_2$. Sad moramo eliminirati element na poziciji 3,2. Ponavljamo postupak koji smo opisali do sad. G_3 ima vrijednosti c i s na kombinacijama 2 i 3 jer imamo 2,3 redak/stupac. Sad možemo izračunati R kao=$G_3 A_3$ jer smo eliminirali sve elemente koje smo trebali. Matricu Q dobivamo kao $Q = G_1^T G_2^T G_3^T$

8.2.1. Primjer QR dekompozicije koristeći Givensove rotacije

Uzet ćemo za primjer matricu A koja je identična matrici korištenoj za QR dekompoziciju pomoću Gram-Schmidt metode.

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Prvo računamo pripadajuće vrijednosti $x,\ y,\ r,\ s,\ c$ za matricu A

$$x = 0,\, y = 1,\, r = \sqrt{0^2 + 1^2} = 1,\, c = \tfrac{0}{1} = 0,\, s = \tfrac{-1}{1} = -1$$

Sad računamo vrijednosti za matricu G_1

$$G_1 = \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Računamo vrijednosti matrice $A_2 = G_1 A$

$$A_2 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

Računamo pripadajuće vrijednosti $x,\ y,\ r,\ s,\ c$ za matricu A_2

$$x = 1,\, y = 1,\, r = \sqrt{1^2 + 1^2} = \sqrt{2},\, c = \tfrac{1}{\sqrt{2}},\, s = \tfrac{1}{\sqrt{-2}} = -\tfrac{1}{\sqrt{2}}$$

Sad računamo vrijednosti za matricu G_2

$$G_2 = \begin{bmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s & 0 & c \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{-1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Računamo vrijednosti matrice $A_3 = G_2 A_2$

$$A_3 = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{-1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & -1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & -1 & -1 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

Računamo pripadajuće vrijednosti $x,\ y,\ r,\ s,\ c$ za matricu A_3

$$x = -1,\, y = \tfrac{1}{\sqrt{2}},\, r = \sqrt{(-1)^2 + (\tfrac{1}{\sqrt{2}})^2} = \tfrac{\sqrt{6}}{2},\, c = \frac{-1}{\frac{\sqrt{6}}{2}} = \frac{-\sqrt{6}}{3},\, s = \frac{\frac{1}{\sqrt{2}}}{\frac{\sqrt{6}}{2}} = -\frac{\sqrt{3}}{3}$$

Sad računamo vrijednosti za matricu G_3

$$G_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{-\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \\ 0 & \frac{-\sqrt{3}}{3} & \frac{-\sqrt{6}}{3} \end{bmatrix}$$

Računamo vrijednosti matrice $R = G_3 A_3$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{-\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \\ 0 & \frac{-\sqrt{3}}{3} & \frac{-\sqrt{6}}{3} \end{bmatrix} \begin{bmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & -1 & -1 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{6} \\ 0 & 0 & \frac{2\sqrt{3}}{3} \end{bmatrix}$$

Na kraju trebamo dobiti matricu Q uvrštavanjem u formulu $G_1^T G_2^T G_3^T$

$$Q = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{-\sqrt{6}}{3} & \frac{\sqrt{-3}}{3} \\ 0 & \frac{\sqrt{3}}{3} & \frac{-\sqrt{6}}{3} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & \frac{-\sqrt{6}}{6} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} & \frac{-\sqrt{3}}{3} \end{bmatrix}$$

Vidimo da smo dobili identično rješenje koje smo dobili i primjenom Gram-Schmidt metode pa možemo zaključiti da bez obzira na to koju metodu koristimo uvijek dobivamo isto rješenje.

8.3. QR dekompozicija koristeći Householderove transformacije

QR dekompozicija koristeći Householderove transformacije temelji se transformiranju matrice A u gornjetrokutastu matricu korištenjem transformacija koje se primjenjuju na stupce matrice A. Da bi mogli razumijeti Householderove transformacije uvodimo pojam **Householderove matrice**.

$$H = 1 - 2vv^T, \qquad v^T v = 1, \qquad v \in \mathbb{R}^m, \qquad H \in^{(m \times n)}$$

Matrica H je Householderova matrica. Vektor v nazivamo **Householderov vektor**. Vidimo da je matrica simetrična i ortogonalna jer vrijedi $H = H^T$. Vrijedi:

$$HH^T = (I - 2vv^T)(I - 2vv^T) = I - 4VV^T + 4V(V^T V)V^T = I$$

Time je dokazano da vrijedi

$$H^T H = HH^T = I$$

Važno svojstvo ove metode je da za dva vektora a i b jednake euklidske norme postoji Householderov vektor v, tako da je $H_a = b$, odnosno, zbog simetričnosti matrice H, $H_b = a$. Ovu tvrdnju dokazuje Scitovski [9, str 53.-54.] pa nećemo trošiti vrijeme na izvod tog dokaza. Ranije smo naveli da je osnovni cilj ove metode transformiranje matrice A na gornjetrokutastu matricu pa možemo njene stupce promatrati kao vektore M_{m1}. Neka je barem jedna od komponenti

$a_k, ..., a_m$ vektora a različita od nule za $k \geq 1$. Householderovu matricu H konstruirat ćemo tako da se prvih $k-1$ komponenti vektora b := H_a i vektora a podudara, a da se posljednjih $m-k$ komponenti vektora b poništi. Time dobivamo

$$(H_a)_i = b_i = \begin{cases} a_i \ za \ i = 1, ..., k-1 \\ +/- \ \lambda \ za \ i = k \\ 0 \ za \ i = k+1, ..., m \end{cases} \qquad , \lambda = \sqrt{a_k^2 + ... + a_m^2}$$

Vektor v ima oblik :

$$v_1 = \frac{u_1}{||u_1||}\text{[9][17]}$$

Odnosno u prvom koraku se eliminiraju svi elementi (postavljaju se na 0) koji se nalaze ispod pivot elementa prvog stupca. U drugom koraku se eliminiraju svi elementi koji se nalaze ispod pivot elementa drugog stupca i tako sve dok ne eliminiramo sve potrebne elemente. Prilikom svake eliminacije računamo matricu H_i prema formuli $I - 2v_iv_i^T$ i matricu R_i prema formuli $R_i = R_{i-1} \times H_i$. Kada eliminiramo sve potrebne elemente dobivamo matricu R tako da pomnožimo posljednje dobivenu matricu H s matricom R dobivenom u prethodnom koraku. Matricu Q dobivamo kao umnožak matrica H dobivenih iz svih prethodnih koraka.

8.3.1. Primjer QR dekompozicije koristeći Householderove transformacije

Uzet ćemo istu matricu A koju smo koristili u prethodnim primjerima QR dekompozicije.

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Prvo uzmemo prvi stupac matrice kao referentni stupac jer moramo dobiti 0 ispod prve vrijednosti prvog stupca i računamo $||x_1||$

$$x_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, ||x_1|| = \sqrt{0^2 + 1^2 + 1^2} = \sqrt{2}$$

Sad računamo u_1 prema formuli $u_1 = x_1 - ||x_1||e_1$

$$u_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - \sqrt{2} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\sqrt{2} \\ 1 \\ 1 \end{bmatrix}$$

Računamo v_1 tako da matricu u_1 podijelimo s $||u_1||$

$$||u_1|| = \sqrt{\sqrt{2}^2 + 1^2 + 1^2} = 2$$

$$v_1 = \frac{u_1}{||u_1||} = \begin{bmatrix} \frac{-\sqrt{2}}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

Sada računamo matricu H_1 prema formuli $I - 2v_1v_1^T$

$$H_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - 2 \cdot \begin{bmatrix} \frac{-\sqrt{2}}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} \frac{-\sqrt{2}}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{-1}{2} \\ \frac{\sqrt{2}}{2} & \frac{-1}{2} & \frac{1}{2} \end{bmatrix}$$

Odredimo R_1 i Q_1

$$R_1 = H_1A = \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{-1}{2} \\ \frac{\sqrt{2}}{2} & \frac{-1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{\sqrt{2}-1}{2} & \frac{1+\sqrt{2}}{2} \\ 0 & \frac{1+\sqrt{2}}{2} & \frac{\sqrt{2}-1}{2} \end{bmatrix}$$

$$Q_1 = H_1 = \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{-1}{2} \\ \frac{\sqrt{2}}{2} & \frac{-1}{2} & \frac{1}{2} \end{bmatrix}$$

Uzmemo drugi stupac matrice R_1 s tim da vrijednost prvog reda zamijenimo s 0 i računamo $||x_2||$

$$x_2 = \begin{bmatrix} 0 \\ \frac{\sqrt{2}-1}{2} \\ \frac{1+\sqrt{2}}{2} \end{bmatrix}, ||x_2|| = \sqrt{0^2 + (\frac{\sqrt{2}-1}{2})^2 + (\frac{1+\sqrt{2}}{2})^2} = \frac{\sqrt{6}}{2}$$

Sad računamo u_2 prema formuli $u_2 = x_2 - ||x_2||e_2$ zbog jednostavnosti brojeve ćemo zaokružiti na 4 decimale

$$u_2 = \begin{bmatrix} 0 \\ \frac{\sqrt{2}-1}{2} \\ \frac{1+\sqrt{2}}{2} \end{bmatrix} - \frac{\sqrt{6}}{2} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1.0176 \\ \frac{1+\sqrt{2}}{2} \end{bmatrix}$$

Računamo v_2 tako da matricu u_2 podijelimo s $||u_2||$

$$||u_2|| = \sqrt{0^2 + (-1.0176)^2 + (\frac{1+\sqrt{2}}{2})^2} = 1.5788$$

$$v_2 = \frac{u_2}{||u_2||} = \begin{bmatrix} 0 \\ -0.6446 \\ 0.7646 \end{bmatrix}$$

Sada računamo matricu H_2 prema formuli $I - 2v_2v_2^T$

$$H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - 2 \cdot \begin{bmatrix} 0 \\ -0.6446 \\ 0.7646 \end{bmatrix} \cdot \begin{bmatrix} 0 & -0.6446 & 0.7646 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1590 & 0.9867 \\ 0 & 0.9867 & -0.1590 \end{bmatrix}$$

Odredimo R_2 i Q_2

$$R_2 = H_2 R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1692 & 0.9857 \\ 0 & 0.9857 & -0.1692 \end{bmatrix} \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{-1}{2} \\ \frac{\sqrt{2}}{2} & \frac{-1}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1.4142 & 0.7071 & 0.7071 \\ 0 & 1.2248 & 0.4084 \\ 0 & 0 & 1.1548 \end{bmatrix}$$

$$Q_2 = H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1590 & 0.9867 \\ 0 & 0.9867 & -0.1590 \end{bmatrix}$$

Sad možemo odrediti matricu Q kao umnožak $Q_1 Q_2$

$$Q = \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{-1}{2} \\ \frac{\sqrt{2}}{2} & \frac{-1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1590 & 0.9867 \\ 0 & 0.9867 & -0.1590 \end{bmatrix} = \begin{bmatrix} 0 & 0.8167 & 0.5774 \\ 0.7071 & -0.4083 & 0.5774 \\ 0.7071 & 0.4083 & -0.5774 \end{bmatrix}$$

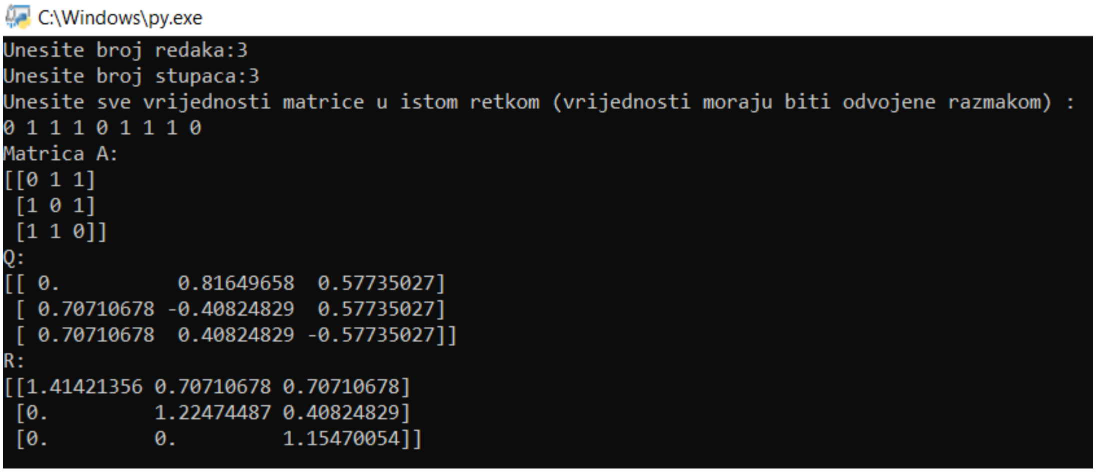
Vidimo da su vrijednosti identične dobivenim vrijednostima iz prethodnih primjera (postoji razlika u zadnjim decimalama zbog zaokruživanja). Vidimo da se u ovoj metodi susrećemo sa decimalnim brojevima pa zbog toga može biti problematična kod zaokruživanja brojeva kad je potrebna vrijednost koja je točna na decimalu .

8.4. Implementacija QR dekompozicije u Pythonu

```
import numpy as np

redak = int(input("Unesite broj redaka:"))
stupac=int(input("Unesite broj stupaca:"))
print("Unesite sve vrijednosti matrice u istom retkom (vrijednosti moraju biti
    odvojene razmakom) : ")
unos = list(map(int, input().split()))
A = np.array(unos).reshape(redak, stupac)

print("Matrica A:")
print(A)

def qrdekompozicija(A):
    m = len(A)
    n = len(A[0])
    Q = np.zeros((n, n), dtype=float)
    R = np.zeros((n, n), dtype=float)

    for j in range(n):
        q = A[:, j].astype(float)
        for i in range(j):
            R[i, j] = np.dot(Q[:, i], q)
            q = q - R[i, j] * Q[:, i]
        R[j, j] = np.linalg.norm(q)
        Q[:, j] = q / R[j, j]
    return Q, R
```

```
26
27 Q, R = qrdekompozicija(A)
28
29 print("Q:")
30 print(Q)
31 print("R:")
32 print(R)
```

Linije od 1-11 su već objašnjene u prethodnim poglavljima pa ćemo krenuti od linije 12 u kojoj definiramo funkciju **qrdekompozicija** koja kao parametar prima matricu A koja je prethodno definirana. U linijama 13-16 određujemo broj redaka i stupaca matrice i kreiramo matrice Q i R koje imaju isti broj redaka i stupaca kao što ima matrica A i popunjavamo ih nulama. U liniji 18 kreiramo petlju s iteratorom j, a linija 19 služi za dohvaćanje stupca po stupca matrice A ovisno o vrijednosti iteratora. U liniji 20 imamo petlju s iteratorom i unutar koje računamo vrijednosti matrice $R[i,j]$ i varijable q. Linija 23 služi za izračun vrijednosti dijagonala matrice R dok se u liniji 24 računa matrica Q stupac po stupac ovisno o vrijednosti itertora j. Funkcija $qrdekompozicija$ vraća matrice Q i R. Nakon što pozovemo navedenu funkciju i pridružimo odgovarajućim varijablama, ispišemo matrice Q i R.


```
C:\Windows\py.exe
Unesite broj redaka:3
Unesite broj stupaca:3
Unesite sve vrijednosti matrice u istom retkom (vrijednosti moraju biti odvojene razmakom) :
0 1 1 1 0 1 1 1 0
Matrica A:
[[0 1 1]
 [1 0 1]
 [1 1 0]]
Q:
[[ 0.          0.81649658  0.57735027]
 [ 0.70710678 -0.40824829  0.57735027]
 [ 0.70710678  0.40824829 -0.57735027]]
R:
[[1.41421356 0.70710678 0.70710678]
 [0.         1.22474487 0.40824829]
 [0.         0.         1.15470054]]
```

Slika 4: Matrice Q i R dobivene na temelju Python programa opisanog iznad slike (QR dekompozicija)

Vidimo da je rješenje dobiveno korištenjem Pythona identično rješenjima dobivenim u prethodnim izračunima QR dekompozicija (Neke se vrijednosti razlikuju od 4. decimale zbog zaokruživanja).

9. SVD dekompozicija

Dekompoziciju singularnih vrijednosti ili skraćeno SVD definiramo kao:

$$A = U\Sigma V^T \text{ za } A \in \mathbb{C}^{m\times n}$$

Matrica U je unitarna matrica oblika $m\times m$, matrica V je unitarna matrica oblika $n\times n$, a matrica Σ je oblika $m\times n$ zadane matrice A oblika $m\times n$. Matrice U i V moraju biti ortogonalne matrice dok matrica Σ mora biti dijagonalna matrica.

$$\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_{min(m,n)})$$

pri čemu mora vrijediti $\sigma_1 \geq \sigma_2 \geq \sigma_{min(m,n)} \geq 0.\sigma$ nazivamo **sigularnim vrijednostima** matrice A, a dobivamo ih korjenovanjem rješenja jednadžbe (λ). Stupce matrice U nazivamo **lijevim singularnim vektorima**, a stupce matrice V nazivamo **desnim singularnim vektorima**. Matrica Σ ima r singularnih vrijednosti na dijagonali dok su ostale vrijednosi 0, s toga stupci $r+1, ..., m$ matrice U i retci $r+1, ..., n$ matrice V^T ne pridonose umnošku. Uklananjem $m-r$ stupaca matrice U, $m-r$ null redaka matrice Σ te $n-r$ redaka matrice V^T dobivamo **reduciranu SVD dekompoziciju**. Prednost ove dekompozicije je da postoji za svaku matricu. Kada je $m \geq n$ SVD dekompozicija se provodi za matricu A, a slučaju kada ne vrijedi $m \geq n$ SVD dekompozicija se provodi za matricu A^T. Prvi korak SVD dekompozicije je množenje matrice A i njezine transponirana matrice. Kada je $m \geq n$ množimo matricu A i njezinu transponiranu matricu, a kada je m<n množimo transponiranu matricu matrice A s matricom A. Jedinični vektor u_1 dobivamo iz formule:

$$u_1 = \frac{Av_1}{\sigma_1}$$

Nadopunom u_1 s $m-1$ vektora do baze $\mathbb{C}^m$ i primjenom Gram-Schmidtovog postupka ortogonalizacije dobijemo bazu $u_1,\ u_2, ..., u_m$. Time smo dobili elemente matrice $U_1 = [u_1,\ u_2, ..., u_m]$. Slično za $v_1 = v$ postoji $n-1$ vektora $v_1,\ v_2, ..., vn \in \mathbb{C}$ takvih da je matrica $V_1 = [v_1,\ v_2, ..., v_n]$ tada je

$$A_1 = U_1^T A V_1 = \begin{bmatrix} \sigma_1 & 0 \\ 0 & A_2 \end{bmatrix}$$

Postupak ponavljamo za matricu $A_2 \in \mathbb{C}^{(m-1)\times(n-1)}$ i dobivamo matrice U i V kao produkt unitarnih matrica dobivenih nakon svakog koraka.

Neka je $A = U\Sigma V^T$ SVD dekompozicija matrice i neka su v_i i u_i oznake za i-te stupce matrica V i U tada vrijede sljedeće tvrdnje:

1.$r(A) = r$

2.$KerA = [v_{r+1}, ..., v_n]$

3.$ImA = [u_1, ..., u_r]$

4.$A = U_r \Sigma_r V_r$ gdje je Σ_r gornja lijeva podmatrica matrice Σ reda r, a U_r i V_r matrice sastavljene od prvih r stupaca matrica U i V.

5.$||A||_F^2 = \sigma_1^2 + ... + \sigma_r^2$

7.$||A||_2 = \sigma_1$
8.$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma\Sigma^T U^T$ isto vrijedi i za matricu $A^T A$
Dokaz tih tvrdnji nalazi se na [16, str.1.-2.]. Neka je $A = U\Sigma V^T$ i neka je $A \in \mathbb{R}^{m\times n}$ ranga r i $1 \leq k < r$ tada vrijedi:

$$min_{r(x)=k}||C - X||_2 = \sigma_{k+1}$$

Minimum se ostvaruje za matricu $X = U_k \Sigma_k V_k$ gdje je Σ_k gornja lijeva podmatrica matrice Σ reda k, a U_k i V_k matrice sastavljene od prvih k stupaca matrica U i V. Sad kad smo se upoznali s osnovnim pravilima možemo prijeći na primjer SVD dekompozicije.[15][16]

9.1. Primjer SVD dekompozicije

Uzet ćemo za primjer matricu

$$A = \begin{bmatrix} 2 & -2 & 1 \\ -4 & -8 & -8 \end{bmatrix}$$

Tada je

$$A^T = \begin{bmatrix} 2 & -4 \\ -2 & -8 \\ 1 & -8 \end{bmatrix}$$

Računamo $A^T A$

$$A^T A = \begin{bmatrix} 2\cdot 2 - 4\cdot(-4) & 2\cdot(-2) - 4\cdot(-8) & 2\cdot 1 - 4\cdot(-8) \\ -2\cdot 2 - 8\cdot(-4) & -2\cdot(-2) - 8\cdot(-8) & -2\cdot 1 - 8\cdot(-8) \\ 1\cdot 2 - 4\cdot(-8) & 1\cdot(-2) - 8\cdot(-8) & 1\cdot 1 - 8\cdot(-8) \end{bmatrix} = \begin{bmatrix} 20 & 28 & 34 \\ 28 & 68 & 62 \\ 34 & 62 & 65 \end{bmatrix}$$

Matricu $A^T A$ možemo zapisati kao

$$\begin{bmatrix} 20-\lambda & 28 & 34 \\ 28 & 68-\lambda & 62 \\ 34 & 62 & 65-\lambda \end{bmatrix}$$

Računamo λ iz $\lambda^3 - S_1\lambda^2 + S_2\lambda - S_3$

$$S_1 = 20 + 68 + 65 = 153$$

$$S_2 = \begin{vmatrix} 68 & 62 \\ 62 & 65 \end{vmatrix} + \begin{vmatrix} 20 & 34 \\ 34 & 65 \end{vmatrix} + \begin{vmatrix} 20 & 28 \\ 28 & 68 \end{vmatrix} = 576+144+576=1296$$

$$S_3 = det(A^T A) = 0$$

Uvrštavanjem izračunatih vrijednosti S dobivamo

$$\lambda^3 - 153\lambda^2 + 1296\lambda - 0 = 0$$

Rješavanjem ove kubne jednadžbe dobivamo da je

$$\lambda_1 = 144,\ \lambda_2 = 9,\ \lambda_3 = 0$$

Sada moramo naći matricu V. Prvo nalazimo matricu v_1 uvrštavanjem λ_1 u matricu A^TA

$$A^TA = \begin{bmatrix} 20-\lambda_1 & 28 & 34 \\ 28 & 68-\lambda_1 & 62 \\ 34 & 62 & 65-\lambda_1 \end{bmatrix} = \begin{bmatrix} 20-144 & 28 & 34 \\ 28 & 68-144 & 62 \\ 34 & 62 & 65-144 \end{bmatrix} \begin{bmatrix} -124 & 28 & 34 \\ 28 & -76 & 62 \\ 34 & 62 & -79 \end{bmatrix}$$

Sada rješavamo ovaj sustav korištenjem Gaussovih eliminacija

$$\begin{bmatrix} -124 & 28 & 34 \\ 28 & -76 & 62 \\ 34 & 62 & -79 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0 \Rightarrow \begin{bmatrix} -124 & 28 & 34 \\ 0 & \frac{-2160}{31} & \frac{2160}{31} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\tfrac{-2160}{31}x_2 + \tfrac{2160}{31}x_3 = 0 \Rightarrow \tfrac{-2160}{31}x_2 = \tfrac{-2160}{31}x_3$$

Moramo odabrati vrijednosti tako da vrijedi jednakost iznad. Uzmimo da je $x_3 = -2$ tada je i $x_2 = -2$ jer vrijedi $\frac{-2160}{31}x_2 = \frac{-2160}{31}x_3$. Iz prve jednadžbe dobivamo

$$-124x_1 + 28x_2 + 34x_3 = 0 \Rightarrow -124x_1 = 56 + 68 \Rightarrow x_1 = -1,\ \mathsf{v}_1 = \begin{bmatrix} -1 \\ -2 \\ -2 \end{bmatrix}$$

Sada nalazimo matricu v_2 uvrštavanjem λ_2 u matricu A^TA

$$A^TA = \begin{bmatrix} 20-\lambda_2 & 28 & 34 \\ 28 & 68-\lambda_2 & 62 \\ 34 & 62 & 65-\lambda_2 \end{bmatrix} = \begin{bmatrix} 20-9 & 28 & 34 \\ 28 & 68-9 & 62 \\ 34 & 62 & 65-9 \end{bmatrix} \begin{bmatrix} 11 & 28 & 34 \\ 28 & 59 & 62 \\ 34 & 62 & 56 \end{bmatrix}$$

Sada rješavamo ovaj sustav korištenjem Gaussovih eliminacija

$$\begin{bmatrix} 11 & 28 & 34 \\ 28 & 59 & 62 \\ 34 & 62 & 56 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0 \Rightarrow \begin{bmatrix} 11 & 28 & 34 \\ 0 & \frac{-135}{11} & \frac{-270}{11} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\tfrac{-135}{11}x_2 + \tfrac{-270}{11}x_3 = 0 \Rightarrow \tfrac{-135}{11}x_2 = \tfrac{270}{11}x_3$$

Uzmimo da je $x_3 = 1$ tada je i $x_2 = -2$ jer vrijedi $\frac{-135}{11}x_2 = \frac{270}{11}x_3$. Iz prve jednadžbe dobivamo

$$11x_1 + 28x_2 + 34x_3 = 0 \Rightarrow 11x_1 = 56 - 34 \Rightarrow x_1 = 2,\ \mathsf{v}_2 = \begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix}$$

Sada nalazimo matricu v_3 uvrštavanjem λ_3 u matricu A^TA

$$A^TA = \begin{bmatrix} 20-\lambda_3 & 28 & 34 \\ 28 & 68-\lambda_3 & 62 \\ 34 & 62 & 65-\lambda_3 \end{bmatrix} = \begin{bmatrix} 20-0 & 28 & 34 \\ 28 & 68-0 & 62 \\ 34 & 62 & 65-0 \end{bmatrix} \begin{bmatrix} 20 & 28 & 34 \\ 28 & 68 & 62 \\ 34 & 62 & 65 \end{bmatrix}$$

Sada rješavamo ovaj sustav korištenjem Gaussovih eliminacija

$$\begin{bmatrix} 20 & 28 & 34 \\ 28 & 68 & 62 \\ 34 & 62 & 65 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0 \Rightarrow \begin{bmatrix} 20 & 28 & 34 \\ 0 & \frac{144}{5} & \frac{72}{5} \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\tfrac{144}{5} x_2 + \tfrac{72}{5} x_3 = 0 \Rightarrow \tfrac{144}{5} x_2 = \tfrac{-72}{5} x_3$$

Uzmimo da je $x_2 = -1$ tada je i $x_3 = 2$ jer vrijedi $\frac{144}{5} x_2 = \frac{-72}{5} x_3$. Iz prve jednadžbe dobivamo

$$20x_1 + 28x_2 + 34x_3 = 0 \Rightarrow 20x_1 = 28 - 68 \Rightarrow x_1 = -2, \text{v}_3 = \begin{bmatrix} -2 \\ -1 \\ 2 \end{bmatrix}$$

Matricu V pretvorimo u V^T

$$V = \begin{bmatrix} -1 & 2 & -2 \\ -2 & -2 & -1 \\ -2 & 1 & 2 \end{bmatrix}, V^T = \begin{bmatrix} -1 & -2 & -2 \\ 2 & -2 & 1 \\ -2 & -1 & 2 \end{bmatrix}$$

Normaliziramo matricu i prebacimo ju u Euklidsku normu

1.red=$\sqrt{(-1^2) + 2^2 + (-2^2)} = 3$, 2.red=$\sqrt{2^2 + (-2)^2 + 1^2} = 3$, 3.red=$\sqrt{(-2)^2 + 1^2 + 2^2} = 3$

$$V^T = \begin{bmatrix} \frac{-1}{3} & \frac{-2}{3} & \frac{-2}{3} \\ \frac{2}{3} & \frac{-2}{3} & \frac{1}{3} \\ \frac{-2}{3} & \frac{-1}{3} & \frac{2}{3} \end{bmatrix}$$

Računamo Σ

$$\Sigma = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{144} & 0 & 0 \\ 0 & \sqrt{9} & 0 \end{bmatrix} = \begin{bmatrix} 12 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$$

Sada trebamo dobiti matricu U. Prvo računamo u_1 zatim u_2. u_3 ne postoji jer $\lambda_3 = 0$

$$u_1 = \frac{1}{\sqrt{\lambda_1}} A v_1 \Rightarrow \frac{1}{12} \begin{bmatrix} 2 & -2 & 1 \\ -4 & -8 & -8 \end{bmatrix} \begin{bmatrix} \frac{-1}{3} \\ \frac{-2}{3} \\ \frac{-2}{3} \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{1}{6} & \frac{-1}{6} & \frac{1}{12} \\ \frac{-1}{3} & \frac{-2}{3} & \frac{-2}{3} \end{bmatrix} \begin{bmatrix} \frac{-1}{3} \\ \frac{-2}{3} \\ \frac{-2}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$u_2 = \frac{1}{\sqrt{\lambda_2}} A v_2 \Rightarrow \frac{1}{3} \begin{bmatrix} 2 & -2 & 1 \\ -4 & -8 & -8 \end{bmatrix} \begin{bmatrix} \frac{2}{3} \\ \frac{-2}{3} \\ \frac{1}{3} \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{2}{3} & \frac{-2}{3} & \frac{1}{3} \\ \frac{-4}{3} & \frac{-8}{3} & \frac{-8}{3} \end{bmatrix} \begin{bmatrix} \frac{2}{3} \\ \frac{-2}{3} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, U = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Rješenje SVD dekompozicije za ovaj primjer je

$$A = U \Sigma V^T = \begin{bmatrix} 2 & -2 & 1 \\ -4 & -8 & -8 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 12 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} \frac{-1}{3} & \frac{-2}{3} & \frac{-2}{3} \\ \frac{2}{3} & \frac{-2}{3} & \frac{1}{3} \\ \frac{-2}{3} & \frac{-1}{3} & \frac{2}{3} \end{bmatrix}$$

9.1.1. Implementacija SVD dekompozicije u Pythonu

Implementirati ćemo programski kod koji radi SVD dekompoziciju za matricu iz primjera 9.1. Postoji ugrađena funkcija **scipy.linalg.svd** koja unutar sebe ima ugrađenu logiku za svd dekompoziciju pa ćemo mi zbog jednostavnosti pokazati samo implementaciju svd dekompozicije za matricu iz primjera 9.1. (2×3) dok se za ostale dekompozicije može koristiti gore navedena funkcija.

```
1  import numpy as np
2  np.seterr(divide='ignore', invalid='ignore')
3
4  def svddekompozicija(matrix):
5      ATA = np.dot(matrix.T, matrix)
6
7      eigenvalues, eigenvectors = np.linalg.eig(ATA)
8      eigenvalues = np.round(eigenvalues).astype(float)
9      sortirane_vrijednosti = np.argsort(eigenvalues)[::-1]
10     eigenvalues = eigenvalues[sortirane_vrijednosti]
11     eigenvectors = eigenvectors[:,sortirane_vrijednosti]
12
13     val = np.sqrt(np.abs(eigenvalues))
14     val = np.nan_to_num(val)
15     sigma = np.diag(val[:matrix.shape[1]])
16     sigma = sigma[~np.all(sigma == 0, axis=1)]
17     Vt = eigenvectors.T
18
19     red=len(matrix)
20     val = np.sqrt(np.abs(eigenvalues))
21     val=np.trim_zeros(val)
22     U = np.zeros((red, red))
23     V= eigenvectors
24     for i in range(red):
25         vi = V[:, i]
26         ui=(1/val[i])*(np.dot(matrix, vi))
27         U[:, i] = ui
28     U = np.round(U).astype(float)
29
30     return  U, sigma, Vt
31
32 A = np.array([[2, -2, 1],
33               [-4, -8, -8]])
34 svddekompozicija(A)
35 U, sigma, Vt = svddekompozicija(A)
36
37 print("U:")
38 print(U)
39 print(":")
40 print(sigma)
41 print("V^T:")
42 print(Vt)
```

Krenut ćemo od linije 5 jer smo prethodne linije objasnili u prethodnim poglavljima. U liniji

5 množimo transponiranu matricu A i matricu A. Za množenje koristimo funkciju **np.dot()**. U liniji 7 računamo λ vrijednosti i matricu V. Za izračun koristimo funkciju **np.linalg.eig()**. Python ponekad vraća rezultat u znastvenom zapisu da bi taj zapis pretvorili u normalni broj koristimo **np.round(eigenvalues).astype(float)**. Ova linija koda je suvišna kada Python ne vraća znastveni zapis broja. U liniji 9 sortiramo vrijednosti dobivene u liniji 7 pomoću funkcije **np.argsort()** i te sortirane vrijednosti onda pridružujemo korespondirajućim varijablama u liniji 10 i 11. Znamo da nam za izračun Σ treba drugi korijen od λ. Ta logika je implementirana u linijama 13 i 14 . Linije 15 i 16 služe za formiranje matrice Σ. U liniji 15 kreiramo matricu koja na dijagonalama ima vrijednosti varijable val dok su ostali elementi 0. Rješavanjem kubne jednadžbe za ovaj primjer dobivamo $\lambda_3 = 0$ to znači da je matrica Σ tipa 3×3, a znamo od ranije da treba biti tipa 2×3 pa da bismo obrisali zadnji red koristimo naredbu iz linije 16. U liniji 17 dobivamo transponiranu vrijednost matrice $eigenvectors$. Linije od 19 do 28 služe za izračun matrice U. Ranije smo vidjeli da matrica U za matricu oblika 2×3 mora biti 2×2 da bismo to dobili prvo uklanjamo 0 iz matrice val koja sadrži vrijednosti λ. Taj dio je implementiran u liniji 21 korištenjem funkcije **np.trim.zeros()**. U liniji 22 kreiramo matricu U koja je oblika (broj redova matrice A, broj redova matrice A). Sama logika izračuna vrijednosti matrice U je u linijama od 24-27. Matricu U računamo prema formuli $u_i = \frac{1}{\sqrt{\lambda_i}} A \times v_i$. Funkcija **svddekompozicija** vraća U, $sigma$ i Vt. Kod poziva funkcije odgovarajućim varijablama pridružimo odgovarajuće vrijednosti i ispišemo te vrijednosti.

Slika 5: Rezultat SVD dekompozicije u Pythonu

10. Zaključak

Danas se najčešće sustav linearnih jednadžbi zapisuje u matrični oblik kako bi se brže došlo do rješenja. Zapisom sustava u matrični oblik dobiva se jasniji i elegantniji oblik sustava jednadžbi pa je jednostavnije pratiti sami proces dobivanja rješenja pogotovo kod velikih sustava. Matricama se pojednostavljuje zapis podataka. Klasični način rješavanja sustava jednadžbi korištenjem Gaussovih eliminacija je problematičan kod velike količine podataka i zato je potrebno transformirati matrice u oblike koji su pogodni za dekompozicije. Gore navedene dekompozicije imaju veliku primijenu u stvarnosti. LU dekompozicija se koristi kod izračuna tokova snage kod velikih dimenzija, Cholesky dekompozicija se koristi se za efikasnije provođenje simulacija kao što su varijante Monte Carlo simulacija, QR dekompozicije se koriste kod kompresija slika dok se SVD dekompozicija koristi kod kvantnih podataka. Svaka od navedenih metoda ima neke nedostatke i prednosti, a odabir vrste dekompozicije ovisi o našim potrebama.

Python nudi brojne mogućnosti zbog svojih raznolikih biblioteka. Biblioteka Numpy ima ugrađene funkcije za matrice. Postoje i ugrađene funkcije za dekompozicije. Za LU dekompoziciju se koristi podbiblioteka scipy.linalg.lu, za cholesky dekompoziciju koristimo linalg.cholesky, za QR dekompoziciju koristimo linalg.qr, a za SVD dekompoziciju koristimo linalg.svd. Mi nismo u ovom radu koristili gotove funkcije već smo napravili svoju implementaciju. Pisanje programskog koda je prilično jednostavno i ima velikih broj funkcija kojima se maksimalno pojednostavlja pisanje programa. Glavni nedostatak je sporo izvođenje programa.

Područje dekompozicija matrica vrlo kompleksno područje koje se i dalje razvija u potrazi za algoritmima kojima bi se još brže dobilo rješenje sustava jednadžbi. Traže se algoritmi kojima bi se još više pojednostavilo rješavanje sustava jednadžbi. Gore navedene dekompozicije su dobar primjer takvih algoritama pogotovo kod velikih količina podataka i predstavljaju temelj za razvoj drugih algoritama.

11. Navođenje literature

[1] Geologija, Znanost o okolišu (bez dat.), Matematika 1 [Na internetu]. Dostupno:https://web.math.pmf.unizg.hr/~fran/predavanja.pdf [pristupano 20.06.2023.]

[2] Leksikogradski zavod M. Krleža (bez dat.), Matrica [Na internetu]. Dostupno: https://www.enciklopedija.hr/natuknica.aspx?ID=39492 [pristupano 22.06.2023.]

[3] Z. Drmac, V. Hari, M. Marušic, M. Rogina, S. Singer i S. Singer, „Numericka analiza," PMF-Matematicki odjel, Sveucilište u Zagrebu, 2003., [Na internetu]. Dostupno: https://web.math.pmf.unizg.hr/nastava/unm/materijali/Hari%20--%20Numericka%20analiza%20-%20osnovni%20udzbenik.pdf [pristupano 22.06.2023.]

[4] D. Bakic, „Linearna algebra," Školska knjiga, Zagreb, 2008., [Na internetu]. Dostupno:https://web.math.pmf.unizg.hr/~bakic/la/linearna_algebra_sk_7.pdf [pristupano 14.06.2023.]

[5] UNIOS (bez dat.), Determinante [Na internetu]. Dostupno:https://www.mathos.unios.hr/matefos/Files/predavanja/p12n.pdf [pristupano 25.06.2023.].

[6] D.Catalin „Determinanta",(bez dat.),[Na internetu]. Dostupno:https://www.math10.com/sr/visha-matematika/matrice/determinanta-matrice.html [pristupano 25.06.2023.].

[7] Geeks for Geeks, (bez dat.), Python Features, [Na internetu]. Dostupno: https://www.geeksforgeeks.org/python-features/ [pristupano 27.06.2023.].

[8] PC Chip, (28.11.2017), Programiranje 101: Od kuda početi s Pythonom, [Na internetu]. Dostupno: https://pcchip.hr/softver/korisni/programiranje-101-od-kuda-poceti-s-pythonom/ [pristupano 29.06.2023.].

[9] R. Scitovski, „Numerička matematika", 2004., [Na internetu]. Dostupno:https://www.mathos.unios.hr/nm/materijali/Num.PDF [pristupano 07.07.2023.].

[10] V. Per, „Metode za određivanje QR dekompozicije", 2018., [Na internetu]. Dostupno:https://dokumen.tips/documents/metode-za-odredivanje-qr-dekompozicije-mdjumicuploadsdiplomskiper23pdf-sa.html, [pristupano 12.07.2023.].

[11] J. Brownlee, „Matrix Decomposition", 2018., [Na internetu]. Dostupno:https://machinelearningmastery.com/introduction-to-matrix-decompositions-for-machine-learning/ [pristupano 23.08.2023.].

[12] I. P. Stanimirović, „Algoritmi za simbolička matrična izračunavanja i optimizaciju", 2012,[Na internetu]. Dostupno:https://dokumen.tips/documents/algoritmi-za-simbolickaa-matricna-izraa-cunavanjaa-i-jedan-od-naasih.html?page=2, [pristupano 03.07.2023.].

[13] Cuemath, (bez dat.), Orthogonal Matrix, [Na internetu]. Dostupno: https://www.cuemath.com/algebra/orthogonal-matrix/ [pristupano 10.07.2023.].

[14] M. Ugrica, „QR dekompozicija koristeći Givensove rotacije”, 2013., [Na internetu]. Dostupno:https://www.mathos.unios.hr/~mdjumic/uploads/diplomski/UGR02.pdf [pristupano 13.07.2023.].

[15] "The Singular Value Decomposition (SVD)" (bez dat.). [Na internetu]. Dostupno:https://math.mit.edu/classes/18.095/2016IAP/lec2/SVD_Notes.pdf [pristupano 14.07.2023.].

[16] "SVD dekompozicija" (bez dat.). [Na internetu]. Dostupno: https://web.math.pmf.unizg.hr/nastava/unm/vjezbe/NM_SVD_vejzbe.pdf [pristupano 15.07.2023.].

[17] N. Truhar, „Numerička linearna algebra”, 2010., [Na internetu]. Dostupno:https://www.mathos.unios.hr/nla/NLA.pdf [pristupano 16.07.2023.].

[18] B. Divjak, „Matematika (PITUP) Dio IV "Matrice i determinante",(bez dat.), [Na internetu]. Dostupno:https://elfarchive1920.foi.hr/pluginfile.php/6620/mod_page/content/35/Matematika1-IVdio.pdf [pristupano 24.08.2023.].

Popis slika

1. Python logo 9
2. Matrice L i U dobivene na temelju Python programa opisanog iznad slike 20
3. Matrice L i L^T dobivene na temelju Python programa opisanog iznad slike (Cholesky dekompozicija) 27
4. Matrice Q i R dobivene na temelju Python programa opisanog iznad slike (QR dekompozicija) 37
5. Rezultat SVD dekompozicije u Pythonu 43