# Aplikacija za rješavanje sustava linearnih jednadžbi

**Spremo, Ena**

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIHTEHNOLOGIJA OSIJEK

**Sveučilišni studij**

# Aplikacija za rješavanje sustava linearnih jednadžbi

**Završni rad**

**Ena Spremo**

**Osijek, 2022.**

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

**Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

**Osijek, 21.09.2022.**

**Odboru za završne i diplomske ispite**

# Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

| | |
|---|---|
| **Ime i prezime Pristupnika:** | Ena Spremo |
| **Studij, smjer:** | Preddiplomski sveučilišni studij Računarstvo |
| **Mat. br. Pristupnika, godina upisa:** | R 4423, 22.07.2019. |
| **OIB Pristupnika:** | 35026756446 |
| **Mentor:** | Izv. prof. dr. sc. Alfonzo Baumgartner |
| **Sumentor:** | Anja Šteko, mag. educ. math. et inf. |
| **Sumentor iz tvrtke:** | |
| **Naslov završnog rada:** | Aplikacija za rješavanje sustava linearnih jednadžbi |
| **Znanstvena grana rada:** | **Programsko inženjerstvo (zn. polje računarstvo)** |
| **Zadatak završnog rad:** | [Rezervirano: Ena Spremo] Potrebno je napraviti aplikaciju koja omogućava unos broja jednadžbi i nepoznanica te na osnovu toga dodatno unosi sve potrebne parametre tih jednadžbi. Aplikacija treba Gauss - Jordanovom metodom eliminacije riješiti sustav jednadžbi te nakon toga ispisati i prikazati postupak dolaska do rješenja. Program se treba napisati u programskom jeziku C#. Sumentorica: Anja Šteko |
| **Prijedlog ocjene završnog rada:** | Izvrstan (5) |
| **Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:** | Primjena znanja stečenih na fakultetu: 3 bod/boda<br>Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda<br>Jasnoća pismenog izražavanja: 3 bod/boda<br>Razina samostalnosti: 3 razina |
| **Datum prijedloga ocjene od strane mentora:** | 21.09.2022. |
| **Datum potvrde ocjene od strane Odbora:** | 22.09.2022. |

| | |
|---|---|
| Potvrda mentora o predaji konačne verzije rada: | *Mentor elektronički potpisao predaju konačne verzije.*<br>Datum: |

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

# IZJAVA O ORIGINALNOSTI RADA

**Osijek, 30.09.2022.**

| | |
|---|---|
| **Ime i prezime studenta:** | Ena Spremo |
| **Studij**: | Preddiplomski sveučilišni studij Računarstvo |
| **Mat. br. studenta, godina upisa:** | R 4423, 22.07.2019. |
| **Turnitin podudaranje [%]:** | 5 |

Ovom izjavom izjavljujem da je rad pod nazivom**: Aplikacija za rješavanje sustava linearnih jednadžbi**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Alfonzo Baumgartner

i sumentora Anja Šteko, mag. educ. math. et inf.

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

# 1. UVOD

U ovom završnom radu bilo je potrebno napraviti aplikaciju za rješavanje sustava linearnih jednadžbi Gauss - Jordanovom metodom eliminacije.

Ova aplikacija će pomoći i poslužiti učenicima, studentima i bilo kome tko se bavi učenjem linearnih jednadžbi, u provjeri valjanosti njihovih zadataka i lakšem razumijevanju ovog dijela matematičkog gradiva.

Kako bi aplikacija bila ispravna i kvalitetno napravljena bilo je nužno imati dobro stečeno znanje iz matematike. Tijekom pisanja rada korištena su znanja iz fakultetskih kolegija „Algoritmi i strukture podataka“, „Objektno orijentirano programiranje“ i „Linearna algebra”.

Pisanje koda aplikacije rađeno je u C#-u, jeziku koji se bazira na objektno orijentiranom programiranju.

U početnim poglavljima završnog rada obrazložena je teorija C# jezika te je dan osvrt na tehnologiju koja je bila korištena za izradu potrebne aplikacije.

Potom je dan opis nastanka koda i algoritma aplikacije, testiranje istoga, te na kraju zaključak konačnih rezultata.

## 1.1. Zadatak završnog rada

Temeljni zadatak bio je izraditi aplikaciju za rješavanje sustava linearnih jednadžbi. Aplikacija će omogućiti unos broja jednadžbi i nepoznanica te na osnovu unesenih podataka, aplikacija dodatno unosi sve potrebne parametre tih jednadžbi. Nakon što se unesu parametri jednadžbi od strane korisnika, aplikacija treba prvo ispisati navedene podatke u matričnom obliku, zatim prikazati izgled matrice nakon Gauss-Jordanove eliminacije, te nakon toga ispisati jedan od tri slučaja. Ako sustav ima rješenje, program ispisuje koja su to rješenja. Ako sustav nema rješenja ili ih ima beskonačno, program ispisuje odgovarajuću poruku. Microsoft Visual Studio i jezik C# korišteni su pri pisanju koda.

# 2. PROGRAMSKI JEZIK C#

## 2.1. Pregled područja rada i postojećih rješenja

Trenutno se za rješavanje sustava linearnih jednadžbi najviše koriste aplikacije Wolfram Alpha, Symbolab, Photomath te preko računala Matrix Calculator, Gauss-Jordan Elimination Calculator i mnoge druge. Wolfram Alpha je iOS aplikacija koja se može skinuti na mobitel, no mora se platiti. Aplikacija je vrlo jednostavna za korištenje. U sebi ima opciju step-by-step rješavanja gdje se svi koraci do rješenja prikazuju detaljno. Također se može birati metoda rješavanja sustava linearnih jednadžbi. Preko računala, aplikacija prikazuje konačno rješenje sustava, a ukoliko korisnik želi da se prikaže detaljan postupak rješavanja tada to mora uzeti pretplatu (mjesečnu ili godišnju). Symbolab za razliku od Wofram Alphe je potpuno besplatan. Jednostavan je za korištenje, no ne daje izbor metode rješavanja, dakle, korisnik mora sam znati upisati metodu koju želi na engleskom jeziku. Također je besplatan na računalu, ali i kao mobilna aplikacija (iOS, android). Photomath je besplatna aplikacija koja nudi opciju skeniranja napisanog zadatka na papiru sa kamerom. Nudi detaljna rješenja zadataka korak po korak te su koraci potkrijepljeni i tekstualnim objašnjenjem. Što se tiče stranica za rješavanje sustava preko računala, Matrix Calculator [1] koristi matrični zapis, dok kod gore navedenih programa jednadžbu korisnik može upisati kako je zadana, npr 2x+3y-1=0. Gauss-Jordan Elimination Calculator [2] omogućuje matrični unos elemenata i prikazuje sve korake rješenja. Aplikacija napravljena u završnom radu omogućuje korisniku unos parametara u matričnom obliku, ispisuje zadanu matricu na konzolu te zatim ispisuje samo krajnje rješenu matricu nakon Gauss-Jordanove eliminacije. Ispisuje rješenja ako postoje, u suprotnom ispisuje odgovarajuću poruku.

## 2.2. Povijest C# u odnosu na C++

C++ je razvio Bjarne Stroustrup[1] u AT & T Bell Laboratoriju. Bjarne je bio snažna podrška C programskog jezika i veliki obožavatelj Simule67 - prvog objektno

[1] Bjarne Stroustrup (rođen 30. prosinca 1950.) je danski informatičar najpoznatiji po izumu i razvoju programskog jezika C++. Gostujući je profesor na Sveučilištu Columbia i radi u Morgan Stanleyu kao izvršni direktor u New Yorku.

orijentiranog jezika. Njegov cilj je bio spojiti najbolje od oba jezika, te tako stvoriti jezik koji zadržava moć C – a, no u isto vrijeme i podržava objektno orijentirano programiranje. To je rezultiralo stvaranjem C++ programskog jezika. On sadrži visoke i niske značajke jezične razine. On se, dakle, smatra jezikom srednje razine. Ranije je imao naziv "*C s klasama*" jer je imao sva svojstva jezika C programskog jezika.

Anders Hejlsberg[2] dao je ključni doprinos razvoju C# jezika. Godine 1999. on i njegov tim programera izgradili su i razvili novi jezik koji je tada imao naziv “Cool.”. U srpnju 2000. godine projekt je odobren te objavljen na .Net Developers Conference-u. Kasnije je ovaj jezik preimenovan u C#.

Glavne razlike C# i C++ programskih jezika su:

- C# je jezik visoke razine, dok se C++ smatra programskim jezikom niske razine koji svom osnovnom jeziku C-u dodaje objektno orijentirane značajke.
- C# kompilira se do CLR(Common Language Runtime), dok se C++ kompilira do strojnog koda.
- Razvoj C#-a treba biti jednostavan, moderan, opće namjene, objektno orijentirani programski jezik, dok u C++ razvoj treba slijediti specifičnu arhitekturu i mora biti prenosiv.
- U C++ se mora ručno upravljati memorijom, dok se C# izvodi u virtualnom stroju, koji upravlja memorijom automatski. [3]

## 2.3. Teorija funkcionalnosti C#

C-Sharp, odnosno C# je programski jezik koji je razvijen od Microsoft-a. On je objektno orijentiran. On radi na .Net Frameworku. Ima značajke kao što su

[2] Anders Hejlsberg (rođen 2. prosinca 1960.) je danski softverski inženjer koji je su-dizajnirao nekoliko programskih jezika i razvojnih alata. Bio je izvorni autor Turbo Pascala i glavni arhitekt Delphija. Trenutno radi za Microsoft kao vodeći arhitekt C#-a i glavni je programer na TypeScript-u.

imperativno, deklarativno, jako tipkanje, te objektno orijentirano programiranje, koje je bazirano na klasama i objektima, te se orijentira na komponente.

Naziv "C-Sharp" inspiriran je glazbenim notama. Simbol '#' označava da napisana nota treba činiti poluton višeg tona. [3]

Namjera C# programskog jezika je da bude jednostavan za korištenje, moderan, te objektno-orijentiran programski jezik prikladan za opću namjenu. Razvojni tim C#-a predvodi Anders Hejlsberg.

Jezik, kao i njegove implementacije, imaju za cilj pružiti podršku za principe softverskog inžinjerstva kao što su: provjera granica niza, stroga provjera tipkanja, pokušaji provjere korištenja nepozvanih varijabli, te automatsko upravljanje memorijom (prikupljanje smeća).

C# je prikladan za pisanje aplikacija za sustave s domaćinom (engl. *host*) i za ugrađene sustave koji u rasponu mogu varirati od vrlo velikih koji koriste sofisticirane operativne sustave, do onih vrlo malih koji sadrže namjenske značajke.

Iako možemo reći da su C# aplikacije poprilično ekonomične, u smislu zahtjeva za memorijom te procesorskom snagom, ipak se ovaj jezik ne natječe izravno u veličini i izvedbi s programskim jezikom C ili asemblerom. [4]

## 2.4. Primjene aplikacija linearnih jednadžbi

Koje su uobičajene primjene linearnih jednadžbi u stvarnom životu?

Upotreba linearnih jednadžbi u stvarnom životu, dolazi u obzir u situacijama u kojima postoji neka nepoznata količina podataka. Većinu vremena mentalni izračuni koriste se u nekim stvarnim situacijama bez potrebe za crtanjem linearnog grafa.

Primjene linearnih jednadžbi koriste se svakodnevno čak i bez korištenja linearnih grafova jer se ljudi suočavaju sa situacijama u kojima može doći do nekakve nepoznate količine podataka, a koja se upravo može predstaviti kao linearna jednadžba. To može biti na primjer: izračun određenog stopa kilometraže, izračun prihoda tijekom zadanog vremena, izračun potrošnje novca u toku jedne godine itd. Vrlo su važne četiri glavne aritmetičke operacije - zbrajanje, oduzimanje, množenje i dijeljenje.

Glavni cilj primjene linearnih jednadžbi ili linearnih sustava je rješavanje različitih problema korištenjem dvije (ili više) varijable gdje je jedna od njih poznata, a druga nepoznata, no ovisna o prvoj. Neke od primjena korištenja linearnih jednadžbi s dvijema nepoznanicama su:

- geometrijski problemi
- problemi s novcem
- mješavina različitih problema pomoću dvije varijable
- Distance-Rate-Time problemi
- primjena linearnih jednadžbi u poslovanju i ekonomiji. [5]

Kod matrica i vektora možemo razmatrati mnoge koncepte linearnih jednadžbi. Kod matrice možemo promatrati eliminaciju matrice, determinantu matrice, inverze matrice, množenje matrice, matrice projekcije potprostora, matrice refleksije, matrice rotacije i matrice smicanja. Kod vektora možemo promatrati kuteve, normalizaciju, skalarni produkt i vektorski produkt te projekciju.

# 3. TEORIJSKA PODLOGA

## 3.1. Teorija sustava

Teoriju sustava definira se kao interdisciplinarnu znanost. Navedena teorija bavi se zakonitostima razvoja sustava. Također, proučava način postanka sustava, te proučava organizaciju i globalno ponašanje istog. Ova teorija ima za cilj stvaranje metoda i pronalazak načina rješavanja problema. Navedeno se radi razvijanjem novog sustava koji se promatra kao apstraktan. Njega opisujemo kao rezultat stvarnoga sustava. Ludwig von Bertalanffy promicao je jedan od prvih oblika opće teorije, no teorija sustava zaživila je tek 1950-ih, na temeljnim načelima fizike, inženjerstva i filozofije. Kasnije je primijenjena i na mnoga druga znanstvena područja. U današnje vrijeme imamo šest znanstvenih područja usko vezanih uz teoriju sustava. To su dijelovi informatike i semiotike, općenita teorija sustava kibernetika, matematička teorija i teorija informacija.

Što je to sustav? Sustav opisujemo kao skup objekata, odnosno elemenata, gdje ti elementi mogu biti elementi sustava ili podsustava. Objekti sustava međusobno su spojeni vezama na način da skupno izgledaju kao jedna velika cjelina. Za potrebe rješavanja zadataka, podsustave možemo rastaviti na njegove komponente, ali sa elementima to ne možemo činiti. Za svaki sustav možemo reći da je podsustav nekoga nadsustava. Taj nadsustav je njegova okolina s kojom razmjenjuje nekakve određene informacije. Nasuprot klasičnoj analizi teorije sustava, kod sustavne analize mi proučavamo okruženje kojega je zadani sustav dio, a različite promjene u toj okolini, objašnjavamo kao prilagođavanje sustava njegovoj okolini. Dio sustava koji se smatra njegovim obilježjem je proces kojim ulazne veličine postaju u izlazne veličine. Jedan dio ulaznih veličina u sustavu troši se na funkcionalnost njega samog, time dobivamo izlazne veličine koje se ne mogu iskoristiti. No, drugi dio se pretvori u korisne te tako možemo ostvariti rješenje sustava. Kako bi postavljeni ciljevi sustava mogli biti realizirani, sustavom se mora moći omogućiti upravljanje. Kada sustavom ne možemo upravljati tada on odlazi ka stanju maksimalne entropije. Navedeno stanje se objašnjava kao nered unutar sustava.

Kako bi se sustavom moglo dobro upravljati, cilj nam je održati ga u željenom stanju, a to možemo postići smanjivanjem entropije. Entropiju možemo smanjiti dodavanjem novih informacija. [6]

## 3.2. Linearni sustavi

Kada govorimo o linearnoj algebra jedan od ključnih i poznatijih pojmova je sustav linearnih jednadžbi. Njega možemo opisati sljedećim primjerom:

$$-129x_1 - 15x_2 + 3x_3 = 141$$

$$0.5x_1 - 15x_2 - 99x_3 = -64$$

$$2.55x_1 - 63x_2 + 2.334x_3 = 76.$$

Glavni problem rješavanja ovog sustava je utvrđivanje da li postoji takav skup rješenja za $x_1$, $x_2$ i $x_3$ iz gore navedenog primjera, da zadovolji sve tri jednadžbe u isto vrijeme. Ako takav skup postoji, navedenog treba i pronaći. Hoće li skup rješenja postojati, ovisi o svakoj pojedinoj jednadžbi te o parametrima ispred nepoznanica, koji mogu biti prirodni, cijeli i realni brojevi.

Navedene sustave svrstavamo među jedne od najstarijih matematičkih problema. Za njih imamo mnogobrojne primjere gdje se primjenjuju. Neki od njih su: procjene, predviđanja, linearno programiranje, obrađivanje digitalnih signala itd. Postoje mnogi načini za rješavanje sustava linearnih jednadžbi. Međutim, u ovom završnom radu mi ćemo se baviti onim najučinkovitijim, a to je Gaussov postupak eliminacije.

Poopćeno, sustav u kojem imamo $m$ linearnih jednadžbi te uz njega $n$ nepoznanica zapisuje se ovako:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & = & b_m. \end{array}$$

U navedenom sustavu $x_1, x_2, \dots, x_n$ predstavljaju nepoznanice sustava, a brojevi $a_{11}, a_{12}, \dots, a_{mn}$ predstavljaju koeficijente sustava. Koeficijente sustava u matričnom obliku zapisujemo kao:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$ [7] (3-1)

Skalari $a_{ij}$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, zovu se koeficijenti sustava, dok su $b_1, \dots, b_m$ slobodni članovi.

Radi jednostavnosti uz sustav naveden pod (3-1) uobičajeno se vežu matrice:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}, X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, B = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, A_p = \begin{bmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_m \end{bmatrix}.$$

Dakle, $A$ je matrica sustava, $X$ je matrica nepoznanica, $B$ je matrica slobodnih članova, te poslijednja, $A_p$ je proširena matrica sustava.

Pomoću gore navedenih matrica sustav (3-1) možemo u ekvivalentnom obliku zapisati:

$$\boldsymbol{AX = B}.$$

U gore navedenoj jednadžbi $\boldsymbol{A}$ predstavlja matricu dimenzija $m \times n$, $\boldsymbol{X}$ nam predstavlja vektor stupac koji sadrži $\boldsymbol{n}$ članova, a $\boldsymbol{B}$ predstavlja vektor redak koji sadrži $\boldsymbol{m}$ članova u sebi. Gauss-Jordanovu eliminaciju može se primjeniti na sve navedene sustave. Koeficijenti jednadžbi mogu biti proizvoljni. [8]

Rješenje ovakvog sustava čine svi parovi $(x_1, x_2, \dots, x_n)$ koji zadovoljavaju sve navedene linearne jednadžbe istovremeno. Drugim riječima, govorimo o točkama koje leže na ravninama koje su određene točno ovim zadanim jednadžbama.

U slučaju da je polje beskonačno, a to se može dogoditi ako imamo realne ili kompleksne brojeve, onda su moguća isključivo i samo sljedeća tri slučaja za svaki gore opisan i navedeni sustav linearnih jednadžbi, a to su:

1. zadani sustav jednadžbi nema rješenja (kažemo da je sustav nemoguć),

2. zadani sustav jednadžbi ima točno i samo jedno rješenje (kažemo da je sustav istinit),

3. zadani sustav jednadžbi ima beskonačno mnogo rješenja (kažemo da je sustav neodređen). [7]

### 3.2.1 Homogeni sustav

Posebna vrsta sustava linearnih jednadžbi je homogeni sustav. On je homogen ako su mu svi slobodni koeficijenti jednaki nuli.

Odnosno, ako vrijedi:

$$b_1 = \ldots = b_m = 0.$$

onda je sustav (3-1) homogen.

Njegov opći oblik zapisujemo kao:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & 0 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & 0, \end{array} \qquad (3\text{-}2)$$

odnosno

$$AX = 0.$$

Za homogeni sustav vrijedi da je uvijek rješiv (jedno rješenje je uvijek trivijalno rješenje $x_1 = x_2 = \cdots = x_n = 0$). Vektorskim prostorom nazivamo skup svih rješenja nekog homogenog sustava (3-2).[8]

## 3.3. Cramerovo pravilo

Sustav lineranih jednadžbi možemo rješavati Cramerovim pravilom. Prilikom takvog rješavanja sustava potrebno je poznavati računanje determinanti. Kako bismo mogli računati determinante, matrica sustava $A$ mora biti dimenzije $n \times n$, tj. broj jednadžbi mora biti jednak broju nepoznanica. Prikažimo sustav linearnih jednadžbi u matričnom obliku:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

Rješenje sustava su uređene $n$-torke $(x_1, x_2, ..., x_n)$ gdje $i$-tu komponentu računamo kao:

$$x_i = \frac{D_i}{D}. \ (3\text{-}3)$$

U (3-3) varijabla $D$ predstavlja determinantu matrice $A$. Varijablom $D_i$ predstavljena je izmjenjena determinanta. Cijeli $i$-ti stupac ove determinante $D_i$ zamjenili smo sa stupcem $b$.

U slučaju kada je $D = \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix} = 0$ , zadani sustav može imati beskonačno mnogo rješenja ili sustav ne mora imati niti jedno rješenje. Postojanje rješenja ovisi o determinantama $D_i$:

- Kada je $D_i = 0, \forall i$, govorimo o beskonačno rješenja $(x_1, x_2, ..., x_n)$.
- Ako postoji $D_i \neq 0$, sustav jednadžbi nema rješenje.

U slučaju kada je $D = \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix} \neq 0$, sustav ima jedno jedinstveno rješenje $(x_1, x_2, ..., x_n)$. [9]

U sljedećem primjeru prikazat ćemo rješavanje sustava linearnih jednadžbi:

$$2x_1 + x_2 + 2x_3 = 2$$

$$x_1 + 3x_2 + x_3 = 3$$

$$x_1 + x_2 + 4x_3 = -1.$$

Kako bi rješili sustav, najprije ga zapisujemo u matričnom zapisu:

$$\begin{bmatrix} 2 & 1 & 2 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix},$$

Sljedeći korak je računanje determinante:

$$D = \begin{vmatrix} 2 & 1 & 2 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{vmatrix} = 2 * \begin{vmatrix} 3 & 1 \\ 1 & 4 \end{vmatrix} - 1 * \begin{vmatrix} 1 & 2 \\ 1 & 4 \end{vmatrix} + 1 * \begin{vmatrix} 1 & 2 \\ 3 & 1 \end{vmatrix} =$$

$$= 2 * (12 - 1) - (4 - 2) + (1 - 6) =$$

$$= 22 - 2 - 5 = 15 \neq 0.$$

Budući da je $D \neq 0$, zaključujemo da sustav ima jedinstveno rješenje. Kako bismo mogli odrediti $x_i$, $\forall i$ odredit ćemo determinante $D_i$.

Determinantu $D_1$ dobit ćemo tako da prvi stupac determinante $D$ zamijenimo sa stupcem slobodnih koeficijenata:

$$D_1 = \begin{vmatrix} 2 & 1 & 2 \\ 3 & 3 & 1 \\ -1 & 1 & 4 \end{vmatrix} = 2 * \begin{vmatrix} 3 & 1 \\ 1 & 4 \end{vmatrix} - 3 * \begin{vmatrix} 1 & 2 \\ 1 & 4 \end{vmatrix} - 1 * \begin{vmatrix} 1 & 2 \\ 3 & 1 \end{vmatrix} =$$

$$= 2 * (12 - 1) - 3 * (4 - 2) - (1 - 6) =$$

$$= 22 - 6 + 5 = 21.$$

Analogno ćemo odrediti i determinante $D_2$ i $D_3$:

$$D_2 = \begin{vmatrix} 2 & 2 & 2 \\ 1 & 3 & 1 \\ 1 & -1 & 4 \end{vmatrix} = 2 * \begin{vmatrix} 3 & 1 \\ -1 & 4 \end{vmatrix} - 1 * \begin{vmatrix} 2 & 2 \\ -1 & 4 \end{vmatrix} + 1 * \begin{vmatrix} 2 & 2 \\ 3 & 1 \end{vmatrix} =$$

$$= 2 * (12 + 1) - (8 + 2) + (2 - 6) =$$

$$= 26 - 10 - 4 = 12,$$

$$D_3 = \begin{vmatrix} 2 & 1 & 2 \\ 1 & 3 & 3 \\ 1 & 1 & -1 \end{vmatrix} = 2 * \begin{vmatrix} 3 & 3 \\ 1 & -1 \end{vmatrix} - 1 * \begin{vmatrix} 1 & 2 \\ 1 & -1 \end{vmatrix} + 1 * \begin{vmatrix} 1 & 2 \\ 3 & 3 \end{vmatrix} =$$

$$= 2 * (-3 - 3) - (-1 - 2) + (3 - 6) =$$

$$= -12 + 3 - 3 = -12.$$

Prema formuli $x_i = \frac{D_i}{D}$ može se izračunati $x_1$, $x_2$ i $x_3$:

$$x_1 = \frac{21}{15} = \frac{7}{5} = 1.4,$$

$$x_2 = \frac{12}{15} = \frac{4}{5} = 0.8,$$

$$x_3 = \frac{-12}{15} = \frac{-4}{5} = -0.8.$$

Rješenje zadanog sustava linearnih jednadžbi je: $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.4 \\ 0.8 \\ -0.8 \end{bmatrix}$.

## 3.4. Gaussova metoda eliminacije

Sustav linearnih jednadžbi možemo rješavati pomoću Gaussove metode eliminacije.

Pri korištenju ove metode, sustav moramo najprije preoblikovati u matrični zapis, nakon toga sustav (3-1) elementarnim transformacijama svodi se na ekvivalentni sustav, iz kojega ćemo onda sa lakoćom moći odrediti njegovo rješenje.

Uspoređivajem dvaju sustava kažemo da su oni ekvivalentni ako sadrže isti skup rješenja. Pri procesu svođenja sustava na ekvivalentan koristimo tri osnovne operacije nad retcima (te operacije nazivaju se elementarne transformacije):

- operacija zamjena dvaju redaka,
- operacija množenje jednog retka sa skalarom $\alpha \neq 0$,
- operacija dodavanja nekog retka nekom drugom retku. [9]

Glavni bi cilj opisanog algoritma bio matricu sustava svesti na gornje trokutastu matricu, korištenjem elementarnih transformacija nad retcima:

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ 0 & a_4 & a_5 \\ 0 & 0 & a_6 \end{bmatrix}.$$

U navedenoj matrici vrijedi da su $a_{1,} a_4, a_6 \neq 0$. Ona se dobije sa osnovnim operacijama nad retcima.

Navodimo primjer 3D sustava koji ćemo rješiti Gaussovom metodom:

$$2x_1 + x_2 - 2x_3 = 4$$

$$-x_1 + 2x_2 + 3x_3 = 8$$

$$4x_1 + 8x_2 - 2x_3 = 12.$$

Kako bi ovaj način rješavanja bio jednostavniji i lakše shvatljiv, prvu linearnu jednadžbu označit ćemo s R1, drugu s R2, a treću s R3.

Najprije ćemo napisati matrični zapis sustava, te onda gledamo kako najbrže i najlakše možemo dobiti gornju trokutastu matricu.

$$\left[\begin{array}{ccc|c} 2 & 1 & -2 & 4 \\ -1 & 2 & 3 & 8 \\ 4 & 8 & -2 & 12 \end{array}\right] \rightarrow \frac{1}{2} * \text{R1} + \text{R2} \rightarrow \text{R2, također i (-2)} * \text{R1} + \text{R3} \rightarrow \text{R3.}$$

Redak R1 pomožili smo skalarom $\frac{1}{2}$ te ga zbrojili sa retkom R2 kako bi dobili nulu na mjestu gdje se nalazi -1. Redak R1 zatim smo pomnožili sa -2 i dodali ga retku R3 kako bi dobili nulu na mjestu gdje se nalazi vrijednost 4. Preostaje nam još samo dobiti nulu na mjestu gdje je vrijednost broj 8.

$$\left[\begin{array}{ccc|c} 2 & 1 & -2 & 4 \\ 0 & \frac{5}{2} & 2 & 10 \\ 0 & 6 & 2 & 4 \end{array}\right] \rightarrow -\frac{12}{5} \text{ R2} + \text{R3} \rightarrow \text{R3}$$

Množimo redak R2 sa skalarom $\frac{-12}{5}$ te ga dodajemo retku R3, kako bi dobili nulu na traženom mjestu.

$$\left[\begin{array}{ccc|c} 2 & 1 & -2 & 4 \\ 0 & \frac{5}{2} & 2 & 10 \\ 0 & 0 & -\frac{14}{5} & -20 \end{array}\right]$$

U prethodnom prikazu matrice, vidimo kako je ona svedena na gornju trokutastu. Nakon toga se iz nje zapisuje dobiveni sustav:

$$2x_1 + x_2 - 2x_3 = 4$$

$$\frac{5}{2}x_2 + 2x_3 = 10$$

$$-\frac{14}{5}x_3 = -20 \rightarrow x_3 = \frac{50}{7}.$$

Nadalje, nakon što je $x_3$ izračunat, uvrštava se u srednju jednadžbu kako bismo pronašli $x_2$ te zatim rezultate uvrštavamo u gornju jednadžbu kako bismo pronašli $x_1$:

$$\frac{5}{2}x_2 + 2 * \frac{50}{7} = 10 \rightarrow x_2 = -\frac{12}{7}$$

$$2x_1 + \frac{-12}{7} - 2 * \frac{50}{7} = 4 \rightarrow x_1 = 10.$$

Rješenje zadanog sustava je uređena trojka ($x_1$, $x_2$, $x_3$) = (10, $-\frac{12}{7}$, $\frac{50}{7}$).

### 3.4.1. Gauss-Jordanova metoda eliminacije

Gauss-Jordanovom metodom eliminacije služit ćemo se u kodu aplikacije. Naime, Gauss-Jordanova eliminacija je algoritam koji ćemo koristiti u rješavanju sustava. Također može služiti i za pronalaženje inverza bilo koje invertibilne matrice.

Oslanja se na tri osnovne operacije retka koje se mogu koristiti nad matricom:

- operacija zamjene položaja dva retka,
- operacija množenja jednog od redaka sa skalarom koji nije nula,
- operacija množenja bilo koje jednadžbe sustava brojem, te zatim dodavanje rezultata bilo kojoj drugoj jednadžbi sustava.

Cilj Gauss-Jordanove metode eliminacije jest dobiti jediničnu matricu.

Jedinična matrica je matrica kojoj se na glavnoj dijagonali nalaze samo jedinice, a svi ostali elementi su nule.

Primjer jedinične matrice dimenzije $3 \times 3$:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Primjer rješenja sustava Gauss-Jordanovom metodom:

$$x_1 + 2x_2 + x_3 = 10$$

$$x_1 + 3x_2 - x_3 = -2$$

$$2x_1 + x_2 + 4x_3 = 8.$$

Radi lakšeg shvaćanja prvu linearnu jednadžbu označiti ćemo sa R1, drugu sa R2, a treću sa R3.

Najprije zapišemo linearne jednadžbe kao matricu:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 1 & 3 & -1 & -2 \\ 2 & 1 & 4 & 8 \end{array}\right].$$

Prvi redak množimo sa -1 i dodajemo ga drugom retku:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 1 & 3 & -1 & -2 \\ 2 & 1 & 4 & 8 \end{array}\right] \text{R2} - 1 * \text{R1} \rightarrow \text{R2} \sim \left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & -2 & -12 \\ 2 & 1 & 4 & 8 \end{array}\right].$$

Prvi redak smo pomnožili sa -2 i dodali ga trećem retku:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & -2 & -12 \\ 2 & 1 & 4 & 8 \end{array}\right] \text{R3} - 2 * \text{R1} \rightarrow \text{R3} \sim \left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & -2 & -12 \\ 0 & -3 & 2 & -12 \end{array}\right].$$

Drugi redak množimo sa 3 i dodajemo ga trećem retku:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & -2 & -12 \\ 0 & -3 & 2 & -12 \end{array}\right] \text{R3} - (-3) * \text{R2} \rightarrow \text{R3} \sim \left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & -2 & -12 \\ 0 & 0 & -4 & -48 \end{array}\right].$$

Treći redak množimo sa $\frac{-1}{4}$ kako bismo se riješili broja -4:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & -2 & -12 \\ 0 & 0 & -4 & -48 \end{array}\right] \text{R3/(–4)} \rightarrow \text{R3} \sim \left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & -2 & -12 \\ 0 & 0 & 1 & 12 \end{array}\right].$$

Treći redak množimo s 2 i dodajemo ga drugom retku:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & -2 & -12 \\ 0 & 0 & 1 & 12 \end{array}\right] \text{R2}-(-2)\cdot\text{R3}\rightarrow\text{R2} \sim \left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & 0 & 12 \\ 0 & 0 & 1 & 12 \end{array}\right].$$

Treći redak množimo s -1 i dodajemo ga prvom retku:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 10 \\ 0 & 1 & 0 & 12 \\ 0 & 0 & 1 & 12 \end{array}\right] \text{R1}-1\cdot\text{R3}\rightarrow\text{R1} \sim \left[\begin{array}{ccc|c} 1 & 2 & 0 & -2 \\ 0 & 1 & 0 & 12 \\ 0 & 0 & 1 & 12 \end{array}\right].$$

Drugi redak množimo s -2 i dodajemo prvom.

$$\left[\begin{array}{ccc|c} 1 & 2 & 0 & -2 \\ 0 & 1 & 0 & 12 \\ 0 & 0 & 1 & 12 \end{array}\right] \text{R1}-2\cdot\text{R2}\rightarrow\text{R1} \sim \left[\begin{array}{ccc|c} 1 & 0 & 0 & -26 \\ 0 & 1 & 0 & 12 \\ 0 & 0 & 1 & 12 \end{array}\right].$$

Na taj način smo prvi dio proširene matrice sustava sveli na jediničnu matricu:

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & -26 \\ 0 & 1 & 0 & 12 \\ 0 & 0 & 1 & 12 \end{array}\right].$$

Budući da je s lijeve strane formirana jedinična matrica, možemo iščitati rješenja:

$$x_1 = -26$$

$$x_2 = 12$$

$$x_3 = 12$$

Rješenje zadanog sustava je uređena trojka $(x_1, x_2, x_3) = (-26,12,12)$.

## 3.6. Kronecker-Capellijev teorem

Pomoću ovog teorema pobliže se opisuje struktura rješenja koju dobivamo rješavanjem sustava linearnih jednadžbi, koje prikazujemo jednadžbom $AX = B$. Spomenuta struktura rješenja uvijek ovisi o dva bitna parametra, a to su:

- rang matrice $A$,
- rang njezine proširene matrice $A_p$.

$A$ je općenita matrica nekog sustava sa elementima $a_{11}$ do $a_{mn}$:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix},$$

$X$ je matrica sustava sa jednim stupcem od $x_1$ pa sve do $x_n$:

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},$$

Dok je $A_p$ proširena matrica sustava:

$$A_p = \begin{bmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_n \end{bmatrix}.$$ [10]

Ako se u obzir uzmu rang matrice $A$ te rang njene proširene matrice $A_p$, te također uz njih još i broj redaka ($m$) i stupaca ($n$) matrice $A$, možemo doći do sljedećih bitnih zaključaka vezana uz rješenja jednadžbi:

1. Ako za sustav $AX = B$ vrijedi $r(A) < r(A_p)$, tada ih nema.
2. Ako za sustav $AX = B$ vrijedi $r(A) = r(A_p) = n$ , tada je jedinstveno.
3. Ako za sustav $AX = B$ vrijedi $r(A) = r(A_p) < n$ , ima ih beskonačno.

U sljedećim primjerima prikazat ćemo gore navedene zaključke. [9]

Primjer 1.

$$2x_1 + 2x_2 = 7$$

$$2x_1 + 2x_2 = 9$$

Zapišimo ovaj sustav u matričnom obliku:

$$\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 9 \end{bmatrix}.$$

Pomoću osnovnih matematičkih operacija nad stupcima, moramo najprije dobiti rang matrice. Radi preglednosti, u zapisu jednadžbi R1 će predstavljati prvu, a R2 će predstavljati drugu linearnu jednadžbu. Najprije se računa rang matrice $A$:

$$\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \rightarrow (-2) * \text{R1} + \text{R2} \rightarrow \text{R2} .$$

Nakon gornje operacije matrica je: $\begin{bmatrix} 2 & 2 \\ 0 & 0 \end{bmatrix}$, stoga možemo zaključiti da je rang matrice jednak 1, jer mamo jedan ne nul-redak od ukupno dva retka. Odnosno $r(A) = 1$. Nadalje računamo rang proširene matrice $A_p$:

$$\begin{bmatrix} 2 & 2 & 7 \\ 2 & 2 & 9 \end{bmatrix} \rightarrow (-2) * \text{R1} + \text{R2} \rightarrow \text{R2}.$$

Nakon elementarne transformacije, matrica ima sljedeći izgled:

$$\begin{bmatrix} 2 & 2 & 7 \\ 0 & 0 & -5 \end{bmatrix}.$$

Zaključujemo da je rang proširene matrice jednak 2, jer imamo dva ne-nul retka. Matematički zapisano kao $r(A_p) = 2$. Promotrimo li 1. zaključak ($r(A) < r(A_p)$), jasno je da ovaj sustav nema rješenja.

Primjer 2.

$$4x_1 + 4x_2 = 20$$

$$x_1 + x_2 = 5$$

Prikažimo sustav kao matricu:

$$\begin{bmatrix} 4 & 4 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 20 \\ 5 \end{bmatrix}.$$

Najprije tražimo $r(A)$ preko Gaussovih transformacija, a zatim rang proširene matrice.

$\begin{bmatrix} 4 & 4 \\ 1 & 1 \end{bmatrix} \rightarrow (-4) * \mathrm{R2} + \mathrm{R1} \rightarrow \mathrm{R1}$

$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$ → vidimo da je $r(A) = 1$.

$\begin{bmatrix} 4 & 4 & 20 \\ 1 & 1 & 5 \end{bmatrix} \rightarrow (-4) * \mathrm{R2} + \mathrm{R1} \rightarrow \mathrm{R1}$

$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 5 \end{bmatrix}$ → vidimo da je $r(A_p) = 1$.

Pošto je $r(A) < n$ $(n = 2)$, nadalje se mora provjeriti $r(A_p)$. Zaključujemo da je on jednak $r(A)$, te onda možemo reći da sustav ima beskonačno rješenja (prema 3. zaključku).

Primjer 3.

$$6x_1 + 6x_2 = 12$$

$$6x_1 - 12x_2 = 30$$

Zapišemo sustav u matričnom obliku, pa tražimo rang matrice $A$, pa rang proširene matrice $A_p$.

$\begin{bmatrix} 6 & 6 \\ 6 & -12 \end{bmatrix} \rightarrow 2 * \text{R1} + \text{R2} \rightarrow \text{R2}$

$\begin{bmatrix} 6 & 6 \\ 12 & 0 \end{bmatrix} \rightarrow (\frac{-1}{2}) * \text{R2} + \text{R1} \rightarrow \text{R1}$, iz čega slijedi:

$\begin{bmatrix} 0 & 6 \\ 12 & 0 \end{bmatrix}$ ovdje vidimo da je $r(A) = 2$.

Nadalje se računa rang proširene matrice:

$\begin{bmatrix} 6 & 6 & 12 \\ 6 & -12 & 30 \end{bmatrix} \rightarrow 2 * \text{R1} + \text{R2} \rightarrow \text{R2}$,

$\begin{bmatrix} 6 & 6 & 12 \\ 12 & 0 & 24 \end{bmatrix} \rightarrow (\frac{-1}{2}) * \text{R2} + \text{R1} \rightarrow \text{R1}$,

$\begin{bmatrix} 0 & 6 & 0 \\ 12 & 0 & 24 \end{bmatrix} \rightarrow r(A_p) = 2$

Prema 2. zaključku koji glasi ($r(A) = r(A_p) = n$), vidimo da sustav ima jedinstveno rješenje.

# 4. APLIKACIJA ZA RJEŠAVANJE SUSTAVA LINERANIH JEDNADŽBI

## 4.1. Ideja i model

Na fakultetima koji se bave područjima prirodnih znanosti, studenti često moraju učiti gradivo vezano uz matrice te rješavanje sustava linearnih jednadžbi koristeći Gaussove metode eliminacije. Ova aplikacija služi upravo kako bi pomogla studentima pri učenju ovog dijela gradiva, no i profesorima pri provjeri rezultata određenih zadataka sa sustavima linearnih jednadžbi s $n$ nepoznanica.

Prije početka razvoja aplikacije bilo je potrebno analizirati problem rješavanja sustava linearnih jednadžbi Gaussovom metodom eliminacije, a zatim i Gauss-Jordanovom metodom eliminacije. Navedeno je učinjeno uz pomoć dostupne literature i predznanja iz kolegija „Linearna algebra' te „Algoritmi i strukture podataka".

Osnova svake aplikacije je njezin model kojim određujemo na koji će ona način biti prikazana korisniku. Za potrebe ovog završnog rada, komunikacija s korisnikom vrši se preko konzole. Korisniku treba biti jasno za što aplikacija služi, kako se koristiti njome i na koji način se unose podatci te iščitavaju rezultati koje daje ista. Sve navedeno je omogućeno s porukama koje se ispisuju na konzolu u obliku uputa kako bi se korisniku olakšalo korištenje aplikacije. Program rješava uneseni sustav linearnih jednadžbi koristeći Gauss-Jordanovu metodu eliminacije, te daje rješenja istih, a ukoliko ih nema, ili ih je beskonačno, ispisuje adekvatnu poruku.

## 4.2. Metoda parcijalnog pivotiranja

Za rješavanje sustava lineranih jednadžbi Gauss-Jordanovom metodom eliminacije u kodu je korištena metoda parcijalnog pivotiranja.

Parcijalno pivotiranje se koristi jer je to algoritam koji je lakše shvatljiv računalu prilikom rješavanja zadatka Gauss-Jordanovom metodom.

Parcijalno pivotiranje je samo dio algoritma zapisanog unutar glavne metode po nazivu **GaussJordan()**.

Kod parcijalnog pivotiranja u svakom koraku izvršavanja petlje postoji jedan glavni element koji se naziva pivot. Pivot element je prvi element u retku koji nema vrijednost nula. U slučaju da pivot element ima vrijednost nula radimo zamjenu redaka na način da se element koji je nula makne sa glavne dijagonale matrice.

Operacije koje se izvršavaju sa pivot elementom su:

1. svođenje pivot elementa na jedinicu,
2. svođenje svih elemenata u stupcu ispod i iznad pivot elementa na nule.

Pri izvršavanju ovih operacija ne smijemo zanemariti ostale elemente matrice.

Kako bi se pivot element sveo na jedinicu, moramo cijeli redak u kojem se nalazi pivot element podijeliti sa njegovom vrijednošću.

Zatim, kako bi sveli prvi element ispod pivot elementa u stupcu na nulu moramo pomnožiti pivot element s vrijednosti elementa ispod njega i onda dobivenu vrijednost oduzeti od svakog elementa u retku ispod pivot elementa. Postupak se ponavlja za svaki element u stupcu gdje se nalazi pivot element.

## 4.3. Administracijski dio

```
1 reference
public static double[,] GaussJordan(double[,] a, double[] b, int n)
{
    for (int k = 0; k < n; k++) //osnovna petlja za parcijalno pivotiranje
    {

        if (Math.Abs(a[k, k]) == 0)
        {
            for (int i = k + 1; i < n; i++)
            {
                if (Math.Abs(a[i, k]) > Math.Abs(a[k, k]))
                //setanje po glavnoj dijagonali i provjera
                //je li na njoj element sa vrijednosti nula,

                //ukoliko je radi se zamjena redaka
                {
                    for (int j = k; j < n; j++)
                    {

                        double temp = a[k, j];
                        a[k, j] = a[i, j];
                        a[i, j] = temp;

                        double tempb = b[k];
                        b[k] = b[i];
                        b[i] = tempb;

                        break;
                    }
                }
            }
        }

        //dijeljenje pivot(glavnog) retka
        double pivot = a[k, k];
        if (pivot != 0)
        {
            for (int j = k; j < n; j++)
            {
                a[k, j] /= pivot;
            }

            b[k] /= pivot;

            //petlja za eliminaciju, pravljenje nula ispod i iznad glavne dijagonale i jedinica na glavnoj
            for (int i = 0; i < n; i++)
            {
                if (i == k || a[i, k] == 0) continue;
                double factor = a[i, k];

                for (int j = k; j < n; j++)
                {
                    a[i, j] -= factor * a[k, j];

                }

                b[i] -= factor * b[k];
            }
        }


    }

    return a;
}
```

*Slika 4.3.1. Metoda za Gauss-Jordanovu metodu eliminacije*

```
2 references
public static void DisplayMatrix(double[,] matrix, double[] b)  //metoda za prikazivanje matricnog oblika zapisa
{
    int length;
    length = Convert.ToInt32(Math.Sqrt(matrix.Length));
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length; j++)
        {
            Console.Write("{0}\t", matrix[i, j]);
        }

        Console.Write("|{0}", b[i]);   //ispisuje rijesenja jednadzbi u zadnjem stupcu
        Console.WriteLine();
    }

}
```

*Slika 4.3.2. Metoda za prikaz matričnog oblika zapisa na konzoli*

```
using System;

namespace GaussJordan
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Dobrodošli!\nOvaj program je napravljen za rješavanje sustava linearnih jednadžbi\nGauss - Jordanovom metodom eliminacije.");
            int equations;
            double[,] Matrix;
            double[] b; // 1D matrica b su rijesenja jednadzbi
                        //unos broja jednadzbi dok se ne zadovolje zadani kriteriji
            int brojac;
            int pozicija;
            do
            {
                Console.WriteLine("Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):");
                equations = Int32.Parse(Console.ReadLine());

                if (equations < 0) Console.WriteLine("Ne može postojati negativan broj jednadžbi.");

                Matrix = new double[equations, equations];
                b = new double[equations];

                if (equations == 0) Console.WriteLine("Ne postoje jednadžbe na kojima bi se program izvodio.");
                if (equations == 1) Console.WriteLine("Program zahtijeva najmanje dvije jednadžbe za izvođenje Gauss - Jordanove eliminacije.");



            }
            while (equations <= 1);
```

*Slika 4.3.3. Do-While petlja za pravilan unos podataka*

```
Console.WriteLine("Unesite parametre jednadžbi:");
Console.WriteLine("Matrica b predstavlja rezultate vaših jednadžbi.");

//Ovdje nam "i" predstavlja broj redaka u matrici (rows), a varijabla equations predstavlja broj jednadzbi odnosno broj nepoznanica
//Varijabla "j" predstavlja broj stupaca u matrici (columns)
for (int i = 0; i < equations; i++)
{
    for (int j = 0; j < equations; j++)
    {
        Console.Write("Matrix[{0},{1}]: ", i, j);
        //Ispisuje redom indekse svakog mjesta u matrici koje korisnik onda treba popuniti sa vrijednostima
        double v = double.Parse(Console.ReadLine());
        //Zapisivanje parametara
        Matrix[i, j] = v;

    }
    Console.Write("b[{0}]: ", i);
    b[i] = double.Parse(Console.ReadLine());
    Console.WriteLine();
}

Console.WriteLine("Prikaz zadane matrice:");

DisplayMatrix(Matrix, b);
Console.WriteLine("__________________________________");

GaussJordan(Matrix, b, equations);
```

*Slika 4.3.4. Omogućavanje unosa svih podataka u konzolu korisniku*

```
Console.WriteLine("Matrica nakon Gauss-Jordanove eliminacije:");

DisplayMatrix(Matrix, b);
Console.WriteLine("__________________________________");
brojac = PronadiBrojBrojaca(Matrix, b, equations);
pozicija = PronadiPoziciju(Matrix, b, equations);
if (brojac == equations && b[pozicija] == 0)
{
    Console.WriteLine("Sustav ima beskonačno mnogo rješenja.");

}
else if (brojac == equations && b[pozicija] != 0)

    Console.WriteLine("Sustav nema rješenja.");

else
{
    Console.WriteLine("Rezultati:");
    for (int i = 0; i < equations; i++)
    {
        Console.WriteLine("x[{0}] = {1}", i + 1, b[i]); //petlja za ispis svakog pojedinacnog rezultata x-a u novom retku
            }
}
```

*Slika 4.3.5. Ispis rezultata sustava ako postoje, odnosno odgovarajuće poruke ako ne postoje ili ih je beskonačno*

```
2 references
public static void DisplayMatrix(double[,] matrix, double[] b)
//metoda za prikazivanje matricnog oblika zapisa
{
    int length;
    length = Convert.ToInt32(Math.Sqrt(matrix.Length));
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length; j++)
        {
            Console.Write("{0}\t", matrix[i, j]);
        }

        Console.Write("|{0}", b[i]);   //ispisuje onaj parametar nakon jednako iz jednadzbi u zadnjem stupcu
        Console.WriteLine();
    }


}
```

*Slika 4.3.6. Metoda za prikazivanje matričnog oblika zapisa*

```
1 reference
public static int PronadiBrojBrojaca(double[,] a, double[] b, int n)
{
    int brojac = 0;

    for (int i = 0; i < n; i++)
    {
        brojac = 0;
        for (int j = 0; j < n; j++)
        {
            if (a[i, j] == 0)
                brojac = brojac + 1;
        }
            if (brojac == n && b[i] == 0)
            {
                break;
            }
            if (brojac == n && b[i] != 0)
            {
                break;
            }
    }
    return brojac;
}
```

*Slika 4.3.7. Metoda za pronalazak broja brojača*

```
1 reference
public static int PronadiPoziciju(double[,] a, double[] b, int n)
{
    int pozicija = 0;

    for (int i = 0; i < n; i++)
    {
        int brojac = 0;
        for (int j = 0; j < n; j++)
        {
            if (a[i, j] == 0)
                brojac = brojac + 1;
        }

            if (brojac == n && b[i] == 0)
            {
                    pozicija = i;
                    break;
            }

            if (brojac == n && b[i] != 0)
            {
                    pozicija = i;
                    break;
            }
    }
    return pozicija;
}
```

*Slika 4.3.8. Metoda za pronalazak pozicije brojača*

Zadnje dvije metode služe za pronalazak nul redaka unutar zadane matrice, te pomažu odrediti ima li zadani sustav beskonačno mnogo rješenja ili ih nema uopće.

## 4.4. Testiranje aplikacije

Pri pokretanju aplikacije otvara se konzola gdje se objašnjava koja je svrha zadane aplikacije te kako ju koristiti.

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
```

*Slika 4.4.1. Prikaz konzole pri prvom otvaranju aplikacije*

U slučaju da korisnik unese negativan broj jednadžbi, ispisuje se odgovarajuća poruka sa objašnjenjem, te se od korisnika traži ponovni unos broja jednadžbi.

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
-1
Ne može postojati negativan broj jednadžbi.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
-34
Ne može postojati negativan broj jednadžbi.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
-1000
Ne može postojati negativan broj jednadžbi.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
-5
Ne može postojati negativan broj jednadžbi.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
```

*Slika 4.4.2. Konzola u slučaju unosa negativnog broja jednadžbi*

U slučaju da korisnik unese nulu kao broj jednadžbi ispisuje se poruka gdje se objašnjava korisniku da program nema jednadžbe na kojima bi izvršavao ikakve matematičke operacije. Program traži ponovni unos broja jednadžbi od korisnika.

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
0
Ne postoje jednadžbe na kojima bi se program izvodio.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
```

*Slika 4.4.3. Konzola u slučaju unosa nule*

Ako korisnik unese jedinicu, program ispisuje na konzolu kako je potrebno najmanje dvije jednadžbe kako bi se mogla izvoditi Gauss-Jordanova metoda eliminacije. Također, nakon poruke, traži ponovan unos broja jednadžbi od korisnika sve dok se ne unese dobar broj jednadžbi koji zadovoljava sve navedene kriterije.

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
1
Program zahtijeva najmanje dvije jednadžbe za izvođenje Gauss-Jordanove eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
```

*Slika 4.4.4. Konzola u slučaju unosa jedinice*

Kada korisnik unese pozitivan broj jednadžbi koji je veći od jedan, program nudi unos parametara jednadžbi koji se nalaze uz nepoznanice. Unos parametara napravljen je u matričnom obliku. Unos svakog pojedinog elementa sprema se u dvodimenzionalnu matricu Matrix[,] počevši od elementa Matrix[0,0] pa sve do elementa Matrix[n-1, n-1] gdje ***n*** predstavlja broj jednadžbi, odnosno nepoznanica koje je korisnik unio. Također je potreban unos rješenja jednadžbi koja se pohranjuju unutar jednodimenzionalne matrice b[n-1].

### 4.4.1. Sustav ima rješenje

Primjer 1. Unos matrice 3x3

Unosimo zadanu matricu:

$$\left[\begin{array}{ccc|c} 1 & -4 & -5 & 1 \\ 2 & 4 & -2 & -22 \\ -4 & 4 & -6 & -34 \end{array}\right]$$

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
3
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: 1
Matrix[0,1]: -4
Matrix[0,2]: -5
b[0]: 1

Matrix[1,0]: 2
Matrix[1,1]: 4
Matrix[1,2]: -2
b[1]: -22

Matrix[2,0]: -4
Matrix[2,1]: 4
Matrix[2,2]: -6
b[2]: -34
```

*Slika 4.4.5. Unos parametara matrice 3x3*

Nakon unosa svih parametara jednadžbi, odnosno elemenata matrice, ispisuje se matrični oblik unesenih *n* linearnih jednadžbi sa njihovim rješenjima. Rješenja su prikazana sa desne strane te su odvojena okomitom isprekidanom crtom radi jednostavnije preglednosti.

Nakon toga primjenjuje se Gauss-Jordanova eliminacija nad zadanom matricom te se ispisuje matrica nakon provedenog postupka eliminacije.

Posebno se još ispod prikaza matrice ispisuju konačna rješenja.

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
3
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: 1
Matrix[0,1]: -4
Matrix[0,2]: -5
b[0]: 1

Matrix[1,0]: 2
Matrix[1,1]: 4
Matrix[1,2]: -2
b[1]: -22

Matrix[2,0]: -4
Matrix[2,1]: 4
Matrix[2,2]: -6
b[2]: -34

Prikaz zadane matrice:
1       -4      -5      |1
2       4       -2      |-22
-4      4       -6      |-34
__________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0       0       |0
0       1       0       |-4
0       0       1       |3
__________________________________
Rezultati:
x[1] = 0
x[2] = -4
x[3] = 3

C:\Users\GamingPC\source\repos\GaussJordan\GaussJordan\bin\Debug\netc
reapp3.1\GaussJordan.exe (process 7224) exited with code 0.
To automatically close the console when debugging stops, enable Tools
>Options->Debugging->Automatically close the console when debugging s
ops.
```

*Slika 4.4.6. Prikaz rješenja trodimenzionalne matrice*

Primjer 2. Unos matrice 2x2

Unosimo zadanu matricu:

$$\left[\begin{array}{cc|c} -4 & 6 & 6 \\ -3 & 2 & 16 \end{array}\right]$$

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
2
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: -4
Matrix[0,1]: 6
b[0]: 8

Matrix[1,0]: -3
Matrix[1,1]: 2
b[1]: 16

Prikaz zadane matrice:
-4      6       |8
-3      2       |16
___________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0       |-8
0       1       |-4
___________________________________
Rezultati:
x[1] = -8
x[2] = -4

C:\Users\GamingPC\source\repos\GaussJordan\GaussJordan\bin\Debug\netcor
pp3.1\GaussJordan.exe (process 11000) exited with code 0.
To automatically close the console when debugging stops, enable Tools->
```

*Slika 4.4.7. Prikaz rješenja dvodimenzionalne matrice*

Primjer 3. Unos matrice 4x4

Unosimo zadanu matricu:

$$\left[\begin{array}{cccc|c} -4 & 6 & 2 & 4 & 8 \\ -3 & 2 & 11 & 5 & 16 \\ -1 & 0 & 2 & 1 & -1 \\ 2 & 1 & 0 & 9 & -6 \end{array}\right]$$

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
4
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: -4
Matrix[0,1]: 6
Matrix[0,2]: 2
Matrix[0,3]: 4
b[0]: 8

Matrix[1,0]: -3
Matrix[1,1]: 2
Matrix[1,2]: 11
Matrix[1,3]: 5
b[1]: 16

Matrix[2,0]: -1
Matrix[2,1]: 0
Matrix[2,2]: 2
Matrix[2,3]: 1
b[2]: -1

Matrix[3,0]: 2
Matrix[3,1]: 1
Matrix[3,2]: 0
Matrix[3,3]: 9
b[3]: -6

Prikaz zadane matrice:
-4      6       2       4       |8
-3      2       11      5       |16
-1      0       2       1       |-1
2       1       0       9       |-6
__________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0       0       0       |4.358208955223882
0       1       0       0       |4.761194029850748
0       0       1       0       |2.7611940298507474
0       0       0       1       |-2.164179104477613
__________________________________
Rezultati:
x[1] = 4.358208955223882
x[2] = 4.761194029850748
x[3] = 2.7611940298507474
x[4] = -2.164179104477613
```

*Slika 4.4.8. Prikaz rješenja četverodimenzionalne matrice*

## 4.4.2. Sustav nema rješenja

Primjer 1. Unos matrice 3x3:

$$\left[\begin{array}{ccc|c} 1 & -1 & 1 & 2 \\ 2 & 2 & 3 & 1 \\ 3 & 1 & 4 & 5 \end{array}\right]$$

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss-Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
3
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: 1
Matrix[0,1]: -1
Matrix[0,2]: 1
b[0]: 2

Matrix[1,0]: 2
Matrix[1,1]: 2
Matrix[1,2]: 3
b[1]: 1

Matrix[2,0]: 3
Matrix[2,1]: 1
Matrix[2,2]: 4
b[2]: 5

Prikaz zadane matrice:
1       -1      1       |2
2       2       3       |1
3       1       4       |5
__________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0       NaN     |-?
0       1       NaN     |-?
0       0       NaN     |?
__________________________________
Rezultati:
x[1] = -?
x[2] = -?
x[3] = ?
```

*Slika 4.2.9. Prikaz na konzoli kada sustav nema rješenja sa primjerom trodimenzionalne matrice*

Sustav nema rješenja kada ne postoji uređena *n*-torka $(x_1, x_2, \ldots, x_n)$ koja zadovoljava svih *n* linearnih jednadžbi. U zadanom primjeru možemo vidjeti kako zadnji redak matrice govori da je 0 = 2, što je nemoguće, pa time zaključujemo kako ovaj sustav nema rješenja. Kada zadani sustav nema rješenja, program ispisuje odgovarajuću poruku na konzolu.

Primjer 2. Unos matrice 2x2

Unosimo matricu:

$$\left[\begin{array}{cc|c} 3 & 0 & 16 \\ 17 & 0 & 0 \end{array}\right]$$

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss - Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
2
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: 3
Matrix[0,1]: 0
b[0]: 16

Matrix[1,0]: 17
Matrix[1,1]: 0
b[1]: 0

Prikaz zadane matrice:
3       0       |16
17      0       |0
__________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0       |5.333333333333333
0       0       |-90.66666666666666
__________________________________
Sustav nema rješenja.
```

*Slika 4.2.10. Prikaz na konzoli kada sustav nema rješenja sa primjerom dvodimenzionalne matrice*

Primjer 3. Unos matrice 4x4

Unosimo matricu:

$$\left[\begin{array}{cccc|c} 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss - Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
4
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: 1
Matrix[0,1]: 0
Matrix[0,2]: 1
Matrix[0,3]: 1
b[0]: 1

Matrix[1,0]: 1
Matrix[1,1]: 0
Matrix[1,2]: 1
Matrix[1,3]: 0
b[1]: 0

Matrix[2,0]: 1
Matrix[2,1]: 0
Matrix[2,2]: 0
Matrix[2,3]: 0
b[2]: 1

Matrix[3,0]: 1
Matrix[3,1]: 0
Matrix[3,2]: 0
Matrix[3,3]: 0
b[3]: 0

Prikaz zadane matrice:
1       0       1       1       |1
1       0       1       0       |0
1       0       0       0       |1
1       0       0       0       |0
________________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0       0       0       |1
0       0       0       -1      |-1
0       0       1       1       |-0
0       0       0       0       |-1
________________________________________
Sustav nema rješenja.
```

*Slika 4.2.11. Prikaz na konzoli kada sustav nema rješenja sa primjerom četverodimenzionalne matrice*

## 4.4.3. Sustav ima beskonačno rješenja

Primjer 1. Unos matrice 3x3

Unosimo matricu:

$$\left[\begin{array}{ccc|c} 3 & 2 & -1 & 5 \\ 1 & -1 & 1 & -7 \\ 4 & 1 & 0 & -2 \end{array}\right]$$

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss - Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
3
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: 1
Matrix[0,1]: -1
Matrix[0,2]: 1
b[0]: -7

Matrix[1,0]: 3
Matrix[1,1]: 2
Matrix[1,2]: -1
b[1]: 5

Matrix[2,0]: 4
Matrix[2,1]: 1
Matrix[2,2]: 0
b[2]: -2

Prikaz zadane matrice:
1       -1      1       |-7
3       2       -1      |5
4       1       0       |-2
__________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0       0.19999999999999996     |-1.7999999999999998
0       1       -0.8    |5.2
0       0       0       |0
__________________________________
Sustav ima beskonačno mnogo rješenja.
```

*Slika 4.2.12. Prikaz na konzoli kada sustav ima beskonačno mnogo rješenja sa primjerom trodimenzionalne matrice*

Sustav ima beskonačno mnogo rješenja kada se nakon Gauss-Jordanovog postupka eliminacije nađe redak kojemu su svi elementi nule. To znači da zbroj prve i druge jednadžbe daje treću jednadžbu. Rješenja ovakvog sustava mogu se zapisati koristeći pomoćni parametar *t* na sljedeći način:

$$x_1 = -\frac{1}{5}t - \frac{9}{5}$$

$$x_2 = \frac{4}{5}t + \frac{26}{5}$$

$$x_3 = t, t \in \mathbb{R}$$

Sve tri jednadžbe međusobno su ovisne o parametru *t* koji može biti bilo koji realan broj, zato ovakav sustav ima beskonačno mnogo rješenja.

Primjer 2. Unos matrice 2x2

Unosimo matricu:

$$\left[\begin{array}{cc|c} 13 & 10 & -13 \\ 0 & 0 & 0 \end{array}\right]$$

```
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss - Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
2
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: 13
Matrix[0,1]: 10
b[0]: -13

Matrix[1,0]: 0
Matrix[1,1]: 0
b[1]: 0

Prikaz zadane matrice:
13      10      |-13
0       0       |0
__________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0.7692307692307693      |-1
0       0       |0
__________________________________
Sustav ima beskonačno mnogo rješenja.
```

*Slika 4.2.13. Prikaz na konzoli kada sustav ima beskonačno mnogo rješenja sa primjerom dvodimenzionalne matrice*

Primjer 3. Unos matrice 4x4

Unosimo matricu:

$$\left[\begin{array}{cccc|c} 13 & 10 & -13 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

```
Dobrodošli!
Ovaj program je napravljen za rješavanje sustava linearnih jednadžbi
Gauss - Jordanovom metodom eliminacije.
Unesite broj jednadžbi (broj jednadžbi mora biti veći od jedan):
4
Unesite parametre jednadžbi:
Matrica b predstavlja rezultate vaših jednadžbi.
Matrix[0,0]: 13
Matrix[0,1]: 10
Matrix[0,2]: -13
Matrix[0,3]: -10
b[0]: 0

Matrix[1,0]: 0
Matrix[1,1]: 0
Matrix[1,2]: 0
Matrix[1,3]: 0
b[1]: 0

Matrix[2,0]: 1
Matrix[2,1]: 2
Matrix[2,2]: 3
Matrix[2,3]: 4
b[2]: 5

Matrix[3,0]: 0
Matrix[3,1]: 0
Matrix[3,2]: 0
Matrix[3,3]: 0
b[3]: 0

Prikaz zadane matrice:
13      10      -13     -10     |0
0       0       0       0       |0
1       2       3       4       |5
0       0       0       0       |0
__________________________________
Matrica nakon Gauss-Jordanove eliminacije:
1       0       0       0.423076923076923       |-3.125
0       1       0       0       |4.0625
0       0       1       1.1923076923076923      |0
0       0       0       0       |0
__________________________________
Sustav ima beskonačno mnogo rješenja.
```

*Slika 4.2.14. Prikaz na konzoli kada sustav ima beskonačno mnogo rješenja sa primjerom četverodimenzionalne matrice*

# 5. ZAKLJUČAK

Osnovni zadaća ovog završnog rada bila je izraditi aplikaciju kojoj je funkcija pomoći studentima i profesorima prilikom rješavanja sustava linearnih jednadžbi. Aplikacija omogućava korisniku unos broja jednadžbi, odnosno, nepoznanica te na osnovu unesenih podataka, aplikacija dodatno unosi sve potrebne parametre tih jednadžbi. Aplikacija najprije ispiše unesene jednadžbe od strane korisnika u matričnom obliku te zatim Gauss - Jordanovom metodom eliminacije rješava zadani sustav jednadžbi te isti nakon toga ispisuje u matričnom obliku. Aplikacija posebno nakon toga ispisuje rješenja zadanog sustava, ukoliko postoje. Ako zadani sustav nema rješenja ispisuje se odgovarajuća poruka, isto tako se ispisuje poruka ako sustav ima beskonačno rješenja. Aplikacija je razvijena u Microsoft Visual Studio-u te je napravljena u C#-u. U teorijskom dijelu rada opisan je programski jezik C#, njegove temeljne značajke i funkcije te njegove razlike u usporedbi sa C++ jezikom, te su također obrađene teme iz polja linearne algebre naziva „*Rješavanje sustava linearnih jednadžbi*", „*Gaussova metoda eliminacije*", „*Cramerovo pravilo*" te „*Kronecker-Capellijev teorem*". Također je dan opis metode parcijanog pivotiranja koji se primjenjuje u kodu za rješavanje sutava. U poglavlju naziva „*Aplikacija za rješavanje sustava linearnih jednadžbi*", dan je detaljajn opis funkcioniranja koda te izlistanje istoga. Nakon toga navedeni kod je testiran i rezultati su objašnjeni na više različitih primjera. U budućnosti bi trebalo doraditi aplikaciju tako da korisnik ne mora unjeti jednak broj jednadžbi i nepoznanica. Rad bi se nadalje mogao unaprijediti pisanjem algoritma koji bi se uvrstio u sadašnji kod, a čija bi funkcija bila pronalaženje općenitog rješenja sustava jednadžbi u slučaju kada sustav ima beskonačno mnogo rješenja. Također, uz to, trebalo bi napisati metodu za ispis tih rješenja na pravilan i korisniku lako čitljiv način na konzolu. U slučaju kada bi imali beskonačno mnogo rješenja program ne bi samo ispisao poruku, nego bi uz poruku bila ispisana i općenita rješenja. Također aplikaciji bi se moglo napraviti grafičko sučelje za iOS, android ili računalo, gdje bi korisnik upisivao podatke i iščitavao ih umjesto da to obavlja na konzoli.

# 6. LITERATURA

[1] Matrix Calculator stranica na računalu https://matrix.reshish.com/gauss-jordanElimination.php (stranica posjećena: 13. rujna 2022.)

[2] Gauss-Jordan Elimination Calculator stranica na računalu https://onlinemschool.com/math/assistance/equation/gaus/ (stranica posjećena: 13. rujna 2022.)

[3] Razlika između C# i C++ jezika https://www.guru99.com/cpp-vs-c-sharp.html (stranica posjećena: 6. srpnja 2022.)

[4] Wikipedia C# https://bs.wikipedia.org/wiki/C%E2%99%AF (stranica posjećena: 6. srpnja 2022.)

[5] Aplikacije za linearne jednadžbe https://www.cuemath.com/algebra/applications-linear-equations/ (stranica posjećena: 7. srpnja 2022.)

[6] Teorija sustava. Hrvatska enciklopedija, mrežno izdanje. http://www.enciklopedija.hr/Natuknica.aspx?ID=60892 (pristupljeno: 8. srpnja 2022.)

[7] Wikipedia, Sustav linearnih jednadžbi https://hr.wikipedia.org/wiki/Sustav_linearnih_jednad%C5%BEbi (stranica posjećena: 8. srpnja 2022.)

[8] D. Bakić, Linearna Algebra, Školska knjiga, Masarykova 28, Zagreb, 2008.

[9] N. Elezović, Linearna Algebra, Zagreb, 2006.

[10] Kronecker-Capellijev teorem http://www.mathematics.digital/matematika1/predavanja/node38.html (stranica posjećena: 8. srpnja 2022.)

# 7. SAŽETAK

Aplikacija za rješavanje sustava linearnih jednadžbi

Ovim završnim radom opisan je postupak izrade aplikacije za riješavanje sustava linearnih jednadžbi Gauss-Jordanovom metodom eliminacije. Prvi korak pri izradi programa bila je detaljna obrada matematičke teorije vezane uz linearne jednadžbe, sustave linearnih jednadžbi, matrice, a potom i razne metode i načini rješavanja istih. Zatim je uslijedila razrada ideje, pisanje koda i testiranje istog. Sljedeći korak je bila izrada dizajna i izgleda konzole pri pokretanju programa. Pošto se preko nje odvija komunikacija sa korisnikom, cilj je bio napraviti konzolu koja će ispisati i objasniti korisniku porukama za što služi zadana aplikacija, te kako se ona koristi. Objašnjen je način unosa podataka te su jasno prikazani dobiveni rezultati koje korisnika najviše i zanimaju. Kao integrirano razvojno okruženje, za izradu aplikacije korišten je Microsoft Visual Studio te je kod aplikacije pisan u programskom jeziku C#. Bitno je bilo dobro istražiti i informirati se o mogućnostima i načinima implementacije željenog dizajna, te poboljšanju jednostavnosti i funkcionalnosti koda. Napravljena aplikacija služit će mnogim korisnicima koji budu imali potrebu raditi sa sustavima linearnih jednadžbi.

**Ključne riječi:** sustav linearnih jednadžbi, C#, matrica, Gaussova metoda eliminacije, Gauss-Jordan

# 8. ABSTRACT

Application for solving systems of linear equations

This final paper describes the process of creating an application for solving systems of linear equations using the Gauss-Jordan elimination method. The first step in creating the program was a detailed processing of the mathematical theory related to linear equations, systems of linear equations, matrices, and then various methods and ways of solving them. Then, followed the development of the idea, writing the code and testing it. The next step was to create the design and appearance of the console when launching the program. Since communication with the user takes place through it, the goal was to create a console that will print and explain to the user with messages what the default application is for and how it is used. The data entry method is described, and the results are presented, which the user is most interested in. As an integrated development environment, Microsoft Visual Studio was used to create the application, and the application code was written in the C# programming language. It was essential to research and get informed about the possibilities and ways of implementing the desired design and improving the code's simplicity and functionality. The created application will serve many users who will need to work with systems of linear equations.

**Keywords:** a system of linear equations, C#, matrix, Gauss elimination method, Gauss-Jordan

# 9. ŽIVOTOPIS

Ena Spremo rođena je 13. listopada 2000. godine u Osijeku, Hrvatska. Osnovnoškolsko obrazovanje započinje 2007. godine u Osnovnoj Školi Ivana Filipovića u Osijeku. Nakon završetka osnovnoškolskog obrazovanja, 2015. se upisuje u III. Gimnaziju u Osijeku, smjer Matematička gimnazija. Upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija, 2019. godine, smjer Računarstvo, koji još uvijek pohađa. Godine 2020. bila je članica IEEE udruge. Posjeduje vrlo dobro znanje u govoru, čitanju i pisanju engleskog jezika. Posjeduje i vrlo dobro znanje u radu s Microsoft Office alatima, osnovno znanje opisnog jezika HTML i CSS-a, dobro znanje programskih jezika C, C++ i C#, te osnovno znanje VHDL i Kotlin programskih jezika.