

**Gaussin eliminaatio**

Koululuokka: K12

**Sisältö**

[Määritelmä 3](#_Toc126589375)

[Menetelmän ratkaisumenetelmä 3](#_Toc126589376)

[Toteutus 5](#_Toc126589377)

[Algoritmi 7](#_Toc126589378)

[Ohjelma c++ 9](#_Toc126589379)

[Harjoitus 1 11](#_Toc126589380)

[Harjoitus 2 12](#_Toc126589381)

[Harjoitus 3 15](#_Toc126589382)

[Harjoitus 4 (BAC-kokeista) 18](#_Toc126589383)

[Harjoitus 5 19](#_Toc126589384)

[Harjoitus 6 20](#_Toc126589385)

[Harjoitus 7 23](#_Toc126589386)

[Lähteet 25](#_Toc126589387)

# Määritelmä

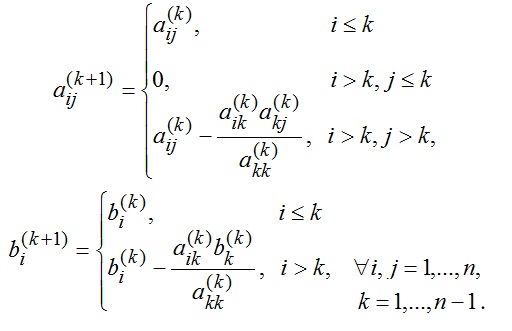
Matematiikassa Gaussin eliminaatio (kutsutaan myös rivin vähentämiseksi) on menetelmä, jota käytetään lineaaristen yhtälöjärjestelmien ratkaisemiseen. Se on nimetty kuuluisan saksalaisen matemaatikon Carl Friedrich Gaussin mukaan, joka kirjoitti tästä menetelmästä, mutta ei keksinyt sitä.

Gaussin eliminaatio on tekniikka matriisin A muuttamiseksi ylempään kolmiomaiseen muotoon. Muunnosmatriisi T on unitaarinen alempi kolmiomatriisi, joka saadaan muodon T = Tn-1Tn-2 alkeiskolmiomuunnosten sekvenssinä (tulona). . . T1, jossa matriisit Tp ovat alemman kolmion muotoisia, kertaluokkaa n:

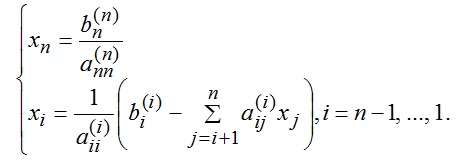
# ****Menetelmän ratkaisumenetelmä****

Huomioimme aluksi, A(1)=A, b(1)=b, yläindeksi, joka edustaa vaihetta.

Gaussin eliminointimenetelmän toistuvuussuhteet ovat:



Systeemi ratkaistaan ​​käänteiskorvausmenetelmällä suhteiden mukaan:



Gaussin eliminaatio on menetelmä muotoisten matriisiyhtälöiden ratkaisemiseksi Ax=b. Algoritmi ei ole monimutkainen, mutta se esiintyy melko usein ohjelmointikilpailuissa ja sillä on mielenkiintoisia sovelluksia.

Oletetaan, että meillä on seuraava järjestelmä:

\begin{pmatrix}
a_{11} &  a_{12}  & \ldots & a_{1n}\\
a_{21} &  a_{22}  & \ldots & a_{2n}\\
\vdots & \vdots   & \ddots & \vdots\\
a_{n1} &  a_{n2} & \ldots  & a_{nn}
\end{pmatrix} * 
\begin{pmatrix}
x_{1} \\
x_{2} \\
\vdots \\
x_{n}
\end{pmatrix} = 
\begin{pmatrix}
b_{1} \\
b_{2} \\
\vdots \\
b_{n}
\end{pmatrix}

Systeemin ratkaisemiseksi muunnamme kaikki laajennetun matriisin päädiagonaalin alapuolella olevat alkiot nollaksi, jotta jokainen tuntematon voidaan kirjoittaa vain tuntemattomien, joilla on korkeammat indeksit..


\begin{pmatrix}
a_{11} &  a_{12} & \ldots & a_{1n} & b_{1}\\
a_{21} &  a_{22} & \ldots & a_{2n} & b_{2}\\
a_{31} &  a_{32} & \ldots & a_{3n} & b_{3}\\
\vdots & \vdots  & \ddots & \vdots \\
a_{n1} &  a_{n2} & \ldots  & a_{nn} & b_{n}
\end{pmatrix}

\rightarrow

\begin{pmatrix}
a_{11} &  a_{12} & \ldots & a_{1n} & b_{1}\\
0      &  a'_{22} & \ldots & a'_{2n} & b'_{2}\\
0      &  0      & \ldots & a'_{3n} & b'_{3}\\
\vdots & \vdots  & \ddots & \vdots \\
0 &  0 & \ldots  & a'_{nn} & b'_{n}
\end{pmatrix}


Kun matriisi on tässä muodossa, voimme helposti löytää jokaisen tuntemattoman yhtälöstä, jossa pienempien indeksien tuntemattomien kerroin on 0:  
![
\:
x_{n}=\frac{b'_{n}}{a'_{nn}}](data:image/gif;base64,R0lGODlhRgAeAPMPAAAAABERESIiIjMzM0RERFVVVWZmZnd3d4iIiJmZmaqqqru7u8zMzN3d3e7u7v///yH5BAAAAAAALAAAAABGAB4AAATx8MlJq7046827/5cAjuSoGGWqSouQTMeyzmTQTAWte+3EvLtg5jBYJIDCpIUgewSUUIrowXhGo7mHAXGNKhQJ7kMRA3eVDAaBeugoFgcE4rA+Z8iPg8KxSfAXAQ4/HweFhodtJAcMeQuMGjcPCFlnAJaXmJmaAHUWBWJ2SgGPMmQJCwqUFYisI1MfCk8LACw3ZG0OtF0nJkUMcQuRDwWMqWcxQlMGqHxYQgxZegpBR3Avg6EzCCh509k6Ao+v3zNWDQQOTeQqR2BbEmSoQF8LcOsgaWu/EqcyBs333LTR04zbAIABNyxq9Igbk4QQI2aIAAA7)  
x_{n-1}=\frac{b'_{n-1}-a'_{n-1n}*x_{n}}{a'_{n-1n-1}}  
 \vdots  
 x_{i}=\frac{b'_{i}-\sum\limits_{j=i+1}^n a'_{ij}*x_{j}}{a'_{ii}}

Nyt kun tiedämme kuinka löytää tuntemattomat matriisin kolmiomuodosta, jäljellä on vain muuntaa matriisi.

Muuttaaksemme matriisin kolmion muotoiseksi käytämme kahta operaatiota:  
L_{i} \longleftrightarrow L_{j} : kahden rivin vaihto  
L_{j} \longleftarrow L_{j}+a*L_{i} missä L_{i} on laajennetun matriisin rivi.

Esimerkiksi:


\begin{pmatrix}
2  &  1 & -1 & 8 \
-3 & -1 & 2 & -11 \
-2 & 1 & 2 & -3
\end{pmatrix} \longrightarrow

\begin{pmatrix}
2 & 1 & -1 & 8 \
0 & \frac{1}{2} & \frac{1}{2} & 1 \
0 & 2 & 1 & 5
\end{pmatrix} \longrightarrow
\begin{pmatrix}
2 & 1 & -1 & 8 \
0 & \frac{1}{2} & \frac{1}{2} & 1 \
0 & 0 & -1 & 1
\end{pmatrix}


Saadakseni toisen matriisin kerroin ensimmäisen rivin ja lisäsin sen toiselle riville ja sitten lisäsin sen viimeiselle riville (\frac{2}{2}=1). Saadakseni viimeisen matriisin kerroin toisen rivin  -\frac{2}{\frac{1}{2}}=-4.

Kuten esimerkistä voidaan nähdä, rakennetaan jokaisessa vaiheessa sarake ja viiva lopullisesta matriisista, sarake täytetään 0:lla kiinteän viivan alapuolella. Oletetaan, että haluamme muuntaa kaikki sarakkeen j rivin i alapuolella olevat elementit nollaksi. Jokaisella rivillä k ( k>i ) kerromme rivin i  -\frac{a_{kj}}{a_ij} ja lisäämme sen riville k, jolloin sarakkeen j elementti muuttuu 0 a_{ij}=0 meidän on etsittävä linjaa k(k > i) sellasta a_{kj}\neq 0. Jos tätä riviä ei ole olemassa, järjestelmällä ei ole ratkaisua. Näitä vaiheita soveltamalla pääsemme lopulta kolmiomatriisiin, josta löydämme tuntemattomat. Algoritmin monimutkaisuus on O(N^3)

# Toteutus

Alla oleva koodi ratkaisee myös tapauksen, jossa yhtälöitä on enemmän kuin tuntemattomia.

void elim(int n,int m,double s[][]) {*//* *järjestelmä, jossa on n tuntematonta m yhtälöä*

for(int i=1,j=1,k;i<=n && j<=m;) {

for(k=i;k<=n; ++k)

if(s[k][j]!=0) break;*//* *etsimme riviä nollien muodostamiseen sarakkeeseen j*

if(k>n) {*// En löytänyt riviä, jolle s[i][j] on nolla, joten siirrymme seuraavaan sarakkeeseen, rivi i ei ole viimeinen*

++j;

continue;

}

if(k!=i)for(int l=1; l<=m+1; ++l) swap(s[i][l],s[k][l]);*//* *vaihdamme rivit siten, että rivillä i ja sarakkeessa j on nollaelementti*

for(k=i+1; k<=n; ++k)

for(int l=m+1; l>=j; --l)

s[k][l]-=((s[k][j]\*s[i][l])/s[i][j]);*//* *käytämme muunnosa jokaiselle riville, joka on suurempi kuin i, jotta 0 sarakkeessa j rivin i alapuolella*

++i; ++j;

}

*//* *opimme tuntematonta*

for(int i=n; i;--i)

for(int j=1; j<=m+1; ++j) if(fabs(s[i][j])>EPS) {

*//* *koska yhtälöitä voi olla enemmän kuin tuntemattomia*

*//Etsimme jokaiselta riviltä ensimmäistä nollakerrointa oikealta vasemmalle*

if(j==m+1) {*//the line has no non-zero coefficients, so we have no solution*

g<<" Mahdotonta ";

exit(0);

}

x[j]=s[i][m+1];

for(int k=j+1; k<=m; ++k) x[j]-=s[i][k]\*x[k];

x[j]/=s[i][j];

break;*//we go to the previous line*

}

}

# Algoritmi

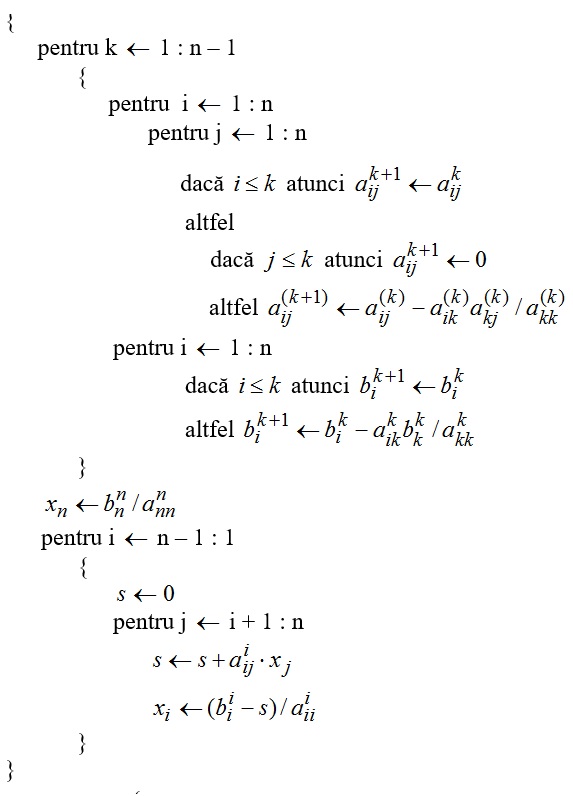
Gaussin eliminointimenetelmään liittyvä algoritmi on:

Tulot:

* n = järjestelmän yhtälöiden ja tuntemattomien lukumäärä
* A = järjestelmämatriisi
* b = vapaiden termien vektori

Lähdöt:

* x = ratkaisuvektori



# Ohjelma c++

#include "stdafx.h"

#include

using namespace std;

#define max 15

void main(void)

{

float s;

float a[max + 1][max + 1][max + 1], b[max + 1][max + 1], x[max + 1];

int n, i, j, k;

cout << "dati dimensiunea matricii" << endl; cin >> n;

cout << "dati matricea sistemului " << endl;

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

{

cout << "a[" << i << j << "]=";

cin >> a[i][j][1];

}

cout << endl;

cout << "dati vectorul termenilor liberi " << endl;

for (i = 1; i <= n; i++)

{

cout << "b[" << i << "]=";

cin >> b[i][1];

}

cout << endl;

for (k = 1; k <= n - 1; k++)

{

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

{

if (i <= k)a[i][j][k + 1] = a[i][j][k];

else if (j <= k)a[i][j][k + 1] = 0;

else a[i][j][k + 1] = a[i][j][k] - a[i][k][k] \* a[k][j][k] / a[k][k][k];

}

for (i = 1; i <= n; i++)

if (i <= k)b[i][k + 1] = b[i][k];

else b[i][k + 1] = b[i][k] - a[i][k][k] \* b[k][k] / a[k][k][k];

}

x[n] = b[n][n] / a[n][n][n];

for (i = n - 1; i >= 1; i--)

{

s = 0;

for (j = i + 1; j <= n; j++)

s = s + a[i][j][i] \* x[j];

x[i] = (b[i][i] - s) / a[i][i][i];

}

cout << "Likimääräinen ratkaisu on:" << endl;

for (i = 1; i <= n; i++)

cout << x[i] << ' ';

cout << endl;

system("pause");

}

# Harjoitus 1

Seuraavaa järjestelmää harkitaan:



Kertoimet kirjoitetaan taulukkomuodossa ja oikealla erillisessä sarakkeessa - vapaat jäsenet. Pylväs, jossa on ilmaisia ​​jäseniä, on erotettu kätevyyden vuoksi. Tämän sarakkeen sisältävää taulukkoa kutsutaan laajennetuksi.

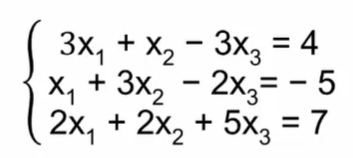


Lisäksi pääkerroinmatriisi on vähennettävä ylempään kolmiomaiseen muotoon. Tämä on pääkohta järjestelmän ratkaisemisessa Gaussin menetelmällä. Yksinkertaisesti, muutaman manipuloinnin jälkeen matriisin pitäisi näyttää tältä, jotta sen vasemmassa alakulmassa on vain nollia:

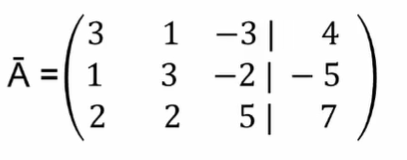


# Harjoitus 2

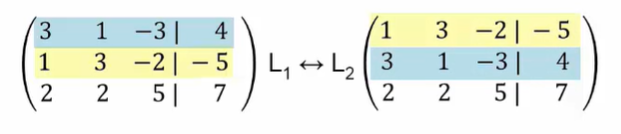
Seuraavaa järjestelmää harkitaan:

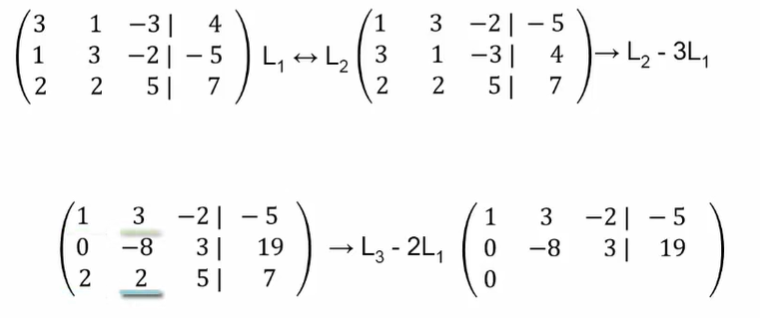


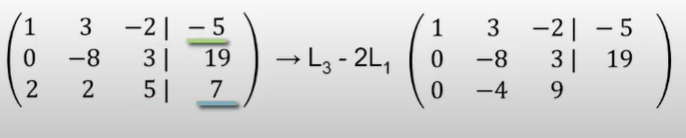
Järjestelmään liittyvä laajennettu matriisi on:



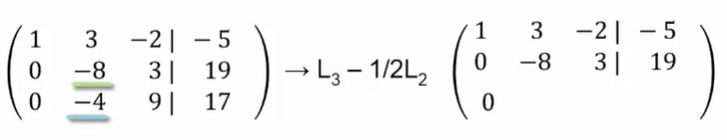
Vaihdoimme riviä L1 rivillä L2, jotta sillä olisi alhaisin arvo ensimmäisellä rivillä ensimmäisessä paikassa

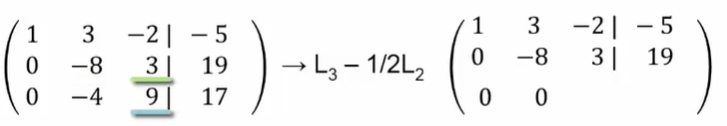


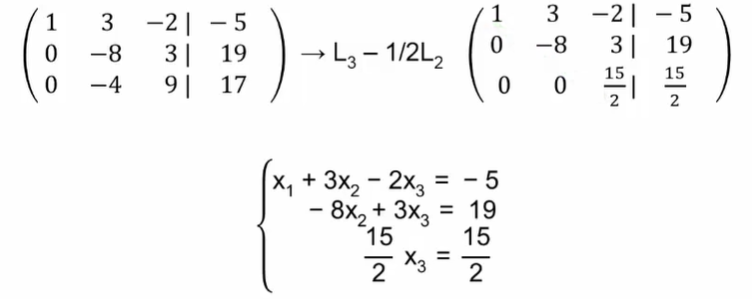


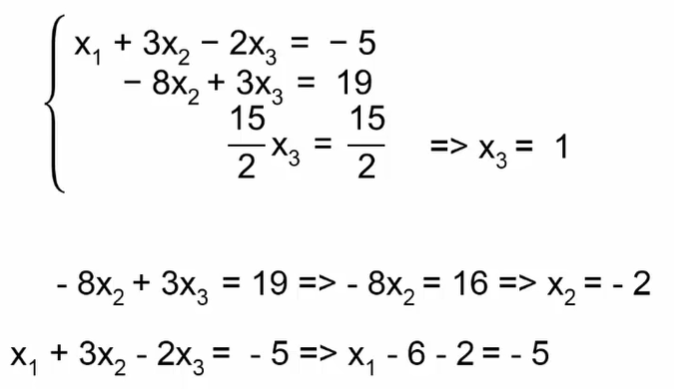












# Harjoitus 3

Käytä Gaussin eliminaatiomenetelmää järjestelmän ratkaisemiseen:



**Ratkaiseminen**:

Järjestelmään liittyvä matriisi A (vaiheessa 1) ja vapaiden termien vektori b ovat:





Ratkaisut ovat:

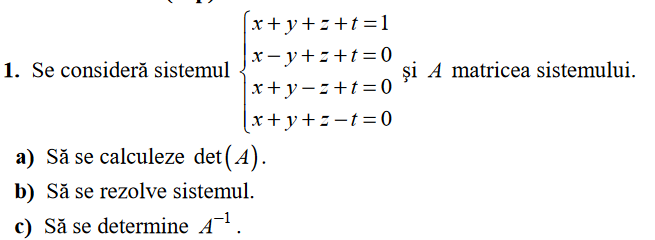


eli järjestelmän ratkaisu on:

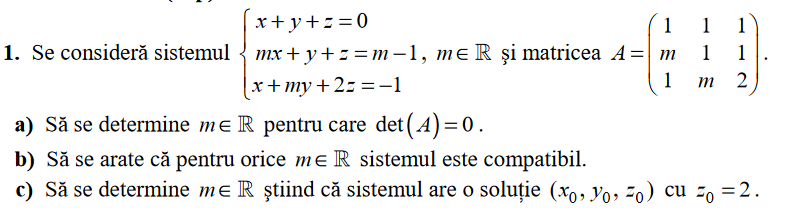


# Harjoitus 4 (BAC-kokeista)

BAC kokeista 2018



BAC kokeista 2018



# Harjoitus 5

Seuraavaa järjestelmää harkitaan:



Kertoimet kirjoitetaan taulukkomuodossa ja oikealla erillisessä sarakkeessa - vapaat jäsenet. Pylväs, jossa on ilmaisia ​​jäseniä, on erotettu kätevyyden vuoksi. Tämän sarakkeen sisältävää taulukkoa kutsutaan laajennetuksi.



Lisäksi pääkerroinmatriisi on vähennettävä ylempään kolmiomaiseen muotoon. Tämä on pääkohta järjestelmän ratkaisemisessa Gaussin menetelmällä. Yksinkertaisesti, pienen manipuloinnin jälkeen matriisin pitäisi näyttää tältä, jotta sen vasemmassa alakulmassa on vain nollia:



# 

# Harjoitus 6

Seuraavaa järjestelmää harkitaan:

Diagram

Description automatically generated with low confidence

Järjestelmään liittyvä laajennettu matriisi on:

A picture containing text, clock

Description automatically generated

Vaihdoimme riviä L1 rivillä L2, jotta sillä olisi alhaisin arvo ensimmäisellä rivillä ensimmäisessä paikassa

Diagram

Description automatically generated

Calendar

Description automatically generated with medium confidence

A picture containing text, clock, gauge

Description automatically generated



A picture containing logo

Description automatically generated

Logo

Description automatically generated

Diagram

Description automatically generated

Text, letter

Description automatically generated

# Harjoitus 7

Käytä Gaussin eliminaatiomenetelmää järjestelmän ratkaisemiseen:



**Ratkaiseminen**:

Järjestelmään liittyvä matriisi A (vaiheessa 1) ja vapaiden termien vektori b ovat:





Ratkaisut ovat:



eli järjestelmän ratkaisu on:



# Lähteet

<https://en.wikipedia.org/wiki/Regular_polygon>

<https://www.youtube.com/watch?v=qetSusATv2w>