

**Eliminazione gaussiana**

Grado scolastico: Scuola secondaria

**Indice**

Definizione

Il metodo di risoluzione del problema

Attuazione

Algoritmo

Il programma c++

Esercizio 1

Esercizio 2

Esercizio 3

Esercizio 4 (esami BAC)

Esercizio 5

Esercizio 6

Esercizio 7

Risorse

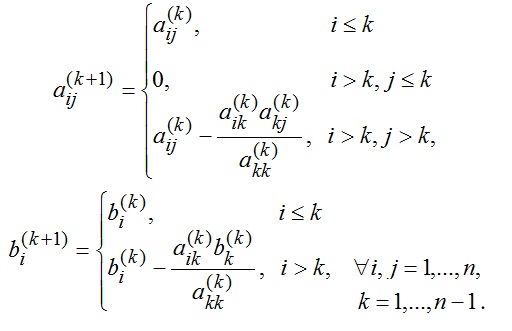
# Definizione

# In matematica, l'eliminazione gaussiana (detta anche riduzione di riga) è un metodo utilizzato per risolvere sistemi di equazioni lineari. Prende il nome da Carl Friedrich Gauss, un famoso matematico tedesco che scrisse di questo metodo ma non lo inventò.

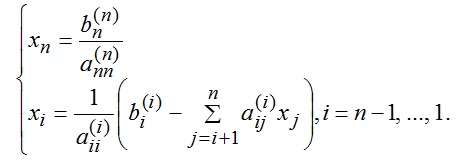
# L'eliminazione gaussiana è una tecnica per trasformare la matrice A in forma triangolare superiore. La matrice di trasformazione T è una matrice triangolare inferiore unitaria ottenuta come sequenza (prodotto) di trasformazioni triangolari inferiori elementari della forma T = Tn-1Tn-2 . . . T1, dove le matrici Tp sono triangolari inferiori di ordine n:

**Il metodo di risoluzione del metodo**

Notiamo, inizialmente, A(1)=A, b(1)=b, l'apice rappresenta lo stadio.

Le relazioni di ricorrenza del metodo di eliminazione gaussiana sono:

Il sistema viene risolto con il metodo della sostituzione inversa secondo le relazioni:



L'eliminazione gaussiana è un metodo per risolvere equazioni matriciali della forma Ax=b. L'algoritmo non è complicato, ma compare spesso nelle gare di programmazione e ha applicazioni interessanti.

Supponiamo di avere il seguente sistema:

\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}

Per risolvere il sistema trasformeremo a 0 tutti gli elementi sotto la diagonale principale della matrice espansa, in modo da poter scrivere ogni incognita solo in termini delle incognite con indici più alti.


\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}


Avendo la matrice in questa forma, possiamo facilmente trovare ogni incognita nell'equazione, dove le incognite con indici più bassi hanno il coefficiente 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}}

Ora che sappiamo come trovare le incognite dalla forma triangolare della matrice, non resta che trasformare la matrice.

Per trasformare la matrice in una forma triangolare applicheremo due operazioni  
L_{i} \longleftrightarrow L_{j} : lo scambio di due linee  
L_{j} \longleftarrow L_{j}+a*L_{i} dove  L_{i} è una riga della matrice estesa.

Ad esempio


\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}


Per ottenere la seconda matrice ho moltiplicato la prima riga per \frac{3}{2} e l'ho aggiunto alla seconda riga, e poi l'ho aggiunto all'ultima riga (\frac{2}{2}=1). Per ottenere l'ultima matrice ho moltiplicato la seconda riga per -\frac{2}{\frac{1}{2}}=-4.

Come si può vedere dall'esempio, a ogni passo costruiamo una colonna e una riga dalla matrice finale, la colonna viene riempita con 0 sotto la riga fissa. Supponiamo di voler convertire in 0 tutti gli elementi sotto la riga i sulla colonna j. Per ogni riga k ( k>i ) moltiplicheremo la riga i per -\frac{a_{kj}}{a_ij} e lo aggiungeremo alla riga k, facendo diventare 0 l'elemento della colonna j. Nel caso di a_{ij}=0 dobbiamo cercare una linea k (k > i) tale che a_{kj}\neq 0. Se questa linea non esiste, il sistema non ha soluzione. Applicando questi passaggi arriveremo infine a una matrice triangolare dalla quale troveremo le incognite. La complessità dell'algoritmo è O(N^3)

# Implementazione

Il codice seguente risolve anche il caso in cui ci siano più equazioni che incognite.

void elim(int n,int m,double s[][]) {*//* *sistema con n equazioni ad incognita m*

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

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

if(s[k][j]!=0) break;*//* *stiamo cercando una linea da utilizzare per formare gli zeri sulla colonna j*

if(k>n) {*// Non ho trovato nessuna riga per la quale s[i][j] è zero, quindi passiamo alla colonna successiva, la riga i non è quella finale* ++j;

continua;

}

if(k!=i)for(int l=1; l<=m+1; ++l) swap(s[i][l],s[k][l]);*//* *scambiamo le righe per avere un elemento nullo sulla riga i e sulla colonna j*

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]);*//* *applichiamo la trasformazione per ogni riga maggiore di i in modo da avere 0 sulla colonna j sotto la riga i*

++i; ++j;

}

*//* *impariamo l'ignoto*

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

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

*//* *perché è possibile avere più equazioni che incognite*

*//* *cerchiamo il primo coefficiente zero su ogni riga, che appare da destra a sinistra*

if(j==m+1) {*//* *la retta non ha coefficienti non nulli, quindi non abbiamo una soluzione*

g<<"Impossibile";

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;*//* *passiamo alla riga precedente*

}

}

# Algoritmo

L'algoritmo associato al metodo di eliminazione di Gauss è il seguente:

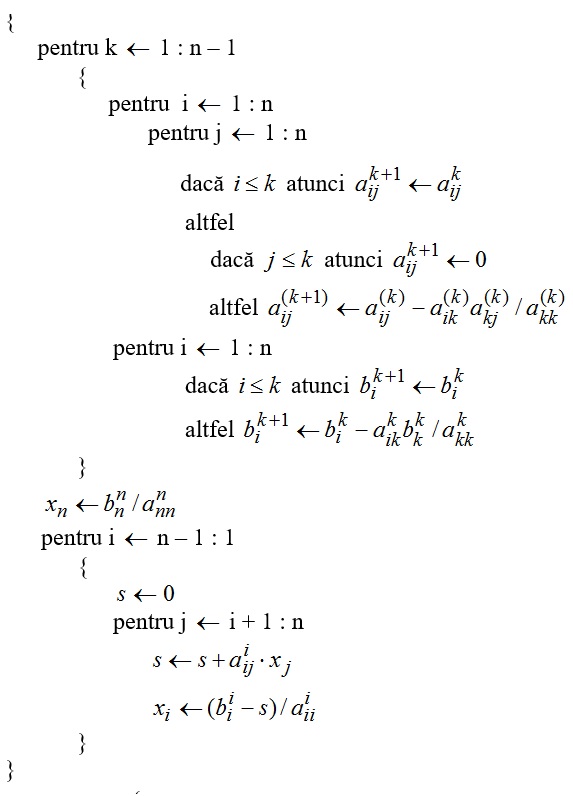
Ingressi:

- n = numero di equazioni e incognite del sistema

- A = matrice del sistema

- b = vettore dei termini liberi

Uscite: - x = il vettore soluzione



# Il programma 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 << "The approximate solution is:" << endl;

for (i = 1; i <= n; i++)

cout << x[i] << ' ';

cout << endl;

system("pause");

}

# Esercizio 1

Si considera il seguente sistema:



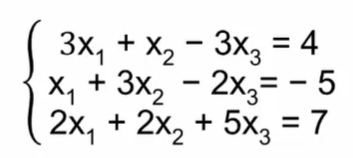
I coefficienti sono scritti in forma tabellare e, a destra, in una colonna separata - membri liberi. La colonna con i membri liberi è separata per comodità. L'array che include questa colonna è chiamato esteso.

Inoltre, la matrice dei coefficienti principali deve essere ridotta in forma triangolare superiore. Questo è il punto principale della risoluzione del sistema con il metodo gaussiano. Semplicemente, dopo alcune manipolazioni, la matrice dovrebbe avere questo aspetto, in modo che ci siano solo zeri nella sua parte inferiore sinistra:

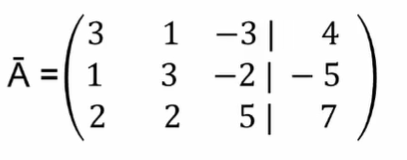


# Esercizio 2

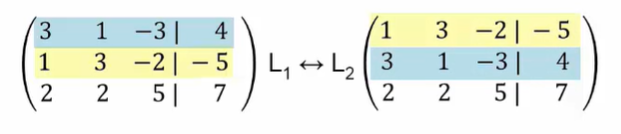
Si considera il seguente sistema:

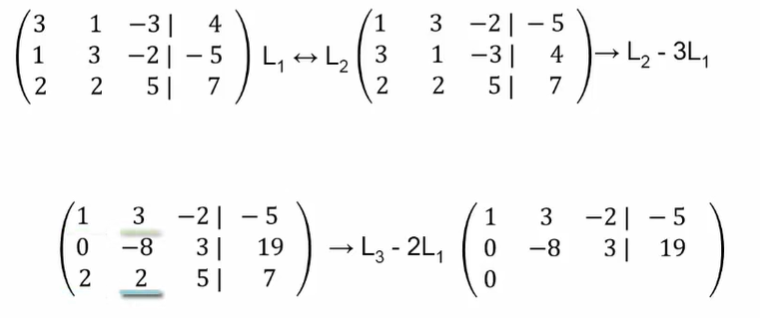


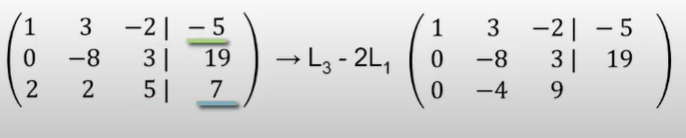
La matrice estesa associata al sistema è:



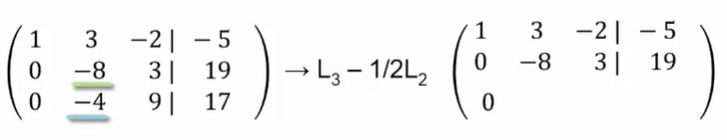
Stiamo cambiando la riga L1 con la riga L2 in modo da avere il valore più basso sulla prima riga nella prima posizione.

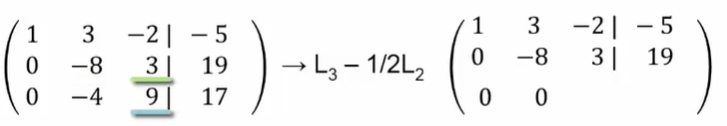


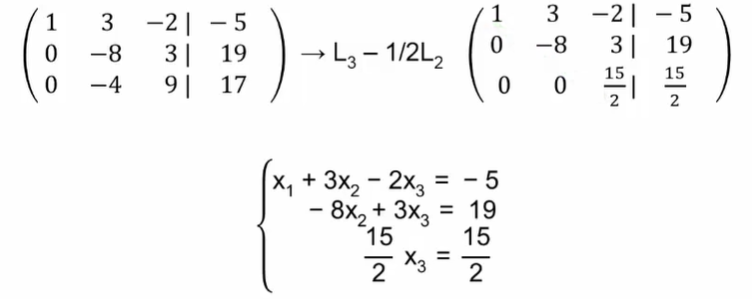


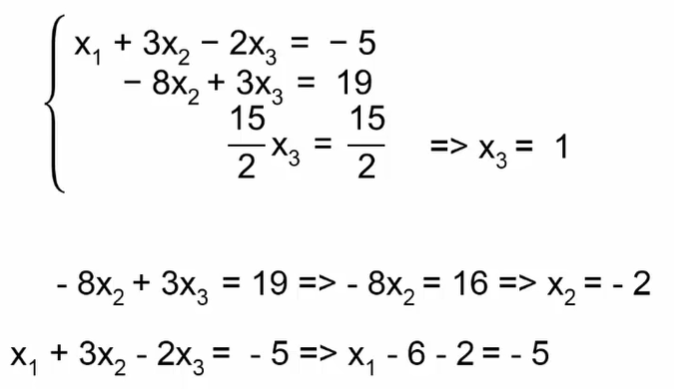












# Esercizio 3

Applicare il metodo di eliminazione gaussiana per risolvere il sistema:

**Procedimento**: La matrice A associata al sistema (al passo 1) e il vettore dei termini liberi b sono:





Le soluzioni sono:

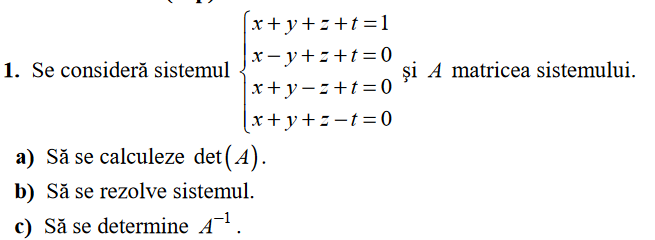


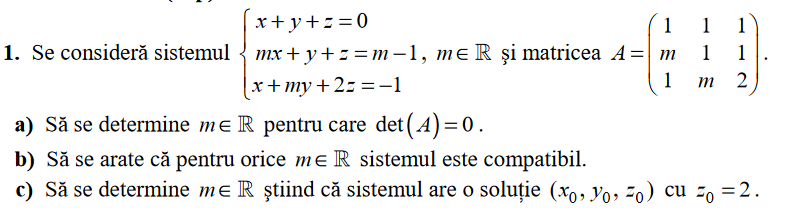
La soluzione del sistema è



# Esercizio 4 (esame BAC)

Esame BAC 2018



Esame BAC 2017

# Esercizio 5

Si considera il seguente sistema:



I coefficienti sono scritti in forma tabellare e, a destra, in una colonna separata - membri liberi. La colonna con i membri liberi è separata per comodità. La matrice che include questa colonna è chiamata estesa



Inoltre, la matrice dei coefficienti principali deve essere ridotta in forma triangolare superiore. Questo è il punto principale della risoluzione del sistema con il metodo gaussiano. Semplicemente, dopo alcune manipolazioni, la matrice dovrebbe avere questo aspetto, in modo che ci siano solo zeri nella sua parte inferiore sinistra:

# https://i2.wp.com/fb.ru/misc/i/gallery/86139/2455721.jpg

# Esercizio 6

Si considera il seguente sistema:

Diagram

Description automatically generated with low confidence

La matrice estesa associata al sistema è:

A picture containing text, clock

Description automatically generated

Stiamo cambiando la riga L1 con la riga L2 in modo da avere il valore più basso sulla prima riga nella prima posizione.

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

# Esercizio 7

Applicare il metodo di eliminazione gaussiana per risolvere il sistema:

**Soluzione**:

La matrice A associata al sistema (al passo 1) e il vettore dei termini liberi b sono:



Le soluzioni sono:



cioè la soluzione del sistema è:



# Risorse

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

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