

## Laboratorio Analisis Numérico

### Metodo de Gauss

Lehi Quincho Mamani

Cui: 20122586

Se realizará una prueba para con la siguiente matriz y se evaluará si coincide con nuestro resultado final .

$$\begin{pmatrix} 4 & 7 & 8 & 6 & 4 \\ 6 & 7 & 3 & 10 & 2 \\ 3 & 8 & 1 & 10 & 4 \\ 7 & 1 & 7 & 3 & 7 \end{pmatrix} \quad \begin{array}{l} X1 \\ X2 \\ X3 \\ X4 \end{array} = \begin{array}{l} -1124 / 119 \\ -1514 / 119 \\ 810 / 119 \\ 1515 / 119 \end{array} = \begin{array}{l} -9.4453781513 \\ -12.7226990756 \\ 6.8067226891 \\ 12.73109243 \end{array}$$

### Función Escalona

```

77 {
78     for(int i=0;i<m;i++)
79     {
80         int indexMax;
81         maxIndex(A,n,m,i,i,indexMax);
82         pivoteo(A,n,m,indexMax,i,i);
83         for(int j=i+1;j<n;j++)
84         {
85             double num=A[j][i];
86             double denom=A[i][i];
87             for(int k=i;k<m;k++)
88                 A[j][k]=A[j][k]-((A[i][k]/denom)*num);
89         }
90     }
91 }
92 }
93
94 int main()
95 {
96     double **A;
97     int n=5,m=6;
98     crear(A,n,m);
99     llenar(A,n,m);
100     imprimir(A,n,m);
101     cout<<"-----"<<endl;
102     escalona(A,n,m);
103     imprimir(A,n,m);
104 }
105
106 return 0;
107 }

```

```

lehi@lehi-Lenovo-U310: ~/Documentos/UNIVERSIDAD/Analisis-Numerico
Archivo Editar Ver Buscar Terminal Ayuda
lehi@lehi-Lenovo-U310:~/Documentos/UNIVERSIDAD/Analisis-Numerico$ g++ Gauss.cpp
lehi@lehi-Lenovo-U310:~/Documentos/UNIVERSIDAD/Analisis-Numerico$ ./Gauss
4 7 8 6 4 6
7 3 10 2 3 8
1 10 4 7 1 7
3 7 2 9 8 10
3 1 3 4 8 6
-----
4 7 8 6 4 6
0 -9.25 -4 -8.5 -4 -2.5
0 0 -1.56757 -2.08108 1.56757 3.27027
0 0 9.2069 15.069 -4.89655
0 0 0 1.3839 2.85581
lehi@lehi-Lenovo-U310:~/Documentos/UNIVERSIDAD/Analisis-Numerico$

```

```

29
30 void escalona(double **A,int n,int m)
31 {
32     for(int i=0;i<m;i++)
33         for(int j=i+1;j<n;j++)
34         {
35             double num=A[j][i];
36             double denom=A[i][i];
37             for(int k=i;k<m;k++)
38                 A[j][k]=A[j][k]-((A[i][k]/denom)*num );
39         }
40 }
41

```

## Usando Sustitución Regresiva para Resolver el sistema de Ecuaciones

```

112 void metGaussPiv(double **A, int n, int m, double *res)
113 {
114     cout<<endl<<"Matriz escalonada con pivoteo"<<endl;
115     escalonaPiv(A,n,m);
116     imprimir(A,n,m);
117
118     sustRegre(A,n,m,res);
119 }
120
121 int main()
122 {
123     double **A;
124     int n=4,m=5;
125     double *res = new double[m-1];
126     crear(A,n,m);
127     llenar(A,n,m);
128     cout<<"Matriz sin escalonar"<<endl;
129     imprimir(A,n,m);
130
131     metGauss(A,n,m,res);
132
133     /*
134     metGaussPiv(A,n,m,res);
135     */
136     cout<<endl<<"La respuesta del sistema de ecuaciones es : "<<
137     for(int i=0;i<m-1;i++)
138         cout<<res[i]<<" ";
139
140
141     return 0;
142 }

```

```

lehi@lehi-Lenovo-U310: ~/Documentos/UNIVERSIDAD/Analisis-Numerico$ g++ Gauss
s.cpp -o Gauss
lehi@lehi-Lenovo-U310: ~/Documentos/UNIVERSIDAD/Analisis-Numerico$ ./Gauss
Matriz sin escalonar
4 7 8 6 4
6 7 3 10 2
3 8 1 10 4
7 1 7 3 7

Matriz escalonada
4 7 8 6 4
0 -3.5 -9 1 -4
0 0 -12.0714 6.28571 -2.14286
0 0 0 0.704142 8.9645

La respuesta del sistema de ecuaciones es :
-9.44538 -12.7227 6.88672 12.7311
lehi@lehi-Lenovo-U310: ~/Documentos/UNIVERSIDAD/Analisis-Numerico$

```

```

87
88 void sustRegre(double **A, int n, int m, double *&res)
89 {
90     double suma=0;
91     res[n-1]=A[n-1][m-1]/A[n-1][n-1];
92     for(int i=n-2;i>=0;i--)
93     {
94         suma=0;
95         for(int j=i+1;j<n;j++)
96             suma=suma+A[i][j]*res[j];
97         res[i]=(A[i][m-1]-suma)/A[i][i];
98     }
99 }
100

```

```

100
101 void metGauss(double **A, int n, int m, double *res)
102 {
103     cout<<endl<<"Matriz escalonada"<<endl;
104     escalona(A,n,m);
105     imprimir(A,n,m);
106     sustRegre(A,n,m,res);
107 }
108

```

## Función Escalona con Pivoteo

```

58     temp=A[irowMax][i];
59     A[irowMax][i]=A[rowtoChange][i];
60     A[rowtoChange][i]=temp;
61 }
62
63 }
64
65 void escalonaPiv(double **A,int n,int m)
66 {
67     for(int i=0;i<m;i++)
68     {
69         int indexMax;
70         maxIndex(A,n,m,i,i,indexMax);
71         pivoteo(A,n,m,indexMax,i,i);
72         for(int j=i+1;j<n;j++)
73         {
74             double num=A[j][i];
75             double denom=A[i][i];
76             for(int k=i;k<m;k++)
77             {
78                 A[j][k]=A[j][k]-((A[i][k]/denom)*num );
79             }
80             imprimir(A,n,m);
81             cout<<endl<<"-----"<<endl;
82         }
83     }
84
85 void sustRegre(double **A, int n, int m, double *res)
86 {
87     double suma=0;

```

```

s.cpp -o Gauss
lehi@lehi-Lenovo-U310:~/Documentos/UNIVERSIDAD/Analisis-Numerico$ ./Gauss
Matriz sin escalonar
4 7 8 6 4
6 7 3 10 2
3 8 1 10 4
7 1 7 3 7

Matriz escalonada con pivoteo
7 1 7 3 7
0 6.14286 -3 7.42857 -4
0 7.57143 -2 8.71429 1
0 6.42857 4 4.28571 0

-----
7 1 7 3 7
0 7.57143 -2 8.71429 1
0 0 -1.37736 0.358491 -4.81132
0 0 5.69811 -3.11321 -0.849057

-----
7 1 7 3 7
0 7.57143 -2 8.71429 1
0 0 5.69811 -3.11321 -0.849057
0 0 0 -0.39404 -5.01656

-----
7 1 7 3 7
0 7.57143 -2 8.71429 1
0 0 5.69811 -3.11321 -0.849057
0 0 0 -0.39404 -5.01656

lehi@lehi-Lenovo-U310:~/Documentos/UNIVERSIDAD/Analisis-Numerico$

```

```

59 void maxIndex(double **A,int n, int m,int irow ,int icol ,int &indexMax)
60 {
61     indexMax=irow;
62     for(int i=irow;i<n;i++)
63         if(A[i][icol] > A[indexMax][icol])
64             indexMax=i;
65
66 }
67
68 void escalonaPiv(double **A,int n,int m)
69 {
70     for(int i=0;i<m;i++)
71     {
72         int indexMax;
73         maxIndex(A,n,m,i,i,indexMax);
74         pivoteo(A,n,m,indexMax,i,i);
75         for(int j=i+1;j<n;j++)
76         {
77             double num=A[j][i];
78             double denom=A[i][i];
79             for(int k=i;k<m;k++)
80             {
81                 A[j][k]=A[j][k]-((A[i][k]/denom)*num );
82             }
83         }
84     }
85

```



## Sistema de ecuaciones Resuelto con Gauss y Pivoteo

```

110 }
111
112 void metGaussPiv(double **A,int n, int m,double *res)
113 {
114     cout<<endl<<"Matriz escalonada con pivoteo"<<endl;
115     escalonaPiv(A,n,m);
116     imprimir(A,n,m);
117     sustRegre(A,n,m,res);
118 }
119
120 int main()
121 {
122     double **A;
123     int n=4,m=5;
124     double *res = new double[m-1];
125     crear(A,n,m);
126     llenar(A,n,m);
127     cout<<"Matriz sin escalonar"<<endl;
128     imprimir(A,n,m);
129     // metGauss(A,n,m,res);
130     metGaussPiv(A,n,m,res);
131
132     cout<<endl<<"La respuesta del sistema de ecuaciones es : "<<
133     for(int i=0;i<m-1;i++)
134         cout<<res[i]<<" ";
135     cout<<endl;
136
137
138
139     return 0;
140 }

```

```

Matriz escalonada con pivoteo
7 1 7 3 7
0 6.14286 -3 7.42857 -4
0 7.57143 -2 8.71429 1
0 6.42857 4 4.28571 0

-----
7 1 7 3 7
0 7.57143 -2 8.71429 1
0 0 -1.37736 0.358491 -4.81132
0 0 5.69811 -3.11321 -0.849057

-----
7 1 7 3 7
0 7.57143 -2 8.71429 1
0 0 5.69811 -3.11321 -0.849057
0 0 0 -0.39404 -5.01656

-----
7 1 7 3 7
0 7.57143 -2 8.71429 1
0 0 5.69811 -3.11321 -0.849057
0 0 0 -0.39404 -5.01656

-----
7 1 7 3 7
0 7.57143 -2 8.71429 1
0 0 5.69811 -3.11321 -0.849057
0 0 0 -0.39404 -5.01656

-----
La respuesta del sistema de ecuaciones es :
-9.44538 -12.7227 6.88672 12.7311
lehi@lehi-Lenovo-U310:~/Documentos/UNIVERSIDAD/Analisis-Numerico$

```

```

108
109 void metGaussPiv(double **A,int n, int m,double *res)
110 {
111     cout<<endl<<"Matriz escalonada con pivoteo"<<endl;
112     escalonaPiv(A,n,m);
113     imprimir(A,n,m);
114     sustRegre(A,n,m,res);
115 }
116

```

## Conclusión

Los resultados del sistema de ecuaciones dado coinciden con los resultados que obtuvimos del algoritmo implementado en C++ , con la salvedad que la computadora redondea el 5 dígito del decimal para poder mostrarlo en pantalla.

Ya que la computadora no trabaja con todos los decimales al tener una memoria limitada, se genera un error en el cálculo que se va acumulando de operación en operación. El método de Gauss con pivoteo busca amortiguar ese error.