UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA

REPORTE DE ALGORITMOS

GAUSS JORDAN

Nombre	Expediente
Zuñiga Fragoso Diego Joel	317684

Asignatura: Método Numéricos 2023-2

Docente: Vargas Vázquez Damián

I. Antecedentes teóricos

El método de eliminación de Gauss-Jordan es una técnica utilizada en álgebra lineal para resolver sistemas de ecuaciones lineales y encontrar matrices inversas. Lleva el nombre de los matemáticos Carl Friedrich Gauss y Wilhelm Jordan, quienes contribuyeron al desarrollo de la eliminación gaussiana y su aplicación para resolver sistemas de ecuaciones lineales.

1. Eliminación Gaussiana:

La eliminación gaussiana es un método para transformar un sistema de ecuaciones lineales en un sistema equivalente en forma triangular superior (o forma escalonada).

La idea central es utilizar operaciones elementales sobre las ecuaciones (como intercambio de ecuaciones, multiplicación de ecuaciones por escalares y suma/resta de ecuaciones) para simplificar el sistema y llevarlo a una forma más manejable.

2. Matrices y Sistemas de Ecuaciones Lineales:

La representación matricial de un sistema de ecuaciones lineales es fundamental para entender el método de Gauss-Jordan. Un sistema Ax=b se puede expresar como una matriz aumentada [A | b].

La matriz aumentada se somete a operaciones elementales hasta alcanzar una forma escalonada reducida, lo que facilita la resolución del sistema.

3. Teorema Fundamental de los Sistemas de Ecuaciones Lineales:

Este teorema establece que dos sistemas de ecuaciones lineales son equivalentes si y solo si tienen las mismas soluciones. La eliminación de Gauss-Jordan aprovecha esta equivalencia para llevar el sistema original a una forma más simple y fácil de resolver.

II. Algoritmos y sus resultados

Cada algoritmo esta seccionado e incluye descripciones de lo que sucede. Además de contar con capturas de sus resultados

Código

```
#include <conio.h>
#include <iostream>

using namespace std;

void IMPMATF(float* A, int f, int c);

void main()
{
	int B;
	do
	{
		int i, j, k, aux;
		float A[3][4], RES[4], det;

		system("cls");

		cout << "Este programa recibe un sistema de ecuaciones 3X3 y lo 
resuelve paso por paso.";

		for (i = 0; i < 3; i++) // Llenado de ecuaciones
		{
			cout << "\n\nEcuacion " << i + 1 << ":\n";
			cout << "\nIngrese el coeficiente de la 'x':\t"; cin >> 
A[i][0];
			cout << "\nIngrese el coeficiente de la 'y':\t"; cin >> 
A[i][1];
			cout << "\nIngrese el coeficiente de la 'z':\t"; cin >> 
A[i][2];
			cout << "\nIngrese el resultado de la ecuacion:\t"; cin >> 
A[i][3];

			cout << "\nLa ecuacion ingresada fue:\t(" << A[i][0] << 
"x) + (" << A[i][1] << "y) + (" << A[i][2] << "z) = " << A[i][3] << endl;
		}

		system("pause");
		system("cls");

		cout << "La matriz de las ecuaciones es:\n\n\tx\ty\tz\n\n";
		IMPMATF(*A, 3, 4);
		_getch();

		// Calculamos determinante para ver si el sistema tendra 
solucion

		det = A[0][0] * (A[1][1] * A[2][2] - A[1][2] * A[2][1]);
		det -= A[0][1] * (A[1][0] * A[2][2] - A[2][0] * A[1][2]);
		det += A[0][2] * (A[1][0] * A[2][1] - A[2][0] * A[1][1]);

		// Intercambiar filas si en algun valor de la diagonal hay 0 al 
comenzar
		for (i = 0; i < 3; i++)
		{
			if (A[i][i] == 0)
			{
```

```
                        for (j = 0; (A[j][i] == 0) && (j < 3); j++);
                        if (j == 3) // No encontro ningun numero diferente
a 0 en su columna
                                break;
                        else // Intercambiar Filas i y j
                        {
                                for (k = 0; k < 4; k++)
                                {
                                        RES[k] = A[i][k]; // RES = Fila i
                                        A[i][k] = A[j][k]; // Fila i = Fila j
                                        A[j][k] = RES[k]; // Fila j = RES
                                }
                                cout << "\nR" << i + 1 << " <--> R" << j + 1;
                                cout << "\n\tx\ty\tz\n\n";
                                IMPMATF(*A, 3, 4);
                                _getch();
                        }
                }
        }

        for (j = 0; j < 3; j++) // Desepejar las variables operando las
filas
        {
                for (i = 0; i < 3; i++)
                {
                        if ((i != j) && (A[i][j] != 0))
                        {
                                aux = A[i][j];
                                for (k = 0; k < 4; k++)
                                {
                                        A[i][k] = (A[j][j] * A[i][k]) - (aux *
A[j][k]);
                                }
                                cout << "\nR" << i + 1 << " --> (" << A[j][j]
<< "R" << i + 1 << ") - (" << aux << "R" << j + 1 << ")\n";
                                cout << "\n\tx\ty\tz\n\n";
                                IMPMATF(*A, 3, 4);
                                _getch();
                        }
                        else {}
                }
        }

        system("cls");

        if (det == 0)
                cout << "\tEL SISTEMA DE ECUACIONES NO TIENE SOLUCION";
        else
        {
                cout << endl << "R E S U L T A D O S:\n";
                for (i = 0; i < 3; i++)
                {
                        aux = A[i][i];
                        A[i][i] /= aux;
                        A[i][3] /= aux;
                }
```

```
				cout << "\n\tx\ty\tz\n\n";
				IMPMATF(*A, 3, 4);
				cout << "\n\nPor lo tanto:\n\nx = " << A[0][3] << "\ny = " 
<< A[1][3] << "\nz = " << A[2][3];
			}

			cout << "\n\nPresione la tecla '1' para repetir el proceso, de 
lo contrario presione '0' para salir";

			do
			{
				fflush(stdin);
				B = _getch();
			} while ((B != 48) && (B != 49));

		} while (B == 49);
}

void IMPMATF(float* A, int f, int c)
{
	int i, j;
	for (i = 0; i < f; i++)
	{
		cout << "\t";
		for (j = 0; j < c; j++)
			cout << A[i * c + j] << "\t";
		cout << "\t\n";
	}
}
```

Resultado

```
Este programa recibe un sistema de ecuaciones 3X3 y lo resuelve paso por paso.

Ecuacion 1:

Ingrese el coeficiente de la 'x':       12

Ingrese el coeficiente de la 'y':       54

Ingrese el coeficiente de la 'z':       12

Ingrese el resultado de la ecuacion:    5

La ecuacion ingresada fue:      (12x) + (54y) + (12z) = 5


Ecuacion 2:

Ingrese el coeficiente de la 'x':       8

Ingrese el coeficiente de la 'y':       45

Ingrese el coeficiente de la 'z':       65

Ingrese el resultado de la ecuacion:    21

La ecuacion ingresada fue:      (8x) + (45y) + (65z) = 21


Ecuacion 3:

Ingrese el coeficiente de la 'x':       32

Ingrese el coeficiente de la 'y':       2

Ingrese el coeficiente de la 'z':       5

Ingrese el resultado de la ecuacion:    4

La ecuacion ingresada fue:      (32x) + (2y) + (5z) = 4
Presione una tecla para continuar . . .
```

```
La matriz de las ecuaciones es:

        x       y       z

        12      54      12      5
        8       45      65      21
        32      2       5       4

R2 --> (12R2) - (8R1)

        x       y       z

        12      54      12      5
        0       108     684     212
        32      2       5       4

R3 --> (12R3) - (32R1)

        x       y       z

        12      54      12      5
        0       108     684     212
        0       -1704   -324    -112

R1 --> (108R1) - (54R2)

        x       y       z

        1296    0       -35640  -10908
        0       108     684     212
        0       -1704   -324    -112

R3 --> (108R3) - (-1704R2)

        x       y       z

        1296    0       -35640  -10908
        0       108     684     212
        0       0       1.13054e+06     349152
```

```
R E S U L T A D O S:

        x       y       z

        1       0       0       0.0763067
        0       1       0       0.00700548
        0       0       1       0.308835


Por lo tanto:

x = 0.0763067
y = 0.00700548
z = 0.308835

Presione la tecla '1' para repetir el proceso, de lo contrario presione '0' para salir
```

III. Conclusiones

En conclusión, el método de Gauss-Jordan es una poderosa herramienta en álgebra lineal que se basa en la eliminación gaussiana y utiliza operaciones elementales sobre matrices para llevar un sistema de ecuaciones lineales a una forma escalonada reducida por filas. Este proceso simplifica significativamente la resolución de sistemas de ecuaciones y la obtención de matrices inversas, aprovechando la equivalencia entre sistemas de ecuaciones lineales y sus soluciones