



Universidad Nacional  
Autónoma de México



**Fes Acatlán**

***Métodos Numéricos***

*Profesora: Carrillo Ramírez Teresa*

*Semestre 2025-1*

**Programa 2**

Mariana Alejandra López Ramírez

Andrea Estephanie Fierro Ibáñez

Jessica Esmeralda Alcantar  
Hernández

## Programa de Métodos Numéricos: El Determinante

El objetivo principal del programa es resolver un sistema de ecuaciones lineales representado por una matriz de coeficientes y un vector independiente. El programa permite al usuario ingresar los valores de la matriz y vector, verificar si los datos ingresados son correctos y realizar correcciones en caso necesario. Una vez que los datos son correctos, el programa verifica si la matriz es diagonalmente dominante. Si lo es, el programa procede con el cálculo del determinante y, si es no nulo, resolver el sistema.

Para la implementación del programa se tuvimos que considerar los conceptos principales para poder dar un eficiente código.

- **Sistemas de Ecuaciones Lineales:** Consisten en un conjunto de ecuaciones lineales donde se busca resolver el valor de múltiples variables.
- **Determinante de una Matriz:** Valor único que puede indicar si una matriz es invertible o no. Un determinante igual a cero sugiere que la matriz no tiene una inversa y que el sistema lineal asociado no tiene solución única.
- **Dominancia Diagonal:** Una matriz es diagonalmente dominante (DD) si, en cada fila, el valor absoluto del elemento diagonal es mayor o igual a la suma de los valores absolutos de los otros elementos en la misma fila. Esta propiedad facilita la convergencia en métodos numéricos de solución de sistemas lineales.

Ahora bien para la estructura del programa este contiene seis funciones además de la función main. Cada función tiene un papel específico en el ingreso de datos, verificación de propiedades de la matriz y cálculo de soluciones.

```
void leermatriz(int n, double matriz[n][n], double vector[n]) {
```

Esta función permite al usuario ingresar los valores de los elementos de la matriz y del vector independiente. La función solicita al usuario ingresar los coeficientes de la matriz de forma individual, indicando la posición de cada elemento.

También recoge el vector independiente, donde cada entrada representa el término independiente de la ecuación correspondiente y se verifica que cada entrada sea numérica, y si no lo es, el programa finaliza con un mensaje de "Entrada inválida".

```
void mostrarmatriz(int n, double matriz[n][n], double vector[n]) {
```

Imprimirá la matriz y el vector independiente para el usuario, permitiendo verificar los datos ingresados. La matriz se muestra con tres decimales para facilitar la lectura y análisis de los datos. También se imprime el vector independiente, al final de cada fila, separando claramente los coeficientes de la matriz del vector.

```
void corregircoeficiente(int n, double matriz[n][n], double vector[n]) {
```

Permitirá al usuario corregir cualquier entrada de la matriz o del vector independiente sin necesidad de volver a ingresar todos los datos. La función pregunta al usuario por la posición del elemento a corregir.

Si el usuario ingresa "0" como columna, se modifica el valor correspondiente en el vector independiente; de lo contrario, se corrige el valor en la matriz en la fila y columna especificadas.

```
int DD(int n, double matriz[n][n]) {
```

En esta parte determina si la matriz es diagonalmente dominante.

Así recorrerá cada fila de la matriz, sumando los valores absolutos de los elementos no diagonales. Si el valor absoluto del elemento diagonal de una fila es menor que esta suma, la función retorna 0, indicando que la matriz no es diagonalmente dominante.

Si la matriz cumple esta condición para todas las filas, la función retorna 1, indicando que es diagonalmente dominante.

```
double determinante(int n, double matriz[n][n]){
```

Calcula el determinante de la matriz usando el método de **triangulación** o eliminación gaussiana. De igual manera crea una copia de la matriz original para evitar modificar los datos originales.

Se usará la eliminación gaussiana para transformar la matriz en una forma triangular superior. A medida que se realizan operaciones en las filas, se acumula el producto de los elementos diagonales, que representa el determinante.

Si en cualquier paso el pivote es cero, el determinante se establece en cero y la función termina.

```
void resolver(int n, double matriz[n][n], double vector[n]) {
```

Resuelve el sistema de ecuaciones lineales mediante sustitución hacia atrás, una vez que la matriz está en forma triangular superior.

Almacena la solución en un arreglo soluciones que se construye de la última variable hacia la primera. El valor de cada incógnita se determina resolviendo las ecuaciones de forma sucesiva, restando los valores ya conocidos y se imprime cada valor de  $x_i$ , mostrando la solución completa del sistema al usuario

```
int main(){
```

Se hará la ejecución de todas las funciones anteriores, gestionando el flujo de interacción con el usuario.

Para esto se...

- Solicita la dimensión de la matriz al usuario.
- Llama a la función leermatriz para permitir la entrada inicial de datos.
- Da al usuario la opción de verificar y corregir datos con corregircoeficiente.
- Usa la función DD para verificar la dominancia diagonal de la matriz.
- Si la matriz es diagonalmente dominante, el programa puede proceder a calcular el determinante y resolver el sistema si es distinto de cero.
- Muestra un mensaje indicando si el determinante es cero y si el sistema tiene o no solución.

En conclusión este programa proporciona un método interactivo para ingresar, verificar y corregir datos de una matriz de coeficientes y vector independiente de un sistema de ecuaciones lineales. Implementa métodos básicos de álgebra lineal, como el cálculo de determinantes y la sustitución hacia atrás, para resolver el sistema en caso de que el determinante sea diferente de cero. La verificación de la dominancia diagonal agrega al programa, asegurando mayor estabilidad y precisión en los cálculos.