



TAREA 13. PSEUDOCÓDIGO PARA RESOLVER MATRICES MEDIANTE EL MÉTODO DE GAUSS-SIEDEL.

Adolfo Hernández Ramírez (427560)

Correo: a.hernandezramirez3@ugto.mx.

Licenciatura Ingeniera Química Sustentable. Universidad de Guanajuato. División de Ciencias e Ingenierías. Campus León. Loma del Bosque 103, Lomas del Campestre. León, Gto, México.

Pseudocódigo.

- 0- INICIO.
- 1- Leer n, max-iter, tolerancia.
- 2- Declarar matrices y vectores.

```
DECLARAR A[n][n]
DECLARAR b[n]
DECLARAR x[n]
DECLARAR x_anterior[n]
DECLARAR error.
```

- 3- Leer la matriz A.

```
PARA i = 0 HASTA n-1 HACER
  PARA j = 0 HASTA n-1 HACER
    LEER A[i][j]
  FIN PARA
FIN PARA
```

- 4- Leer vector b.

```
PARA i = 0 HASTA n-1 HACER
  LEER b[i]
FIN PARA
```

- 5- Inicializar vector solución con ceros.

```
PARA i = 0 HASTA n-1 HACER
  x[i] = 0.0
  x_ant[i] = 0.0
FIN PARA
```

- 6- Verificar criterio de convergencia con el método de la matriz diagonalmente dominante.

```
PARA i = 0 HASTA n-1 HACER
  suma = 0.0
  PARA j = 0 HASTA n-1 HACER
    SI i != j ENTONCES
      suma = suma + ABS(A[i][j])
    FIN SI
  FIN PARA
  SI ABS(A[i][i]) <= suma ENTONCES
    es_diagonal_dominante = FALSO
    IMPRIMIR "Advertencia: La matriz no es diagonalmente dominante"
  SALIR DEL PARA
```

```
FIN SI
FIN PARA
```

7- Realizar el método de Gauss-Siedel.

```
iter = 0
error = tolerancia + 1 // para entrar al bucle

MIENTRAS iter < max_iter Y error > tolerancia HACER
```

7.1- Copiar solución actual a anterior.

```
PARA i = 0 HASTA n-1 HACER
    x_ant[i] = x[i]
FIN PARA
```

7.2- Actualizar cada componente de x

```
PARA i = 0 HASTA n-1 HACER
    suma = 0.0
```

7.3- Suma de $A[i][j] * x[j]$ para $j < i$.

```
PARA j = 0 HASTA i-1 HACER
    suma = suma + A[i][j] * x[j]
FIN PARA
```

7.4- Suma de $A[i][j] * x_{ant}[j]$ para $j > i$.

```
PARA j = i+1 HASTA n-1 HACER
    suma = suma + A[i][j] * x_ant[j]
FIN PARA
```

7.5- Calcular nuevo $x[i]$

```
x[i] = (b[i] - suma) / A[i][i]
FIN PARA
```

7.6- Calcular error.

```
error = 0.0
PARA i = 0 HASTA n-1 HACER
    error_actual = ABS(x[i] - x_ant[i])
    SI error_actual > error ENTONCES
        error = error_actual
FIN SI
FIN PARA
```

7.7- Mostrar información de la iteración.

```
IMPRIMIR "Iteración:", iter, "Error:", error
IMPRIMIR "Solución actual:"
PARA i = 0 HASTA n-1 HACER
    IMPRIMIR "x[" , i, "] =", x[i]
FIN PARA
```

```
iter = iter + 1
FIN MIENTRAS
```

8- Mostrar resultados finales.

```
SI error <= tolerancia ENTONCES
    IMPRIMIR "CONVERGENCIA ALCANZADA"
    IMPRIMIR "Número de iteraciones:", iter
    IMPRIMIR "Error final:", error
SINO
    IMPRIMIR "CONVERGENCIA NO ALCANZADA"
    IMPRIMIR "Máximo de iteraciones alcanzado"
FIN SI
```

```
IMPRIMIR "Solución final:"
PARA i = 0 HASTA n-1 HACER
    IMPRIMIR "x[" , i, "] =", x[i]
FIN PARA
```

9- Verificar solución.

```
IMPRIMIR "Verificación (A*x - b):"
PARA i = 0 HASTA n-1 HACER
    residuo = 0.0
    PARA j = 0 HASTA n-1 HACER
        residuo = residuo + A[i][j] * x[j]
    FIN PARA
    residuo = residuo - b[i]
    IMPRIMIR "Residuo[" , i, "] =", residuo
FIN PARA
```

10- FIN.