Física Computacional

Escuela de Física

M.R.Fulla[1]

[1]Escuela de Física, Universidad Nacional de Colombia Sede Medellín

marlonfulla@yahoo.com- Oficina:21-408

https://sites.google.com/view/fiscomunalmed/

October 16, 2023

Método de Gauss-Jordan

Wilhelm Jordan (1842-1899)

Variación del método de la eliminación gaussiana

$$\begin{aligned} 5x_1 - x_2 - 2x_3 &= 11 \\ -x_1 + 5x_2 - 2x_3 &= 0 \\ -2x_1 - 2x_2 + 7x_3 &= 0 \end{aligned}$$

$$lin = \begin{bmatrix} 5 & -1 & -2 & 11 \\ -1 & 5 & -2 & 0 \\ -2 & -2 & 7 & 0 \end{bmatrix}$$

Se busca el pivote (entrada con el valor más grande en valor absoluto) y se normaliza la fila con ese valor.

UNIVERSIDAD NACIONAL DE COLOMBIA

$$lin = \begin{bmatrix} 5 & -1 & -2 & 11 \\ -1 & 5 & -2 & 0 \\ -2 & -2 & 7 & 0 \end{bmatrix}$$

normalizando a 5

$$lin = \begin{bmatrix} 1.0 & -0.2 & -0.4 & 2.2 \\ -1 & 5 & -2 & 0 \\ -2 & -2 & 7 & 0 \end{bmatrix}$$

UNIVERSIDAD NACIONAL DE COLOMBIA

$$lin = \begin{bmatrix} 1.0 & -0.2 & -0.4 & 2.2 \\ -1 & 5 & -2 & 0 \\ -2 & -2 & 7 & 0 \end{bmatrix}$$

$$fila_2 = fila_2 + 1 * fila_1$$
$$fila_3 = fila_3 + 2 * fila_1$$

$$lin = \begin{bmatrix} 1.0 & -0.2 & -0.4 & 2.2 \\ 0 & 4.8 & -2.4 & 2.2 \\ 0 & -2.4 & 6.2 & 4.2 \end{bmatrix}$$

buscamos el pivote en la segunda columna y normalizamos su fila a ese valor.

UNIVERSIDAD NACIONAL DE COLOMBIA

$$lin = \begin{bmatrix} 1.0 & -0.2 & -0.4 & 2.2 \\ 0 & 4.8 & -2.4 & 2.2 \\ 0 & -2.4 & 6.2 & 4.2 \end{bmatrix}$$

normalizando

$$lin = \begin{bmatrix} 1.0 & -0.2 & -0.4 & 2.2 \\ 0 & 1.0 & -0.5 & 0.4583 \\ 0 & -2.4 & 6.2 & 4.2 \end{bmatrix}$$

$$fila_1 = fila_1 + 0.2 * fila_2$$
$$fila_3 = fila_3 + 2.4 * fila_2$$

UNIVERSIDAD NACIONAL DE COLOMBIA

$$lin = \begin{bmatrix} 1.0 & 0.0 & -0.5 & 2.292 \\ 0 & 1.0 & -0.5 & 0.4583 \\ 0 & 0 & 5.0 & 5.5 \end{bmatrix}$$

buscamos pivote en la tercera columna y normalizamos su fila a ese valor.

$$lin = \begin{bmatrix} 1.0 & 0.0 & -0.5 & 2.292 \\ 0 & 1.0 & -0.5 & 0.4583 \\ 0 & 0 & 1.0 & 1.1 \end{bmatrix}$$

$$fila_1 = fila_1 + 0.5 * fila_3$$
$$fila_2 = fila_2 + 0.5 * fila_3$$

UNIVERSIDAD NACIONAL DE COLOMBIA

$$lin = \begin{bmatrix} 1.0 & 0.0 & 0 & 2.842 \\ 0 & 1.0 & 0 & 1.008 \\ 0 & 0 & 1.0 & 1.1 \end{bmatrix}$$

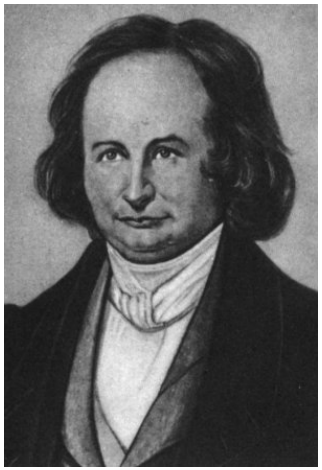
Finalmente:

$$\begin{aligned} x_1 &= 2.842 \\ x_2 &= 1.008 \\ x_3 &= 1.100 \end{aligned}$$

UNIVERSIDAD NACIONAL DE COLOMBIA

1. Introducir coeficientes matriciales y vector constante.
2. Variando a i desde 1 hasta n, hacer lo siguiente:
   - 2.1 Encontrar la entrada $lin(k,i),\ k = i, i+1, \ldots n$ que tenga el máximo valor absoluto y definirlo como pivote.
   - 2.2 Si el pivote es cero, entonces desplegar un mensaje que informe que el sistema es singular y terminar el programa. En otro caso, continuar.
   - 2.3 Intercambiar la fila i y k.
   - 2.4 Normalizar la fila i dividiendo cada entrada por $lin(i,i)$.
   - 2.5 Variando j desde 1 hasta n realizar lo siguiente: si $j \neq i$, adicionar $-lin(j,i)$ veces la fila i-ésima a la fila j-ésima de lin para eliminar $x(i)$ de la j-ésima ecuación.
3. Variando i desde 1 hasta n, realice lo siguiente: haga $x(i) = lin(i, n+1)$

Método de Jacobi



Carl Gustav Jacob Jacobi (1804-1851)

Método de punto fijo:

$$f(x) = 0 \;\rightarrow\; x = g(x)$$

$$x_{n+1} = g(x_n)$$

(Fórmula de iteración)

Consideremos

$$\begin{aligned} 10x_1 + x_2 + 2x_3 &= 13 \\ x_1 + 10x_2 + x_3 &= 12 \\ x_1 + x_2 + 10x_3 &= 12 \end{aligned}$$

Tiene solución exacta: $x_1 = x_2 = x_3 = 1$

Método de Jacobi

UNIVERSIDAD NACIONAL DE COLOMBIA

despejando la variable x_i de la ecuación i-ésima:

$$x_1 = (13 - x_2 - 2x_3)/10$$
$$x_2 = (12 - x_1 - x_3)/10$$
$$x_3 = (12 - x_1 - x_2)/10$$

Usando una iteración tipo "punto fijo", comenzamos con una solución inicial aproximada:

$$x_1 = 1.3,\ x_2 = 1.2,\ x_3 = 1.2$$

para obtener la siguiente solución aproximada:

$$x_1^* = (13 - x_2 - 2x_3)/10 = (13 - 1.2 - 2.4)/10 = 0.94$$
$$x_2^* = (12 - x_1 - x_3)/10 = (12 - 1.3 - 1.2)/10 = 0.95$$
$$x_3^* = (12 - x_1 - x_2)/10 = (12 - 1.3 - 1.2)/10 = 0.95$$

Método de Jacobi

Esta solución puede ser ustada para obtener otra aproximación reemplazando $x_i \rightarrow x_i^*$:

$$x_1^* = (13 - x_2 - 2x_3)/10 = (13 - 0.95 - 1.9)/10 = 1.015$$
$$x_2^* = (12 - x_1 - x_3)/10 = (12 - 0.94 - 0.95)/10 = 1.011$$
$$x_3^* = (12 - x_1 - x_2)/10 = (12 - 0.94 - 0.95)/10 = 1.011$$

En general, las fórmulas iterativas de Jacobi para un sistema lineal Ax=b son:

$$x_i^* = (b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^{n} A_{ij}x_j)/A_{ii} \quad i = 1, 2, ..., n$$

NOTA: se debe cumplir que todos los términos de las diagonales $A_{ii} \neq 0$, si esto no se cumple, se deben reorganizar las ecuaciones.

Método de Jacobi

UNIVERSIDAD NACIONAL DE COLOMBIA

El programa debe finalizar cuando dos soluciones aproximadas sucesivas *xold* y *xnew* difieran por una cantidad *epsil* (tolerancia):

$$|xold(i) - xnew(i)| < epsil, \quad \text{para toda } i$$

La convergencia a una única solución se garantiza para cualquier aproximación inicial, si la matriz A es **diagonalmente dominante**, lo cual puede significar:

Método de Jacobi

UNIVERSIDAD NACIONAL DE COLOMBIA

1. Las entradas de la diagonal dominan las filas:
   para cada $i = 1, \ldots, n \quad |A_{ii}| > \sum_{j=1}^{i-1} |A_{ij}| + \sum_{j=i+1}^{n} |A_{ij}|$
2. Las entradas de la diagonal dominan las columnas:
   para cada $i = 1, \ldots, n \quad |A_{ii}| > \sum_{j=1}^{i-1} |A_{ji}| + \sum_{j=i+1}^{n} |A_{ji}|$

$$
\begin{aligned}
10x_1 + x_2 + 2x_3 &= 13 \\
x_1 + 10x_2 + x_3 &= 12 \\
x_1 + x_2 + 10x_3 &= 12
\end{aligned}
$$

Método de Jacobi

UNIVERSIDAD NACIONAL DE COLOMBIA

```
1   ! Programa para encontrar una solución aproximada de un sistema
2   ! lineal Ax=B usando el método de Jacobi. El proceso iterativo
3   ! se realiza hasta que se alcance un número máximo de iteraciones
4   ! o si dos aproximaciones sucecivas difieren en cada componente
5   ! a lo sumpo por un valor epsil.
6   ! Entrada: numeqn,a,b,numits,epsil,xold
7   ! Salida:  una secuencia de soluciones aproximadas o un mensaje
8   !          indicando que el límite de iteraciones fue alcanzado
9   PROGRAM jacobi
10  IMPLICIT NONE
11  !lim    : número máximo de ecuaciones
12  !numeqn : número de ecuaciones (número de incógnitas)
13  !a      : matriz con coeficientes
14  !b      : vector con constantes
15  !numits : límite del número de iteraciones
16  !epsil  : tolerancia
17  !done   : bandera (lógica) que indica la terminación satisfactoria
18  !xold   : aproximación anterior
19  !xnew   : aproximación nueva
20  !suma   : sumatoria de la fórmula
21  !n      : número de iteraciones
22  !i,j    : subíndices
23  INTEGER:: numeqn,numits,n,i,j
24  INTEGER, PARAMETER:: lim=10
25  REAL::a(lim,lim),b(lim),xold(lim),xnew(lim)
26  REAL::epsil,suma
27  LOGICAL::done
28
29  WRITE(*,*) "Escriba el numero de ecuaciones"
30  READ(*,*)   numeqn
```

Método de Jacobi

UNIVERSIDAD NACIONAL DE COLOMBIA

```
31
32    WRITE(*,*) "Escriba los coeficientes matriciales (fila a fila)"
33    DO i=1,numeqn
34        READ(*,*) (a(i,j),j=1,numeqn)
35    ENDDO
36
37    WRITE(*,*) "Escriba el vector constante"
38    READ(*,*) (b(i),i=1,numeqn)
39
40    WRITE(*,*) "Escriba la tolerancia y el maximo de iteraciones"
41    READ(*,*) epsil,numits
42
43    WRITE(*,*) "Introduzca una solucion inicial"
44    READ(*,*) (xold(i),i=1,numeqn)
45
46    done=.FALSE.
47    n=0
48
49    !mientras no sea satisfactoria la convergencia
50    DO WHILE(.NOT.done)
51        n=n+1
52
53        DO i=1,numeqn
54            suma=b(i)
55
56            DO j=1,i-1
57                suma=suma-a(i,j)*xold(j)
58            ENDDO
59            DO j=j+1,numeqn
60                suma=suma-a(i,j)*xold(j)
61            ENDDO
62            xnew(i)=suma/a(i,i)
63        ENDDO
```

Método de Jacobi

UNIVERSIDAD NACIONAL DE COLOMBIA

```
64
65    !chequea si la condición de convergencia se satisface
66    i=1
67    done=ABS(xold(i)-xnew(i))<epsil
68    !mientras done=false e i<numeqn, realice:
69    DO WHILE(done .AND. i<numeqn)
70        i=i+1
71        done=ABS(xold(i)-xnew(i))<epsil
72    ENDDO
73    done=done .OR. (n>numits)
74
75    !despliega en la consola una solución aproximada
76    WRITE(*,*)
77    WRITE(*,*) "ITERACION #:",n
78    DO i=1,numeqn
79        WRITE(*,1) "X(",i,")=",xnew(i)
80    1   FORMAT(1X,A2,I3,A6,F8.4)
81    ENDDO
82
83    !copia xnew en xold
84    DO i=1,numeqn
85        xold(i)=xnew(i)
86    ENDDO
87
88    ENDDO
89
90    END PROGRAM
```

Método de Jacobi-Test

UNIVERSIDAD NACIONAL DE COLOMBIA

```
"C:\Users\GMCSMC\source\re
 Escriba el numero de ecuaciones
3
 Escriba los coeficientes matriciales (fila a fila)
10 1 1
1 10 1
1 1 10
 Escriba el vector constante
15 24 33
 Escriba la tolerancia y el maximo de iteraciones
1e-5 20
 Introduzca una solucion inicial
0 0 0
```

```
 ITERACION #:           1
 X(  1    )=  1.5000
 X(  2    )=  2.4000
 X(  3    )=  3.3000

 ITERACION #:           2
 X(  1    )=  0.9300
 X(  2    )=  1.9200
 X(  3    )=  2.9100

 ITERACION #:           3
 X(  1    )=  1.0170
 X(  2    )=  2.0160
 X(  3    )=  3.0150
```

```
ITERACION #:           4
X(  1    )=  0.9969
X(  2    )=  1.9968
X(  3    )=  2.9967

ITERACION #:           5
X(  1    )=  1.0007
X(  2    )=  2.0006
X(  3    )=  3.0006

ITERACION #:           6
X(  1    )=  0.9999
X(  2    )=  1.9999
X(  3    )=  2.9999

ITERACION #:           7
X(  1    )=  1.0000
X(  2    )=  2.0000
X(  3    )=  3.0000

ITERACION #:           8
X(  1    )=  1.0000
X(  2    )=  2.0000
X(  3    )=  3.0000

ITERACION #:           9
X(  1    )=  1.0000
X(  2    )=  2.0000
X(  3    )=  3.0000
Press any key to continue . . .
```

Método de Gauss-Seidel

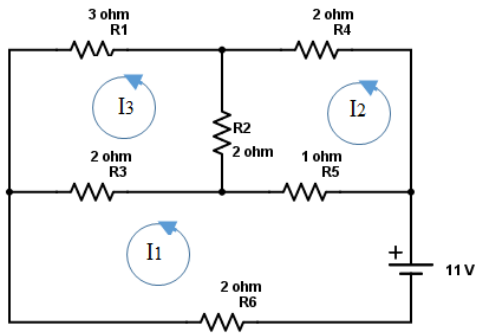
UNIVERSIDAD NACIONAL DE COLOMBIA

La implementación de soluciones nuevas x_i^* en la fórmula iterativa de Jacobi proporciona mayor convergencia.

$$x_i^* = (b_i - \sum_{j=1}^{i-1} A_{ij} x_j^* - \sum_{j=i+1}^{n} A_{ij} x_j)/A_{ii} \quad i = 1, 2, \ldots, n$$

Actividad

1. Realizar una modificación del programa de eliminación Gaussiana y adaptarlo al algoritmo de Gauss-Jordan.
2. Realizar una adaptación del programa de Jacobi y adaptarlo al algoritmo de Gauss-Seidel. Evaluar la capacidad de convergencia de ambos programas con el ejemplo suministrado y concluir.
3. Encontrar las corrientes de malla en el siguiente circuito mediante las técnicas estudiadas y concluya acerca de su convergencia:



$$\begin{aligned} 5I_1 - I_2 - 2I_3 &= 11 \\ -I_1 + 5I_2 - 2I_3 &= 0 \\ -2I_1 - 2I_2 + 7I_3 &= 0 \end{aligned}$$