

# Práctica 2

## Hoja de Actividades

Curso 2013–2014

**Actividad 1.** Determina si la matriz de coeficientes de los siguientes sistemas de ecuaciones lineales, es (o no es) estrictamente diagonalmente dominante.

$$a) \left. \begin{array}{rcl} 10x + y + 2z & = & 3 \\ 4x - 6y - z & = & 9 \\ -2x + 3y + 8z & = & 51 \end{array} \right\} \quad b) \left. \begin{array}{rcl} 2x + y + t & = & 1 \\ x + y + z + 2t & = & 1 \\ 2x + y + 3z + t & = & 1 \\ x + 2y + z - t & = & 2 \end{array} \right\}$$

*Solución*

Las matrices de coeficientes de estos sistemas son

$$A = \begin{bmatrix} 10 & 1 & 2 \\ 4 & -6 & -1 \\ -2 & 3 & 8 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 1 & 1 & 2 \\ 2 & 1 & 3 & 1 \\ 1 & 2 & 1 & -1 \end{bmatrix}$$

Así pues,  $A$  es estrictamente diagonal dominante ( $10 > 1 + 2$ ,  $6 > 4 + 1$  y  $8 > 2 + 3$ ) y  $B$  no lo es.

**Actividad 2.** a) Calcula de forma directa las soluciones de los sistemas de ecuaciones de la Actividad 1.

b) Aplica los métodos de Jacobi y Gauss-Seidel a los anteriores sistemas, haciendo sólo 6 iteraciones y tomando el vector nulo como vector inicial. ¿Son convergentes?

*Solución*

a) Introducimos las matrices  $A$  y  $B$ , los términos independientes y resolvemos con la instrucción rref

```
-->A=[10 1 2;4 -6 -1;-2 3 8]
```

```
A =
```

```
10.    1.    2.
 4.   - 6.   - 1.
- 2.    3.    8.
```

```
-->B=[2 1 0 1;1 1 1 2;2 1 3 1;1 2 1 -1]
```

```
B =
```

```
2.    1.    0.    1.
1.    1.    1.    2.
2.    1.    3.    1.
1.    2.    1.   - 1.
```

```
-->b1=[3;9;51]
```

```
b1 =
```

```
3.
9.
51.
```

```
-->R1=rref([A b1])
R1 =

    1.    0.    0. - 0.85
    0.    1.    0. - 3.3
    0.    0.    1.    7.4
```

La solución de este sistema es el vector  $(-0,85, -3,3, 7,4)$ .

```
-->b2=[1;1;1;2]
b2 =

    1.
    1.
    1.
    2.
```

```
-->rref([B b2])
ans =

    1.    0.    0.    0.    0.
    0.    1.    0.    0.    1.
    0.    0.    1.    0.    0.
    0.    0.    0.    1.    0.
```

La solución de este sistema es el vector  $(0, 1, 0, 0)$ .

b) Apliquemos Jacobi al primer sistema

```
-->D=diag([diag(A)])
D =

    10.    0.    0.
    0.   - 6.    0.
    0.    0.    8.
```

```
-->U=triu(A)-D
U =

    0.    1.    2.
    0.    0.   - 1.
    0.    0.    0.
```

```
-->L=tril(A)-D
L =

    0.    0.    0.
    4.    0.    0.
   - 2.    3.    0.
```

```
-->x=[0;0;0];
-->for i=1:6
-->x=inv(D)*(b1-(L+U)*x)
-->end
```

```

x =

    0.3
   - 1.5
    6.375

x =

   - 0.825
   - 2.3625
    7.0125
x =
   - 0.86625
   - 3.21875
    7.0546875
x =
   - 0.7890625
   - 3.2532812
    7.3654687
x =
   - 0.8477656
   - 3.2536198
    7.3977148
x =
   - 0.8541810
   - 3.2981296
    7.383166

```

A continuación aplicamos Gauss-Seidel al primer sistema. Primero ejecutamos el fichero *SustitucionProgresiva.sci* y después calculamos las 6 iteraciones con un bucle

```

-->x=[0;0;0]
-->for i=1:6 x=SustitucionProgresiva(L+D,b1-U*x)
-->end
x =

    0.3
   - 1.3
    6.9375

x =

   - 0.9575
   - 3.2945833
    7.3710938
x =
   - 0.8447604
   - 3.2916892
    7.3981934

```

```

x =

- 0.8504697
- 3.3000121
  7.3998871
x =

- 0.8499762
- 3.2999653
  7.3999929
x =

- 0.8500021
- 3.3000002
  7.3999996

```

Ambos métodos convergen porque  $A$  es estrictamente diagonal dominante. Observa que la convergencia es bastante rápida (el vector  $(-0,85, -3,3, 7,4)$  es la solución tal y como hemos visto en el apartado a)).

Aplicamos Jacobi al segundo sistema:

```

-->D=diag([diag(B)])
D =

    2.    0.    0.    0.
    0.    1.    0.    0.
    0.    0.    3.    0.
    0.    0.    0.   -1.

-->U=triu(B)-D
U =

    0.    1.    0.    1.
    0.    0.    1.    2.
    0.    0.    0.    1.
    0.    0.    0.    0.

-->L=tril(B)-D
L =

    0.    0.    0.    0.
    1.    0.    0.    0.
    2.    1.    0.    0.
    1.    2.    1.    0.

-->x=[0;0;0;0];
-->for i=1:6 x=inv(D)*(b2-(L+U)*x)
-->end
x =

    0.5
    1.
    0.3333333
    -2.

```

```
x =
    1.
    4.1666667
    0.3333333
    0.8333333
```

```
x =
    - 2.
    - 2.
    - 2.
    7.6666667
```

```
x =
    - 2.3333333
    - 10.333333
    - 0.2222222
    - 10.
```

```
x =
    10.666667
    23.555556
    8.6666667
    - 25.222222
```

```
x =
    1.3333333
    32.111111
    - 6.2222222
    64.444444
```

Aplicamos ahora Gauss-Seidel

```
-->x=[0;0;0;0];
-->for i=1:6    x=SustitucionProgresiva(L+D,b2-U*x)
-->end
```

```
x =
    0.5
    0.5
    - 0.1666667
    - 0.6666667
```

```
x =
    0.5833333
    1.9166667
    - 0.4722222
    1.9444444
```

```
x =
    - 1.4305556
    - 0.9861111
    0.9675926
    - 4.4351852
```

```

x =
    3.2106481
    5.6921296
- 2.2260802
    10.368827
x =
- 7.5304784
- 9.9810957
    5.2244084
- 24.268261
x =
    17.624678
    26.687436
- 12.222844
    56.77670

```

A la vista de los resultados obtenidos podemos concluir que ninguno de los dos métodos converge.

### Actividad 3. Sea el sistema de ecuaciones

$$\begin{aligned}
 0,2x + 2,2y + 4,5z &= 0,7 \\
 1,3x + 3,7y + 2,1z &= 1,2 \\
 4,2x + 3,1y + 0,4z &= 5,2
 \end{aligned}$$

- (a) Tomando el vector nulo como aproximación inicial, obtén 20 aproximaciones aplicando el método de Jacobi ¿Es convergente dicho método?
- (b) Reordena las ecuaciones de este sistema para que su matriz asociada sea estrictamente diagonal dominante. Comprueba que en ese caso el método de Jacobi converge y calcula la aproximación obtenida usando 20 iteraciones.

#### Solución

a) Introducimos la matriz del sistema y la columna de términos independientes y aplicamos Jacobi

```

-->A=[0.2 2.2 4.5;1.3 3.7 2.1;4.2 3.1 0.4]
A =

```

```

    0.2    2.2    4.5
    1.3    3.7    2.1
    4.2    3.1    0.4

```

```

-->b=[0.7;1.2;5.2]
b =

```

```

    0.7
    1.2
    5.2

```

```

-->D=diag([diag(A)])
D =

    0.2    0.    0.
    0.    3.7    0.
    0.    0.    0.4

-->L=tril(A)-D
L =

    0.    0.    0.
    1.3    0.    0.
    4.2    3.1    0.

-->U=triu(A)-D
U =

    0.    2.2    4.5
    0.    0.    2.1
    0.    0.    0.

-->x=[0;0;0];
-->for i=1:20    x=inv(D)*(b-(L+U)*x)
-->end

x =

    3.5
    0.3243243
    13.

x =

- 292.56757
- 8.2837838
- 26.263514

.
.
.

x =

1.0D+19 *

    9.2466361
    0.7855516
    17.338312

```

```

x =

    1.0D+20 *

    - 39.875308
    - 1.3089481
    - 10.31777
x =

    1.0D+21 *

    24.654826
    1.9866275
    42.883508
x =

    1.0D+22 *

    - 98.673183
    - 3.3001795
    - 27.427204

```

A la vista de los resultados podemos concluir que el método no converge. Observar que la matriz  $A$  no es diagonal dominante.

b) En este caso es posible reordenar las ecuaciones para obtener una matriz estrictamente diagonal dominante (basta intercambiar las ecuaciones 1 y 3). Así pues, el método de Jacobi convergerá. Veamos cual sería la aproximación en la iteración 20.

Introducimos en Scilab la reordenación

```

-->A([1,3],:)=A([3,1],:)
A =

    4.2    3.1    0.4
    1.3    3.7    2.1
    0.2    2.2    4.5

-->b([1,3],:)=b([3,1],:)
b =

    5.2
    1.2
    0.7

```

Ahora para estas matrices calculamos  $D$ ,  $L$  y  $U$  y aplicamos Jacobi

```

-->D=diag([diag(A)])
D =

    4.2    0.    0.
    0.    3.7    0.
    0.    0.    4.5

```



```
-->L=tril(A)-D
```

```
L =
```

```
0.    0.    0.
1.3    0.    0.
0.2    2.2    0.
```

```
-->U=triu(A)-D
```

```
U =
```

```
0.    3.1    0.4
0.    0.    2.1
0.    0.    0.
```

```
-->x=[0;0;0];
```

```
-->for i=1:20 x=inv(D)*(b-(L+U)*x)
```

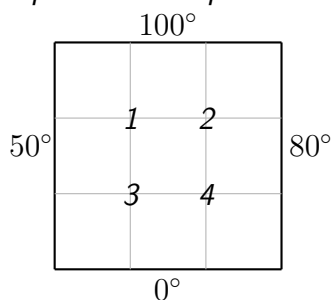
```
-->end
```

La aproximación obtenida en la iteración numero 20 es

```
x =
```

```
1.4569683
- 0.3327386
0.2516418
```

**Actividad 4.** Una placa metálica cuadrada tiene una temperatura constante en cada uno de sus cuatro bordes. Para calcular la temperatura en puntos del interior de la misma, se superpone una rejilla virtual conectando puntos del borde con puntos interiores (ver el ejemplo en la figura) y se supone que la temperatura en cada punto interior es el promedio de las temperaturas en los 4 puntos a los que está conectado por medio de la rejilla.



Así, si  $T_i$  denota la temperatura en el punto  $i$ , tenemos que, por ejemplo,

$$T_1 = \frac{1}{4}(50 + 100 + T_2 + T_3).$$

Calcula las aproximaciones de las temperaturas en los 4 puntos internos de la rejilla con los métodos de Jacobi y de Gauss-Seidel, usando 11 iteraciones y a partir de una aproximación inicial nula.

**Solución**

Escribiendo la temperatura en cada punto como el promedio de las temperaturas de los cuatro puntos vecinos obtenemos el sistema de ecuaciones:

$$\left. \begin{aligned} T_1 &= \frac{1}{4}(50 + 100 + T_2 + T_3) \\ T_2 &= \frac{1}{4}(100 + T_1 + T_4 + 80) \\ T_3 &= \frac{1}{4}(T_1 + 50 + 0 + T_4) \\ T_4 &= \frac{1}{4}(T_2 + T_3 + 0 + 80) \end{aligned} \right\} \quad \left. \begin{aligned} 4T_1 - T_2 - T_3 &= 150 \\ -T_1 + 4T_2 - T_4 &= 180 \\ -T_1 + 4T_3 - T_4 &= 50 \\ -T_2 - T_3 + 4T_4 &= 80 \end{aligned} \right\}$$

La matriz de coeficientes de este sistema es estrictamente diagonal dominante, por tanto ambos métodos convergerán.

Jacobi

```
-->A=[4 -1 -1 0;-1 4 0 -1;-1 0 4 -1;0 -1 -1 4]
```

A =

```
    4.    -1.    -1.     0.
   -1.     4.     0.    -1.
   -1.     0.     4.    -1.
    0.    -1.    -1.     4.
```

```
-->b=[150;180;50;80]
```

b =

```
    150.
    180.
     50.
     80.
```

```
-->D=diag([diag(A)])
```

D =

```
    4.     0.     0.     0.
    0.     4.     0.     0.
    0.     0.     4.     0.
    0.     0.     0.     4.
```

```
-->U=triu(A)-D
```

U =

```
    0.    -1.    -1.     0.
    0.     0.     0.    -1.
    0.     0.     0.    -1.
    0.     0.     0.     0.
```

```
-->L=tril(A)-D
```

L =

```
    0.     0.     0.     0.
   -1.     0.     0.     0.
   -1.     0.     0.     0.
    0.    -1.    -1.     0.
```

```
-->x=[0;0;0;0];
```

```
-->for i=1:11 x=inv(D)*(b-(L+U)*x)
```

```
-->end
```

x =

```
    37.5
    45.
    12.5
    20.
```

```

x =

    51.875
    59.375
    26.875
    34.375
    .
    .
    .
x =

    66.193848
    73.693848
    41.193848
    48.693848
x =

    66.221924
    73.721924
    41.221924
    48.721924

```

Gauss-Seidel

```

-->x=[0;0;0;0];
-->for i=1:11 x=SustitucionProgresiva(L+D,b-U*x)
-->end
x =

    37.5
    54.375
    21.875
    39.0625

x =

    56.5625
    68.90625
    36.40625
    46.328125
    .
    .
    .
x =

    66.249852
    73.749926
    41.249926
    48.749963
x =

    66.249963
    73.749982
    41.249982
    48.749991

```