

# PEC 3 - Sistemas de ecuaciones lineales

Fecha de entrega: 15/03/2022



## Descripción del problema

### 1. Resolución numérica de sistemas de ecuaciones lineales

En este apartado se utilizarán los métodos numéricos que hemos aprendido para dar solución a dos sistemas de ecuaciones. Además, se aconsejará a los alumnos cuándo es más conveniente utilizar los métodos numéricos directos y cuándo los iterativos, para que tengan criterios claros antes de empezar a calcular.

#### 1.1. Métodos numéricos directos

Dada una matriz  $A$ , si esta matriz es invertible y no tiene muchas variables, los métodos directos son los más adecuados. En este apartado se pide resolver el sistema de ecuaciones (S) del tipo  $Ax = b$ :

$$\begin{pmatrix} 1 & 2 & -2 & -1 & 1 \\ 0 & -3 & 1 & 2 & 2 \\ 2 & -2 & -4 & 2 & 7 \\ 3 & -3 & -1 & 5 & 9 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} -2 \\ 3 \\ -3 \\ 13 \\ 2 \end{pmatrix} \quad (1)$$

Se pide:

1. Hacer la descomposición  $LU$  de la matriz  $A$  del sistema (S).
2. Resolver el sistema (S) haciendo uso de las matrices  $L$  y  $U$ .

##### 1.1.1. Solución:

La descomposición  $LU$  de  $A$  es:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 \\ 3 & 3 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad U = \begin{pmatrix} 1 & 2 & -2 & -1 & 1 \\ 0 & -3 & 1 & 2 & 2 \\ 0 & 0 & -2 & 0 & 1 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} \quad (2)$$

Y la solución del sistema de ecuaciones usando métodos directos es:

$$x_1 = 1, \quad x_2 = 2, \quad x_3 = 3, \quad x_4 = 2, \quad x_5 = 1. \quad (3)$$

El código R para resolver este apartado es:

```

1  ## Solucion de sistema de ecuaciones mediante metodos directos
2
3  # Resolucion del sistema mediante factorizacion LU
4
5  require(pracma)
6  # Definicion del sistema Ax = b
7  A = matrix(c(1, 2, -2, -1, 1,
8              0, -3, 1, 2, 2,
9              2, -2, -4, 2, 7,
10             3, -3, -1, 5, 9,
11             0, 0, 0, 0, 2), nrow=5, byrow=TRUE)
12 b = c(-2, 3, -3, 13, 2)
13
14 #Factorizacion LU
15 mlu = lu(A, scheme='ijk') # metodo de Doolittle
16 L = mlu$L
17 U = mlu$U
18 print(L)
19 print(U)
20
21 # Resolucion del sistema a partir de L y U
22 y = solve(L, b)
23 x = solve(U, y)
24 print(x)
25
26 #Comprobamos que obtenemos el mismo resultado, resolviendo de forma directa con
27   solve
28 x_ = solve(A, b)
29 print(x_)

```

## 1.2. Métodos numéricos iterativos

Si una matriz es muy grande y los elementos no nulos son una fracción pequeña de la totalidad de elementos de la matriz, entonces los métodos directos no son los mejores métodos para resolver el sistema. Ello es debido a que, en general, tendríamos problemas de lentitud en el manejo de la memoria por tener que almacenar y manipular toda la matriz.

Los métodos iterativos se usan para resolver problemas con matrices muy grandes, con muchos ceros y con ciertos patrones (matrices tridiagonales, etc), habitualmente problemas que aparecen al discretizar ecuaciones diferenciales en derivadas parciales.

En este apartado se pide resolver el sistema de ecuaciones (S) del tipo  $Ax = b$ :

$$\begin{pmatrix} 1 & -1/2 & 0 & \dots & \dots & 0 \\ -1/2 & 1 & -1/2 & 0 & \dots & 0 \\ 0 & -1/2 & 1 & -1/2 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & -1/2 & 1 & -1/2 \\ 0 & \dots & \dots & 0 & -1/2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ \dots \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} 1/2 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} \quad (4)$$

Considerando  $n = 10$ . Se pide:

1. Estudiar la convergencia del método de Jacobi para resolver el sistema (S). Para ello emplea alguno de los métodos introducidos en la guía para estudiar la convergencia de los métodos iterativos.
2. Convertir el sistema (S) a la iteración de punto fijo  $x^{(k+1)} = Jx^{(k)} + c$  donde  $J$  es la matriz de iteración de Jacobi. Es decir, resolver este sistema de ecuaciones por el método numérico iterativo de **Jacobi**. Indicar las iteraciones para obtener un error inferior a  $10^{-6}$  y el error cometido si solo se realizan 80 iteraciones. Calcular a mano la primera iteración del método.
3. Convertir el sistema (S) a la iteración de punto fijo  $x^{(k+1)} = Gx^{(k)} + d$  donde  $G$  es la matriz de iteración de Gauss-Seidel. Es decir, resolver este sistema de ecuaciones por el método numérico iterativo de **Gauss-Seidel**. Indicar las iteraciones para obtener un error inferior a  $10^{-6}$  y el error cometido si solo se realizan 80 iteraciones. Calcular a mano la primera iteración del método.
4. ¿Cuál de los dos métodos converge más rápido?, es decir, ¿cuál de los 2 tiene menos iteraciones para dar la misma solución?.
5. Considerando  $n = 20$  y  $n = 40$  calcular la solución del sistema por el método de Jacobi y Gauss-Seidel, dar el error de la solución con cada uno de los dos métodos considerando 80 iteraciones, y el número de iteraciones en cada caso para obtener un error inferior a  $10^{-6}$ . Compara estos resultados para los distintos valores de  $n$  que se han considerado.

En ambos casos se pide resolver el sistema de ecuaciones por los métodos numéricos iterativos de Jacobi y Gauss-Seidel, para ello habrá que preparar un código que permita resolver el sistema de ecuaciones usando estos métodos iterativos. Se recomienda usar el paquete de R que más se ajuste al ejercicio, aunque hay muchos otros paquetes que encontraréis tras una breve búsqueda en internet ("Iterative methods"). Se valorará que los alumnos expliquen el funcionamiento de las funciones utilizadas para resolver estos sistemas de ecuaciones por los métodos de Jacobi o Gauss-Seidel, y no se limiten simplemente a aplicar las funciones y resolverlos.

### 1.2.1. Solución

La sucesión iterante en el método de Jacobi viene dada por

$$Dx^{(k+1)} = (L + U)x^{(k)} + b \quad (5)$$

por lo que,

$$J = D^{-1}(L + U) \quad y \quad c = D^{-1}b. \quad (6)$$

Para este caso,

$$J = \begin{pmatrix} 0 & 1/2 & 0 & \dots & \dots & 0 \\ 1/2 & 0 & 1/2 & 0 & \dots & 0 \\ 0 & 1/2 & 0 & 1/2 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 1/2 & 0 & 1/2 \\ 0 & \dots & \dots & 0 & 1/2 & 0 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 1/2 \\ 0 \\ 0 \\ \dots \\ \dots \\ 0 \end{pmatrix} \quad (7)$$

Calculando ahora el radio espectral de  $J$ , es decir, el máximo de los valores absolutos de los autovalores, se obtiene

$$\rho = \max_i |\lambda_i| = 0.959493$$

Como  $\rho < 1$  concluimos que el método de Jacobi nos permite obtener una solución aproximada que converja a la solución del sistema.

Con ello, realizamos la primera iteración de Jacobi, tomando  $x^{(0)} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$ ,

$$x^{(1)} = \begin{pmatrix} 0 & 1/2 & 0 & \dots & \dots & 0 \\ 1/2 & 0 & 1/2 & 0 & \dots & 0 \\ 0 & 1/2 & 0 & 1/2 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 1/2 & 0 & 1/2 \\ 0 & \dots & \dots & 0 & 1/2 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dots \\ \dots \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1/2 \\ 0 \\ 0 \\ \dots \\ \dots \\ 0 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 0 \\ 0 \\ \dots \\ \dots \\ 0 \end{pmatrix}$$

Prescribiendo una tolerancia de  $10^{-6}$ , la solución obtenida mediante el método iterativo de Jacobi es:

$$\begin{aligned} x_1 &= 0.9090899, & x_2 &= 0.8181799, & x_3 &= 0.7272699, & x_4 &= 0.6363604, & x_5 &= 0.5454509, \\ x_6 &= 0.4545420, & x_7 &= 0.3636330, & x_8 &= 0.2727246, & x_9 &= 0.1818162, & x_{10} &= 0.0909081. \end{aligned}$$

El número de iteraciones requeridas con este método es 292, obteniendo un error relativo de 0.03042246 si solo se emplean 80 iteraciones.

La sucesión interante en el método de Gauss-Seidel viene dada por

$$(D - L)x^{(k+1)} = Ux^{(k)} + b \quad (8)$$

por lo que,

$$G = (D - L)^{-1}U \text{ y } d = (D - L)^{-1}b \quad (9)$$

Para este caso,

$$G = \begin{pmatrix} 0 & 1/2 & 0 & \dots & \dots & 0 \\ 0 & 1/2^2 & 1/2 & 0 & \dots & 0 \\ 0 & 1/2^3 & 1/2^2 & 1/2 & \dots & \dots \\ \dots & 1/2^4 & 1/2^3 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & 1/2^2 & 1/2 \\ 0 & 1/2^n & 1/2^{n-1} & \dots & 1/2^3 & 1/2^2 \end{pmatrix} \text{ y } d = \begin{pmatrix} 1/2 \\ 1/2^2 \\ 1/2^3 \\ \dots \\ \dots \\ 1/2^n \end{pmatrix} \quad (10)$$

Con ello, realizamos la primera iteración Gauss-Seidel, tomando  $x^{(0)} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$ ,

$$x^{(1)} = \begin{pmatrix} 0 & 1/2 & 0 & \dots & \dots & 0 \\ 0 & 1/2^2 & 1/2 & 0 & \dots & 0 \\ 0 & 1/2^3 & 1/2^2 & 1/2 & \dots & \dots \\ \dots & 1/2^4 & 1/2^3 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & 1/2^2 & 1/2 \\ 0 & 1/2^n & 1/2^{n-1} & \dots & 1/2^3 & 1/2^2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dots \\ \dots \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1/2 \\ 1/2^2 \\ 1/2^3 \\ \dots \\ \dots \\ 1/2^n \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2^2 \\ 1/2^3 \\ \dots \\ \dots \\ 1/2^n \end{pmatrix} \quad (11)$$

Prescribiendo una tolerancia de  $10^{-6}$ , la solución obtenida mediante el método iterativo de Gauss-Seidel es:

$$\begin{aligned} x_1 &= 0.90908934, & x_2 &= 0.81817893, & x_3 &= 0.72726885, & x_4 &= 0.63635916, & x_5 &= 0.54544988, \\ x_6 &= 0.45454097, & x_7 &= 0.36363241, & x_8 &= 0.27272412, & x_9 &= 0.18181602, & x_{10} &= 0.09090801. \end{aligned}$$

El número de iteraciones requeridas con este método es 141, obteniendo un error relativo de 0.0009494007 con 80 iteraciones.

Como era de esperar, el método de Gauss-Seidel converge más rápido. En general, este método necesita menos iteraciones para alcanzar la solución con el nivel de precisión establecido. Esto es importante ya que, si las matrices fueran mucho más grandes, utilizar un método iterativo que sea más rápido supondría un menor gasto de recursos para obtener la solución.

Si ahora consideramos  $n = 20$ , el número de iteraciones para obtener un error inferior a  $10^{-6}$  es 987 . El error obtenido empleando un total de 80 iteraciones ha sido 0.3288015

En el caso  $n = 40$ , el número de iteraciones para obtener un error inferior a  $10^{-6}$  es 3432 . El error obtenido empleando un total de 80 iteraciones ha sido 0.6470177

Como se puede observar, cuando la dimensión de la matriz es muy grande, observamos que la función `itersolve` para el método de Jacobi realiza más de 1000 iteraciones para obtener un error inferior a  $10^{-6}$  cuando  $n = 40$ , siendo necesario introducir el parámetro  $nmax = 6000$  en la función `itersolve`.

Si ahora consideramos el método de Gauss-Seidel con  $n = 20$ , realizándose un total de 475 iteraciones se obtienen un error inferior a  $10^{-6}$ . Si consideramos solo 80 iteraciones, el error relativo obtenido es: 0.1211957.

En el caso  $n = 40$ , la solución del sistema con el método de Gauss-Seidel tiene un error inferior a  $10^{-6}$  si se realizan 1648 iteraciones. Si consideramos solo 80 iteraciones, el error relativo obtenido es: 0.4772296.

En el caso de utilizar el algoritmo de Gauss-Seidel, vemos que a mayor dimensión de la matriz, el número de iteraciones, también aumenta, siendo necesarias más de 1000 iteraciones cuando  $n = 40$ , sin embargo, como Gauss-seidel converge más rápido que Jacobi, en ambos métodos observamos que el error es mayor para una mayor dimensión, pero que el método de Gauss-seidel, nos ofrece errores menores al converger más rápido cuando comparamos los valores para 80 iteraciones.

El código R para resolver este apartado es:

```
1 ## Solucion de sistema de ecuaciones mediante metodos iterativos
2 # Empleamos el comando de R itersolve(A, b, x0, tol, method), cuyos argumentos son:
3 # A: la matriz del sistema
4 # b: el lado derecho del sistema
5 # x0: aproximacio inicial
6 # tol: condici?n de parada en base al error cometido
7 # method: metodo a utilizar ("Gauss-Seidel" o "Jacobi")
8
9 require(pracma)
10
11 # Resolucion del sistema mediante Jacobi
12
13 n = 10
14 A = diag(n)
15 for(i in 1:n) { A[i,i+1] <- -1/2}
16 for(i in 2:n) { A[i,i-1] <- -1/2}
17 b = rep(0,n)
18 b[1] <- -1/2
19
```

```

20 # Primera iteracion
21 D = diag(diag(A))
22 L = -tril(A, -1)
23 U = -triu(A, 1)
24
25 J = inv(D)%%(L + U)
26 c = inv(D)%%b
27 print(J)
28 print(c)
29
30 x0 = rep(0,n)
31 x1 = J%%x0 + c
32 print(x1)
33
34 # Resolucion iterativa
35 #sol = itersolve(A, b, x0, nmax = 100, method = "Jacobi")
36 #sol = itersolve(A, b, x0, tol = 1e-8, method = "Jacobi", nmax = 6000)
37 sol = itersolve(A, b, x0, tol = 1e-8, method = "Jacobi")
38 print(sol)
39
40 # Solucion exacta
41 x_ = solve(A, b)
42 dif_J = sol$x - x_
43 error_J = norm(dif_J, "2")
44 error_rel_J = error_J/norm(x_, "2")
45 print(error_rel_J)
46
47 # Resolucion del sistema mediante Gauss-Seidel
48
49 n = 10, #20, #40
50 A = diag(n)
51 for(i in 1:n-1) { A[i,i+1] <- -1/2}
52 for(i in 2:n) { A[i,i-1] <- -1/2}
53 b = rep(0,n)
54 b[1] <- -1/2
55
56
57 # Primera iteracion
58 D = diag(diag(A))
59 L = -tril(A, -1)
60 M = D - L
61 U = -triu(A, 1)
62
63 G = inv(M)%%U
64 d = inv(M)%%b
65 print(G)
66 print(d)
67
68 x0 = rep(0,n)
69 x1 = G%%x0 + d

```



```

70 print(x1)
71
72 # Resolucion iterativa
73 #sol = itersolve(A, b, x0, nmax = 100, method = "Gauss-Seidel")
74 #sol = itersolve(A, b, x0, tol = 1e-8, method = "Gauss-Seidel", nmax = 6000)
75 sol = itersolve(A, b, x0, tol = 1e-8, method = "Gauss-Seidel")
76 print(sol)
77
78 #Comprobamos que obtenemos el mismo resultado, resolviendo de forma directa con
    solve
79 # x_ = solve(A, b)
80 print(x_)
81
82
83 x_ = solve(A, b)
84 dif_GS = sol$x - x_
85 error_J = norm(dif_GS, "2")
86 error_rel_GS = error_GS/norm(x_, "2")
87 print(error_rel_GS)

```

## Criterios de corrección y puntuación de los apartados.

Resolución de sistemas de ecuaciones lineales. Esta actividad se evaluará como sigue:

- 1. Resolución de sistemas de ecuaciones lineales. Esta actividad se evaluará como sigue:
  - Tarea 1.1: Esta tarea se puntuará con 2.5 puntos. El cálculo correcto de las matrices,  $L$  y  $U$ , vale 2 puntos (1 por cada matriz bien calculada). La resolución del sistema se evaluará con 0.5 puntos.
  - Tarea 1.2: En esta tarea se pide resolver el sistema ( $S$ ) por el método de Jacobi y por el método de Gauss-Seidel. Además, se pide comparar la convergencia de ambos métodos. La puntuación será de 7.5 puntos distribuidos en: 1 punto por estudiar la convergencia por el método de Jacobi (apartado 1). 1 punto por calcular la solución mediante el método de Jacobi y 0.5 por el cálculo de la primera iteración, 0.6 por dar el error y 0.4 por dar el número de iteraciones (apartado 2). De la misma forma, por dar la solución por el método de Gauss-Seidel (apartado 3) se puntuará con 1 punto, el cálculo de la primera iteración se puntuará con 0.5, por dar el error 0.6 y por dar el número de iteraciones se puntuará 0.4. Además se puntuará 1 por el apartado (4), es decir, por razonar con los resultados obtenidos en los apartados (2) y (3) cuál de los 2 métodos converge más rápido. Finalmente, el apartado (5) se puntuará con 0.5 puntos.