

Problema 1

Sobre el eje x hay situadas N cargas eléctricas puntuales Q_1, \dots, Q_N que pueden moverse únicamente en la dirección del mismo. Además del campo eléctrico generado por ellas mismas, las cargas están sometidas al campo eléctrico generado por dos hilos infinitos de densidad lineal de carga λ , perpendiculares al plano del papel, que pasan por los puntos $P_1 = (0, 0)$ y $P_2 = (L, 0)$, véase Fig. 1.

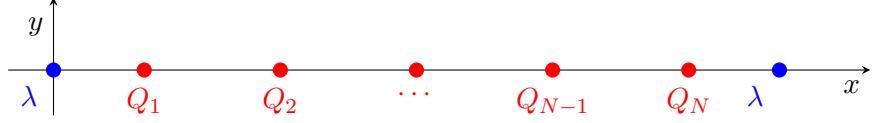


Figura 1: Esquema de las variables del Problema 1.

Se desea hallar la posición de equilibrio estático de las cargas. Se sabe que esto ocurre cuando el campo eléctrico sobre cada carga Q_i , dado por

$$E_i(\mathbf{x}) = 2K\lambda \left(\frac{1}{x_i} - \frac{1}{L-x_i} \right) + K \sum_{k=1, k \neq i}^N Q_k \frac{x_i - x_k}{|x_i - x_k|^3}, \quad i = 1, \dots, N,$$

es nulo. En la ecuación anterior, x_i es la coordenada horizontal de la carga Q_i , $\mathbf{x} = [x_1, \dots, x_N]^T$ y K es la constante de Coulomb.

Cuestión 1 (1pt): Demostrar que

$$\frac{\partial E_i}{\partial x_j} = \begin{cases} -2K\lambda \left(\frac{1}{x_i^2} + \frac{1}{(L-x_i)^2} \right) - 2K \sum_{k=1, k \neq i}^N Q_k \frac{1}{|x_i - x_k|^3}, & i = j, \\ 2KQ_j \frac{1}{|x_i - x_j|^3}, & i \neq j. \end{cases}$$

Solución: Empleando la regla de la cadena y recordando que $\partial x_i / \partial x_j = \delta_{ij}$, siendo δ_{ij} la delta de Kronecker, el término pedido puede escribirse como

$$\begin{aligned} \frac{\partial E_i}{\partial x_j} &= 2K\lambda \left(-\frac{1}{x_i^2} - \frac{1}{(L-x_i)^2} \right) \delta_{ij} + \\ &\quad + K \sum_{k=1, k \neq i}^N Q_k \left(\frac{\delta_{ij} - \delta_{kj}}{|x_i - x_k|^3} - (x_i - x_k) \frac{-3}{|x_i - x_k|^4} \frac{x_i - x_k}{|x_i - x_k|} (\delta_{ij} - \delta_{kj}) \right), \end{aligned}$$

donde se ha hecho uso de que $d|u|/du = u/|u|$. Notando que $(x_i - x_k)^2 = |x_i - x_k|^2$, se tiene entonces

$$\begin{aligned} \frac{\partial E_i}{\partial x_j} &= 2K\lambda \left(-\frac{1}{x_i^2} - \frac{1}{(L-x_i)^2} \right) \delta_{ij} + K \sum_{k=1, k \neq i}^N Q_k \left(\frac{\delta_{ij} - \delta_{kj}}{|x_i - x_k|^3} - 3 \frac{|x_i - x_k|^3}{|x_i - x_k|^5} (\delta_{ij} - \delta_{kj}) \right) \\ &= 2K\lambda \left(-\frac{1}{x_i^2} - \frac{1}{(L-x_i)^2} \right) \delta_{ij} - 2K \sum_{k=1, k \neq i}^N Q_k \frac{\delta_{ij} - \delta_{kj}}{|x_i - x_k|^3}. \end{aligned}$$

Si $j = i$, queda entonces

$$\frac{\partial E_i}{\partial x_j} = 2K\lambda \left(-\frac{1}{x_i^2} - \frac{1}{(L-x_i)^2} \right) - 2K \sum_{k=1, k \neq i}^N Q_k \frac{1}{|x_i - x_k|^3},$$

mientras que, si $j \neq i$, entonces

$$\frac{\partial E_i}{\partial x_j} = -2K \sum_{k=1, k \neq i}^N Q_k \frac{-\delta_{kj}}{|x_i - x_k|^3} = 2KQ_j \frac{1}{|x_i - x_j|^3}.$$

□

Cuestión 2 (2.5pt): Implementar una función de la forma

```
function [E,J] = CargasElectricasResiduo(K,lambda,L,Q,x,CalcJ)
```

que reciba los valores de K , λ y L , un vector columna $\mathbf{Q} = [Q_1, \dots, Q_N]^T$ con los valores de las cargas puntuales, el vector columna $\mathbf{x} = [x_1, \dots, x_N]^T$ con las posiciones de las mismas y una variable booleana $CalcJ$ y devuelva un vector columna $\mathbf{E} = [E_1, \dots, E_N]^T$ con el campo eléctrico en cada carga y, si $CalcJ=true$, el Jacobiano del mismo, definido como $J_{ij} := \partial E_i / \partial x_j$. Subir la función a Moodle. Puede ser útil saber que la función valor absoluto está implementada bajo el comando `abs`.

Solución: Véase archivo `CargasElectricasResiduo1.m` aparte. □

De ahora en adelante, utilícese la función codificada `CargasElectricasResiduo1` disponible en Moodle. Asimismo, considérense los parámetros $K = 2.1$, $\lambda = 0.4$ y $L = 8$, así como $N = 10$ cargas puntuales de idéntico valor $Q_i = 0.5$, $i = 1, \dots, N$.

A continuación, se implementará una función de la forma `function Problema1`, sin argumentos de entrada ni de salida, que realice los siguientes pasos:

1. Definir los parámetros K , λ y L y el vector de cargas \mathbf{Q} .
2. Resolver el sistema de ecuaciones $E_i(\mathbf{x}) = 0$, $i = 1, \dots, N$, mediante el método de Newton–Raphson para hallar la posición de equilibrio estático de cada carga. Tómese, como condición inicial \mathbf{x}^0 , un vector de N componentes equiespaciadas entre $0.1L$ y $0.9L$, así como una tolerancia de 10^{-8} y un máximo de 20 iteraciones. Para ello, puede ser útil definir la función auxiliar

```
fun = @(x,CalcJ) CargasElectricasResiduo1(K,lambda,L,Q,x,CalcJ);
```

y llamar al método de Newton–Raphson mediante `NewtonRaphson1(fun, ...)`.

3. Dibujar, en una nueva figura, la posición de cada carga con un círculo azul (opción `'ob'`) en el comando `plot`. Supóngase que la coordenada vertical de cada carga es cero.

Cuestión 3 (1.5pt): Crear la función `Problema1` y subirla a Moodle.

Solución: Véase archivo `Problema1.m` aparte. □

Problema 2

Sea $\mathbf{A} \in \mathbb{R}^{N \times N}$ una matriz cuyos elementos no nulos son

$$a_{ii} = 12, \quad a_{i+\alpha_i, i} = -1, \quad a_{i, i+\beta_i} = -5, \quad i = 1, \dots, N-1,$$

y $a_{NN} = 12$, donde

$$\alpha_i = \left\lceil (N-i) \left(0.01 + 0.49 \left(1 - \cos \left(\frac{5\pi i}{N} \right) \right) \right) \right\rceil, \quad \beta_i = \left\lceil (N-i) \left(0.01 + 0.49 \left(1 + \cos \left(\frac{5\pi i}{N} \right) \right) \right) \right\rceil,$$

y $\lceil \cdot \rceil$ representa la función techo (orden `ceil`).

Cuestión 4 (2pt): Crear una función de la forma `function A=MatrizExamen(N)` que, recibido el valor de N , construya la matriz dispersa \mathbf{A} . Subir la función a Moodle.

Solución: Véase archivo `MatrizExamen1.m` aparte. □

De ahora en adelante, utilícese la función codificada `MatrizExamen1` disponible en Moodle.

Se desea implementar una función de la forma `function Problema2(N)`, sin argumentos de salida, que realice los siguientes pasos:

1. Crear la matriz \mathbf{A} anterior llamando a la función `MatrizExamen1` y representar el patrón de elementos no nulos (orden `spy`) en una nueva figura. Obsérvese que la figura indica el número total de elementos no nulos.
2. Crear los vectores columna de N componentes definidos por $\mathbf{x}^* = [1, \dots, 1]^T$ y $\mathbf{b} := \mathbf{A}\mathbf{x}^*$.
3. Hallar la factorización LU más adecuada, mostrando el tiempo empleado por pantalla, y representar el patrón de elementos no nulos de $\mathbf{L} + \mathbf{U}$ en una nueva figura.
4. Resolver el sistema $\mathbf{A}\mathbf{x} = \mathbf{b}$ utilizando dicha factorización, mostrando el tiempo empleado por pantalla.

Existe una factorización LU aproximada, $\mathbf{A} \approx \mathbf{LU}$, llamada *factorización LU incompleta* o *ILU0*, que consiste en realizar la factorización LU estándar y anular los términos de llenado (*fill-in*). De esta forma, $\mathbf{L} + \mathbf{U}$ tiene el mismo patrón de elementos no nulos que \mathbf{A} y $a_{ij} = \sum_{k=1}^N l_{ik} u_{kj}$ para todos los elementos no nulos a_{ij} . Dicha factorización puede obtenerse con la instrucción `[L, U]=ilu0(A)`, siendo `ilu0` una función disponible en Moodle. Realizar los siguientes pasos adicionales:

5. Hallar la factorización ILU0 llamando a la función `ilu0`, mostrando el tiempo empleado por pantalla, y representar el patrón de elementos (orden `spy`) no nulos de $\mathbf{L} + \mathbf{U}$ en una nueva figura. (No confundir estas matrices \mathbf{L} y \mathbf{U} con las del paso 3.)
6. Resolver el sistema

$$\mathbf{g}(x) := \mathbf{U}^{-1} \mathbf{L}^{-1} (\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{0} \tag{1}$$

mediante el método de Anderson, mostrando por pantalla el tiempo empleado y el número de iteraciones. Tómese el vector nulo como condición inicial, una tolerancia de 10^{-8} , un máximo de 100 iteraciones y 20 pasos almacenados.

Cuestión 5 (1pt): Justificar la definición del residuo precondicionado \mathbf{g} dado por la Ec. (1).

Solución: Queremos resolver la ecuación lineal $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$ mediante el método de Anderson. Para que la resolución sea eficiente, debemos definir un residuo precondicionado de la forma $\mathbf{g}(\mathbf{x}) = (\mathbf{A}^*)^{-1} \mathbf{f}(\mathbf{x}) = (\mathbf{A}^*)^{-1} (\mathbf{A}\mathbf{x} - \mathbf{b})$, donde \mathbf{A}^* es un operador que

- (i) aproxima bien la matriz del sistema \mathbf{A} , de tal modo que la matriz jacobiana $\partial \mathbf{g} / \partial \mathbf{x} \approx \mathbb{I}$,
- (ii) es fácil de invertir, de tal forma que el cálculo de \mathbf{g} sea rápido.

En este caso, tomamos $\mathbf{A}^* = \mathbf{LU} \approx \mathbf{A}$, siendo \mathbf{L} y \mathbf{U} las matrices de la factorización ILU0. Como vemos, \mathbf{A}^* es una aproximación razonable a \mathbf{A} y al mismo tiempo es fácil de invertir puesto que los factores \mathbf{L} y \mathbf{U} son triangulares y dispersos. □

Cuestión 6 (2pt): Crear la función *Problema2* y subirla a Moodle. Ejecutar la función para $N = 10^5$ (reducir sucesivamente este valor a la mitad si el ordenador tarda más de dos minutos) y completar la Tabla 1 (con dos cifras significativas es suficiente). ¿Qué método es más eficiente? ¿A qué se debe la diferencia?

Solución: Véase el archivo *Problema2.m* aparte. La Tabla 1 está completada para un ordenador IntelCore i9-7900X de diez núcleos a 3.30 GHz de frecuencia y 32GB de RAM¹.

Los métodos a comparar son el directo (factorización $\mathbb{P}\mathbb{A}\mathbb{Q} = \mathbb{L}\mathbb{U}$, al tratarse de una matriz dispersa no simétrica) y el método iterativo (Anderson precondicionado con ILU0). En el caso del método directo, podemos ver que, a pesar de que las permutaciones de filas y columnas reducen el fenómeno de llenado en las matrices \mathbb{L} y \mathbb{U} respecto de otras factorizaciones LU, éste se sigue produciendo y es notable, véase Fig. 2. En particular, puede verse que el número de elementos no nulos se multiplica por un factor de casi 10^3 . Esto hace que el método directo sea muy lento cuando N es grande.

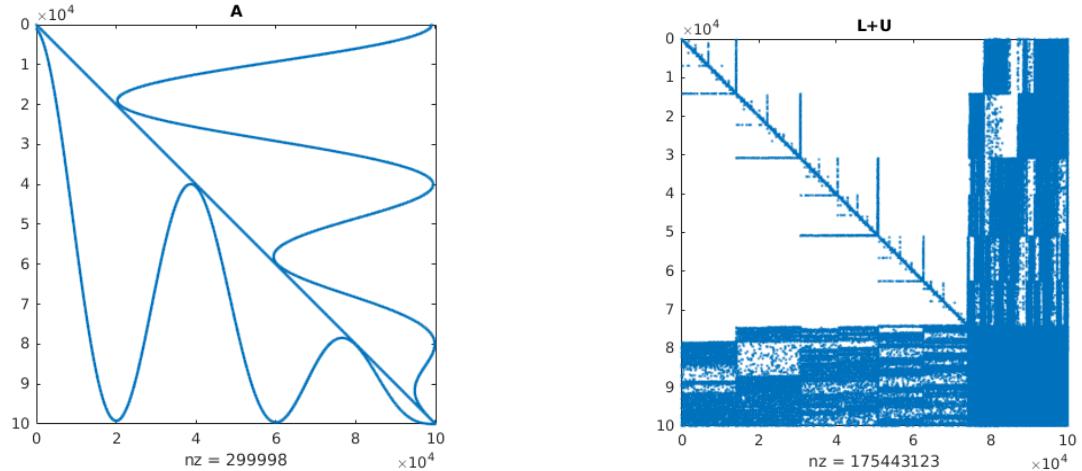


Figura 2: Patrones de elementos no nulos de \mathbb{A} y los factores \mathbb{L} y \mathbb{U} de la factorización $\mathbb{P}\mathbb{A}\mathbb{Q} = \mathbb{L}\mathbb{U}$.

En cambio, con la factorización ILU0, el número de elementos no nulos en los factores \mathbb{L} y \mathbb{U} es el mismo que el de la matriz original, lo cual hace que su cálculo sea económico. Asimismo, esta factorización resulta ser un buen precondicionador para el método de Anderson, ya que éste requiere pocas iteraciones (nueve) para encontrar la solución. Por tanto, el método iterativo Anderson + ILU0 requiere pocas iteraciones y cada una de ellas es muy económica, de ahí que resulte mucho más eficiente que el método directo. \square

Tabla 1: Resultados del Problema 2.

N	10^5
Coste [s] factorización LU	31
Coste [s] resolución a partir de LU	0.25
Número de elementos no nulos en $\mathbb{L} + \mathbb{U}$	$1.8 \cdot 10^8$
Coste [s] factorización ILU0	0.19
Coste [s] resolución con Anderson	0.66
Número de elementos no nulos en $\mathbb{L} + \mathbb{U}$	$3.0 \cdot 10^5$
Número de iteraciones Anderson	9

¹Este comentario es únicamente a título informativo, no se requiere poner estos datos en la solución. El número de núcleos y la frecuencia del procesador determinan la velocidad de los cálculos, mientras que la memoria RAM determina el tamaño de las matrices que pueden almacenarse.