Análisis Numérico — Entrega Práctica 4

Segundo Cuatrimestre de 2012

1. El problema:

Considerar el problema Poisson en un cuadrado:
Dada $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$, se quiere hallar $u : \mathbb{R}^2 \longrightarrow \mathbb{R}$ tal que:

$$-\Delta u = f(x,y) \qquad (x,y) \in [0,1] \times [0,1], \tag{1}$$

Físicamente f representa la densidad de carga eléctrica, mientras que la incógnita u es el potencial eléctrico. Nos interesa calcular tanto u como el campo eléctrico generado por la distribución de cargas f, que viene dado por $E : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$,

$$E = -\nabla u$$

El objetivo de esta entrega es implementar un algoritmo de Elementos Finitos para resolver este problema, utilizando elementos lineales en cada triángulo. En particular, se desean estudiar las siguientes situaciones:

Dirichlet homogéneas: Resolver (1) con condiciones

$$u(0,y) = u(1,y) = 0 \quad y \in [0,1]$$

$$u(x,0) = u(x,1) = 0 \quad x \in [0,1]$$

Estudiar los casos:

(1) *Fuente puntual:*

$$f(x,y) = \begin{cases} 1 & \text{si } x = y = \frac{1}{2} \\ 0 & \text{si no} \end{cases}$$

(2) *Dipolo:*

$$f(x,y) = \begin{cases} 1 & \text{si } x = \frac{1}{4},\ y = \frac{3}{4} \\ -1 & \text{si } x = \frac{3}{4},\ y = \frac{1}{4} \\ 0 & \text{si no} \end{cases}$$

(3) *Espira cargada:*

$$f(x,y) = \begin{cases} 1 & \text{si } 0{,}124 \leq (x-\frac{1}{2})^2 + (y-\frac{1}{2})^2 \leq 0{,}126 \\ 0 & \text{si no} \end{cases}$$

Dirichlet-Neumann: Resolver (1) con condiciones de contorno:

$$u(0,y) = u(x,0) = 0 \quad x,y \in [0,1]$$

$$\frac{\partial u}{\partial x}(1,y) = 0 \quad y \in (0,1)$$

$$\frac{\partial u}{\partial y}(x,1) = 0 \quad x \in (0,1)$$

Estudiar todos los casos anteriores.

Pila (simplificada): Esquemáticamente, una pila está formada por dos placas paralelas de material conductor, encerradas en una cápsula blindada. Al aplicar una diferencia de potencial entre las placas se genera un campo eléctrico que induce una corriente de una capa hacia la otra. Para simular esta situación, podemos imponer las siguientes condiciones de borde:

$$u(x,0) = -1 \quad x \in [0,1]$$

$$u(x,1) = 1 \quad x \in [0,1]$$

$$\frac{\partial u}{\partial y}(0,y) = \frac{\partial u}{\partial y}(1,y) = 0 \quad y \in (0,1)$$

De este modo las condiciones de Neumann en el *techo* y en el *piso* simulan el blindaje de la pila, mientras que los valores de u en los laterales dan la diferencia de potencial. ¿Qué sucede si los valores de potencial -1 y 1 se reemplazan por 1 y 3? ¿Qué sucede si las placas no están colocadas paralelas, sino adyacentes?

2. Sugerencias:

2.1. Estructura general

Para poder tomar en cuenta todas las variantes propuestas sin necesidad de implementar un programa para cada item, recomendamos estructurar el trabajo en torno a un programa principal, lo más general posible, que haga uso de subrutinas para la realización de cada tarea específica.

El algoritmo consta de las siguientes partes:

1. Generador de triangulación.
2. Detector de nodos Dirichlet.
3. Iteración principal: recorre los triángulos y construye el sistema lineal. Para ello hace uso de 4 y 5.
4. Generador de la base nodal local para un triángulo dado.
5. Cuadratura en triángulos.
6. Resolución del sistema lineal: obtención de u.
7. Cálculo de $E = -\nabla u$.
8. Graficos.

Salvo 3, que es el centro del algoritmo, los demás ítems pueden formar parte del programa principal o estar implementados en rutinas independientes, de acuerdo al gusto de cada cual.

2.2. Triangulación:

Para definir la triangulación del dominio, recomendamos generar primero una malla rectangular, con el comando `meshgrid`. Para ello los pasos espaciales en x e y deben ser parámetros del programa. A partir de esta malla rectangular, almacenar las coordenadas de los nodos en una matriz de $N \times 2$, siendo N el número total de nodos. Finalmente, utilizar el comando `delaunay` para generar la malla triangular.

La grilla rectangular no es sólo una herramienta para facilitar la creación de la triangulación: también será útil para graficar el campo E.

2.3. Condiciones de borde:

Recuerden que las condiciones de Dirichlet son *esenciales* y deben imponerse al espacio de funciones, mientras que las condiciones Neumann se cumplen naturalmente. Por esta razón será necesario distinguir los nodos con condición Dirichlet de aquellos con condición de tipo Neumann. Para los ejemplos que queremos estudiar lo ideal es que el usuario indique qué lados tienen condición Dirichlet y que haya una rutina sencilla que detecte los nodos ubicados sobre estos lados.

2.4. Bases nodales:

Es conveniente implementar una rutina que reciba como dato los vértices de un triángulo y devuelva un arreglo (comando `cell`) de 3 funciones ϕ_i, $i = 1, 2, 3$ de modo que ϕ_i valga 1 en el nodo i y se anule en los otros.

2.5. Cuadratura:

Preferentemente, implementen la cuadratura del Ejercicio 19 de la Práctica 4.

2.6. Cálculo de E:

Para simplificar el cálculo de ∇u es recomendable reensamblar el vector solución en forma de matriz. Pueden utilizar el comando `reshape`. Tengan en cuenta que, en Matlab, dada la matriz con coeficientes $U(i, j)$, el coeficiente i representa las filas, y por lo tanto se corresponde, en el dibujo, con la variable y.

2.7. Graficos:

Para graficar el potencial u utilicen el comando `trimesh`. Para E, el comando `quiver`.