Análisis Numérico - TP 3: **Elementos Finitos: Ecuación de Poisson**

Segundo Cuatrimestre de 2014

1. Descripción del problema

En esta práctica queremos encontrar el potencial eléctrico u en un dominio Ω generado por una densidad de carga f. Para ello, resolveremos la ecuación de Poisson

$$-\Delta u = f(x,y) \qquad (x,y) \in \Omega, \tag{1}$$

con $f, u : \Omega \longrightarrow \mathbb{R}$, usando el método de los elementos finitos. Consideraremos 3 configuraciones:

1. **Fuente Puntual en un octógono:** Tomaremos:
   - Dominio Ω igual al octógono regular de apotema 1.
   - Condición de frontera dirichlet igual 0: $u(x) = 0 \ \forall (x,y) \in \partial\Omega$.
   - Densidad de carga:

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

2. **Espira en un hexadecágono:** Tomaremos:
   - Dominio Ω igual al hexadecágono regular de apotema 1.
   - Condición de frontera dirichlet igual 0: $u(x) = 0 \ \forall (x,y) \in \partial\Omega$.
   - Densidad de carga:

   $$f(x,y) = \begin{cases} 1 & \text{si } 0{,}24 \leq x^2 + y^2 \leq 0{,}25 \\ 0 & \text{si no} \end{cases}$$

3. **Pila (simplificada) en un cuadrado:** Tomaremos:
   - Dominio Ω igual al cuadrado $[-2,2] \times [-1,1]$.
   - Condición de frontera:
     - $u\big|_{\Gamma_1} = -1$, con $\Gamma_1 = \{x = -2\}$
     - $u\big|_{\Gamma_2} = 1$, con $\Gamma_1 = \{x = 2\}$
     - $\frac{\partial u}{\partial \eta}\Big|_{\Gamma_3} = 0$, con $\Gamma_3 = \{y = -1\} \cup \{y = 1\}$
   - Densidad de carga: $f(x,y) \equiv 0$

2. Tarea

Concretamente, lo que se les pide en este Trabajo práctico es lo siguiente: Completar el archivo `solveP_FEM.m`, donde la funcón `solveP_FEM(caso, ... )` permite elegir qué caso de los descritos nos interesa y resuelve la ecuación variacional usando el método de elementos finitos en una triangulacón del tamaño `hmax` deseado.

Para ello puede necesitar más rutinas de las especificadas en el archivo (ver secección *Sugerencias*) pero, particularmente importante, deberá escribir una rutina `fem(P,T,ND,B,f)` que calcule la matriz de rigidez A asociada los puntos libres de la triangulacón y el término independiente $rhs = \mathcal{L}(f)$.

2.1. OPCIONAL: Cálculo de ∇u

En algún momento puede ser interesante para ustedes calcular el campo $E = -\nabla u$ asociado a un potencial. Si les interesa, pueden practicar haciendolo para los casos que se estudian en este ejercicio. Para ello se puede seguir la siguiente guía: Tomar $P \in \mathbb{R}^{N\times 2}$ la matriz de nodos y $T \in \mathbb{N}^{NT\times 3}$ la de triángulos. Para cada nodo i:

- Hallar el primer triángulo t (fila de T) que contiene a i.
- Llamar $a_1 = i$ y a_2 y a_3 los otros nodos que participan en t.
- Observar que son conocidos los valores de u en los nodos: $u(a_1)$, $u(a_2)$, $u(a_3)$.
- Considerar $r_1 = a_2 - a_1$ y $r_2 = a_3 - a_2$, los vectores correspondientes a los lados de t adyacentes a i.
- Observar que las derivadas direccionales satisfacen la siguiente propiedad:

  $$\frac{\partial u}{\partial r_1}(a_1) = \nabla u \cdot \frac{r_1}{\|r_1\|} \qquad \frac{\partial u}{\partial r_2}(a_1) = \nabla u \cdot \frac{r_2}{\|r_2\|}$$

  y que a su vez pueden estimarse a través de los cocientes incrementales:

  $$\frac{\partial u}{\partial r_1}(a_1) \sim \frac{u(a_2) - u(a_1)}{\|r_1\|} \qquad \frac{\partial u}{\partial r_2}(a_1) \sim \frac{u(a_3) - u(a_1)}{\|r_2\|}$$

- Teniendo en cuenta el item anterior se tiene que si:

  $$A = \begin{bmatrix} \frac{r_1}{\|r_1\|} \\ \frac{r_2}{\|r_2\|} \end{bmatrix} \qquad b = \begin{pmatrix} \frac{u(a_2)-u(a_1)}{\|r_1\|} \\ \frac{u(a_3)-u(a_1)}{\|r_2\|} \end{pmatrix}$$

  entonces:

  $$A \cdot \nabla u = b$$

  $\nabla u(a_1)$ puede calcularse construyendo la matriz A y el vector b y tomando `grad=A\b`

Es conveniente guardar los vectores `grad` de cada nodo i como las filas de una matriz $G \in \mathbb{R}^{N\times 2}$. De este modo G tiene el mismo tamao que P, y cada fila de G contiene el gradiente de u en el nodo dado por la correspondiente fila de P.

ESTE APARTADO NO SE EVALUARÁ.

3. Sugerencias:

3.1. Estructura general

La idea es que el trabajo se estructure en torno a un programa principal, lo más general posible, que haga uso de subrutinas para la realización de cada tarea específica. La rutina `solveP_FEM` se encargará de seleccionar el poblema concreto a solucionar y pasar la información necesaria a este programa principal.

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. (OPCIONAL) Cálculo del campo $E = -\nabla u$.
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.

3.2. 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.

Alternativamente, puede usar el comando `initmesh` de matlab. En este caso (si no partimos de la malla rectangular) deberá definir primero la geometria del dominio. Para ello puede usar los comandos `pdeGeometryFromEdges` y `decsg`. Para describir un poligono arbitrario, se utiliza un vector columna donde: la primera fila es un 2, la segunda contiene el nmero N de lneas rectas que componen dicho polgono, las N siguientes contienen las coordenadas x de cada uno de los puntos de partida de las rectas, y las N siguientes contienen las coordenadas y de cada uno de los puntos de partida de las rectas.

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

3.3. 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 haya una rutina sencilla que detecte los nodos ubicados sobre los lados en los que se define esta condición.

3.4. 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.

3.5. Cuadratura:

Pueden implementarla (por ejemplo, usar la cuadratura del Ejercicio 22 de la Práctica 4) o usar funciones propias de MATLAB, como `integral2`.

3.6. Cálculo de $E = \nabla u$:

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.

3.7. Graficos:

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