
ANÁLISIS NUMÉRICO

Segundo Cuatrimestre 2019

TP N° 2 - Tema 1: Campo eléctrico

1 El problema:

Considerar el problema de Poisson:

Dada $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, se quiere hallar $u : \mathbb{R}^2 \rightarrow \mathbb{R}$ tal que:

$$-\Delta u = f(x, y) \quad (x, y) \in \Omega = B(0, 1) \quad (1)$$

con condiciones de Dirichlet homogéneas.

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 $E : \mathbb{R}^2 \rightarrow \mathbb{R}^2$,

$$E = -\nabla u$$

El objetivo de este trabajo es implementar un algoritmo de elementos finitos para resolver este problema y estimar el orden de convergencia.

2 Formulación Discreta

Dada una triangulación \mathcal{T}_h de Ω , considerar el espacio V_h de funciones continuas y lineales en cada triángulo. Llamemos $\{\phi_i\}_i$ a la base nodal de V_h . La discretización de la formulación débil del problema conduce a la necesidad de calcular la matriz de rigidez R y el vector b , dados por:

$$R_{i,j} = \int_{\Omega} \nabla \phi_i \nabla \phi_j, \quad b_i = \int_{\Omega} f \phi_i$$

El problema discreto se reduce a resolver $R \cdot U = b$, donde U son los coeficientes de u_h . En nuestro caso nos interesa también calcular ∇u , para lo cual necesitaremos una función que tome como datos U y la triangulación y calcule, en cada nodo, el valor de ∇u_h . Para ello, se sugiere seguir una lógica similar a la de la construcción de la matriz de rigidez. Es decir: hacer un ciclo sobre triángulos y calcular el gradiente de u_h en cada nodo localmente. Estos valores se suman y al finalizar el valor acumulado sobre cada nodo se divide por el número de triángulos incidentes sobre el nodo. Así, se calcula, en cada nodo el promedio de los gradientes de u_h sobre los triángulos incidentes.

3 Error

Se quiere también estimar el error de aproximación en L^2 . Para ello es necesario calcular:

$$\left(\int_{\Omega_h} (u - u_h)^2 \right)^{\frac{1}{2}} = \sum_T \left(\int_T (u - u_h)^2 \right)^{\frac{1}{2}}.$$

Donde Ω_h es el polígono mallado, que aproxima a Ω . Es importante observar que hay otro error involucrado en la resolución, que es el error de aproximación entre Ω_h y Ω . Utilizando elementos lineales y siendo $\partial\Omega$ suave, esto no afectará el orden de convergencia, por lo que no lo tomaremos en cuenta. Sin embargo, cuando se trabaja con elementos de mayor orden es necesario incluir este error en el análisis.

Para calcular el error local, utilizaremos la cuadratura:

$$\int_T f(x) dx \sim |T|(-27f(q_0) + 25f(q_1) + 25f(q_2) + 25f(q_3))/48,$$

donde los nodos pueden calcularse como:

$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{5} & \frac{3}{5} \\ \frac{1}{5} & \frac{1}{5} \\ \frac{3}{5} & \frac{1}{5} \end{pmatrix} \cdot \begin{pmatrix} p_1 - p_0 & \vdots & p_2 - p_0 \end{pmatrix} + \begin{pmatrix} p_0 \\ p_0 \\ p_0 \\ p_0 \end{pmatrix}.$$

siendo p_0, p_1, p_2 los vértices del triángulo.

Es importante tener en cuenta que como u_h es lineal en T , los valores $u_h(q_i)$ pueden calcularse sencillamente:

$$\begin{pmatrix} u_h(q_0) \\ u_h(q_1) \\ u_h(q_2) \\ u_h(q_3) \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{5} & \frac{3}{5} \\ \frac{1}{5} & \frac{1}{5} \\ \frac{3}{5} & \frac{1}{5} \end{pmatrix} \cdot \begin{pmatrix} u_h(p_1) - u_h(p_0) & \vdots & u_h(p_2) - u_h(p_0) \end{pmatrix} + \begin{pmatrix} u_h(p_0) \\ u_h(p_0) \\ u_h(p_0) \\ u_h(p_0) \end{pmatrix}.$$

4 Datos

Resolver (1) para:

- $f(x, y) = 4$, que da por solución exacta: $u(x, y) = 1 - x^2 - y^2$
- para una fuente puntual f tal que $\int f v = v(0, 0)$, que da por solución exacta: $u(x, y) = \frac{1}{2\pi} \log(\sqrt{x^2 + y^2})$.

Resolver para distintos valores de h (por ejemplo: $h = 2^{-2}, 2^{-3}, 2^{-4}$) y graficar, en escala logarítmica, el error en función de $\frac{1}{N^2}$, donde N el número de vértices de la malla.

4.1 Comandos útiles:

Pueden resultar útiles las siguientes funciones:

- `Delaunay`, de la biblioteca `scipy.spatial` para la construcción de la triangulación. Alternativamente, puede usarse `Triangulation` de `matplotlib.tri`.
- `csr_matrix` de la biblioteca `scipy.sparse` para el ensamblado de la matriz de rigidez.
- `spsolve` o `cg` de `scipy.sparse.linalg` para la resolución de sistemas.
- `plot_trisurf`, de `matplotlib.pyplot` para el gráfico del potencial u y `quiver` para el gráfico de $-\nabla u$. Preferentemente, graficar ambos en una misma figura, con dos subplots.
- `loglog` de `matplotlib.pyplot` para gráficos en escala logarítmica.