Análisis Numérico

Segundo Cuatrimestre 2025

FEM 2D: Problemas de reacción-difusión singularmente perturbados.

El objetivo es resolver el siguiente problema de reacción-difusión con condiciones de Dirichlet homogeneas:

$$\begin{cases} -\varepsilon^2 \Delta u + u = f & \Omega = [0,1]^2 \\ u = 0 & \Gamma = \partial\Omega. \end{cases}$$

Supongamos que se toma f tal que

$$u(x,y) = \left(1 - e^{-(\frac{1-x}{\varepsilon})}\right)\left(1 - e^{-(\frac{1-y}{\varepsilon})}\right)\left(1 - e^{\frac{-x}{\varepsilon}}\right)\left(1 - e^{\frac{-y}{\varepsilon}}\right)$$

es solución (notar que dicha función se anula en Γ).

Ejercicio 1 Utilizando el paquete `DelaunayTriangulation` y las rutinas dadas en la sección *Sugerencias*, implementar una función que dada una cantidad N de subdivisiones del intervalo $[0,1]$ retorne, la matriz de vértices, la matriz de elementos y el vector con la numeración de los nodos que están en Γ.

Ejercicio 2 Graficar $u(x,y)$ para $\varepsilon = 1, 0.1, 0.01, 0.001$ utilizando una malla suficientemente fina. ¿Qué se observa? *Ver sugerencias.*

Ejercicio 3 Implementar un método de elementos finitos utilizando una malla uniforme de triángulos y funciones lineales a trozos. El programa debe tomar como parámetros el valor de ε y N (cantidad de subdivisiones del intervalo $[0,1]$). Resolver para $\varepsilon = 1$ con $N = 8, 16, 32, 64$, y graficar la solución en cada caso.

Ejercicio 4 Ahora queremos calcular el error y estimar el orden de convergencia.

(a) Implementar dos funciones que calcule los errores $\|u - u_h\|_{L^2}$ y $\|u - u_h\|_{H^1}$ respectivamente. *Sugerencia:* Tener en cuenta que deberá conocer ∇u_h.

(b) Implementar una función que dado un vector con distintos valores de h (por ej, con $h = 1/N$) calcule los errores en $\|\cdot\|_{L^2}$ y $\|\cdot\|_{H^1}$. En un mismo gráfico comparar $log(h)$ vs. $log(error)$ para cada tipo de error. Además, la función deberá imprimir el orden obtenido. (*Sugerencia:* Para esto último puede ser útil la función `fit` de la librería `Polynomials` para encontrar la pendiente de la recta en ambos casos).

Ejercicio 5 Para los distintos valores de ε dados en el Ejercicio 2, volver a calcular la solución u_h y graficar cada caso ¿Qué se observa?

Sugerencias

Mallas y gráficos

Para construir y armar la matriz de vertices y de elementos pueden usar

```
using DelaunayTriangulation

tri = triangulate_rectangle(0, 1, 0, 1, N, N,delete_ghosts = true)

#matriz de nodos
points = get_points(tri)
nodes = zeros(size(points)[1],2)
for i=1:size(nodes)[1]
    nodes[i,1] = points[i][1]
    nodes[i,2] = points[i][2]
end

#matriz de elementos
triangles = get_triangles(tri)
elements = zeros(length(triangles),3)
j=1
for k in triangles
    elements[j,1],elements[j,2],elements[j,3] = k[1], k[2], k[3]
    j=j+1
end
elements = Int.(elements)
```

Para graficar las superficies se puede usar

```
using GLMakie

function plot_u(nodes,u)
    z=[u(x,y) for x in nodes[:,1], y in nodes[:,2]]
    #donde u es la solución exacta
    GLMakie.surface(nodes[:,1],nodes[:,2],z,axis=(type=Axis3,))
end

 #funcion para graficar uh
 function plot_uh(nodes,elements,uh)
    nodes3 = hcat(nodes, uh)
    fig = Figure()
    ax = Axis3(fig[1,1])
    mesh!(ax, nodes3, elements, color = uh)
    fig
end
```

Cálculo de ∇u_h

Supongamos que $P \in \mathbb{R}^{N\times 2}$ es la matriz de nodos y $T \in \mathbb{R}^{M\times 2}$ la matriz de triangulos. Se puede seguir el siguiente esquema para el calulo del gradiente. Para cada nodo i:

1. Hallar el primer triangulo t (fila de T) que contiene a i.
2. Llamar $a_1 = i$ y a_2, a_3 los otros nodos que participan en t.
3. Considerar $r_1 = a_2 - a_1$ y $r_2 = a_3 - a_1$, los vectores correspondientes a los lados de t adyacentes a i.
4. Observar que las derivadas direccionales satisfacen la siguiente propiedad:
$$\frac{\partial u_h}{\partial r_1} = \nabla u_h \cdot \frac{r_1}{\|r_1\|} \qquad \frac{\partial u_h}{\partial r_2} = \nabla u_h \cdot \frac{r_2}{\|r_2\|}$$
que pueden estiamarse de la siguiente forma:
$$\frac{\partial u_h}{\partial r_1} \approx \frac{u_h(a_2) - u_h(a_1)}{\|r_1\|} \qquad \frac{\partial u_h}{\partial r_2} \approx \frac{u_h(a_3) - u_h(a_1)}{\|r_2\|}.$$
Notar que los valores de $u_h(a_j)$ son conocidos.
5. Teniendo en cuenta lo anterior:
$$A = \begin{pmatrix} \frac{r_1}{\|r_1\|} \\ \frac{r_2}{\|r_2\|} \end{pmatrix} \qquad b = \begin{pmatrix} \frac{u_h(a_2)-u_h(a_1)}{\|r_1\|} \\ \frac{u_h(a_3)-u_h(a_1)}{\|r_2\|} \end{pmatrix}$$
Luego podemos calcular $\nabla u_h(a_1)$ resolviendo el sistema. Es decir, $\nabla u_h(a_1) = A\backslash b$.

   Puede ser conveniente guardar estos vectores de cada nodo i como filas de una nueva matriz $G \in \mathbb{R}^{N\times 2}$. Así, G tiene el mismo tamaño que P, y cada fila de G contiene el gradiente de u_h en el nodo dado por la correspondiente fila de P.