Elementi Finiti - Esercitazione 5
Il problema di Poisson con condizioni di Dirichlet

Prof. Giancarlo Sangalli   Ivan Bioli

30 Aprile 2025

1 Condizioni al bordo di Dirichlet omogenee

Ripartiamo da quanto visto durante la scorsa esercitazione. Nel caso di condizioni di Dirichlet omogenee, lo spazio $\mathrm{V}_h \subset \mathrm{V}$ è un sottospazio di $\mathrm{H}^1_0(\Omega)$ e contiene quindi funzioni nulle su $\partial\Omega$. Pertanto, si ha che

$$\mathrm{V}_h = \mathrm{Span}\{\varphi_i \,:\, \mathbf{v}_i \notin \partial\Omega\}.$$

L'assemblaggio iniziale ha incluso tutti i gradi di libertà, compresi quelli di bordo (non conviene distinguere tra indici interni e indici di bordo durante l'assemblaggio). Tuttavia, per il calcolo effettivo, è sufficiente rimuovere le righe e colonne corrispondenti ai nodi di bordo. Se indichiamo con $\tilde{\mathbf{A}}$ la matrice assemblata su tutti i nodi e con `F` e `D` rispettivamente l'insieme degli indici dei nodi liberi (cioè, in questo caso, quelli interni) e di Dirichlet (cioè, in questo caso, di bordo), dove $\mathtt{F} \cap \mathtt{D} = \emptyset$ e $\mathtt{F} \cup \mathtt{D} = \{1, \ldots, N_{\text{points}}\}$, allora la matrice risultante è la sottomatrice di $\tilde{\mathbf{A}}$ associata agli indici `F`. Usando la notazione di Julia

$$\mathbf{A}_{\text{cond}} = \tilde{\mathbf{A}}[\mathtt{F}, \mathtt{F}].$$

Esercizio 1

Determinare gli indici dei nodi interni e dei nodi di bordo. A tal fine, utilizzare la funzione `get_boundary_nodes` presente in `Meshing.jl`. Successivamente, integrare queste informazioni nella struttura della mesh utilizzando la funzione `set_dirichletdofs!`. Il codice di riferimento è il seguente:

```julia
# Build the mesh with mesh-size h
out_file = mesh_circle(h)
T, p = get_nodes_connectivity(out_file)
msh = Mesh(T, p)
# Get Dirichlet dofs
bnd_tags, bnd_coords = get_boundary_nodes(out_file)
set_dirichletdofs!(msh, bnd_tags)
```

Esercizio 2

Verificare che $\mathbf{A}_{\text{cond}} = \tilde{\mathbf{A}}[\mathtt{F},\mathtt{F}]$ ha nucleo $\{\mathbf{0}\}$.
Suggerimento: Potete usare la funzione `LinearAlgebra.nullspace`.

1.1 Test per il problema di Poisson con condizioni di Dirichlet omogenee

Esercizio 3

Consideriamo il problema di Poisson sul cerchio unitario con condizioni di Dirichlet omogenee:

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = 0 & \text{su } \partial\Omega \end{cases}. \tag{1}$$

Per il caso particolare $f = 1$, la soluzione esatta è

$$u(x, y) = \frac{1}{4}(1 - x^2 - y^2).$$

Vogliamo approssimare questa soluzione mediante il metodo degli elementi finiti e tracciare il diagramma degli errori $\|u_h - u\|_{\mathrm{L}^2(\Omega)}$ e $|u_h - u|_{\mathrm{H}^1(\Omega)}$ in funzione di h.

Parte I

Assemblare e risolvere il sistema lineare

$$\mathbf{A}_{\text{cond}}\mathbf{u}_{\mathtt{F}} = \mathbf{b}_{\text{cond}},$$

dove $\mathbf{b}_{\text{cond}} = \mathbf{b}_{\mathtt{F}} := \mathbf{b}[\mathtt{F}]$.

Parte II

Ricostruire l'intero vettore dei coefficienti

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_{\mathtt{F}} \\ \mathbf{u}_{\mathtt{D}} \end{bmatrix}, \qquad \mathbf{u}_{\mathtt{D}} = \mathbf{0}.$$

Parte III

Tracciare il grafico della soluzione approssimata u_h utilizzando le funzioni `plot_flat` e `plot_surf` in `Meshing.jl`. Dovreste ottenere dei risultati simili a quelli riportati nella Figura 1.

Parte IV

Calcolare l'errore:

1. $\max_i |u_i - u(\mathbf{v}_i)| \approx \|u_h - u\|_{\mathrm{L}^\infty(\Omega)}$
2. Approssimare $\|u_h - u\|_{\mathrm{L}^2(\Omega)}$ usando le regole di quadratura Q_0 e Q_2. Sfruttare che u_h è lineare affine sui triangoli della mesh, e dunque per valutarla in un punto $\mathbf{p} = \alpha\mathbf{v}_{i_1} + \beta\mathbf{v}_{i_2} + \gamma\mathbf{v}_{i_3}$ combinazione convessa dei vertici $\mathbf{v}_{i_1}, \mathbf{v}_{i_2}, \mathbf{v}_{i_3}$ di un triangolo della mesh è sufficiente calcolare
$$u_h(\mathbf{p}) = \alpha u_h(\mathbf{v}_{i_1}) + \beta u_h(\mathbf{v}_{i_2}) + \gamma u_h(\mathbf{v}_{i_3}) = \alpha u_{i_1} + \beta u_{i_2} + \gamma u_{i_3}.$$
3. Costruire il grafico dell'errore rispetto alla mesh size h e stimare l'ordine di accuratezza del metodo.
4. *Facoltativo:* approssimare $|u_h - u|_{\mathrm{H}^1(\Omega)}$ usando le regole di quadratura Q_0 e Q_2. Confrontare i risultati ottenuti per diversi valori di h e discutere le differenze osservate con le due regole di quadratura.

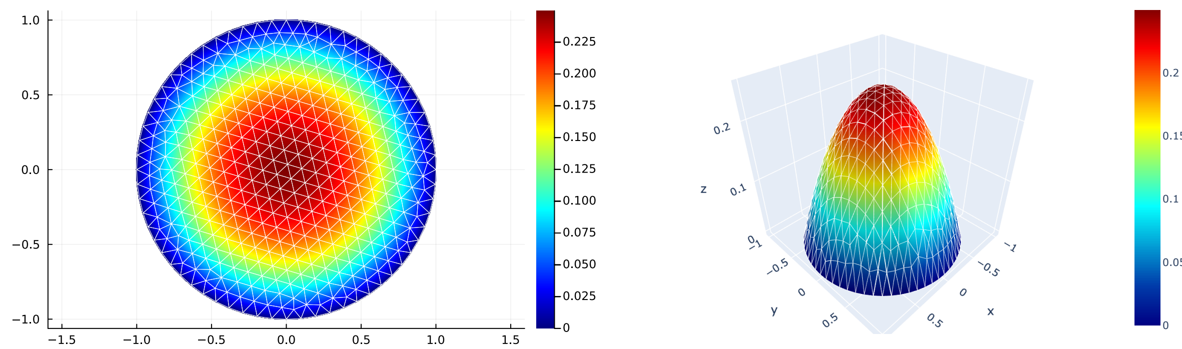

Figura 1: Plot della soluzione $u(x, y) = \frac{1}{4}(1 - x^2 - y^2)$.

2 Condizioni di Dirichlet non omogenee

Consideriamo adesso il problema di Poisson con condizioni di Dirichlet *non* omogenee

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = g & \text{su } \partial\Omega \end{cases}. \tag{2}$$

Se $u|_{\partial\Omega} = g$, possiamo "rilevare" $g : \partial\Omega \to \mathbb{R}$ all'interno del dominio costruendo un'estensione $\tilde{g} : \Omega \to \mathbb{R}$ tale che $\tilde{g}|_{\partial\Omega} = g$. Definiamo quindi una nuova incognita $\tilde{u} \coloneqq u - \tilde{g}$ che soddisfa condizioni al bordo di Dirichlet *omogenee*. Sostituendo questa espressione nel problema variazionale, otteniamo

$$\text{Trovare } \tilde{u} \text{ tale che } a(\tilde{u} + \tilde{g}, v) = \langle f, v \rangle_{\mathrm{V}' \times \mathrm{V}}, \qquad \forall v \in \mathrm{V},$$

dove sia $\tilde{u}$ che v sono nulle al bordo. Portando i termini noti al secondo membro, il problema si riscrive come:

$$\text{Trovare } \tilde{u} \text{ tale che } a(\tilde{u}, v) = \langle f, v \rangle_{\mathrm{V}' \times \mathrm{V}} - a(\tilde{g}, v), \qquad \forall v \in \mathrm{V},$$

Nel metodo agli elementi finiti, una $\tilde{g}$ può essere calcolata in modo semplice imponendo che $\tilde{g} = g$ sui nodi di bordo e $\tilde{g} = 0$ sui nodi interni. Si noti che con questa scelta, $\tilde{g}$ coincide con g su $\partial\Omega$ solo se quest'ultima è una funzione lineare a tratti; altrimenti, $\tilde{g}$ rappresenta un'interpolazione discreta di g sui nodi di bordo. Denotando con `F` l'insieme degli indici dei nodi interni e con `D` quello dei nodi di bordo, possiamo scrivere

$$\tilde{g}(\mathbf{x}) = \sum_{j \in \mathtt{D}} g(\mathbf{v}_j)\varphi_j(\mathbf{x}), \qquad \tilde{u}(\mathbf{x}) = \sum_{j \in \mathtt{F}} u_j \varphi_j(\mathbf{x}).$$

In forma matriciale, indicando $g_j = g(\mathbf{v}_j)$, si ha:

$$a(\tilde{g}, \varphi_i) = \sum_{j \in \mathtt{D}} a(\varphi_j, \varphi_i) g_j = \tilde{\mathbf{A}}_{i\mathtt{D}} \mathbf{g}_{\mathtt{D}}$$
$$a(\tilde{u}, \varphi_i) = \sum_{j \in \mathtt{F}} a(\varphi_j, \varphi_i) u_j = \tilde{\mathbf{A}}_{i\mathtt{F}} \mathbf{u}_{\mathtt{F}}.$$

Se dividiamo la matrice $\tilde{\mathbf{A}}$ a blocchi, dividendo nodi interni da quelli di bordo (e assumendo che gli indici siano ordinati mettendo prima quelli interni e poi quelli di bordo), questa assume una del tipo

$$\tilde{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{A}}_{\mathtt{FF}} & \tilde{\mathbf{A}}_{\mathtt{FD}} \\ \tilde{\mathbf{A}}_{\mathtt{DF}} & \tilde{\mathbf{A}}_{\mathtt{DD}} \end{bmatrix}.$$

A questo punto il vettore dei coefficienti $\mathbf{u}$ è $\mathbf{u} = \begin{bmatrix} \mathbf{u}_{\mathtt{F}} \\ \mathbf{u}_{\mathtt{D}} \end{bmatrix}$ dove $\mathbf{u}_{\mathtt{D}} = \mathbf{g}_{\mathtt{D}}$ e $\mathbf{u}_{\mathtt{F}}$ risolve

$$a(\tilde{g}, \varphi_i) = \sum_{j \in \mathtt{F}} a(\varphi_j, \varphi_i) u_j = \langle f, \varphi_i \rangle_{\mathrm{V}' \times \mathrm{V}} - a(\tilde{g}, \varphi_i) \qquad \forall i \in \mathtt{F},$$

cioè in forma algebrica

$$\tilde{\mathbf{A}}_{\mathtt{FF}} \mathbf{u}_{\mathtt{F}} = \mathbf{b}_{\mathtt{F}} - \tilde{\mathbf{A}}_{\mathtt{FD}} \mathbf{g}_{\mathtt{D}}.$$

Dunque, usando la notazione di Julia, nel caso di condizioni di Dirichlet non omogenee abbiamo

$$\mathbf{A}_{\text{cond}} = \tilde{\mathbf{A}}[\mathtt{F}, \mathtt{F}], \qquad \mathbf{b}_{\text{cond}} = \mathbf{b}[\mathtt{F}] - \tilde{\mathbf{A}}[\mathtt{F}, \mathtt{D}]\mathbf{g}[\mathtt{D}]. \tag{3}$$

Esercizio 4

Implementare la funzione

```
function impose_dirichlet(A, b, g, mesh)
```

che impone le condizioni di Dirichlet al sistema lineare risultante dalla discretizzazione agli elementi finiti di un problema variazionale, cioè a partire da $\tilde{\mathbf{A}}$ e $\mathbf{b}$ assembla il sistema (2). La funzione prende in input:

- `A`: la matrice di stiffness globale;
- `b`: il vettore dei termini noti globale;
- `g`: la funzione che definisce le condizioni di Dirichlet;
- `mesh`: l'oggetto mesh contenente la discretizzazione del dominio;

e restituisce:

- `A_cond`: la matrice di stiffness modificata estraendo solamente i nodi interni;
- `b_cond`: il vettore dei termini noti modificato con l'imposizione delle condizioni di Dirichlet;
- `uh`: il vettore soluzione con le condizioni di Dirichlet già applicate.

Potete trovare un codice da completare in `Assembly.jl`.

2.1 Test per il problema di Poisson con condizioni di Dirichlet non omogenee

Esercizio 5

Verificare la correttezza della propria implementazione risolvendo il problema di Poisson con condizioni non omogenee (2). Scegliere una soluzione u non polinomiale e da essa stessa definire $f = -\Delta u$ e $g = u|_{\partial\Omega}$. Diagrammare l'errore in funzione della mesh-size come nell'Esercizio 3.