

Elaborato per l'esame di Complementi di Analisi Numerica

Bruno Degli Esposti

Luglio 2019

Sommario

In questo elaborato ho cercato sia di approfondire alcuni argomenti visti a lezione, sia di esporre le conoscenze di FreeFEM che ho acquisito con il seminario della professoressa Falini, la Modelling Week di Madrid e qualche giorno di studio individuale. L'elaborato è diviso in tre parti. La prima parte consiste in un confronto tra il metodo delle differenze finite e il metodo degli elementi finiti per la soluzione dell'equazione di Laplace su una corona circolare con condizioni al bordo di Dirichlet. La seconda parte consiste nella soluzione di un problema ADR con condizioni al bordo miste in cui il termine convettivo domina su quello diffusivo. Ho confrontato l'efficacia di varie tecniche di stabilizzazione, come il metodo upwind e streamline-diffusion. La terza parte consiste nell'applicazione del θ -metodo per la soluzione dell'equazione del calore con condizioni al bordo di Robin. Il codice relativo al metodo delle differenze finite (FDM) è stato sviluppato con MATLAB, mentre quello relativo al metodo degli elementi finiti (FEM) è stato sviluppato con FreeFEM.

Indice

1	Equazione di Laplace	2
1.1	Metodo delle differenze finite	2
1.2	Metodo degli elementi finiti	3
1.3	Confronto tra FDM e FEM	5
2	Problema ADR	11
2.1	Formulazione variazionale	12
2.2	Tecniche di stabilizzazione	12
2.3	Confronto dei risultati	13
3	Equazione del calore	18
3.1	Formulazione variazionale	18
3.2	Applicazione del θ -metodo	18
3.3	Convergenza alla soluzione	19

1 Equazione di Laplace

Ho scelto il seguente problema per confrontare i due metodi:

$$\begin{cases} \Delta u(x, y) = 0 & \text{in } \Omega \subseteq \mathbb{R}^2 \\ u(x, y) = f(x, y) & \text{su } \partial\Omega \text{ regolare} \end{cases} \quad (1)$$

Il dominio Ω è una corona circolare centrata nell'origine, con raggio interno 1 e raggio esterno 2. La funzione f è armonica, così da risultare essa stessa soluzione del problema. Questo permetterà in seguito di calcolare con precisione l'errore commesso dai due metodi. Ho scelto

$$f = \operatorname{Re} \left(\frac{1}{z} \right) = \frac{x}{x^2 + y^2}$$

È immediato verificare che f sia armonica, in quanto parte reale di una funzione olomorfa su Ω .

1.1 Metodo delle differenze finite

Per poter applicare il metodo delle differenze finite, è necessario innanzitutto trovare un mapping da un rettangolo di \mathbb{R}^2 a Ω . Questo problema è in generale difficile da risolvere, ma in questo caso basta passare alle coordinate polari ρ e θ . Sia g l'espressione di f in queste coordinate, ovvero $g: (\rho, \theta) \mapsto f(\rho \cos(\theta), \rho \sin(\theta))$. L'analogo del problema (1) in coordinate polari è il seguente:

$$\begin{cases} \frac{\partial^2 v}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial v}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2 v}{\partial \theta^2} = 0 & \text{in } \Psi = (1, 2) \times (0, 2\pi) \subseteq \mathbb{R}^2 \\ v(1, \theta) = g(1, \theta) & \text{per ogni } \theta \in [0, 2\pi] \\ v(2, \theta) = g(2, \theta) & \text{per ogni } \theta \in [0, 2\pi] \\ v(\rho, 0) = g(\rho, 2\pi) & \text{per ogni } \rho \in [1, 2] \end{cases} \quad (2)$$

Dopo aver cambiato dominio, è comparso nell'equazione differenziale un termine di trasporto orizzontale in Ψ , radiale in Ω . Il rapporto tra il coefficiente di trasporto e quello di diffusione non supera 1 su tutto Ψ , pertanto non si rende necessaria l'applicazione di uno schema upwind. Sono cambiate anche le condizioni al bordo, adesso miste tra Dirichlet e periodiche.

Ora che il problema (1) è stato riformulato in modo da avere come dominio un rettangolo, si può procedere con la definizione di una griglia G su Ψ . Siano N_ρ e N_θ il numero di suddivisioni uniformi rispettivamente lungo ρ e lungo θ . Allora

$$G = \{(1 + (i - 1)/N_\rho, 2\pi(j - 1)/N_\theta) \mid i = 1, \dots, N_\rho + 1 \quad j = 1, \dots, N_\theta\}.$$

Gli indici partono da 1 e non da 0 per rispettare la convenzione di MATLAB. Siano h_ρ e h_θ i passi di discretizzazione orizzontali e verticali della griglia, cioè $1/N_\rho$ e $2\pi/N_\theta$. Dalla griglia sono stati esclusi i punti con $\theta = 2\pi$, perché la condizione al bordo periodica li rende superflui (basta usare al loro posto i punti con $\theta = 0$). In seguito sarà utile pensare a G anche come vettore di \mathbb{R}^N , dove $N = |G| = (N_\rho + 1)N_\theta$. In tal caso verrà utilizzato k come indice monodimensionale, al posto di i e j . La funzione contenuta nel file `get_k.m` permette di calcolare k a partire da i , j e N_θ .

A questo punto si può applicare il metodo delle differenze finite per trasformare il problema continuo (2) in un problema discreto della forma $A_h v_h = b_h$, ovvero un sistema lineare N -dimensionale. Una volta risolto tale sistema, l'elemento k -esimo di v_h è l'approssimazione di $v(G_k)$ fornita dal metodo. C'è da dire che in questo caso non si può applicare nessuno dei teoremi visti a lezione circa l'esistenza dell'inversa di A_h o l'uniforme limitatezza della norma di A_h^{-1} . Per fortuna, entrambe le proprietà continuano a valere, come ho potuto verificare numericamente. In particolare, per quanto riguarda la stabilità del metodo, ho ottenuto la seguente stima empirica:

$$\|A_h^{-1}\|_\infty \leq M, \quad \text{con } M \approx 1.13 \text{ per ogni } h > 0.$$

Per quanto riguarda invece la consistenza del metodo, l'errore di troncamento locale τ_h è sicuramente in $\mathcal{O}(h_\rho^2 + h_\theta^2)$, perché ho scelto di usare il metodo delle differenze finite centrate del secondo ordine e la soluzione f ha la regolarità richiesta (è addirittura analitica). Dunque mi aspetto che

$$\|e_h\|_\infty = \|v_h - v\|_\infty \in \mathcal{O}(h^2), \quad \text{con } h = \sqrt{h_\rho^2 + h_\theta^2}.$$

Fissato un punto (ρ_i, θ_j) interno a G , diciamo G_k , la k -esima equazione lineare del sistema $A_h v_h = b_h$ è

$$\begin{aligned} & \frac{v_h(\rho_{i+1}, \theta_j) - 2v_h(\rho_i, \theta_j) + v_h(\rho_{i-1}, \theta_j)}{h_\rho^2} + \frac{1}{\rho_i} \frac{v_h(\rho_{i+1}, \theta_j) - v_h(\rho_{i-1}, \theta_j)}{2h_\rho} \\ & + \frac{1}{\rho_i^2} \frac{v_h(\rho_i, \theta_{j+1}) - 2v_h(\rho_i, \theta_j) + v_h(\rho_i, \theta_{j-1})}{h_\theta^2} = 0 \end{aligned} \quad (3)$$

Se invece (ρ_i, θ_j) appartiene al bordo di G , l'equazione precedente va modificata in modo opportuno; i dettagli sono presenti all'interno del codice. Ho trovato utile scrivere l'equazione (3) anche sotto forma di tabella:

$$\sum \begin{bmatrix} \left(\frac{1}{\rho_i^2} \frac{1}{h_\theta^2}\right) v_h(\rho_i, \theta_{j+1}) \\ \left(\frac{1}{h_\rho^2} - \frac{1}{\rho_i} \frac{1}{2h_\rho}\right) v_h(\rho_{i-1}, \theta_j) \quad \left(-\frac{2}{h_\rho^2} - \frac{1}{\rho_i^2} \frac{2}{h_\theta^2}\right) v_h(\rho_i, \theta_j) \quad \left(\frac{1}{h_\rho^2} + \frac{1}{\rho_i} \frac{1}{2h_\rho}\right) v_h(\rho_{i+1}, \theta_j) \\ \left(\frac{1}{\rho_i^2} \frac{1}{h_\theta^2}\right) v_h(\rho_i, \theta_{j-1}) \end{bmatrix} = 0$$

Il codice MATLAB per la soluzione del problema (2) è contenuto nel file `differenze_finite.m`. Ho preferito imporre la condizione al bordo di Dirichlet tramite diagonalizzazione, senza eliminare gradi di libertà. Mi pare che il codice risulti così più chiaro.

1.2 Metodo degli elementi finiti

Anche per il metodo degli elementi finiti è necessario riformulare il problema (1). Occorre infatti costruire una triangolazione regolare \mathcal{T}_h di Ω e trasformare l'equazione di Laplace dalla forma forte alla forma debole. Per quanto riguarda la generazione della mesh regolare, questa viene effettuata in automatico da FreeFEM a partire da una parametrizzazione di $\partial\Omega$, in questo caso due circonferenze. Per avere un confronto significativo con il metodo delle differenze

finite, il numero di suddivisioni delle circonferenze in segmenti verrà nuovamente indicato con N_θ . A partire da queste suddivisioni e dai vertici così fissati, viene generata una triangolazione di Delaunay all'interno, con una densità di vertici proporzionale a quella sul bordo. La documentazione di FreeFEM descrive così il procedimento:

A triangulation is built by the keyword **buildmesh**. This keyword calls a triangulation subroutine based on the Delaunay test, which first triangulates with only the boundary points, then adds internal points by subdividing the edges. How fine the triangulation becomes is controlled by the size of the closest boundary edges.

Una volta generata la triangolazione \mathcal{T}_h , si può passare alla definizione di V_h . FreeFEM non usa la tecnica di rilevamento del dato sul bordo: il vincolo costituito dalle condizioni di Dirichlet viene imposto tramite penalizzazione durante la fase di algebra lineare. Pertanto all'interno del codice l'insieme V_h coincide con tutto X_h^r . Questo aspetto è messo in luce in modo inequivocabile dal fatto che, nel caso di elementi lineari, la dimensione di V_h è uguale al numero di vertici della mesh, compresi quelli sul bordo. Un'altra peculiarità di FreeFEM è che V_h , una volta definito con il comando **fespace**, diventa un nuovo tipo di dato. Così, almeno da un punto di vista sintattico, basta assegnare f a una variabile di tipo V_h per trovare la proiezione di f su V_h . Molto comodo!

Per quanto riguarda la formulazione variazionale, basta moltiplicare l'equazione di Laplace per una funzione test $v \in V = H_0^1(\Omega)$, integrare su Ω e applicare il teorema della divergenza:

$$0 = - \int_{\Omega} \Delta u v \, dx = - \int_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}} v \, d\gamma + \int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} \nabla u \cdot \nabla v \, dx = a(u, v)$$

In questo caso il funzionale lineare $F(v)$ è nullo. Indichiamo con $\{\varphi_i\}$ gli elementi della base lagrangiana di V_h . Seguendo il metodo di Galerkin, FreeFEM può ora assemblare la matrice di stiffness A_h e il vettore di carico f_h :

$$(A_h)_{ij} = a(\varphi_i, \varphi_j) \quad (f_h)_i = F(\varphi_i) = 0$$

Una volta calcolati A_h e f_h , FreeFEM apporta loro delle modifiche tramite il comando **on**, che serve a imporre le condizioni al bordo di Dirichlet tramite penalizzazione. Per ogni elemento φ_i che vale 1 su un punto base P_i appartenente al bordo della mesh, si pone

$$(A_h)_{ii} += \mathbf{tg}v, \quad (f_h)_i += f(P_i) \cdot \mathbf{tg}v, \quad \text{con } \mathbf{tg}v = 10^{30} \text{ di default.}$$

In questo modo la soluzione u_h del sistema lineare $A_h u_h = f_h$ soddisfa a precisione di macchina la condizione $(u_h)_i = f(P_i)$. Ho trovato utile a riguardo la lettura del paragrafo *A.2 Imposing essential boundary conditions*, in appendice al testo di Formaggia, Saleri, Veneziani.

La soluzione u del problema (1) è analitica, quindi $u \in H^{r+1}(\Omega)$ e vale la seguente stima dell'errore per il metodo degli elementi finiti:

$$\|u - u_h\|_{H^1} \leq \frac{M}{\alpha} Ch^r |u|_{r+1}$$

Di default, FreeFEM usa il metodo iterativo GMRES per risolvere il sistema lineare. Visto che in questo caso la matrice A_h è simmetrica definita positiva, ho

provato anche a usare il metodo dei gradienti coniugati. Non ho però ottenuto alcun beneficio, pur scegliendo adeguatamente le tolleranze. In ogni caso, il metodo iterativo scelto non usa preconditionamento.

1.3 Confronto tra FDM e FEM

Le figure 4 e 5 mostrano l'aspetto delle soluzioni numeriche ottenute con i due metodi per $N_\theta = 126$. Purtroppo il software di visualizzazione incluso in FreeFEM ha delle serie limitazioni, ovvero non permette l'output in formato vettoriale o l'ingrandimento della legenda che riporta i valori delle curve di livello. Le tabelle 1, 2 e 3 indicano in dettaglio i risultati ottenuti con i vari metodi, rispettivamente differenze finite del second'ordine, elementi finiti lineari ed elementi finiti quadratici. La colonna t_a riporta il numero di secondi richiesto per l'assemblaggio del sistema lineare, includendo nel caso degli elementi finiti il costo di generazione della mesh. La colonna t_s riporta invece il tempo richiesto dalla fase di algebra lineare. Nelle tabelle 1 e 2, il numero di vertici della mesh e la dimensione del sistema lineare coincidono. Ho riportato pertanto solo quest'ultimo valore. Nella tabella 3 non ho riportato le informazioni su h , perché coincidono con quelle della tabella 2. Infatti i due metodi FE, anche se di ordine diverso, usano la stessa mesh a parità di N_θ . Un esempio di tale mesh per $N_\theta = 63$ è riportato in figura 3b. Per quanto riguarda il metodo delle differenze finite, la figura 3a mostra la griglia G a cui è stato applicato il mapping delle coordinate polari.

Ottenere anche solo una cifra significativa corretta nelle stime numeriche degli errori per il metodo degli elementi finiti si è rivelato veramente difficile. Alla fine, ho ottenuto risultati soddisfacenti solo nel seguente modo:

- Ho generato una triangolazione di riferimento \mathcal{T}_{ref} molto fitta.
- Ho proiettato u su V_{ref} , spazio degli elementi P4 su \mathcal{T}_{ref} .
- Ho elevato u_h da V_h a V_{ref} .
- Ho calcolato $\|u_h - u\|$ usando formule di quadratura di ordine 10.

I dettagli sono presenti all'interno del codice. Senza questi accorgimenti, cioè lavorando direttamente in V_h , ho ottenuto stime fino a due ordini di grandezza più piccole. Talvolta gli errori non avevano nemmeno un andamento monotono rispetto a h . Dall'analisi dei dati riportati, emergono le seguenti considerazioni:

- Nel codice MATLAB, il tempo t_a richiesto per l'assemblaggio aumenta con il quadrato del numero di vertici della mesh, il che significa che ho scritto un programma poco efficiente. La colpa va attribuita senz'altro al modo in cui viene costruita A_h . Ho preallocato in memoria una matrice sparsa CSC con `spalloc()` e poi ho assegnato tutti i valori v alle entrate non nulle con la sintassi `Ah(k,1) = v`. La documentazione di MATLAB suggerisce invece di costruire prima i tre vettori della rappresentazione COO di A_h , poi di chiamare la funzione `sparse()`, con cui si ottiene la forma CSC.
- Nel codice FreeFEM, il tempo t_a richiesto per l'assemblaggio aumenta linearmente con il numero di vertici della mesh, il che è ottimale. Nel caso di elementi lineari, la fase di assemblaggio costa più della soluzione del

sistema lineare (di un 25% circa). Nel caso di elementi quadratici, è la fase di algebra lineare a costare di più (oltre il doppio).

- Per il metodo delle differenze finite, l'andamento asintotico della norma infinito dell'errore è effettivamente $\mathcal{O}(h^2)$, in accordo con la teoria. Si osservi a riguardo la figura 1. Nella stessa figura si può vedere come anche il metodo degli elementi finiti presenti il medesimo andamento asintotico dell'errore, sia per $r = 1$ che per $r = 2$. In questo caso non conosco risultati teorici di riferimento.

Per il metodo delle differenze finite, l'errore è calcolato in modo puntuale sui vertici della griglia G . Per il metodo FEM, invece, ho cercato di dare una stima su tutto \mathcal{T}_h . Tale stima è circa 10 volte più grande di quella che avrei potuto ottenere confrontando i valori solo sui punti base della triangolazione. A bilanciare questa scelta, riducendo le stime, c'è il fatto che nel grafico 1 ho riportato i valori h_{max} delle mesh, e non delle ipotetiche medie h_{avg} .

In conclusione, direi che i metodi FD e FEM con elementi quadratici forniscono soluzioni della stessa qualità, per quel che riguarda l'errore in norma infinito. Il metodo FD richiede però la soluzione di un sistema lineare più piccolo, quindi in questo caso mi sembra preferibile (si confrontino le colonne t_s).

- Per il metodo degli elementi finiti, l'andamento asintotico della norma H^1 dell'errore è effettivamente $\mathcal{O}(h^r)$, in accordo con la teoria. Si osservi a riguardo la figura 2. Nella stessa figura si può vedere anche l'andamento della norma L^2 dell'errore. Trovo interessante il fatto che nel caso di elementi finiti quadratici, $\|e\|_{L^2}$ vada a zero come $h^{2.5}$, anziché h^3 .
- La famiglia \mathcal{F} di triangolazioni generata da FreeFEM è quasi-uniforme, perché il rapporto tra h_{min} e h_{max} è limitato inferiormente da 0.26 circa.

Tabella 1: Metodo delle differenze finite

N_θ	h	$\ e\ _\infty$	$\ e\ _{L^2}$	$\ e\ _{H^1}$	$\dim(A_h)$	t_a	t_s
63	1.41e-1	2.57e-4	-	-	693	2.5e-3	1.6e-3
88	1.01e-1	1.32e-4	-	-	1320	5.3e-3	3.0e-3
126	7.06e-2	6.52e-5	-	-	2646	1.4e-2	7.9e-3
176	5.05e-2	3.33e-5	-	-	5104	3.8e-2	1.7e-2
251	3.53e-2	1.63e-5	-	-	10291	1.2e-1	3.8e-2
358	2.48e-2	8.05e-6	-	-	20764	4.1e-1	7.3e-2
503	1.77e-2	4.09e-6	-	-	40743	1.5	1.6e-1
710	1.25e-2	2.05e-6	-	-	80940	7.1	3.6e-1

Tabella 2: Metodo degli elementi finiti P1

N_θ	h_{max}	$\frac{h_{min}}{h_{max}}$	$\ e\ _\infty$	$\ e\ _{L^2}$	$\ e\ _{H^1}$	$\dim(A_h)$	t_a	t_s
63	2.82e-1	0.30	5.30e-3	3.16e-3	1.48e-1	510	7.0e-3	7.0e-3
88	2.00e-1	0.30	2.56e-3	1.57e-3	1.04e-1	983	1.3e-2	9.0e-3
126	1.41e-1	0.27	1.24e-3	7.75e-4	7.25e-2	1986	2.7e-2	1.7e-2
176	1.02e-1	0.26	6.72e-4	4.05e-4	5.22e-2	3705	4.9e-2	3.3e-2
251	7.04e-2	0.26	3.25e-4	1.96e-4	3.60e-2	7695	1.0e-1	7.8e-2
358	4.95e-2	0.27	1.52e-4	9.63e-5	2.52e-2	15138	2.1e-1	1.6e-1
503	3.54e-2	0.27	8.57e-5	4.68e-5	1.73e-2	32517	4.7e-1	3.8e-1
710	2.52e-2	0.27	3.91e-5	2.52e-5	1.27e-2	57982	9.7e-1	7.0e-1

Tabella 3: Metodo degli elementi finiti P2

N_θ	$\ e\ _\infty$	$\ e\ _{L^2}$	$\ e\ _{H^1}$	vertici	$\dim(A_h)$	t_a	t_s
63	6.22e-4	1.27e-4	2.71e-2	505	1894	9.0e-3	1.7e-2
88	3.19e-4	4.54e-5	1.39e-2	983	3756	1.7e-2	3.4e-2
126	1.55e-4	1.60e-5	6.73e-3	1986	7692	3.5e-2	7.4e-2
176	7.90e-5	6.41e-6	3.43e-3	3705	14468	6.1e-2	1.5e-1
251	3.92e-5	2.41e-6	1.66e-3	7695	30278	1.3e-1	3.5e-1
358	1.89e-5	9.83e-7	7.92e-4	15138	59836	2.7e-1	7.8e-1
503	9.75e-6	4.12e-7	3.78e-4	32517	129062	6.9e-1	1.8
710	4.87e-6	1.77e-7	1.77e-4	57982	230508	1.3	3.5

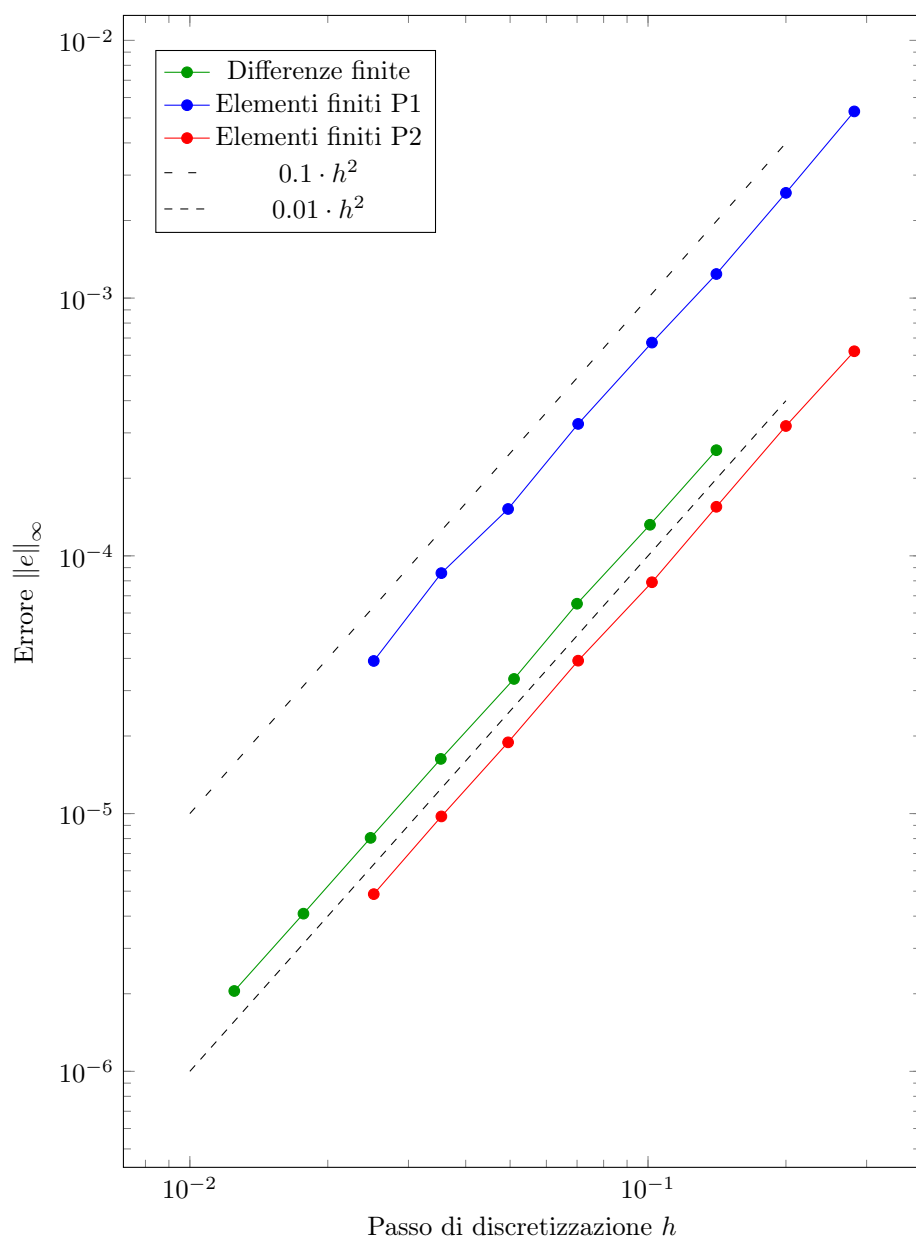


Figura 1: Confronto degli errori in norma infinito.

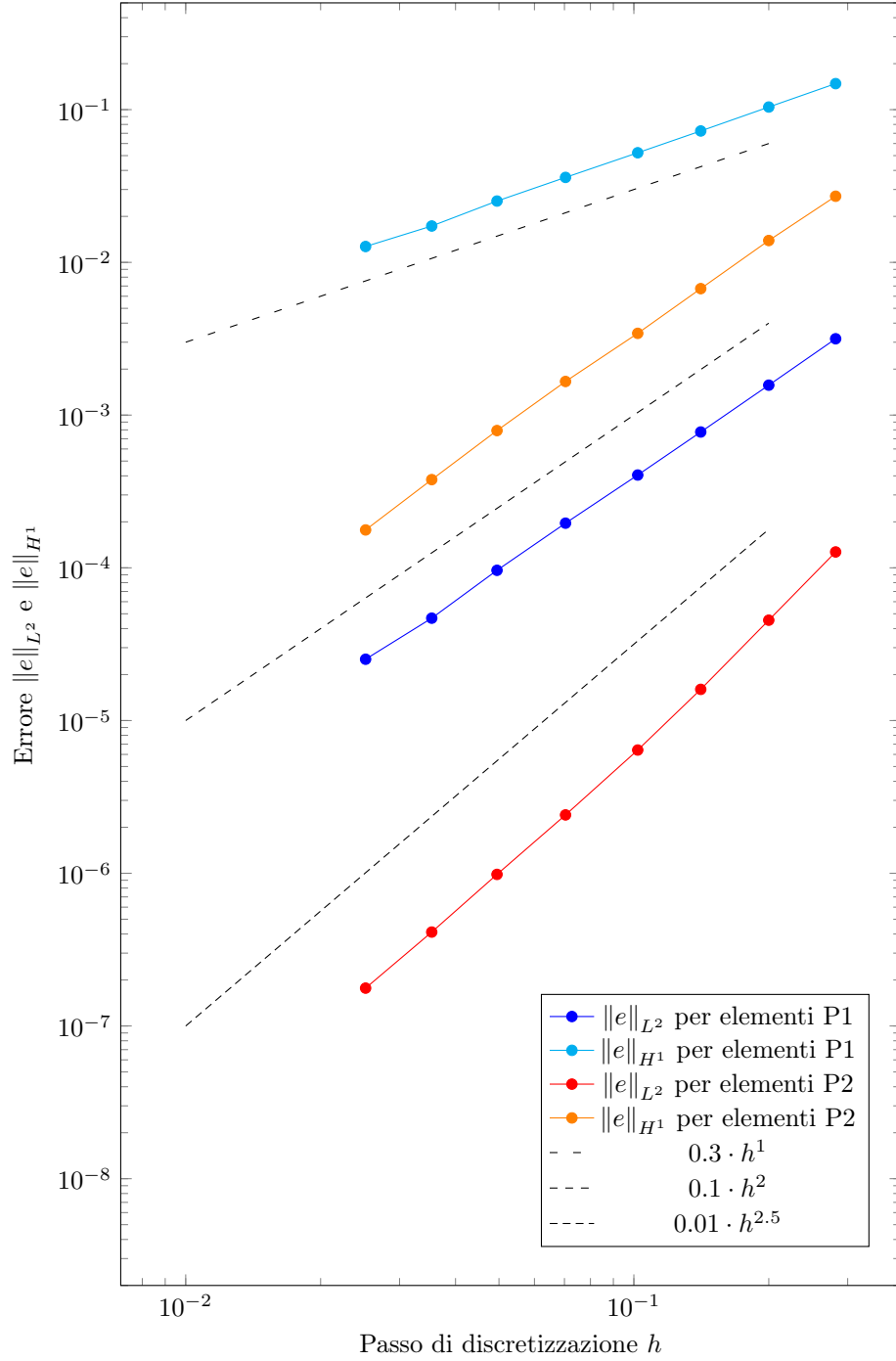


Figura 2: Confronto degli errori in norma L^2 e H^1 .

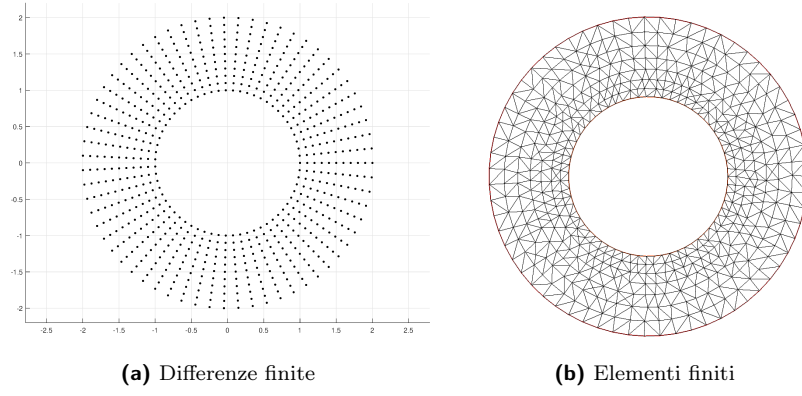


Figura 3: Mesh a confronto con $N_\theta = 10$.

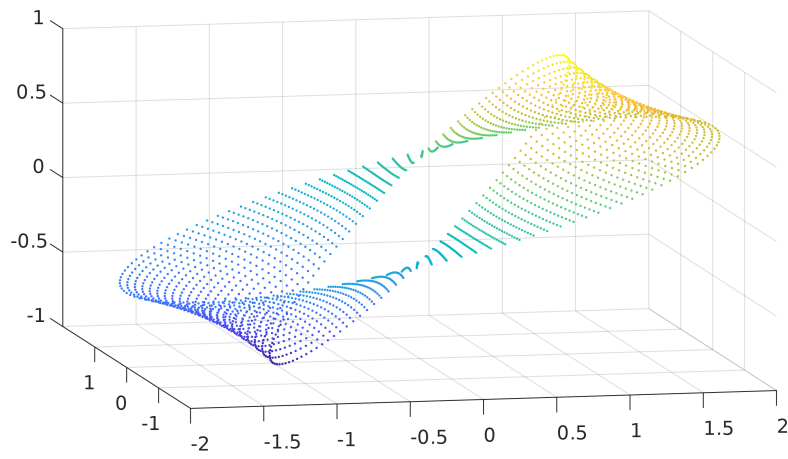


Figura 4: Soluzione FDM, con $N_\theta = 20$.

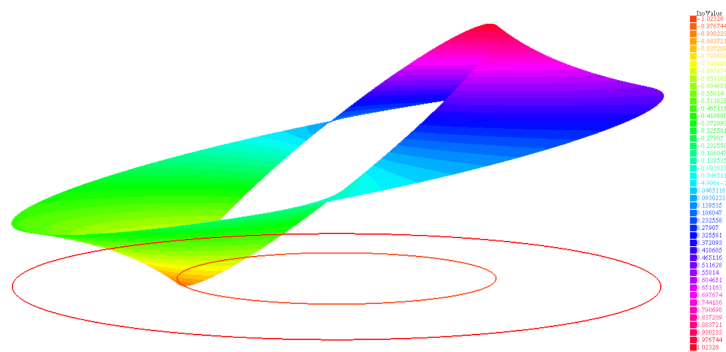


Figura 5: Soluzione FEM P2, con $N_\theta = 20$.

2 Problema ADR

Si consideri il seguente problema di diffusione-trasporto-reazione:

$$\begin{cases} -\mu\Delta u + \mathbf{b} \cdot \nabla u + \sigma u = 0 & \text{in } \Omega \subseteq \mathbb{R}^2 \\ u = 1 & \text{su } \Gamma_1 \subseteq \partial\Omega \\ \frac{\partial u}{\partial \mathbf{n}} = 0 & \text{su } \Gamma_2 \subseteq \partial\Omega \\ u = -1 & \text{su } \Gamma_3 \subseteq \partial\Omega \\ \frac{\partial u}{\partial \mathbf{n}} = 0 & \text{su } \Gamma_4 \subseteq \partial\Omega \end{cases} \quad (4)$$

Il dominio Ω è un trapezio con vertici $\{(-1, 3), (-1, -3), (1, -1), (1, 1)\}$. Le porzioni di bordo $\Gamma_1, \dots, \Gamma_4$ corrispondono a lati del trapezio come in figura.

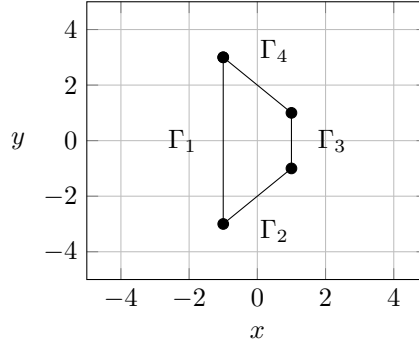


Figura 6: Il dominio Ω e il suo bordo.

I coefficienti μ , \mathbf{b} e σ sono costanti, per semplicità. All'interno del codice, queste costanti sono state scelte in modo da garantire esistenza e unicità della soluzione. Sono anche state scelte affinché si formi uno strato limite nei pressi di Γ_3 , a causa del termine di trasporto \mathbf{b} che domina su quello di diffusione μ . Per il problema (4) non ho previsto nessuna soluzione in forma chiusa con cui confrontare i risultati numerici ottenuti. Ho cercato pertanto di calcolare con FreeFEM una soluzione numerica di riferimento con la più alta accuratezza possibile. A tal fine, ho trovato utile la funzione `adaptmesh()`, che permette di rendere la mesh più fitta in corrispondenza dello strato limite.

2.1 Formulazione variazionale

Siano $\Gamma_D = \Gamma_1 \cup \Gamma_3$, $\Gamma_N = \Gamma_2 \cup \Gamma_4$ e $v \in H_D^1(\Omega)$. Allora si ha che

$$\begin{aligned}
& - \int_{\Omega} \mu \Delta u v \, dx + \int_{\Omega} (\mathbf{b} \cdot \nabla u) v \, dx + \int_{\Omega} \sigma u v \, dx = 0 \\
& - \int_{\partial\Omega} \mu \frac{\partial u}{\partial \mathbf{n}} v \, d\gamma + \int_{\Omega} \mu \nabla u \cdot \nabla v \, dx + \int_{\Omega} (\mathbf{b} \cdot \nabla u) v \, dx + \int_{\Omega} \sigma u v \, dx = 0 \\
& - \int_{\Gamma_D} \mu \frac{\partial u}{\partial \mathbf{n}} v \, d\gamma - \int_{\Gamma_N} \mu \frac{\partial u}{\partial \mathbf{n}} v \, d\gamma + \int_{\Omega} \mu \nabla u \cdot \nabla v \, dx + \int_{\Omega} (\mathbf{b} \cdot \nabla u) v \, dx + \int_{\Omega} \sigma u v \, dx = 0 \\
& - \int_{\Gamma_D} \mu \frac{\partial u}{\partial \mathbf{n}} 0 \, d\gamma - \int_{\Gamma_N} \mu 0 v \, d\gamma + \int_{\Omega} \mu \nabla u \cdot \nabla v \, dx + \int_{\Omega} (\mathbf{b} \cdot \nabla u) v \, dx + \int_{\Omega} \sigma u v \, dx = 0 \\
& a(u, v) - F(v) = \int_{\Omega} \mu \nabla u \cdot \nabla v \, dx + \int_{\Omega} (\mathbf{b} \cdot \nabla u) v \, dx + \int_{\Omega} \sigma u v \, dx = 0
\end{aligned}$$

2.2 Tecniche di stabilizzazione

La soluzione ottenuta con il metodo di Galerkin standard

$$\text{Calcolare } u_h \text{ tale che } a(u_h, v_h) = F(v_h) \text{ per ogni } v_h \in V_h$$

può presentare oscillazioni spurie, se il numero di Peclet locale

$$Pe = \frac{\|\mathbf{b}\|_2 h}{2\mu}$$

è maggiore di 1. Una rimedio consiste nello scegliere $h < 2\mu / \|\mathbf{b}\|_2$, ma se $\|\mathbf{b}\|_2 \gg \mu$ il costo associato al calcolo della soluzione può essere proibitivo. A lezione sono state presentate due tecniche che permettono di recuperare il corretto andamento qualitativo della soluzione senza dover ridurre h : la *diffusione artificiale* e la *streamline-diffusion*. Il fondamento teorico di entrambe le tecniche consiste nel metodo di Galerkin generalizzato

$$\begin{aligned}
& \text{Calcolare } u_h \text{ tale che } a(u_h, v_h) + b_h(u_h, v_h) \\
& = F(v_h) + G_h(v_h) \text{ per ogni } v_h \in V_h
\end{aligned}$$

in cui $b_h(u_h, v_h)$ e $G_h(v_h)$ sono detti *termini di stabilizzazione*. Nel caso della diffusione artificiale upwind, si ha

$$b_h(u_h, v_h) = \mu Pe \int_{\Omega} \nabla u_h \cdot \nabla v_h \, dx \quad G_h(v_h) = 0$$

Questa scelta risulta equivalente all'aggiunta di $-\mu Pe \Delta u$ nell'equazione differenziale associata al problema (4). L'ordine di convergenza che si ottiene è $\mathcal{O}(h)$ indipendentemente da r . La diffusione artificiale di tipo Scharfetter-Gummel è analoga alla precedente, con la differenza che al numero di Peclet locale viene applicata la funzione

$$\varphi(t) = t - 1 + B(2t), \quad \text{con } B(t) = \begin{cases} \frac{t}{e^t - 1} & \text{se } t > 0 \\ 1 & \text{se } t = 0 \end{cases}$$

L'ordine di convergenza migliora e diventa $\min\{r, 2\}$. Sia $\hat{\mathbf{b}}$ il vettore \mathbf{b} normalizzato. Nel caso della streamline-diffusion, si pone

$$b_h(u_h, v_h) = -\mu Pe \int_{\Gamma_N} (\nabla u_h \cdot \hat{\mathbf{b}})(\mathbf{n} \cdot \hat{\mathbf{b}}) v_h d\gamma + \mu Pe \int_{\Omega} (\nabla u_h \cdot \hat{\mathbf{b}})(\nabla v_h \cdot \hat{\mathbf{b}}) dx$$

Questa scelta corrisponde all'aggiunta di $-\mu Pe \operatorname{div}((\nabla u \cdot \hat{\mathbf{b}}) \hat{\mathbf{b}})$ nell'equazione differenziale associata al problema (4). Per dimostrarlo, basta ripetere il procedimento utilizzato per passare alla formulazione variazionale:

$$\begin{aligned} - \int_{\Omega} \mu \Delta u v dx - \int_{\Omega} \mu Pe \operatorname{div}((\nabla u \cdot \hat{\mathbf{b}}) \hat{\mathbf{b}}) v dx + \int_{\Omega} (\mathbf{b} \cdot \nabla u) v dx + \int_{\Omega} \sigma u v dx &= 0 \\ \cdots - \mu Pe \int_{\partial\Omega} (\nabla u \cdot \hat{\mathbf{b}})(\mathbf{n} \cdot \hat{\mathbf{b}}) v d\gamma + \mu Pe \int_{\Omega} (\nabla u \cdot \hat{\mathbf{b}})(\nabla v \cdot \hat{\mathbf{b}}) dx + \cdots &= 0 \\ \cdots - \mu Pe \int_{\Gamma_N} (\nabla u \cdot \hat{\mathbf{b}})(\mathbf{n} \cdot \hat{\mathbf{b}}) v d\gamma + \mu Pe \int_{\Omega} (\nabla u \cdot \hat{\mathbf{b}})(\nabla v \cdot \hat{\mathbf{b}}) dx + \cdots &= 0 \end{aligned}$$

L'ordine di convergenza che si ottiene è $\mathcal{O}(h)$ indipendentemente da r , ma in linea di principio aver aggiunto viscosità solamente nella direzione di \mathbf{b} dovrebbe portare a soluzioni più accurate rispetto alla tecnica upwind. Anche per la streamline-diffusion si può scegliere di applicare φ a Pe , portando così l'ordine di convergenza a $\min\{r, 2\}$.

2.3 Confronto dei risultati

La figura 7 permette di confrontare l'aspetto qualitativo di alcune soluzioni numeriche che ho ottenuto con i programmi `adr_galerkin_standard.edp`, `adr_upwind.edp` e `adr_streamline_diffusion.edp`. Ho scelto i seguenti parametri per il problema:

$$\mu = 1 \quad \mathbf{b} = (30, 0) \quad \sigma = 1$$

Si noti come il metodo di Galerkin standard presenti forti oscillazioni nei pressi dello strato limite. La soluzione ottenuta con il metodo della viscosità artificiale upwind presenta invece un andamento monotono, sicuramente più vicino a quello della soluzione numerica di riferimento (che ai fini della stima dell'errore viene considerata come se fosse esatta).

Purtroppo, non sono riuscito a replicare i risultati di convergenza teorici che ho riassunto nella sottosezione precedente. In particolare, ho notato che la qualità della soluzione migliora in modo marginale nel passaggio da elementi finiti lineari a elementi finiti quadratici. Sospetto quindi che la soluzione debole u di questo problema ADR non abbia la regolarità $H^2(\Omega)$ richiesta dalle stime di interpolazione alla base dei risultati teorici. La mancanza di regolarità potrebbe essere dovuta alle condizioni al bordo miste, o alla geometria particolare di Ω con due angoli interni di 135° .

Le tabelle 4, 5 e 6 indicano in dettaglio i risultati ottenuti con il metodo degli elementi finiti P1 al variare delle tecniche di stabilizzazione. La colonna superiore riporta il valore massimo assunto dalla funzione $u_h - u$. Delle tante quantità possibili per rilevare la presenza di oscillazioni nella soluzione, questa è risultata la più affidabile tra quelle che ho provato. Dall'analisi dei dati riportati, emergono le seguenti considerazioni:

- Quando il numero di Peclet locale è grande, sono i metodi stabilizzati a presentare l'errore più piccolo. Quando invece Pe scende sotto una certa soglia (intorno a 3), il metodo di Galerkin standard torna a essere più accurato. Le oscillazioni non scompaiono del tutto (si osservi la colonna superiore), ma si smorzano abbastanza da confondersi con il restante profilo dell'errore.
- L'ordine di convergenza del metodo risulta inferiore a 1, ammesso che la soluzione di riferimento sia attendibile. Si vede bene nella figura 8 come le curve degli errori tendano ad allontanarsi dalla retta di riferimento tratteggiata, invece che proseguire in parallelo.
- Per questo problema (in particolare direi per questo dominio Ω) la tecnica di streamline-diffusion sembra non apportare alcun tipo di beneficio rispetto alla diffusione artificiale upwind. Anzi, il sistema lineare associato risulta leggermente più costoso da risolvere.

Tabella 4: Metodo degli elementi finiti P1 standard

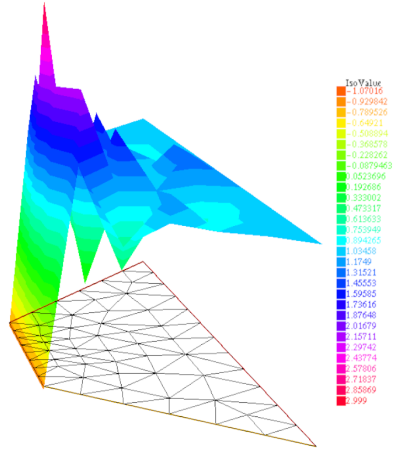
h	Pe	$\ e\ _{H^1}$	$\sup e$	$\dim(A_h)$	t_a	t_s
5.00e-1	7.50	1.22e1	1.97	67	1.0e-3	2.0e-3
3.54e-1	5.30	1.05e1	1.38	137	2.0e-3	1.0e-3
2.50e-1	3.75	9.12	1.07	230	3.0e-3	2.0e-3
1.77e-1	2.65	6.92	5.92e-1	511	5.0e-3	7.0e-3
1.25e-1	1.87	5.48	4.10e-1	892	9.0e-3	1.1e-2
8.84e-2	1.33	4.19	2.24e-1	1599	1.6e-2	2.1e-2
6.25e-2	9.37e-1	3.08	1.28e-1	3305	3.3e-2	4.5e-2
4.42e-2	6.63e-1	2.29	6.14e-2	6856	7.3e-2	1.0e-1
3.12e-2	4.69e-1	1.69	3.97e-2	13167	1.6e-1	2.2e-1
2.21e-2	3.31e-1	1.28	1.84e-2	25765	3.1e-1	4.2e-1
1.56e-2	2.34e-1	9.30e-1	8.52e-3	52104	6.8e-1	9.2e-1
1.10e-2	1.66e-1	7.04e-1	4.33e-3	104038	1.5	2.0

Tabella 5: Metodo degli elementi finiti P1 con viscosità artificiale upwind

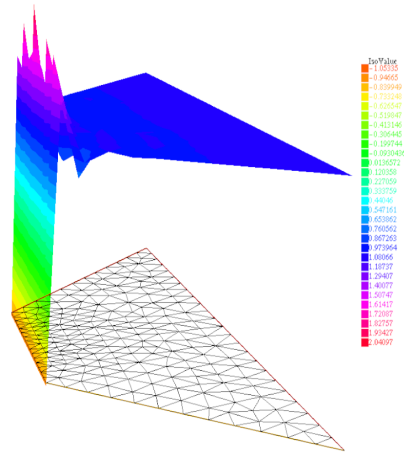
h	$\ e\ _{H^1}$	$\sup e$	$\dim(A_h)$	t_a	t_s
5.00e-1	9.77	6.13e-3	67	1.0e-3	2.0e-3
3.54e-1	9.19	4.97e-3	137	2.0e-3	1.0e-3
2.50e-1	8.58	3.92e-3	230	3.0e-3	2.0e-3
1.77e-1	7.64	2.88e-3	511	6.0e-3	6.0e-3
1.25e-1	6.73	2.09e-3	892	9.0e-3	1.1e-2
8.84e-2	5.73	1.50e-3	1599	1.7e-2	2.2e-2
6.25e-2	4.71	1.06e-3	3305	3.5e-2	4.8e-2
4.42e-2	3.78	7.54e-4	6856	7.2e-2	1.0e-1
3.12e-2	2.96	5.33e-4	13167	1.4e-1	2.0e-1
2.21e-2	2.28	3.76e-4	25765	2.9e-1	4.2e-1
1.56e-2	1.71	2.65e-4	52104	6.5e-1	9.0e-1
1.10e-2	1.28	1.87e-4	104038	1.5	2.0

Tabella 6: Metodo degli elementi finiti P1 con streamline-diffusion

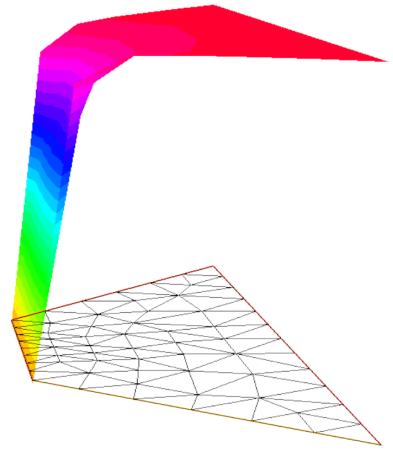
h	$\ e\ _{H^1}$	$\sup e$	$\dim(A_h)$	t_a	t_s
5.00e-1	9.77	3.60e-4	67	1.0e-3	2.0e-3
3.54e-1	9.21	1.12e-3	137	1.0e-3	2.0e-3
2.50e-1	8.60	1.11e-3	230	2.0e-3	4.0e-3
1.77e-1	7.68	9.13e-4	511	5.0e-3	9.0e-3
1.25e-1	6.77	7.37e-4	892	9.0e-3	1.6e-2
8.84e-2	5.77	5.77e-4	1599	1.7e-2	2.9e-2
6.25e-2	4.75	4.40e-4	3305	3.3e-2	6.2e-2
4.42e-2	3.82	3.30e-4	6856	7.4e-2	1.3e-1
3.12e-2	2.99	2.44e-4	13167	1.4e-1	2.7e-1
2.21e-2	2.31	1.78e-4	25765	2.9e-1	5.4e-1
1.56e-2	1.73	1.28e-4	52104	6.4e-1	1.1
1.10e-2	1.30	9.23e-5	104038	1.5	2.6



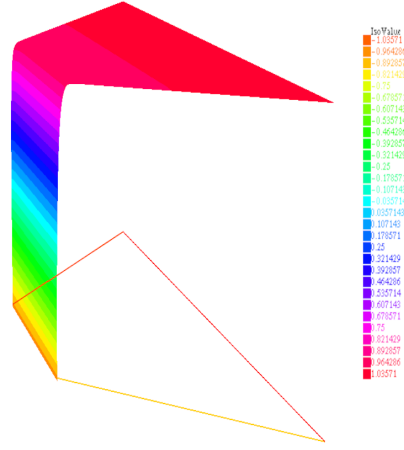
(a) Galerkin standard con $h = 0.5$



(b) Galerkin standard con $h = 0.25$



(c) Viscosità artificiale upwind con $h = 0.5$



(d) Soluzione di riferimento

Figura 7: Aspetto delle soluzioni a confronto.

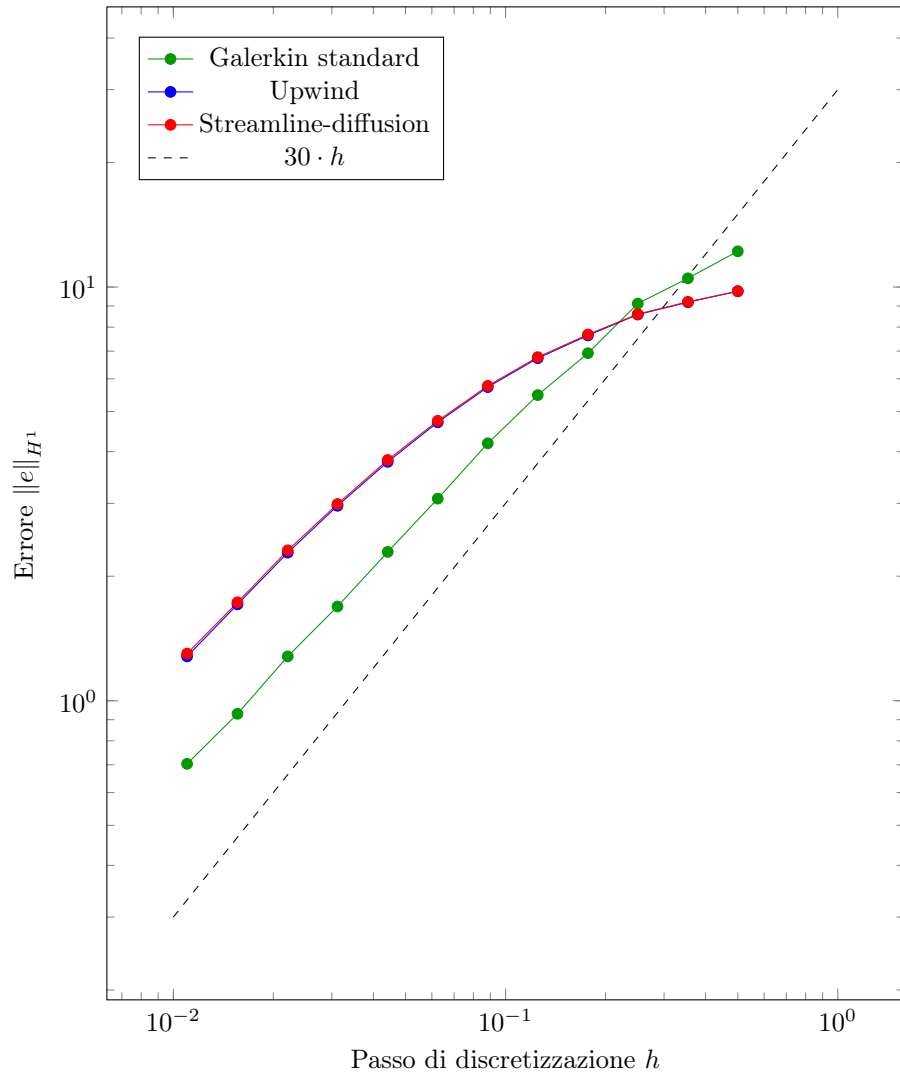


Figura 8: Confronto degli errori in norma H^1 .

3 Equazione del calore

Si consideri la seguente equazione del calore con condizioni al bordo di Robin:

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u = f & \text{in } \Omega \times (0, +\infty) \\ 2u + \frac{\partial u}{\partial \mathbf{n}} = 0 & \text{su } \partial\Omega \times (0, +\infty) \\ u(x, y, 0) = -x^2 - y^2 + 2 & \text{in } \Omega \subseteq \mathbb{R}^2 \end{cases} \quad (5)$$

Il dominio Ω è un disco di raggio 1 centrato nell'origine. La funzione f è stata scelta affinché la soluzione u del problema sia

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

Si dimostra facilmente che

$$f(x, y, t) = \frac{4t + x^2 + y^2 + 2}{(t+1)^2}$$

e che u soddisfa le condizioni al bordo di Robin per ogni $t > 0$.

3.1 Formulazione variazionale

Sia $v \in H^1(\Omega)$. Allora si ha che

$$\begin{aligned} \int_{\Omega} \frac{\partial u}{\partial t} v \, dx - \int_{\Omega} \Delta u v \, dx &= \int_{\Omega} f v \, dx \\ \int_{\Omega} \frac{\partial u}{\partial t} v \, dx - \int_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}} v \, d\gamma + \int_{\Omega} \nabla u \cdot \nabla v \, dx &= \int_{\Omega} f v \, dx \\ \int_{\Omega} \frac{\partial u}{\partial t} v \, dx + \int_{\partial\Omega} 2uv \, d\gamma + \int_{\Omega} \nabla u \cdot \nabla v \, dx &= \int_{\Omega} f v \, dx \\ \int_{\Omega} \frac{\partial u}{\partial t} v \, dx + a(u, v) &= F(v) \end{aligned}$$

A questo punto si può applicare il metodo di Galerkin (con le stesse modalità viste a lezione) per ottenere il sistema di equazioni differenziali ordinarie

$$M\mathbf{U}'(t) + A\mathbf{U}(t) = \mathbf{f}(t) \quad (6)$$

L'incognita $\mathbf{U}(t)$ è il vettore dei coefficienti di $u_h(t)$ nella sua scrittura rispetto alla base lagrangiana di V_h . Stavolta la matrice A contiene anche il contributo dato dall'integrale di $2uv$ sul bordo della mesh. Questo contributo aumenta la coercività del problema e non cambia il fatto che A sia SDP, quindi non lo reputo problematico. La questione meriterebbe però di essere approfondita.

3.2 Applicazione del θ -metodo

Sia Δt il passo temporale scelto per la soluzione dell'equazione differenziale (6). Indichiamo con \mathbf{U}^k l'approssimazione di $\mathbf{U}(t)$ calcolata al passo k -esimo, cioè relativa all'istante $k\Delta t$. Sia \mathbf{f}^k il vettore dei valori $F(\varphi_i)$ calcolati all'istante $k\Delta t$ mediante formule di quadratura. Gli integrali su \mathcal{T}_h vengono approssimati di

default con una formula di quadratura gaussiana (**qf5pT**) che usa 7 punti su ogni triangolo. La formula ha ordine 6, quindi è esatta su polinomi di grado al più 5.

Sia $\theta \in [0, 1]$. Il cosiddetto θ -metodo consiste nel seguente schema di avanzamento temporale:

$$(M + \Delta t \theta A) \mathbf{U}^{k+1} = (M - \Delta t (1 - \theta) A) \mathbf{U}^k + (\theta \mathbf{f}^{k+1} + (1 - \theta) \mathbf{f}^k) \Delta t$$

$$\text{o, più sinteticamente,} \quad S \mathbf{U}^{k+1} = T \mathbf{U}^k + (\theta \mathbf{f}^{k+1} + (1 - \theta) \mathbf{f}^k) \Delta t$$

Il costo di ogni iterazione è dominato dal calcolo della soluzione del sistema lineare $S \mathbf{U}^{k+1} = \mathbf{y}^k$. Dato che la matrice S rimane costante a ogni passo, risulta conveniente fattorizzarla una volta per tutte all'inizio del codice. A tal fine, FreeFEM mette a disposizione una versione della fattorizzazione di Cholesky in grado di preservare il più possibile la struttura sparsa di S . Questo approccio si rivela più veloce che risolvere a ogni passo un sistema lineare con un metodo iterativo, che sia GMRES o CG. Introduciamo

$$u_h^k = \sum_{i=0}^{N_h} (\mathbf{U}^k)_i \varphi_i \in V_h, \quad p(\theta) = \begin{cases} 4 & \text{se } \theta = 1/2 \\ 2 & \text{altrimenti} \end{cases}$$

Per quanto riguarda la convergenza di u_h^k alla soluzione esatta $u(\cdot, k\Delta t)$, si ha la seguente stima:

$$\|u_h^k - u(\cdot, k\Delta t)\|_{L^2}^2 + 2\alpha \Delta t \sum_{j=1}^k \|u_h^j - u(\cdot, j\Delta t)\|_{H^1}^2 \leq G(u(\cdot, 0), f) \left(\Delta t^{p(\theta)} + h^{2r} \right)$$

A partire da questa, ne ho ricavata una più semplice:

$$\|e_h^k\|_{H^1} = \|u_h^k - u(\cdot, k\Delta t)\|_{H^1} \leq C \sqrt{\Delta t^{p(\theta)} + h^{2r}}$$

Per mancanza di tempo, ho verificato numericamente solo quest'ultima con i dati raccolti dall'esecuzione del programma **calore.edp**. Durante la scrittura di tale programma, mi sono imbattuto in un bug di FreeFEM piuttosto serio: le espressioni della forma $A - cB$ con A e B matrici sparse non vengono valutate correttamente. Per l'appunto, T è di questa forma. Ho provveduto a segnalare il bug agli sviluppatori di FreeFEM, che nel giro di un giorno hanno risolto il problema (il ticket, ancora aperto, si trova all'indirizzo <https://github.com/FreeFem/FreeFem-sources/issues/102>). Per arrivare a capire perché il programma non funzionasse (un valore errato di T finiva per dare soluzioni addirittura negative!) ho trovato molto utile scrivere del codice FreeFEM per trasferire le matrici M , A e T in MATLAB. Tale codice si trova nel file **calore_export_matrix.m**. Quando ho visto che la simulazione dava un risultato corretto in MATLAB, ho capito che la colpa era di FreeFEM.

A parte questa disavventura, un altro buon motivo per trasferire le matrici in MATLAB è che FreeFEM non ha funzionalità equivalenti a **spy()**, **condest()** o **normest()**, molto utili quando si lavora con matrici sparse.

3.3 Convergenza alla soluzione

Siano $r = 2$ e $\theta = 1/2$. Così, nell'ipotesi in cui h sia uguale a Δt , la velocità di convergenza risulta quadratica in h . La figura 9 mostra la norma H^1 dell'errore al variare di $\sqrt{\Delta t^4 + h^4}$. L'indice k è stato scelto affinché $k\Delta t = t_{final} = 1$. L'andamento asintotico è in accordo con la teoria.

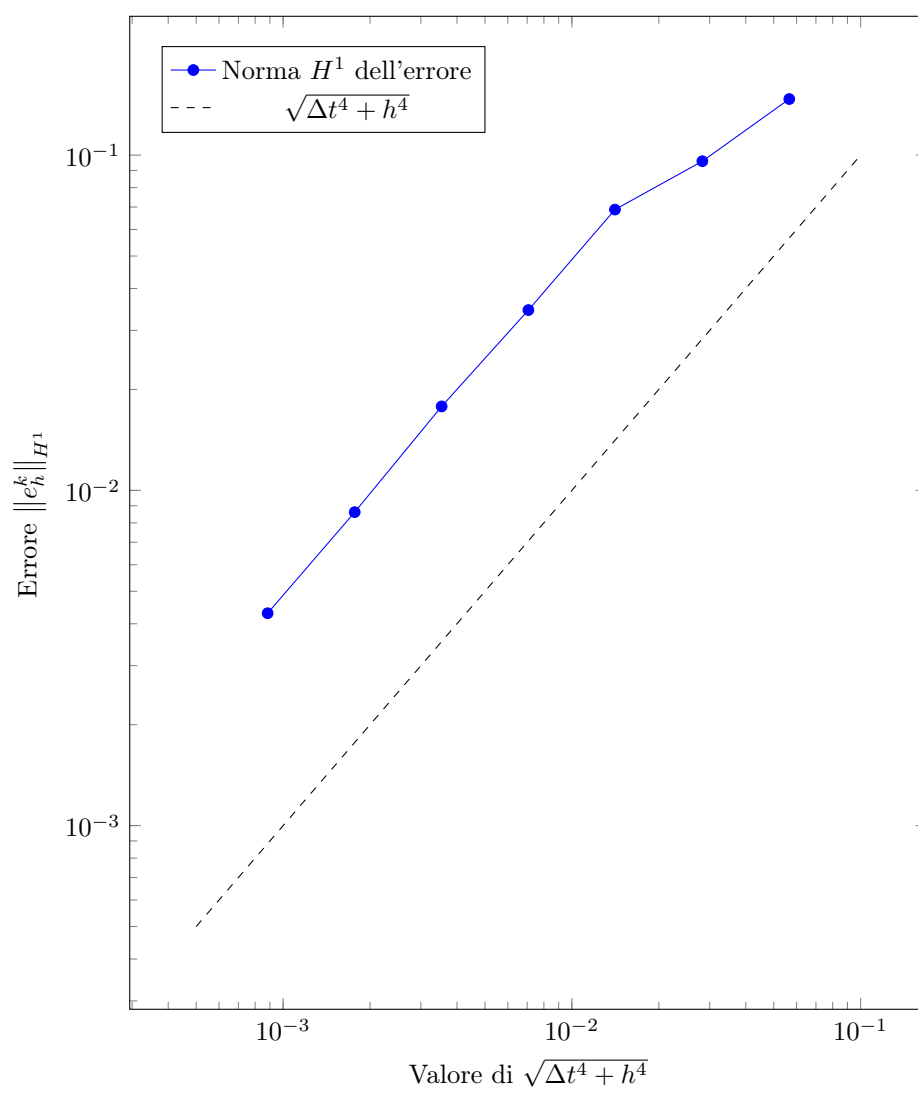


Figura 9: Convergenza del θ -metodo.