



UNIVERSITÀ DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE,
FISICHE E INFORMATICHE

CORSO DI LAUREA IN MATEMATICA

FEM adattivo per problemi di trasporto e diffusione

Relatore

Prof. Mauro Diligenti

Correlatrice

Prof.ssa Alessandra Aimi

Laureando

Ariel Surya Boiardi

Anno Accademico

2017-2018

Indice

1	Introduzione al FEM	1
1.1	Analisi di un problema differenziale in forma debole	1
1.1.1	Esistenza ed unicità della soluzione debole	1
1.2	Discretizzazione: i metodi di Galerkin	3
1.2.1	Analisi del problema di Galerkin	3
1.2.2	Formulazione algebrica del metodo di Galerkin	5
1.3	Il metodo degli elementi finiti nel caso monodimensionale	7
1.3.1	Lo spazio degli elementi finiti lineari	7
1.3.2	Interpolazione e stima dell'errore di interpolazione	8
1.3.3	Il condizionamento della matrice di rigidità	9
2	Analisi di un problema di trasporto e diffusione	11
2.0.1	Formulazione debole del problema	13
2.1	Risoluzione numerica del problema modello	14
2.1.1	Esistenza ed unicità della soluzione	14
2.1.2	Approssimazione in $X_h^1(0, 1)$	14
2.1.3	Sperimentazioni numeriche	16
2.1.4	Un altro esempio numerico	17
3	Il metodo degli elementi finiti adattivo	23
3.1	Stima residuale dell'errore	23
3.1.1	Rappresentazione esplicita del residuo	25
3.1.2	Stima superiore dell'errore	26
3.2	Tecniche di raffinamento	27
3.2.1	<i>Marking strategies</i>	27
3.3	Esempi numerici	29
3.3.1	Problema modello di trasporto-diffusione	30
3.3.2	Problema con <i>layer</i> interno	31
3.3.3	Problema con soluzione oscillante	31
A	Codici MATLAB	47
A.1	Risolutori ad elementi finiti	47
A.2	Risolutori adattivi	49

Introduzione

Numerosi problemi differenziali derivanti da modelli della fisica e dell'ingegneria presentano soluzioni con forti gradienti in regioni localizzate del dominio, o altre singolarità locali. In generale per approssimare tali soluzioni, caratterizzate da brusche variazioni, un approccio adattivo, basato cioè su raffinamenti mirati della discretizzazione, risulta più efficiente rispetto ai metodi che prevedono un raffinamento uniforme.

Uno degli strumenti più diffusi per la risoluzione numerica di problemi differenziali a derivate parziali con condizioni al bordo è l'analisi agli elementi finiti (FEA), ancora oggi alla base di gran parte dei software utilizzati in ambito industriale per l'analisi di strutture e la simulazione in meccanica del continuo, fluidodinamica e altri ambiti tecnici. Il fondamento del metodo degli elementi finiti (FEM) può essere collocato nei metodi di Ritz [19] e Galerkin [9], che proposero un approccio sistematico per l'approssimazione di problemi variazionali in spazi di dimensione finita [10]. La particolarità del FEM è che esso consente di scomporre il problema in contributi locali [15], grazie ad una opportuna costruzione dello spazio di approssimazione. Ampliando lo spazio di approssimazione miglioriamo la sua capacità di esprimere la soluzione del problema, aumentando però anche la complessità della risoluzione numerica del problema.

Nel presente lavoro di tesi abbiamo confrontato il classico metodo degli elementi finiti con un suo successivo sviluppo in direzione adattiva. Negli ultimi 35 anni i metodi automatici di discretizzazione adattiva sono stati ampiamente studiati nell'ambito del FEM [21], con lo scopo di ottenere, idealmente, approssimazioni entro una data tolleranza mantenendo la discretizzazione più efficiente possibile. L'idea del metodo degli elementi finiti adattivo (AFEM) è di raffinare lo spazio di approssimazione in modo da renderlo più flessibile solo dove necessario. Questo è reso possibile dal fatto che le funzioni che compongono lo spazio di approssimazione nel metodo degli elementi finiti hanno uno stretto legame con la geometria del dominio. Lo strumento principale nel metodo adattivo sono le stime residuali dell'errore, quantità facilmente calcolabili che danno informazioni globali e locali sull'errore commesso nell'approssimazione. Abbiamo affrontato con questi metodi i problemi di trasporto e diffusione monodimensionali, le cui soluzioni presentano, per alcune scelte dei parametri, marcati strati limite (*boundary layers*), e rappresentano quindi un buon *benchmark* per verificare le potenzialità del FEM adattivo.

Nel primo capitolo proponiamo un'introduzione alla teoria generale del metodo degli elementi finiti. Nel secondo prendiamo in esame il problema modello di trasporto e diffusione e proponiamo alcuni risultati numerici. Nel terzo capitolo, infine, mettiamo a punto un metodo agli elementi finiti adattivo e lo confrontiamo con il metodo classico in alcune simulazioni numeriche.

Capitolo 1

Il metodo di Galerkin agli elementi finiti

Nel presente capitolo proponiamo un'introduzione ai metodi di Galerkin agli elementi finiti, in modo da fissare la notazione ed uno schema di riferimento per i capitoli successivi. Dopo aver presentato alcuni risultati di analisi funzionale necessari per sviluppare la teoria successiva, procediamo presentando i metodi di Galerkin a livello formale e costruendo esplicitamente il metodo degli elementi finiti nel caso monodimensionale.

1.1 Analisi di un problema differenziale in forma debole

Consideriamo un problema differenziale al contorno sul dominio $\Omega \subset \mathbb{R}^m$

$$\begin{cases} \mathcal{L}u = f & \text{in } \Omega \\ \mathcal{B}u = 0 & \text{in } \partial\Omega, \end{cases} \quad (1.1.1)$$

con operatore differenziale \mathcal{L} del secondo ordine di tipo ellittico e \mathcal{B} operatore che descrive le condizioni al bordo. Assumiamo che il problema (1.1.1) ammetta una *formulazione debole*, ovvero:

$$\text{trovare } u \in V : a(u, v) = \mathcal{F}(v), \quad \forall v \in V, \quad (1.1.2)$$

essendo V un opportuno spazio di Hilbert [5], $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ una forma bilineare e $\mathcal{F} : V \rightarrow \mathbb{R}$ un funzionale lineare continuo che esprime il contributo del termine noto f e di eventuali condizioni di bordo non omogenee.

1.1.1 Esistenza ed unicità della soluzione debole

Prima di cercare un'approssimazione numerica del problema (1.1.2) è necessario capire se - e sotto quali condizioni - il problema debole sia risolubile almeno analiticamente. Riportiamo da [17] il seguente *Lemma di Lax-Milgram*, che risponde completamente a questa domanda.

Teorema 1.1 (Lemma di Lax-Milgram [13]). *Sia V uno spazio di Hilbert (reale) dotato della norma $\|\cdot\|$, sia $a : V \times V \rightarrow \mathbb{R}$ una forma bilineare ed $\mathcal{F} : V \rightarrow \mathbb{R}$ un funzionale lineare continuo nel duale V' di V . Si assuma inoltre che $a(\cdot, \cdot)$ sia continua, cioè*

$$\exists \gamma > 0 : |a(u, v)| \leq \gamma \|u\| \|v\|, \quad \forall u, v \in V \quad (1.1.3)$$

e coerciva, cioè

$$\exists \alpha > 0 : a(v, v) \geq \alpha \|v\|^2, \quad \forall v \in V. \quad (1.1.4)$$

Allora esiste un'unica soluzione $u \in V$ del problema (1.1.2) e si ha la stima a priori

$$\|u\| \leq \frac{1}{\alpha} \|\mathcal{F}\|_{V'}. \quad (1.1.5)$$

Dimostrazione. Grazie al teorema di rappresentazione di Riesz (si veda [5]), possiamo rappresentare \mathcal{F} attraverso un elemento di V , $\mathbf{R}\mathcal{F}$:

$$\mathcal{F}(v) = (\mathbf{R}\mathcal{F}, v)_V, \quad \forall v \in V,$$

e per ogni $w \in V$ fissato si ha

$$a(w, v) = (\mathcal{A}w, v)_V, \quad \forall v \in V,$$

ove $(\cdot, \cdot)_V : V \times V \rightarrow \mathbb{R}$ è il prodotto scalare di V , $\mathbf{R} : V' \rightarrow V$ e $\mathcal{A} : V \rightarrow V$ sono operatori lineari e continui. In particolare \mathbf{R} è un'isometria in quanto

$$\|\mathbf{R}\mathcal{F}\| = \sup_{\substack{v \in V \\ v \neq 0}} \frac{(\mathbf{R}\mathcal{F}, v)_V}{\|v\|} = \sup_{\substack{v \in V \\ v \neq 0}} \frac{\mathcal{F}(v)}{\|v\|} = \|\mathcal{F}\|_{V'}$$

inoltre per la continuità di $a(\cdot, \cdot)$ in (1.1.3)

$$\|\mathcal{A}w\| = \sup_{\substack{v \in V \\ v \neq 0}} \frac{(\mathcal{A}w, v)_V}{\|v\|} = \sup_{\substack{v \in V \\ v \neq 0}} \frac{a(w, v)}{\|v\|} \leq \gamma \|w\|.$$

Tornando al problema (1.1.2) si tratta di determinare $u \in V$ per cui $(\mathcal{A}u, v) = (\mathbf{R}\mathcal{F}, v)$ per ogni $v \in V$, ovvero (1.1.2) risulta equivalente a al problema:

$$\forall \mathcal{F} \in V', \text{ trovare } u \in V : \mathcal{A}u = \mathbf{R}\mathcal{F}. \quad (1.1.6)$$

Pertanto la risolubilità di (1.1.2) è equivalente alla biiettività dell'operatore \mathcal{A} . La (1.1.4) ci dà che

$$\|v\|^2 \leq \frac{1}{\alpha} (\mathcal{A}v, v)_V \leq \frac{1}{\alpha} \|\mathcal{A}v\| \|v\|, \quad (1.1.7)$$

e quindi, se $\mathcal{A}v = 0$, deve essere *a fortiori* $v = 0$, ovvero \mathcal{A} è iniettiva.

Consideriamo una successione $\{\mathcal{A}v_n\} \subset \text{Im}(\mathcal{A})$ convergente a $w \in V$, da 1.1.7 abbiamo che $\|v\| \leq \frac{1}{\alpha} \|\mathcal{A}v\|$ e pertanto

$$\|v_n - v_m\| \leq \frac{1}{\alpha} \|\mathcal{A}v_n - \mathcal{A}v_m\|,$$

ovvero v_n è una successione di Cauchy in V ; sia $v := \lim_{n \rightarrow \infty} v_n$. Per la continuità di \mathcal{A} si ha $\mathcal{A}v = \mathcal{A} \lim_{n \rightarrow \infty} v_n = \lim_{n \rightarrow \infty} \mathcal{A}v_n = w$, pertanto $w \in \text{Im}(\mathcal{A})$, ovvero $\text{Im}(\mathcal{A})$ è un chiuso. Sia $z \in \text{Im}(\mathcal{A})^\perp$:

$$0 = (\mathcal{A}z, z)_V = a(z, z) \geq \alpha \|z\|^2,$$

dove l'ultima disuguaglianza si è ottenuta dalla coercività di a . Ma allora $z = 0$, ovvero $\text{Im}(\mathcal{A})^\perp = \{0\}$. La teoria degli spazi di Hilbert permette di concludere che, essendo $\text{Im}(\mathcal{A})$ chiusa in V e $\text{Im}(\mathcal{A})^\perp = \{0\}$, $\text{Im}(\mathcal{A}) = V$ i.e. \mathcal{A} è surgettiva.

Per provare la disuguaglianza (1.1.5) consideriamo in particolare che v coincida con l'unica soluzione u di (1.1.2), allora si ha dalla (1.1.4) che

$$\|u\|^2 \leq \frac{1}{\alpha} a(u, u) = \frac{1}{\alpha} \mathcal{F}(u) \leq \frac{1}{\alpha} \|\mathcal{F}\|_{V'} \|u\|,$$

il che prova la stima (1.1.5). □

1.2 Discretizzazione: i metodi di Galerkin

Una volta che il problema differenziale sia stato posto in forma debole come in (1.1.2), sotto le ipotesi del Lemma di Lax-Milgram (Teorema 1.1) abbiamo garantite esistenza ed unicità della soluzione. Per poter calcolare numericamente la soluzione del problema, sarà necessario procedere ad una sua discretizzazione.

Sia (V_h) una famiglia di spazi dipendente da un parametro positivo h , tali che $V_h \subset V$ e $\dim V_h = N_h < +\infty$, $\forall h$. Consideriamo quindi il problema debole (1.1.2) approssimato nel sottospazio finito-dimensionale V_h :

$$\text{trovare } u_h \in V_h : a(u_h, v_h) = \mathcal{F}(v_h), \quad \forall v_h \in V_h. \quad (1.2.1)$$

Il problema debole nella forma approssimata (1.2.1) è anche detto *problema di Galerkin*.

1.2.1 Analisi del problema di Galerkin

In questa sezione proviamo l'esistenza e l'unicità della soluzione del problema di Galerkin (1.2.1), proviamo successivamente la convergenza della soluzione del problema di Galerkin in V_h alla soluzione del problema debole (1.1.2).

Teorema 1.2 (Esistenza e unicità della soluzione del problema di Galerkin). *Sotto le stesse ipotesi del Lemma di Lax-Milgram (Teorema 1.1), esiste un'unica soluzione u_h del problema di Galerkin (1.2.1) e tale soluzione risulta stabile in quanto*

$$\|u_h\| \leq \frac{1}{\alpha} \|\mathcal{F}\|_{V'}. \quad (1.2.2)$$

Dimostrazione. Lo spazio V_h è uno spazio di Hilbert, essendo sottospazio chiuso dello spazio di Hilbert V ; inoltre la forma bilineare $a(\cdot, \cdot)$ e il funzionale $\mathcal{F}(\cdot)$ in (1.2.1) sono i medesimi del problema in forma debole (1.1.2). Abbiamo quindi tutte le ipotesi del Lemma di Lax-Milgram, da cui si ottiene che esiste ed è unica la soluzione $u_h \in V_h$ del problema di Galerkin (1.2.1).

Consideriamo poi in (1.2.1) $v_h = u_h$ l'unica soluzione del problema, dalla coercività della forma bilineare a si ottiene

$$\alpha \|u_h\|^2 \leq a(u_h, u_h) = \mathcal{F}(u_h) \leq \|\mathcal{F}\|_{V'} \|u_h\|,$$

da cui la stabilità (1.2.2). \square

Convergenza Vorremmo a questo punto garantire che la soluzione del problema di Galerkin converga alla soluzione del problema debole (1.1.2) quando h tende a zero. Di conseguenza, a patto di scegliere h sufficientemente piccolo si potrà approssimare con precisione arbitraria la soluzione esatta u del problema debole con una approssimazione $u_h \in V_h$ ricavata con il metodo di Galerkin.

Proposizione 1.3 (Ortogonalità di Galerkin [4]). *La soluzione u_h del problema di Galerkin (1.2.1) soddisfa la seguente proprietà di ortogonalità (o forte consistenza):*

$$a(u - u_h, v_h) = 0, \quad \forall v_h \in V_h \quad (1.2.3)$$

Dimostrazione. Essendo $V_h \subset V$, la soluzione esatta u soddisfa il problema debole (1.1.2) considerando in particolare le funzioni test in V_h :

$$a(u, v_h) = \mathcal{F}(v_h), \quad \forall v_h \in V_h;$$

sottraendo a quest'espressione la (1.2.1) si ha $a(u, v_h) - a(u_h, v_h) = 0$ per ogni $v_h \in V_h$, ove u_h è la soluzione del problema di Galerkin. Allora, grazie alla bilinearità di $a(\cdot, \cdot)$ si ha la (1.2.3). \square

Osservazione. Se $a(\cdot, \cdot)$ è simmetrica, essa definisce un prodotto scalare su V ; allora la proprietà di consistenza si interpreta come l'ortogonalità, rispetto al prodotto scalare $a(\cdot, \cdot)$ tra l'errore di approssimazione di u mediante u_h e il sottospazio V_h . In questo senso possiamo dire che la soluzione u_h del metodo di Galerkin è la proiezione su V_h della soluzione esatta u ed è quindi la migliore approssimazione di u nella norma dell'energia, vale a dire nella norma indotta dalla forma $a(\cdot, \cdot)$ pensata come prodotto scalare su V .

In termini della norma di V la soluzione approssimata u_h non è ottimale in V_h , ma il seguente *Lemma di Céa* [6] garantisce che u_h è ottima in V_h a meno di una costante che dipende dalla forma $a(\cdot, \cdot)$.

Teorema 1.4 (Lemma di Céa [6]). *Nelle condizioni del Teorema 1.2, siano $u \in V$ la soluzione del problema debole (1.1.2) e u_h la soluzione del problema di Galerkin (1.2.1). Allora vale la seguente stima dell'errore*

$$\|u - u_h\| \leq \frac{\gamma}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|. \quad (1.2.4)$$

Dimostrazione. Consideriamo $a(u - u_h, u - u_h) = a(u - u_h, u - v_h) + a(u - u_h, v_h - u_h)$ e poiché grazie alla (1.2.3) per ogni $v_h \in V_h$ il secondo addendo è nullo, si ha

$$a(u - u_h, u - u_h) = a(u - u_h, u - v_h).$$

Sfruttando la continuità della forma bilineare (1.1.3)

$$|a(u - u_h, u - v_h)| \leq \gamma \|u - u_h\| \|u - v_h\|;$$

la coercività (1.1.4) invece garantisce che

$$a(u - u_h, u - u_h) \geq \alpha \|u - u_h\|^2.$$

Mettendo insieme queste due disuguaglianze e dividendo per $\|u - u_h\|$:

$$\|u - u_h\| \leq \frac{\gamma}{\alpha} \|u - v_h\|, \quad \forall v_h \in V_h,$$

e valendo tale disuguaglianza per ogni $v_h \in V_h$, passando all'estremo inferiore otteniamo la (1.2.4). \square

Naturalmente $\inf_{v_h \in V_h} \|u - v_h\| \leq \|u - u_h\|$, quindi se $\frac{\gamma}{\alpha}$ in (1.2.4) è dell'ordine dell'unità, l'approssimazione u_h costruita con il metodo di Galerkin risulta essere la migliore approssimazione di u ottenibile in V_h . In ogni caso i due errori hanno lo stesso ordine di infinitesimo rispetto ad h .

Grazie al Lemma di Céa è evidente che per avere garantita la convergenza del metodo di Galerkin basterà porre sulla famiglia degli spazi V_h la *condizione di densità*

$$\forall v \in V, \quad \lim_{h \rightarrow 0} \left[\inf_{v_h \in V_h} \|v - v_h\|_V \right] = 0, \quad (1.2.5)$$

per cui al tendere a zero del parametro h lo spazio V_h tende a saturare tutto lo spazio V . La famiglia di spazi V_h andrà quindi scelta opportunamente in modo da garantire la proprietà (1.2.5). Una volta che questo requisito risulta soddisfatto, si avrà

$$\lim_{h \rightarrow 0} \|u - u_h\| = 0,$$

indipendentemente da come si sia scelto V_h e da come sia fatta u . Dalla scelta di V_h e dalla regolarità di u dipenderà invece la velocità con cui la soluzione discreta converge alla soluzione esatta rispetto al parametro h .

1.2.2 Formulazione algebrica del metodo di Galerkin

Sia $\{\varphi_j : j = 1, 2, \dots, N_h\}$ una base per il sottospazio V_h , per linearità basta che la (1.2.1) sia verificata per ogni funzione della base:

$$a(u_h, \varphi_i) = \mathcal{F}(\varphi_i), \quad i = 1, 2, \dots, N_h \quad (1.2.6)$$

e naturalmente, poiché cerchiamo $u_h \in V_h$,

$$u_h = \sum_{j=1}^{N_h} u_j \varphi_j(x),$$

ove gli u_j , $j = 1, \dots, N_h$ sono coefficienti incogniti che rappresentano la soluzione approssimata rispetto alla base scelta per V_h . Riformuliamo quindi il problema di Galerkin (1.2.1), sfruttando la bilinearità di a :

$$\sum_{j=1}^{N_h} u_j a(\varphi_j, \varphi_i) = \mathcal{F}(\varphi_i), \quad i = 1, 2, \dots, N_h. \quad (1.2.7)$$

Denotiamo con

$$\mathbf{A} = [a_{ij}], \quad a_{ij} = a(\varphi_j, \varphi_i) \quad (1.2.8)$$

la matrice del sistema (detta di *rigidezza* o *stiffness*) e

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_h} \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} \mathcal{F}(\varphi_1) \\ \mathcal{F}(\varphi_2) \\ \vdots \\ \mathcal{F}(\varphi_{N_h}) \end{bmatrix}$$

i vettori delle incognite e dei termini noti. Allora le equazioni (1.2.7) si scrivono nel sistema di N_h equazioni in N_h incognite

$$\mathbf{A}\mathbf{u} = \mathbf{f}. \quad (1.2.9)$$

Alcune proprietà della matrice di rigidezza (1.2.8) sono indipendenti dalla scelta dello spazio V_h e della sua base, ma dipendono direttamente dalle caratteristiche del problema che si sta approssimando, ovvero della forma bilineare a .

Teorema 1.5. *Sia $\mathbf{A} \in \mathbb{R}^{N_h \times N_h}$ la matrice di rigidezza associata alla discretizzazione con il metodo Galerkin in un sottospazio $V_h \subset V$ di dimensione N_h del problema debole (1.1.2). Se la forma bilineare $a(\cdot, \cdot)$ del problema debole è coerciva, allora la matrice \mathbf{A} risulta definita positiva. Si ha inoltre che \mathbf{A} è simmetrica se e solo se la forma bilineare $a(\cdot, \cdot)$ è simmetrica.*

Dimostrazione. Dato un generico vettore $\mathbf{v} = [v_i]_{i=1, \dots, N_h} \in \mathbb{R}^{N_h}$ abbiamo che dalla definizione di \mathbf{A} e dalla bilinearità di a

$$\begin{aligned} \mathbf{v}^T \mathbf{A} \mathbf{v} &= \sum_{j=1}^{N_h} \sum_{i=1}^{N_h} v_i a_{ij} v_j = \sum_{j=1}^{N_h} \sum_{i=1}^{N_h} v_i v_j a(\varphi_j, \varphi_i) \\ &= \sum_{j=1}^{N_h} \sum_{i=1}^{N_h} a(v_j \varphi_j, v_i \varphi_i) = a\left(\sum_{j=1}^{N_h} v_j \varphi_j, \sum_{i=1}^{N_h} v_i \varphi_i\right) \\ &= a(v_h, v_h) \geq \alpha \|v_h\|^2 \geq 0, \end{aligned} \quad (1.2.10)$$

dove l'ultima disuguaglianza si ottiene dalla coercività di a in (1.1.4). Inoltre se fosse $\mathbf{v}^T \mathbf{A} \mathbf{v} = 0$ dalla precedente (1.2.10) dovrebbe essere $\|v_h\| = 0$ ovvero $v_h = 0$ e quindi $\mathbf{v} = 0$.

Per provare la simmetria basta ricordare la (1.2.8). \square

Le proprietà specifiche della matrice di rigidezza dipenderanno dalla scelta dello spazio e dalla base considerata. Alcune proprietà importanti di questo tipo sono il numero di condizionamento

e la struttura della matrice. Ad esempio la scelta di una base formata da funzioni a supporto piccolo (come vedremo nel caso degli elementi finiti) darà luogo ad una matrice sparsa. Questo è un fatto importante da tenere a mente quando si sceglie lo spazio e soprattutto la base: può essere che la possibilità o impossibilità di risolvere numericamente un dato problema dipendano esclusivamente dal modo in cui si è scelta più o meno opportunamente la base dello spazio V_h .

1.3 Il metodo degli elementi finiti nel caso monodimensionale

Il metodo degli elementi finiti (FEM) consiste nella costruzione di una soluzione al problema di Galerkin (1.2.1) in uno spazio di funzioni a supporto locale.

Per un dominio Ω monodimensionale, ovvero $\Omega = [a, b] \subset \mathbb{R}$, introduciamo una partizione \mathcal{T}_h dell'intervallo $[a, b]$ consistente di N_h intervalli $K_j = (x_{j-1}, x_j)$ di ampiezza $h_j = x_j - x_{j-1}$, $j = 1, \dots, N_h$, con

$$a = x_0 < x_1 < \dots < x_{N_h-1} < x_{N_h} = b \tag{1.3.1}$$

e poniamo $h = \max_j h_j$. Denotiamo $\mathcal{N}_h = \{x_j, j = 0, \dots, N_h\}$ l'insieme dei *odi* della partizione \mathcal{T}_h . Denotando con \mathbb{P}_r lo spazio dei polinomi di grado al più r , definiamo lo spazio di elementi finiti di grado al più $r \geq 1$ sulla partizione \mathcal{T}_h

$$X_h^r = \left\{ v_h \in \mathcal{C}^0(\Omega) : [v_h]_{K_j} \in \mathbb{P}_r, \forall K_j \in \mathcal{T}_h \right\}. \tag{1.3.2}$$

Gli spazi X_h^r sono tutti sottospazi di $H^1(\Omega)$ in quanto costituiti da funzioni derivabili tranne che in al più un numero finito di punti (i nodi x_j della partizione \mathcal{T}_h) [16].

1.3.1 Lo spazio degli elementi finiti lineari

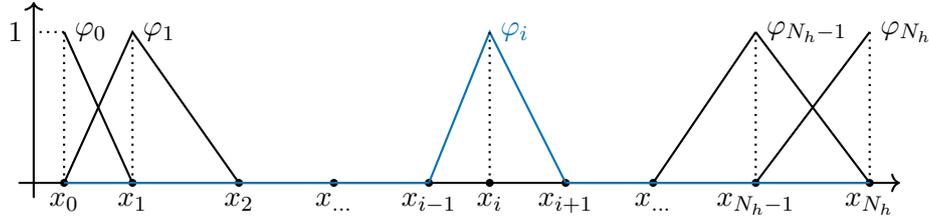
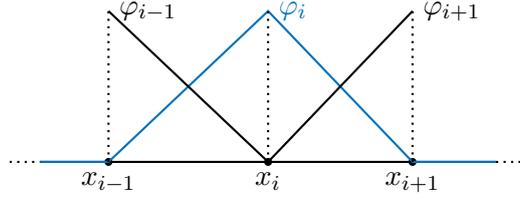
Data una partizione \mathcal{T}_h di Ω della forma (1.3.1), lo spazio X_h^1 è costituito dalle funzioni continue in Ω e lineari sugli intervalli della partizione; ogni tratto lineare è quindi determinato dai punti di passaggio agli estremi dell'intervallo su cui è definito, inoltre per continuità tratti consecutivi si raccordano. Pertanto per determinare in modo univoco una funzione in X_h^1 sarà sufficiente assegnare i valori da essa assunti negli $N_h + 1$ nodi della partizione \mathcal{T}_h . Le funzioni di questo spazio hanno $N_h + 1$ *gradi di libertà*, ovvero lo spazio X_h^1 ha dimensione $N_h + 1$.

Possiamo definire una base per X_h^1 formata da funzioni della forma

$$\varphi_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & x_i \leq x \leq x_{i+1} \\ 0 & \text{altrove} \end{cases} \quad \text{per } i = 1, \dots, N_h - 1 \tag{1.3.3}$$

$$\varphi_0 = \begin{cases} \frac{x_1-x}{x_1-x_0} & x_0 \leq x \leq x_1 \\ 0 & \text{altrove} \end{cases} \quad \varphi_{N_h} = \begin{cases} \frac{x-x_{N_h-1}}{x_{N_h}-x_{N_h-1}} & x_{N_h-1} \leq x \leq x_{N_h} \\ 0 & \text{altrove} \end{cases}$$

La rappresentazione della base in Fig. 1.1 rende evidente che ogni φ_i della base ha supporto dato da $[x_{i-1}, x_i] \cup [x_i, x_{i+1}]$ se $i = 1, \dots, N_h - 1$, mentre sarà $[x_0, x_1]$ nel caso di φ_0 e $[x_{N_h-1}, x_{N_h}]$ per φ_{N_h} . Soltanto le funzioni di base φ_{i-1} e φ_{i+1} avranno supporto ad intersezione non vuota con quello di φ_i e quindi la matrice di rigidezza risulterà tridiagonale in quanto $a_{ij} = 0$ ogni volta che $j \notin \{i - 1, i, i + 1\}$, si veda Fig. 1.2.

Figura 1.1: Rappresentazione della base di X_h^1 Figura 1.2: Intersezioni dei supporti delle funzioni di base di X_h^1

1.3.2 Interpolazione e stima dell'errore di interpolazione

Riportiamo brevemente alcuni risultati per stimare l'errore commesso nell'interpolazione con funzioni X_h^r . Consideriamo il generico intervallo $\Omega = (a, b)$ e $v \in \mathcal{C}^0(\overline{\Omega})$

Definizione 1. Definiamo l'operatore di interpolazione

$$\Pi_{\mathcal{T}_h}^r : \mathcal{C}^0(\overline{\Omega}) \rightarrow X_h^r(\overline{\Omega}), \quad r \geq 1, \quad (1.3.4)$$

che a v associa il suo interpolante nello spazio X_h^r determinato dalla partizione \mathcal{T}_h di Ω :

$$\Pi_{\mathcal{T}_h}^r v(x_i) = v(x_i), \quad \forall x_i \in \mathcal{N}_h.$$

Definizione 2. Indichiamo con $|v|_{\mathbb{H}^{r+1}(\Omega)}$ la seminorma di $\mathbb{H}^{r+1}(\Omega)$ [5]:

$$|v|_{\mathbb{H}^{r+1}(\Omega)} := \left\| D^{(r+1)}v \right\|_{L^2(\Omega)}.$$

Per l'errore di interpolazione in X_h^r si ha la seguente stima [16]:

Teorema 1.6. Sia $v \in \mathbb{H}^{r+1}(\Omega)$ per $r \geq 1$, vale la seguente stima:

$$\|v - \Pi_h^r v\|_{\mathbb{H}^k(\Omega)} \leq C_{k,r} h^{r+1-k} |v|_{\mathbb{H}^{r+1}(\Omega)}, \quad k = 0, 1, \quad (1.3.5)$$

dove le costanti $C_{k,r}$ sono indipendenti da v ed h .

Osservazione 1.3.1. Se la funzione interpolata v è solo in $\mathbb{H}^1(\Omega)$ si può comunque provare il seguente risultato

$$\|v - \Pi_h^1 v\|_{L^2(\Omega)} \leq h |v|_{\mathbb{H}^1(\Omega)}, \quad (1.3.6)$$

come riportato in [8].

1.3.3 Il condizionamento della matrice di rigidezza

Si è già provato che la matrice di rigidezza $A = [a(\varphi_j, \varphi_i)]$ associata al problema di Galerkin, ed in particolare al metodo degli elementi finiti, è definita positiva ed è simmetrica se la forma bilineare a è simmetrica. In questo caso il numero di condizionamento in norma spettrale è dato da

$$K_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \quad (1.3.7)$$

ove $\lambda_{\max}(A)$ e $\lambda_{\min}(A)$ denotano rispettivamente il massimo e il minimo degli autovalori della matrice A . Per il numero di condizionamento della matrice di rigidezza abbiamo la seguente stima che riportiamo da [17].

Definizione 3. Diciamo che una partizione \mathcal{T}_h della forma (1.3.1) è *quasi-uniforme* se esiste una costante $\tau > 0$ tale che

$$\min_{K \in \mathcal{T}_h} h_K \geq \tau h, \quad \forall h > 0, \quad (1.3.8)$$

con h_K il diametro dell'elemento $K \in \mathcal{T}_h$.

Teorema 1.7. Se $a(\cdot, \cdot)$ è simmetrica e le partizioni $\{\mathcal{T}_h, h > 0\}$ sono quasi-uniformi, vale la seguente relazione per il numero di condizionamento (1.3.7)

$$K_2(A) = \frac{C}{h^2}, \quad (1.3.9)$$

con C una costante indipendente da h , ma dipendente dal grado degli elementi finiti utilizzati.

Allora, anche se la teoria assicura che l'approssimazione agli elementi finiti migliora diminuendo il passo h della partizione, difficilmente potremo ottenere in questo modo un miglioramento arbitrario del risultato. Infatti a fronte di sistemi lineari di dimensioni sempre maggiori, avremo anche un deterioramento della qualità della soluzione numerica dovuto all'aumento del numero di condizionamento.

Capitolo 2

Analisi di un problema di trasporto e diffusione

Consideriamo il problema di trasporto e diffusione monodimensionale con condizioni al bordo di Dirichlet:

$$\begin{cases} -\mu u'' + bu' = 0, & 0 < x < 1 \\ u(0) = 0, \quad u(1) = 1 \end{cases} \quad \begin{matrix} (2.0.1) \\ (2.0.2) \end{matrix}$$

con μ, b costanti reali positive. Questo problema è un buon *benchmark* in quanto la sua soluzione è calcolabile analiticamente e presenta un *boundary layer* la cui ampiezza è controllabile al variare dei parametri (Fig. 2.1). Potremo quindi confrontare esplicitamente le soluzioni approssimate con la soluzione esatta.

In molte applicazioni pratiche il termine di diffusione $-\mu u''$ è dominato dal termine convettivo (o di trasporto) bu' ; in questi casi, come vedremo, la soluzione può dare luogo a *strati limite* (*boundary layer*), ovvero regioni, in genere in prossimità della frontiera del dominio, in cui la soluzione è caratterizzata da forti gradienti [16].

Risoluzione analitica Il problema (2.0.1) è risolubile per via analitica, l'equazione caratteristica ad esso associata è $-\mu\lambda^2 + b\lambda = 0$ che ha come radici $\lambda_1 = 0$, $\lambda_2 = \frac{b}{\mu}$. L'integrale generale dell'equazione è quindi

$$u(x) = C_1 + C_2 e^{\frac{b}{\mu}x}, \quad \text{con } C_1, C_2 \in \mathbb{R}. \quad (2.0.3)$$

Imponendo le condizioni al bordo

$$\begin{cases} u(0) = C_1 + C_2 = 0 \\ u(1) = C_1 + C_2 e^{\frac{b}{\mu}} = C_2(1 - e^{\frac{b}{\mu}}) = 1 \end{cases}$$

si ottiene la soluzione analitica del problema (2.0.1)

$$u(x) = \frac{e^{\frac{b}{\mu}x} - 1}{e^{\frac{b}{\mu}} - 1}. \quad (2.0.4)$$

Per $\frac{b}{\mu}$ prossimo a zero, possiamo espandere in serie gli esponenziali e troncando lo sviluppo al primo ordine otteniamo

$$\frac{e^{\frac{b}{\mu}x} - 1}{e^{\frac{b}{\mu}} - 1} = \frac{(1 + \frac{b}{\mu}x + \dots) - 1}{(1 + \frac{b}{\mu} + \dots) - 1} \approx \frac{\frac{b}{\mu}x}{\frac{b}{\mu}} = x,$$

quindi se il termine diffusivo prevale su quello di trasporto la soluzione è prossima ad una retta interpolante i dati al bordo. Se invece il rapporto $\frac{b}{\mu}$ è grande gli esponenziali prevalgono e

$$u(x) \approx \frac{e^{\frac{b}{\mu}x}}{e^{\frac{b}{\mu}}} = e^{-\frac{b}{\mu}(1-x)}.$$

Pertanto la soluzione ha un andamento esponenziale ed è molto prossima a zero in quasi tutto l'intervallo, salvo presentare una crescita molto rapida in prossimità di 1, dove si raccorda con il dato al bordo unitario. L'ampiezza della regione in cui la soluzione inizia a crescere è dell'ordine di $\frac{\mu}{b}$: la soluzione presenta quindi uno *strato limite* (*boundary layer*) di ampiezza $\mathcal{O}\left(\frac{\mu}{b}\right)$.

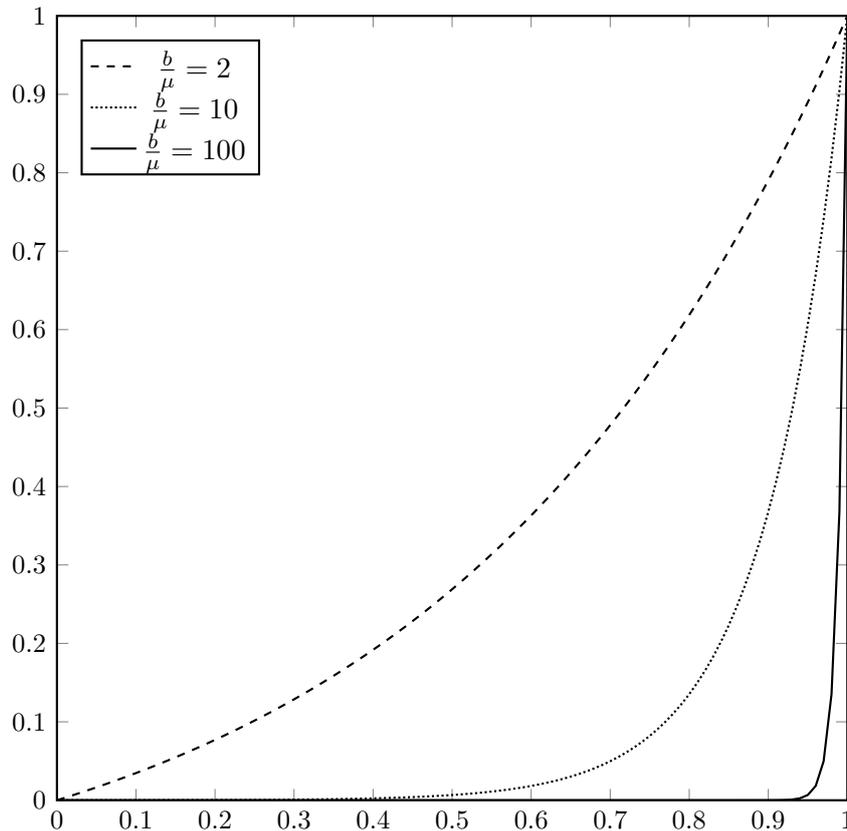


Figura 2.1: Andamento della soluzione del problema (2.0.1) al variare del rapporto $\frac{b}{\mu}$: si noti in particolare il progressivo inasprimento del *boundary layer*.

2.0.1 Formulazione debole del problema

I metodi di approssimazione che abbiamo presentato hanno come punto di partenza una riformulazione del problema differenziale in forma debole. Ci limitiamo nel seguito a presentare alcuni passaggi formali sul problema modello di trasporto e diffusione per ottenerne la formulazione debole. Per una presentazione più completa e approfondita dell'argomento è possibile consultare [4, 5, 16].

Per dare significato alle operazioni successive definiamo lo *spazio di Sobolev*

$$H^1(0, 1) = \{v \in L^2(0, 1) : v' \in L^2(0, 1)\},$$

dove la derivata è intesa in senso debole (si veda [5]). Definiamo anche il sottospazio costituito dalle funzioni di $H^1(0, 1)$ che sono nulle agli estremi dell'intervallo:

$$H_0^1(0, 1) = \{v \in H^1(0, 1) : v(0) = v(1) = 0\}.$$

Dall'equazione (2.0.1), scelta a piacere una *funzione test* $v \in H_0^1(0, 1)$ abbiamo

$$-\mu u''v + bu'v = 0. \quad (2.0.5)$$

Integrando la (2.0.5) su $[0, 1]$ ricaviamo

$$\int_0^1 (-\mu u''v + bu'v) dx = [-\mu u'v]_0^1 + \int_0^1 (\mu u'v' + bu'v) dx = \int_0^1 (\mu u'v' + bu'v) dx.$$

La formulazione debole del problema sarà quindi

$$\text{trovare } u \in H^1(0, 1) : a(u, v) = 0, \quad \forall v \in H_0^1(0, 1), \quad (2.0.6)$$

ove si è posto

$$a(u, v) = \int_0^1 (\mu u'v' + bu'v) dx. \quad (2.0.7)$$

Passaggio a dati al bordo omogenei La teoria sviluppata per i metodi di Galerkin richiede che si cerchi la soluzione del problema debole nello stesso spazio in cui si scelgono le funzioni test. Si rende quindi necessario riscrivere il problema debole (2.0.6) in modo da poter cercare la soluzione in $H_0^1(0, 1)$.

Sia $u(x)$ soluzione del problema non omogeneo, fissato un opportuno *rilevamento dei dati al bordo* $B \in H^1(0, 1)$, tale che $B(0) = u(0)$, $B(1) = u(1)$, ponendo $\hat{u} = u - B \in H_0^1(0, 1)$, con la notazione precedente possiamo formulare il problema debole (2.0.6) come

$$\text{trovare } \hat{u} \in H_0^1(0, 1) : a(\hat{u}, v) = F(v), \quad \forall v \in H_0^1(0, 1); \quad (2.0.8)$$

ove

$$F(v) = -a(B, v) \quad (2.0.9)$$

è il contributo al termine noto dovuto al rilevamento dei dati al bordo. Una volta determinata la soluzione di (2.0.8), la soluzione $u \in H^1(0, 1)$ del problema debole (2.0.6) è data da $u = \hat{u} + B$.

2.1 Risoluzione numerica del problema modello

In questa sezione formuliamo la discretizzazione del problema modello con il FEM e presentiamo i risultati numerici.

2.1.1 Esistenza ed unicità della soluzione

Prima di tutto è necessario verificare che il problema modello in forma debole (2.0.8) rispetti le ipotesi del lemma di Lax-Milgram. Si tratta quindi di provare continuità (1.1.3) e coercività (1.1.4) della forma bilineare $a(\cdot, \cdot)$ in (2.0.7).

Per la continuità consideriamo $u, v \in H_0^1(0, 1)$, si ha che

$$|a(u, v)| \leq \int_0^1 |\mu u' v'| \, dx + \int_0^1 |b u' v| \, dx \leq \mu \|u'\|_{L^2} \|v'\|_{L^2} + b \|u'\|_{L^2} \|v\|_{L^2},$$

dove l'ultima disuguaglianza è la disuguaglianza di Hölder [5] applicata ad entrambi i membri. Ponendo $\gamma = \max(\mu, b)$ abbiamo

$$|a(u, v)| \leq \gamma (\|u'\|_{L^2} \|v'\|_{L^2} + \|u'\|_{L^2} \|v\|_{L^2}) \leq 2\gamma \|u\|_{H^1} \|v\|_{H^1}$$

che è la continuità della forma bilineare.

Sia ora $v \in H_0^1(0, 1)$,

$$a(v, v) = \int_0^1 (\mu (v')^2 + b v' v) \, dx \quad (2.1.1)$$

e si ha che

$$\int_0^1 \mu (v')^2 \, dx = \mu \|v'\|_{L^2}^2.$$

Inoltre

$$\int_0^1 b v v' \, dx = [b v^2]_0^1 - \int_0^1 b v' v \, dx = - \int_0^1 b v' v \, dx,$$

pertanto $\int_0^1 b v v' \, dx = 0$.

Allora dalla (2.1.1)

$$a(v, v) = \mu \|v'\|_{L^2}^2,$$

e poiché per la disuguaglianza di Poincaré [5] si ha che la norma in $H_0^1(0, 1)$ risulta equivalente alla norma $L^2(0, 1)$ della derivata, esiste una costante $\alpha > 0$ tale che $a(v, v) \geq \alpha \|v\|_{H_0^1}^2$, ovvero la forma a è coerciva.

2.1.2 Approssimazione in $X_h^1(0, 1)$

Come primo approccio consideriamo una partizione \mathcal{T}_h a passo costante $h = \frac{1}{N}$ con nodi $x_i = ih$, $i = 0, \dots, N$ e proviamo ad approssimare (2.0.6) nello spazio $X_h^1(0, 1)$ basato su tale partizione. Per incorporare le condizioni di bordo di Dirichlet omogenee nella scelta dello spazio funzionale introduciamo

$$\mathring{X}_h^1(0, 1) = \{v_h \in X_h^1(0, 1) : v_h(0) = v_h(1) = 0\} = \text{Span} \{\varphi_1, \dots, \varphi_{N-1}\}, \quad (2.1.2)$$

ove le φ_i sono le funzioni della base di $X_h^1(0,1)$ in (1.3.3). In pratica si tratta di tralasciare le due funzioni terminali, come si può facilmente vedere in Fig. 1.1.

Il problema in forma debole approssimato in $\overset{\circ}{X}_h^1(0,1)$ sarà

$$\text{trovare } u_h \in \overset{\circ}{X}_h^1 : a(u_h, \varphi_i) = F(\varphi_i), \quad i = 1, \dots, N-1, \quad (2.1.3)$$

con la forma bilineare a definita in (2.0.7) e il funzionale F come in (2.0.9). Naturalmente essendo $u_h \in \overset{\circ}{X}_h^1(0,1)$, si avrà

$$u_h = \sum_{j=1}^{N-1} u_j \varphi_j,$$

con i coefficienti u_j da determinare; potremo pertanto riscrivere la (2.1.3) come un sistema di $N-1$ equazioni nelle $N-1$ incognite u_j :

$$\sum_{j=1}^{N-1} u_j a(\varphi_j, \varphi_i) = F(\varphi_i), \quad i = 1, \dots, N-1. \quad (2.1.4)$$

Una volta che siano determinati i coefficienti u_j , la soluzione del problema non omogeneo sarà data da

$$\tilde{u}_h = u_h + u_0 \varphi_0 + u_N \varphi_N, \quad (2.1.5)$$

dove φ_0 e φ_N sono le due funzioni di base definite agli estremi dell'intervallo in (1.3.3) che non appartengono alla base di $\overset{\circ}{X}_h^1(0,1)$ e i valori della soluzione al bordo $u_0 = u(0)$, $u_N = u(1)$ sono noti dalle condizioni (2.0.2). In altre parole abbiamo implicitamente introdotto come rilevamento dei dati al bordo

$$B(x) = u_0 \varphi_0(x) + u_N \varphi_N(x) = \varphi_N(x), \quad (2.1.6)$$

che sarà il contributo al termine noto nella (2.1.4).

Assemblaggio Nel procedimento del FEM la fase di assemblaggio è quella in cui viene costruita la matrice di rigidità. Dalla definizione della forma bilineare (2.0.7) si ha che gli elementi della matrice di rigidità sono

$$a_{ij} = a(\varphi_j, \varphi_i) = \int_0^1 (\mu \varphi_j' \varphi_i' + b \varphi_j' \varphi_i) dx. \quad (2.1.7)$$

Come già osservato in precedenza, per ogni i , $a_{ij} \neq 0$ solo per $j \in \{i-1, i, i+1\}$, come si vede dalla Fig. 1.2.

Calcoliamo quindi i coefficienti non nulli della matrice di rigidità:

$$\begin{aligned}
a(\varphi_{i-1}, \varphi_i) &= \int_{x_{i-1}}^{x_i} (\mu \varphi'_{i-1}(x) \varphi'_i(x) + b \varphi'_{i-1}(x) \varphi_i(x)) \, dx \\
&= \int_{x_{i-1}}^{x_i} \left(-\frac{\mu}{h^2} - \frac{b}{h^2} (x - x_{i-1}) \right) \, dx = -\frac{\mu}{h} - \frac{b}{2}, \\
a(\varphi_i, \varphi_i) &= \int_{x_{i-1}}^{x_{i+1}} \left(\mu (\varphi'_i(x))^2 + b \varphi'_i(x) \varphi_i(x) \right) \, dx \\
&= \int_{x_{i-1}}^{x_i} \frac{\mu}{h^2} \, dx + \int_{x_{i-1}}^{x_{i+1}} b \varphi'_i(x) \varphi_i(x) \, dx \\
&= \frac{2\mu}{h} + \int_{x_{i-1}}^{x_i} \frac{b}{h^2} (x - x_{i-1}) \, dx + \int_{x_i}^{x_{i+1}} -\frac{b}{h^2} (x_{i+1} - x) \, dx = \frac{2\mu}{h}, \\
a(\varphi_{i+1}, \varphi_i) &= \int_{x_i}^{x_{i+1}} \left(-\frac{\mu}{h^2} + \frac{b}{h^2} (x_{i+1} - x) \right) \, dx = -\frac{\mu}{h} + \frac{b}{2}.
\end{aligned}$$

Quanto al termine noto, si avrà $F(\varphi_i) \neq 0$ solo per $i = N - 1$ e

$$F(\varphi_{N-1}) = - \int_0^1 (\mu \varphi'_N(x) \varphi'_{N-1}(x) + b \varphi'_N(x) \varphi_{N-1}(x)) \, dx = \frac{\mu}{h} - \frac{b}{2}.$$

Possiamo quindi finalmente scrivere il sistema per ricavare i valori della u soluzione di (2.0.6) sui nodi della partizione \mathcal{T}_h :

$$\mathbf{A} \mathbf{u} = \mathbf{f} \tag{2.1.8}$$

dove

$$\mathbf{A} = \begin{bmatrix} \frac{2\mu}{h} & -\frac{\mu}{h} + \frac{b}{2} & & & \\ -\frac{\mu}{h} - \frac{b}{2} & \frac{2\mu}{h} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -\frac{\mu}{h} + \frac{b}{2} \\ & & & & & -\frac{\mu}{h} - \frac{b}{2} & \frac{2\mu}{h} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{\mu}{h} - \frac{b}{2} \end{bmatrix}.$$

2.1.3 Sperimentazioni numeriche

Procediamo ora con la risoluzione numerica del sistema (2.1.8); per la risoluzione del sistema lineare abbiamo usato l'algoritmo di Thomas [18] implementato nelle *routines* di MATLAB [14].

Avendo a disposizione la soluzione analitica ricavata in (2.0.4) possiamo facilmente confrontare con essa le soluzioni ottenute numericamente e valutare l'errore commesso. In Fig. 2.2 vediamo che l'errore puntuale si accumula attorno alla *boundary layer*. Rappresentiamo in Fig. 2.3 l'andamento dell'errore globale (in norma L^2) al crescere del numero di gradi di libertà (DoF), ovvero della dimensione dello spazio di approssimazione. Come garantito dalla teoria, al crescere dello spazio di approssimazione l'errore diminuisce, con un andamento simile per vari valori del rapporto $\frac{b}{\mu}$. I valori numerici dell'errore sono invece fortemente dipendenti dal rapporto $\frac{b}{\mu}$ come si può vedere dalla Fig. 2.3.

2.1.4 Un altro esempio numerico

Siccome abbiamo visto che le difficoltà nel problema modello (2.0.1) sono dati dalla presenza di un *boundary layer* nella soluzione, ha senso domandarsi cosa succeda qualora la soluzione presenti un *layer* interno al dominio. Rimaneggiando la funzione di Runge, consideriamo di voler ottenere come soluzione del problema di trasporto e diffusione una funzione del tipo

$$u(x) = \frac{1}{1 + \alpha \left(x - \frac{1}{2}\right)^2} - \frac{1}{1 + \frac{\alpha}{4}}, \quad \alpha \in \mathbb{R}_+, \quad (2.1.9)$$

rappresentata in Fig. 2.4 per alcuni valori di α . Osserviamo che al bordo dell'intervallo la funzione è nulla per costruzione.

Problema differenziale per $u(x)$ Possiamo facilmente costruire un problema differenziale del secondo ordine che abbia $u(x)$ in (2.1.9) come soluzione:

$$\begin{cases} -u''(x) + u'(x) = \frac{\alpha(3-2x)}{[\alpha(x-\frac{1}{2})^2+1]^2} - \frac{2\alpha^2(2x-1)^2}{[\alpha(x-\frac{1}{2})^2+1]^3} := f_\alpha(x) & 0 < x < 1 \\ u(0) = u(1) = 0. \end{cases} \quad (2.1.10)$$

Procedendo come nell'esempio precedente, ricaviamo la formulazione debole:

$$\text{trovare } u \in H_0^1(0,1) : a(u,v) = F(v) \quad \forall v \in H_0^1(0,1), \quad (2.1.11)$$

con $a(u,v) = \int_0^1 (u'v - u'v') dx$ e $F(v) = \int_0^1 f_\alpha v dx$.

Approssimazione con elementi finiti lineari Fissato N costruiamo lo spazio X_h^1 per una partizione a passo costante $h = \frac{1}{N}$ dell'intervallo $[0,1]$. Approssimiamo il problema debole (2.1.11) in X_h^1 :

$$\text{trovare } u_h \in X_h^1 : a(u_h, \varphi_i) = F(\varphi_i), \quad i = 1, \dots, N-1 \quad (2.1.12)$$

da cui ricaviamo come prima una matrice di rigidità tridiagonale. Per ottenere il termine noto del sistema (2.1.12) ricorriamo invece all'integrazione numerica fornita dalle *routines* nel comando `integral` di MATLAB.

Dai risultati riportati in Fig. 2.5 possiamo vedere che anche in questo caso l'errore massimo si localizza nelle zone di forte gradiente della soluzione. Anche per questo esempio in Fig. 2.6 abbiamo studiato la convergenza (in norma L^2). I risultati sono molto simili all'esempio precedente mostrati in Fig. 2.3.

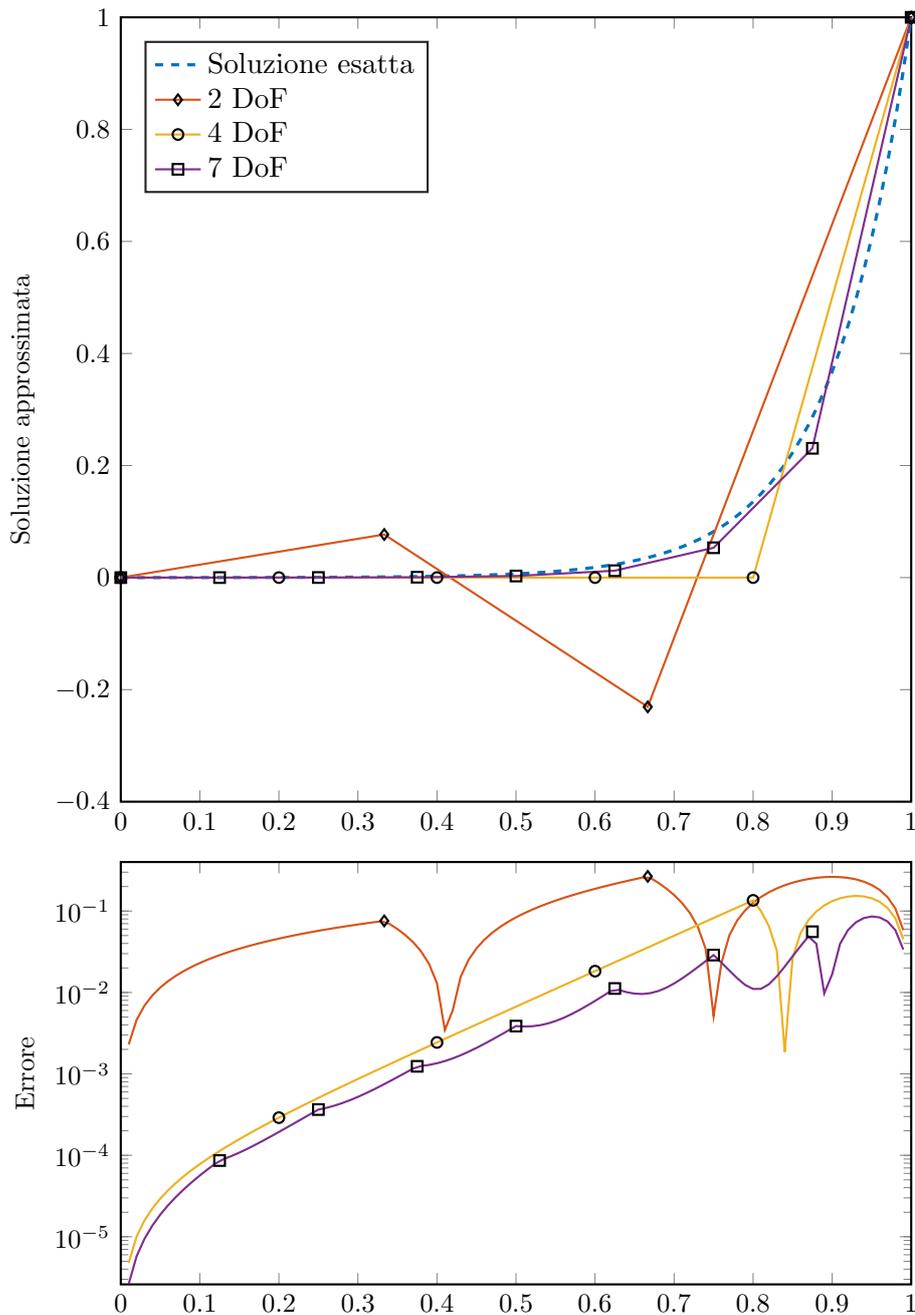


Figura 2.2: Per il problema modello con $\frac{b}{\mu} = 10$ alcune approssimazioni ottenute con metodo Galerkin in X_h^1 aumentando il numero di gradi di libertà. Sopra abbiamo confrontato alcune approssimazioni con la soluzione esatta del problema. Sotto abbiamo rappresentato puntualmente l'errore commesso nell'approssimazione. I punti evidenziati dai marcatori segnalano i nodi della partizione, ovvero i valori u_i ricavati dal sistema lineare (2.1.8).

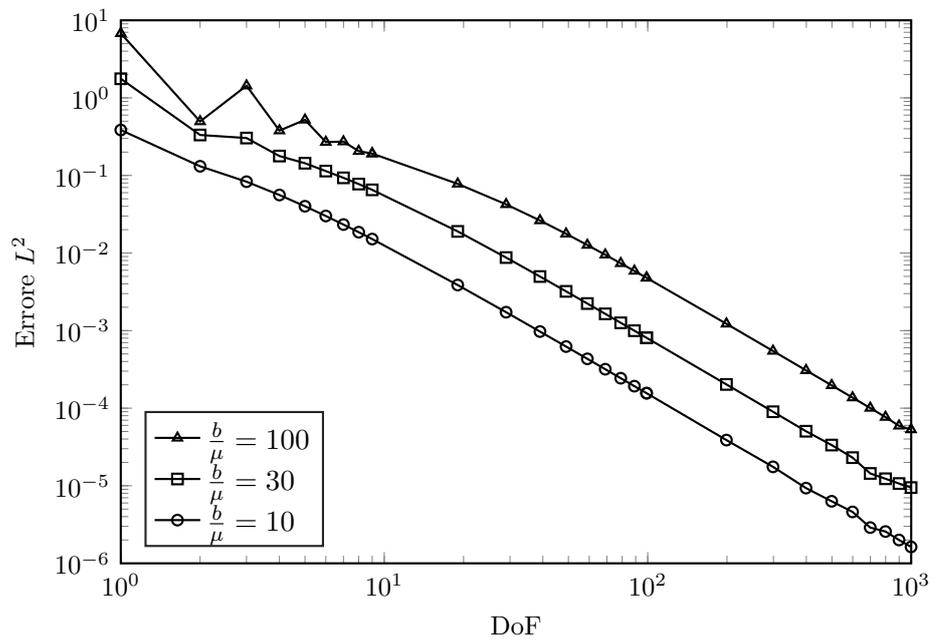


Figura 2.3: Andamento dell'errore nell'approssimazione del problema modello al variare del numero di gradi di libertà per alcuni valori di $\frac{b}{\mu}$.

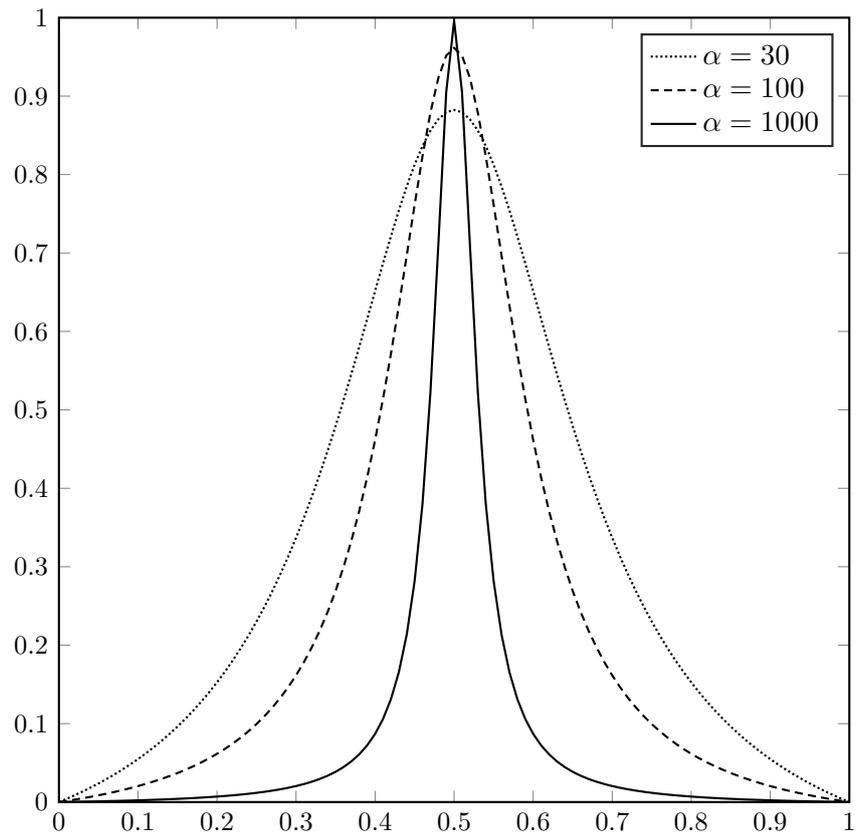


Figura 2.4: Rappresentazione della soluzione (2.1.9) per alcuni valori di α

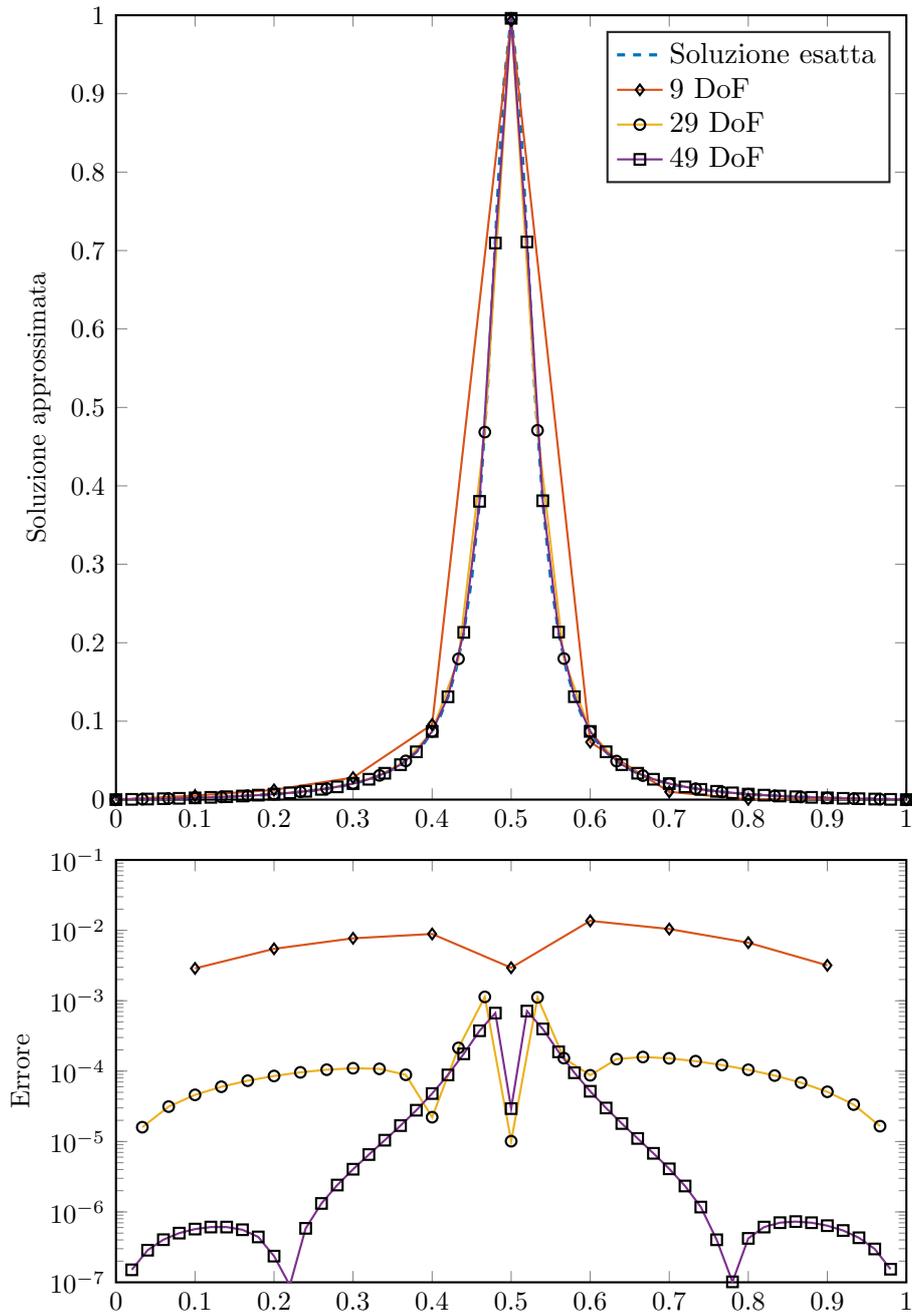


Figura 2.5: Per il problema con *layer* interno con $\alpha = 1000$ alcune approssimazioni ottenute con metodo Galerkin in X_h^1 aumentando il numero di gradi di libertà. Sopra abbiamo confrontato alcune approssimazioni con la soluzione esatta del problema. Sotto abbiamo rappresentato puntualmente l'errore commesso nell'approssimazione. I punti evidenziati dai marcatori segnalano i nodi della partizione.

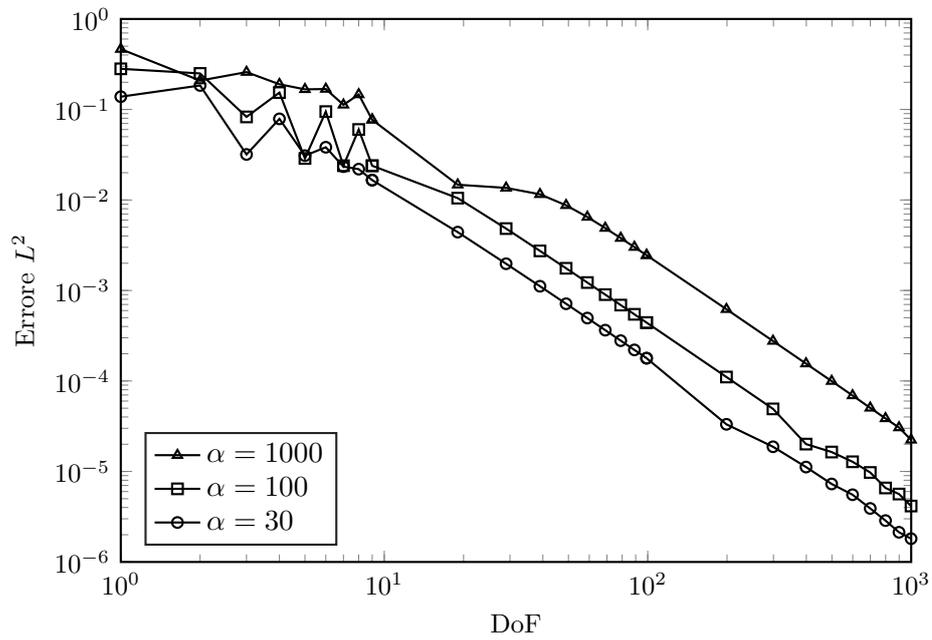


Figura 2.6: Andamento dell'errore nell'approssimazione del problema con *layer* interno al variare del numero di gradi di libertà per alcuni valori di α .

Capitolo 3

Il metodo degli elementi finiti adattivo

Gli esempi presentati nel capitolo precedente mostrano problematiche interessanti, in particolare relativamente alla loro risoluzione numerica. Accade spesso che nella risoluzione numerica di problemi provenienti da modelli della fisica e dell'ingegneria l'accuratezza complessiva dell'approssimazione risulti deteriorata da singolarità locali come, ad esempio, angoli rientranti nel dominio, *layer* interni o al bordo, o fronti d'onda acuti [22].

Il rimedio che proponiamo in questo capitolo è il metodo degli elementi finiti *adattivo* (AFEM), che permetterà di migliorare la precisione complessiva della soluzione numerica andando a raffinare la discretizzazione solo in prossimità delle regioni *critiche*. Naturalmente questo approccio necessita di riuscire ad identificare tali regioni critiche senza aver a disposizione la soluzione del problema.

Queste considerazioni mostrano la necessità di uno stimatore dell'errore calcolabile a partire da quantità note, che fornisca una stima affidabile dell'accuratezza della soluzione numerica calcolata. Vorremo anche essere in grado di stimare i contributi locali all'errore, in modo da capire dove raffinare la discretizzazione.

L'Algoritmo 1 mostra la struttura generale di un metodo FEM adattivo; discuteremo nel seguito nel capitolo le componenti dell'algoritmo.

3.1 Stima residuale dell'errore

La classica analisi degli errori del FEM fornisce stime a priori dell'errore che valgono solo asintoticamente per $h \rightarrow 0$ e danno informazioni globali [4]. Come abbiamo già rilevato, per l'algoritmo adattivo è necessario disporre di uno stimatore dell'errore che sia significativo ad ogni passo e che dia informazioni locali [3]. Presentiamo di seguito un semplice stimatore a posteriori che si ricava una volta calcolata un'approssimazione della soluzione.

Sia $u_h \in V_h$ soluzione esatta di (1.2.1), approssimazione agli elementi finiti di $u \in V$ soluzione di (1.1.2); denotiamo con $e_h = u - u_h$ l'errore commesso nell'approssimazione, abbiamo

$$a(e_h, v) = \mathcal{F}(v) - a(u_h, v) = R(v) \quad \forall v \in V, \quad (3.1.1)$$

Algoritmo 1 Algoritmo adattivo generale

Input: I dati del problema differenziale, una partizione iniziale \mathcal{T}_0 , una tolleranza ϵ , un numero massimo di iterazioni k_{\max}

Output: Una soluzione approssimata del problema con errore stimato minore di ϵ , se possibile; altrimenti l'approssimazione ottenuta in k_{\max} iterazioni

Poniamo $\eta = +\infty$

per $k = 0 \dots k_{\max}$

Risolvi il problema discretizzato sulla partizione \mathcal{T}_k

per ogni $K \in \mathcal{T}_k$

Calcola il residuo locale η_K su K

fine per

$\eta \leftarrow \left(\sum_{K \in \mathcal{T}_k} \eta_K^2 \right)^{\frac{1}{2}}$ stima l'errore globale

se $\eta \leq \epsilon$ **allora**

Si è raggiunta l'accuratezza richiesta, il procedimento termina

fine se

In base ai residui locali $(\eta_K)_{K \in \mathcal{T}_k}$ determinare un sottoinsieme $\tilde{\mathcal{T}}_k$ di elementi da raffinare

In base a $\tilde{\mathcal{T}}_k$ determinare una nuova partizione \mathcal{T}_{k+1} tale che $\mathcal{N}_{k+1} \supset \mathcal{N}_k$

$k \leftarrow k + 1$

fine per

che definisce implicitamente il funzionale $R \in V'$, detto *residuo*. Dalla relazione di ortogonalità di Galerkin (1.2.3) si ha immediatamente che il residuo si annulla su V_h .

Equivalenza fra errore e residuo Calcolando la (3.1.1) con $v = e_h$, grazie alla coercività della forma bilineare a (1.1.4) possiamo stimare

$$\alpha \|e_h\|^2 \leq |a(e_h, e_h)| = |R(e_h)| \leq \|R\|_{V'} \|e_h\|.$$

Lo spazio V è di Hilbert quindi di Banach e riflessivo, pertanto¹ il funzionale R assume massimo sulla palla unitaria, e siccome $R(v_h) = 0$, $\forall v_h \in V_h$ sottospazio di V il massimo sarà non negativo; sia quindi $w \in V$ con $\|w\| \leq 1$ tale che $R(w) = \|R\|_{V'}$. Calcolando la relazione (3.1.1) su w tenendo conto della continuità di a (1.1.3) si ricava

$$\|R\|_{V'} = R(w) = a(e_h, w) \leq \gamma \|e_h\| \|w\| \leq \gamma \|e_h\|$$

e quindi complessivamente

$$\frac{1}{\gamma} \|R\|_{V'} \leq \|e_h\| \leq \frac{1}{\alpha} \|R\|_{V'}. \quad (3.1.2)$$

Anche se il residuo è esplicitamente disponibile una volta determinata una approssimazione u_h di u , bisogna notare che il calcolo della norma duale in (3.1.2) è spesso difficile, cercheremo

¹Fatto generale sugli spazi riflessivi. Una caratterizzazione degli spazi riflessivi in questi termini viene data dal teorema di James[12].

pertanto di stimare la norma del residuo mediante norme in spazi di Lebesgue [21], più facilmente calcolabili. Un'altra difficoltà deriva dal fatto che le disuguaglianze (3.1.2) valgono solo globalmente e non sono immediatamente riconducibili a quantità locali.

3.1.1 Rappresentazione esplicita del residuo

La costruzione esplicita del residuo dipende in modo essenziale dal problema; ci occupiamo quindi nel seguito di trovare una rappresentazione del residuo relativamente a problemi di diffusione-trasposto nella forma del nostro problema modello (2.0.1). Per una trattazione più generale si può vedere [21].

Consideriamo il problema modello in forma debole con condizioni omogenee (2.0.8) ed una soluzione approssimata agli elementi finiti $u_h \in V_h = \dot{X}_h^r(0, 1)$, $v \in V = H_0^1(0, 1)$:

$$\begin{aligned} R(v) &= F(v) - a(u_h, v) = -a(B, v) - a(u_h, v) = -a(B + u_h, v) \\ &= - \int_0^1 (\mu(B' + u_h') v' + b(B' + u_h') v) \, dx. \end{aligned} \quad (3.1.3)$$

Possiamo quindi suddividere il residuo nei contributi locali sugli intervalli della partizione $\mathcal{T}_h = \{I_k = [x_{k-1}, x_k], k = 1, \dots, N\}$

$$\begin{aligned} R(v) &= - \sum_{I_k \in \mathcal{T}_h} \left[\int_{I_k} (\mu(B' + u_h') v' + b(B' + u_h') v) \, dx \right] \\ &= \sum_{I_k \in \mathcal{T}_h} \left[\int_{I_k} R_A v \, dx - [\mu(B' + u_h') v]_{\partial I_k} \right], \end{aligned} \quad (3.1.4)$$

avendo definito per brevità di notazione

$$R_A = \mu(B'' + u_h'') - b(B' + u_h') \quad (3.1.5)$$

e indicando

$$[\mu(B' + u_h') v]_{\partial I_k} = \lim_{t \rightarrow x_k^-} \mu(B'(t) + u_h'(t))v(t) - \lim_{t \rightarrow x_{k-1}^+} \mu(B'(t) + u_h'(t))v(t)$$

il salto all'interfaccia fra elementi della partizione \mathcal{T}_h .

Osservazione 3.1.1. Se il problema differenziale presenta anche un termine noto f , con passaggi analoghi otteniamo per il residuo la seguente espressione

$$R(v) = \sum_{I_k \in \mathcal{T}_h} \left[\int_{I_k} R_A v \, dx - [\mu u_h' v]_{\partial I_k} \right], \quad (3.1.6)$$

con

$$R_A = f + \mu(B'' + u_h'') - b(B' + u_h') \quad (3.1.7)$$

e

$$[\mu u_h' v]_{\partial I_k} = \lim_{t \rightarrow x_k^-} \mu u_h'(t) v(t) - \lim_{t \rightarrow x_{k-1}^+} \mu u_h'(t) v(t).$$

Nel seguito considereremo questo caso più generale.

3.1.2 Stima superiore dell'errore

Si è già rilevato che $R(v_h) = 0$, $\forall v_h \in V_h$; sia quindi $v_h = \Pi_{\mathcal{T}_h}^1 v$ l'interpolante lineare di $v \in H_0^1$ sulla partizione \mathcal{T}_h (1.3.4):

$$R(v) = R(v - v_h) = \sum_{I_k \in \mathcal{T}_h} \int_{I_k} R_A(v - v_h) \, dx, \quad (3.1.8)$$

dove la prima uguaglianza vale per linearità e la seconda poiché $v(x_i) = v_h(x_i)$ su tutti i nodi della partizione. Applichiamo quindi la diseguaglianza di Hölder, la stima dell'errore di interpolazione (1.3.6) e la disuguaglianza di Cauchy-Schwarz per le somme

$$\begin{aligned} |R(v)| &\leq \sum_{I_k \in \mathcal{T}_h} \|R_A\|_{L^2(I_k)} \|v - v_h\|_{L^2(I_k)} \leq \sum_{I_k \in \mathcal{T}_h} \|R_A\|_{L^2(I_k)} Ch_k \|v\|_{H^1(I_k)} \\ &\leq \left(\sum_{I_k \in \mathcal{T}_h} C^2 h_k^2 \|R_A\|_{L^2(I_k)}^2 \right)^{\frac{1}{2}} \left(\sum_{I_k \in \mathcal{T}_h} \|v\|_{H^1(I_k)}^2 \right)^{\frac{1}{2}} \\ &= \left(\sum_{I_k \in \mathcal{T}_h} C^2 h_k^2 \|R_A\|_{L^2(I_k)}^2 \right)^{\frac{1}{2}} \|v\|_{H^1(\Omega)} \\ &= C \left(\sum_{I_k \in \mathcal{T}_h} \eta_{R,k}^2 \right)^{\frac{1}{2}} \|v\|_{H^1(\Omega)}, \end{aligned} \quad (3.1.9)$$

avendo definito il *residuo locale*

$$\eta_{R,k} = h_k \|R_A\|_{L^2(I_k)} = h_k \|f + \mu(B'' + u_h'') - b(B' + u_h')\|_{L^2(I_k)}. \quad (3.1.10)$$

Combinando (3.1.10) e (3.1.2), per l'arbitrarietà di $v \in H_0^1(0, 1)$ abbiamo la stima dell'errore

$$\|e_h\| \leq \frac{C}{\alpha} \left(\sum_{I_k \in \mathcal{T}_h} \eta_{R,k}^2 \right)^{\frac{1}{2}}. \quad (3.1.11)$$

Espressione del residuo locale nel caso lineare Nel caso lineare, ovvero $V_h = X_h^1$, gli integrali nella norma L^2 in (3.1.10) sono piuttosto semplici da calcolare simbolicamente. Per $k = 2, \dots, N - 1$ il contributo del rilevamento dei dati al bordo è nullo e quindi, ricordando la definizione delle φ_i in (1.3.3), abbiamo:

$$\begin{aligned} \eta_{R,k} &= h_k \left(\int_{I_k} |f - b(u_h')|^2 \, dx \right)^{\frac{1}{2}} \\ &= h_k \left(\int_{I_k} |f - b(u_{k-1}\varphi'_{k-1} + u_k\varphi'_k)|^2 \, dx \right)^{\frac{1}{2}} \\ &= h_k \left(\int_{I_k} \left| f - \frac{b}{h_k}(u_k - u_{k-1}) \right|^2 \, dx \right)^{\frac{1}{2}}. \end{aligned} \quad (3.1.12)$$

Per $k = 1$ e $k = N$ invece si ha anche il contributo dovuto al rilevamento dei dati al bordo, ma u_h è formata da un solo tratto, quindi abbiamo:

$$\begin{aligned}
\eta_{R,1} &= h_1 \left(\int_{I_1} |f - b(B' + u'_h)|^2 dx \right)^{\frac{1}{2}} \\
&= h_1 \left(\int_{I_1} |f - b(u_0\varphi'_0 + u_1\varphi'_1)|^2 dx \right)^{\frac{1}{2}} \\
&= h_1 \left(\int_{I_1} \left| f - \frac{b}{h_1}(u_1 - u_0) \right|^2 dx \right)^{\frac{1}{2}}, \\
\eta_{R,N} &= h_N \left(\int_{I_N} |f - b(B' + u'_h)|^2 dx \right)^{\frac{1}{2}} \\
&= h_N \left(\int_{I_N} |f - b(u_N\varphi'_N + u_{N-1}\varphi'_{N-1})|^2 dx \right)^{\frac{1}{2}} \\
&= h_N \left(\int_{I_N} \left| f - \frac{b}{h_N}(u_N - u_{N-1}) \right|^2 dx \right)^{\frac{1}{2}}.
\end{aligned} \tag{3.1.13}$$

Notiamo quindi che complessivamente l'espressione (3.1.12) vale per $k = 1, \dots, N$.

3.2 Tecniche di raffinamento

Il procedimento adattivo richiede un metodo per costruire un raffinamento di una data partizione basato sulle stime locali dell'errore. Gli ingredienti fondamentali sono

1. Una *marking strategy* che stabilisce quali siano gli elementi da raffinare basandosi sull'errore locale.
2. Regole di raffinamento che determinano come ogni elemento marcato al passaggio precedente debba essere suddiviso per ottenere un raffinamento della partizione data. Nel caso monodimensionale in cui stiamo lavorando il raffinamento viene in genere effettuato per bisezione degli intervalli marcati.

3.2.1 *Marking strategies*

Presentiamo di seguito due popolari *marking strategies*: la *strategia del massimo* e quella dell'*equilibrio* [21]. Presentiamo quindi due strategie aggiuntive per un controllo più diretto della procedura di *marking*.

Strategia del massimo La strategia del massimo schematizzata nell'Algoritmo 2 seleziona tutti gli elementi che hanno residuo locale maggiore o uguale del residuo massimo moltiplicato per una costante $\theta \in (0, 1)$. Osserviamo che scegliendo $\theta \approx 0$ raffiniamo quasi globalmente, pertanto effettueremo poche iterazioni dell'algoritmo adattivo ma con molti gradi di libertà inutili, il

Algoritmo 2 Strategia del massimo

Input: Una partizione \mathcal{T} , stime dell'errore locale $(\eta_K)_{K \in \mathcal{T}}$, una soglia $\theta \in (0, 1)$.**Output:** Un sottoinsieme $\tilde{\mathcal{T}} \subset \mathcal{T}$ di elementi da raffinare. $\tilde{\mathcal{T}} \leftarrow \emptyset$ $\eta \leftarrow \max_{K \in \mathcal{T}} \eta_K$ **per ogni** $K \in \mathcal{T}$ **se** $\eta_K \geq \theta \eta$ **allora** $\tilde{\mathcal{T}} \leftarrow \tilde{\mathcal{T}} \cup \{K\}$ **fine se****fine per**

Algoritmo 3 Strategia dell'equilibrio o di Dörfler [7]

Input: Una partizione \mathcal{T} , stime dell'errore locale $(\eta_K)_{K \in \mathcal{T}}$, una soglia $\theta \in (0, 1)$.**Output:** Un sottoinsieme $\tilde{\mathcal{T}} \subset \mathcal{T}$ di elementi da raffinare. $\tilde{\mathcal{T}} \leftarrow \emptyset, \Sigma \leftarrow 0, \Theta \leftarrow \sum_{K \in \mathcal{T}} \eta_K^2$ **finché** $\Sigma < \theta \Theta$ $\eta \leftarrow \max_{K \in \mathcal{T} \setminus \tilde{\mathcal{T}}} \eta_K$ **per ogni** $K \in \mathcal{T} \setminus \tilde{\mathcal{T}}$ **se** $\eta_K = \eta$ **allora** $\tilde{\mathcal{T}} \leftarrow \tilde{\mathcal{T}} \cup \{K\}, \Sigma \leftarrow \Sigma + \eta_K^2$ **fine se****fine per****fine finché**

che va contro all'idea stessa dell'adattività. D'altra, fissando $\theta \approx 1$ andremo a selezionare pochi elementi ad ogni iterazione, e se, come spesso accade, l'errore è distribuito in modo non omogeneo, sarà necessario effettuare molte iterazioni per abbattere significativamente l'errore, e si avrà quindi un aumento del costo computazionale della procedura dovuto alle molteplici risoluzioni ad elementi finiti.

Strategia dell'equilibrio La strategia schematizzata nell'Algoritmo 3 è stata sviluppata in [7] per risolvere le debolezze della strategia del massimo elencate nel paragrafo precedente. In questo caso l'idea è di selezionare solo gli elementi della partizione che danno il maggior contributo all'errore globale. Alla fine della procedura avremo selezionato un insieme $\tilde{\mathcal{T}}$ tale che

$$\sum_{K \in \tilde{\mathcal{T}}} \eta_K^2 \geq \theta \sum_{K \in \mathcal{T}} \eta_K^2,$$

e quindi è garantita una riduzione dell'errore globale stimato.

Premarking Entrambe le strategie risultano poco efficaci quando una piccola parte degli elementi della partizione hanno errore stimato molto più alto di tutti gli altri. In questo caso infatti sfuggono alla marcatura non solo gli elementi con errore stimato molto piccolo, ma anche quelli che hanno errore stimato relativamente piccolo, ma abbastanza grande da deteriorare la precisione dell'algoritmo adattivo. Per ovviare a questo inconveniente è in generale consigliabile stabilire una piccola percentuale p di elementi da marcare a priori, e marcare quindi il $p\%$ degli elementi che hanno massimo errore stimato per poi sottoporre ad una procedura di marcatura tutti i restanti.

Sensibilità alla partizione iniziale Le strategie di *marking* fino ad ora esposte sono volte ad abbassare il residuo globale prendendo in esame il residuo locale; questo porta in molti casi ad una forte dipendenza dalla scelta della partizione iniziale. Per rendere più stabile e prevedibile la procedura di raffinamento, è possibile introdurre un marcatore a soglia, che marca per il raffinamento tutti gli elementi della partizione che hanno residuo locale maggiore di una certa soglia σ .

Con l'introduzione di un criterio di *marking* così stretto siamo certi di ottenere approssimazioni che localmente raggiungono elevati livelli di precisione con poche iterazioni dell'algoritmo adattivo, il prezzo da pagare è però un aumento molto più rapido del numero di gradi di libertà. La scelta della soglia σ è piuttosto delicata: una soglia troppo bassa (relativamente al problema e alla scelta di ϵ) fa degenerare l'algoritmo adattivo in una successione - computazionalmente onerosa - di raffinamenti quasi uniformi. Suggeriamo di basare la scelta sul problema in esame, a partire ad esempio da informazioni ottenute con una prima approssimazione grossolana (come mostrato nell'esempio nel listato A.4).

3.3 Esempi numerici

In questa sezione testiamo l'algoritmo adattivo in tutte le sue componenti sul problema modello di diffusione-trasporto. Proviamo quindi alcune simulazioni anche sul problema con *layer* in-

terno al dominio (2.1.10) e su un nuovo problema con soluzione oscillante. Le sperimentazioni numeriche riportate sono state quasi integralmente condotte con i codici riportati in appendice A.

3.3.1 Problema modello di trasporto-diffusione

Confronto fra lo stimatore residuale e l'errore analitico Abbiamo visto formalmente che errore e residuo risultano equivalenti nelle rispettive norme naturali, si è inoltre stimata la norma del residuo attraverso quantità più facilmente calcolabili (3.1.11). Avendo a disposizione la soluzione analitica del problema (2.0.4) possiamo confrontare il residuo locale con l'errore ottenuto analiticamente.

Nella Fig. 3.1 possiamo osservare che la correlazione fra l'errore effettivo e lo stimatore $\eta_{R,k}$ in (3.1.10) è piuttosto forte e diventa man mano più chiara al crescere di N , cioè della finezza della partizione. Siccome saremo interessati a stimare l'errore su griglie non uniformi, verificiamo che quanto osservato fino ad ora valga anche per questo tipo di partizioni in Fig. 3.2. Chiaramente lo stimatore segue molto bene l'andamento dell'errore anche in questo caso.

Marcatori e partizioni Esaminiamo la dipendenza del risultato finale dell'algoritmo adattivo dalla partizione iniziale: in Fig. 3.3, a partire da una stessa partizione uniforme, con gli stessi marcatori otteniamo risultati piuttosto differenti, soprattutto per quanto riguarda la partizione finale del dominio e la distribuzione dell'errore. Ripetendo in Fig. 3.4 l'esperimento appena illustrato avendo introdotto il marcatore a soglia $\sigma = 10^{-3}$, vediamo che i due risultati sono molto più omogenei per numero di gradi di libertà finali, stima dei residui, distribuzione dell'errore e partizioni costruite. Osserviamo che la distribuzione puntuale dell'errore risulta meno omogenea rispetto a Fig. 3.3a, ed è più simile a Fig. 3.3b. In effetti però entrambi i risultati ottenuti con il marcatore a soglia sono piuttosto simili ad una partizione uniforme. Per evitare questo fenomeno e mettere in risalto le caratteristiche del metodo adattivo abbiamo evitato di utilizzare questo marcatore nei restanti esempi, ove non specificato.

Raffinamenti adattivi Rappresentiamo in Fig.3.5 i successivi raffinamenti dello spazio di discretizzazione in cui costruiamo la soluzione approssimata. La figura è sempre basata sul problema modello con $\frac{b}{\mu} = 100$, avendo marcato gli elementi con il solo marcatore di Dörfler con $\theta = 0.5$.

Risultati Avendo verificato che tutte le componenti dell'algoritmo funzionano come previsto, possiamo confrontare i risultati ottenibili con il metodo adattivo rispetto ai raffinamenti uniformi della sezione precedente. In Fig. 3.6 possiamo vedere chiaramente che il metodo adattivo rende possibile un abbattimento drastico dell'errore con un aumento modesto del numero di gradi di libertà. Osserviamo che in questo confronto il rapporto $\frac{b}{\mu}$ non è eccessivamente discriminante.

Si osserva anche che le prime iterazioni dei due metodi portano a risultati essenzialmente identici in termini di errore. Questo fatto è abbastanza naturale: quando la partizione è molto larga l'algoritmo adattivo effettua qualche raffinamento uniforme, solo dopo alcune iterazioni i

due metodi differiscono in modo sostanziale. La differenza fra i due metodi è più evidente al crescere del rapporto $\frac{b}{\mu}$, ovvero all'evidenziarsi del *boundary layer*.

3.3.2 Problema con *layer* interno

Consideriamo il problema costruito per avere un *layer* interno al dominio (2.1.10).

Stima residuale dell'errore Per questo problema il residuo locale è definito da

$$\eta_{R,k}^2 = h_k^2 \|f_\alpha - u'_h - u''_h\|_{L^2(I_k)}^2, \quad (3.3.1)$$

che nel caso lineare, cioè con $u_h \in X_h^1(0,1)$, possiamo esprimere come

$$\begin{aligned} \eta_{R,k}^2 &= h_k^2 \int_{x_{k-1}}^{x_k} |f_\alpha - u'_h|^2 dx \\ &= h_k^2 \int_{x_{k-1}}^{x_k} |f_\alpha - (u_{k-1}\varphi'_{k-1} + u_k\varphi'_k)|^2 dx \\ &= h_k^2 \int_{x_{k-1}}^{x_k} \left| f_\alpha - \left(\frac{u_{k-1}}{-h_k} + \frac{u_k}{h_k} \right) \right|^2 dx \\ &= h_k^2 \int_{x_{k-1}}^{x_k} \left| f_\alpha - \frac{1}{h_k} (u_k - u_{k-1}) \right|^2 dx. \end{aligned}$$

Anche in questo caso possiamo vedere una chiara aderenza fra la distribuzione dell'errore locale in norma e quella del residuo locale in Fig. 3.7. In Fig. 3.8 confrontiamo un risultato ottenuto con il metodo adattivo rispetto all'analogo risultato prodotto dal FEM classico: a partire dalla stessa prima approssimazione i due metodi producono risultati piuttosto diversi con lo stesso numero di gradi di libertà. Osserviamo che l'algoritmo adattivo ha raffinato la partizione proprio nelle zone di forte gradiente della soluzione. Nella Fig. 3.9 confrontiamo più quantitativamente il metodo adattivo e il FEM; come nell'esempio precedente il metodo adattivo risulta particolarmente vantaggioso quando il *layer* è più marcato.

3.3.3 Problema con soluzione oscillante

Un altro caso interessante da esaminare è quello in cui la soluzione del problema presenti numerosi strati interni o al bordo del dominio. Consideriamo ad esempio una funzione oscillante della forma

$$u(t) = (\cos(kx))^\alpha, \quad (3.3.2)$$

dove il parametro $k \in \mathbb{N}$ determina il numero di oscillazioni, mentre dal parametro $\alpha \geq 1$ dipende l'ampiezza delle delle onde; in particolare per α grandi le onde diventano sempre meno lisce e degenerano in singolarità locali, come vediamo in Fig. 3.10. Possiamo facilmente verificare che la u definita in (3.3.2) è soluzione del problema differenziale

$$\begin{cases} -u''(x) + u'(x) = -k\alpha \cos^{\alpha-2}(kx) [k\alpha \sin^2(kx) - k + \cos(kx) \sin(kx)] \\ u(0) = 1, \quad u(\pi) = -1. \end{cases} \quad (3.3.3)$$

Risolviamo quindi il problema (3.3.3) con i metodi FEM e AFEM e confrontiamo i risultati. Vediamo in Fig. 3.11 il caso in cui la soluzione è liscia: i due metodi risolutivi conducono a discretizzazioni del dominio indistinguibili. Analogamente l'andamento dell'errore in norma L^2 (Fig. 3.13a) mostra che in questo caso i raffinamenti uniformi risultano preferibili, conducendo a livelli di precisione almeno uguali a quelli adattivi, comportando però un costo minore per la loro costruzione.

Aumentando il valore del parametro α le onde diventano sempre più strette e degenerano in singolarità locali. In questo caso il metodo adattivo inizia a differenziarsi dai raffinamenti uniformi appena mostrati, fornendo discretizzazioni più mirate (Fig. 3.12). Vediamo in Fig. 3.13b che al crescere di α il metodo adattivo diventa più vantaggioso portando ad una riduzione dell'errore di oltre un ordine di grandezza a parità del numero di gradi di libertà.

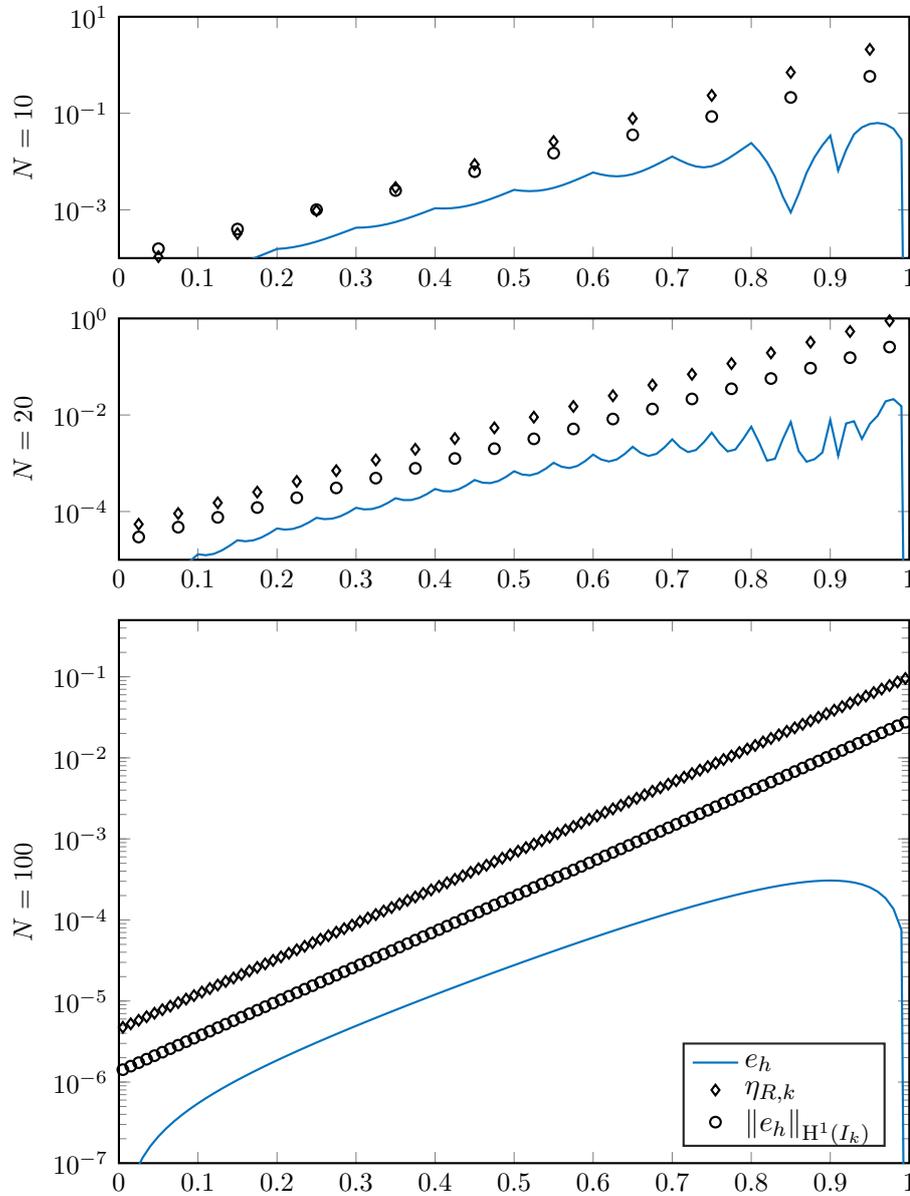


Figura 3.1: Rappresentazione della distribuzione dell'errore puntuale e_h e in norma su una partizione uniforme per il problema modello con $\frac{b}{\mu} = 10$. L'errore in norma e il residuo sono rappresentati al centro dell'intervallo su cui sono calcolati.

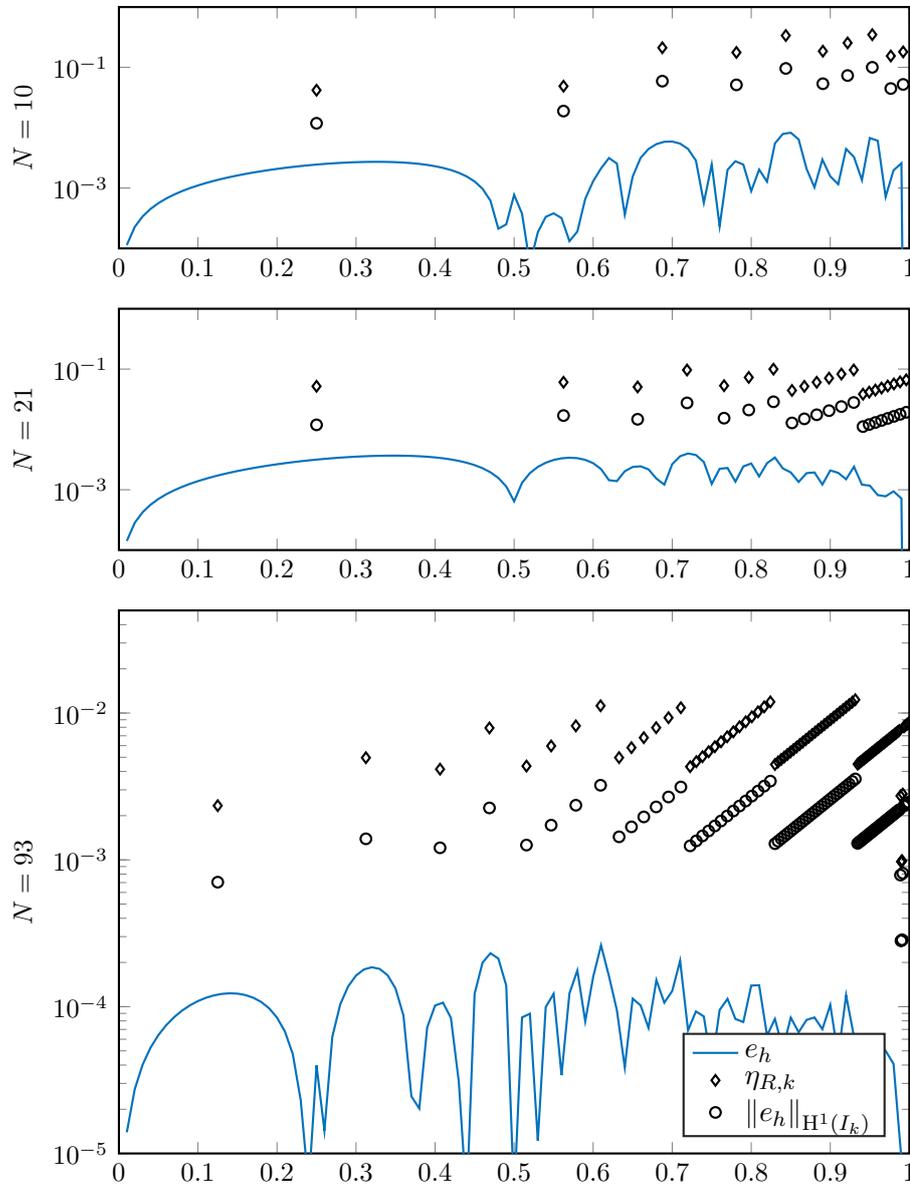
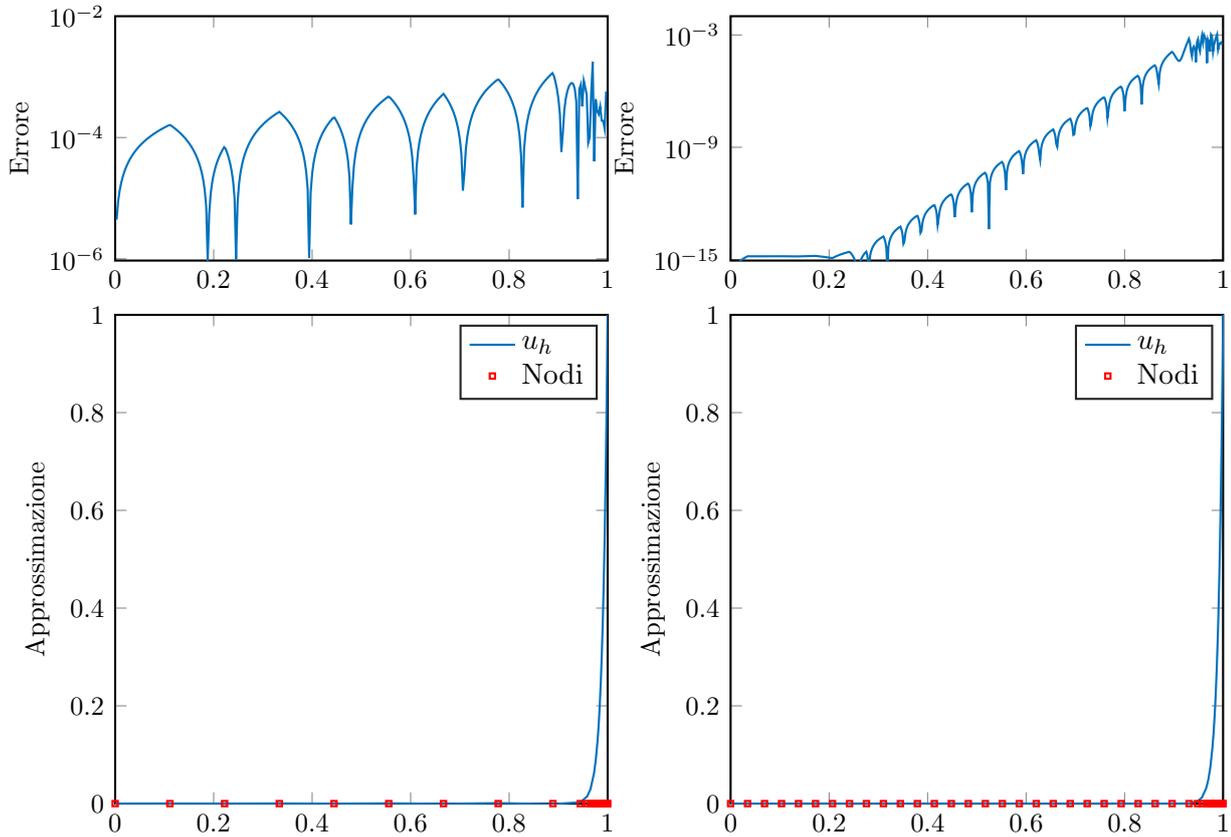


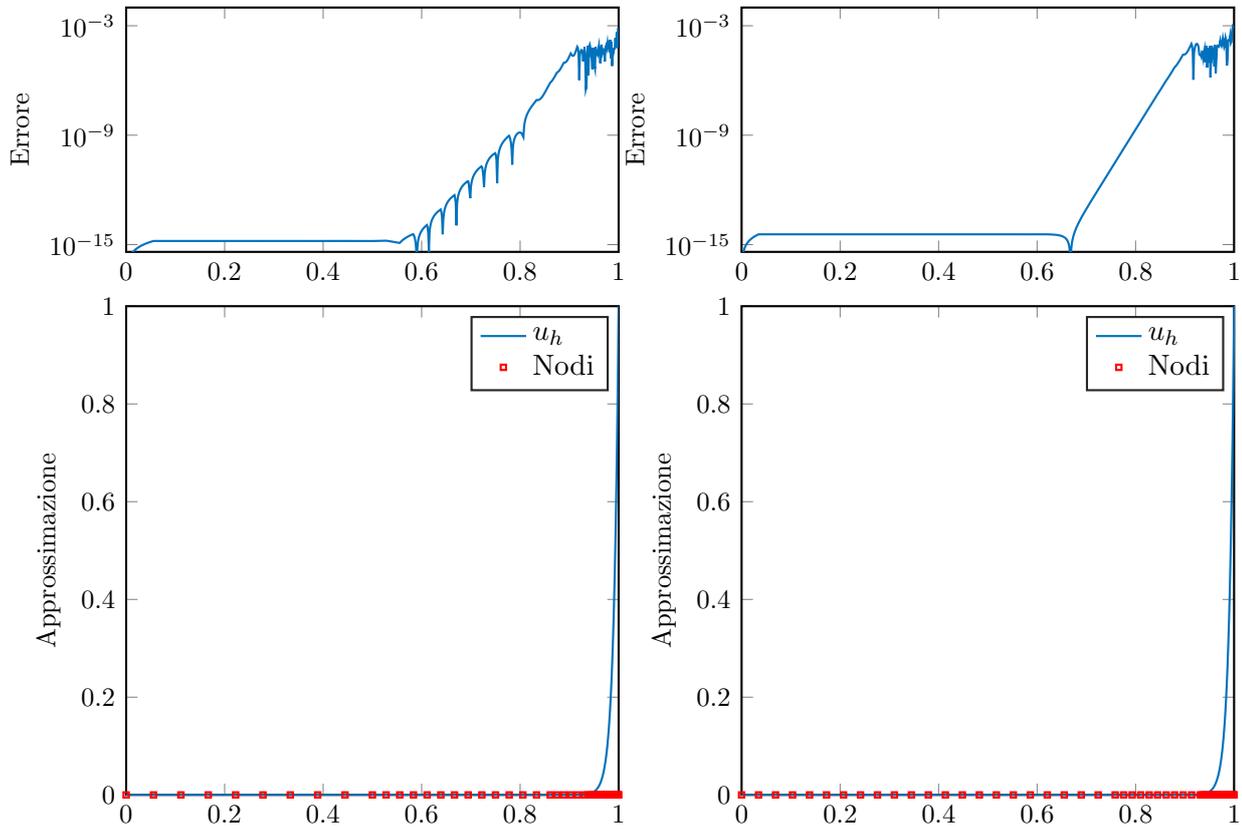
Figura 3.2: Rappresentazione della distribuzione dell'errore puntuale e_h e in norma su una partizione adattiva per il problema modello con $\frac{b}{\mu} = 10$. L'errore in norma e il residuo sono rappresentati al centro dell'intervallo su cui sono calcolati.



(a) Il procedimento iniziato con 10 DoF ha terminato a 33 DoF in 10 iterazioni con errore locale stimato fra 0.000054 e 0.002643, errore globale stimato 0.008290.

(b) Il procedimento iniziato con 30 DoF ha terminato a 49 DoF in 5 iterazioni con errore locale stimato fra 0 e 0.003357, errore globale stimato 0.009508.

Figura 3.3: Dipendenza del risultato dalla partizione iniziale per il problema modello con $\frac{b}{\mu} = 100$. In questo esperimento abbiamo eseguito *premarking* al 5%, marcato gli elementi restanti con marcatore Dörfler con $\theta = 0.5$; le diverse scelte della partizione iniziale conducono a diverse partizioni finali e diverse distribuzioni dell'errore.



(a) Il procedimento iniziato con 10 DoF ha terminato a 75 DoF in 7 iterazioni con errore locale stimato fra 0 e 0.002451, errore globale stimato 0.00618.

(b) Il procedimento iniziato con 30 DoF ha terminato a 75 DoF in 5 iterazioni con errore locale stimato fra 0 e 0.003357, errore globale stimato 0.007642.

Figura 3.4: Abbiamo ripetuto l'esperimento mostrato nella precedente figura 3.3 imponendo il raffinamento degli elementi con residuo locale maggiore di $\sigma = 10^{-3}$. Con questo accorgimento i due risultati sono più simili, si è quindi diminuita la dipendenza dalla scelta della partizione iniziale \mathcal{T}_0 ; rileviamo però un aumento considerevole del numero di gradi di libertà già con poche iterazioni dell'algoritmo.

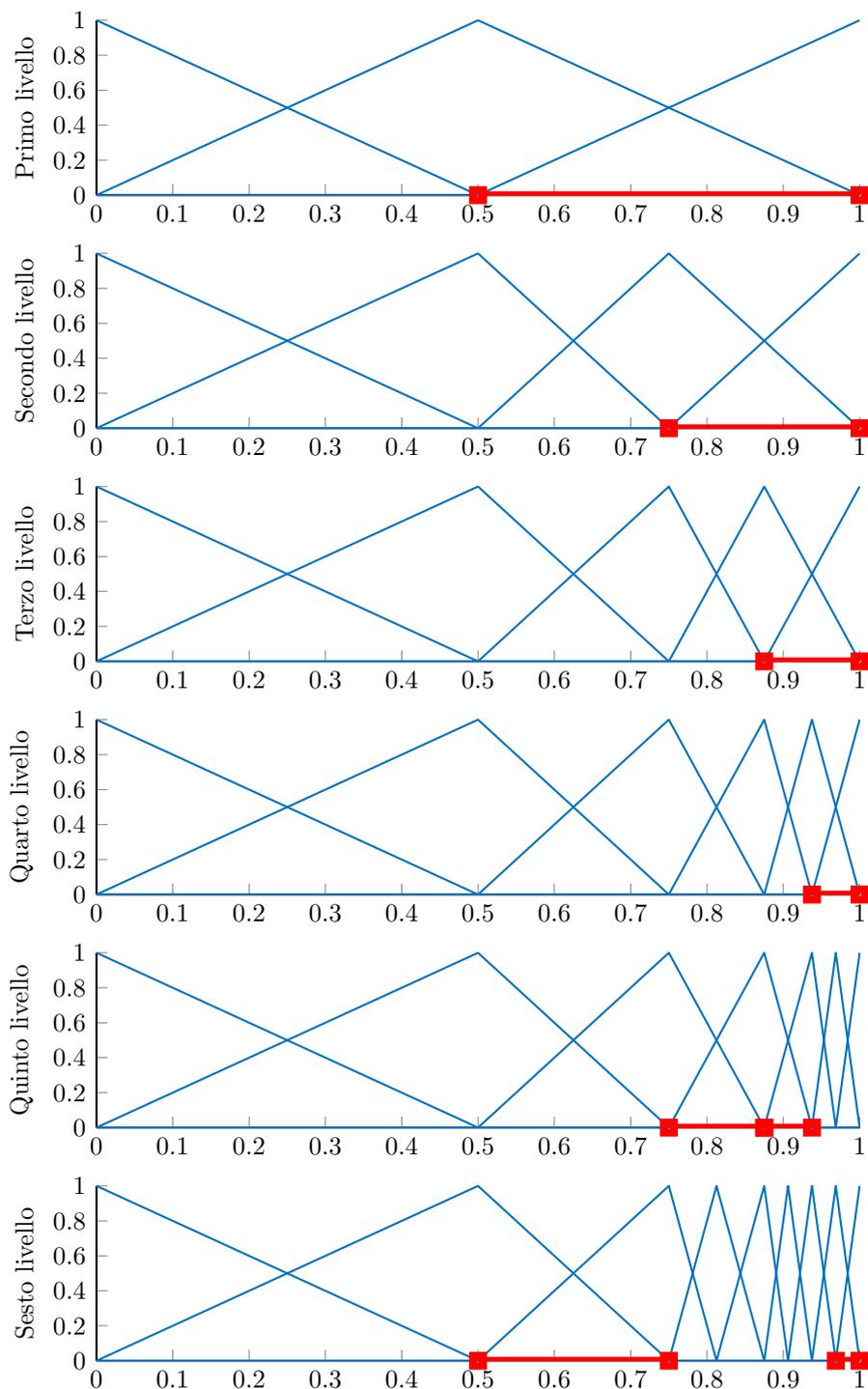


Figura 3.5: Raffinamenti adattivi successivi dello spazio X_h^1 in cui cerchiamo la soluzione approssimata del problema modello di trasporto-diffusione. La scelta degli elementi da raffinare è affidata al marcatore di Dörfler per $\theta = 0.1$ senza *premarking*. Ad ogni livello abbiamo segnato gli elementi marcati e il loro raffinamento nel livello successivo.

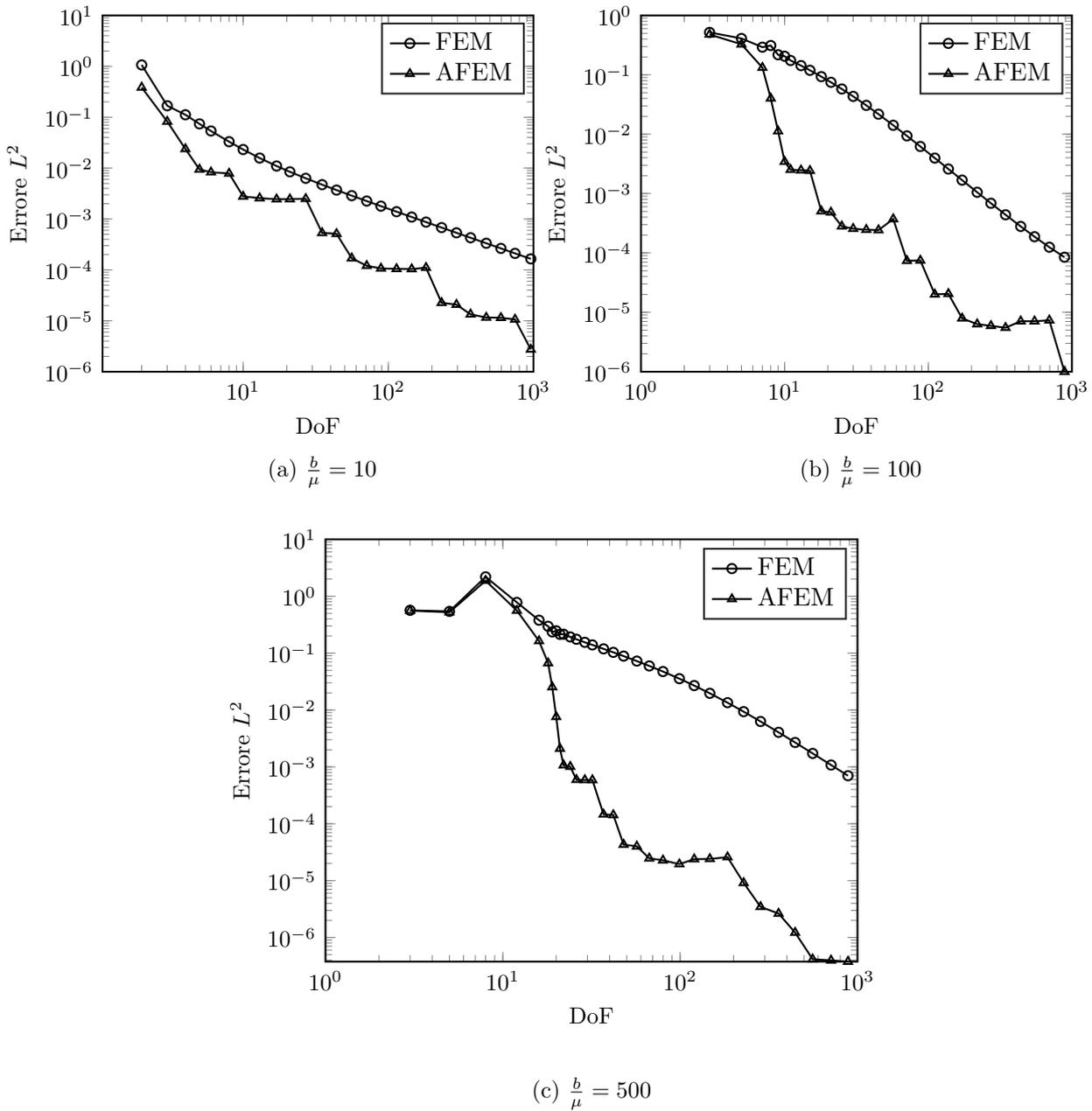


Figura 3.6: Confronto della convergenza FEM vs. AFEM per il problema modello. Nell'esempio mostrato l'algoritmo adattivo utilizza il marcatore di Dörfler con $\theta = 0.5$ e nessun *premarking*.

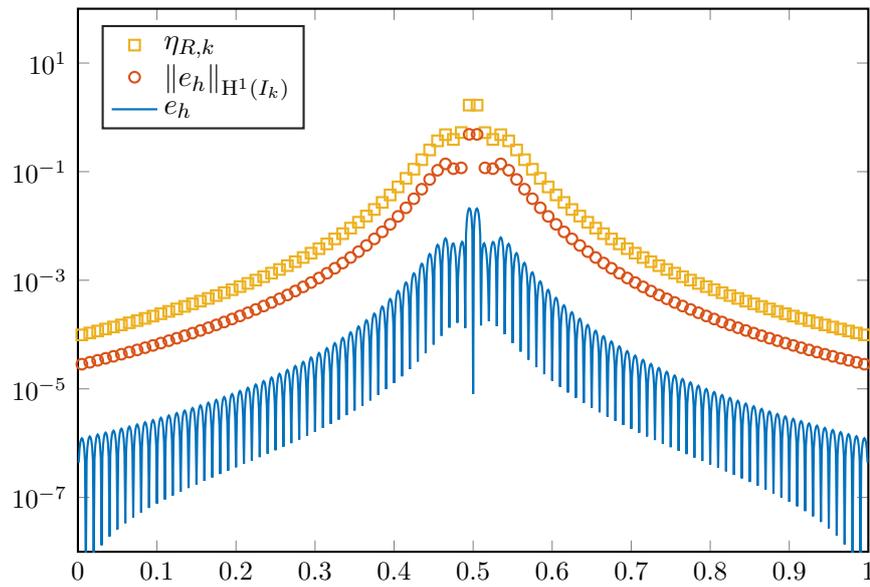
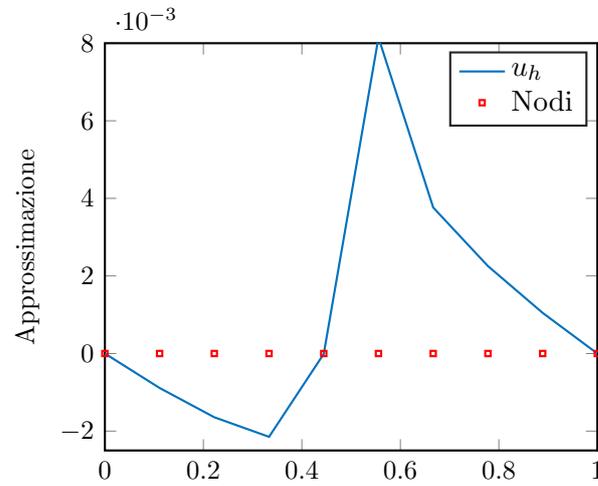
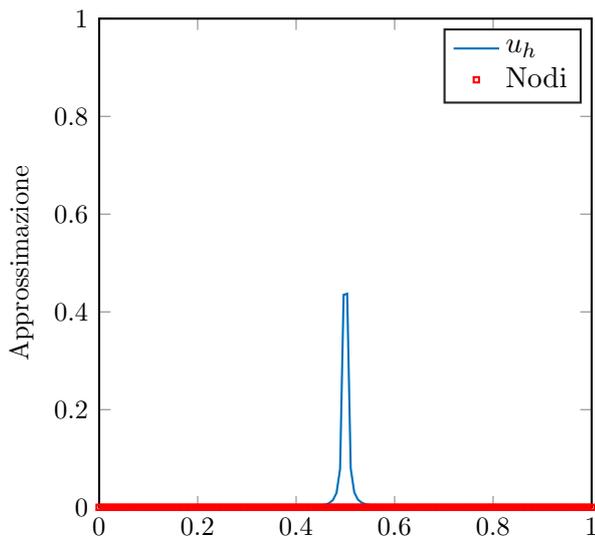


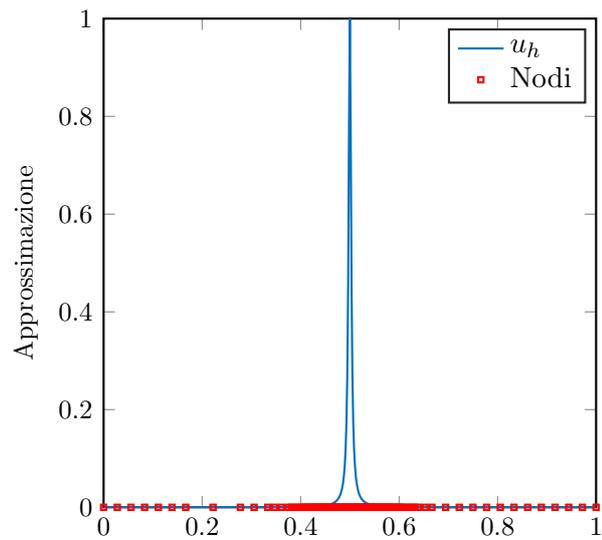
Figura 3.7: Distribuzione dell'errore puntuale, errore in norma e residuo locale su una partizione uniforme di 101 nodi per il problema con *layer* interno con $\alpha = 1000$. L'errore in norma e il residuo locale sono rappresentati al centro dell'intervallo su cui sono calcolati.



(a) Prima approssimazione (10 DoF)



(b) Raffinamento uniforme a 140 DoF



(c) Raffinamento adattivo a 140 DoF

Figura 3.8: Confronto fra due raffinamenti ottenuti dalla stessa prima approssimazione con evidenziati i nodi della partizione. In questo esempio abbiamo cercato una soluzione del problema con *layer* interno per $\alpha = 100000$. La marcatura è stata effettuata con il marcatore Dörfler con $\theta = 0.5$ e premarking al 5%.

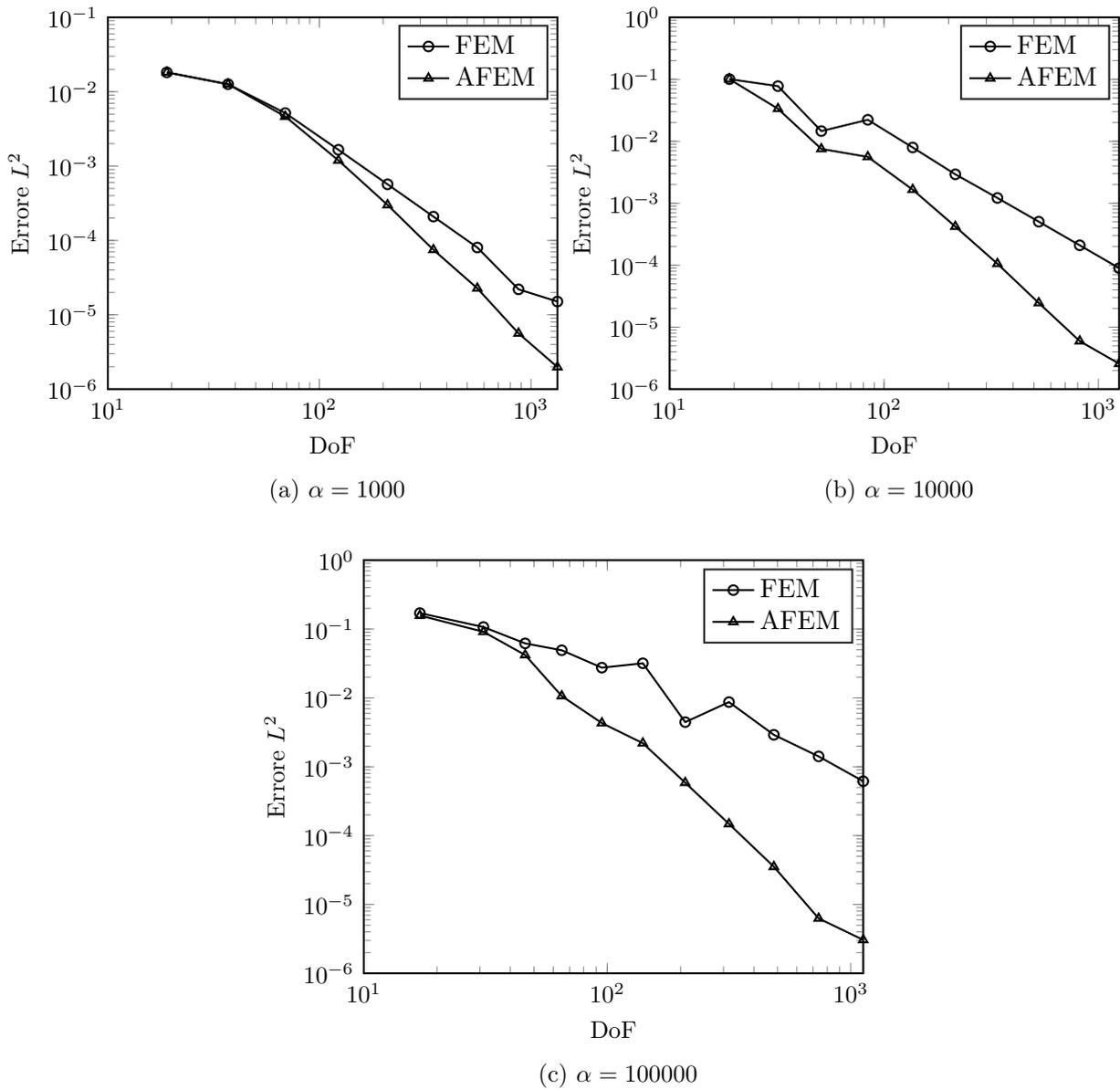


Figura 3.9: Confronto della convergenza FEM vs. AFEM per il problema con *layer* interno. Nell'esempio mostrato l'algoritmo adattivo utilizza il marcatore di Dörfler con $\theta = 0.5$ e premarking al 5%.

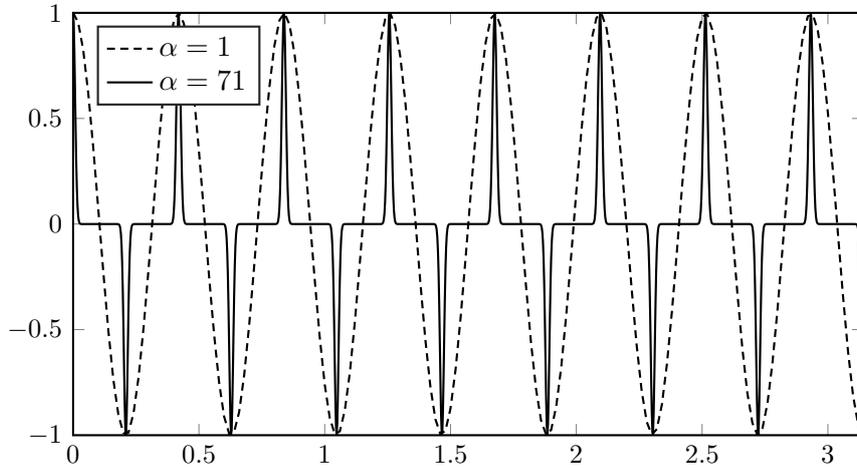


Figura 3.10: Rappresentazione della u definita in (3.3.2) - ovvero soluzione del problema (3.3.3) - per $k = 15$

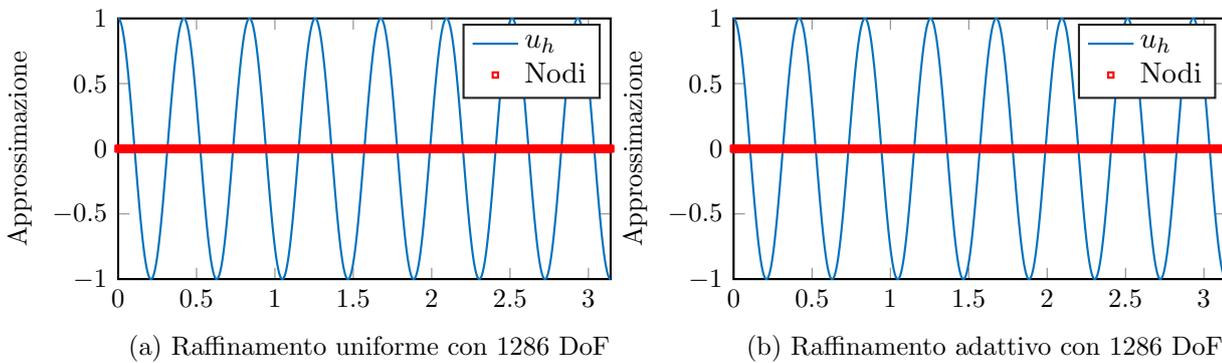


Figura 3.11: Confronto fra i risultati ottenuti con FEM e AFEM per il problema con soluzione oscillante per $\alpha = 1$. Nell'esempio mostrato l'algorithmo adattivo utilizza il marcatore di Dörfler con $\theta = 0.5$, residuo locale sotto $\sigma = 0.1$ e *premarking* al 5%.

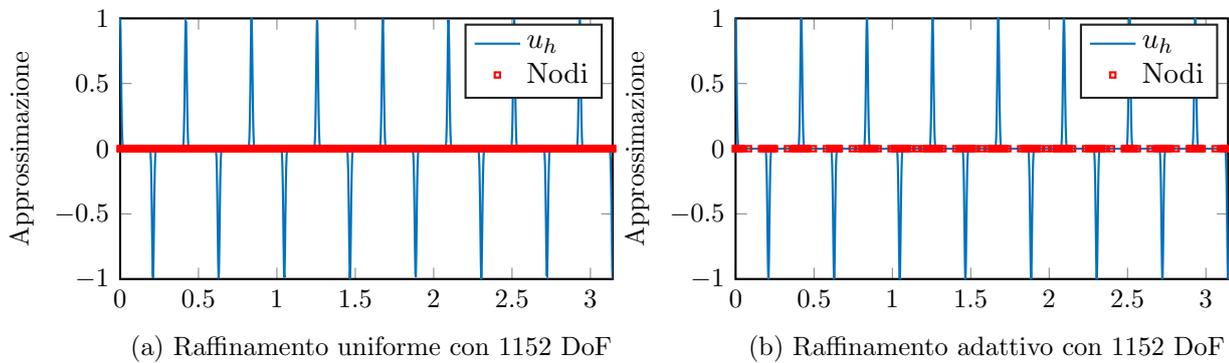


Figura 3.12: Confronto fra i risultati ottenuti con FEM e AFEM per il problema con soluzione oscillante per $\alpha = 121$. Nell'esempio mostrato l'algoritmo adattivo utilizza il marcatore di Dörfler con $\theta = 0.5$, residuo locale sotto $\sigma = 0.1$ e *premarking* al 5%.

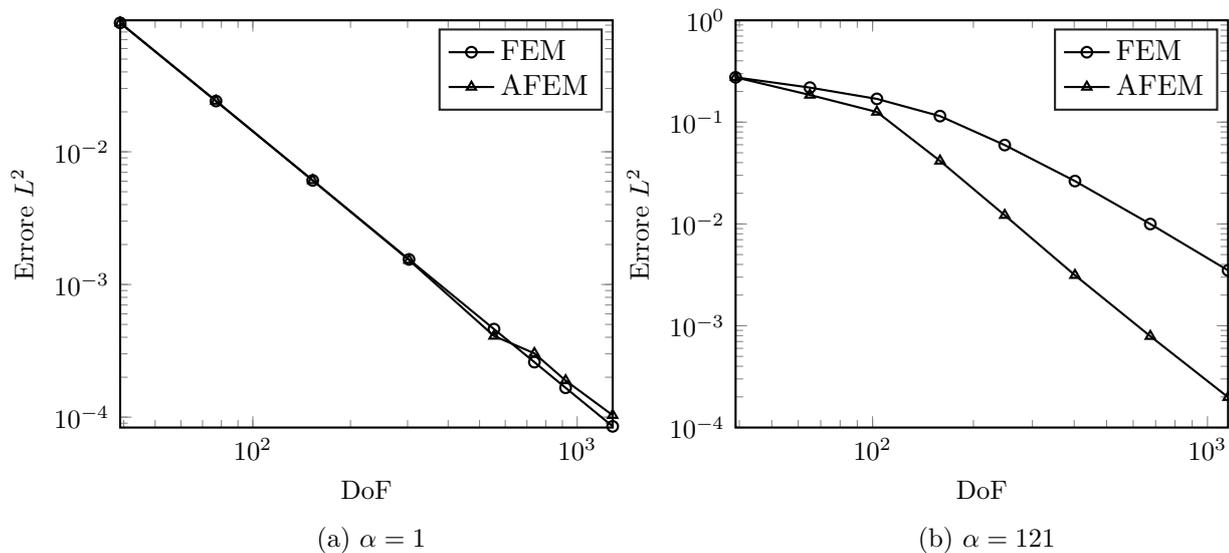


Figura 3.13: Confronto della convergenza FEM vs. AFEM per il problema (3.3.3). Nell'esempio mostrato l'algoritmo adattivo utilizza il marcatore di Dörfler con $\theta = 0.5$, $\sigma = 0.1$ e *premarking* al 5%.

Conclusioni

In questa tesi abbiamo proposto un'introduzione alla teoria di base del metodo degli elementi finiti e abbiamo visto un possibile sviluppo adattivo, in particolare per la risoluzione numerica di problemi di trasporto e diffusione. Il metodo adattivo consiste di successive risoluzioni agli elementi finiti del problema, operate in spazi di approssimazione raffinati in modo tale da riuscire ad esprimere meglio le peculiarità locali della soluzione cercata, nel nostro caso strati limite e forti gradienti in punti interni. Per identificare le regioni in cui è necessario il raffinamento, abbiamo ricavato per i problemi affrontati uno stimatore residuale, che si è dimostrato efficace nel rilevamento dell'errore globale e locale. Nelle simulazioni numeriche che abbiamo svolto il metodo adattivo ha dato buoni risultati, consentendo di raggiungere elevati livelli di precisione con un modesto aumento del numero di gradi di libertà. Il metodo adattivo risulta particolarmente vantaggioso rispetto al classico FEM proprio per l'approssimazione di soluzioni con marcate singolarità locali, poco adatto invece per la risoluzione di problemi con soluzioni regolari.

Appendice A

Codici MATLAB

Tutte le simulazioni che abbiamo riportato nei capitoli della tesi sono state svolte in ambiente MATLAB. In questa appendice abbiamo raccolto i listati dei principali codici scritti.

A.1 Risolutori ad elementi finiti

Listato A.1: Risolutore FEM lineare

```
function [uh,Uh]=solFEM_lin(probdata,space)
% Determina la soluzione approssimata nello spazio X^1 definito da space
% con dati probdata del problema
%
%   -m u'' + b u' = f su [a,b]
%   u(a)=u0, u(b)=uN
%
% Input:   probdata : struct che rappresenta i dati del problema
%           .Omega : Vettore 1x2 con estremi a,b dell'intervallo
%           .m, .b : Parametri dell'equazione
%           .u0, .uN : Valori al bordo
%           .f : termine noto dell'equazione differenziale
%
%           space : struct contenente le informazioni dello spazio di
%                   approssimazione
%                   .dim : Dimensione dello spazio di approssimazione
%                   .T : Vettore dei nodi della partizione
%
% Output:  uh: vettore con i valori nodali di uh
%           Uh: soluzione Uh come function handle (facoltativo)
%
% Calcolo la soluzione
A=FEMassembly(probdata,space);
F=FEMload(probdata,space);
uh=A\F;

% completo la soluzione con i dati di bordo
uh=[probdata.u0, uh', probdata.u1];
```

```

if nargin==2
% Se l'utente ne ha bisogno fornisco già una versione funzionale della
% soluzione
    Uh=@(s) interp1(space.T,uh,s);
end
end

```

Listato A.2: Assemblaggio della matrice di rigidezza

```

function A=FEMassembly(probdata,space)
% Costruisce la matrice di rigidezza del problema probdata nello spazio
% space.

m=probdata.m;
b=probdata.b;
T=space.T;
N=space.dim-1;
H=diff(T); % Passi della partizione

A=zeros(N-1,N-1);
for idx=1:N-1
    if idx~=1
        A(idx,idx-1)=-m/H(idx)-b/2;
    end
    A(idx,idx)=m*(1/H(idx)+1/H(idx+1));
    if idx~=N-1
        A(idx,idx+1)=-m/H(idx+1)+b/2;
    end
end
end

```

Listato A.3: Vettore di carico per la discretizzazione agli elementi finiti

```

function F=FEMload(probdata,space)
% Costruisce il vettore di carico per il problema definito in probdata
% nello spazio space.

m=probdata.m; % Estraggo i dati
b=probdata.b;
T=space.T;
N=space.dim-1;

H=diff(T); % Passi della partizione
fn=probdata.f;
F=zeros(space.dim-2, 1); % Inizializzo

for kdx=1:N-1
    fn1=@(t) fn(t).*(t-T(kdx)); % Contributi del termine noto
    fn2=@(t) fn(t).*(T(kdx+2)-t);
    Q1=integral(fn1,T(kdx),T(kdx+1));
    Q2=integral(fn2,T(kdx+1),T(kdx+2));
    F(kdx)=Q1/H(kdx)+Q2/H(kdx+1);
end

```

```

end
F(1)=F(1)+(m/H(1) + 0.5*b)*probddata.u0; % Contributo dei dati al bordo
F(N-1)=F(N-1)+(m/H(N-1) - 0.5*b)*probddata.u1;
end

```

A.2 Risolutori adattivi

Presentiamo nel listato A.4 un esempio completo di risoluzione adattiva, ed in seguito includiamo tutte le componenti utilizzate.

Listato A.4: Esempio di risoluzione adattiva

```

%% Risoluzione adattiva di un problema del tipo
%
% -mu''+bu'=f su [a,b]
% u(a)=u0, u(b)=u1
%
% dove i dati del problema sono contenuti nella struct probdata. La prima
% approssimazione è cercata nello spazio space. La struct method fornisce i
% parametri per l'algoritmo adattivo, come dettagliato nelle functions
% corrispondenti.
clc; clear; close all;
%% Dati del problema
% I dati riportati sono relativi ad un problema di trasporto-diffusione, ma
% si possono modificare a piacere.

probddata.Omega=[0,1]; % Dominio
probddata.m=1; % Parametri
probddata.b=100;
b=probddata.b; m=probddata.m;
probddata.u0=0; % Dati al bordo
probddata.u1=1;
probddata.f=@(t) 0.*t; % Termine noto

%% Prima approssimazione
% Calcoliamo una prima approssimazione in uno spazio piccolo per avere
% un'idea della forma della soluzione e degli errori (stimati) commessi.

% Spazio di approssimazione
space.dim=30; % con quanti gradi di libertà iniziamo
space.T=linspace(probddata.Omega(1),probddata.Omega(2),space.dim);

% Prima approssimazione
[uh]=solFEM_lin(probddata, space);

% Grafico: produco una figura con la prima approssimazione in modo che
% l'utente possa capire grossomodo con che problema ha a che fare.
figure
plot(space.T,uh)
titolo=sprintf("Approssimazione con %i DoF",space.dim);
title(titolo,'interpreter','latex')

```

```

% Stima residuale : calcolo il residuo locale e informo l'utente sul
% residuo locale e globale nella prima apporssimazione

etaR=LocRes(uh,probddata,space);
etaLoc=max(etaR);
eta=norm(etaR,2);

fprintf("La soluzione determinata ha residuo locale fra %f e %f\n",...
        min(etaR),etaLoc)
fprintf("L'errore globale stimato è %f\n",eta)

stop=input("Si vogliono modificare i parametri della risoluzione adattiva? s/n\n",'s
');
if stop=='s'
% Se l'utente vuole dare nuovi parametri alla luce del residuo locale
% appena mostratogli, interrompiamo l'esecuzione per consentire di
% modificare il codice.
    return
end

%% Metodo di approssimazione
% In questa sezione vengono raccolti i parametri per il metodo adattivo
% quali il tipo di marker e le soglie.

method.maxResLoc=1e-2; % Massimo errore su ogni elemento della partizione

method.maxRes=1e-6; % Massimo errore globale

method.maxIter=10; % Massimo numero di iterazioni dell'algorithmo adattivo

method.marker='Dor'; % Tipo di marcatore
% method.marker='Max';
method.theta=0.5; % Parametro del marcatore

method.PreMark=true; % Sceglie se si effettua premarking
% method.PreMark=false;
method.PreMarkPerc=5; % Percentuale di elementi da premarcare

%% Risoluzione adattiva
% Vedere la function AFEM.m
[uh, spacef, err]=AFEM(probddata,space,method);

%% Soluzione finale
% Presentiamo all'utente la soluzione finale ottenuta.

% Visualizza la soluzione finale
plot(spacef.T,uh)

visnodi=input("Vuoi vedere i nodi della partizione finale? s/n\n",'s');
if visnodi=='s'
    hold on
    plot(spacef.T,min(uh)*ones(size(spacef.T)),'rs')
end

```

```

titolo=sprintf("Approssimazione AFEM con %i DoF",spacef.dim);
title(titolo,'interpreter','latex')

% Informa sull'errore stimato nella soluzione finale
fprintf("Soluzione trovata con %i gradi di libert  in %i iterazioni\n",...
        spacef.dim, err.NoIter)
fprintf("Abbiamo stimato l'errore locale fra %f e %f\n",min(err.ResLoc),...
        max(err.ResLoc))
fprintf("L'errore globale stimato   %f\n",err.Res)

```

Listato A.5: Risolutore adattivo

```

function [uh, spacef, err]=AFEM(probdata, space, method)
% Realizza l' algoritmo degli elementi finiti adattivo per un problema della
% forma
%
%  $-m u'' + b u' = f$  su  $[a,b]$ 
%  $u(a)=u_0, u(b)=u_N$ 
%
% dove i dati del problema sono contenuti nella struct probdata. La prima
% approssimazione   cercata nello spazio space. La struct method fornisce i
% parametri per l' algoritmo adattivo.
%
% Input:   probdata : struct che rappresenta i dati del problema
%           .Omega : Vettore 1x2 con estremi a,b dell' intervallo
%           .m, .b : Parametri dell' equazione
%           .u0, .uN : Valori al bordo
%           .f : Termine noto dell' equazione differenziale
%
%           space : struct contenente le informazioni dello spazio di
%                   approssimazione iniziale
%                   .dim : Dimensione dello spazio di approssimazione
%                   .T : Vettore dei nodi della partizione
%           method : struct contenente i parametri dell' algoritmo adattivo
%                   .maxIter : Massimo numero di iterazioni
%                   .maxDoF : Massimo numero di gradi di libert 
%                   .maxRes : Massimo errore stimato mediante residuo accettato
%                   .maxResLoc : Massimo residuo accettato su ogni elemento
%                               della partizione
%                   .marker : Stringa == 'Dor' per il marcatore di Dorfler
%                               == 'Max' per il criterio del massimo
%                   .theta : Parametro per il marcatore
%                   .PreMark : Booleano che indica se il premarking sia attivo
%                   .PreMarkPerc : Percentuale di elementi da premarcare
%
% Output:  uh : Coordinate della soluzione approssimata rispetto alla base
%           dello spazio descritto da spacef
%           spacef : struct uguale a space contenente le informazioni dello
%                   spazio di approssimazione costruito con il metodo
%                   adattivo
%           err : struct con le informazioni relative all' errore stimato
%                   .Res : Residuo globale dell' ultima approssimazione
%                   .ResLoc : Residuo locale dell' ultima approssimazione

```

```

%               .NoIter : Numero di iterazioni richiesto

% Controllo che la partizione su cui si costruisce lo spazio space sia
% compatibile con il problema assegnato.
if (space.T(1)~=probddata.Omega(1)) || (space.T(end)~=probddata.Omega(2))
    error("La partizione space.T non è valida per il problema assegnato")
end

% Inizializzazioni
eta=Inf;      % Per essere certi che almeno venga svolta un'iterazione
space_ref=space; % Lo spazio inizia come quello dato, poi viene modificato

for kdx=0:method.maxIter
    % Aggiornamento dati
    spacef=space_ref;

    % Approssimazione
    uh=solFEM_lin(probddata, spacef);

    % Stima il residuo locale su tutti gli elementi della partizione
    etaR=LocRes(uh,probddata, spacef);
    eta=norm(etaR,2); % Residuo globale

    err.ResLoc=etaR;
    err.Res=eta;

    % Test per uscita a soglia del residuo globale
    if eta<=method.maxRes
        break
    end

    % Marcatore a soglia
    % Marca per il raffinamento tutti gli elementi con errore stimato
    % maggiore di sigma=method.maxResLoc
    [marked, etaR]=ThreshMark(etaR, method.maxResLoc);

    % Premark
    % marca il (method.PreMarkPerc)% degli elementi con errore massimo se
    % l'utente lo ha chiesto
    if method.PreMark
        [marked1, etaR]=PreMark(etaR,method.PreMarkPerc, spacef);
        % Unisco gli elementi fino ad ora marcati
        marked.id=union(marked.id,marked1.id);
        marked.numel=marked.numel+marked1.numel;
    end

    % Marcatori
    switch method.marker
        case 'Dor'
            marked1=DorflerMark(etaR,method.theta, spacef);
        case 'Max'
            marked1=MaxMark(etaR,method.theta, spacef);
    end
end

```



```

marked.id=find(indici);
marked.numel=nnz(indici);

etaR_rest=etaR;
etaR_rest(marked.id)=0;
end

```

Listato A.8: *Premarker*

```

function [marked, etaR_rest]=PreMark(etaR, eps, space)
% Premarker all'eps%
% Input:   etaR: vettore col residuo locale
%          eps: percentuale degli elementi da premarcare
% Output:  marked: struct con
%          - id: indici degli elementi marcati
%          - numel: numero di elementi marcati
%          etaR_rest: vettore dei residui azzerato sugli elementi
%          premarcati, in modo che non vengano marcati successivamente

N=space.dim-1; % numero elementi di etaR
etaR_rest=etaR; % lo salvo

NoMarked=fix(N*eps*0.01); % NoMarked è il p% degli elementi (intero)
if nnz(etaR)>NoMarked
    [~, idMarked]=sort(etaR, 'descend');
    idMarked=idMarked(1:NoMarked);
    etaR_rest(idMarked)=0; % Azzero il residuo corrispondente agli elementi
                          % marcati
    marked.id=idMarked;
    marked.numel=NoMarked;
else
    marked.id=[];
    marked.numel=0;
end
end

```

Listato A.9: Marcatore di Dörlfer

```

function marked=DorflerMark(etaR, theta, space)
% Algoritmo di marcatura di Dorfler
% Input:   etaR: vettore con i residui locali
%          theta: soglia 0<theta<1
%          space: struct con le informazioni dello spazio di splines
% Output:  marked: struct con i seguenti campi
%          - id: indici degli elementi marcati
%          - numel: numero di elementi marcati

N=space.dim-1;
idMarked=[];
NoMarked=0;
Sigma=0; % Rappresenta (errore dovuto agli elementi selezionati)^2
err_glob=sum(etaR.^2); % Rappresenta (errore globale)^2

while Sigma<theta*err_glob

```

```

Tt=setdiff([1:N],idMarked);
eta=max(etaR(Tt));
for K=Tt
    if etaR(K)==eta
        idMarked=[idMarked,K];
        NoMarked=NoMarked+1;
        Sigma=Sigma+etaR(K)^2;
    end
end
end
marked.id=idMarked;
marked.numel=NoMarked;
end

```

Listato A.10: Marcatore del massimo

```

function marked=MaxMark(etaR,theta,space)
% Algoritmo dorfler per marcatura: certa riduzione dell'errore.
% Input:   etaR: vettore con i residui locali
%          theta: soglia 0<theta<1
%          space: struct con le informazioni dello spazio (vedi AFEM.m)
% Output:  marked: struct con i seguenti campi
%          - id: indici degli elementi marcati
%          - numel: numero di elementi marcati
N=space.dim-1;
idMarked=[];
NoMarked=0;
eta=max(etaR);
for idK=1:N
    if etaR(idK)>=theta*eta
        idMarked=[idMarked,idK];
        NoMarked=NoMarked+1;
    end
end
marked.id=idMarked;
marked.numel=NoMarked;
end

```

Listato A.11: Raffinamento diadico della partizione

```

function refined_space=DyadRef(marked,space)
% Raffinamento diadico degli elementi della mesh dello spazio space marcati
% in marked.
% Input:   marked: struct (si veda una function di marcatura)
%          space: informazioni dello spazio (si veda AFEM.m)
% Output:  refined_space: struct basata su space che rappresenta lo spazio
%          raffinato.

refined_space=space; % iniziamo dallo spazio dato
T=space.T; % la partizione su cui è basato è

% Siccome costruisco una riga controllo che la partizione sia in formato

```

```
% riga. Se non la è la metto in riga.
temp=size(T);
if temp(1)<temp(2)
    T=reshape(T,1,[]);
end
clear temp

for jdx=1:marked.numel
    k=marked.id(jdx);
    h=(T(k+1)-T(k));
    if h>0
        T=[T(1:k),T(k)+0.5*h,T(k+1:end)];
        refined_space.dim=refined_space.dim+1;
        marked.id=marked.id+1;
    end
end
refined_space.T=T;
end
```

Bibliografia

- [1] M. Ainsworth e J. T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Pure and Applied Mathematics: A Wiley-Interscience Series of Texts, Monographs and Tracts. Wiley, 2000.
- [2] I. Babuška e W. C. Rheinboldt. «A-posteriori error estimates for the finite element method». In: *International Journal for Numerical Methods in Engineering* 12.10 (1978), pp. 1597–1615.
- [3] S. Bertoluzza et al. *Multiscale and Adaptivity: Modeling, Numerics and Applications: C.I.M.E. Summer School, Cetraro, Italy 2009*. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 2012.
- [4] S. C. Brenner e R. Scott. *The mathematical theory of finite element methods*. Vol. 15. Springer Science & Business Media, 2007.
- [5] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. 1^a ed. Universitext. Springer-Verlag New York, 2010.
- [6] J. Cea. «Approximation variationnelle des problèmes aux limites». In: *Annales de l'Institut Fourier* 14.2 (1964), pp. 345–444.
- [7] W. Dörfler. «A Convergent Adaptive Algorithm for Poisson's Equation». In: *SIAM Journal on Numerical Analysis* 33.3 (1996), pp. 1106–1124.
- [8] A. Ern e J.L. Guermond. *Theory and Practice of Finite Elements*. Applied Mathematical Sciences. Springer New York, 2004.
- [9] B. G. Galerkin. «Rods and plates. Series occurring in various questions concerning the elastic equilibrium of rods and plates». In: *Engineers Bulletin (Vestnik Inzhenerov)* 19 (1915), pp. 897–908.
- [10] M. J. Gander e G. Wanner. «From Euler, Ritz, and Galerkin to Modern Computing». In: *SIAM Review* 54.4 (gen. 2012), pp. 627–666.
- [11] E. Geerardyn et al. *Matlab2Tikz*. 2016.
- [12] R. C. James. «Reflexivity and the Supremum of Linear Functionals». In: *Annals of Mathematics* 66.1 (1957), pp. 159–169.
- [13] P. D. Lax e A. N. Milgram. «IX. Parabolic Equations». In: *Contributions to the Theory of Partial Differential Equations. (AM-33)*. Princeton University Press, 1955.
- [14] The MathWorks. *MATLAB*. Ver. R2018b.

- [15] T. Oden. «Some Historic Comments on Finite Elements». In: *Proceedings of the ACM Conference on History of Scientific and Numeric Computation*. HSNC '87. ACM, 1987, pp. 125–130.
- [16] A. Quarteroni. *Modellistica Numerica per Problemi Differenziali*. 6^a ed. Springer Milan, gen. 2016.
- [17] A. Quarteroni e A. Valli. *Numerical Approximation of Partial Different Equations*. Springer, 1994.
- [18] A. Quarteroni et al. *Matematica Numerica*. Springer Milan, 12 apr. 2014.
- [19] W. Ritz. «Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik». In: *Journal für die reine und angewandte Mathematik* 135 (1909), pp. 1–61.
- [20] G. Sangalli. «Robust a-posteriori estimator for advection-diffusion-reaction problems». In: *Mathematics of Computation* 77.261 (gen. 2008), pp. 41–70.
- [21] R. Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Numerical mathematics and scientific computation. Oxford University Press, 2013.
- [22] R. Verfürth. *A Review of A Posteriori Error Estimation Adaptive Mesh-Refinement Techniques*. Wiley Teubner, 1996.