

POLITECNICO DI MILANO

FACOLTÀ DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
Corso di Laurea in Ingegneria Matematica

PROGETTO PER IL CORSO DI ANALISI NUMERICA PER LE EQUAZIONI A DERIVATE PARZIALI

Titolo

sottotitolo

Candidati:

Claudia Bonomi matr. 804378

Edoardo Arbib matr.

Relatori:

Simona Perotto

Ilario Mazzieri

Abstract

Contents

1	Introduzione	1
2	Analisi del problema continuo	3
3	Analisi del problema discreto	7
3.1	Semidiscretizzazione temporale	7
3.1.1	Caratteristiche dello schema numerico	7
3.1.2	Semidiscretizzazione del problema di stato	8
3.1.3	Semidiscretizzazione del problema aggiunto	9
4	Discretizzazione variazionale	11
4.0.4	Il caso generale	11
4.0.5	Il caso in oggetto	12
4.1	Metodo di punto fisso	13
4.2	Metodo semi-Newton	14
5	Descrizione Implementazione	17
5.1	Strumenti di sviluppo	17
5.2	Implementazione	18
5.2.1	Calcolo dell'errore	19
5.2.2	Punto Fisso	21
5.2.3	Damped Newton	23
6	Risultati numerici	25
6.1	Test Case 01	25
6.2	Test Case 02	27
7	Conclusioni	29

Chapter 1

Introduzione

Il lavoro qui presentato tratta lo studio di un problema di controllo ottimo parabolico attraverso l'analisi proposta da [MAIN].

Per l'equazione di stato in tempo viene utilizzato uno schema Petrov-Galerkin con un approccio costante a tratti per la funzione di stato ed uno lineare a tratti per la funzione test. Questa scelta degli spazi funzionali ha una ripercussione sullo schema di discretizzazione temporale sia del problema di stato che del problema aggiunto. Per entrambi, infatti, sarà utilizzata una variante dello schema di Crank-Nicolson consistente con la teoria di Rannacher descritta in [Ran84]. In [MAIN] viene provato analiticamente che questa scelta permette di raggiungere un ordine due di convergenza temporale sia per l'errore di controllo che per l'errore dello stato proiettato sulla griglia duale. Per la discretizzazione spaziale si è fatto riferimento all'analisi proposta in [MV11].

Attraverso l'utilizzo del software **FreeFem++** l'approccio teorico proposto precedentemente è stato implementato. I risultati numerici ottenuti confermano quelli teorici e sono consistenti con quelli presentati in [MAIN]. Per il calcolo dell'errore di controllo è stato utilizzato inizialmente il metodo di Cavalieri-Simpson. In seguito è stato calcolato un secondo algoritmo meno soggetto agli errori di approssimazione, con esso si trova un ordine di convergenza maggiore di 2 per l'errore di controllo.

Il report è strutturato nel seguente modo. Nel Capitolo 2 viene analizzata la soluzione teorica del problema di controllo ottimo ed introdotti i risultati di regolarità per l'equazione di stato e per l'equazione aggiunta. Nel Capitolo 3 viene analizzata la regolarità del problema discontinuo e introdotte la semi-discretizzazione temporale e la discretizzazione spaziale. Nel Capitolo 5 sono contenute le informazioni riguardanti l'implementazione dell'algoritmo. Nel Capitolo 6 sono raccolti i risultati numerici. Nel Capitolo 7 sono contenute le conclusioni e spunti per lavori futuri.

Chapter 2

Analisi del problema continuo

In questo studio vengono considerati un dominio poligonale convesso $\Omega \in \mathbb{R}^n$ dove $n = 2, 3$, il cui bordo viene indicato con $\partial\Omega$, ed un intervallo temporale $I = (0, T) \subset \mathbb{R}$, $T < \infty$. Per l'analisi seguente viene introdotta la terna hilbertiana $(H_1^0(\Omega), L^2(\Omega), H^{-1}(\Omega))$. Il problema di controllo ottimo lineare quadratico analizzato è definito come:

$$\begin{aligned} \min_{y \in Y, u \in U_{ad}} \quad & J(y, u) = \frac{1}{2} \|y - y_d\|_{L^2(I, L^2(\Omega))}^2 + \frac{\alpha}{2} \|u\|_U^2 \\ \text{s.t.} \quad & y = S(Bu, y_0) \end{aligned} \quad (\mathbb{P})$$

dove y_d è una funzione scelta $\in L^2(I, L^2(\Omega))$.

Problema di Stato

Il problema di stato è definito in forma forte e in forma debole rispettivamente in 2.1 e 2.2.

$$\begin{aligned} \partial_t y - \Delta y &= f && \text{in } I \times \Omega \\ y &= 0 && \text{in } I \times \Omega \\ y(0) &= \kappa && \text{in } \Omega \end{aligned} \quad (2.1)$$

$y \in W(I)$ con $y(0) = \kappa$ e con $(f, \kappa) \in L^2(I, H^{-1}(\Omega)) \times L^2(\Omega)$ tale che:

$$\begin{aligned} \int_0^T \langle \partial_t y(t), v(t) \rangle_{H^{-1}(\Omega), H_1^0(\Omega)} dt + \int_0^T a(y(t), v(t)) dt \\ = \\ \int_0^T \langle f(t), v(t) \rangle_{H^{-1}(\Omega), H_1^0(\Omega)} dt \quad \forall v \in L^2(I, H_1^0(\Omega)) \end{aligned} \quad (2.2)$$

dove $y(t) \in H_1^0(\Omega)$ e la forma bilineare $a(y(t), v(t)) : H_1^0(\Omega) \times H_1^0(\Omega) \rightarrow \mathbb{R}$ è definita come:

$$a(y, v) = \int_{\Omega} \nabla y(x) \cdot \nabla v(x) dx \quad (2.3)$$

Lo spazio dello stato Y è definito come:

$$Y = W(I) = \{v \in L^2(I, H_1^0(\Omega)), \partial_t v \in L^2(I, H^{-1}(\Omega))\} \quad (2.4)$$

ed in particolare vale che:

$$Y \hookrightarrow C([0, T], L^2(\Omega)) \quad (2.5)$$

l'operatore associato alla soluzione debole di 2.4 è

$$S : L^2(I, H^{-1}(\Omega)) \times L^2(\Omega) \rightarrow Y, (f, \kappa) \mapsto y = S(f, \kappa) \quad (2.6)$$

Applicando l'integrazione per parti sul 2.2 si ricava che:

$$A(y, v) = \int_0^T \langle f(t), v(t) \rangle_{H^{-1}(\Omega) H_1^0(\Omega)} dt + (\kappa, v(0))_{L^2(\Omega)} \quad (2.7)$$

dove $y \in Y$ è la soluzione di 2.2, $v \in Y$ è la funzione test e la forma bilineare $A(y, v) : Y \times Y \rightarrow \mathbb{R}$ è definita come:

$$A(y, v) = \int_0^T -\langle \partial_t v(t), y(t) \rangle_{H^{-1}(\Omega) H_1^0(\Omega)} dt + \int_0^T a(y(t), v(t)) dt + (y(T), v(T))_{L^2(\Omega)} \quad (2.8)$$

Per i risultati di stabilità, la consistenza e la convergenza di 2.1, noti in letteratura, si definisce y come soluzione unica di 2.7. L'equazione 2.8 è necessaria per la definire lo schema di approssimazione numerica per l'equazione di stato come descritto nel Capitolo 3.

Spazio del Controllo

Nello scenario descritto precedentemente la scelta per lo spazio di controllo non è unica. Seguendo le linee guida di [MAIN] questo viene definito come $U = L^2(I, \mathbb{R}^d)$, $d \in \mathbb{N}$. Presi dunque $a_i, b_i \in \mathbb{R}$ t.c. $a_i < b_i \forall i = 1 : d$ la regione ammissibile, costituita da un insieme chiuso e convesso, è definita come:

$$U_{ad} = \{u \in U | a_i \leq u_i(t) \leq b_i \forall i = 1 : d\} \quad (2.9)$$

in questo caso, introdotti i funzionali $g_i \in H^{-1}(\Omega)$ l'operatore di controllo B , lineare e limitato, è definito da 2.10.

$$B : U \rightarrow L^2(I, H^{-1}(\Omega)), u \mapsto \left(t \mapsto \sum_{i=1}^d u_i(t) g_i \right) \quad (2.10)$$

Si nota che l'operatore di controllo B può essere sostituito con l'operatore lineare affine \tilde{B} definito come:

$$\tilde{B} : U \rightarrow L^2(I, H^{-1}(\Omega)), u \mapsto g_0 + Bu \quad (2.11)$$

Affinchè non si perda la validità dei risultati che verranno esposti in seguito si suppone $g_0 \in L^2(I, L^2(\Omega))$ ed $g_0(0) \in H_1^0(\Omega)$. È quindi possibile introdurre l'operatore di proiezione ortogonale

$$P_{U_{ad}} : L^2(I, \mathbb{R}^d) \rightarrow U_{ad} \quad (2.12)$$

Problema Aggiunto

Il problema \mathbb{P} ammette un'unica soluzione $(\bar{y}, \bar{u}) \in Y \times U$ dove $\bar{y} = S(B\bar{u}, y_0)$. Intodotti ora la variabile aggiunta $(\bar{p}, \bar{q}) \in L^2(I, H_1^0(\Omega) \times L^2(\Omega))$, soluzione unica di 2.15, e l'operatore aggiunto $B' : L^2(I, H_1^0(\Omega)) \rightarrow L^2(I, \mathbb{R}^d)$ definito da:

$$B'q(t) = (\langle g_1, q(t) \rangle_{H^{-1}(\Omega)H_1^0(\Omega)}, \dots, \langle g_d, q(t) \rangle_{H^{-1}(\Omega)H_1^0(\Omega)})^T \quad (2.13)$$

Il controllo ottimo, utilizzando l'operatore di proiezione ortogonale 2.12, è caratterizzato dalla seguente condizione necessaria e sufficiente di prim'ordine:

$$\bar{u} = P_{U_{ad}} \left(-\frac{1}{\alpha} B' \bar{p} \right) \quad (2.14)$$

$$\begin{aligned} \int_0^T \langle \partial_t \tilde{y}(t), \bar{p}(t) \rangle_{H^{-1}(\Omega)H_1^0(\Omega)} dt + \int_0^T a(\tilde{y}(t), \bar{p}(t)) dt + (\tilde{y}(0), \bar{q})_{L^2(\Omega)} \\ = \\ \int_0^T \int_{\Omega} (\bar{y}(t, x) - y_d(t, x)) \tilde{y}(t, x) dx dt \quad \forall \tilde{y} \in Y \end{aligned} \quad (2.15)$$

Si nota che per $v \in L^2(I, \mathbb{R}^d)$ vale che:

$$P_{U_{ad}}(v)(t) = (P_{[a_i, b_i]}(v_i(t)))_{i=1}^d \quad (2.16)$$

considerati $a, b, z \in \mathbb{R}$ $P_{[a, b]}(z) = \max(a, \min(z, b))$. Poichè $\bar{y} - y_d \in L^2(I, L^2(\Omega))$ in 2.15, si ha $\bar{p} \in Y$ ed integrando per parti con funzione di Y si trova che:

$$\begin{aligned} \int_0^T \langle -\partial_t \bar{p}(t), \tilde{y}(t) \rangle_{H^{-1}(\Omega)H_1^0(\Omega)} dt + \int_0^T a(\tilde{y}(t), \bar{p}(t)) dt \\ + (\tilde{y}(0), \bar{q})_{L^2(\Omega)} + (\tilde{y}(T), \bar{p}(T))_{L^2(\Omega)} - (\tilde{y}(0), \bar{p}(0))_{L^2(\Omega)} \\ = \end{aligned} \quad (2.17)$$

$$\int_0^T \int_{\Omega} (\bar{y}(t, x) - y_d(t, x)) \tilde{y}(t, x) dx dt \quad \forall \tilde{y} \in Y$$

Il problema 2.17 può essere riscritto in forma forte come:

$$\begin{aligned} \partial_t \bar{p} - \Delta \bar{p} &= h && \text{in } I \times \Omega \\ \bar{p} &= 0 && \text{in } \times \partial \Omega \\ \bar{p}(T) &&& \text{su } \Omega \end{aligned} \quad (2.18)$$

dove $h = \bar{y} - y_d$ e $\bar{q} = \bar{p}(0)$.

Regolarità

Per lo studio della regolarità di 2.1 e 2.16 è importante assumere che:

- i) $y_d \in H^1(I, L^2(\Omega))$ e $y_d(T) \in H_1^0(\Omega)$ ed $g_i \in H_1^0(\Omega) \forall i = 1 : d$ e $y_0 \in H_1^0(\Omega)$ con $\Delta y_0 \in H_1^0(\Omega)$.

Per le dimostrazioni dei risultati esposti in questa sezione si rimanda a [MAIN] o [MV11]

Chapter 3

Analisi del problema discreto

3.1 Semidiscretizzazione temporale

3.1.1 Caratteristiche dello schema numerico

La sezione inizia introducendo la griglia e gli spazi funzionali utilizzati per la semidiscretizzazione temporale dei problemi di stato e aggiunto.

Si partizioni l'intervallo $[0, T)$ in sottointervalli $I_m = [t_{m-1}, t_m)$, dove $0 = t_0 < t_1 < \dots < t_M = T$ sono punti appartenenti al segmento $[0, T)$. Inoltre si affianchi a tale griglia una seconda partizione dell'intervallo $[0, T)$, fondamentale per ottenere i risultati di convergenza dei problemi semidiscretizzati, detta *griglia duale*; essa è composta dagli intervalli $I_m^* = [t_{m-1}^*, t_m^*)$, con $0 = t_0^* < t_1^* < \dots < t_M^* = T$ e $t_m^* = \frac{t_{m-1} + t_m}{2}$ per $m = 1, \dots, M$. Riassumendo si può contare su due suddivisioni dell'intervallo $[0, T)$ tali che $[0, T) = \bigcup_{m=1}^{M+1} I_m$ e $[0, T) = \bigcup_{m=1}^{M+1} I_m^*$.

Chiaramente la griglia primale avrà un parametro che ne descrive "l'accuratezza" che per noi sarà

$$k = \max_{1 \leq m \leq M} k_m, \quad k_m = t_m - t_{m-1} \quad \text{per } m = 1, \dots, M.$$

Per quanto riguarda l'ambientazione funzionale definiamo

$$\begin{aligned} P_k &:= \left\{ v \in C([0, T], H_0^1(\Omega)) \mid v|_{I_m} \in \mathcal{P}_1(I_m, H_0^1(\Omega)) \right\} \hookrightarrow W(I), \\ P_k^* &:= \left\{ v \in C([0, T], H_0^1(\Omega)) \mid v|_{I_m^*} \in \mathcal{P}_1(I_m^*, H_0^1(\Omega)) \right\} \hookrightarrow W(I), \\ Y_k &:= \left\{ v : [0, T] \rightarrow H_0^1(\Omega) \mid v|_{I_m} \in \mathcal{P}_0(I_m, H_0^1(\Omega)) \right\} \hookrightarrow W(I). \end{aligned}$$

Qui, $\mathcal{P}_i(J, H_0^1(\Omega))$, $J \subset \bar{I}$, $i \in \{0, 1\}$, denota l'insieme dei polinomi in tempo al più di grado i sull'intervallo J a valori in $H_0^1(\Omega)$.

Nei risultati che si menzioneranno in seguito si farà spesso uso dei seguenti operatori di interpolazione:

$$1. \mathcal{P}_{Y_k} : L^2(I, H_0^1(\Omega)) \rightarrow Y_k$$

$$\mathcal{P}_{Y_k} v|_{I_m} := \frac{1}{k_m} \int_{t_{m-1}}^{t_m} v dt \quad \text{for } m = 1, \dots, M, \quad \text{e } \mathcal{P}_{Y_k} v(T) := 0$$

$$2. \Pi_{Y_k} : C([0, T], H_0^1(\Omega)) \rightarrow Y_k$$

$$\Pi_{Y_k} v|_{I_m} := v(t_m^*) \quad \text{per } m = 1, \dots, M, \quad \Pi_{Y_k} v(T) := v(T)$$

$$3. \pi_{P_k^*} : C([0, T], H_0^1(\Omega)) \cup Y_k \rightarrow P_k^*$$

$$\pi_{P_k^*} v|_{I_1^* \cup I_2^*} := v(t_1^*) + \frac{t - t_1^*}{t_2^* - t_1^*} (v(t_2^*) - v(t_1^*)),$$

$$\pi_{P_k^*} v|_{I_m^*} := v(t_{m-1}^*) + \frac{t - t_{m-1}^*}{t_m^* - t_{m-1}^*} (v(t_m^*) - v(t_{m-1}^*)), \quad \text{per } m = 3, \dots, M-1,$$

$$\pi_{P_k^*} v|_{I_M^* \cup I_{M+1}^*} := v(t_{M-1}^*) + \frac{t - t_{M-1}^*}{t_M^* - t_{M-1}^*} (v(t_M^*) - v(t_{M-1}^*)).$$

3.1.2 Semidiscretizzazione del problema di stato

Uno degli elementi di novità del lavoro DHV è dato proprio dallo schema adottato per la discretizzazione in tempo dell'equazione di stato, che è totalmente variazionale di tipo Petrov-Galerkin. Verranno infatti utilizzati uno spazio di funzioni costanti a tratti per la soluzione e continue lineari a tratti per le test.

Supponendo di estendere la forma bilineare A di [MAIN] in una mappa $A : W(I) \cup Y_k \times \rightarrow \mathbb{R}$, si tratta dunque di risolvere il seguente problema:

dati $(f, \kappa) \in L^2(I, H^{-1}(\Omega)) \times L^2(\Omega)$, trovare $y_k \in Y_k$ tale che

$$A(y_k, v_k) = \int_0^T \langle f(t), v_k(t) \rangle_{H^{-1}(\Omega) H_0^1(\Omega)} dt + (\kappa, v_k(0))_{L^2(\Omega)} \quad \forall v_k \in P_k. \quad (3.1)$$

Data la scelta dello spazio Y_k per la soluzione, essa può essere espressa come

$$y_k = \alpha_{M+1} \chi_{\{T\}} + \sum_{i=1}^M \alpha_i \chi_{I_i}, \quad \alpha_i \in H_0^1(\Omega) \quad \text{per } i = 1, \dots, M+1,$$

e attraverso facili calcoli si arriva alla formulazione del problema (3.1) in termini delle funzioni α_i :

1. Trovare α_1 tale che

$$\frac{1}{t_1 - t_0} (\alpha_1 - \kappa, g)_{L^2(\Omega)} + \frac{1}{2} a(\alpha_1, g) = \frac{1}{2} (f(0), g)_{L^2(\Omega)} \quad \forall g \in H_0^1(\Omega), \quad (3.2)$$

2. Trovare $\alpha_i, i = 2, \dots, M$ tale che

$$\frac{1}{t_i - t_{i-1}} (\alpha_i - \alpha_{i-1}, g)_{L^2(\Omega)} + \frac{1}{2} a(\alpha_i + \alpha_{i+1}, g) = \frac{1}{2} (f(t_{i-1}), g)_{L^2(\Omega)} \quad \forall g \in H_0^1(\Omega), \quad (3.3)$$

3. Trovare α_{M+1} tale che

$$\frac{1}{t_M - t_{M-1}}(\alpha_{M+1} - \alpha_M, g)_{L^2(\Omega)} + \frac{1}{2}a(\alpha_M, g) = \frac{1}{2}(f(t_M), g)_{L^2(\Omega)} \quad \forall g \in H_0^1(\Omega), \quad (3.4)$$

Alla luce di questa seconda formulazione si evince che lo schema di Petrov-Galerkin introdotto è equivalente ad una variante del metodo di Crank-Nicolson con primo e ultimo passo temporale di Rannacher, e sempre dalla seconda formulazione risulta chiaro che il problema semidiscretizzato ammette unica soluzione $y_k \in Y_k$.

A questo punto vengono enunciati senza dimostrazione alcuni risultati tratti da [MAIN] che garantiscono la stabilità e convergenza del metodo di Petrov-Galerkin. Per la stabilità:

Lemma 1. *Sia $y_k \in Y_k$ la soluzione di (3.1) con $f \in L^2(I, L^2(\Omega))$ e $\kappa \in L^2(\Omega)$ assegnati. Allora esiste una costante $C > 0$ indipendente dal parametro di griglia k tale che*

$$\|y_k\|_I \leq C(\|f\|_I + \|\kappa\|_{L^2(\Omega)})$$

Passando ora all'analisi di convergenza è normale aspettarsi un'approssimazione di ordine $\mathcal{O}(k)$, poiché la y_k è costante a tratti in tempo; tuttavia la proiezione di y_k tramite l'operatore $\pi_{P_k^*}$ permette di ottenere stime di ordine $\mathcal{O}(k^2)$. Infatti,

Lemma 2. *Siano $f \in H^1(I, L^2(\Omega))$, $f(0) \in H_0^1(\Omega)$ e $\kappa \in H_0^1(\Omega)$ con $\Delta\kappa \in H_0^1(\Omega)$ e siano y, y_k le soluzioni dei problemi 2.1 e (3.1) con dati (f, κ) . Allora vale*

$$\|\pi_{P_k^*} y_k - y\|_I \leq Ck^2(\|\partial_t^2 y\|_I + \|\partial_t^2 \Delta y\|_I)$$

3.1.3 Semidiscretizzazione del problema aggiunto

Anche il problema aggiunto viene affrontato con uno schema in tempo di tipo Petrov-Galerkin, dove però vengono scambiati gli spazi funzionali delle soluzioni e delle test. Poiché tale scelta conduce ad un metodo di Crank-Nicolson all'indietro, la sua analisi è standard. Si riportano qui di seguito la formulazione e risultati principali.

Problema aggiunto semidiscretizzato in tempo:

Dato $h \in L^2(I, H^{-1}(\Omega))$ trovare $p_k \in P_k$ tale che

$$A(\tilde{y}, p_k) = \int_0^T \langle h(t), \tilde{y}(t) \rangle_{H^{-1}(\Omega) H_0^1(\Omega)} dt \quad \forall \tilde{y} \in Y_k \quad (3.5)$$

Se si scrive p_k nella forma

$$p_k(t) = \sum_{i=0}^M \beta_i b_i(t)$$

con coefficienti $\beta_i \in H_0^1(\Omega)$ e $b_i \in C([0, T])$, $b_i(t_j) = \delta_{ij}$ per $i, j = 0, \dots, M$, l'equazione (3.5) diventa equivalente a

1. $\beta_M = 0$,
2. trovare $\beta_i, i = 0, \dots, M - 1$ tale che

$$\frac{1}{t_{i+1} - t_i}(\beta_i - \beta_{i+1}, g)_{L^2(\Omega)} + \frac{1}{2}a(\beta_i + \beta_{i+1}, g) = \frac{1}{2}(h(t_i) + h(t_{i+1}), g)_{L^2(\Omega)} \quad \forall g \in H_0^1(\Omega) \quad (3.6)$$

Stabilità e convergenza:

Lemma 3. *Sia $p_k \in P_k$ la soluzione di (3.5) con $h \in L^2(I, L^2(\Omega))$. Allora esiste una costante $C > 0$ indipendente da k tale che*

$$\|p_k\|_{H^1(I, L^2(\Omega))} + \|p_k(0)\|_{H^1(\Omega)} \leq C\|h\|_I.$$

Lemma 4. *Siano p, p_k le soluzioni di 2.18 e di (3.5) rispettivamente, con $h \in L^2(I, L^2(\Omega))$. Allora vale*

$$\|p_k - p\|_I \leq Ck^2(\|\partial_t^2 p\|_I + \|\partial_t \Delta p\|_I).$$

Chapter 4

Discretizzazione variazionale

Per approssimare il problema di controllo ottimo [P] viene applicato un metodo detto di discretizzazione variazionale, introdotto da M. Hinze in [Hin05]. Nella prossima sezione viene brevemente presentato questo metodo nel caso generale di problemi di controllo quadratici, dopodiché lo si vedrà in azione per la risoluzione di \mathbb{P} .

4.0.4 Il caso generale

Si consideri il seguente problema di controllo ottimo quadratico

$$\min_{(y,u) \in Y \times U} J(y, u) \text{ s.t. } y = Su \text{ and } u \in U_{ad}, \quad (4.1)$$

dove $U = U^*$ denota lo spazio di Hilbert del controllo, Y lo spazio di Banach dello stato, $S : U \rightarrow Y \subseteq U$ l'operatore lineare e limitato tra controllo e stato, e $U_{ad} \subseteq U$ l'insieme chiuso e convesso dei controlli ammissibili. Inoltre per $\alpha > 0$ sia il funzionale J dato da

$$J(y, u) = \frac{1}{2} \|y - z\|_Z^2 + \frac{\alpha}{2} \|u\|_U^2, \quad (4.2)$$

dove $Z = Z^*$ denota uno spazio di Hilbert, $z \in Z$ e $Y \hookrightarrow Z \hookrightarrow Y^*$.

La presente discretizzazione di (4.1) si basa sulla discretizzazione dei soli spazi di stato e aggiunto, utilizzando implicitamente le condizioni di ottimalità del primo ordine per la discretizzazione del controllo. Tra i vantaggi di tale approccio si consideri, nel caso si utilizzino schemi ad elementi finiti, il disaccoppiamento dell'approssimazione dell'*active set* dai nodi della griglia per gli EF.

Per definire il controllo discreto sia $S_h : U \rightarrow Y_h \subset Y \subseteq U$ l'operatore lineare limitato tra il controllo e lo stato discretizzato, dove $Y_h \subseteq Y$ è uno sottospazio finito-dimensionale equipaggiato con la norma di Y .

Definizione 1. $u_h^* \in U_{ad}$ è detto controllo discreto ottimale \iff

$$(\hat{J}'_h(u_h^*), v - u_h^*)_U \geq 0 \quad \forall v \in U_{ad}, \quad (4.3)$$

dove $\hat{J}'_h(u) := J(S_h u, u)$.

Osservazione 1. Nel caso di problemi di controllo quadratici vincolati (come per il presente lavoro) la disuguaglianza variazionale (4.3) è condizione di ottimalità necessaria e sufficiente di

$$u_h^* = \arg \min_{u \in U_{ad}} J(S_h u, u). \quad (4.4)$$

. Se esprimiamo la condizione di ottimalità (4.3) qualora $U_{ad} = U$ appare chiaro come sia possibile realizzare uno schema numerico per risolvere un problema di ottimizzazione senza discretizzare lo spazio di controllo:

$$u_h^* = \frac{-1}{\alpha} S_h^* (S_h u_h^* - z), \quad (4.5)$$

Infatti, sebbene u_h^* sia in U , esso è implicitamente un oggetto discreto per via dell'operatore aggiunto discreto.

Ora si riporta senza dimostrazione il risultato principale legato alla discretizzazione variazionale, che poi sarà utile in seguito:

Teorema 1. *Se gli operatori S_h, S_h^* soddisfano le condizioni*

- $\|(S^* - S_h^*)z\|_U \leq Ch^2 \|z\|_Z$
- $\|(S^* S - S_h^* S_h)u^*\|_U \leq Ch^2 \|u^*\|_U$

allora per $h > 0$ sufficientemente piccolo la disuguaglianza variazionale (4.3) ammette un'unica soluzione $u_h^ \in U_{ad}$ che soddisfa*

$$\|u^* - u_h^*\|_U \leq Ch^2 \{\|u^*\|_U + \|z\|_Z\}. \quad (4.6)$$

Qui, $u^ \in U_{ad}$ denota l'unica soluzione del problema (4.1).*

4.0.5 Il caso in oggetto

Ora si vuole applicare il metodo di discretizzazione variazionale al problema [RIF] nei confronti della variabile tempo, dove gli operatori S_h e S_h^* sono individuati dagli schemi di Petrov-Galerkin esposti nei capitoli precedenti.

Dunque, il problema (semi)discretizzato da considerare è

$$\min_{y_k \in Y_k, u \in U_{ad}} J(y_k, u) = \frac{1}{2} \|y_k - y_d\|_{L^2(I, L^2(\Omega))}^2 + \frac{\alpha}{2} \|u\|_U^2, \quad (4.7)$$

$$\text{s.t. } y_k = S_k(Bu, y_0), \quad (4.8)$$

dove S_k è l'operatore delle soluzioni associato a [RIF]. Grazie ai risultati della precedente sezione questo problema ammette un'unica soluzione $(\bar{y}_k, \bar{u}_k) \in Y_k \times U_{ad}$, con $\bar{y}_k = S_k(B\bar{u}_k, y_0)$ e la condizione di ottimalità del primo ordine recita

$$\bar{u}_k = P_{U_{ad}} \left(-\frac{1}{\alpha} B' \bar{p}_k \right), \quad (4.9)$$

dove $\bar{p}_k \in P_k$ denota la variabile aggiunta discreta, che è l'unica soluzione di [RIF] con $h := \bar{y}_k - y_d$. A partire dalla formulazione del problema (??) è possibile ottenere delle stime di convergenza che somigliano alla stima standard ottenuta per problemi con discretizzazione variazionale. Per completezza vengono trascritti qui di seguito tali risultati.

Teorema 2. *Siano \bar{u} e \bar{u}_k le soluzioni di [RIF] e di (4.7) rispettivamente. Allora*

$$\alpha|\bar{u}_k - \bar{u}|_I \leq Ck^2(\|\bar{u}\|_{H^1(I, \mathbb{R}^D)} + \|\bar{u}(0)\|_{\mathbb{R}^D} + \|y_d\|_{H^1(I, L^2(\Omega))} + \|y_d(T)\|_{H^1(\Omega)} + \|y_0\|_{H^1(\Omega)} + \|\Delta y_0\|_{H^1(\Omega)}) \quad (4.10)$$

è soddisfatta.

Teorema 3. *Siano \bar{u} e \bar{u}_k le soluzioni di [RIF] e (4.7) rispettivamente. Allora vale*

$$\|\bar{y} - \pi_{P_k^*} \bar{y}_k\|_I \leq (|a|_I + \|\bar{u}\|_{H^1(I, \mathbb{R}^D)} + \|\bar{u}(0)\|_{\mathbb{R}^D} + \|y_d\|_{H^1(I, L^2(\Omega))} + \|y_d(T)\|_{H^1(\Omega)} + \|y_0\|_{H^1(\Omega)} + \|\Delta y_0\|_{H^1(\Omega)}). \quad (4.11)$$

4.1 Metodo di punto fisso

L'equazione (4.9) è incline ad essere risolta numericamente nonostante il controllo non sia discretizzato esplicitamente; nel presente lavoro vengono scelti schemi di punto fisso e Newton con damping (semi-Newton).

Per quanto riguarda il metodo di punto fisso, se si indica con S_h^* l'operatore di soluzione dell'equazione [RIF], l'algoritmo è il seguente:

Algoritmo 1. 1. Inizializzare $u_h^0 \in U_{ad}$, $n := 0$.

2. Ripetere fino a convergenza

- (a) calcolare Bu_h^n ,
- (b) calcolare $y_h^n = S_h(y_0, Bu_h^n)$,
- (c) calcolare $p_h^n = S_h^*(y_h^n - y_d)$,
- (d) calcolare $u_h^{n+1} = P_{U_{ad}}(-\frac{1}{\alpha}B'p_h^n)$,
- (e) porre $n=n+1$.

Come criterio di arresto è stato scelto

$$\|B'(p_h^{n+1} - p_h^n)\|_{L^\infty(\Omega \times I)} < \varepsilon, \quad (4.12)$$

con ε tolleranza prefissata.

Seguendo [RIF] il metodo di punto fisso per l'equazione (4.9) converge globalmente per $\alpha > \|S_h\|_{\mathcal{L}(L^2(\Omega), L^2(\Omega))}^2$, motivo per cui è stato implementato anche lo schema semi-Newton, che ha invece la pretesa di convergere globalmente per qualsiasi valore di α .

4.2 Metodo semi-Newton

Affinché il metodo di Newton converga globalmente, è stato scelto di applicare ad ogni passo una minimizzazione monodimensionale con criterio di Armijo; per ottenere ciò è opportuno riformulare (4.7) come problema di minimizzazione non vincolata; ci si basa dunque sulla seguente funzione lagrangiana duale $\varphi : L^2(I, L^2(\Omega)) \rightarrow \mathbb{R}$ data da

$$\varphi(w) = - \inf_{u, y \in L^2(I, L^2(\Omega))} \underbrace{\left(\frac{1}{2} \|y - y_d\|_{L^2(I, L^2(\Omega))}^2 + \frac{\alpha}{2} \|u\|_{L^2(I, L^2(\Omega))}^2 + \chi_{U_{ad}}(u) - (w, y - S_h u)_{L^2(I, L^2(\Omega))} \right)}_{\mathcal{L}(u, y, w)} \quad (4.13)$$

dove $\chi_{U_{ad}}$ indica la funzione caratteristica dell'insieme U_{ad} nel senso dell'analisi convessa, ovvero

$$\chi_{U_{ad}} = \begin{cases} 0, & \text{su } U_{ad}, \\ \infty & \text{su } L^2(I, L^2(\Omega)) \setminus U_{ad}. \end{cases} \quad (4.14)$$

Si verifica che φ è differenziabile con derivata lipschitziana e fortemente convessa, infatti

Lemma 5. *La funzione $\varphi : L^2(I, L^2(\Omega)) \rightarrow \mathbb{R}$ da (4.13) è fortemente convessa e Frechet-differenziabile con gradiente lipschitziano*

$$\nabla \varphi(w) = y(w) - S_h u(w), \quad (4.15)$$

dove $y_h(w) = w + y_d$ e $u(w) = P_{U_{ad}}(-\frac{1}{\alpha} S_h^* w)$ sono gli unici punti di minimo della lagrangiana $\mathcal{L}(u, y, w)$ da (4.13) per ogni $w \in L^2(I, L^2(\Omega))$ data.

Si osservi che, poiché il gradiente $\nabla \varphi(w)$ ammette sottogradiente, è possibile applicare una strategia di Newton semi-regolare per il problema duale

$$\min_{w \in L^2(I, L^2(\Omega))} \varphi(w). \quad (4.16)$$

Grazie alla forte convessità, il problema (4.16) ammette un'unica soluzione w^* che soddisfa $\nabla \varphi(w^*) = 0$. Una strategia di Newton semi-regolare per (4.16) significa risolvere

$$\left(I + \frac{1}{\alpha} S_h \mathbb{I}_{S_h^* w} S_h^* \right) \delta w = -(w + y_d) + S_h P_{U_{ad}} \left(-\frac{1}{\alpha} S_h^* w \right). \quad (4.17)$$

L'equazione (4.17) contiene l'operatore $\mathbb{I}_{p_h(v)}$ che ha il seguente significato: introdotto l'*inactive set* della funzione $p_h \in L^2(I, L^2(\Omega))$ come l'insieme $\mathcal{I}(p_h) = \{ \omega \in \Omega \times [0, T] \mid (-\frac{1}{\alpha} p_h(v))(\omega) \in (a(\omega), b(\omega)) \}$ e $\mathbb{I}_{\mathcal{I}(p_h)}$ come la funzione indicatrice

$$\mathbb{I}_{\mathcal{I}(p_h)} = \begin{cases} 1, & \omega \in \mathcal{I}(p_h), \\ 0, & \omega \in \Omega \times [0, T] \setminus \mathcal{I}(p_h), \end{cases} \quad (4.18)$$

con $\mathbb{1}_{p_h(v)}$ si denota l'endomorfismo auto-aggiunto in $L^2(I, L^2(\Omega))$ dato dalla moltiplicazione puntuale con $\mathbb{1}_{\mathcal{I}(p_h)}$. Poiché l'Hessiano generalizzato a primo membro di (4.17) è un endomorfismo auto-aggiunto e definito positivo, è lecito utilizzare un algoritmo di tipo gradiente coniugato per la computazione del passo di Newton semi-regolare. Anche φ è facile da valutare in quanto dal lemma (5) si evince che

$$\varphi(w) = \frac{1}{2} \|w\|_{L^2(I, L^2(\Omega))}^2 - \frac{\alpha}{2} \|u(w)\|_{L^2(I, L^2(\Omega))}^2 + (w, y_d - S_h u(w))_{L^2(I, L^2(\Omega))}. \quad (4.19)$$

A questo punto è possibile indicare l'algoritmo semi-Newton utilizzato nel presente lavoro:

Algoritmo 2. 1. Inizializzare $w^0 \in L^2(I, L^2(\Omega))$, $\beta \in (0, 1)$, $k = 0$,

2. Ripetere fino a convergenza

- (a) Risolvere l'equazione (4.17) per δw^k tramite CG,
- (b) Porre $\lambda := 1$,
- (c) Finché risulta vera la condizione $\varphi(w^k + \lambda \delta w^k) > \varphi(w) + \frac{1}{3} \lambda (\nabla \varphi(w^k), \delta w^k)_{L^2(I, L^2(\Omega))}$,
porre $\lambda := \beta \lambda$,
- (d) Porre $w^{k+1} = w^k + \lambda \delta w^k$
- (e) Porre $k := k + 1$.

Come criterio di arresto è stato scelto $\|\nabla \varphi(w^k)\| \leq t_0$, con t_0 tolleranza prefissata.

L'analisi della convergenza del metodo e la plausibilità del criterio di arresto sono affidati al seguente lemma

Lemma 6. Sia $L = 1 + \|S_h\|^2/\alpha$ la costante di Lipschitz di $\nabla \varphi$ e $\beta \in (0, 1)$ come nell'algoritmo (2). Siano u_h e w^* le soluzioni di (4.7) e (4.16) rispettivamente. Allora si verifica che

1. un passo con parametro di damping $\lambda \leq \beta^{K(L, \beta)}$ è sempre accettato, dove

$$K(L, \beta) = \frac{\log(2) - \log(3L)}{\log(\beta)}, \quad (4.20)$$

2. l'algoritmo (2) converge per qualsiasi dato iniziale $w^0 \in L^2(I, L^2(\Omega))$, nel senso che $w \rightarrow w^*$, $u(w) \rightarrow u_h$ in $L^2(I, L^2(\Omega))$,

3. il criterio di arresto $\|\nabla \varphi(w)\|_{L^2(I, L^2(\Omega))} \leq \text{Toll.}$ è ragionevole poiché

$$\|\nabla \varphi(w)\|_{L^2(I, L^2(\Omega))} \geq \|w - w^*\|_{L^2(I, L^2(\Omega))}. \quad (4.21)$$

Per completezza si descrive l'algoritmo del gradiente coniugato adoperato per la risoluzione di (4.17)

Algoritmo 3. Per brevità sarà indicato con A_w l'Hessiano generalizzato a primo membro di (4.17), mentre il termine noto a secondo membro con b_w

1. Inizializzare δw^0 in $L^2(I, L^2(\Omega))$, $r^0 := b_w - A_w \delta w^0$, $p^0 = r^0$, $n := 0$.

2. Ripetere fino a convergenza

$$(a) \quad \alpha^n := \frac{(r^n, r^n)_{L^2(I, L^2(\Omega))}}{(p^n, A_w p^n)_{L^2(I, L^2(\Omega))}},$$

$$(b) \quad \delta w^{n+1} := \delta w^n + \alpha^n p^n,$$

$$(c) \quad r^{n+1} := r^n - \alpha^n p^n,$$

$$(d) \quad \beta^n := \frac{(r^{n+1}, r^{n+1})_{L^2(I, L^2(\Omega))}}{(r^n, r^n)_{L^2(I, L^2(\Omega))}},$$

$$(e) \quad p^{n+1} := r^{n+1} + \beta^n p^n,$$

$$(f) \quad k := k + 1.$$

Come criterio di arresto è stato scelto $\|r^n\|_{L^2(I, L^2(\Omega))} \leq t_0$, con t_0 tolleranza prescritta. E' superfluo specificare che qui l'indice n denota l'n-esima iterazione di CG, non di Newton.

Chapter 5

Descrizione Implementazione

In questo capitolo verranno descritte la struttura di base e le funzioni principali del codice sviluppato. Per l'implementazione del codice si è utilizzato il solver `FreeFem++`¹ ed il relativo linguaggio di programmazione.

5.1 Strumenti di sviluppo

`FreeFem++`

`FreeFem++` è un software per la risoluzione di equazioni alle derivate parziali che ha un proprio linguaggio di scripting basato sul C++. Gli script permettono di risolvere sistemi non lineari di più variabili in un dominio 2D o 3D. `FreeFem++` è un software libero, disponibile per i sistemi operativi *Linux*, *Solaris*, *OS X* and *MS Windows*.

`GitHub`

`GitHub`² è un sistema di controllo versione utilizzabile direttamente da linea di comando, molto diffuso e utile per tenere traccia delle varie fasi di sviluppo del codice. `GitHub` gestisce in modo adeguato i contributi al codice provenienti da agenti esterni e permette la condivisione del codice.

Il codice del progetto è reperibile su `GitHub` ed è possibile scaricarlo e collaborare allo sviluppo clonando il codice dalla repository `GitHub`:

```
git clone https://github.com/ClaB90/Progetto_ANEDP
```

Nella cartella principale è contenuto anche un file `.gitignore`, in cui sono specificate le estensioni dei files e le sottocartelle che non devono essere visionati in una repository `GitHub`. In particolare non si è interessati ai files temporanei che vengono eventualmente generati dagli editor.

¹<http://www.freefem.org/>

²<https://github.com/>

5.2 Implementazione

In questo lavoro sia l'algoritmo di punto fisso, **Algoritmo 1**, sia l'algoritmo di Newton, **Algoritmo** [REFDN], sono stati implementati per la soluzione del problema parabolico di controllo ottimo (\mathbb{P}) descritto nei capitoli precedenti [REFCHAP]. Il corpo del programma è costituito dal file **main.edp** che permette di risolvere il problema 4.7 con uno dei due algoritmi su una o più griglie temporali. Il processo di raffinamento temporale considera griglie uniformi tali che il numero di nodi N_k al livello l di infittimento sia:

$$N_k = (2^l + 1) \quad (5.1)$$

I valori della soluzione approssimata u_k del problema di controllo vengono salvati per ogni livello di raffinamento in dei file di testo esterni per poter essere confrontati graficamente coi valori della soluzione esatta, nei casi in cui quest'ultima è nota.

L'errore fra la soluzione esatta e la soluzione approssimata per i problemi di stato, dell'aggiunto e del variabile di controllo viene calcolato e salvato in un file di testo ogni livello. Questo permette il calcolo dell'ordine di convergenza per i diversi errori.

Gli altri file principali sono i seguenti:

- **controlparameters.edp**: contiene i parametri che definiscono il problema di controllo;
- **controlfunction.edp**: contiene la funzione per il procedimento di punto fisso, **Algoritmo 1**, e la funzione per l'algoritmo di damped Newton, **Algoritmo** [RIFDN];
- **UabSet.edp**: contiene i parametri limite dell'insieme U_{ab} e le funzioni di proiezione per uno scalare e per un vettore sullo spazio U_{ab} ;
- **funzioni.edp** contiene le funzioni che caratterizzano l'operatore B , la funzione y_d e la forzante del problema di stato g_0 ;
- **soluzioniEsatte.edp**: definisce le funzioni per le soluzioni esatte del problema di stato y , del problema aggiunto p e per la variabile di controllo u ;
- **mesh.edp**: contiene la definizione del dominio, la mesh ed il parametro N di discretizzazione spaziale;
- **state.edp**: in questo file viene definito il problema di stato. In particolare vengono introdotti lo spazio degli elementi finiti, la soluzione y_0 all'istante iniziale, la definizione delle matrici che costituiscono il problema variazionale e la funzione per la risoluzione del problema di stato;
- **adjoint.edp**: in questo file viene definito il problema aggiunto. Anche in questo caso vengono introdotti lo spazio degli elementi finiti, la soluzione p_T all'istante finale, la definizione delle matrici che costituiscono il problema variazionale e la funzione per la risoluzione del problema di aggiunto;

- `normeeprodotti.edp`: contiene tutte le funzioni implementate per il calcolo delle norme e dei prodotti scalari;
- `funzioniErrore.edp` e `funzioniErroreAdj.edp`: contengono le funzioni per il calcolo dell'errore delle variabili y, p e u ;
- `funzioniCG.edp`: contiene le funzioni necessarie per l'algoritmo del gradiente coniugato [RIFCG];
- `saveme.edp` e `saveerr.edp`: descrivono le funzioni per salvare i valori della soluzione approssimata di controllo e dell'aggiunto e degli errori al termine dell'algoritmo per una data griglia temporale. Inoltre contiene la funzione per il calcolo dell'ordine di convergenza degli errori implementati quando un processo iterativo di infittimento di griglia viene utilizzato;

Per il calcolo di ogni norma e prodotto scalare il metodo numerico di integrazione utilizzato è il metodo di Cavalieri Simpson. È stata fatta questa scelta poiché questo metodo ha un ordine due di convergenza dell'errore. In modo che l'ordine di convergenza degli errori di interesse per questo lavoro non sia influenzato negativamente dall'ordine del metodo di integrazione utilizzato.

per i problemi parabolici di controllo ottimo è necessario salvare matrici molto grandi contenenti le soluzioni del problema di stato e del problema aggiunto ad ogni istante temporale considerato. Questa richiesta può portare a problemi di insufficiente memoria RAM durante l'esecuzione del programma. Tuttavia in questo lavoro non sono stati riscontrati problemi di questo tipo, e per una maggiore velocità di calcolo le soluzioni sono state tenute in memoria.

5.2.1 Calcolo dell'errore

Per il calcolo dell'errore della soluzione approssimata u_k del problema di controllo particolare attenzione è stata posta sulla griglia di integrazione. La soluzione del problema aggiunto p_k è una funzione costante a tratti nel tempo i cui punti di non derivabilità coincidono con la griglia di discretizzazione temporale utilizzata. La soluzione u_k , equazione (4.9), è ancora una funzione costante a tratti nel tempo, ma l'operazione di proiezione $P_{U_{ab}}$ non garantisce che i nodi della griglia temporale utilizzata corrispondano a i punti di non derivabilità della funzione stessa. Come mostrato in Figura 5.1. La funzione:

Listing 5.1: Funzione `controlErrL2`

```
1 func real controlErrL2(real[int] &controlsol)
```

Prende in ingresso la soluzione approssimata u_k e calcola la norma L2 della differenza con la soluzione esatta u_{es} , equazione (5.2), utilizzando il metodo di integrazione di Cavalieri-Simpson sulla griglia della soluzione u_k invece che su quella generale. Anche in questo caso il motivo di questa scelta è quello di ridurre al minimo l'influenza degli errori di approssimazione di integrazione rispetto all'errore dovuto all'approssimazione della soluzione.

$$err_{u_k} = ||u_{es} - u_k||_{L2} \quad (5.2)$$

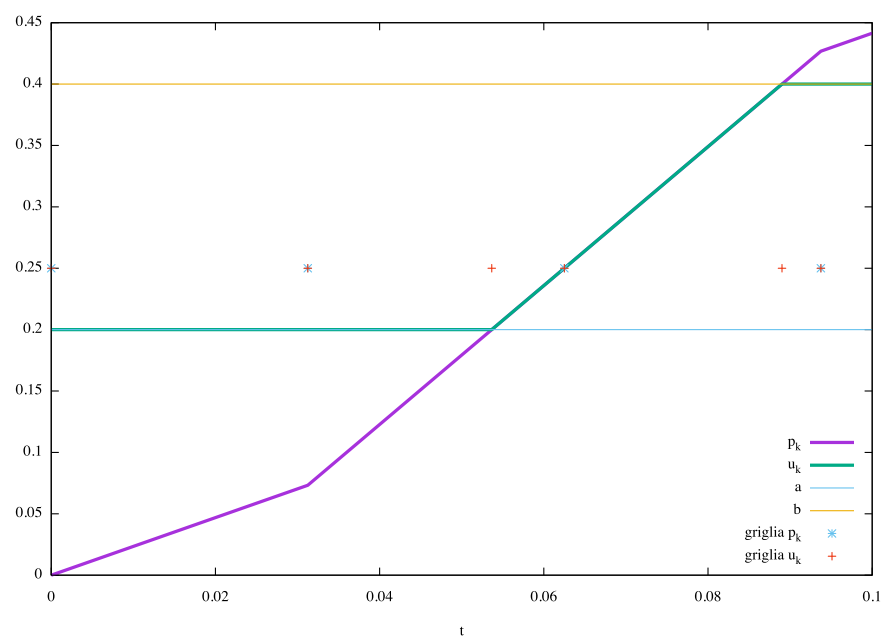


Figure 5.1: Esempio di differenza fra la griglia per la soluzione u_k e la soluzione di p_k

Per questo studio è di particolare interesse anche l'errore della soluzione di stato y_k proietta sullo spazio P_k^* definito come:

$$err_{\pi_{P_k^*} y_k} = \|yes - \pi_{P_k^*} y_k\|_{L2} \quad (5.3)$$

dove yes è la soluzione esatta del problema di stato e $\pi_{P_k^*}$ è definito nel Capitolo 3. Il calcolo dell'errore (5.3) è eseguito dalla funzione `stateprojtrap()`.

La funzione `normL2state()`, invece, calcola l'errore per il problema di stato non proiettato definito da:

$$err_{p_k} = \|ues - u_k\|_{L2} \quad (5.4)$$

La funzione `normL2adj()` calcola infine l'errore per il problema aggiunto definito da:

$$err_{p_k} = \|pes - p_k\|_{L2} \quad (5.5)$$

5.2.2 Punto Fisso

Il primo algoritmo implementato, detto di Punto Fisso segue l'**Algoritmo 1**.

Lo schema di integrazione temporale implementato per il problema di stato è una variante dello schema di Crank-Nicolson(CN). In particolare viene effettuato un primo passo di eulero all'indietro(EI) con passo temporale dimezzato, poi lo schema di Crank-Nicolson viene applicato per ogni fino al penultimo ed infine un passo di Eulero in avanti(EA).

Listing 5.2: Funzione `setscheme()`

```

1 int setscheme(int value)
2 {
3     // EI => value == 0
4     if(value == 0)
5     {
6         gamma1=1./2.;
7         gamma2=0.;
8         gamma3=0.;
9         gamma4=1./2.;
10    }
11    // CN => value == 1
12    else if (value == 1)
13    {
14        gamma1=1./2.;
15        gamma2=1./2.;
16        gamma3=0.;
17        gamma4=1.;
18    }
19    // EA => value == 2
20    else if (value == 2)
21    {
22        gamma1=0.;

```

```

23         gamma2=1./2.;
24         gamma3=1./2.;
25         gamma4=0.;
26     }
27
28     return 0;
29 }

```

Poichè la matrice di Stiffness rimane invariata fino al passo di eulero in avanti per il problema di stato sono state definite due matrici:

Listing 5.3: Matrici **s(w,wtest)** e **statetn(w,wtest)**

```

1 varf s(w,wtest) = int2d(Th)( w*wtest/dt )
2                     + int2d(Th)( gamma1*( dx(w)*dx(wtest) + dy(w)*dy(wtest) ) )
3                     + on(1,2,3,4, w=wb);
4
5 varf statetn(w,wtest) = int2d(Th)( wold*wtest/dt )
6                                     - int2d(Th)( gamma2*( dx(wold)*dx(wtest) + dy(wold)*dy(wtest) ) )
7                                     //contributo di u
8                                     + int2d(Th)( gamma3*( g*ui*wtest ) )
9                                     + int2d(Th)( gamma4*( g*uold*wtest ) )
10                                    //contributo di g0
11                                    + int2d(Th)( gamma3*( g0*wtest ) )
12                                    + int2d(Th)( gamma4*( g0old*wtest ) );

```

In questo modo la matrice di Stiffness **s(w,wtest)** verrà calcolata solamente al primo ed all'ultimo passo dell'algoritmo, permettendo una riduzione del tempo di esecuzione.

La funzione che calcola la soluzione y_k ad ogni passo è la funzione **statet()** che prima di terminare calcola anche l'errore (5.3) ed (5.4).

Anche per il problema aggiunto è stato utilizzato un metodo di Crank-Nicolson ed è possibile trovare due matrici principali:

Listing 5.4: Matrici **adj(w,wtest)** e **adjtn(w,wtest)**

```

1 varf adj(p,ptest) = int2d(Th)( p*ptest/dt )
2                     + int2d(Th)( 0.5*( dx(p)*dx(ptest) + dy(p)*dy(ptest) ) )
3                     + on(1,2,3,4, p=pb);
4
5 varf adjtn(p,ptest) = int2d(Th)( pnext*ptest/dt )
6                                     - int2d(Th)( 0.5*( dx(pnext)*dx(ptest) + dy(pnext)*dy(ptest) ) )
7                                     + int2d(Th)( 0.5*( h*ptest ) )
8                                     + int2d(Th)( 0.5*( hnext*ptest ) );

```

dove **adj(p,ptest)** è invariante ad ogni passo temporale.

Particolare attenzione è stata posta per il calcolo di h ed $hnext$ che rappresentano il valore della forzante al passo i ed $i+1$. Per una corretta implementazione del metodo ad ogni passo i vale che:

$$h_i = y_{k_i} - yd(t_i) \quad hnext_i = y_{k_i} - yd(t_i + dt) \quad (5.6)$$

dove t_i è il tempo considerato all'iterazione i -esima. La funzione che calcola la soluzione p_k ad ogni passo è la funzione **adjointt()** che prima di terminare calcola e salva l'errore per il problema aggiunto definito da (5.5). Inoltre in questa funzione vengono calcolati i valori della soluzione di controllo non proiettata:

$$u_{k_i} = -\frac{1}{\alpha} B p_{k_i} = -\frac{1}{\alpha} * \text{int2d}(Th)(p_{k_i} * g); \quad (5.7)$$

Infine la funzione **puntoFisso()** risolve l'**Algoritmo 1** rispettando il criterio di tolleranza (4.12).

5.2.3 Damped Newton

Chapter 6

Risultati numerici

In questo capitolo verranno esposti i risultati numerici ottenuti. I primi due esempi sono stati presi da [MAIN]. In entrambi questi esempi verrà considerato l'operatore lineare affine \tilde{B} 2.11.

6.1 Test Case 01

Set-up

Il primo esempio considerato, in accordo con l'ipotesi i) introdotta in Chapter ??, Preso il dominio spazio-temporale $\Omega \times I = (0, 1)^2 \times (0, 0.01)$ e $d=1$, si considera l'operatore di controllo lineare affine \tilde{B} che può essere completamente caratterizzato da:

$$g_1(t, x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2)$$

$$g_0(t, x_1, x_2) = -\pi^2 w_a(t, x_1, x_2) - BP_{U_{ad}} \left(-\frac{1}{4\alpha} (\exp(a\pi^2 t) - \exp(a\pi^2 T)) \right) \quad (6.1)$$

dove

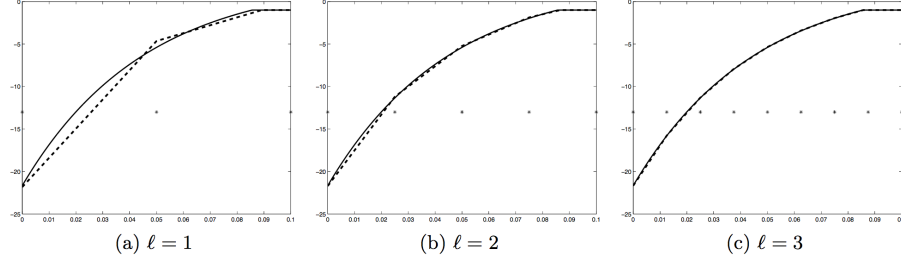
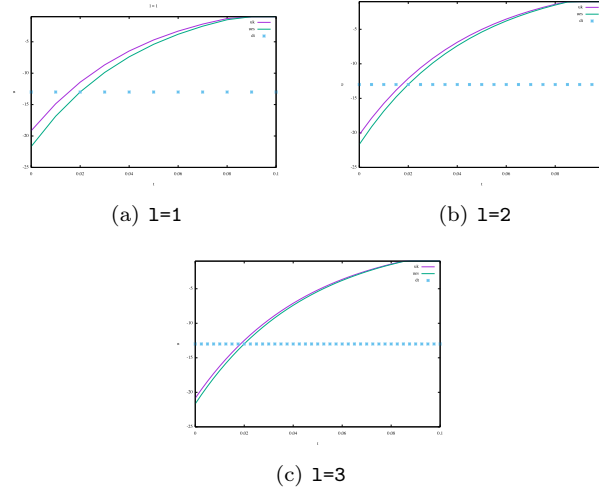
$$w_a(t, x_1, x_2) = \exp(a\pi^2 t) \sin(\pi x_1) \sin(\pi x_2), \quad a \in \mathbb{R} \quad (6.2)$$

In particolare vengono considerate le costanti $a = -\sqrt{5}$ ed $\alpha = \pi^{-4}$. Come conseguenza di ciò si ha che 2.13 verrà riscritta, utilizzando l'aggiunto di B e non \tilde{B} , come:

$$(B'z)(t) = \int_{\Omega} z(t, x_1, x_2) g_1(t, x_1, x_2) dx_1 dx_2 \quad (6.3)$$

Si definiscono ora:

$$\begin{aligned} y_a(t, x_1, x_2) &= \frac{a^2-5}{2+a} \pi^2 w_a(t, x_1, x_2) + 2\pi^2 w_a(T, x_1, x_2) \\ y_0(x_1, x_2) &= \frac{-1}{2+a} \pi^2 w_a(0, x_1, x_2) \end{aligned} \quad (6.4)$$

Figure 6.1: Test Case 01 \bar{u} e u_k risultati di [MAIN]Figure 6.2: Test Case 01 \bar{u} e u_k risultati di questo studio

L'insieme ammissibile U_{ad} è limitato inferiormente da $a_1 = -25$ e superiormente da $b_1 = -1$. Infine definiamo le soluzioni esatte per il problema di controllo ottimo \mathbb{P} :

$$\begin{aligned}
 \bar{u}(t, x_1, x_2) &= P_{U_{ad}} \left(-\frac{1}{4\alpha} (\exp(a\pi^2 t) - \exp(a\pi^2 T)) \right) \\
 \bar{y}(t, x_1, x_2) &= \frac{-1}{2+a} p i^2 w_a(0, x_1, x_2) \\
 \bar{p}(t, x_1, x_2) &= w_a(t, x_1, x_2) - w_a(T, x_1, x_2)
 \end{aligned} \tag{6.5}$$

Risultati Numerici

6.2 Test Case 02

Set-up

Il primo esempio considerato, in accordo con l'ipotesi i) introdotta in Chapter ??, Preso il dominio spazio-temporale $\Omega \times I = (0, 1)^2 \times (0, 0.5)$ e $d=1$, si considera l'operatore di controllo lineare affine \tilde{B} che può essere completamente caratterizzato da:

$$g_1(t, x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2) \quad (6.6)$$

$$g_0(t, x_1, x_2) = g_1(t, x_1, x_2) 2\pi \left(-\frac{a}{T} \sin\left(\frac{t}{T} 2\pi a\right) + \pi \cos\left(\frac{t}{T} 2\pi a\right) \right) - B\bar{u}$$

In particolare vengono considerate le costanti $a = -2$ ed $\alpha = 1$. Si definiscono ora:

$$\begin{aligned} y_d(t, x_1, x_2) &= g_1 \left(\cos\left(\frac{t}{T} 2\pi a\right) (1 - 2\pi^2) - \frac{2\pi a}{T} \sin\left(\frac{t}{T} 2\pi a\right) + 2\pi^2 \cos(2\pi a) \right) \\ y_0(x_1, x_2) &= g_1(t, x_1, x_2) \end{aligned} \quad (6.7)$$

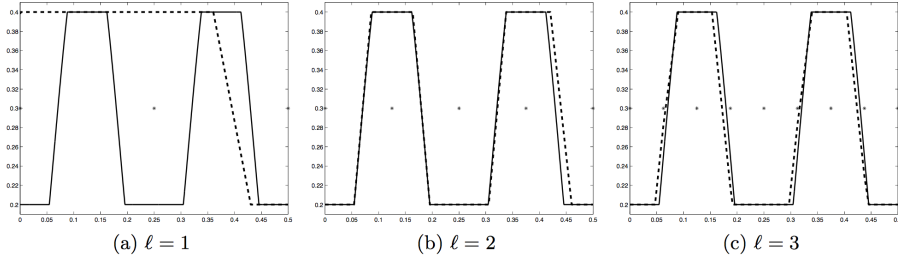
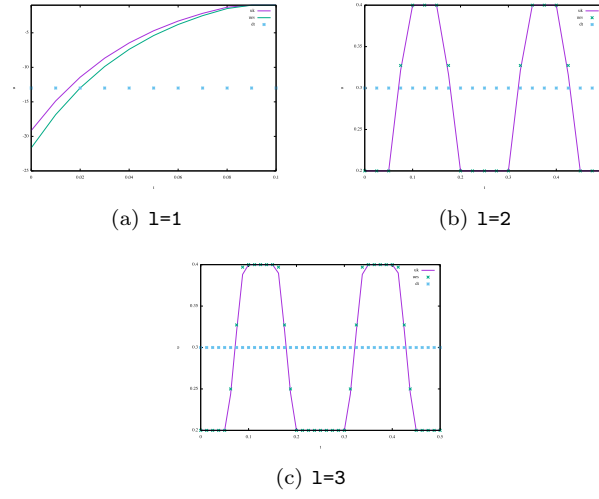
L'insieme ammissibile U_{ad} è limitato inferiormente da $a_1 = 0.2$ e superiormente da $b_1 = 0.4$. Infine definiamo le soluzioni esatte per il problema di controllo ottimo \mathbb{P} :

$$\begin{aligned} \bar{u}(t, x_1, x_2) &= P_{U_{ad}} \left(-\frac{1}{4\alpha} \cos\left(\frac{t}{T} 2\pi a\right) + \frac{1}{4\alpha} \right) \\ \bar{y}(t, x_1, x_2) &= \frac{-1}{2+a} p^2 w_a(0, x_1, x_2) \\ \bar{p}(t, x_1, x_2) &= w_a(t, x_1, x_2) - w_a(T, x_1, x_2) \end{aligned} \quad (6.8)$$

dove

$$w_a(t, x_1, x_2) = \cos\left(\frac{t}{T} 2\pi a\right) \cdot g_1(t, x_1, x_2) \quad (6.9)$$

Risultati Numerici

Figure 6.3: Test Case 02 \bar{u} e u_k risultati di [MAIN]Figure 6.4: Test Case 02 \bar{u} e u_k risultati di questo studio

Chapter 7

Conclusioni

List of Figures

5.1	Esempio di differenza fra la griglia per la soluzione u_k e la soluzione di p_k	20
6.1	Test Case 01 \bar{u} e u_k risultati di [MAIN]	26
6.2	Test Case 01 \bar{u} e u_k risultati di questo studio	26
6.3	Test Case 02 \bar{u} e u_k risultati di [MAIN]	28
6.4	Test Case 02 \bar{u} e u_k risultati di questo studio	28

List of Tables