



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 II

Strategie e analisi dell'errore per problemi di ottimizzazione vincolati regolati da equazioni di evoluzione

Candidati:

Claudia Bonomi matr. 804378

Edoardo Arbib matr. 782037

Relatori:

Simona Perotto

Ilario Mazzieri

Contents

1	Introduzione	3
2	Analisi del problema continuo	3
2.1	Problema di Stato	3
2.2	Spazio del Controllo	4
2.3	Problema Aggiunto	5
2.4	Regolarità	5
3	Analisi del problema discreto	6
3.1	Semidiscretizzazione temporale	6
3.2	Semidiscretizzazione del problema di stato	7
3.2.1	Passo di Rannacher	8
3.3	Semidiscretizzazione del problema aggiunto	8
4	Discretizzazione variazionale	9
4.1	Il caso generale	9
4.2	Il caso in oggetto	10
5	Algoritmi risolutivi	11
5.1	Metodo di Punto Fisso	11
5.2	Metodo semi-Newton	12
6	Descrizione Implementazione	14
6.1	Strumenti di sviluppo	14
6.1.1	FreeFem++	14
6.1.2	GitHub	14
6.2	Implementazione	14
6.2.1	Calcolo dell'errore	16
6.2.2	Punto Fisso	17
6.2.3	Semi-Newton	19
7	Risultati numerici	20
7.1	Test Case 01	20
7.1.1	Set-up	20
7.1.2	Risultati Numerici	21
7.2	Test Case 02	23
7.2.1	Set-up	23
7.2.2	Risultati Numerici	24
8	Conclusioni e Lavori Futuri	24
	References	28

1 Introduzione

Il lavoro qui presentato tratta lo studio di un problema di controllo ottimo parabolico attraverso l'analisi proposta da [NMM15].

Per l'equazione di stato in tempo viene utilizzato uno schema Petrov-Galerkin con funzioni costanti a tratti e continue lineari a tratti. 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 [NMM15] 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 [NMM15].

Il report è strutturato nel seguente modo. Nella sezione 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. Nella sezione 3 vengono introdotte la semi-discretizzazione temporale e la discretizzazione spaziale, mentre in 4 viene descritto il procedimento di discretizzazione variazionale per il problema di controllo. In 5 viene introdotta l'analisi teorica per i due algoritmi studiati. Nella sezione 6 sono contenute le informazioni riguardanti l'implementazione degli algoritmi trattati analiticamente precedentemente. Nella sezione 7 sono raccolti i risultati numerici su due test case. Nel Capitolo 8 sono contenute le conclusioni e gli spunti per lavori futuri.

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 $(H_0^1(\Omega), L^2(\Omega), H^{-1}(\Omega))$. Il problema di controllo ottimo lineare quadratico analizzato è definito come:

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

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

2.1 Problema di Stato

Il problema di stato è definito in forma forte e in forma debole rispettivamente in (1) e (2).

$$\begin{aligned} \partial_t y - \Delta y &= f && \text{in } I \times \Omega \\ y &= 0 && \text{in } I \times \Omega \\ y(0) &= \kappa && \text{in } \Omega \end{aligned} \tag{1}$$

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

$$\begin{aligned} \int_0^T \langle \partial_t y(t), v(t) \rangle_{H^{-1}(\Omega) H_0^1(\Omega)} dt + \int_0^T a(y(t), v(t)) dt \\ = \int_0^T \langle f(t), v(t) \rangle_{H^{-1}(\Omega) H_1^0(\Omega)} dt \quad \forall v \in L^2(I, H_1^0(\Omega)) \end{aligned} \quad (2)$$

dove $y(t), v(t) \in H_1^0(\Omega)$ e la forma bilineare $a(y(t), v(t)) : H_0^1(\Omega) \times H_0^1(\Omega) \rightarrow \mathbb{R}$ è definita come:

$$a(y, v) = \int_{\Omega} \nabla y(x) \nabla v(x) dx \quad (3)$$

Lo spazio dello stato Y è definito come:

$$Y = W(I) = \{v \in L^2(I, H_1^0(\Omega)), \partial_t v \in L^2(I, H^{-1}(\Omega))\} \quad (4)$$

ed in particolare vale che:

$$Y \hookrightarrow C([0, T], L^2(\Omega)) \quad (5)$$

l'operatore associato alla soluzione debole di 4 è

$$S : L^2(I, H^{-1}(\Omega)) \times L^2(\Omega) \rightarrow Y, (f, \kappa) \mapsto y = S(f, \kappa) \quad (6)$$

Applicando l'integrazione per parti sul 2 si ricava che:

$$A(y, v) = \int_0^T \langle f(t), v(t) \rangle_{H^{-1}(\Omega) H_0^1(\Omega)} dt + (\kappa, v(0))_{L^2(\Omega)} \quad (7)$$

dove $y \in Y$ è la soluzione di 2, $v \in Y$ è la funzione test e la forma bilineare $A(y, v) : Y \times Y \rightarrow \mathbb{R}$ è definita come:

$$A(y, v) = \int_0^T -\langle \partial_t v(t), y(t) \rangle_{H^{-1}(\Omega) H_0^1(\Omega)} dt + \int_0^T a(y(t), v(t)) dt + (y(T), v(T))_{L^2(\Omega)} \quad (8)$$

Per i risultati di stabilità, la consistenza e la convergenza di (1), noti in letteratura, si definisce y come soluzione unica di (7). L'equazione (8) è necessaria per la definire lo schema di approssimazione numerica per l'equazione di stato come descritto nella sezione successiva 3.

2.2 Spazio del Controllo

Nello scenario descritto precedentemente la scelta per lo spazio di controllo non è unica. Seguendo le linee guida di [NMM15] 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:

$$U_{ad} = \{u \in U \mid a_i \leq u_i(t) \leq b_i \forall i = 1 : d\} \quad (9)$$

in questo caso, introdotti i funzionali $g_i \in H^{-1}(\Omega)$ l'operatore di controllo B , lineare e limitato, è definito da 10.

$$B : U \rightarrow L^2(I, H^{-1}(\Omega)), u \mapsto \left(t \mapsto \sum_{i=1}^d u_i(t) g_i \right) \quad (10)$$

Si noti che la teoria presentata rimane valida se l'operatore di controllo B viene sostituito con l'operatore lineare affine \tilde{B} definito come:

$$\tilde{B} : U \rightarrow L^2(I, H^{-1}(\Omega)), u \mapsto g_0 + Bu \quad (11)$$

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 $g_0(0) \in H_0^1(\Omega)$ È quindi possibile introdurre l'operatore di proiezione ortogonale

$$P_{U_{ad}} : L^2(I, \mathbb{R}^d) \rightarrow U_{ad} \quad (12)$$

2.3 Problema Aggiunto

Il problema \mathbb{P} ammette un'unica soluzione $(\bar{y}, \bar{u}) \in Y \times U$ dove $\bar{y} = S(B\bar{u}, y_0)$. Vengono ora introdotti la variabile aggiunta $(\bar{p}, \bar{q}) \in L^2(I, H_1^0(\Omega) \times L^2(\Omega))$, soluzione unica di 14, e l'operatore aggiunto $B' : L^2(I, H_1^0(\Omega)) \rightarrow L^2(I, \mathbb{R}^d)$ definito da:

$$B'q(t) = (\langle g_1, q(t) \rangle_{H^{-1}(\Omega)H_1^0(\Omega)}, \dots, \langle g_d, q(t) \rangle_{H^{-1}(\Omega)H_1^0(\Omega)})^T \quad (13)$$

$$\begin{aligned} \int_0^T \langle \partial_t \tilde{y}(t), \bar{p}(t) \rangle_{H^{-1}(\Omega)H_1^0(\Omega)} dt + \int_0^T a(\tilde{y}(t), \bar{p}(t)) dt + (\tilde{y}(0), \bar{q})_{L^2(\Omega)} \\ = \\ \int_0^T \int_{\Omega} (\bar{y}(t, x) - y_d(t, x)) \tilde{y}(t, x) dx dt \quad \forall \tilde{y} \in Y \end{aligned} \quad (14)$$

Il controllo ottimo, utilizzando l'operatore di proiezione ortogonale 12, è caratterizzato dalla seguente condizione necessaria e sufficiente del primo ordine:

$$\bar{u} = P_{U_{ad}} \left(-\frac{1}{\alpha} B' \bar{p} \right) \quad (15)$$

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

$$P_{U_{ad}}(v)(t) = (P_{[a_i, b_i]}(v_i(t)))_{i=1}^d \quad (16)$$

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 14, si ha $\bar{p} \in Y$ ed integrando per parti si trova che:

$$\begin{aligned} \int_0^T \langle -\partial_t \bar{p}(t), \tilde{y}(t) \rangle_{H^{-1}(\Omega)H_1^0(\Omega)} dt + \int_0^T a(\tilde{y}(t), \bar{p}(t)) dt \\ + (\tilde{y}(0), \bar{q})_{L^2(\Omega)} + (\tilde{y}(T), \bar{p}(T))_{L^2(\Omega)} - (\tilde{y}(0), \bar{p}(0))_{L^2(\Omega)} \\ = \\ \int_0^T \int_{\Omega} (\bar{y}(t, x) - y_d(t, x)) \tilde{y}(t, x) dx dt \quad \forall \tilde{y} \in Y \end{aligned} \quad (17)$$

Il problema 17 può essere riscritto in forma forte come:

$$\begin{aligned} \partial_t \bar{p} - \Delta \bar{p} &= h && \text{in } I \times \Omega \\ \bar{p} &= 0 && \text{in } I \times \partial\Omega \\ \bar{p}(T) &= 0 && \text{su } \Omega \end{aligned} \quad (18)$$

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

2.4 Regolarità

Per lo studio della regolarità di 1 e 16 è importante assumere che:

i) $y_d \in H^1(I, L^2(\Omega))$ e $y_d(T) \in H_0^1(\Omega)$ 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 gli enunciati e le dimostrazioni sui risultati di regolarità si rimanda al paragrafo 3 di [NMM15].

3 Analisi del problema discreto

3.1 Semidiscretizzazione temporale

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:

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

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

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

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

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

$$\begin{aligned} \pi_{P_k^*} v|_{I_1^* \cup I_2^*} &:= v(t_1^*) + \frac{t - t_1^*}{t_2^* - t_1^*} (v(t_2^*) - v(t_1^*)), \\ \pi_{P_k^*} v|_{I_m^*} &:= v(t_{m-1}^*) + \frac{t - t_{m-1}^*}{t_m^* - t_{m-1}^*} (v(t_m^*) - v(t_{m-1}^*)), \quad \text{per } m = 3, \dots, M-1, \\ \pi_{P_k^*} v|_{I_M^* \cup I_{M+1}^*} &:= v(t_{M-1}^*) + \frac{t - t_{M-1}^*}{t_M^* - t_{M-1}^*} (v(t_M^*) - v(t_{M-1}^*)). \end{aligned}$$

3.2 Semidiscretizzazione del problema di stato

Uno degli elementi di novità del lavoro [NMM15] 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 [NMM15] 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

$$A(y_k, v_k) = \int_0^T \langle f(t), v_k(t) \rangle_{H^{-1}(\Omega) H_0^1(\Omega)} dt + (\kappa, v_k(0))_{L^2(\Omega)} \quad \forall v_k \in P_k. \quad (19)$$

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 (19) in termini delle funzioni α_i :

1. Trovare α_1 tale che

$$\frac{1}{t_1 - t_0} (\alpha_1 - \kappa, g)_{L^2(\Omega)} + \frac{1}{2} a(\alpha_1, g) = \frac{1}{2} (f(0), g)_{L^2(\Omega)} \quad \forall g \in H_0^1(\Omega), \quad (20)$$

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

$$\frac{1}{t_i - t_{i-1}} (\alpha_i - \alpha_{i-1}, g)_{L^2(\Omega)} + \frac{1}{2} a(\alpha_i + \alpha_{i+1}, g) = \frac{1}{2} (f(t_{i-1}), g)_{L^2(\Omega)} \quad \forall g \in H_0^1(\Omega), \quad (21)$$

3. Trovare α_{M+1} tale che

$$\frac{1}{t_M - t_{M-1}} (\alpha_{M+1} - \alpha_M, g)_{L^2(\Omega)} + \frac{1}{2} a(\alpha_M, g) = \frac{1}{2} (f(t_M), g)_{L^2(\Omega)} \quad \forall g \in H_0^1(\Omega), \quad (22)$$

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 $y_k \in Y_k$.

Prima di enunciare senza dimostrazione alcuni risultati tratti da [NMM15] che garantiscono la stabilità e convergenza del metodo di Petrov-Galerkin, aggiungiamo le seguenti abbreviazioni

$$\|\cdot\|_I := \|\cdot\|_{L^2(I, L^2(\Omega))}, \quad |\cdot|_I := \|\cdot\|_{L^2(I, \mathbb{R}^d)}. \quad (23)$$

Ora, per la stabilità:

Lemma 1. *Sia $y_k \in Y_k$ la soluzione di (19) 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 1 e (19) con dati (f, κ) . Allora vale*

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

3.2.1 Passo di Rannacher

In questa sezione si espone brevemente cosa significhi un passo di Rannacher all'interno di uno schema numerico per la soluzione di EDP.

Come è noto i problemi lineari continui di diffusione esibiscono il cosiddetto “effetto regolarizzante”; infatti per esempio la soluzione dell'equazione del calore in assenza di coefficienti discontinui è di classe C^∞ anche se il dato iniziale è una distribuzione. Le modifiche di uno schema numerico per equazioni paraboliche attraverso alcuni passi di Rannacher servono proprio a restituire questa proprietà anche al problema discreto.

Per essere più concreti, si consideri la seguente equazione lineare di diffusione-convezione

$$\partial_t u + Au = f \text{ in } \Omega \times (0, \infty), \quad (24)$$

$$u(0) = u^0 \text{ in } \Omega, \quad (25)$$

$$u = d \text{ in } \partial\Omega \times (0, \infty). \quad (26)$$

dove A è un operatore differenziale ellittico e sono state scelte condizioni al bordo di solo Dirichlet per semplicità. Si supponga ora che (24) venga discretizzato con elementi finiti in spazio di modo che la semidiscretizzazione conduca al seguente sistema di equazioni differenziali ordinarie

$$\partial_t u_h + A_h u_h = F_h, \quad t \in (0, \infty) \quad (27)$$

$$u_h(0) = u_h^0. \quad (28)$$

Si tratta dunque di affrontare, trascurando il termine noto, un sistema del tipo

$$\frac{\partial \psi}{\partial t} = H(t)\psi(t) \quad (29)$$

$$\psi(0) = \psi_0 \quad (30)$$

la cui soluzione formale è $\psi(t) = \psi_0 \exp(\int_0^t H(s)ds)$. Ora si può pensare, al fine di ricavare uno schema numerico, di sostituire l'esponenziale con una sua approssimante di Padè di ordine (ν, μ) e di supporre l'operatore $H(t)$ costante in tempo; scegliendo per esempio $\nu = 1, \mu = 1$ si ottiene lo schema di Crank-Nicolson, con la premura di aggiungere un termine che tenga conto dell'integrazione in tempo del termine noto. Il problema riscontrato dai metodi con $\nu = \mu$ (schemi di Padè diagonali) è la mancanza di assoluta stabilità. Questa proprietà è invece verificata da algoritmi con $\mu < \nu$ (schemi di Padè sub-diagonali). Ci si aspetta quindi che gli schemi di Padè diagonali propaghino maggiormente gli errori locali dovuti alle discontinuità dei dati rispetto a quelli sub-diagonali. L'idea del passo di Rannacher consiste, qualora si scegliesse di utilizzare uno schema di Padè diagonale, nella sostituzione dei passi in corrispondenza di irregolarità nel termine noto o nelle condizioni al bordo con altri tipici di schemi sub-diagonali. In effetti si dimostra che la sostituzione ottimale per uno schema di ordine (μ, μ) si compie attraverso passi sub-diagonali $(\mu - 1, \mu)$.

3.3 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

$$A(\tilde{y}, p_k) = \int_0^T \langle h(t), \tilde{y}(t) \rangle_{H^{-1}(\Omega) H_0^1(\Omega)} dt \quad \forall \tilde{y} \in Y_k \quad (31)$$

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 (31) diventa equivalente a

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

$$\frac{1}{t_{i+1} - t_i} (\beta_i - \beta_{i+1}, g)_{L^2(\Omega)} + \frac{1}{2} a(\beta_i + \beta_{i+1}, g) = \frac{1}{2} (h(t_i) + h(t_{i+1}), g)_{L^2(\Omega)} \quad \forall g \in H_0^1(\Omega) \quad (32)$$

Stabilità e convergenza:

Lemma 3. *Sia $p_k \in P_k$ la soluzione di (31) 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 18 e di (31) rispettivamente, con $h \in L^2(I, L^2(\Omega))$. Allora vale*

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

4 Discretizzazione variazionale

Per approssimare il problema di controllo ottimo (\mathbb{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}) .

4.1 Il caso generale

Si consideri il seguente problema di controllo ottimo quadratico

$$\min_{(y, u) \in Y \times U} J(y, u) \text{ s.t. } y = Su \text{ and } u \in U_{ad}, \quad (33)$$

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

$$J(y, u) = \frac{1}{2} \|y - z\|_Z^2 + \frac{\alpha}{2} \|u\|_U^2, \quad (34)$$

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

Il presente metodo 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 elementi finiti.

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

$$(\hat{J}'_h(u_h^*), v - u_h^*)_U \geq 0 \quad \forall v \in U_{ad}, \quad (35)$$

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 (35) è condizione di ottimalità necessaria e sufficiente di

$$u_h^* = \arg \min_{u \in U_{ad}} J(S_h u, u). \quad (36)$$

Se esprimiamo la condizione di ottimalità (35) 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:

$$u_h^* = -\frac{1}{\alpha} S_h^* (S_h u_h^* - z), \quad (37)$$

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 (35) ammette un'unica soluzione $u_h^ \in U_{ad}$ che soddisfa*

$$\|u^* - u_h^*\|_U \leq Ch^2 \{\|u^*\|_U + \|z\|_Z\}. \quad (38)$$

Qui, $u^ \in U_{ad}$ denota l'unica soluzione del problema (33).*

4.2 Il caso in oggetto

Ora si vuole applicare il metodo di discretizzazione variazionale al problema (\mathbb{P}) 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:

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

dove S_k è l'operatore delle soluzioni associato a (19). 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

$$\bar{u}_k = P_{U_{ad}} \left(-\frac{1}{\alpha} B' \bar{p}_k \right), \quad (39)$$

dove $\bar{p}_k \in P_k$ denota la variabile aggiunta discreta, che è l'unica soluzione di (\mathbb{P}) con $h := \bar{y}_k - y_d$. A partire dalla formulazione del problema (\mathbb{P}_k) 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 (\mathbb{P}) e (\mathbb{P}_k) rispettivamente. Allora*

$$\alpha |\bar{u}_k - \bar{u}|_I \leq Ck^2 (\|\bar{u}\|_{H^1(I, \mathbb{R}^D)} + \|\bar{u}(0)\|_{\mathbb{R}^D} + \|y_d\|_{H^1(I, L^2(\Omega))} + \|y_d(T)\|_{H^1(\Omega)} + \|y_0\|_{H^1(\Omega)} + \|\Delta y_0\|_{H^1(\Omega)}) \quad (40)$$

è soddisfatta.

Teorema 3. *Siano \bar{u} e \bar{u}_k le soluzioni di (\mathbb{P}) e (\mathbb{P}_k) rispettivamente. Allora vale*

$$\|\bar{y} - \pi_{P_k^*} \bar{y}_k\|_I \leq (|a|_I + \|\bar{u}\|_{H^1(I, \mathbb{R}^D)} + \|\bar{u}(0)\|_{\mathbb{R}^D} + \|y_d\|_{H^1(I, L^2(\Omega))} + \|y_d(T)\|_{H^1(\Omega)} + \|y_0\|_{H^1(\Omega)} + \|\Delta y_0\|_{H^1(\Omega)}). \quad (41)$$

5 Algoritmi risolutivi

5.1 Metodo di Punto Fisso

L'equazione (39) 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 (31), 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 $u_h^{n+1} = P_{U_{ad}} \left(-\frac{1}{\alpha} B' p_h^n \right)$,
- (e) porre $n=n+1$.

Come criterio di arresto è stato scelto

$$\|B'(p_h^{n+1} - p_h^n)\|_{L^\infty(\Omega \times I)} < \varepsilon, \quad (42)$$

con ε tolleranza prefissata.

Seguendo [HV12] il metodo di punto fisso per l'equazione (39) 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 .

5.2 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 (\mathbb{P}_k) 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

$$\begin{aligned} \varphi(w) = & \\ - \inf_{u, y \in L^2(I, L^2(\Omega))} & \underbrace{\left(\frac{1}{2} \|y - y_d\|_{L^2(I, L^2(\Omega))}^2 + \frac{\alpha}{2} \|u\|_{L^2(I, L^2(\Omega))}^2 + \chi_{U_{ad}}(u) - (w, y - S_h u)_{L^2(I, L^2(\Omega))} \right)}_{\mathcal{L}(u, y, w)} \end{aligned} \quad (43)$$

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

$$\chi_{U_{ad}} = \begin{cases} 0, & \text{su } U_{ad}, \\ \infty & \text{su } L^2(I, L^2(\Omega)) \setminus U_{ad}. \end{cases} \quad (44)$$

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 (43) è fortemente convessa e Frechet-differenziabile con gradiente lipschitziano*

$$\nabla \varphi(w) = y(w) - S_h u(w), \quad (45)$$

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 (43) 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

$$\min_{w \in L^2(I, L^2(\Omega))} \varphi(w). \quad (46)$$

Grazie alla forte convessità, il problema (46) ammette un'unica soluzione w^* che soddisfa $\nabla \varphi(w^*) = 0$. Una strategia di Newton semi-regolare per (46) significa risolvere ad ogni passo

$$(I + \frac{1}{\alpha} S_h \mathbb{1}_{S_h^* w} S_h^*) \delta w = -(w + y_d) + S_h P U_{ad}(-\frac{1}{\alpha} S_h^* w). \quad (47)$$

L'equazione (47) contiene l'operatore $\mathbb{1}_{p_h(v)}$ 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 $\mathbb{1}_{\mathcal{I}(p_h)}$ come la funzione indicatrice

$$\mathbb{1}_{\mathcal{I}(p_h)} = \begin{cases} 1, & \omega \in \mathcal{I}(p_h), \\ 0, & \omega \in \Omega \times [0, T] \setminus \mathcal{I}(p_h), \end{cases} \quad (48)$$

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 (47) 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

$$\varphi(w) = \frac{1}{2} \|w\|_{L^2(I, L^2(\Omega))}^2 - \frac{\alpha}{2} \|u(w)\|_{L^2(I, L^2(\Omega))}^2 + (w, y_d - S_h u(w))_{L^2(I, L^2(\Omega))}. \quad (49)$$

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 (47) 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

$$\|\nabla\varphi(w^k)\| \leq t_0 \quad (50)$$

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 (\mathbb{P}_k) e (46) rispettivamente. Allora si verifica che

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

$$K(L, \beta) = \frac{\log(2) - \log(3L)}{\log(\beta)}, \quad (51)$$

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

$$\|\nabla\varphi(w)\|_{L^2(I, L^2(\Omega))} \geq \|w - w^*\|_{L^2(I, L^2(\Omega))}. \quad (52)$$

Per completezza si descrive l'algoritmo del gradiente coniugato adoperato per la risoluzione di (47)

Algoritmo 3. Per brevità sarà indicato con A_w l'Hessiano generalizzato a primo membro di (47), 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) $\alpha^n := \frac{(r^n, r^n)_{L^2(I, L^2(\Omega))}}{(p^n, A_w p^n)_{L^2(I, L^2(\Omega))}}$,
- (b) $\delta w^{n+1} := \delta w^n + \alpha^n p^n$,
- (c) $r^{n+1} := r^n - \alpha^n p^n$,
- (d) $\beta^n := \frac{(r^{n+1}, r^{n+1})_{L^2(I, L^2(\Omega))}}{(r^n, r^n)_{L^2(I, L^2(\Omega))}}$,
- (e) $p^{n+1} := r^{n+1} + \beta^n p^n$,

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

Come criterio di arresto è stato scelto

$$\|r^n\|_{L^2(I, L^2(\Omega))} \leq t_0 \quad (53)$$

con t_0 tolleranza prescritta. E' superfluo specificare che qui l'indice n denota l'n-esima iterazione del gradiente coniugato e non di Newton.

6 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. Durante tutte le fasi di sviluppo del codice è stata utilizzata la piattaforma di hosting web GitHub². Questo ha sia reso più fluida la collaborazione fra gli autori che garantito un maggiore controllo sull'avanzamento del codice.

6.1 Strumenti di sviluppo

6.1.1 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*.

6.1.2 GitHub

Git 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.

6.2 Implementazione

In questo lavoro sia l'algoritmo 1 di punto fisso che l'algoritmo 2 di Newton sono stati implementati per la soluzione del problema parabolico di controllo ottimo (\mathbb{P}) descritto nei capitoli precedenti. Il corpo del programma è costituito dal file `main.edp` che permette di risolvere il problema \mathbb{P}_k con uno dei due algoritmi su una o più griglie temporali. Il processo di raffinamento temporale considera

¹<http://www.freefem.org/>

²<https://github.com/>

griglie, della tipologia introdotta nella sezione 3, uniformi tali che il numero di nodi N_k al livello l di inaffinamento sia:

$$N_k = (2^l + 1) \quad (54)$$

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 esser confrontati graficamente coi valori della soluzione esatta, nei casi in cui quest'ultima sia nota.

L'errore fra la soluzione esatta e la soluzione approssimata per le funzioni di stato, aggiunta e controllo viene calcolato e salvato in un file di testo ad 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 le funzioni per l'algoritmo 1 di punto fisso e per semi Newton 2;
- `UabSet.edp`: contiene i parametri limite dell'insieme U_{ad} e le funzioni di proiezione per uno scalare e per un vettore sullo spazio U_{ad} ;
- `funzioni.edp` contiene le funzioni che caratterizzano l'operatore B , la funzione y_d e la forzante del problema di stato g_0 , che verrà definita nella sezione 7;
- `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 di Stiffness 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 di Stiffness 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 3;
- `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 dell'errore in tempo;

Per il calcolo di ogni norma e prodotto scalare il metodo numerico di integrazione in tempo utilizzato è il metodo di Cavalieri Simpson. È stata fatta questa scelta poiché questa formula di Newton-Cotes ha un errore di quadratura sufficientemente basso da non influenzare l'analisi degli errori oggetto di studio.

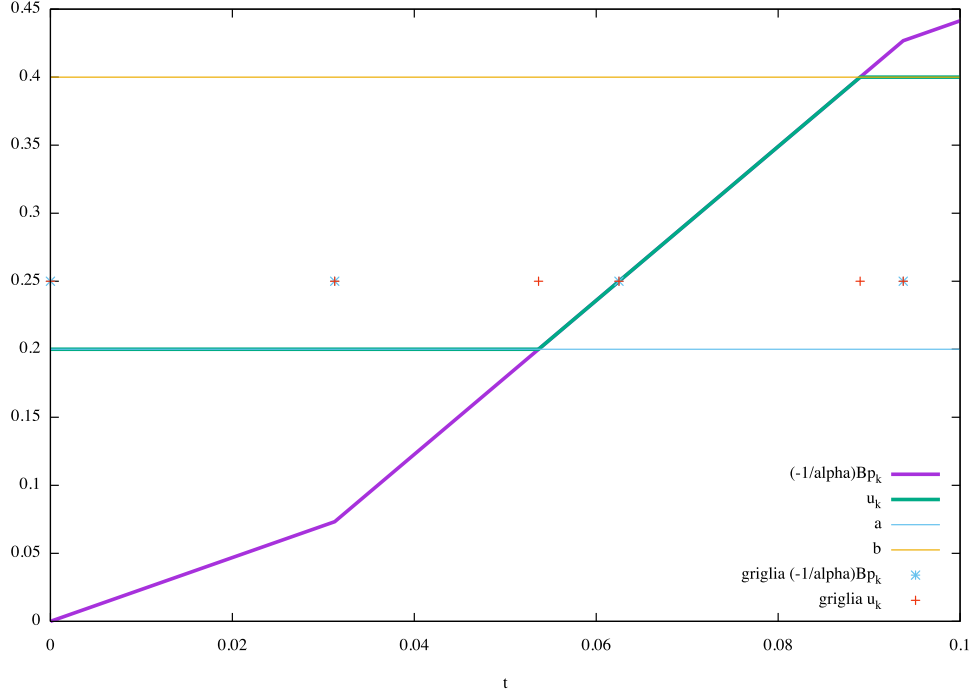


Figure 1: Esempio di differenza fra la griglia per la soluzione u_k e la soluzione di p_k

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.

6.2.1 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 continua lineare a tratti nel tempo i cui punti di non derivabilità coincidono con i nodi della griglia. La soluzione u_k , equazione (39), è ancora una funzione continua lineare 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 1. La funzione:

Listing 1: Funzione `controlErrL2`

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

Prende in ingresso la soluzione approssimata u_k e calcola la norma $L^2(I, L^2(\Omega))$ della differenza con la soluzione esatta \bar{u} , equazione (55), 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 integrazione rispetto a quello

dovuto all'approssimazione della soluzione.

$$err_{u_k} = \|\bar{u} - u_k\|_{L^2(I, L^2(\Omega))} \quad (55)$$

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

$$err_{\pi_{P_k^*} y_k} = \|\bar{y} - \pi_{P_k^*} y_k\|_{L^2(I, L^2(\Omega))} \quad (56)$$

dove \bar{y} è la soluzione esatta del problema di stato e $\pi_{P_k^*}$ è definito nella sezione 3. Il calcolo dell'errore (56) è eseguito dalla funzione `stateprojtrap()`.

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

$$err_{y_k} = \|\bar{y} - y_k\|_{L^2(I, L^2(\Omega))} \quad (57)$$

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

$$err_{p_k} = \|\bar{p} - p_k\|_{L^2(I, L^2(\Omega))} \quad (58)$$

6.2.2 Punto Fisso

Il primo procedimento implementato, di tipo Punto Fisso segue l'**Algoritmo 1**.

Lo schema di integrazione temporale 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 parametro di griglia $\frac{\Delta t}{2}$, poi lo schema di Crank-Nicolson ed infine un passo di Eulero in avanti(EA).

Listing 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
29

return 0;

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 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)
7                                           + dy(wold)*dy(wtest) ) )
8                           //contributo di u
9                           + int2d(Th)( gamma3*( g*ui*wtest ) )
10                          + int2d(Th)( gamma4*( g*uold*wtest ) )
11                          //contributo di g0
12                          + int2d(Th)( gamma3*( g0*wtest ) )
13                          + 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. Il termine forzante g_0 verrà introdotto nella sezione 7.

La funzione che calcola la soluzione y_k ad ogni passo è `statet()` che prima di terminare calcola anche l'errore (56) ed (57).

Anche per il problema aggiunto è stato utilizzato un metodo di Crank-Nicolson ed è possibile trovare una matrice di Stiffness principale invariante nel tempo:

Listing 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:

$$h_i = y_{k_i} - y_d(t_i), \quad hnext_i = y_{k_i} - y_d(t_i + \Delta t) \quad (59)$$

dove t_i è il tempo considerato all'iterazione i -esima³. La funzione che calcola la soluzione p_k ad ogni passo è la funzione `adjoinntt()` che prima di terminare valuta e salva l'errore per il problema aggiunto definito da (58). Inoltre in questa funzione vengono calcolati i valori della soluzione di controllo non proiettata:

$$u_{k_i} = -\frac{1}{\alpha} B' p_{k_i} \quad (60)$$

³Come si vede dalla formulazione debole (31)

Infine la funzione `puntoFisso()` risolve l'**Algoritmo 1** rispettando il criterio di tolleranza (42).

6.2.3 Semi-Newton

Il secondo procedimento implementato, di tipo semi Newton, segue l'**Algoritmo 2** e richiede l'ausilio del metodo del gradiente coniugato, 3.

Per la soluzione del problema di stato son state implementate due diverse funzioni. La prima chiamata `stateCG(real[int] &xx)` risolve il problema di stato ad ogni passo temporale dato un termine noto e crea/restituisce il vettore delle soluzioni. L'operatore implementato è S_h applicato a $P_{U_{ad}}(xx)$. In particolare il termine noto, per una corretta proiezione sullo spazio U_{ad} , deve essere della forma:

$$xx = -\frac{1}{\alpha} * S_h^*(w) \quad (61)$$

La seconda, denominata `mat1(real[int] &xx, real[int] &ww)`, rappresenta invece l'operatore $S_h \mathbb{1}_{S_h^* w}$ applicato sempre a un termine noto xx . Questa operazione è necessaria per la costruzione di (47). Il vettore in ingresso ww contiene i valori dell'indicatrice $\mathbb{1}_{S_h^* w}$. L'algoritmo seguito da queste funzioni è simile. Le differenze principali si trovano nella definizione del termine noto per il problema di stato. Nel primo caso verrà introdotta:

Listing 5: Matrice `tnnew(w,wtest)` per `stateCG(real[int] &xx)`

```

1 varf tnnew(w,wtest) = int2d(Th)( wold*wtest/dt )
2                       - int2d(Th)( gamma2*( dx(wold)*dx(wtest) + dy(wold)*dy(wtest) ) )
3                       //contributo di u
4                       + int2d(Th)(gamma3*g*(PUabScal(listate)*wtest))
5                       + int2d(Th)(gamma4*g*(PUabScal(loldstate)*wtest))
6                       //contributo di go
7                       + int2d(Th)( gamma3*( g0*wtest ))
8                       + int2d(Th)( gamma4*( g0old*wtest ));

```

dove le variabili *listate* e *loldstate* rappresentano i valori dell'operazione di controllo dovuta al moltiplicatore di lagrange, mentre g_0 è il termine forzante.

Listing 6: Matrice `tnnew(w,wtest)` per `mat1(real[int] &xx, real[int] &ww)`

```

1 varf tnnew(w,wtest) = int2d(Th)( wold*wtest/dt )
2                       - int2d(Th)( gamma2*( dx(wold)*dx(wtest) + dy(wold)*dy(wtest) ) )
3                       + int2d(Th)(gamma3*g*((indnew<=b)*(indnew>=a)*listate*wtest))
4                       + int2d(Th)(gamma4*g*((indold<=b)*(indold>=a)*loldstate*wtest));

```

dove le variabili *listate*, *loldstate* rappresentano i valori di $S_h^* \delta w$, mentre *indnew* ed *indold* quelli dell'indicatrice $\mathbb{1}_{S_h^* w}$. In entrambe lo schema di integrazione temporale per il problema di stato segue la variante dello schema di Crank-Nicolson(CN) introdotta precedentemente per l'**Algoritmo 1** di punto fisso.

La funzione `adjCG(real[int,int] &xx)` implementa l'operatore S_h^* ; risolve il problema aggiunto ad ogni passo temporale dato un termine noto xx e restituisce la soluzione in un opportuno vettore. Il procedimento utilizzato rispecchia quello precedentemente implementato per il metodo di punto fisso. La funzione `CG(real[int,int] &xx, real[int,int] &ww)` risolve il sistema di equazioni (47) tramite il metodo del gradiente coniugato introdotto in 3. Il vettore ww corrisponde al valore iniziale δw^0 mentre xx la funzione w per l'iterazione di Newton in questione. Infine la funzione `DampedNewton()` risolve l'**Algoritmo 2** rispettando il criterio di tolleranza (50).

7 Risultati numerici

In questo capitolo verranno esposti i risultati numerici ottenuti. I due esempi utilizzati come casi test sono stati presi da [NMM15]. In entrambi questi esempi verrà considerato l'operatore lineare affine \tilde{B} (11).

Si considera come dominio spaziale il quadrato $\Omega = (0, 1)^2$ ed una mesh uniforme definita da

Listing 7: mesh in spazio square(150, 150, flags=1)

-- Square mesh : nb vertices = 22801 , nb triangles = 45000 , nb boundary edges 600

7.1 Test Case 01

7.1.1 Set-up

Il primo esempio considerato è in accordo con l'ipotesi i) introdotta nella sezione 2. 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:

$$\begin{aligned} g_1(x_1, x_2) &= \sin(\pi x_1) \sin(\pi x_2) \\ g_0(t, x_1, x_2) &= -\pi^2 w_a(t, x_1, x_2) - BP_{U_{ad}} \left(-\frac{1}{4\alpha} (\exp(a\pi^2 t) - \exp(a\pi^2 T)) \right) \end{aligned} \quad (62)$$

dove

$$w_a(t, x_1, x_2) = \exp(a\pi^2 t) \sin(\pi x_1) \sin(\pi x_2), \quad a \in \mathbb{R} \quad (63)$$

In particolare vengono considerate le costanti $a = -\sqrt{5}$ ed $\alpha = \pi^{-4}$. Come conseguenza di ciò si ha che (13) verrà riscritta, utilizzando l'aggiunto di B e non \tilde{B} , come:

$$(B'z)(t) = \int_{\Omega} z(t, x_1, x_2) g_1(x_1, x_2), dx_1 dx_2 \quad (64)$$

Si definiscono ora:

$$\begin{aligned} y_d(t, x_1, x_2) &= \frac{a^2-5}{2+a} \pi^2 w_a(t, x_1, x_2) + 2\pi^2 w_a(T, x_1, x_2) \\ y_0(x_1, x_2) &= \frac{-1}{2+a} \pi^2 w_a(0, x_1, x_2) \end{aligned} \quad (65)$$

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} :

$$\begin{aligned} \bar{u}(t, x_1, x_2) &= P_{U_{ad}} \left(-\frac{1}{4\alpha} (\exp(a\pi^2 t) - \exp(a\pi^2 T)) \right) \\ \bar{y}(t, x_1, x_2) &= \frac{-1}{2+a} \pi^2 w_a(0, x_1, x_2) \\ \bar{p}(t, x_1, x_2) &= w_a(t, x_1, x_2) - w_a(T, x_1, x_2) \end{aligned} \quad (66)$$

Table 1: Punto fisso per Test case 01: errori e EOC

1	$\ \bar{u} - u_{kh}\ _{L^2(L^2)}$	$\ \bar{y} - y_{kh}\ _{L^2(L^2)}$	EOC_u	EOC_y
1	0.31667	0.981285	—	—
2	0.0835064	0.496296	2.60937	1.33449
3	0.0209608	0.248822	2.35165	1.17464
4	0.00500916	0.124494	2.25065	1.08882
5	0.00109219	0.0622586	2.29624	1.04473
6	0.000497644	0.0311327	1.15957	1.02236

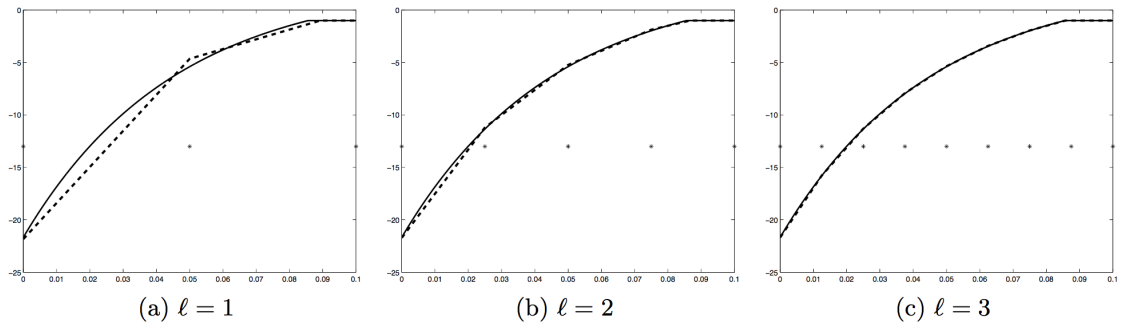
Table 2: Punto fisso per Test case 01: errori e EOC

1	$\ \bar{y} - \pi_{P_k^*} y_{kh}\ _{L^2(L^2)}$	$\ \bar{p} - p_{kh}\ _{L^2(L^2)}$	EOC_{π_y}	EOC_p
1	0.520894	0.00660747	—	—
2	0.15134	0.00173155	2.41965	2.6216
3	0.0393476	0.00043334	2.29181	2.35673
4	0.00970087	0.000103613	2.20164	2.24982
5	0.00221619	0.000022824	2.2259	2.28078
6	0.000432024	0.000010724	2.41203	1.11429

7.1.2 Risultati Numerici

La tabelle 1, 2, 3 e 4 mostrano il comportamento degli errori considerati e il valore dell'ordine di convergenza per diverse griglie temporali. Si può notare che sia per l'algoritmo di punto fisso che quello di semi-Newton l'andamento dell'errore 55 è di $O(k^2)$ in accordo col risultato teorico 38 e con i risultati numerici di [NMM15]. In particolare si rileva che l'ordine di convergenza diminuisce per l'ultimo livello in cui la griglia temporale non è più indipendente da quella spaziale. Essendo la variabile di stato costante a tratti nel tempo l'ordine di convergenza riscontrato è solo uno, ma un ordine due è ottenuto per la proiezione della variabile di stato $\pi_{P^*_k} y_k$ conformemente con il **Lemma 2**.

In figura 62 sono riportati i risultati del paper [NMM15] nei primi tre livelli di raffinamento di griglia. Mentre in 63 e 64 quelli ottenuti con il codice implementato per questo lavoro per ogni

Figure 2: Test Case 01 \bar{u} e u_k risultati di [NMM15]

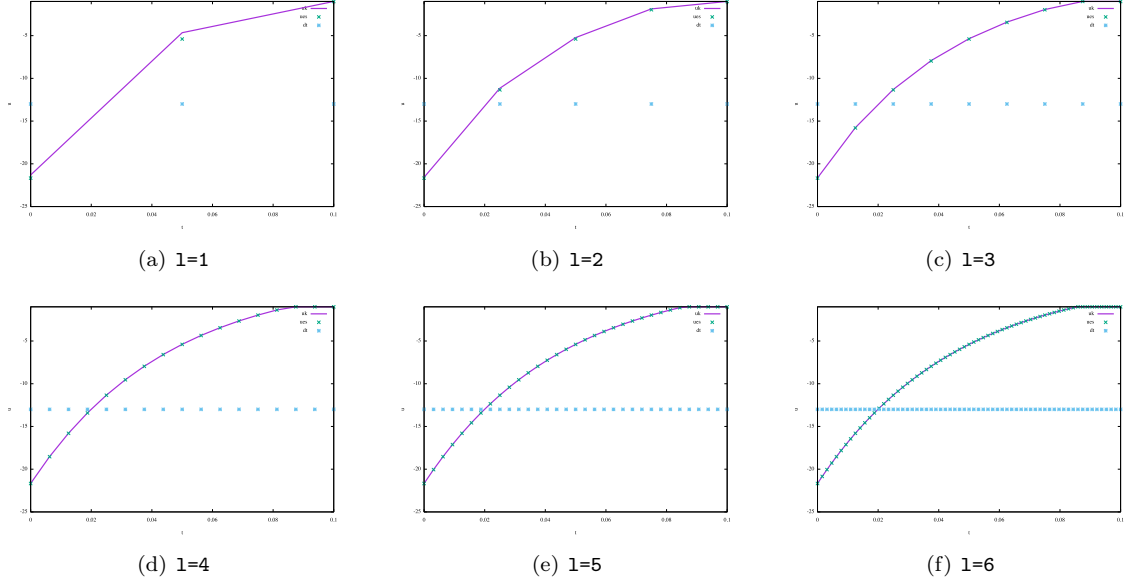


Figure 3: Test Case 01 \bar{u} e u_k risultati dell'algoritmo di punto fisso

Table 3: Newton per Test case 01: errori e EOC

l	$\ \bar{u} - u_{kh}\ _{L^2(L^2)}$	$\ \bar{y} - y_{kh}\ _{L^2(L^2)}$	EOC_u	EOC_y
1	0.316669	0.981285	—	—
2	0.0835064	0.496296	2.60937	1.33449
3	0.0209612	0.248822	2.35161	1.17464
4	0.00500971	0.124494	2.25051	1.08882
5	0.00109312	0.0622586	2.29512	1.04473
6	0.000497833	0.0311327	1.16028	1.02236

Table 4: Newton per Test case 01: errori e EOC

l	$\ \bar{y} - \pi_{P_k^*} y_{kh}\ _{L^2(L^2)}$	$\ \bar{p} - p_{kh}\ _{L^2(L^2)}$	$EOC_{\pi y}$	EOC_p
1	0.520894	0.00660747	—	—
2	0.15134	0.00173155	2.41965	2.6216
3	0.0393476	0.00043334	2.29181	2.35672
4	0.00970089	0.000103614	2.20164	2.24981
5	0.00221621	0.0000228246	2.22589	2.28078
6	0.000432023	0.0000107235	2.41204	1.11436

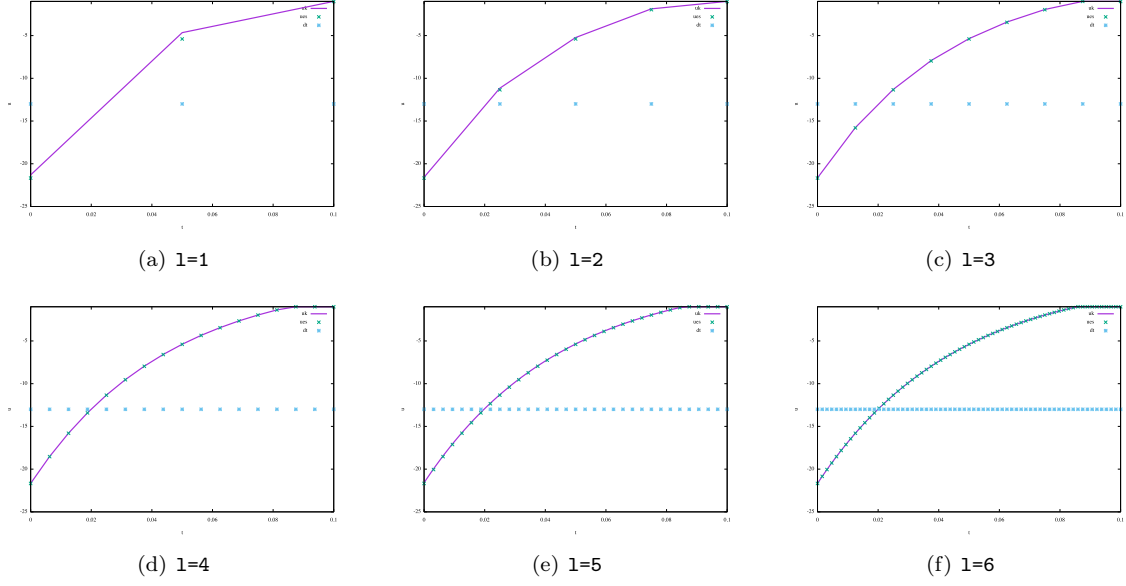


Figure 4: Test Case 01 \bar{u} e u_k : risultati dell'algoritmo semi Newton

livello considerato. In entrambi i casi si vede che una buona approssimazione della soluzione esatta viene raggiunta.

Per l'algoritmo di punto fisso sono richieste cinque o sei iterazioni per ogni griglia temporale prima di arrivare a convergenza, considerando un valore di tolleranza di $tol = 10^{-5}$. Mentre per quello di Newton ne bastano quattro.

7.2 Test Case 02

7.2.1 Set-up

Il secondo esempio considerato è in accordo con l'ipotesi i) introdotta in 2. 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:

$$g_1(x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2) \quad (67)$$

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

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

$$y_d(t, x_1, x_2) = g_1 \left(\cos\left(\frac{t}{T} 2\pi a\right) (1 - 2\pi^2) - \frac{2\pi a}{T} \sin\left(\frac{t}{T} 2\pi a\right) + 2\pi^2 \cos(2\pi a) \right) \quad (68)$$

$$y_0(x_1, x_2) = g_1(x_1, x_2)$$

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} :

$$\begin{aligned}\bar{u}(t, x_1, x_2) &= P_{U_{ad}} \left(-\frac{1}{4\alpha} \cos \left(\frac{t}{T} 2\pi a \right) + \frac{1}{4\alpha} \right) \\ \bar{y}(t, x_1, x_2) &= \frac{-1}{2+a} \pi^2 w_a(0, x_1, x_2) \\ \bar{p}(t, x_1, x_2) &= w_a(t, x_1, x_2) - w_a(T, x_1, x_2)\end{aligned}\tag{69}$$

dove

$$w_a(t, x_1, x_2) = \cos \left(\frac{t}{T} 2\pi a \right) \cdot g_1(t, x_1, x_2)\tag{70}$$

7.2.2 Risultati Numerici

I risultati per gli errori di convergenza sono gli stessi tratti per il caso 01 e rispecchiano quelli esposti in [NMM15]. Essendo la soluzione esatta \bar{u} più complessa in questo test case, si nota che i primi due livelli di griglia non danno risultati rappresentativi. Infatti il numero di punti di non derivabilità della \bar{u} è maggiore dei nodi della griglia temporale considerata. La perdita riscontrata nell'ordine di convergenza per l'ultima mesh temporale è invece da attribuire all'influenza dell'errore in spazio.

In figura 65 sono riportati i risultati del paper [NMM15] nei primi tre livelli di raffinamento di griglia. Mentre in 66 e 67 quelli ottenuti con il codice implementato per questo lavoro per ogni livello considerato. In entrambi i casi si vede che una buona approssimazione della soluzione esatta viene raggiunta.

Per l'algoritmo di punto fisso sono richieste due iterazioni per ogni griglia temporale prima di arrivare a convergenza, considerando un valore di tolleranza di $toll = 10^{-5}$, mentre per quello di Newton ne bastano tre se viene introdotto un criterio di tolleranza aggiuntivo tale per cui:

$$\|\nabla\varphi(w^k)\| - \|\nabla\varphi(w^{k+1})\| > t_0\tag{71}$$

oppure quattro se viene considerato invece:

$$\|\nabla\varphi(w^k) - \nabla\varphi(w^{k+1})\| > t_0\tag{72}$$

Col criterio 71 l'algoritmo converge in tre passi per ognuna delle mesh considerate. Mentre utilizzando 72 si ha la convergenza dopo nove passi per la prima griglia utilizzata e quattro dalla successiva in avanti.

8 Conclusioni e Lavori Futuri

Entrambi gli algoritmi implementati confermano i risultati teorici per l'ordine di convergenza dell'errore nei problemi di controllo ottimo parabolico se viene utilizzato uno schema di Petrov Galerkin. Per riscontrare questi esiti è necessario che l'errore in tempo non raggiunga la dimensione dell'errore in spazio. Inoltre la griglia temporale considerata deve avere un numero sufficiente di nodi per approssimare eventuali punti di non derivabilità nella funzione esatta.

Lavori futuri potrebbero vertere sull'analisi dell'estensione dei teoremi di semi-Newton nel caso per i problemi di controllo ottimo parabolici.

Nel codice potrebbe essere introdotta la lettura e scrittura su file per le soluzioni del problema di stato ed aggiunto in modo da ridurre il rischio di problemi di memoria, non riscontrati nei casi trattati da questo lavoro.

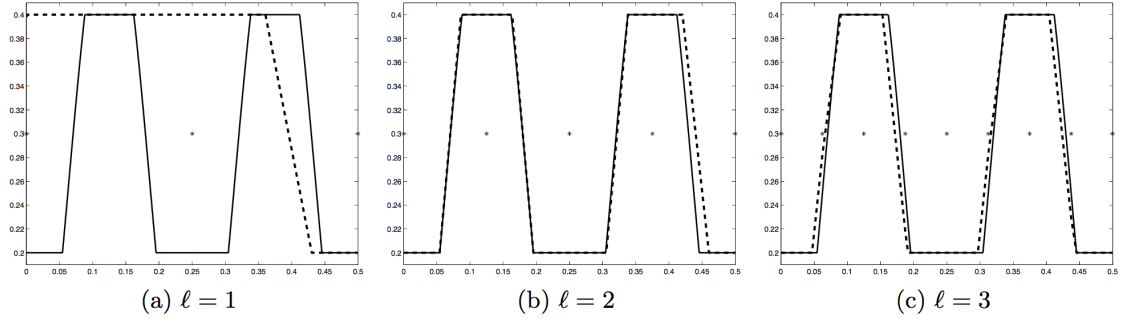


Figure 5: Test Case 02 \bar{u} e u_k risultati di [NMM15]

Table 5: Punto fisso per Test case II: errori e EOC

1	$\ \bar{u} - u_{kh}\ _{L^2(L^2)}$	$\ \bar{y} - y_{kh}\ _{L^2(L^2)}$	EOC_u	EOC_y
1	0.11547	0.577323	—	—
2	0.049267	0.463462	1.66741	0.430045
3	0.0229418	0.136413	1.30029	2.08074
4	0.0036817	0.0594452	2.87676	1.30605
5	0.000908002	0.0286805	2.1105	1.09882
6	0.000229782	0.0142118	2.02708	1.0358
7	0.0000652737	0.0070899	1.83615	1.01455
8	0.0000267529	0.003543	1.29406	1.00643

Table 6: Punto fisso per Test case II: errori e EOC

1	$\ \bar{y} - \pi_{P_k^*} y_{kh}\ _{L^2(L^2)}$	$\ \bar{p} - p_{kh}\ _{L^2(L^2)}$	$EOC_{\pi y}$	EOC_p
1	0.408219	0.57735	—	—
2	0.428289	0.181146	0.0939571	2.26916
3	0.108057	0.0846821	2.34293	1.29366
4	0.0223162	0.0224906	2.48014	2.08464
5	0.00443799	0.00571816	2.43498	2.06462
6	0.000929298	0.00145281	2.3065	2.02122
7	0.000203934	0.000385864	2.21269	1.93423
8	0.0000465654	0.000130114	2.14277	1.57715

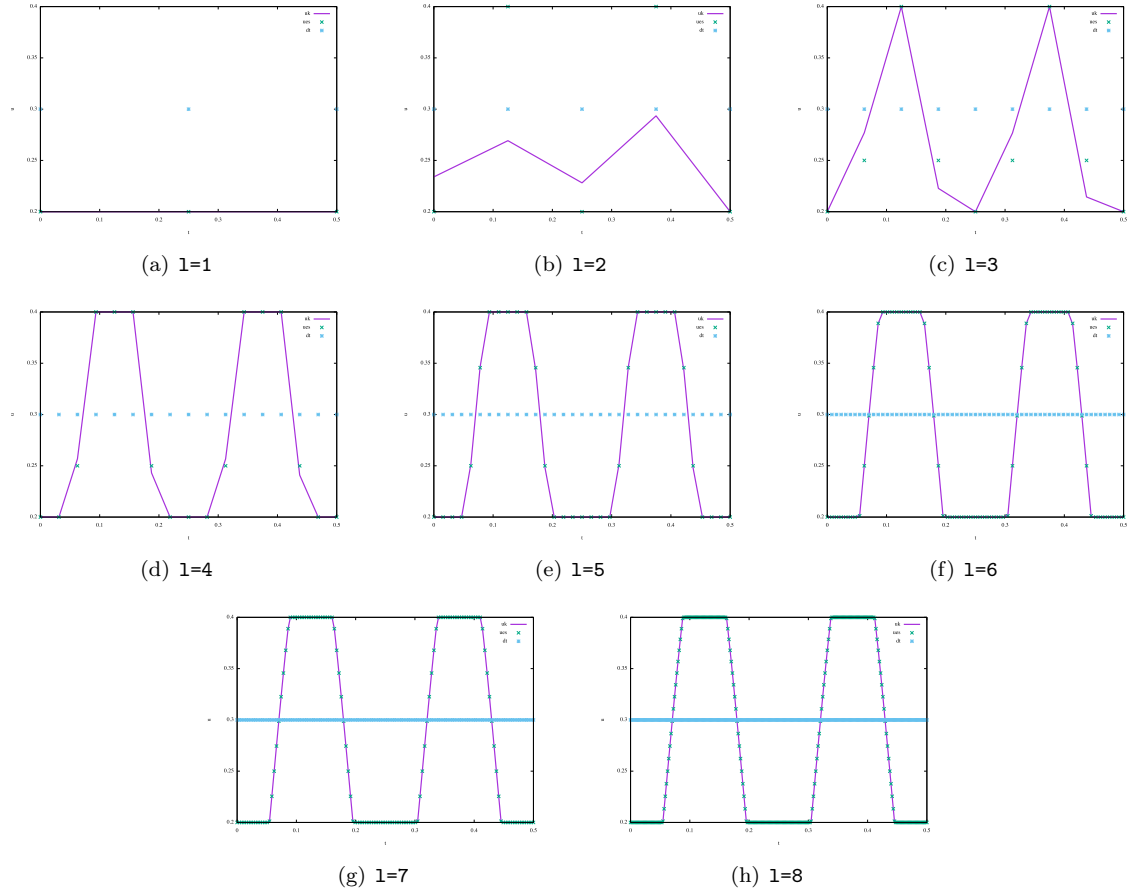


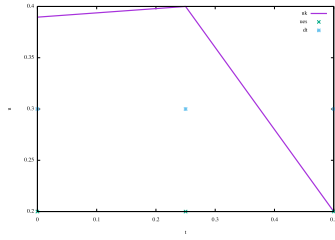
Figure 6: Test Case 02 \bar{u} e u_k risultati dell'algoritmo di punto fisso

Table 7: Newton per Test case II: errori e EOC

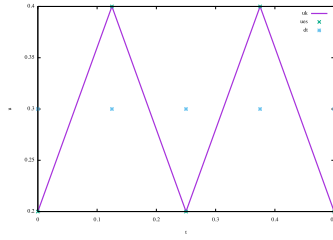
l	$\ \bar{u} - u_{kh}\ _{L^2(L^2)}$	$\ \bar{y} - y_{kh}\ _{L^2(L^2)}$	EOC_u	EOC_y
1	0.115495	0.579976	—	—
2	0.076157	0.462076	0.815213	0.444884
3	0.0127336	0.136402	3.04286	2.07579
4	0.00222359	0.0594457	2.74395	1.30591
5	0.000541362	0.0286803	2.12996	1.09884
6	0.000138463	0.0142117	2.01139	1.03579
7	0.0000349821	0.0070899	2.00717	1.01455
8	0.0000178053	0.003543	0.979797	1.00643

Table 8: Newton per Test case II: errori e EOC

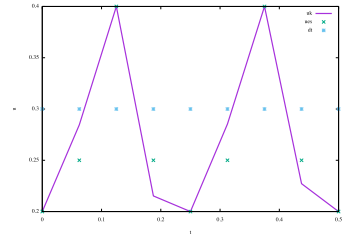
l	$\ \bar{y} - \pi_{P_k^*} y_{kh}\ _{L^2(L^2)}$	$\ \bar{p} - p_{kh}\ _{L^2(L^2)}$	EOC_{π_y}	EOC_p
1	0.410488	0.57748	—	—
2	0.427682	0.181205	0.080328	2.26897
3	0.108041	0.0846824	2.34076	1.29421
4	0.0223131	0.0224907	2.48014	2.08463
5	0.00443687	0.00571824	2.43516	2.06461
6	0.0009289	0.00145285	2.30675	2.02121
7	0.000203852	0.000385873	2.21265	1.93423
8	0.0000465544	0.000130118	2.14254	1.57714



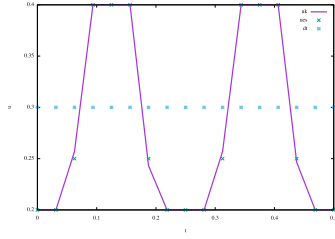
(a) l=1



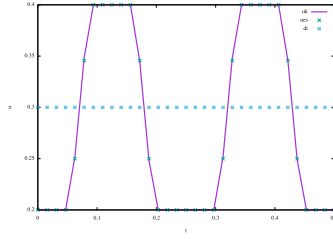
(b) l=2



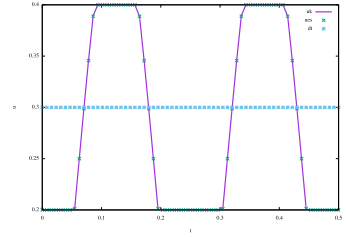
(c) l=3



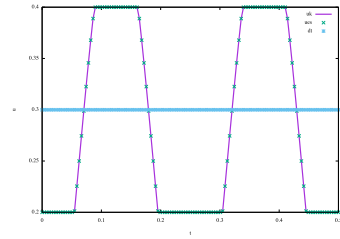
(d) l=4



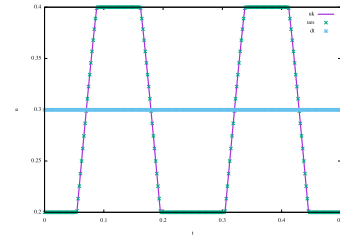
(e) l=5



(f) l=6



(g) l=7



(h) l=8

Figure 7: Test Case 02 \bar{u} e u_k : risultati dell'algoritmo di semi newton

Il codice sviluppato permette l'utilizzo di diversi spazi funzionali in Ω per il problema di stato e di aggiunto nel caso di punto fisso ma non di semi-Newton. Questo potrebbe essere introdotto anche per il secondo algoritmo.

References

- [BGS06] P. Benner, S. Görner, and J. Saak. “Numerical Solution of Optimal Control Problems for Parabolic Systems”. In: *Lecture Notes in Computational Science and Engineering* 52.1 (2006), pp. 151–169.
- [BIK99] M. Bergounioux, K. Ito, and K. Kunisch. “Primal-Dual Strategy for Constrained Optimal Control Problems”. In: *SIAM Journal on Control and Optimization* 37.4 (1999), pp. 1176–1194.
- [BMV08] R. Becker, D. Meidner, and B. Vexler. “Efficient numerical solution of parabolic optimization problems by finite element methods”. In: *Optimisation Methods and Software* 22.5 (2008), pp. 813–833.
- [BV14] Z. Brakerski and V. Vaikuntanathan. “Downloaded 03 / 29 / 16 to 131 . 180 . 145 . 249 . Redistribution subject to SIAM license or copyright ; see <http://www.siam.org/journals/ojsa.php>”. In: 43.2 (2014), pp. 831–871.
- [CMM97] Z. Cai, T. A. Manteuffel, and S. F. McCormick. “Downloaded 01 / 18 / 16 to 131 . 94 . 117 . 94 . Redistribution subject to SIAM license or copyright ; see <http://www.siam.org/journals/ojsa.php>”. In: *SIAM Journal on Numerical Analysis* 34.5 (1997), pp. 1727–1741.
- [GK09] C. Graeser and R. Kornhuber. “Nonsmooth Newton Methods for Set-Valued Saddle Point Problems”. In: *SIAM Journal on Numerical Analysis* 47.2 (2009), pp. 1251–1273.
- [Hin05] M. Hinze. “A variational discretization concept in control constrained optimization: The linear-quadratic case”. In: *Computational Optimization and Applications* 30.1 (2005), pp. 45–61.
- [HPH14] F. Hecht, O. Piro, and A. L. Hyaric. “Freefem++”. In: (2014).
- [HV09] M. Hinze and M. Vierling. “constrained optimization with control constraints arXiv : 0912 . 0488v1 [math . OC] 2 Dec 2009”. In: (2009), pp. 1–14.
- [HV10] M. Hinze and M. Vierling. “A globalized semi-smooth Newton method for variational discretization of control constrained elliptic optimal control problems”. In: (2010), pp. 1–12.
- [HV12] M. Hinze and M. Vierling. “The semi-smooth Newton method for variationally discretized control constrained elliptic optimal control problems ; Implementation , convergence and globalization A globalized semi-smooth Newton method for variational discretization of control constrained”. In: March 2016 (2012).
- [KV07] K. Kunisch and B. Vexler. “Constrained Dirichlet Boundary Control in L^2 for a Class of Evolution Equations”. In: *SIAM Journal on Control and Optimization* 46.5 (2007), pp. 1726–1753.
- [MPC15] J. Melorose, R. Perroy, and S. Careas. “No Title No Title”. In: *Statewide Agricultural Land Use Baseline 2015* 1 (2015).
- [MV11] D. Meidner and B. Vexler. “A priori error analysis of the Petrov Galerkin Crank Nicolson scheme for parabolic optimal control problems”. In: 49.5 (2011), pp. 2183–2211.

- [NMM15] V. D. Nikolaus, H. Michael, and V. Morten. “Crank-Nicolson time stepping and variational discretization of control-constrained parabolic optimal control problems”. In: *SIAM J. Control and Optimization* 53(3) (March 6, 2015), pp. 1182–1198.
- [NW06] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Science and Business Media, 2006.
- [Opf11] G. Opfer. “Hamburger Beitrage”. In: *SciencesNew York* 2011.9 (2011), p. 22.
- [Qua13] A. Quarteroni. *Modellistica Numerica per Problemi Differenziali*. Springer Science and Business Media, 30 luglio 2013.
- [Ran84] R. Rannacher. “Numerische MathemaUk Finite Element Solution of Diffusion Problems with Irregular Data”. In: 327 (1984), pp. 309–327.
- [SYK14] Z. K. Seymen, H. Yücel, and B. Karasözen. “Journal of Computational and Applied Distributed optimal control of time-dependent diffusion – convection – reaction equations using space – time discretization”. In: 261 (2014), pp. 146–157.
- [Ulb02] M. Ulbrich. “Nonsmooth Newton-like methods for variational inequalities and constrained optimization problems in function spaces”. In: *Convergence* February (2002).
- [Vie14] M. Vierling. “Parabolic optimal control problems on evolving surfaces subject to pointwise box constraints on the control - Theory and numerical realization”. In: *Interfaces and Free Boundaries* 16.2 (2014), pp. 137–173.