

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI  
Corso di Laurea Magistrale in Matematica

**METODI DI REGRESSIONE  
PER PROGRAMMAZIONE  
DINAMICA STOCASTICA**

Tesi di Laurea in Equazioni Differenziali Stocastiche

Relatore:  
Chiar.mo Prof.  
ANDREA PASCUCCI

Presentata da:  
ELENA SGUBBI

Sessione V  
Anno Accademico 2016 - 2017



*Alla mia famiglia:  
a babbo e mamma,  
a Lorisa e Mirco,  
a Silvano e Katia,  
a Gabriele e Chiara,  
ad Andrea.*



# Indice

<b>Indice</b>	<b>I</b>
<b>Introduzione</b>	<b>III</b>
<b>1 Metodo di Eulero</b>	<b>1</b>
1.1 Metodo di Eulero per equazioni ordinarie . . . . .	1
1.2 Metodo di Eulero per equazioni stocastiche . . . . .	4
1.3 Moto Browniano . . . . .	6
1.4 Moto Browniano geometrico . . . . .	9
1.4.1 Algoritmi per il moto Browniano geometrico . . . . .	10
1.4.2 Il moto Browniano geometrico in finanza . . . . .	13
1.5 Modelli Gaussiani . . . . .	14
1.6 Square-root diffusions . . . . .	15
<b>2 Il Metodo Monte Carlo</b>	<b>17</b>
2.1 Un po' di storia . . . . .	17
2.2 La tecnica Monte Carlo . . . . .	18
2.3 Stima dell'errore . . . . .	19
2.4 Esempi . . . . .	22
2.5 Tecniche di riduzione della varianza . . . . .	24
<b>3 Metodo di Longstaff Schwartz</b>	<b>29</b>
3.1 Procedimento di Longstaff e Schwartz . . . . .	30
3.2 Esempio numerico . . . . .	37
3.3 Esempi e tabelle di put Americane . . . . .	44
3.4 Grafici e confronti . . . . .	48
<b>4 Conclusioni</b>	<b>71</b>

<b>A Codici Matlab</b>	<b>73</b>
A.1 Algoritmo di Longstaff-Schwartz . . . . .	73
A.2 Simulazione del moto Browniano geometrico . . . . .	75
A.3 Funzioni base per la regressione . . . . .	76
<b>Bibliografia</b>	<b>81</b>

# Introduzione

La trattazione dei derivati finanziari di tipo Americano è un argomento attuale e ancora oggetto di ricerche.

La difficoltà nella trattazione di queste opzioni risiede principalmente nella loro caratteristica di esercizio: possono essere esercitate in qualsiasi momento fino alla data di scadenza. La modellazione di un derivato Americano deve quindi tener conto della continua possibilità di esercizio, oltre all'andamento stocastico del sottostante.

Questa tesi vuole descrivere il metodo di regressione per la programmazione dinamica stocastica, con particolare attenzione all'approccio di Longstaff e Schwartz, al fine di valutare un'opzione Americana.

L'idea di fondo consiste nell'utilizzare la programmazione dinamica stocastica discretizzando il tempo per modellizzare l'opzione; si valuta il derivato al tempo di scadenza e, attraverso il metodo di regressione, si va a ritroso nel tempo per valutare se conviene esercitare l'opzione o portarla a scadenza.

Precisamente, nel *primo capitolo* si analizza il metodo di Eulero, che consente la modellazione del derivato finanziario.

In particolare si considera il metodo per le equazioni differenziali stocastiche che descrivono il moto Browniano geometrico e si confrontano diversi approcci implementativi al fine di ottenere il più efficiente.

Nel *secondo capitolo* si considera il metodo Monte Carlo.

Dopo aver visto velocemente la sua storia si analizza la tecnica, che trova fondamento nel campionamento aleatorio e nel teorema del limite centrale. In ambito finanziario il metodo Monte Carlo viene utilizzato per la valutazione dei derivati, in questo capitolo si applica nella valutazione del payoff di un'opzione call e di un'opzione Asiatica. Si stima inoltre l'errore dovuto al campionamento aleatorio e, per contenerlo, si applicano tecniche di riduzione della varianza in quanto l'errore è direttamente proporzionale alla varianza.

Il *terzo capitolo* tratta il metodo di Longstaff-Schwartz per la valutazione di un'opzione Americana.

Il cuore di questa tecnica è dato dal metodo di regressione applicato alla programmazione dinamica. In questa sezione si descrivono in dettaglio i passaggi che portano all'implementazione finale del metodo, elencando inoltre le diverse tipologie di polinomi applicabili

nel metodo di regressione. Dopo aver visto un esempio numerico semplice che permette di osservare meglio il funzionamento di tale tecnica, si applica l'algoritmo ad esempi concreti variando il valore iniziale, la volatilità ed il tempo di scadenza dell'opzione in modo da osservare l'andamento in diversi casi. Attraverso l'utilizzo di tabelle e grafici si confrontano i risultati ottenuti con Longstaff-Schwartz rispetto ai valori calcolati con altri metodi, inoltre si possono confrontare le valutazioni ottenute dai metodi di regressione utilizzando i polinomi visti.

Nel *quarto capitolo* vengono brevemente riepilogate alcune caratteristiche del metodo e si forniscono alcuni spunti per progetti futuri.

Infine in *Appendice* vengono riportati gli algoritmi in linguaggio di programmazione Matlab, utilizzati per implementare l'approccio di Longstaff e Schwartz al metodo di regressione per programmazione dinamica stocastica.



# Capitolo 1

## Metodo di Eulero

Il metodo di Eulero è una tecnica di approssimazione numerica.

L'approssimazione numerica risulta necessaria nei casi in cui non sia possibile determinare in modo esplicito la soluzione di un'equazione.

Si hanno diversi metodi numerici per risolvere un'equazione, ma tutti hanno la stessa idea di base: approssimare le derivate con rapporti incrementali.

In generale, per mostrare la validità di uno schema numerico, ovvero per dimostrare che la soluzione dell'equazione discretizzata approssima la soluzione esatta dell'equazione data, è necessario verificare tre elementi:

- la regolarità della soluzione
- la consistenza della discretizzazione
- la stabilità dello schema numerico

### 1.1 Metodo di Eulero per equazioni ordinarie

Innanzitutto analizziamo il metodo di Eulero per le equazioni differenziali ordinarie.

Sia  $\mu : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$  continua, tale che:

-  $\mu$  sia a crescita lineare:

$$|\mu(t, x)| \leq k(1 + |x|), \quad x \in \mathbb{R},$$

-  $\mu$  sia Lipschitziana in entrambe le variabili:

$$|\mu(t, x) - \mu(s, y)| \leq k(|t - s| + |x - y|), \quad x, y \in \mathbb{R}, \quad t, s \in [0, T],$$

e considero l'equazione differenziale ordinaria:

$$X'_t = \mu(t, X_t), \quad t \in [0, T]. \quad (1.1)$$

Lo schema numerico di Eulero consiste nel fissare un valore  $N \in \mathbb{N}$  e suddividere l'intervallo  $[0, T]$  in  $N$  intervalli più piccoli  $[t_n, t_{n+1}]$ , per  $n = 0, \dots, N$ . In questo modo i nuovi intervalli definiti hanno lunghezza pari al passo della discretizzazione, ovvero  $\delta = \frac{T}{N}$ .

Si approssima quindi la derivata  $X'_t$  con il rapporto incrementale:

$$X'_t = \frac{X_{t_{n+1}}^\delta - X_{t_n}^\delta}{\delta}.$$

Sostituendo il rapporto incrementale nell'equazione (1.1), si ottiene la seguente discretizzazione:

$$X_{t_{n+1}}^\delta = X_{t_n}^\delta + \mu(t_n, X_{t_n}^\delta)\delta, \quad n = 1, \dots, N. \quad (1.2)$$

Ponendo il valore iniziale  $X_0^\delta = X_0$ , la (1.2) definisce la ricorsione per l'algoritmo che a sua volta determina l'approssimazione della soluzione esatta  $X$ .

Per analizzare la regolarità, la consistenza e la stabilità del metodo di Eulero consideriamo la versione integrale dell'equazione differenziale (1.1):

$$L_t X = 0, \quad t \in [0, T], \quad (1.3)$$

con operatore  $L_t$  definito come:

$$L_t X = X_t - X_0 - \int_0^t \mu(s, X_s) ds, \quad t \in [0, T]. \quad (1.4)$$

Analogamente a prima, si ottiene l'equazione discretizzata  $L_t^\delta X^\delta = 0$ , con operatore discretizzato  $L_t^\delta$ :

$$L_t^\delta X = X_t - X_0 - \int_0^t \sum_{n=1}^N \mu(t_{n-1}, X_{t_{n-1}}) \mathbb{1}_{]t_{n-1}, t_n]}(s) ds, \quad t \in [0, T]. \quad (1.5)$$

L'equazione discretizzata definisce la stessa approssimazione di  $X$  vista in (1.2).

Analizziamo ora la convergenza dello schema di Eulero per un'equazione differenziale. Per le dimostrazioni dei risultati che seguono, rimandiamo ad [1].

**Proposizione 1** (Regolarità). *Sia  $X$  la soluzione dell'equazione differenziale integrale data dall'operatore  $L_t$ , come in (1.5).*

*Allora  $X$  è tale che:*

$$|X_t - X_s| \leq K_1 |t - s|, \quad t, s \in [0, T],$$

*con  $K_1$  costante che dipende solo da  $X_0, T$  e dalla costante  $K$  della crescita lineare di  $\mu$ .*

**Proposizione 2** (Consistenza). Considero l'operatore  $L_t$  definito in (1.1) e  $L_t^\delta$  di (1.5). Sia  $Y$  una funzione Lipschitziana su  $[0, T]$ , con costante di Lipschitz  $K_1$ .

Per ogni  $t \in [0, T]$  vale:

$$|L_t Y - L_t^\delta Y| \leq C\delta,$$

con  $C$  costante che dipende solo da  $T, K_1$  e dalla costante  $K$  della crescita lineare di  $\mu$ .

**Proposizione 3** (Stabilità). Sia  $L_t^\delta$  l'operatore discretizzato in (1.5).

Siano  $X, Y$  funzioni continue su  $[0, T]$ .

Allora vale:

$$\max_{t \in [0, T]} |X_t - Y_t| \leq e^{KT} \left( |X_0 - Y_0| + \max_{t \in [0, T]} |L_t^\delta X - L_t^\delta Y| \right).$$

Per analizzare l'ordine di convergenza della discretizzazione di Eulero, consideriamo il seguente teorema.

**Teorema 1.** Sia  $X$  soluzione di  $L_t X = 0$  e  $X^\delta$  soluzione di  $L_t^\delta X = 0$ , aventi lo stesso dato iniziale  $X_0 = X_0^\delta$ . Allora esiste una costante  $C$  tale che:

$$\max_{t \in [0, T]} |X_t - X_t^\delta| \leq C\delta,$$

dove  $C$  dipende solo da  $T, X_0$  e dalla costante  $K$  della crescita lineare di  $\mu$ .

Da questo teorema segue che lo schema di Eulero per le equazioni differenziali ha ordine di convergenza pari a 1.

Per aumentare l'ordine di convergenza è possibile sviluppare lo schema di Eulero per un qualsiasi ordine  $p \in \mathbb{N}$ , attraverso lo sviluppo di Taylor di  $X$ .

Se l'applicazione  $\mu$  è sufficientemente regolare, deriviamo l'equazione (1.1) e, definendo  $\mu_x = \partial_x \mu(x, t)$  e  $\mu_t = \partial_t \mu(x, t)$ , si ottiene:

$$X' = \mu;$$

$$X'' = \mu_t + \mu_x X' = \mu_t + \mu_x \mu;$$

$$X''' = \mu_{tt} + 2\mu_{tx} X' + \mu_{tt} (X')^2 + \mu_x X'' = \mu_{tt} + 2\mu_{tx} \mu + \mu_{xx} \mu^2 + \mu_x \mu_t + \mu_x^2 \mu.$$

Si consideri ora lo sviluppo di Taylor di ordine  $p \in \mathbb{N}$ :

$$X_{t_{n+1}} = X_{t_n} + X'_{t_n} \delta + \dots + \frac{1}{p!} X_{t_n}^{(p)} \delta^p.$$

Sostituendo nello sviluppo di Taylor le derivate calcolate precedentemente, si ottiene lo schema di Eulero di ordine  $p$ .

Per esempio lo schema di Eulero di ordine 3 è dato dal seguente sviluppo di Taylor di  $X$ :

$$X_{t_{n+1}} = X_{t_n} + X'_{t_n} \delta + \frac{\delta^2}{2} X''_{t_n} + \frac{\delta^3}{6} X'''_{t_n},$$

da cui, per sostituzione, deriva la seguente formulazione ricorsiva:

$$\begin{aligned} X_{t_{n+1}} = & X_{t_n} + \mu_t(t_n, X_{t_n})\delta + \frac{\delta^2}{2}(\mu_{tt}(t_n, X_{t_n}) + \mu_x(t_n, X_{t_n})\mu(t_n, X_{t_n})) + \\ & + \frac{\delta^3}{6}(\mu_{ttt}(t_n, X_{t_n}) + 2\mu_{tx}(t_n, X_{t_n})\mu(t_n, X_{t_n}) + \mu_{xx}(t_n, X_{t_n})\mu^2(t_n, X_{t_n})) + \\ & + \mu_x(t_n, X_{t_n})\mu_t(t_n, X_{t_n}) + \mu_x^2(t_n, X_{t_n})\mu(t_n, X_{t_n}). \end{aligned}$$

## 1.2 Metodo di Eulero per equazioni stocastiche

Analizziamo il metodo di Eulero per approssimare numericamente la soluzione di un'equazione differenziale stocastica.

Consideriamo un moto Browniano,  $W$ , nello spazio di probabilità con filtrazione  $(\Omega, \mathcal{F}, P, \mathcal{F}_t)$  e consideriamo le funzioni

$$\begin{aligned} \mu &= \mu(t, x) : [0, T] \times \mathbb{R} \longrightarrow \mathbb{R} \\ \sigma &= \sigma(t, x) : [0, T] \times \mathbb{R} \longrightarrow \mathbb{R} \end{aligned}$$

tali che siano Lipschitziane globalmente nella variabile  $x$ , Hölderiane di esponente  $1/2$  nella variabile  $t$  e che abbiano crescita lineare. Possiamo riassumere queste caratteristiche nella seguente condizione:

$$|\mu(t, x) - \mu(s, y)|^2 + |\sigma(t, x) - \sigma(s, y)|^2 \leq K(|t-s| + |x-y|^2), \quad x, y \in \mathbb{R}, s, t \in [0, T]. \quad (1.6)$$

Definiamo quindi l'operatore  $L_t$ :

$$L_t X = X_t - X_0 - \int_0^t \mu(s, X_s) ds - \int_0^t \sigma(s, X_s) dW_s, \quad t \in [0, T], \quad (1.7)$$

e l'equazione differenziale stocastica:

$$L_t X = 0, \quad t \in [0, T]. \quad (1.8)$$

Analogamente a prima, lo schema di Eulero suddivide l'intervallo  $[0, T]$  in  $N$  intervallini  $[t_n, t_{n+1}]$  di lunghezza  $\delta = \frac{T}{N}$ . Rendendo costante a tratti gli integrandi di  $L_t$ , si ottiene l'operatore discretizzato  $L_t^\delta$ , con  $t \in [0, T]$ :

$$L_t^\delta X = X_t - X_0 - \int_0^T \sum_{n=1}^N \mu(t_{n-1}, X_{t_{n-1}}) \mathbb{1}_{]t_{n-1}, t_n]}(s) ds - \int_0^T \sum_{n=1}^N \sigma(t_{n-1}, X_{t_{n-1}}) \mathbb{1}_{]t_{n-1}, t_n]}(s) dW_s. \quad (1.9)$$

Il processo discretizzato è definito ricorsivamente da:

$$X_{t_{n+1}}^\delta = X_{t_n}^\delta + \mu(t_n, X_{t_n}^\delta)\delta + \sigma(t_n, X_{t_n}^\delta)(W_{t_{n+1}} - W_{t_n}). \quad (1.10)$$

Per dimostrare la convergenza del metodo di Eulero per le equazioni stocastiche, analizziamo la regolarità, la consistenza e la stabilità attraverso tre teoremi.

**Proposizione 4** (Regolarità). *Sia  $X$  soluzione dell'equazione  $L_t X = 0$ . Vale:*

$$E \left[ \sup_{s \in [t, t']} |X_s - X_t|^2 \right] \leq K_1(t' - t), \quad 0 \leq t < t' \leq T$$

con  $K_1$  costante che dipende solo da  $T, E[X_0^2]$  e dalla costante  $K$  di (1.6).

Denotiamo con  $\mathcal{A}_C$  lo spazio dei processi  $(X_t)_{t \in [0, T]}$  continui,  $\mathcal{F}_t$ -adattati, con norma:

$$\|X\|_T = \sqrt{E \left[ \sup_{t \in [0, T]} X_t^2 \right]} < \infty.$$

**Proposizione 5** (Consistenza). *Sia  $Y \in \mathcal{A}_C$  tale che*

$$E \left[ \sup_{s \in [t, t']} |Y_s - Y_t|^2 \right] \leq K_1(t' - t), \quad 0 \leq t < t' \leq T$$

allora

$$\|LY - L^\delta Y\|_T \leq C\sqrt{\delta}$$

con  $C$  costante che dipende da  $T, K_1$  e dalla costante  $K$  di (1.6).

**Proposizione 6** (Stabilità). *Per ogni  $X, Y \in \mathcal{A}_C$ , esiste una costante  $C_0$ , tale che:*

$$\|X - Y\|_T^2 \leq C_0(E[|X_0 - Y_0|^2] + \|L^\delta X - L^\delta Y\|_T^2).$$

La costante  $C_0$  dipende da  $T$  e dalla costante  $K$  di (1.6).

Lo schema di Eulero analizzato per le equazioni stocastiche ha ordine di convergenza forte pari a  $1/2$ , per via del seguente teorema.

**Teorema 2.** *Sia  $X$  soluzione di  $L_t X = 0$  e  $X^\delta$  soluzione di  $L_t^\delta X = 0$ , aventi lo stesso valore iniziale  $X_0 = X_0^\delta$ .*

*Allora esiste una costante  $C$  tale che:*

$$\|X - X^\delta\|_T \leq C\sqrt{\delta}.$$

La costante  $C$  dipende solo da  $T, E[X_0^2]$  e dalla costante  $K$  di (1.6).

Per aumentare l'ordine di convergenza del metodo di Eulero, è possibile applicare lo schema di Milstein che presenta ordine di convergenza forte pari a 1.

Lo schema di Milstein si basa sull'approssimazione del primo ordine del termine diffusivo rispetto alla variabile  $x$ :

$$\int_{t_n}^{t_{n+1}} \sigma(t, X_t) dW_t \sim \int_{t_n}^{t_{n+1}} (\sigma(t_n, X_{t_n}) + \partial_x \sigma(t_n, X_{t_n})(W_t - W_{t_n})) dW_t,$$

da cui si ottiene l'iterazione:

$$X_{t_{n+1}} = X_{t_n} + \mu(t_n, X_{t_n})(t_{n+1} - t_n) + \sigma(t_n, X_{t_n})\sqrt{(t_{n+1} - t_n)}Z + \partial_x \sigma(t_n, X_{t_n}) \frac{(t_{n+1} - t_n)(Z^2 - 1)}{2}$$

con  $Z$  variabile aleatoria normale standard.

Vediamo ora degli esempi per simulare le traiettorie di alcuni moti.

### 1.3 Moto Browniano

Riprendiamo la definizione di moto Browniano.

**Definizione 1** (Moto Browniano). *Dato  $(\Omega, \mathcal{F}, P, (\mathcal{F}_t)_{t \geq 0})$  uno spazio di probabilità con filtrazione, un moto Browniano standard  $(W_t)_{t \geq 0}$  è un qualunque processo stocastico tale che:*

1. *il valore iniziale è  $W_0 = 0$ ;*
2. *il processo  $W$  è continuo e  $\mathcal{F}_t$ -adattato;*
3. *gli incrementi hanno distribuzione normale  $W_t - W_s \sim N_{0, t-s}$  e sono indipendenti dalla filtrazione  $\mathcal{F}_t$ .*

Osserviamo che, come conseguenza delle richieste (1) e (3), il moto Browniano al tempo  $t$  ha distribuzione normale standard:

$$W(t) \sim N(0, t). \quad (1.11)$$

Attraverso il metodo di Eulero, una traiettoria per un moto Browniano standard può essere generata attraverso la seguente iterazione:

$$W(t_{i+1}) = W(t_i) + \sqrt{t_{i+1} - t_i} Z_{i+1}, \quad i = 0, \dots, n-1 \quad (1.12)$$

dove  $Z_1, \dots, Z_n$  sono variabili aleatorie indipendenti con distribuzione normale standard.

Più in generale è possibile definire un moto Browniano  $X(t)$  con drift  $\mu$  e coefficiente di diffusione  $\sigma^2$ .

**Definizione 2.** Date le costanti  $\mu$  e  $\sigma > 0$ , il processo  $X(t)$  è un moto Browniano con drift  $\mu$  e coefficiente di diffusione  $\sigma^2$ ,  $X \sim MB(\mu, \sigma^2)$ , se:

$$\frac{X(t) - \mu t}{\sigma}$$

è un moto Browniano standard.

Dalla Definizione (2), possiamo costruire  $X \sim MB(\mu, \sigma^2)$  dato un mB standard  $W$ , ponendo:

$$X(t) = \mu t + \sigma W(t).$$

Osservando la costruzione del moto Browniano con drift  $\mu$  e coefficiente di diffusione  $\sigma^2$  e considerando la proprietà (1.11) del moto Browniano standard, segue:

$$X(t) \sim N(\mu t, \sigma^2 t). \quad (1.13)$$

Inoltre,  $X$  risolve l'equazione differenziale stocastica:

$$dX(t) = \mu dt + \sigma dW(t).$$

Una traiettoria per un moto  $X \sim MB(\mu, \sigma^2)$ , con valore iniziale  $X(0)$  dato, può essere generata attraverso la seguente iterazione:

$$X(t_{i+1}) = X(t_i) + \mu(t_{i+1} - t_i) + \sigma \sqrt{t_{i+1} - t_i} Z_{i+1}, \quad i = 0, \dots, n-1 \quad (1.14)$$

dove  $Z_1, \dots, Z_n$  sono variabili aleatorie indipendenti con distribuzione normale standard.

In generale è possibile considerare  $\mu$  e  $\sigma$  come applicazioni deterministiche positive in funzione del tempo,  $\mu(t)$  e  $\sigma(t) > 0$ . In tal caso il moto Browniano  $X$  può essere definito attraverso l'equazione differenziale stocastica:

$$dX(t) = \mu(t) dt + \sigma(t) dW(t),$$

che possiamo riscrivere in forma integrale:

$$X(t) = X(0) + \int_0^t \mu(s) ds + \int_0^t \sigma(s) dW(s),$$

con  $X(0)$  una costante arbitraria.

Il processo  $X$  è ancora continuo e adattato, gli incrementi,  $X(t) - X(s)$ , sono indipendenti tra loro e hanno distribuzione normale con media e varianza rispettivamente:

$$E[X(t) - X(s)] = \int_s^t \mu(u) du, \quad \text{Var}[X(t) - X(s)] = \int_s^t \sigma^2(u) du.$$

Si può ottenere una traiettoria per tale moto Browniano con coefficienti dipendenti dal tempo e valore iniziale  $X(0)$  dato, attraverso la seguente iterazione:

$$X(t_{i+1}) = X(t_i) + \int_{t_i}^{t_{i+1}} \mu(s) ds + \sqrt{\int_{t_i}^{t_{i+1}} \sigma^2(u) du} Z_{i+1}, \quad i = 0, \dots, n-1 \quad (1.15)$$

Riscrivendo l'espressione (1.15) attraverso l'approssimazione di Eulero, si ottiene più semplicemente:

$$X(t_{i+1}) = X(t_i) + \mu(t_i)(t_{i+1} - t_i) + \sigma(t_i)\sqrt{t_{i+1} - t_i}Z_{i+1}, \quad i = 0, \dots, n-1 \quad (1.16)$$

con  $X(0)$  valore iniziale dato.

I metodi (1.12), (1.14) e (1.15) sono esatti, nel senso che la distribuzione congiunta dei valori simulati,  $(W(t_1), \dots, W(t_n))$  oppure  $(X(t_1), \dots, X(t_n))$ , coincide con la distribuzione congiunta del corrispondente moto Browniano in  $t_1, \dots, t_n$ . Tali metodi non forniscono informazioni su cosa succede tra i tempi fissati  $t_i$ , ma è possibile estendere i valori simulati agli altri tempi intermedi attraverso una interpolazione lineare a tratti introducendo, tuttavia, degli errori di discretizzazione. Diversamente, l'iterazione (1.16) presenta degli errori di discretizzazione anche sui tempi fissati  $t_i, i = 0, \dots, n$ .

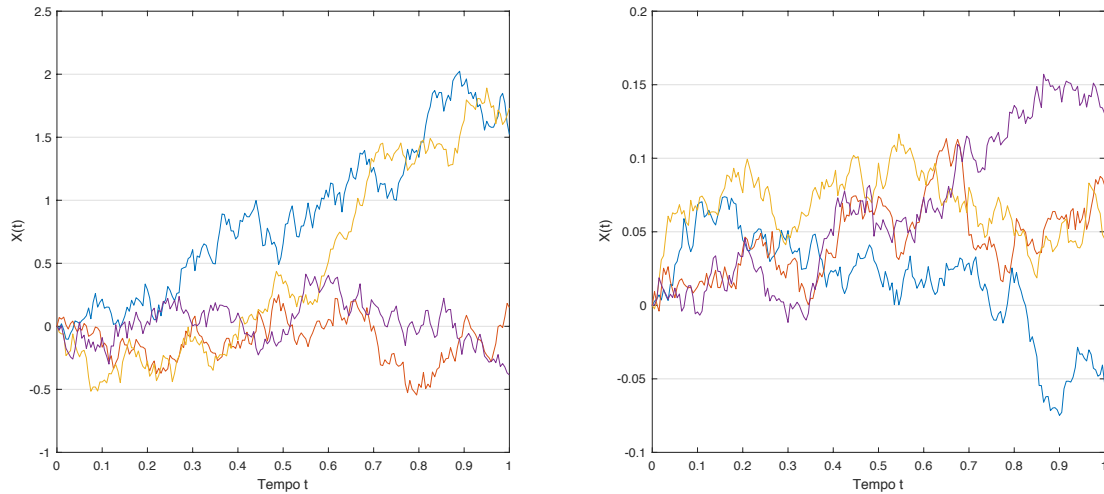


Figura 1.1: A sinistra: moto Browniano standard.  
A destra: moto Browniano con  $\mu = 0.06$ ,  $\sigma = 0.1$ .

La Figura 1.1 mostra, a sinistra, 4 traiettorie per un moto Browniano standard, mentre a destra è rappresentato un moto Browniano con drift  $\mu = 0.06$  e coefficiente di diffusione  $\sigma = 0.1$ . Entrambi i moti hanno valore iniziale nell'origine e sono definiti nell'intervallo  $[0, 1]$  discretizzato con  $n = 200$  punti.



## 1.4 Moto Browniano geometrico

**Definizione 3** (Moto Browniano Geometrico). *Il moto Browniano geometrico è il processo  $\{S(t)\}$  che è soluzione dell'equazione differenziale stocastica:*

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t), \quad (1.17)$$

con  $\mu$  e  $\sigma > 0$  costanti e  $(W(t))_{t \geq 0}$  moto Browniano standard.

Tale processo viene indicato con  $S \sim MBG(\mu, \sigma^2)$ . La costante  $\mu$  rappresenta il drift mentre  $\sigma$  è il parametro di volatilità del moto Browniano geometrico.

**Proposizione 7.** *Il moto Browniano geometrico  $S$ , soluzione dell'equazione (1.17), è della forma:*

$$S(t) = S(0)e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma W(t)} \quad (1.18)$$

*Dimostrazione.* Per prima cosa consideriamo la formula di Ito.

Per  $F = F(t, x) \in C^2(\mathbb{R}^2)$ , si ha che  $F(t, W_t)$  è un processo di Ito e vale la formula:

$$dF(t, W_t) = (\partial_t F)(t, W_t)dt + (\partial_x F)(t, W_t)dW_t + \frac{1}{2}(\partial_{xx} F)(t, W_t)dt.$$

Applichiamo tale formula alla soluzione (1.18) attraverso la funzione:

$$F(t, x) = S(0)e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma x}.$$

Calcoliamo le derivate parziali

$$\partial_t F(t, x) = (\mu - \frac{1}{2}\sigma^2)F(t, x); \quad \partial_x F(t, x) = \sigma F(t, x); \quad \partial_{xx}^2 F(t, x) = \sigma^2 F(t, x);$$

e le sostituiamo nella formula di Ito, ottenendo:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t).$$

□

Poiché, come visto nella Definizione (1), gli incrementi di  $W$  sono indipendenti e con distribuzione normale, allora per simulare una traiettoria di un moto Browniano geometrico  $S$ , con valore iniziale  $S(0)$ , si può utilizzare la seguente procedura ricorsiva:

$$S(t_{i+1}) = S(t_i)e^{(\mu - \frac{1}{2}\sigma^2)(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1}}, \quad i = 0, \dots, n-1 \quad (1.19)$$

con  $Z_1, \dots, Z_n$  variabili aleatorie indipendenti con distribuzione normale standard.

Si può osservare che questa procedura si ricava dall'iterazione (1.14) di  $X \sim MB(\mu, \sigma^2)$

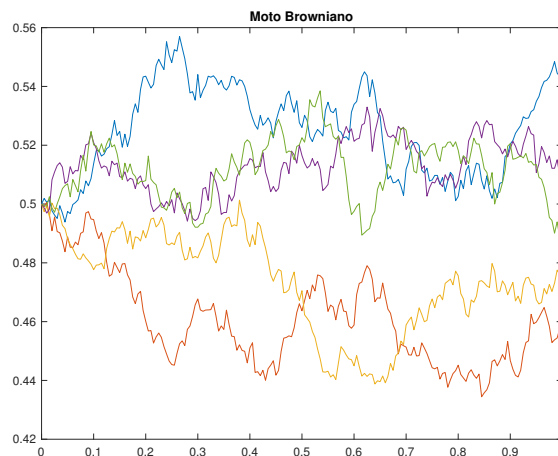


Figura 1.2: Moto Browniano geometrico con:  $S_0 = 0.5$ ,  $\mu = 0.06$ ,  $\sigma = 0.1$  e  $T = 1$ .

elevando entrambi i membri e sostituendo il drift  $\mu$  con  $\mu - \frac{1}{2}\sigma^2$ . Quindi il metodo di simulazione sopra descritto è esatto, ovvero la distribuzione congiunta dei valori simulati  $(S(t_1), \dots, S(t_n))$  coincide con la distribuzione congiunta del corrispondente moto Browniano geometrico  $S$  in  $t_1, \dots, t_n$ .

La Figura 3.1 mostra 5 traiettorie per un moto Browniano geometrico con  $\mu = 0.06$ ,  $\sigma = 0.1$  e valore iniziale  $S(0) = 0.5$ . L'intervallo di definizione  $[0, 1]$  è discretizzato in  $n = 200$  punti. Si può osservare che i valori assunti dalle traiettorie del moto Browniano geometrico sono sempre positivi.

#### 1.4.1 Algoritmi per il moto Browniano geometrico

Costruiamo l'algoritmo per simulare alcune traiettorie del moto Browniano geometrico utilizzando il linguaggio di programmazione Matlab.

Per esempio consideriamo il moto Browniano geometrico con valore iniziale  $S_0 = 36$ , drift  $\mu = 0.06$  e volatilità  $\sigma = 0.2$ . Per utilizzare la tecnica di Eulero discretizziamo l'intervallo di definizione  $[0, 1]$  in  $n = 200$  sottointervalli di lunghezza  $dt$ . Confrontiamo diverse tecniche di calcolo per simulare in modo ottimale  $b = 100000$  traiettorie del moto appena descritto.

Un primo approccio calcola, passo per passo, una traiettoria alla volta. Si simulano i valori della singola traiettoria ad ogni tempo, per poi passare alla traiettoria successiva solo al completamento di questa. Tale metodo prevede l'utilizzo di due cicli *for*: il primo per le traiettorie e, all'interno, un secondo ciclo per il tempo discretizzato.

---

```

function [S]=mBgeom1(b,n,dt,S0,mu,sigma)
S=zeros(n+1,b);
S(1,:)=S0;
for j=1:b
    for i=2:n+1
        Z=randn();
        S(i,j)=S(i-1,j)*exp((mu-0.5*sigma^2)*dt+sigma*sqrt(dt)*Z);
    end
end
end

```

---

Tuttavia proprio l'uso di due cicli *for* comporta un onere in termini di tempo di calcolo, all'aumentare delle discretizzazioni del tempo e del numero delle traiettorie.

Un secondo approccio consiste nel calcolo parallelo delle stime di tutte le traiettorie al variare del tempo. Applicando un solo ciclo *for* sul tempo, l'algoritmo calcola simultaneamente i valori di tutte le traiettorie nel tempo fissato.

---

```

function [S]=mBgeom2(b,n,dt,S0,mu,sigma)
S=zeros(n+1,b);
S(1,:)=S0;
for i=2:n+1
    Z=randn(1,b);
    S(i,:)=S(i-1,:).*exp((mu-0.5*sigma^2)*dt+sigma*sqrt(dt)*Z);
end
end

```

---

L'utilizzo di un unico ciclo *for* permette di migliorare la prestazione del calcolatore, in termini di tempo di calcolo.

Il terzo metodo che analizziamo non prevede l'uso di cicli *for* ma sfrutta il *cumulative product* di Matlab. Osservando la procedura ricorsiva di Eulero (1.19), costruiamo una matrice  $(n + 1) \times b$  che contiene:

- nella prima riga, i valori del dato iniziale  $S_0$  per ogni traiettoria;
- nelle righe successive, i valori del fattore esponenziale di (1.19) per ogni traiettoria, ovvero:

$$e^{(\mu - \frac{1}{2}\sigma^2)(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1}}, \quad i = 0, \dots, n - 1,$$

con  $Z_1, \dots, Z_n$  variabili aleatorie indipendenti con distribuzione normale standard.

Utilizzando il *cumulative product* per colonne, si ottiene una matrice nella quale ogni elemento è dato dal prodotto di sé stesso con gli elementi che lo precedono nella sua stessa colonna.

Applicando il *cumulative product* alla matrice descritta sopra, si ricrea la procedura (1.19) simulando così tutte le traiettorie nell'intervallo di definizione.

---

```
function[S]=mBgeom3(b,n,dt,S0,mu,sigma)
Z=randn(n,b);
e = exp((mu-sigma^2/2)*dt+sigma*sqrt(dt)*Z );
S = cumprod([S0*ones(1,b); e]);
end
```

---

Avendo eliminato un altro ciclo *for*, quest'ultimo approccio risulta leggermente più veloce del precedente.

Confrontiamo gli algoritmi in funzione del tempo di esecuzione.

Per fare ciò, consideriamo il moto Browniano geometrico con i parametri utilizzati precedentemente:  $S(0) = 0.5$ ,  $\mu = 0.06$ ,  $\sigma = 0.2$ . Variamo la discretizzazione dell'intervallo di definizione: partiamo da 10 punti di discretizzazione fino al arrivare a 200, incrementando di 5 punti per volta.

Si ottiene il seguente grafico, nel quale il tempo di esecuzione è espresso in secondi.

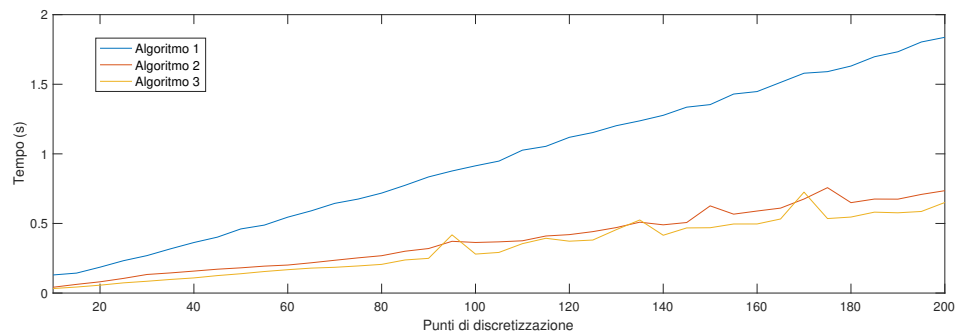


Figura 1.3: Tempo di esecuzione dei 3 diversi algoritmi visti per simulare un moto Browniano geometrico, in funzione del numero degli intervalli di discretizzazione.

Il grafico 1.3, conferma le nostre ipotesi sull'efficienza in termini di tempo di esecuzione dei tre algoritmi analizzati.

Nei prossimi capitoli avremo ancora bisogno di simulare il moto Browniano geometrico. Perciò utilizzeremo il terzo algoritmo visto, che applica il *cumulative product*, in quanto, alla luce delle considerazioni appena fatte, risulta computazionalmente più efficiente.

### 1.4.2 Il moto Browniano geometrico in finanza

Il moto Browniano geometrico è un processo utilizzato in finanza per modellizzare il prezzo di un'opzione, grazie agli studi e al lavoro di Paul Samuelson svolti intorno al 1960. Il suo uso finanziario è dovuto a due importanti caratteristiche di questo moto. Per prima cosa, mentre il moto Browniano può assumere valori negativi, il moto Browniano geometrico è sempre positivo (perché la funzione esponenziale assume sempre valori positivi) e questo è indispensabile per modellizzare il prezzo di un'opzione. Una seconda peculiarità è data dall'indipendenza delle variazioni percentuali:

$$\frac{S(t_2) - S(t_1)}{S(t_1)}, \frac{S(t_3) - S(t_2)}{S(t_2)}, \dots, \frac{S(t_n) - S(t_{n-1})}{S(t_{n-1})}, \quad t_1 < t_2 < \dots < t_n.$$

Diversamente dal moto Browniano standard dove sono indipendenti gli incrementi:

$$S(t_2) - S(t_1), S(t_3) - S(t_2), \dots, S(t_n) - S(t_{n-1}), \quad t_1 < t_2 < \dots < t_n.$$

Consideriamo un'opzione con tasso d'interesse  $r$  continuo costante. Sia  $S$  il sottostante dell'opzione modellizzabile da un moto Browniano geometrico, allora  $S$  soddisferà l'equazione differenziale stocastica:

$$dS(t) = rS(t)dt + \sigma dW(t).$$

Si osserva che il drift di  $S$  corrisponde al tasso di interesse  $r$ .

Più in generale possiamo considerare un tasso di interesse deterministico ma dipendente dal tempo,  $r(t)$ . L'andamento di  $S(t)$  sarà descritto dall'equazione differenziale stocastica:

$$dS(t) = r(t)S(t)dt + \sigma S(t)dW(t),$$

con soluzione

$$S(t) = S(0)e^{\int_0^t r(u) du - \frac{1}{2}\sigma^2 t + \sigma W(t)}.$$

Nei prossimi paragrafi, 1.5 e 1.6, analizzeremo alcuni modelli per i tassi di interesse con andamento stocastico.

## 1.5 Modelli Gaussiani

Analizziamo alcuni modelli per i tassi di interesse con andamento stocastico, in particolare la dinamica del tasso a breve termine.

Il modello classico di Vasicek descrive l'andamento del tasso a breve termine attraverso un processo del tipo:

$$dr(t) = \alpha(b - r(t))dt + \sigma dW(t), \quad (1.20)$$

dove  $W$  è un moto Browniano standard, mentre  $\alpha, b, \sigma$  sono costanti positive.

Osservando l'equazione si nota che il drift  $\alpha(b - r(t))$  è positivo per  $r(t) < b$ , viceversa è negativo per  $r(t) > b$ . Inoltre  $r(t)$  tende a  $b$  e questa proprietà è generalmente indicata come "reversione media". Possiamo quindi considerare  $b$  come il tasso di interesse a lungo termine, mentre  $\alpha$  indica la velocità con la quale  $r(t)$  tende a  $b$ .

Il modello di Vasicek può essere espresso anche con il tasso  $b$  in funzione del tempo. In tal caso la soluzione generale è data da:

$$r(t) = e^{-\alpha t}r(0) + \alpha \int_0^t e^{-\alpha(t-s)}b(s) ds + \sigma \int_0^t e^{-\alpha(t-s)} dW(s).$$

Per ogni tempo  $t$ , la variabile  $r(t)$  ha distribuzione normale con media:

$$e^{-\alpha(t-u)}r(u) + \mu(u, t), \quad \text{dove } \mu(u, t) \equiv \alpha \int_u^t e^{-\alpha(t-s)}b(s) ds,$$

e varianza

$$\sigma_r^2(u, t) \equiv \sigma^2 \int_u^t e^{-2\alpha(t-s)} ds = \frac{\sigma^2}{2\alpha} \left(1 - e^{-2\alpha(t-u)}\right).$$

Per simulare i valori di  $r(t)$  ai tempi  $0 = t_0 < t_1 < \dots < t_n$  ci sono due metodi.

Il primo algoritmo che analizziamo utilizza i valori di media e varianza di  $r(t)$  visti sopra, ottenendo la seguente iterazione:

$$r(t_{i+1}) = e^{-\alpha(t_{i+1}-t_i)}r(t_i) + \mu(t_i, t_{i+1}) + \sigma_r(t_i, t_{i+1})Z_{i+1} \quad (1.21)$$

con  $Z_1, \dots, Z_n$  variabili aleatorie indipendenti con distribuzione normale standard.

Questo algoritmo è una simulazione esatta, quindi la distribuzione congiunta di  $(r(t_1), \dots, r(t_n))$  coincide con il processo di Vasicek al tempo  $t_1, \dots, t_n$  con lo stesso valore iniziale  $r(0)$ .

Un altro metodo è dato dallo schema di Eulero:

$$r(t_{i+1}) = r(t_i) + \alpha(b(t_i) - r(t_i))(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1} \quad (1.22)$$

dove  $Z_1, \dots, Z_n$  sono variabili aleatorie indipendenti con distribuzione normale standard. Quest'ultimo metodo è più semplice e utilizza solo i parametri presenti nell'equazione differenziale stocastica (1.20), tuttavia introduce degli errori di discretizzazione.

Nel caso particolare di  $b(t) = b$  costante, l'iterazione (1.21) si semplifica:

$$r(t_{i+1}) = e^{-\alpha(t_{i+1}-t_i)}r(t_i) + b(1 - e^{-\alpha(t_{i+1}-t_i)}) + \sigma\sqrt{\frac{1}{2\alpha}(1 - e^{-2\alpha(t_{i+1}-t_i)})}Z_{i+1}. \quad (1.23)$$

Lo schema di Eulero rimane quello visto in (1.22), è sufficiente sostituire  $b(t_i)$  con  $b$ .

**Osservazione 1.** Se  $b(t) = b$  costante, allora l'iterazione di Eulero si ricava dalla (1.23) approssimando gli esponenziali:  $e^x \approx 1 + x$ .

*Dimostrazione.* Consideriamo l'iterazione (1.23) e approssimiamo gli esponenziali, ottenendo:

$$r(t_{i+1}) \approx (1 - \alpha(t_{i+1} - t_i))r(t_i) + b(1 - 1 + \alpha(t_{i+1} - t_i)) + \sigma\sqrt{\frac{1}{2\alpha}(1 - 1 + 2\alpha(t_{i+1} - t_i))}Z_{i+1}.$$

Semplificando e riordinando i fattori, si ricava lo schema di Eulero:

$$r(t_{i+1}) = r(t_i) + \alpha(b - r(t_i))(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1}.$$

□

## 1.6 Square-root diffusions

Nel 1985 Cox, Ingersoll e Ross introdussero il modello CIR come estensione del modello Vasicek. Questo modello descrive l'evoluzione dei tassi di interesse, in particolare dei tassi a breve termine.

Nel modello CIR il tasso di interesse  $r(t)$  soddisfa l'equazione differenziale stocastica:

$$dr(t) = \alpha(b - r(t))dt + \sigma\sqrt{r(t)}dW(t), \quad (1.24)$$

con  $W$  moto Browniano standard,  $\sigma$  costante che rappresenta la volatilità,  $\alpha$  e  $b$  costanti positive che indicano rispettivamente la velocità di regolazione e la media.

Il drift del modello CIR è uguale a quello di Vasicek,  $\alpha(b - r(t))$ , quindi, anche in questo caso si ha che  $r(t)$  tende verso  $b$  con velocità controllata da  $\alpha$ . Diversamente da Vasicek, nel modello CIR il termine di diffusione  $\sigma\sqrt{r(t)}$  diventa molto piccolo quando il tasso  $r(t)$  è vicino a zero. Questo impedisce che  $r(t)$  assuma valori negativi.

La discretizzazione di Eulero dell'equazione (1.24), simula  $r(t)$  ai tempi  $t_1, \dots, t_n$  ponendo:

$$r(t_{i+1}) = r(t_i) + \alpha(b - r(t_i))(t_{i+1} - t_i) + \sigma\sqrt{r(t_i)^+}\sqrt{t_{i+1} - t_i}Z_{i+1},$$

con  $Z_1, \dots, Z_n$  variabili aleatorie indipendenti con distribuzione normale standard.

Da notare che dentro la prima radice quadrata si considera la parte positiva di  $r(t_i)$ , questo perché il valore del tasso  $r(t_i)$ , ottenuto dalla discretizzazione di Eulero, potrebbe risultare negativo.





## Capitolo 2

# Il Metodo Monte Carlo

### 2.1 Un po' di storia

Il metodo Monte Carlo è una classe di metodi basati sul campionamento aleatorio al fine di risolvere numericamente un problema la cui soluzione, per via analitica, risulta troppo complessa.

Per indicare questa tecnica che si fonda sul calcolo aleatorio, il matematico N. Metropolis utilizzò il nome “Monte Carlo” facendo riferimento al famoso casinò, sede per antonomasia dell'aleatorietà, in quanto il calcolo delle probabilità nacque proprio da problemi sul lancio dei dadi e sul gioco d'azzardo.

Tuttavia la prima applicazione di un metodo di questo tipo si ha già nel 1777 da parte del matematico francese Georges Louis Leclerc, conte di Buffon, nell'eseguire un esperimento con lo scopo di stimare il valore di  $\pi$  greco. Come descritto dallo stesso Buffon nella sua opera *Essai di arithmetique morale*, l'esperimento consisteva nel far cadere in modo casuale uno spillo su una struttura a strisce parallele della stessa larghezza così, attraverso la ripetizione dell'esperimento e l'applicazione della probabilità e della geometria integrale, è stato possibile stimare statisticamente il numero  $\pi$  greco.

Ma è nel '900 che questi metodi basati sul campionamento casuale trovano piena applicazione. Negli anni '30 Enrico Fermi utilizzava stime ottenute con tecniche di campionamento statistico per lo studio del moto dei neutroni. Negli anni '40 J.Von Neumann e S. Ulam, partecipando al progetto Manhattan per lo studio della dinamica delle esplosioni nucleari, formalizzarono il metodo Monte Carlo. La sua formalizzazione, insieme allo sviluppo dei calcolatori elettronici, permise un ulteriore incremento del metodo che trovò così applicazione in nuovi ambiti. Nel 1964 Monte Carlo approda per la prima volta in economia grazie a David Hertz che, nell'articolo *Risk analysis in capital investment*, propose tale tecnica per la valutazione degli investimenti.

Grazie al fatto che il metodo Monte Carlo riesce a trarre stime attraverso le simulazioni, la sua applicazione continua tuttora ad avere importanza in svariati ambiti, tanto che nel

2000 J.Dongarra e F. Sullivan, in un articolo per la rivista *Computing in science and engineering*, lo inseriscono tra i dieci algoritmi con la più grande influenza sullo sviluppo e sulla pratica della scienza e dell'ingegneria nel XX secolo.

## 2.2 La tecnica Monte Carlo

Il metodo Monte Carlo è una tecnica di approssimazione numerica che permette di stimare il valore atteso di una variabile aleatoria  $X$ .

Precisamente, considero  $X$  una variabile aleatoria e  $f$  una funzione. Per calcolare il valore atteso di  $f(X)$ ,  $E[f(X)]$ :

- si generano  $n$  variabili  $X_1, \dots, X_n$  indipendenti e identicamente distribuite secondo la distribuzione di  $X$ ;
- si valuta la funzione nelle variabili generate;
- si esegue la media aritmetica dei valori calcolati.

Vale:

$$E[f(X)] = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(X_i). \quad (2.1)$$

Inoltre, il metodo Monte Carlo è spesso descritto come un metodo per stimare anche il valore di un integrale per una funzione su un intervallo unitario:

$$\alpha = \int_0^1 f(x) dx.$$

Questo è possibile perché l'integrale può essere rappresentato come un valore atteso,  $E[f(U)]$ , con  $U$  distribuito uniformemente in  $[0, 1]$ .

Così, per ottenere la stima Monte Carlo dell'integrale, è sufficiente generare  $n$  punti  $U_1, \dots, U_n$  indipendenti e distribuiti uniformemente in  $[0, 1]$ , valutare  $f$  in questi punti poi calcolare la media dei valori ottenuti:

$$\hat{\alpha}_n = \frac{1}{n} \sum_{i=1}^n f(U_i).$$

Assumendo  $f$  integrabile su  $[0, 1]$ , la convergenza del metodo Monte Carlo è data dalla legge forte dei grandi numeri.

**Teorema 3** (Legge dei grandi numeri). *Sia  $(X_n)$  una successione di variabili aleatorie i.i.d. sommabili, posto:*

$$M_n = \frac{X_1 + \cdots + X_n}{n}, \quad e \quad \mu = E[X_1]$$

allora

$$\lim_{n \rightarrow \infty} M_n = \mu.$$

Quindi, nel caso dell'integrale considerato:  $\hat{\alpha}_n \rightarrow \alpha$  con probabilità 1, per  $n \rightarrow \infty$ .

## 2.3 Stima dell'errore

Consideriamo l'approssimazione numerica dell'integrale definito sul cubo unitario di  $\mathbb{R}^d$ :

$$\int_{[0,1]^d} f(x) dx.$$

In base a quanto detto nella sezione (2.2), utilizzando il metodo Monte Carlo l'approssimazione dell'integrale è data da:

$$\int_{[0,1]^d} f(x) dx = E[f(Y_1)] = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n f(Y_k),$$

con  $(Y_n)$  successione di variabili aleatorie i.i.d. con distribuzione uniforme su  $[0, 1]^d$ .

Stimo l'errore di approssimazione.

**Proposizione 8.** *Sia  $(X_n)$  una successione di variabili aleatorie reali indipendenti e identicamente distribuite, con  $\mu = E[X_1]$  e  $\sigma^2 = \text{var}(X_1)$  finiti.*

*Pongo:*

$$M_n = \frac{1}{n} \sum_{k=1}^n X_k.$$

Allora la stima dell'errore per il metodo Monte Carlo è data da:

$$P(|M_n - \mu| \leq \epsilon) \geq p, \quad \text{dove } p = 1 - \frac{\sigma^2}{n\epsilon^2}. \quad (2.2)$$

*Dimostrazione.* Per la disuguaglianza di Markov, per ogni  $\epsilon > 0$  vale:

$$P(|M_n - \mu| \geq \epsilon) \leq \frac{\text{var}(M_n)}{\epsilon^2}.$$

Essendo  $M_n$  somma di variabili aleatorie i.i.d. e per le proprietà della varianza, si ha:

$$\frac{\text{var}(M_n)}{\epsilon^2} = \frac{\text{var}(X_1)}{n\epsilon^2} = \frac{\sigma^2}{n\epsilon^2}.$$

Quindi:

$$P(|M_n - \mu| \geq \epsilon) \leq \frac{\sigma^2}{n\epsilon^2},$$

che si può riscrivere ottenendo la formulazione (2.2). □

La variabile  $n$  rappresenta quindi il numero di simulazioni,  $\epsilon$  il massimo errore di propagazione, mentre  $p$  è la probabilità minima per la quale la stima appartiene ad un intervallo di accettabilità,  $M_n \in [\mu - \epsilon, \mu + \epsilon]$ .

Analizzando il risultato della Proposizione (8), l'errore massimo di approssimazione, con  $n$  e  $p$  fissati, è dato da

$$\epsilon = \frac{\sigma}{\sqrt{n(1-p)}}. \quad (2.3)$$

Il metodo Monte Carlo ha quindi ordine di convergenza  $O(n^{-1/2})$ .

La caratteristica centrale per la tecnica Monte Carlo consiste proprio nell'andamento dell'errore:  $\frac{1}{\sqrt{n}}$ .

Tuttavia il metodo Monte Carlo non risulta competitivo per la stima di integrali a 1 dimensione, infatti possiamo trovare altri metodi deterministici con prestazioni superiori. Per esempio la discretizzazione in somme di Riemann ha ordine di convergenza  $O(n^{-q})$  se  $f \in C^q([0, 1])$ .

Ma è nella valutazione di integrali multidimensionali, definiti su  $[0, 1]^d$ , che il metodo Monte Carlo risulta particolarmente efficiente. L'ordine di convergenza rimane  $O(n^{-1/2})$  risultando indipendente dalla dimensione  $d$  del problema. Il fattore  $\sqrt{n}$  al denominatore fa sì che per dimezzare l'errore sia sufficiente aumentare il numero dei punti di un fattore 4, così come per migliorare la precisione della stima di un decimale si aumenta di un fattore 100 il numero dei punti. Diversamente, nei metodi di integrazione deterministici la convergenza dipende dalla dimensione  $d$  del problema. Per esempio, nella tecnica di discretizzazione in somme di Riemann per avere un errore dell'ordine di  $\frac{1}{n}$  è necessario valutare la funzione in  $n^d$  punti, così facendo il numero dei punti dipende in modo esponenziale dalla dimensione del problema.

Questo fa sì che il metodo Monte Carlo non sia un metodo competitivo nella stima di integrali con dimensioni basse,  $d \leq 5$ , perché in tal caso è possibile implementare algoritmi deterministici con prestazioni superiori, tuttavia viene preferito quando la dimensione del problema aumenta.

Osservando la relazione (2.3), si nota che la deviazione standard  $\sigma$ , è direttamente proporzionale all'errore di approssimazione. In generale la deviazione standard non è una variabile

nota, tuttavia è possibile stimare la varianza  $\sigma^2$  utilizzando le variabili generate  $X_n$ :

$$\sigma_n^2 = \frac{1}{n-1} \sum_{k=1}^n (X_k - \mu_n)^2, \quad \text{con} \quad \mu_n = \frac{1}{n} \sum_{k=1}^n X_k. \quad (2.4)$$

La deviazione standard influisce sulla stima dell'errore, quindi per migliorare l'efficienza di Monte Carlo si usano dei metodi di riduzione della varianza, tecniche che, sfruttando le caratteristiche del problema, riducono il valore di  $\sigma_n$  e aumentano la velocità di convergenza. Nel paragrafo 2.5 vedremo in dettaglio una particolare tecnica di riduzione della varianza: il metodo delle variabili antitetiche.

Ricordo il teorema del limite centrale.

**Teorema 4** (del limite centrale). *Sia  $(X_n)$  una successione di variabili aleatorie i.i.d. con  $\sigma^2 < \infty$ . Definisco la successione:*

$$G_n = \sqrt{n} \left( \frac{M_n - \mu}{\sigma} \right), \quad n \in \mathbb{N}; \quad \text{dove } M_n = \frac{X_1 + \dots + X_n}{n}, \quad \mu = E[X_1].$$

Allora

$$G_n \xrightarrow{d} Z, \quad \text{con } Z \sim N_{0,1},$$

ovvero, per ogni  $x \in \mathbb{R}$ , si ha

$$\lim_{n \rightarrow \infty} P(G_n \leq x) = \phi(x),$$

dove  $\phi$  è la funzione di distribuzione normale standard:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{y^2}{2}} dy.$$

Grazie al teorema del limite centrale, per il metodo Monte Carlo vale la convergenza in distribuzione:

$$\sqrt{n} \left( \frac{M_n - \mu}{\sigma} \right) \xrightarrow{n \rightarrow \infty} Z \sim N_{0,1}$$

nel senso che:

$$\lim_{n \rightarrow \infty} P \left( \sqrt{n} \left( \frac{M_n - \mu}{\sigma} \right) \leq x \right) = \phi(x),$$

con  $\phi$  funzione di distribuzione normale standard.

Tale limite si mantiene anche sostituendo il valore esatto di  $\sigma$  con i valori approssimati  $\sigma_n$  calcolati con (2.4), in quanto il rapporto tra la deviazione standard esatta e quella stimata tende a 1 al crescere di  $n$ . Questo è importante perché, come detto, il valore esatto di  $\sigma$

raramente è conosciuto.

Otteniamo una stima precisa dell'errore:

$$P\left(M_n \in \left[\mu - \frac{\sigma x}{\sqrt{n}}, \mu + \frac{\sigma x}{\sqrt{n}}\right]\right) \approx p, \quad \text{con } p = 2\phi(x) - 1.$$

Fissando la probabilità  $p \in ]0, 1[$ , la distanza tra il valore esatto e quello approssimato con  $n$  simulazioni è minore di:

$$\frac{\sigma}{\sqrt{n}} \phi^{-1}\left(\frac{p+1}{2}\right).$$

## 2.4 Esempi

Una prima applicazione del metodo Monte Carlo si può vedere nel calcolo del valore atteso del payoff di un'opzione call.

Sia  $S(t)$  il prezzo del sottostante al tempo  $t$  e  $K$  lo strike al tempo  $T$ . Se al tempo di scadenza  $T$  il prezzo dell'opzione supera il prezzo fissato  $K$  allora il profitto è dato dalla differenza tra i due valori, mentre se  $S(T)$  è minore di  $K$  allora l'opzione ha valore nullo. Il payoff dell'opzione al tempo  $T$  è quindi dato da:

$$(S(T) - K)^+ = \max\{0, S(T) - K\}.$$

Introduciamo un tasso di interesse continuo  $r$ , così il payoff è dato dal valore atteso dal prezzo scontato,  $E[e^{-rT}(S(T) - K)^+]$ . Suppongo che il sottostante  $S$  sia descritto attraverso un'equazione differenziale stocastica:

$$dS(t) = rS(t)dt + \sigma S(t)dW(t), \quad (2.5)$$

dove  $W$  è un moto Browniano standard,  $\sigma$  è la volatilità, e la soluzione è data da:

$$S(T) = S(0)e^{(r - \frac{1}{2}\sigma^2)T + \sigma W(T)}.$$

La variabile aleatoria  $W(T)$  segue una distribuzione normale con media 0 e varianza  $T$ , quindi, per semplificare i passaggi computazionali, è possibile sostituirla con  $\sqrt{T}Z$ , dove  $Z$  è una variabile aleatoria con distribuzione normale standard:

$$S(T) = S(0)e^{(r - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}Z}.$$

Il prezzo del titolo  $S(T)$  ha distribuzione lognormale, ovvero il logaritmo di  $S(T)$  ha distribuzione normale.

Per stimare il valore atteso del payoff di un'opzione call,  $E[e^{-rT}(S(T) - K)^+]$ , si applica un algoritmo che si basa sui seguenti passaggi:

1. Si genera una sequenza di  $n$  variabili aleatorie indipendenti con distribuzione normale standard  $Z_i, i = 1, \dots, n$ .

Quando si generano numeri pseudo-casuali è necessario porre attenzione alla qualità del generatore, in quanto le variabili pseudo-casuali hanno proprietà simili alle estrazioni casuali ma, all'aumentare del numero delle simulazioni, si nota che sono generati in modo non realmente indipendente e questo introduce un errore aggiuntivo.

2. Si calcola il prezzo finale del titolo con la formula ricavata precedentemente, per  $n$  traiettorie che rappresentano i possibili andamenti del sottostante:

$$S_i(T) = S(0)e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z_i}, \quad i = 1, \dots, n.$$

3. Si calcola il prezzo scontato:

$$C_i = e^{-rT}(S_i(T) - K)^+, \quad i = 1, \dots, n.$$

4. Attraverso il metodo Monte Carlo calcolo il valore atteso:

$$\tilde{C}_n = \frac{C_1 + \dots + C_n}{n}.$$

La stima ottenuta,  $\tilde{C}_n$ , tende al valore atteso all'aumentare di  $n$ , ovvero tende al payoff dell'opzione call:

$$\tilde{C}_n \approx E[C_i] \approx E[e^{-rT}(S(T) - K)^+].$$

Abbiamo visto che il payoff di una call Europea dipende solo dal prezzo finale del sottostante,  $S(T)$ , e non da tutto il suo andamento. Tuttavia nella valutazione di derivati più complessi è necessario considerare tutta la dinamica del sottostante  $S$ .

Un esempio si ha nelle opzioni Asiatiche. La dinamica del sottostante segue l'equazione differenziale stocastica (2.5), ma queste opzioni dipendono dall'andamento del sottostante lungo tutto l'intervallo  $[0, T]$ , in quanto il payoff è dato da  $(\bar{S} - K)^+$  con

$$\bar{S} = \frac{1}{m} \sum_{j=1}^m S(t_j), \quad 0 = t_0 < t_1 < \dots < t_m = T.$$

Il payoff di un'opzione Asiatica dipende quindi dall'andamento medio del sottostante durante l'intervallo di definizione  $[0, T]$ .

Per calcolare l'attesa del prezzo scontato  $E[e^{-rT}(\bar{S} - K)^+]$  è quindi necessario simulare tutta la traiettoria del sottostante. Per fare ciò si suddivide l'intervallo  $[0, T]$  in  $m$  sottointervalli e si calcola il valore di  $S$  in tali punti:

$$S(t_{j+1}) = S(t_j)e^{[(r-\frac{1}{2}\sigma^2)(t_{j+1}-t_j)+\sigma\sqrt{t_{j+1}-t_j}Z_{j+1}]}, \quad j = 0, \dots, m-1,$$

dove  $Z_1, \dots, Z_m$  sono variabili aleatorie indipendenti con distribuzione normale standard. L'algoritmo per simulare l'attesa del payoff scontato per un'opzione Asiatica, si basa sui seguenti passi:

1. Si suddivide l'intervallo  $[0, T]$  in  $m$  sottointervalli.
2. Si generano delle variabili aleatorie normali standard:  
 $Z_{i,j}$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ .  
 In particolare  $Z_{i,j}$  indica la variabile al tempo  $j - mo$  per l' $i - ma$  traiettoria.
3. Si calcola il sottostante lungo i punti dell'intervallo, ripetendo il procedimento per ottenere diverse traiettorie:

$$S_i(t_j) = S_i(t_{j-1})e^{[(r - \frac{1}{2}\sigma^2)(t_j - t_{j-1}) + \sigma\sqrt{t_j - t_{j-1}}Z_{i,j}]} \quad i = 1, \dots, n \quad j = 1, \dots, m$$

con valore iniziale  $S(0)$  fissato.

4. Si calcola la media del sottostante:

$$\bar{S}_i = \frac{S_i(t_1) + \dots + S_i(t_m)}{m}, \quad i = 1, \dots, n.$$

5. Per ogni traiettoria si calcola il prezzo scontato:

$$C_i = e^{-rT}(\bar{S} - K)^+, \quad i = 1, \dots, n.$$

6. Si stima l'attesa con il metodo Monte Carlo:

$$\tilde{C} = \frac{C_1 + \dots + C_n}{n}.$$

## 2.5 Tecniche di riduzione della varianza

Le tecniche di riduzione della varianza incrementano l'efficienza della simulazione del metodo Monte Carlo, in quanto riducono la varianza delle stime simulate.

In particolare consideriamo il metodo delle variabili antitetiche. Tale tecnica riduce la varianza associando alle variabili aleatorie generate altre variabili correlate ad esse. L'idea alla base del metodo è data dalla seguente osservazione: se  $U$  è una variabile aleatoria con distribuzione uniforme sull'intervallo  $[0, 1]$ , allora anche  $1 - U$  è una variabile con distribuzione uniforme.

Di conseguenza se generiamo una traiettoria utilizzando  $U_1, U_2, \dots, U_n$  variabili aleatorie distribuite uniformemente, possiamo generare un'ulteriore traiettoria attraverso le variabili  $1 - U_1, 1 - U_2, \dots, 1 - U_n$  senza modificare la legge del processo simulato.

Ad ogni variabile  $U_i$  possiamo quindi associare la corrispondente variabile  $1 - U_i$ . Le coppie  $(U_i, 1 - U_i)$  che si formano, sono dette antitetiche in quanto ad ogni grande o piccolo



valore di  $U_i$  corrisponde un valore antitetico rispettivamente piccolo o grande. Questo fa sì che una variabile aleatoria insolitamente grande o piccola utilizzata per simulare una traiettoria, venga bilanciata dal corrispondente valore antitetico ottenendo una traiettoria antitetica che riduce così la varianza delle stime simulate.

Tali considerazioni si estendono anche alle altre tipologie di distribuzioni attraverso il metodo della funzione di ripartizione inversa: applicando la trasformazione inversa della funzione di ripartizione  $F$  a variabili uniformemente distribuite in  $[0, 1]$ , si ottengono delle variabili aleatorie con distribuzione data da  $F$ .

Nel nostro caso, se  $F$  è una funzione di ripartizione invertibile ed  $U$  una variabile aleatoria distribuita uniformemente allora  $F^{-1}(U)$  e  $F^{-1}(1 - U)$  hanno la stessa distribuzione di  $F$  e sono variabili antitetiche, in quanto  $F^{-1}$  è monotona. In particolare, per una variabile aleatoria  $Z$  normale standard la corrispondente variabile antitetica è data dal valore stesso ma con segno opposto,  $-Z$ .

Per cui ad ogni traiettoria di un moto Browniano simulata attraverso una sequenza di variabili aleatorie  $Z_1, Z_2, \dots, Z_n$  indipendenti e identicamente distribuite con distribuzione normale standard, possiamo associare la traiettoria antitetica simulata a partire dalle corrispondenti variabili antitetiche  $-Z_1, -Z_2, \dots, -Z_n$ . Le traiettorie così ottenute risultano simmetriche rispetto all'asse delle ascisse per l'origine. La figura 2.1 mostra quattro traiettorie di un moto Browniano standard simulate attraverso la tecnica delle variabili antitetiche. Si può notare la simmetria rispetto all'asse delle ascisse.

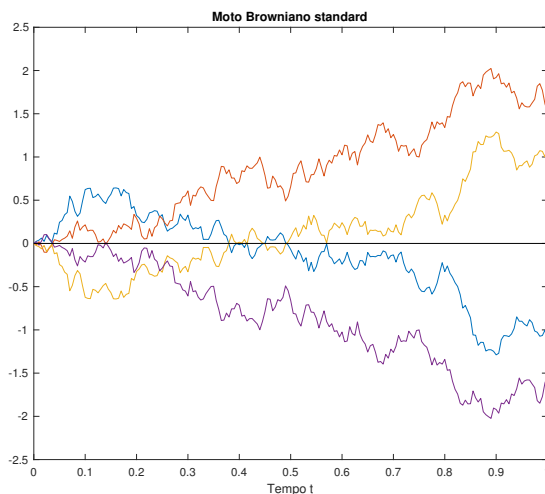


Figura 2.1: Quattro traiettorie del moto Browniano standard simulate con le variabili antitetiche.

Riportiamo l'algoritmo in linguaggio Matlab per simulare un moto Browniano standard  $W$ , riducendo la varianza con la tecnica appena descritta. Seguendo la notazione utilizzata nel Capitolo 1, indichiamo con  $b$  il numero delle traiettorie che si vogliono simulare,  $n$  il numero dei punti di discretizzazione del tempo e  $dt$  il passo della discretizzazione.

---

```
function [W]=mBstandard(b,n,dt)
W=zeros(n+1,b);
for i=2:n+1
    Z=randn(1,b/2);
    W(i,:)=W(i-1,:)+sqrt(dt)*[Z -Z];
end
end
```

---

Analogamente, utilizzando le traiettorie antitetiche possiamo simulare un moto Browniano  $X$  con drift  $\mu$ , coefficiente di diffusione  $\sigma^2$  e valore iniziale  $X_0$ .

---

```
function [X]=mB(b,n,dt,mu,sigma,X0)
X=zeros(n+1,b);
X(1,:)=X0;
for i=2:n+1
    Z=randn(1,b/2);
    X(i,:)=X(i-1,:)+mu*dt+sigma*sqrt(dt)*[Z -Z];
end
end
```

---

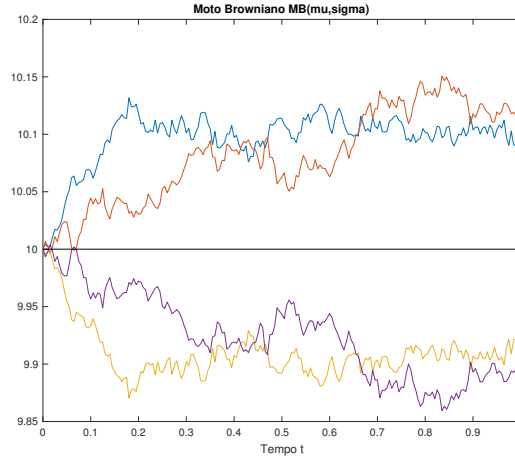
In questo caso si ha una simmetria tra le traiettorie simulate e quelle antitetiche rispetto all'asse delle ascisse passante dal valore iniziale  $X_0$ , come in figura 2.2.

Analizziamo ora il metodo delle variabili antitetiche applicate al metodo Monte Carlo. Sia  $Y$  una variabile aleatoria, per stimare l'attesa  $E[Y]$  l'implementazione della tecnica appena descritta genera una sequenza di coppie antitetiche aventi la stessa distribuzione di  $Y$ :

$$(Y_1, \tilde{Y}_1), (Y_2, \tilde{Y}_2), \dots, (Y_n, \tilde{Y}_n).$$

La stima dell'attesa ottenuta applicando questo metodo,  $\hat{Y}_{VA}$ , è data dalla media di tutte le  $2n$  variabili. Questa si può riscrivere attraverso  $n$  variabili indipendenti ottenute dalla media delle coppie antitetiche:

$$\hat{Y}_{VA} = \frac{1}{2n} \left( \sum_{i=1}^n Y_i + \sum_{i=1}^n \tilde{Y}_i \right) = \frac{1}{n} \sum_{i=1}^n \left( \frac{Y_i + \tilde{Y}_i}{2} \right). \quad (2.6)$$

Figura 2.2: Quattro traiettorie del moto Browniano  $MB(\mu, \sigma^2)$  simulate con le variabili antitetiche.

Poiché le variabili:

$$\frac{Y_1 + \tilde{Y}_1}{2}, \frac{Y_2 + \tilde{Y}_2}{2}, \dots, \frac{Y_n + \tilde{Y}_n}{2}$$

sono indipendenti e hanno la stessa distribuzione, lo stimatore  $\hat{Y}_{VA}$  ha varianza:

$$\begin{aligned} \text{var}[\hat{Y}_{VA}] &= \text{var} \left[ \frac{1}{n} \sum_{i=1}^n \frac{Y_i + \tilde{Y}_i}{2} \right] = \frac{1}{n^2} \sum_{i=1}^n \text{var} \left[ \frac{Y_i + \tilde{Y}_i}{2} \right] = \frac{1}{n} \text{var} \left[ \frac{Y_i + \tilde{Y}_i}{2} \right] = \\ &= \frac{1}{4n} \text{var}[Y_i + \tilde{Y}_i] = \frac{1}{4n} (\text{var}[Y_i] + \text{var}[\tilde{Y}_i] + 2\text{cov}(Y_i, \tilde{Y}_i)) = \\ &= \frac{1}{2n} (\text{var}[Y_i] + \text{cov}(Y_i, \tilde{Y}_i)). \end{aligned} \quad (2.7)$$

Confrontiamo questi risultati con lo stimatore classico di Monte Carlo,  $\hat{Y}_{MC}$ , ottenuto dalla media di  $2n$  variabili aleatorie indipendenti e identicamente distribuite secondo  $Y$ :

$$\hat{Y}_{MC} = \frac{1}{2n} \sum_{i=1}^{2n} Y_i, \quad (2.8)$$

con varianza:

$$\text{var}[\hat{Y}_{MC}] = \text{var} \left[ \frac{1}{2n} \sum_{i=1}^{2n} Y_i \right] = \frac{1}{4n^2} \sum_{i=1}^{2n} \text{var}[Y_i] = \frac{1}{2n} \text{var}[Y_i]. \quad (2.9)$$

Per avere una riduzione della varianza, imponiamo che la varianza nel metodo delle variabili antitetiche sia minore rispetto a quella classica di Monte Carlo:

$$\text{var}[\hat{Y}_{VA}] < \text{var}[\hat{Y}_{MC}],$$

$$\frac{1}{2n}(\text{var}[Y_i] + \text{cov}(Y_i, \tilde{Y}_i)) < \frac{1}{2n}\text{var}[Y_i],$$

ovvero

$$\text{var}[Y_i] + \text{cov}(Y_i, \tilde{Y}_i) < \text{var}[Y_i].$$

Questo si verifica quando:

$$\text{cov}(Y_i, \tilde{Y}_i) < 0. \tag{2.10}$$

Perciò, rispetto all'applicazione del metodo classico, la campionatura antitetica porta ad una riduzione della varianza se si ha una correlazione negativa tra le variabili simulate e le corrispondenti antitetiche.

## Capitolo 3

# Metodo di Longstaff Schwartz

La quotazione delle opzioni, in particolare di tipo Americano, rappresenta un argomento molto importante e tutt'oggi oggetto di ricerca nell'ambito finanziario.

Una delle difficoltà principali è data dalla loro prerogativa di poter essere esercitate in ogni momento dell'intervallo di definizione, diversamente dalle opzioni Europee che sono esercitabili solo a scadenza. Per tale caratteristica la valutazione dell'opzione Americana è data dal valore di esercizio ottimale. Precisamente ad ogni tempo di possibile esercizio, si confronta il payoff immediato con il payoff atteso nel caso in cui l'opzione venga mantenuta. L'opzione viene esercitata solo nel caso in cui il payoff attuale risulti migliore. Il nodo cruciale nella valutazione di un'opzione Americana consiste quindi nello stimare il payoff atteso supponendo di mantenere l'opzione e non esercitarla.

L'approccio ideato da Francis A. Longstaff ed Eduardo S. Schwartz consiste nell'approssimare il valore dell'opzione Americana per simulazione.

Per prima cosa vengono simulati diversi andamenti del sottostante. Attraverso la programmazione dinamica ed al metodo di regressione, andando a ritroso a partire dal tempo di scadenza, si stima il payoff atteso in caso di continuazione. Dal confronto del payoff atteso con quello attuale si ottiene un'approssimazione del valore dell'opzione ad ogni tempo. Il metodo Monte Carlo, infine, stima la quotazione attuale dell'opzione Americana.

In questo capitolo analizzeremo in dettaglio l'approccio di Longstaff e Schwartz. Vedremo innanzitutto come si costruisce l'algoritmo e lo applicheremo ad un esempio semplice per poter osservare ogni passaggio. Presteremo attenzione anche all'andamento delle opzioni al variare dei parametri principali.

### 3.1 Procedimento di Longstaff e Schwartz

Per analizzare il metodo Longstaff-Schwartz consideriamo un'opzione put Americana esercitabile nell'intervallo di tempo  $[0, T]$  con strike  $K$  al tempo di scadenza  $T$  e con tasso d'interesse costante  $r$ . Supponiamo che il sottostante  $X(t)$ ,  $t \in [0, T]$ , sia modellizzabile attraverso un moto Browniano geometrico con drift dato dal tasso di interesse  $r$  e volabilità  $\sigma^2$ :  $GBM(r, \sigma^2)$ .

Perciò:

$$X(t) = X_0 e^{(r - \frac{1}{2}\sigma^2)t + \sigma W_t}, \quad t \in [0, T]$$

dove  $X_0$  è il valore iniziale e  $W_t$  è un moto Browniano standard, come visto nella sezione (1.4).

L'obiettivo dell'algoritmo è stimare il valore al tempo 0,  $V(X_0)$ , del derivato Americano:

$$V(X_0) = \sup_{t \in [0, T]} E[e^{-rt} h(X(t))],$$

dove  $h(X(t))$  rappresenta il payoff al tempo  $t$  che, nel caso dell'opzione put considerata, è dato da:

$$h(X(t)) = (K - X(t))^+.$$

Poichè il valore che si vuole approssimare è dato dall'estremo superiore di un valore atteso rispetto al tempo, è necessario risolvere un problema di ottimizzazione trovando il tempo d'arresto ottimale:

$$t^* = \inf\{t \geq 0; X(t) \leq b^*(t)\}.$$

La variabile  $t^*$  indica il primo momento in cui il sottostante raggiunge un limite di esercizio ottimale  $b^*$  non noto.

In questo modo il valore  $V(X_0)$  dell'opzione corrisponde più semplicemente all'attesa del payoff scontato calcolato al tempo dell'arresto ottimale:

$$V(X_0) = E[e^{-rt^*} h(X(t^*))]. \quad (3.1)$$

Il metodo di Longstaff-Schwartz utilizza l'approccio della programmazione dinamica, che consiste nel risolvere il problema di ottimizzazione dividendolo in sottoproblemi e combinando tra loro le soluzioni ottenute.

Suddividiamo quindi l'intervallo  $[0, T]$  in sottointervalli di uguale lunghezza,  $\Delta t = \frac{T}{m}$ , andando a definire un insieme finito di punti,  $\{0, t_1, t_2, \dots, t_m\}$  con  $t_m = T$ , nel quale ogni punto  $t_i$  rappresenta un'opportunità d'esercizio.

Assumendo il numero dei punti  $m$  sufficientemente grande, l'opzione Americana viene quindi approssimata con un'opzione Bermuda.

Un'opzione Bermuda è un derivato finanziario in cui il titolare ha il diritto, ma non l'obbligo, di esercitare l'opzione solo in alcune date predefinite che si verificano ad intervalli regolari per tutta la durata del contratto. Per questo, esse rappresentano una via di mezzo

tra le opzioni Americane, esercitabili in ogni momento, e quelle Europee, esercitabili solo a scadenza. Lo stesso nome richiama la loro natura ibrida, in quanto le Isole Bermuda sono situate nell'Oceano Atlantico fra America ed Europa.

L'algoritmo Longstaff-Schwartz simula quindi  $b$  traiettorie del sottostante attraverso il moto Browniano geometrico discretizzato secondo lo schema di Eulero, come visto nel Capitolo (1). Nella simulazione delle traiettorie è possibile introdurre una tecnica della riduzione della varianza. In questo caso abbiamo usato il metodo delle traiettorie antitetiche. E' stata così costruita una griglia stocastica di traiettorie indipendenti.

Utilizziamo una nuova notazione:

$X_{i,j}$  indica la  $j$ -ma traiettoria all'opportunità d'esercizio  $i$ -ma, per  $i = 1, \dots, m$  e  $j = 1, \dots, b$ ; mentre  $X_i = (X_{i,1}, \dots, X_{i,b})$  identifica più in generale tutte le traiettorie al tempo d'esercizio  $i$ -mo.

La traccia seguita in questa prima parte è dunque caratterizzata dai seguenti passaggi:

- suddivido  $[0, T]$  in  $m$  sottointervalli;
- $X_{i,j}$  = traiettoria  $j$ -ma del  $MBG(r, \sigma^2)$  al tempo  $i$ , con valore iniziale  $X_0$ .  
 $i = 0, \dots, m$  e  $j = 1, \dots, b$ .

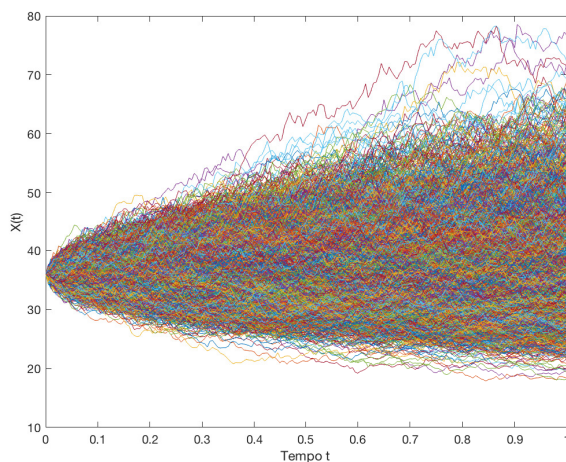


Figura 3.1: Simulazione di 5000 traiettorie del sottostante modellizzate attraverso il moto Browniano geometrico con:  $X_0 = 36$ ,  $r = 0.06$ ,  $\sigma = 0.2$  e  $T = 1$

Per ogni opportunità d'esercizio  $t_i$ ,  $i = 1, \dots, m - 1$ , il titolare dell'opzione deve decidere se risulta conveniente esercitare oppure mantenere l'opzione.

Al fine di compiere la scelta ottimale è necessario confrontare il payoff attuale con il payoff

atteso in caso di continuazione dell'opzione, ed esercitare solo nel caso in cui il payoff attuale sia maggiore.

Per ogni traiettoria, risulta necessario definire il valore di continuazione  $C(X_i)$ , come il valore scontato atteso al tempo successivo  $t_{i+1}$  a condizione che l'opzione non venga esercitata fino al tempo  $t_{i+1}$ :

$$C(X_i) = E[e^{-r\Delta t}V(X_{i+1})|X_i]. \quad (3.2)$$

L'algoritmo Longstaff-Schwartz usa il metodo della regressione lineare dei minimi quadrati per stimare il valore di continuazione.

Questo è possibile perché l'attesa condizionata è un elemento dello spazio delle funzioni a quadrato sommabile,  $L^2$ . Quest'ultimo è uno spazio di Hilbert, possiede una base ortonormale ed ogni suo elemento, compresa l'attesa condizionata, può essere rappresentato come una combinazione lineare delle funzioni base. Il valore di continuazione, essendo definito attraverso l'attesa condizionata, eredita questa proprietà.

Una volta scelto un insieme di  $M$  funzioni base ortogonali,  $\{\psi_1(x), \psi_2(x), \dots, \psi_M(x)\}$ , il valore di continuazione può così essere espresso attraverso una combinazione lineare:

$$C(X_i) = \sum_{r=1}^M \beta_{i,r} \psi_r(X_i), \quad i = 1, \dots, m \quad \text{fissato}, \quad (3.3)$$

dove  $\beta_{i,r}$  indicano le costanti di regressione.

La scelta delle funzioni base da utilizzare, presenta molteplici opzioni. Longstaff e Schwartz, nel loro articolo "Valuing American Option by simulation: a simple least-squares approach", utilizzano i polinomi di Laguerre pesati, come descritto in [4]:

$$l_0(x) = e^{-\frac{x}{2}} \quad l_1(x) = e^{-\frac{x}{2}}(1-x) \quad l_2(x) = e^{-\frac{x}{2}}\left(1-2x+\frac{x^2}{2}\right)$$

$$l_n(x) = e^{-\frac{x}{2}} \frac{e^x}{n!} \frac{d^n}{dx^n} (x^n e^{-x}).$$

Tuttavia vengono ammessi anche i polinomi di Laguerre non pesati:

$$L_0(x) = 1 \quad L_1(x) = 1-x \quad L_2(x) = \frac{1}{2}(2-4x+x^2)$$

$$L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n} (x^n e^{-x})$$

i polinomi di Hermite:

$$H_0(x) = 1 \quad H_1(x) = x \quad H_2(x) = x^2 - 1$$

$$H_n(x) = (-1)^n e^{\frac{x^2}{2}} \frac{d^n}{dx^n} e^{-\frac{x^2}{2}}$$



i polinomi di Legendre:

$$P_0(x) = 1 \quad P_1(x) = x \quad P_2(x) = \frac{1}{2}(3x^2 - 1)$$

$$P_n(x) = (2^n n!)^{-1} \frac{d^n}{dx^n} [(x^2 - 1)^n]$$

i polinomi di Chebyshev:

$$T_0(x) = 1 \quad T_1(x) = x \quad T_2(x) = 2x^2 - 1$$

$$T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x)$$

ma anche più semplicemente i monomi:

$$M_0(x) = 1 \quad M_1(x) = x \quad M_2(x) = x^2$$

$$M_n(x) = x^n.$$

Il numero delle funzioni base necessarie per ottenere una buona approssimazione non cresce esponenzialmente con la dimensione del problema ma rimane basso, spesso sono sufficienti 4-5 funzioni base come approfondiremo nella Sezione (3.3). Questa caratteristica, come vedremo, è dovuta alla stima dei coefficienti di regressione.

Rappresentiamo la formulazione (3.3) del valore di continuazione, in forma vettoriale:

$$C(X_i) = \beta_i^T \psi(X_i), \quad (3.4)$$

con  $\beta_i^T = (\beta_{i,1}, \dots, \beta_{i,M})$  e  $\psi(x) = (\psi_1(x), \dots, \psi_M(x))^T$ .

Al tempo  $t_i$ , i coefficienti di regressione  $\beta_i$  vengono calcolati attraverso la regressione lineare ai minimi quadrati.

Confrontando la definizione (3.2) del valore di continuazione con la forma vettoriale (3.4), si ottiene la seguente formulazione:

$$E[e^{-r\Delta t} V(X_{i+1}) | X_i] = \beta_i^T \psi(X_i), \quad (3.5)$$

con  $\beta_i^T = (\beta_{i,1}, \dots, \beta_{i,M})$  e  $\psi(x) = (\psi_1(x), \dots, \psi_M(x))^T$ .

Dalla (3.5) possiamo ricavare un'espressione per il vettore  $\beta_i$ :

$$\beta_i = (E[\psi(X_i)\psi(X_i)^T])^{-1} E[\psi(X_i)e^{-r\Delta t} V(X_{i+1})] =: B_\psi^{-1} B_{\psi V} \quad (3.6)$$

dove  $B_\psi$  indica una matrice  $M \times M$  non singolare per ipotesi e  $B_{\psi V}$  indica un vettore di lunghezza  $M$ .

Utilizziamo il metodo Monte Carlo per calcolare i valori attesi  $B_\psi$  e  $B_{\psi V}$ .

Per applicare Monte Carlo dovremmo usare tutte le  $b$  traiettorie simulate, tuttavia Longstaff e Schwartz consigliano di utilizzare solo le traiettorie in the money del sottostante.

Per traiettorie in the money si intendono le traiettorie per le quali il payoff è strettamente positivo. Nel caso della put considerata nel nostro esempio, le traiettorie in the money al tempo  $t_i$  fissato hanno valore del sottostante strettamente minore dello strike:  $V(X_i) < K$ . In questo modo si riduce la regione sulla quale viene stimata l'attesa condizionata del valore di continuazione, così sono necessarie molte meno funzioni base per ottenere un'approssimazione accurata.

Usiamo Monte Carlo per stimare la matrice  $B_\psi = E[\psi(X_i)\psi(X_i)^T]$ .

Otteniamo  $\hat{B}_\psi$  matrice  $M \times M$ , che assumiamo per ipotesi non singolare, data dalle combinazioni delle funzioni base calcolate nei valori del sottostante al tempo  $t_i$  per tutte le traiettorie in the money.

L'elemento  $\hat{B}_\psi(p, q)$  è definito da:

$$\hat{B}_\psi(p, q) = \frac{1}{N} \sum_{j=1}^N \psi_p(X_{i,j})\psi_q(X_{i,j}), \quad (3.7)$$

dove  $N$  rappresenta il numero delle traiettorie in the money al tempo  $t_i$ .

Analogamente applichiamo Monte Carlo per stimare il vettore

$B_{\psi V} = E[\psi(X_i)e^{-r\Delta t}V(X_{i+1})]$ . Otteniamo  $\hat{B}_{\psi V}$  vettore di lunghezza  $M$ , dato dalle funzioni base calcolate nei valori del sottostante al tempo  $t_i$  per le sole traiettorie in the money, combinate con i valori stimati scontati al tempo successivo  $t_{i+1}$ .

L'elemento  $\hat{B}_{\psi V}(p)$  è definito da:

$$\hat{B}_{\psi V}(p) = \frac{1}{N} \sum_{j=1}^N \psi_p(X_{i,j})e^{-r\Delta t}V(X_{i+1,j}). \quad (3.8)$$

Possiamo così stimare i coefficienti di regressione:

$$\hat{\beta}_i = \hat{B}_\psi^{-1}\hat{B}_{\psi V}. \quad (3.9)$$

Per semplificare i calcoli, suggeriamo di definire  $L$ , una matrice  $M \times N$  data dalle funzioni di base calcolate nelle traiettorie in the money al tempo  $t_i$  fissato.

Precisamente:

$$L(p, j) = \psi_p(X_{i,j}), \quad (3.10)$$

con  $i = 1, \dots, m - 1$  fissato,  $p = 1, \dots, M$  e  $j = 1, \dots, N$ .

Inoltre definisco  $v$ , il vettore di lunghezza  $N$  dei valori dell'opzione al tempo  $t_{i+1}$  scontati, per le traiettorie in the money. Ovvero:

$$v(j) = e^{-r\Delta t}V(X_{i+1,j}), \quad (3.11)$$

con  $i = 1, \dots, m - 1$  fissato e  $j = 1, \dots, N$ .

Gli elementi  $\hat{B}_\psi$  e  $\hat{B}_{\psi V}$  vengono così calcolati:

$$\hat{B}_\psi = \frac{1}{N}(L \cdot L'); \quad \hat{B}_{\psi V} = \frac{1}{N}(L \cdot v'). \quad (3.12)$$

Sostituiamo le espressioni (3.12) appena trovate, nella formulazione (3.9):

$$\begin{aligned}\hat{\beta}_i &= \hat{B}_\psi^{-1} \hat{B}_{\psi V} = \left( \frac{1}{N} L L' \right)^{-1} \cdot \left( \frac{1}{N} L v' \right) = N(L'^{-1} L^{-1}) \cdot \left( \frac{1}{N} L v' \right) = \\ &= L'^{-1} (L^{-1} L) v' = (L')^{-1} v'.\end{aligned}$$

Otteniamo così una formulazione più semplice e veloce per stimare i coefficienti di regressione:

$$\hat{\beta}_i = (L')^{-1} v'. \quad (3.13)$$

Ora, usando la (3.4), possiamo ottenere una stima dei valori di continuazione,  $\hat{C}(X_i)$ :

$$\hat{C}(X_i) = \hat{\beta}_i^T \psi(X_i). \quad (3.14)$$

Abbiamo quindi calcolato il payoff di esercizio immediato  $h(X_i)$  ed il valore di continuazione  $\hat{C}(X_i)$ , relativi all'opportunità d'esercizio  $t_i$ . Dal confronto di questi due valori decidiamo se conviene esercitare l'opzione oppure mantenerla. Per ogni traiettoria in the money, se il payoff  $h(X_{i,j})$  è maggiore del valore di continuazione  $\hat{C}(X_{i,j})$ , esercitiamo l'opzione e aggiorniamo la stima del valore dell'opzione al tempo  $t_i$  con il valore del payoff. In caso contrario, con  $\hat{C}(X_{i,j})$  maggiore del payoff  $h(X_{i,j})$ , l'opzione non viene esercitata e aggiorniamo la stima del valore dell'opzione al tempo  $t_i$  con il valore scontato dell'opzione relativa al tempo  $t_{i+1}$ .

Con la formulazione seguente riassumiamo il confronto appena descritto:

$$\hat{V}(X_{i,j}) = \begin{cases} h(X_{i,j}), & \text{se } h(X_{i,j}) \geq \hat{C}(X_{i,j}) \\ e^{-r\Delta t} \hat{V}(X_{i+1,j}), & \text{se } h(X_{i,j}) < \hat{C}(X_{i,j}). \end{cases} \quad (3.15)$$

Invece, per le traiettorie on the money, cioè per le quali il payoff al tempo  $t_i$  è nullo, il valore dell'opzione in  $t_i$  viene aggiornato semplicemente con il valore dell'opzione al tempo  $t_{i+1}$  moltiplicato per il fattore di sconto:

$$\hat{V}(X_{i,j}) = e^{-r\Delta t} \hat{V}(X_{i+1,j}). \quad (3.16)$$

Il metodo di regressione lineare dei minimi quadrati, come visto, si basa sulla seguente procedura:

- $\hat{V}(X_{m,j}) = h(X_{m,j})$ , considero il payoff al nodo terminale
- per  $t_i$  da  $t_{m-1}$  a  $t_1$ :
  - $itm = \{j_1, \dots, j_N\}$ , considero le traiettorie in the money
  - calcolo la matrice  $L$  delle funzioni base

- calcolo il vettore  $v$  dei prezzi in  $t_{i+1}$  scontati
- calcolo  $\hat{\beta}_i$ , coefficienti di regressione
- stimo  $\hat{C}(X_{i,j})$ , valore di continuazione
- per tutte le traiettorie  $j$  da 1 a  $b$ , confronto e valuto l'opzione in  $t_i$ :  
 se  $j \in itm$  e  $h(X_{i,j}) > \hat{C}(X_{i,j})$   
 allora  $\hat{V}(X_{i,j}) = h(X_{i,j})$   
 altrimenti  $\hat{V}(X_{i,j}) = e^{-r\Delta t}\hat{V}(X_{i+1,j})$ .

Con il metodo di regressione abbiamo valutato l'opzione secondo una strategia ottimale, andando a ritroso dal tempo di scadenza  $t_m = T$  al tempo  $t_1 = \Delta t$ , per tutte le traiettorie simulate del sottostante.

Essendo  $\hat{V}(X_1)$  il valore ottimale dell'opzione al tempo  $t_1$ , allora tale valore può essere espresso con il payoff in funzione della regola d'arresto ottimale  $t^*$ :

$$\hat{V}(X_1) = e^{-r(t^*-t_1)}h(X_{t^*}).$$

Lavoriamo algebricamente questa espressione moltiplicando entrambi i membri per il fattore di sconto:

$$e^{-r\Delta t}\hat{V}(X_1) = e^{-r\Delta t}e^{-r(t^*-t_1)}h(X_{t^*}) = e^{-r(t_1-t_0)}e^{-r(t^*-t_1)}h(X_{t^*}) = e^{-r(t^*-t_0)}h(X_{t^*}).$$

Poiché  $t_0 = 0$  per ipotesi, si ottiene la relazione:

$$e^{-r\Delta t}\hat{V}(X_1) = e^{-rt^*}h(X_{t^*}). \quad (3.17)$$

Questo ci permette di formulare diversamente l'espressione per la quotazione dell'opzione. Sostituendo la (3.17) in (3.1), possiamo calcolare il valore dell'opzione attraverso il valore  $\hat{V}(X_1)$  stimato con il metodo di regressione, piuttosto che usare esplicitamente il tempo d'arresto ottimale  $t^*$ .

La quotazione dell'opzione Americana è perciò calcolata come:

$$V(X_0) = E[e^{-r\Delta t}\hat{V}(X_1)]. \quad (3.18)$$

Per stimare il valore atteso in (3.18), applichiamo il metodo Monte Carlo.

La quotazione dell'opzione è approssimata dai valori ottimali stimati in  $t_1$  moltiplicati per il fattore di sconto, per tutte le traiettorie:

$$\hat{V}(X_0) = \frac{1}{b} \sum_{j=1}^b e^{-r\Delta t}\hat{V}(X_{1,j}). \quad (3.19)$$

L'algoritmo completo di Longstaff-Schwartz in linguaggio di programmazione Matlab, è riportato in Appendice.

## 3.2 Esempio numerico

Al fine di analizzare in dettaglio l'approccio di Longstaff e Schwartz, consideriamo un semplice esempio numerico andando a ripercorrere i passaggi dell'algoritmo sviluppato nel paragrafo precedente e descritto in Appendice.

Consideriamo una put Americana su un sottostante modellizzabile attraverso un moto Browniano geometrico, con valore iniziale di 1.00, volatilità del 20% e tasso di interesse del 6%. L'opzione put è esercitabile con strike di 1.10 fino alla data di scadenza pari a 3 anni. Lo scopo di questo esempio è vedere come funziona l'algoritmo e l'idea che si sviluppa dietro al metodo di Longstaff-Schwartz. Per non appesantirlo, si considerano solamente 8 traiettorie esercitabili in 3 possibili date: 1, 2, 3.

Ovviamente il numero limitato di traiettorie e di opportunità d'esercizio non consente un'approssimazione accettabile della quotazione dell'opzione, per questa trattazione rimandiamo al paragrafo (3.3). Tuttavia, questo esempio permette una descrizione più intuitiva del metodo che vogliamo analizzare.

Seguendo la notazione vista nella sezione (3.1), le variabili note sono:

$$X_0 = 1.00 \quad \sigma = 0.2 \quad r = 0.06 \quad K = 1.10$$

$$T = 3 \quad m = 3 \quad b = 8$$

L'intervallo  $[0, T]$  viene suddiviso in  $m = 3$  intervalli di lunghezza  $step = 1$ , così le opportunità d'esercizio si hanno in:

$$\mathbf{t} = [0 \quad 1 \quad 2 \quad 3].$$

Simulo ora  $\mathbf{b} = 8$  traiettorie del sottostante  $X$  con il moto Browniano geometrico discretizzato secondo il metodo di Eulero, come visto nella sezione 1.4.1.

Possiamo introdurre anche il metodo delle variabili antitetiche, con 4 traiettorie simulate e 4 antitetiche, per ridurre la varianza nel metodo Monte Carlo ed ottenere un'approssimazione migliore.

---

```
Z=randn(m,b/2);
e=exp((mu-0.5*sigma^2)*dt + sigma*sqrt(dt)*[Z -Z]);
X=cumprod([X0*ones(1,b); e]);
```

---

Nella matrice  $X$  memorizzo i valori delle traiettorie simulate nei tempi di possibile esercizio.

X	path	1	2	3	4	5	6	7	8
t=0		1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
t=1		1.1477	0.9795	1.2432	0.8852	1.1108	0.7391	1.1094	1.0346
t=2		1.4692	1.0812	1.0287	0.5113	0.9941	0.7537	1.2293	1.0419
t=3		1.7684	0.9614	0.8646	0.7096	1.3609	0.7475	1.0762	1.2294

Longstaff e Schwartz massimizzano il valore dell'opzione in ogni opportunità d'esercizio, lungo ogni traiettoria simulata. Per fare questo applichiamo il metodo di regressione: partendo dal tempo di scadenza e andando a ritroso fino al primo tempo d'arresto, si confronta il payoff attuale con il valore di continuazione e si decide se esercitare o meno l'opzione.

Consideriamo quindi il tempo di scadenza  $t = 3$  e determiniamo il payoff nel nodo terminale di ogni traiettoria.

---

```
V=max(K-X(m+1,:),0);
```

---

V	path	1	2	3	4	5	6	7	8
t=3		0	0.1386	0.2354	0.3904	0	0.3525	0.0238	0

---

Procedendo a ritroso, valuto l'opportunità d'esercizio  $t = 2$ .

L'indice con il quale si avvia il ciclo *for* della regressione, è dato da  $i = m = 3$ .

Nella stima del valore di continuazione utilizziamo solamente le traiettorie in the money, per le quali il payoff attuale è strettamente positivo.

Per comodità di calcolo memorizziamo i loro indici nel vettore  $itm$  ed isoliamo i corrispondenti valori al tempo  $t = 2$ , nel vettore  $Xitm$ .

---

```
i=3;
itm=find(X(i,:)<K);
Xitm=X(i,itm);
```

---

In questo esempio si hanno  $N = 6$  traiettorie in the money al tempo  $t = 2$ .

Perciò:

$$itm = [2 \ 3 \ 4 \ 5 \ 6 \ 8]$$

Xitm	path	2	3	4	5	6	8
	t=2	1.0812	1.0287	0.5113	0.9941	0.7537	1.0419

Al fine di semplificare i calcoli nella stima dei valori di continuazione, costruiamo il vettore  $Vitm$  seguendo la formula (3.16) vista nella sezione (3.1). Consideriamo i payoff al tempo  $t = 3$ , memorizzati in  $V$ , per le traiettorie in the money e li moltiplichiamo per il fattore di sconto. Inoltre, poichè il fattore di sconto  $e^{-r\Delta t}$ , dipende dal tasso d'interesse  $r$  e dal passo di discretizzazione, possiamo calcolare a priori tale fattore in quanto non varia durante tutta la durata dell'algoritmo.

In questo caso, il fattore di sconto ed il vettore  $Vitm$  ottenuti, corrispondono a:

$$\text{sconto} = 0.9418$$

Vitm	path	2	3	4	5	6	8
	t=2	0.1305	0.2217	0.3677	0	0.3320	0

Per stimare il valore di continuazione, Longstaff e Schwartz utilizzano il metodo di regressione lineare ai minimi quadrati, proiettando i payoff scontati di  $Vitm$  sulle funzioni base, per ogni traiettoria in the money.

In questo esempio, scegliamo una costante e  $M = 3$  monomi come funzioni base:

$$\psi_0(x) = 1 \quad \psi_1(x) = x \quad \psi_2(x) = x^2 \quad \psi_3(x) = x^3.$$

Per semplificare i calcoli nella stima dei coefficienti di regressione, costruiamo la matrice  $L$  delle funzioni base valutate in tutte le traiettorie in the money al tempo  $t = 2$ .

---

```
L=[ ones(size(Xitm)); Xitm; Xitm.^ 2; Xitm.^ 3];
```

---

Nel nostro particolare caso, si ottiene:

L	path	2	3	4	5	6	8
M=0		1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
M=1		1.0812	1.0287	0.5113	0.9941	0.7537	1.0419
M=2		1.1690	1.0582	0.2614	0.9882	0.5681	1.0856
M=3		1.2639	1.0886	0.1337	0.9824	0.4281	1.1310

Attraverso il vettore  $Vitm$  e la matrice  $L$ , possiamo determinare i coefficienti di regressione  $\hat{\beta}_i$  come in (3.13).

---


$$\text{beta} = L' \backslash Vitm';$$


---

$$\text{beta} = \begin{bmatrix} -5.7720 \\ 25.7560 \\ -34.1725 \\ 14.2494 \end{bmatrix}$$

Questo consente di ottenere il valore di continuazione al tempo  $t = 2$ , per ogni traiettoria in the money.

Si segue la formula (3.14).

---


$$C = \text{beta}' * L;$$


---

C	path	2	3	4	5	6	8
t=2		0.1380	0.0729	0.3681	0.0602	0.3290	0.0837

Infine calcoliamo il valore di esercizio immediato per le traiettorie in the money. Tale valore è determinato dal payoff al tempo  $t = 2$ .

---


$$E = K - Xitm;$$


---

E	path	2	3	4	5	6	8
t=2		0.0188	0.0713	0.5887	0.1059	0.3463	0.0581

Ora abbiamo entrambi gli elementi per determinare la strategia ottimale in  $t = 2$  per tutte le traiettorie.

Tra le traiettorie in the money, i cui indici sono memorizzati nel vettore  $itm$ , consideriamo solo quelle per le quali conviene esercitare, ovvero per le quali il valore di esercizio è maggiore del valore di continuazione. Memorizziamo gli indici che le identificano in un nuovo vettore,  $ex$ .



---

```
ex = itm(C<=E);
```

---

$$\mathbf{ex} = [4 \ 5 \ 6], \quad \text{in } t = 2.$$

Per esclusione memorizziamo gli indici di tutte le traiettorie rimanenti, che comprendono:

- le traiettorie on the money, ovvero quelle con payoff nullo in  $t = 2$ ;
- le traiettorie in the money per le quali conviene mantenere l'opzione in  $t = 2$ , ovvero quelle con il valore di continuazione maggiore del valore di esercizio.

Memorizziamo gli indici corrispondenti in un nuovo vettore.

---

```
cont = setdiff(1:b,ex);
```

---

$$\mathbf{cont} = [1 \ 2 \ 3 \ 7 \ 8], \quad \text{in } t = 2.$$

Avendo trovato la strategia ottimale, determiniamo il valore dell'opzione al tempo  $t = 2$ :

- per le traiettorie nelle quali conviene esercitare, il valore dell'opzione corrisponde al valore d'esercizio;
- per le traiettorie nelle quali conviene mantenere l'opzione senza esercitare, il valore dell'opzione corrisponde al valore stimato al tempo  $t = 3$  moltiplicato per il fattore di sconto.

Per non appesantire l'algoritmo, non creiamo una nuova variabile ma riscriviamo i valori sul vettore  $V$ , che conteneva il valore dell'opzione per  $t = 3$ .

---

```
V(ex) = E(C<E);
```

```
V(cont) = V(cont)*sconto;
```

---

V	path	1	2	3	4	5	6	7	8
	t=2	0	0.1305	0.2217	0.5887	0.1059	0.3463	0.0224	0

Il vettore  $V$  contiene quindi i valori dell'opzione put, stimati al tempo  $t = 2$ .

Ripetiamo ora il procedimento visto per stimare l'opzione al tempo  $t = 1$ , ripercorrendo il ciclo *for* dell'algoritmo con indice  $i = 2$ .

Analogamente a prima identifichiamo le traiettorie in the money alla data  $t = 1$ .

---

```
i=2;
itm=find(X(i,:)≤K);
```

---

$$\text{itm} = [2 \quad 4 \quad 6 \quad 8], \quad \text{per } t = 1.$$

In questo caso si hanno  $N = 4$  traiettorie in the money.  
I corrispondenti valori del sottostante si rimemorizzano in  $X_{itm}$ .

---

```
Xitm=X(i,itm);
```

---

Xitm	path	2	4	6	8
t=1		0.9795	0.8852	0.7391	1.0346

Per la stima dei valori di continuazione ricalcoliamo il vettore  $V_{itm}$  per le traiettorie in the money, moltiplicando i payoff in  $t = 2$  con il fattore di sconto.  
Calcoliamo anche la matrice delle funzioni base, prendendo sempre una costante e  $M = 3$  monomi.

---

```
Vitm=V(itm)*sconto;
L=[ones(size(X)); X; X.^ 2; X. ^ 3];
```

---

Vitm	path	2	4	6	8
t=1		0.1229	0.5544	0.3261	0

$$L = \begin{bmatrix} 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 0.9795 & 0.8852 & 0.7391 & 1.0346 \\ 0.9594 & 0.7836 & 0.5463 & 1.0704 \\ 0.9398 & 0.6936 & 0.4037 & 1.1074 \end{bmatrix}$$

$V_{itm}$  e  $L$  ci permettono di trovare  $\hat{B}_\psi$  e  $\hat{B}_{\psi V}$ , e di conseguenza anche i coefficienti di regressione.

---

```
beta = L'\Vitm';
```

---

$$\text{beta} = \begin{bmatrix} -106.9429 \\ 356.2906 \\ -388.8115 \\ 139.5184 \end{bmatrix}$$

Combinando linearmente i coefficienti di regressione con le funzioni base si stimano i valori di continuazione in  $t = 1$ , per ogni traiettoria in the money.

Calcoliamo anche i valori di esercizio al tempo  $t = 1$  per le traiettorie in the money.

---

```
C = beta' *L;
```

```
E = K-Xitm;
```

---

C	path	2	4	6	8
t=1		0.1229	0.5544	0.3261	-0.0000

E	path	2	4	6	8
t=1		0.1205	0.2148	0.3609	0.0654

Come visto nell'iterazione precedente, distinguiamo le traiettorie in the money con il valore di esercizio immediato maggiore del valore di continuazione.

---

```
ex = itm(C<E);
```

```
cont = setdiff(1:b,ex);
```

---

$$\text{ex} = [6 \ 8] \quad \text{cont} = [1 \ 2 \ 3 \ 4 \ 5 \ 7].$$

Stimiamo quindi il valore dell'opzione al tempo  $t = 1$ , aggiornando il vettore  $V$ .

L'opzione assume il valore di esercizio nelle traiettorie per le quali conviene esercitare. In alternativa l'opzione prende il valore al tempo  $t = 2$  e lo moltiplica per il fattore di sconto.

---

```
V(ex) = E(C<E);
```

```
V(cont) = V(cont)*sconto;
```

---

V	path	1	2	3	4	5	6	7	8
t=1	0	0.1229	0.2088	0.5544	0.0997	0.3609	0.0211	0.0654	

Abbiamo così applicato il metodo di regressione, trovando i valori ottimali per l'opportunità d'esercizio  $t = 1$ .

Come spiegato nella sezione (3.1), la quotazione dell'opzione al tempo iniziale è data dal metodo Monte Carlo, applicato ai valori in  $t = 1$  moltiplicati per un fattore di sconto, seguendo la formula (3.19).

---

$V_0 = \text{mean}(V * \text{sconto});$

---

$$V_0 = 0.1687$$

Il valore ottenuto,  $V_0 = 0.1687$ , è stato stimato utilizzando solo 8 traiettorie e discretizzando il tempo in soli 3 intervalli. Aumentando il numero delle traiettorie e raffinando la discretizzazione, possiamo trovare una quotazione migliore, più vicina al valore esatto. Per esempio simulando 50000 traiettorie con 200 opportunità d'esercizio, la quotazione dell'opzione diventa  $V_0 = 0.1340$ .

Risulta interessante, inoltre, confrontare i valori ottenuti con l'esercizio anticipato rispetto al caso in cui si esercita solo a scadenza. In tal caso il modello equivale ad un'opzione Europea valutabile con la formula di Black and Scholes e la quotazione corrisponde a  $V_0 = 0.0952$ .

Nella sezione 3.3 analizzeremo meglio il confronto tra una valutazione di tipo Americana ad esercizio anticipato ed una valutazione di tipo Europea a scadenza. Vedremo inoltre l'andamento della quotazione al variare di alcuni parametri, quali il numero e il tipo di funzioni basi, il numero di intervalli di tempo, ma anche il dato iniziale e la volatilità del sottostante.

### 3.3 Esempi e tabelle di put Americane

Nella sezione (3.2) abbiamo utilizzato un esempio semplice per illustrare come questo approccio venga applicato nella valutazione di un'opzione put Americana.

Consideriamo ora un esempio più consistente al fine di analizzare il risultato ottenuto con l'algoritmo di Longstaff-Schwartz.

Prendiamo un'opzione put Americana, con sottostante modellizzabile attraverso un moto Browniano geometrico che segue l'equazione differenziale stocastica:

$$dX(t) = rX(t) dt + \sigma X(t) dW_t,$$

con valore iniziale  $X_0$  dato.

Seguendo la notazione vista nelle sezioni precedenti,  $r$  indica il tasso di interesse,  $\sigma$  la volatilità, mentre  $W$  il moto Browniano standard. Assumiamo inoltre che l'opzione sia esercitabile fino al tempo di scadenza  $T$ , con strike  $K$ .

Variamo alcuni parametri della put Americana per osservare l'andamento del metodo di Longstaff e Schwartz, a seconda delle diverse caratteristiche di un'opzione.

Come riportato in Tabella 3.1, incrementiamo il valore iniziale del sottostante di due punti per volta, partendo dal valore  $X_0 = 36$  fino a  $X_0 = 44$ . Manteniamo costante il valore del tasso d'interesse  $r$  del 6%, mentre aumentiamo la volatilità dal 20% al 40%. Raddoppiamo inoltre il tempo  $T$  di possibile durata dell'opzione, considerando prima 1 anno poi 2 anni. Il prezzo dello strike  $K$ , invece, rimane fisso a 40.

Il metodo di Longstaff-Schwartz prevede la suddivisione dell'intervallo  $[0, T]$ . Per ottenere una discretizzazione sufficiente, consideriamo 200 sottointervalli per ogni anno. In questo modo lo step di discretizzazione rimane costante indipendentemente dalla scadenza  $T$  considerata. Vengono inoltre simulate 100000 traiettorie, delle quali 50000 sono antitetiche. Infine, per applicare il metodo di regressione previsto nell'algoritmo di Longstaff-Schwartz, bisogna scegliere la tipologia ed il numero delle funzioni base. Consideriamo allora i primi 5 polinomi di Laguerre, formulati come nella sezione 3.1.

La Tabella 3.1 riporta i valori stimati con il metodo di Longstaff-Schwartz. Tali risultati vengono poi confrontati con le quotazioni ottenute applicando lo schema delle differenze finite. La colonna "LS-Diff" esprime la differenza tra i valori ottenuti con Longstaff-Schwartz e quelli calcolati con le differenze finite. Come si può notare, la variazione tra i risultati stimati dai due metodi è molto piccola, in diversi casi è minore dell'1% in valore assoluto.

La Tabella 3.1 prende in considerazione anche l'eventualità in cui l'opzione non venga esercitata in anticipo, ma solo a scadenza. In tal caso il valore ottenuto equivale ad una put di tipo Europeo, valutabile attraverso il modello Black and Scholes. L'ultima colonna della tabella, "LS-Europea", ci permette di osservare come l'esercizio anticipato permetta una migliore valutazione dell'opzione.

Nell'esempio appena considerato abbiamo stimato il valore dell'opzione con l'algoritmo Longstaff-Schwartz utilizzando i polinomi di Laguerre per il metodo di regressione. Tuttavia si possono applicare diverse tipologie di polinomi, come elencato in sezione (3.1).

In Tabella 3.2 riproponiamo le opzioni già analizzate, valutando l'approccio di Longstaff-Schwartz utilizzando i polinomi di Laguerre, di Hermite, di Legendre, di Chebyshev ed i monomi. Per consentire un confronto incrociato tra i valori della Tabella 3.1 e quelli della Tabella 3.2, manteniamo inalterati i parametri iniziali stabiliti precedentemente.

Tabella 3.1: Consideriamo delle opzioni put Americane variando  $X_0, \sigma, T$ .

Si mantengono fissi  $r = 0.06$  e  $K = 40$ .

In "LS": stime ottenute con il metodo di Longstaff-Shwartz usando 200 discretizzazioni, 100000 traiettorie e 5 polinomi di Laguerre.

In "Diff. finite": stime ottenute con il metodo delle differenze finite.

In "opz. Europea": valori delle opzioni mantenute fino a scadenza.

In "LS-Diff": differenze tra Longstaff-Schwartz e le differenze finite.

In "LS-Europea": differenze tra Longstaff-Schwartz e i valori dell'opzione Europea.

$X_0$	$\sigma$	$T$	LS	Diff. finite	pz. Europea	LS-Diff	LS-Europea
36	0.20	1	4.4872	4.4807	3.844	0.0065	0.643
36	0.20	2	4.8419	4.8432	3.763	-0.0013	1.079
36	0.40	1	7.1082	7.1001	6.711	0.0081	0.397
36	0.40	2	8.5276	8.4943	7.700	0.0333	0.828
38	0.20	1	3.2747	3.2519	2.852	0.0228	0.423
38	0.20	2	3.7696	3.7425	2.991	0.0271	0.780
38	0.40	1	6.1462	6.1502	5.834	-0.0004	0.312
38	0.40	2	7.6727	7.6530	6.979	0.0197	0.694
40	0.20	1	2.3041	2.3143	2.066	-0.0102	0.238
40	0.20	2	2.8862	2.8815	2.356	0.0047	0.530
40	0.40	1	5.3240	5.3093	5.060	0.0147	0.264
40	0.40	2	6.9247	6.8983	6.326	0.0264	0.599
42	0.20	1	1.6259	1.6167	1.465	0.0092	0.161
42	0.20	2	2.2249	2.2120	1.841	0.0129	0.384
42	0.40	1	4.6082	4.5821	4.379	0.0261	0.230
42	0.40	2	6.2606	6.2349	5.736	0.0257	0.525
44	0.20	1	1.1230	1.1095	1.017	0.0135	0.106
44	0.20	2	1.6943	1.6871	1.429	0.0072	0.265
44	0.40	1	3.9566	3.9448	3.783	0.0118	0.174
44	0.40	2	5.6601	5.6337	5.202	0.0264	0.458

Tabella 3.2: Consideriamo delle opzioni put Americane variando  $X_0, \sigma, T$ .

Si mantengono fissi  $r = 0.06$  e  $K = 40$ .

Confrontiamo le stime di Longstaff-Schwartz ottenute con 200 discretizzazioni, 100000 traiettorie e 5 polinomi di Laguerre, Hermite, Chebyshev, Legendre ed i monomi.

$X_0$	$\sigma$	$T$	Laguerre	Hermite	Chebyshev	Legendre	Monomi
36	0.20	1	4.4872	4.4940	4.4881	4.4936	4.4813
36	0.20	2	4.8419	4.8504	4.8539	4.8525	4.8361
36	0.40	1	7.1082	7.1429	7.1117	7.1088	7.1093
36	0.40	2	8.5276	8.5559	8.4995	8.4899	8.5084
38	0.20	1	3.2747	3.2729	3.2648	3.2587	3.2238
38	0.20	2	3.7696	3.7647	3.7347	3.7397	3.7568
38	0.40	1	6.1462	6.1740	6.1485	6.1543	6.1707
38	0.40	2	7.6727	7.6773	7.6629	7.6651	7.6929
40	0.20	1	2.3041	2.3105	2.3060	2.3171	2.3209
40	0.20	2	2.8862	2.8980	2.8880	2.8058	2.8867
40	0.40	1	5.3240	5.3383	5.3348	5.3259	5.3205
40	0.40	2	6.9247	6.8826	6.9291	6.9318	6.9548
42	0.20	1	1.6259	1.6180	1.6289	1.6320	1.6181
42	0.20	2	2.2249	2.2198	2.2151	2.2038	2.2125
42	0.40	1	4.6082	4.5905	4.6227	4.6000	4.5947
42	0.40	2	6.2606	6.2418	6.2260	6.2282	6.2608
44	0.20	1	1.1230	1.1198	1.1103	1.1190	1.1142
44	0.20	2	1.6943	1.6866	1.6789	1.6787	1.6739
44	0.40	1	3.9566	3.9578	3.9422	3.9622	3.9560
44	0.40	2	5.6601	5.6420	5.6611	5.6303	5.6460

### 3.4 Grafici e confronti

Analizziamo in dettaglio nove delle opzioni put Americane viste nelle Tabelle 3.1 e 3.2.

1. Innanzitutto consideriamo un'opzione put con valore iniziale minore del prezzo strike, precisamente:  $X_0 = 36$ ,  $\sigma = 0.2$ ,  $T = 1$ .

Valutiamo tale opzione utilizzando l'algoritmo visto del metodo di regressione con 100000 traiettorie, variando la discretizzazione dell'intervallo di definizione  $[0, T]$ : partiamo da 20 punti fino ad arrivare a 200, aumentando di 5 punti per volta.

L'obiettivo è confrontare gli andamenti della soluzione utilizzando le diverse tipologie dei polinomi usati nel metodo di regressione, come visto in Sezione 3.1.

Per i grafici della figura 3.2, abbiamo valutato l'opzione utilizzando la costante 1 ed i primi 4 polinomi di Laguerre, di Hermite, di Chebyshev, di Lagrange ed i monomi. Per quanto riguarda i polinomi di Laguerre pesati, essi presentano un andamento differente rispetto agli altri polinomi, perciò per ottenere dei risultati analoghi abbiamo dovuto considerarne 5 oltre al valore  $l_0$ , come in sezione 3.1.

A fianco abbiamo rappresentato i corrispettivi errori relativi espressi in percentuale. Come si può notare, gli errori sono inferiori allo 0.4% anche per un numero di discretizzazioni più basso.

Confrontiamo inoltre le valutazioni ottenute variando il numero delle funzioni base. Valutiamo l'opzione prendendo 3, 4 e 5 polinomi di Hermite ed osserviamo anche l'errore relativo corrispondente. Analogamente prendiamo 3, 4 e 5 polinomi di Laguerre pesati. Possiamo osservare che l'andamento dell'opzione varia poco al variare del numero delle basi di Hermite, diversamente aumentando il numero delle basi di Laguerre pesato si ha un netto miglioramento della valutazione. L'andamento per i restanti polinomi è approssimabile al caso di Hermite.

Nella figura 3.3 grafichiamo il tempo di esecuzione dell'algoritmo in base ai diversi polinomi utilizzati. Si osserva che i polinomi di Hermite, Chebyshev, Lagrange ed i monomi si equivalgono, i polinomi di Laguerre sono leggermente più onerosi, mentre i polinomi di Laguerre pesati hanno un costo maggiore. Possiamo inoltre confrontare il tempo di esecuzione al variare del numero delle funzioni base considerate, sia per le basi di Hermite sia per i polinomi di Laguerre pesato.

2. Consideriamo ora la stessa opzione put raddoppiando l'intervallo di definizione. L'opzione sarà caratterizzata da:  $X_0 = 36$ ,  $\sigma = 0.2$ ,  $T = 2$ .

Aumentiamo anche il numero delle discretizzazioni considerate: da 40 punti fino a 400, aumentando di 10 punti per volta.

In figura 3.4 osserviamo i grafici ottenuti prendendo come funzioni base la costante 1 ed i primi 4 polinomi di Laguerre, Hermite, Chebyshev, Lagrange ed i monomi, mentre per i



polinomi di Laguerre pesati consideriamo i primi 6, oltre ad  $l_0$ .

Consideriamo anche il corrispondente grafico dell'errore relativo. Nonostante l'andamento ottenuto con i polinomi di Laguerre pesati si discosti dalle altre tipologie di polinomi, l'errore relativo non supera lo 0.4%.

Confrontiamo anche le valutazioni, ed i corrispondenti errori relativi, al variare del numero delle funzioni base. Consideriamo 3, 4 e 5 polinomi di Hermite e 5, 6 e 7 polinomi di Laguerre pesati.

Per quanto riguarda il tempo di esecuzione, in figura 3.5, si possono riprendere le considerazioni fatte precedentemente, notando tuttavia un generale aumento del costo computazionale dovuto all'aver raddoppiato i punti di discretizzazione: il metodo della regressione ai minimi quadrati deve infatti valutare l'opzione in tutti i 400 punti.

3. Proviamo ora a raddoppiare la volatilità. Consideriamo un'opzione put con  $X_0 = 36, \sigma = 0.4, T = 1$ .

Nella figura 3.6 possiamo osservare i grafici ottenuti. Per valutare l'opzione abbiamo considerato, oltre alla costante 1, 4 polinomi di Laguerre, Hermite, Chebyshev, Legendre ed i monomi, mentre abbiamo preso 8 polinomi di Laguerre pesati, oltre a quello iniziale  $l_0$ .

L'errore relativo per le funzioni base di Laguerre pesato non superano lo 0.8%, mentre per le restanti funzioni l'errore è inferiore allo 0.4% in valore assoluto.

Confrontiamo le valutazioni ed i corrispondenti errori relativi al variare del numero delle funzioni base: consideriamo sempre 3, 4 e 5 funzioni di Hermite (oltre alla costante 1), mentre prendiamo 6, 7 e 8 polinomi di Laguerre pesati (oltre ad  $l_0$ ). Nonostante la volatilità sia maggiore rispetto ai casi precedenti, abbiamo considerato lo stesso numero di funzioni base per i polinomi di Laguerre, Hermite, Chebyshev, Lagrange ed i monomi, mentre abbiamo preso un numero maggiore di polinomi di Laguerre pesato.

Per quanto riguarda il tempo di esecuzione si può notare un aumento del costo computazionale, nel grafico 3.7 rispetto al grafico 3.3, per i polinomi di Laguerre pesati, perchè abbiamo considerato un numero maggiore di basi.

4. Ripetiamo le osservazioni fatte considerando un'opzione put Americana con valore iniziale uguale al prezzo dello strike. L'opzione è perciò caratterizzata da:  $X_0 = 40, \sigma = 0.2, T = 1$ .

In figura 3.8 valutiamo l'opzione considerando i primi 4 polinomi di Laguerre, Hermite, Chebyshev, Legendre ed i monomi, mentre prendiamo 5 polinomi di Laguerre pesati, come visto precedentemente nell'opzione 1.

L'errore relativo non supera lo 0.5% in valore assoluto per tutte le funzioni base considerate.

Nel variare il numero di funzioni base prendiamo 3, 4 e 5 funzioni sia per i polinomi di Hermite che per Laguerre pesato.

In figura 3.9 si ha il tempo di esecuzione che risulta invece inferiore rispetto al caso in figura 3.3, pur mantenendo lo stesso andamento, anche variando il numero di funzioni base.

5. Raddoppiamo ora l'intervallo di definizione e conseguentemente aumentiamo i punti di discretizzazione fino a 400. L'opzione è quindi data da:  $X_0 = 40, \sigma = 0.2, T = 2$ .

In figura 3.10 valutiamo l'opzione attraverso la funzione costante 1 e 4 polinomi di Laguerre, Hermite, Chebyshev, Legendre ed i monomi e 6 polinomi di Laguerre pesati. Abbiamo utilizzato lo stesso numero di funzioni base applicate in figura 3.4 per favorirne il confronto. Avendo un intervallo di definizione più ampio, l'errore relativo è maggiore se si considera un numero basso di discretizzazioni, passando così dallo 0.8% a meno dello 0.2%.

Confrontiamo anche le valutazioni, con i rispettivi errori relativi, considerando 3, 4 e 5 polinomi di Hermite e 5, 6 e 7 polinomi di Laguerre pesati.

Il tempo di esecuzione, rappresentato in figura 3.11, risulta maggiore se paragonato alla figura 3.9 in quanto abbiamo aumentato i punti di discretizzazione. Tuttavia il costo computazionale risulta minore rispetto alla figura 3.5 dove, a parità di discretizzazioni, varia solo il valore iniziale dell'opzione che ora è uguale allo strike.

6. Raddoppiamo nuovamente la volatilità. L'opzione è caratterizzata da:  $X_0 = 40, \sigma = 0.4, T = 1$ .

In figura 3.12 valutiamo l'opzione considerando sempre 4 polinomi di Laguerre, Hermite, Chebyshev, Legendre ed i monomi, mentre prendiamo 8 polinomi di Laguerre pesati.

Osservando il grafico dell'errore relativo, la valutazione dei polinomi di Hermite risulta migliore rispetto ai polinomi di Laguerre pesati, praticamente in ogni discretizzazione considerata.

Confrontiamo anche le valutazioni ottenute variando il numero delle funzioni base considerate: 3, 4 e 5 per Hermite, 6, 7 e 8 per Laguerre pesato.

Il tempo di esecuzione, mostrato in figura 3.13, per un'opzione con prezzo iniziale uguale allo strike risulta inferiore rispetto al caso visto in figura 3.7, per tutti i polinomi considerati.

7. Infine andiamo a considerare il caso in cui l'opzione abbia prezzo iniziale maggiore dello strike. Le caratteristiche dell'opzione put sono:  $X_0 = 44, \sigma = 0.2, T = 1$ .

In figura 3.14 valutiamo l'opzione considerando i primi 4 polinomi di Laguerre, Hermite, Chebyshev, Legendre e monomi. Consideriamo invece 6 polinomi di Laguerre pesati, un numero maggiore rispetto ai casi rappresentati nei grafici 3.2 e 3.8.

Gli errori relativi risultano inferiori allo 0.6% in valore assoluto.

Variamo il numero delle funzioni base prendendo 3, 4 e 5 funzioni per i polinomi di Hermite e 5, 6 e 7 per Laguerre pesato.

Il tempo di esecuzione, rappresentato in figura 3.15, è inferiore rispetto agli esempi visti precedentemente per tutti i polinomi presi in esame, nonostante si utilizzi una base in più per Laguerre pesato.

8. Considerazioni analoghe si possono fare raddoppiando l'intervallo di definizione, prendendo un'opzione con:  $X_0 = 44, \sigma = 0.2, T = 2$ .

In figura 3.16 valutiamo l'opzione prendendo lo stesso numero di funzioni base considerate nelle figure 3.4 e 3.10: 4 per i polinomi per Laguerre, Hermite, Chebyshev, Lagrange e monomi, mentre 6 per i polinomi di Laguerre pesati.

In questo caso gli errori relativi sono vicini per tutte le tipologie di polinomi e sono inferiori allo 0.7% in valore assoluto.

Confrontiamo poi le valutazioni prendendo 3, 4 e 5 polinomi di Hermite per funzioni base e 6, 7 e 8 polinomi di Laguerre pesato.

Anche in questo caso il tempo di esecuzione dell'opzione, in figura 3.17, è inferiore rispetto agli esempi precedenti raffigurati in 3.5 e 3.11.

9. Per ultimo consideriamo il caso di un'opzione con volatilità doppia, ovvero:

$$X_0 = 44, \sigma = 0.4, T = 1.$$

In figura 3.18 valutiamo l'opzione con 4 polinomi di Laguerre, Hermite, Chebyshev, Lagrange e monomi, ed i primi 8 polinomi di Laguerre pesati.

L'errore relativo è inferiore del 6% indipendentemente dalla tipologia di polinomi e dalla discretizzazione considerata.

Confrontiamo le valutazioni al variare del numero delle funzioni base, applicando 3, 4 e 5 polinomi di Hermite e 6, 7 e 8 polinomi di Laguerre pesati. Anche in questo caso il tempo di esecuzione, in figura 3.19, è inferiore rispetto agli esempi precedenti descritti nei grafici 3.7 e 3.13.

In generale, dal confronto incrociato dei grafici, possiamo osservare che gli andamenti delle *valutazioni* ottenute applicando al metodo di regressione i polinomi di Laguerre, Hermite, Chebyshev, Legendre ed i monomi sono simili tra loro e per ottenere una buona approssimazione è sufficiente utilizzare dalle 3 alle 5 funzioni base, indipendentemente dalle caratteristiche dell'opzione.

Gli andamenti delle valutazioni ottenute applicando i polinomi di Laguerre pesati, definiti in sezione 3.1, sono molto diversi rispetto a quelli derivanti dalle altre tipologie di polinomi. La valutazione cambia notevolmente al variare del numero delle funzioni base utilizzate, ma soprattutto è necessario cambiare il numero dei polinomi in base alle caratteristiche dell'opzione: con volatilità bassa,  $\sigma = 0.2$ , e tempo unitario,  $T = 1$ , conviene considerare 5 polinomi di Laguerre pesati, come mostrato in figura 3.2, 3.8 e 3.14; aumentando il tempo di definizione,  $T = 2$ , è necessario aumentare il numero delle funzioni base utilizzando 6, 7 polinomi come nei grafici 3.4, 3.10 e 3.16; infine aumentando la volatilità,  $\sigma = 0.4$ , il metodo di regressione necessita di almeno 7, 8 polinomi di Laguerre pesato per ottenere una buona approssimazione, come risulta dalle figure 3.6, 3.12 e 3.18.

In generale, gli *errori relativi* sono inferiori allo 0.2% in valore assoluto anche per un numero basso di discretizzazioni, mentre raffinando la griglia l'errore cala anche al di sotto dello 0.1%.

Confrontiamo infine i *tempi di esecuzione*. Anche in questo caso i metodi di regressione con i polinomi di Hermite, Chebyshev, Legendre ed i monomi si equivalgono, sebbene l'uso dei polinomi di Laguerre sia leggermente più oneroso. Diversamente il metodo di regressione con i polinomi di Laguerre pesato, secondo la definizione in sezione 3.1, ha un tempo di esecuzione maggiore.

Analizziamo ora l'andamento dei tempi di implementazione del metodo di regressione in base alle caratteristiche dell'opzione da valutare. Osservando parallelamente i grafici possiamo notare che i tempi non subiscono variazioni apprezzabili all'aumentare della volatilità  $\sigma$  dell'opzione.

Le tempistiche crescono invece all'aumentare dell'intervallo di definizione del derivato, in quanto è necessario un numero maggiore di punti per ottenere una griglia ugualmente raffinata e il metodo di regressione deve valutare l'opzione e stimare il suo valore di continuazione in ogni punto della discretizzazione, quindi impiega necessariamente più tempo. Tuttavia, a parità di volatilità  $\sigma$  e di tempo  $T$ , la velocità dell'algoritmo varia al cambiare del valore iniziale del sottostante  $X_0$  rispetto allo strike  $K$ : per  $X_0$  che si avvicina e supera  $K$ , il tempo di esecuzione cala. Questo andamento è dovuto al fatto che il metodo di regressione per la programmazione dinamica valuta solo le traiettorie in the money, come descritto in sezione 3.1. Quindi se il sottostante ha valore iniziale inferiore al payoff il numero di traiettorie in the money sarà maggiore rispetto a quelle di un'opzione con sottostante che parte da un valore uguale o superiore al prezzo del payoff.

Figura 3.2: Opzione:  $X_0 = 36, r = 0.06, \sigma = 0.2$  e  $T = 1$

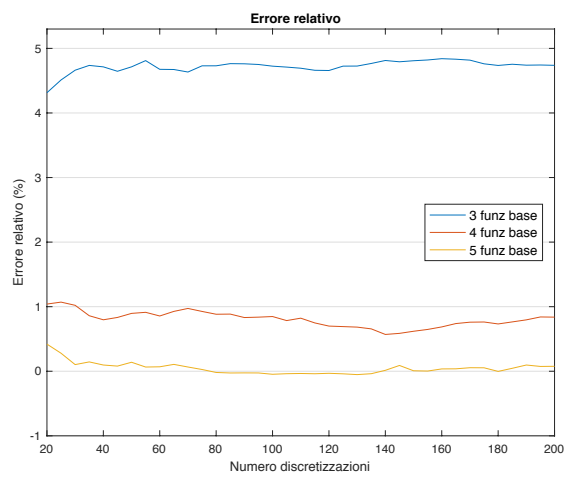
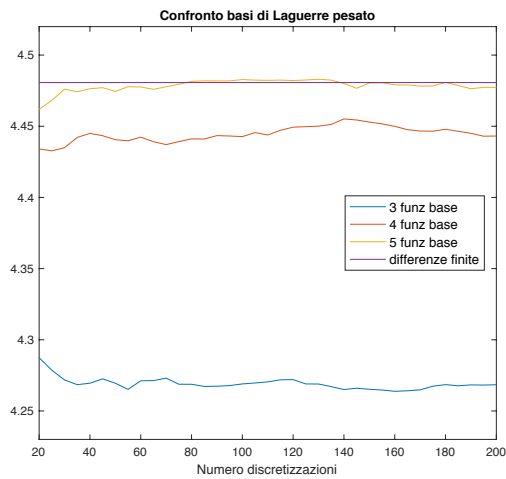
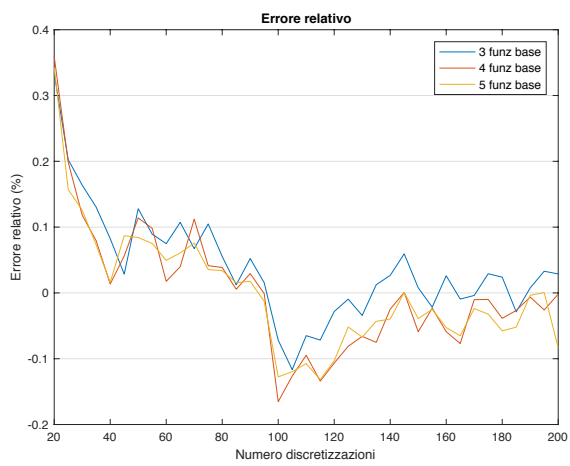
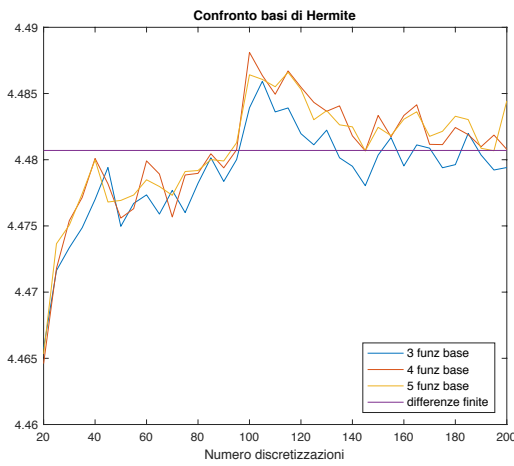
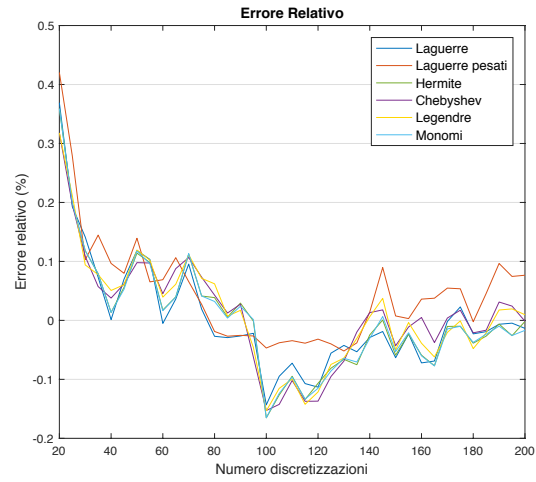
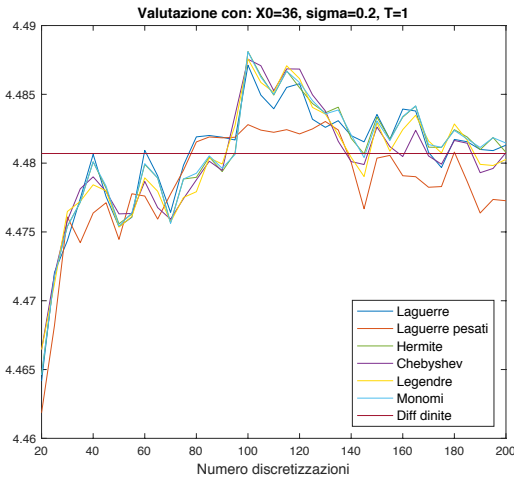


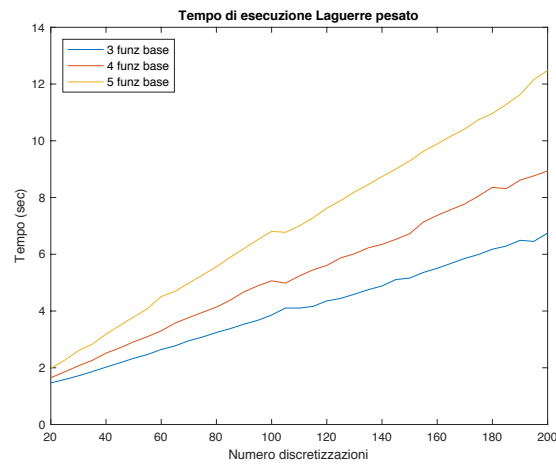
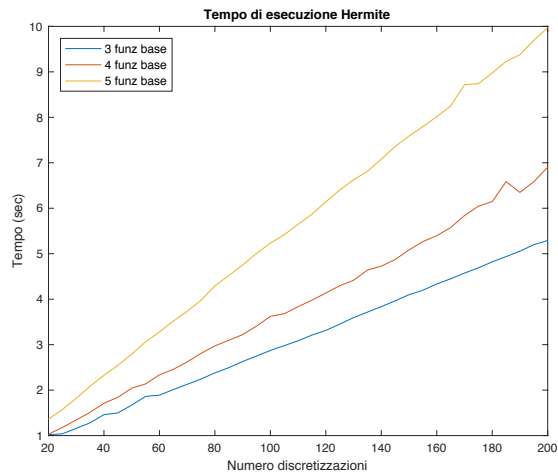
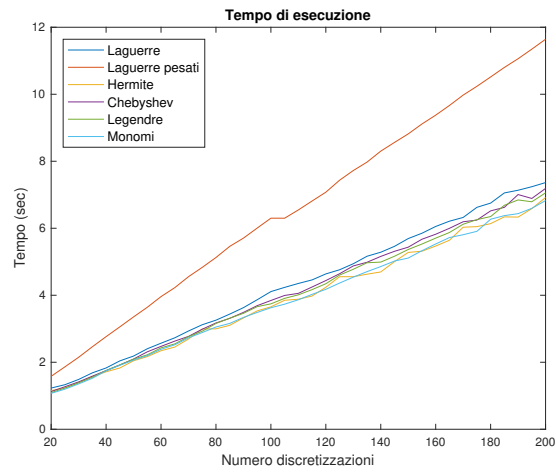
Figura 3.3: Opzione:  $X_0 = 36, r = 0.06, \sigma = 0.2$  e  $T = 1$ 

Figura 3.4: Opzione:  $X_0 = 36, r = 0.06, \sigma = 0.2$  e  $T = 2$

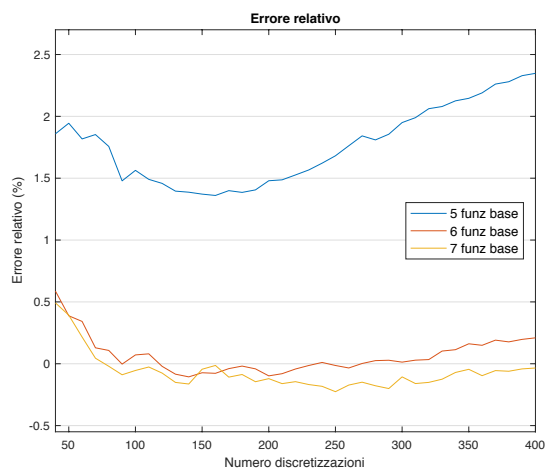
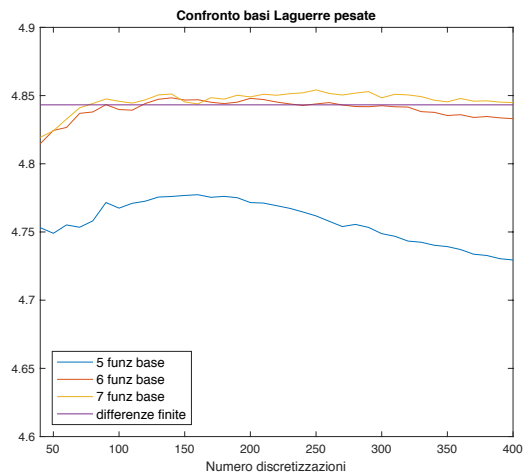
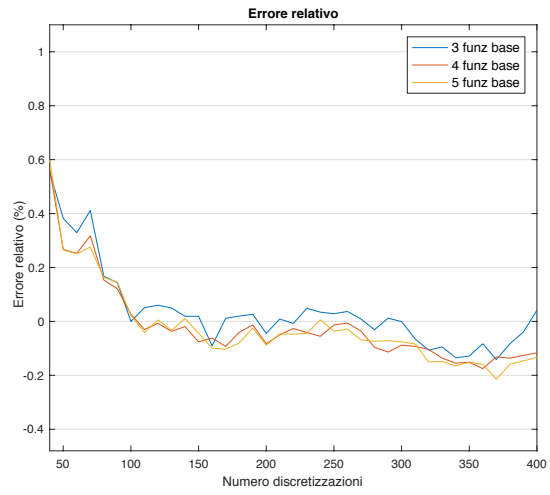
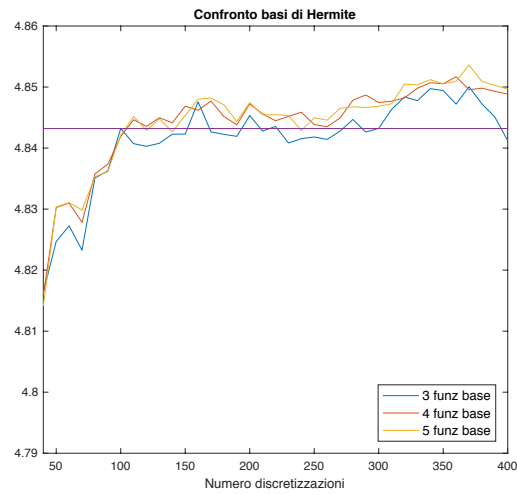
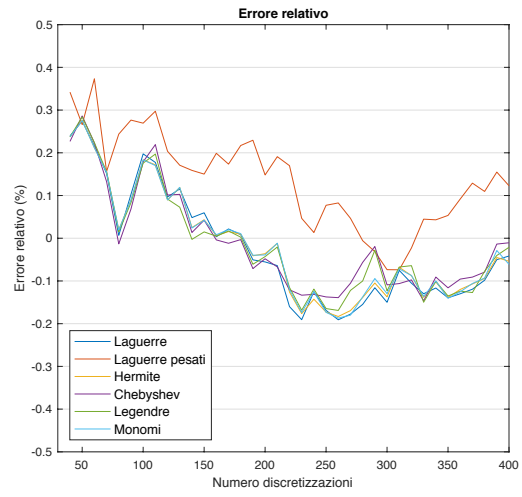


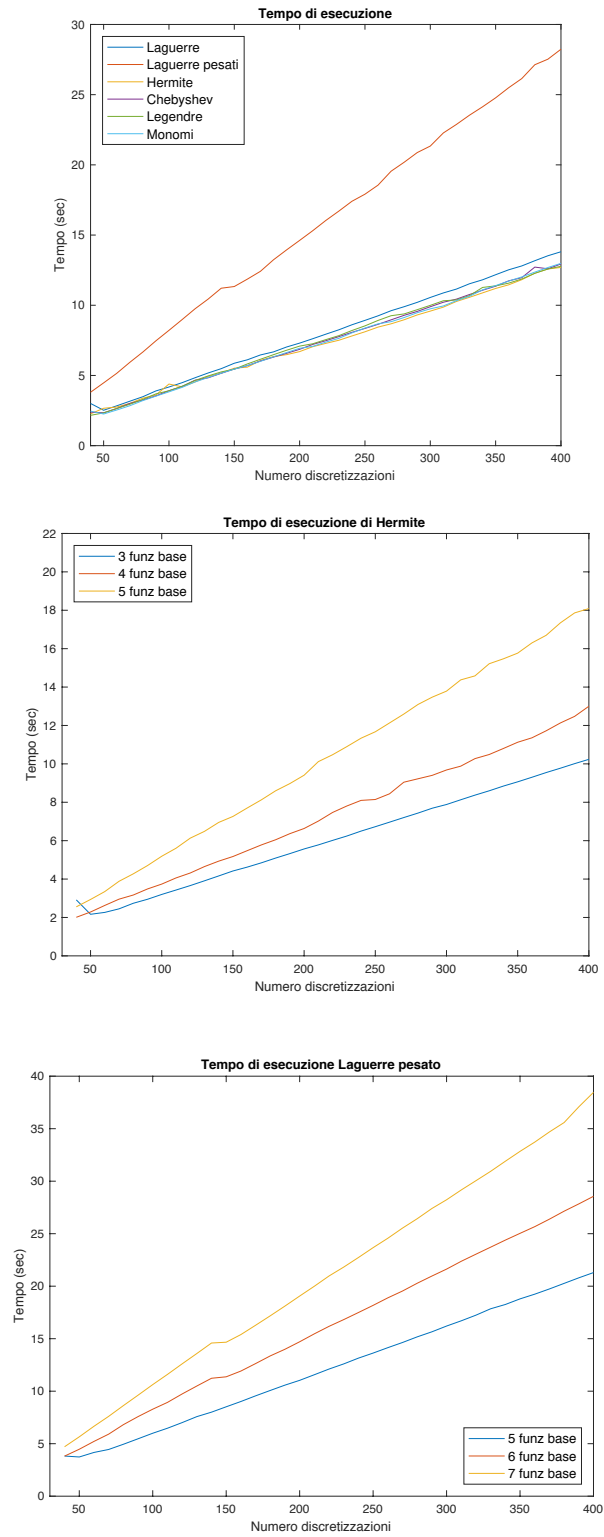
Figura 3.5: Opzione:  $X_0 = 36, r = 0.06, \sigma = 0.2$  e  $T = 2$ 



Figura 3.6: Opzione:  $X_0 = 36, r = 0.06, \sigma = 0.4$  e  $T = 1$

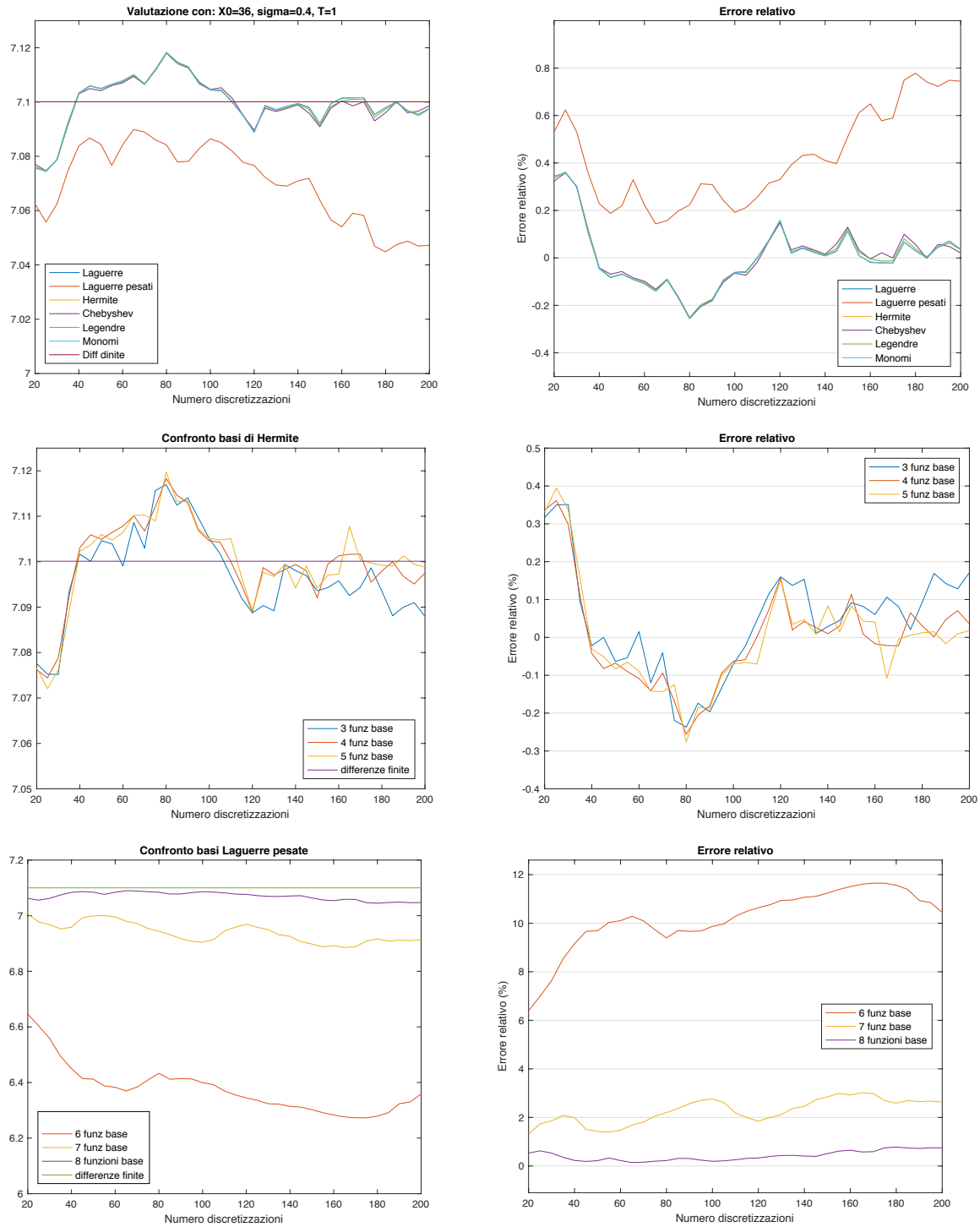


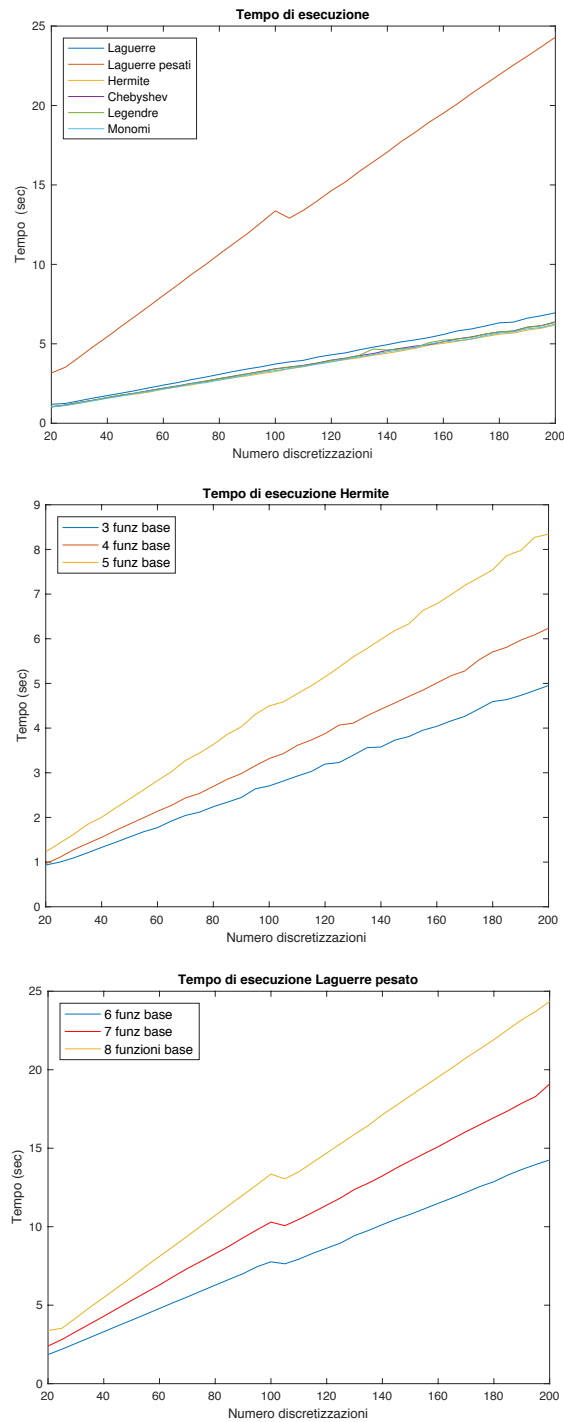
Figura 3.7: Opzione:  $X_0 = 36, r = 0.06, \sigma = 0.4$  e  $T = 1$ 

Figura 3.8: Opzione:  $X_0 = 40, r = 0.06, \sigma = 0.2$  e  $T = 1$

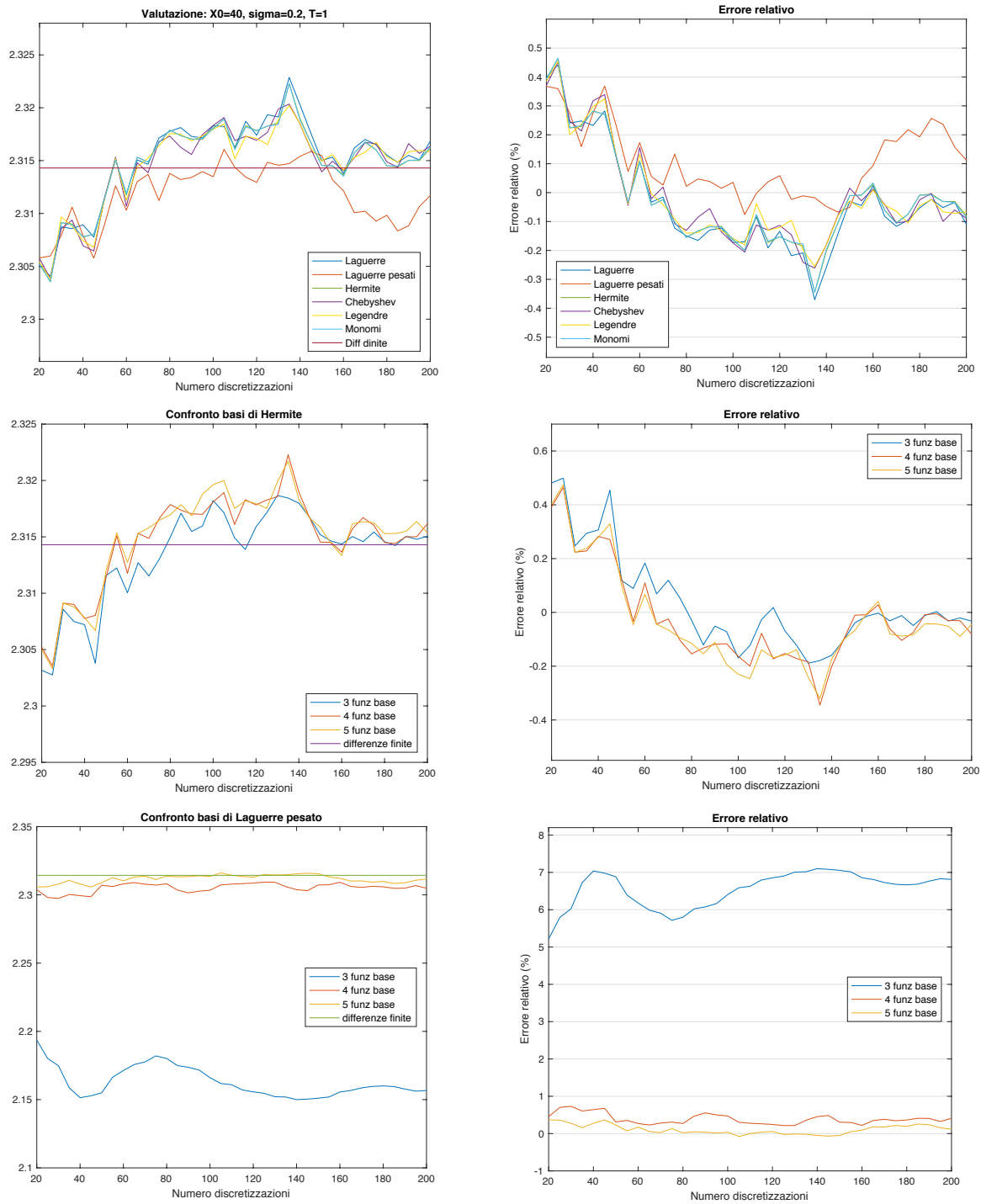


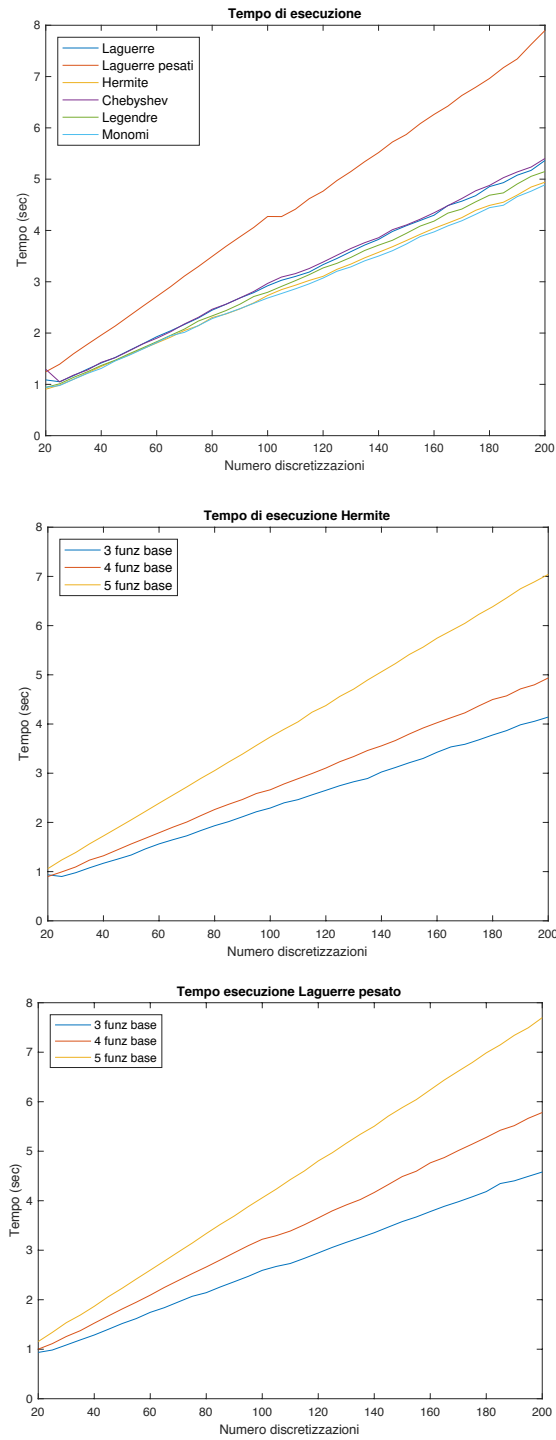
Figura 3.9: Opzione:  $X_0 = 40, r = 0.06, \sigma = 0.2$  e  $T = 1$ 

Figura 3.10: Opzione:  $X_0 = 40, r = 0.06, \sigma = 0.2$  e  $T = 2$

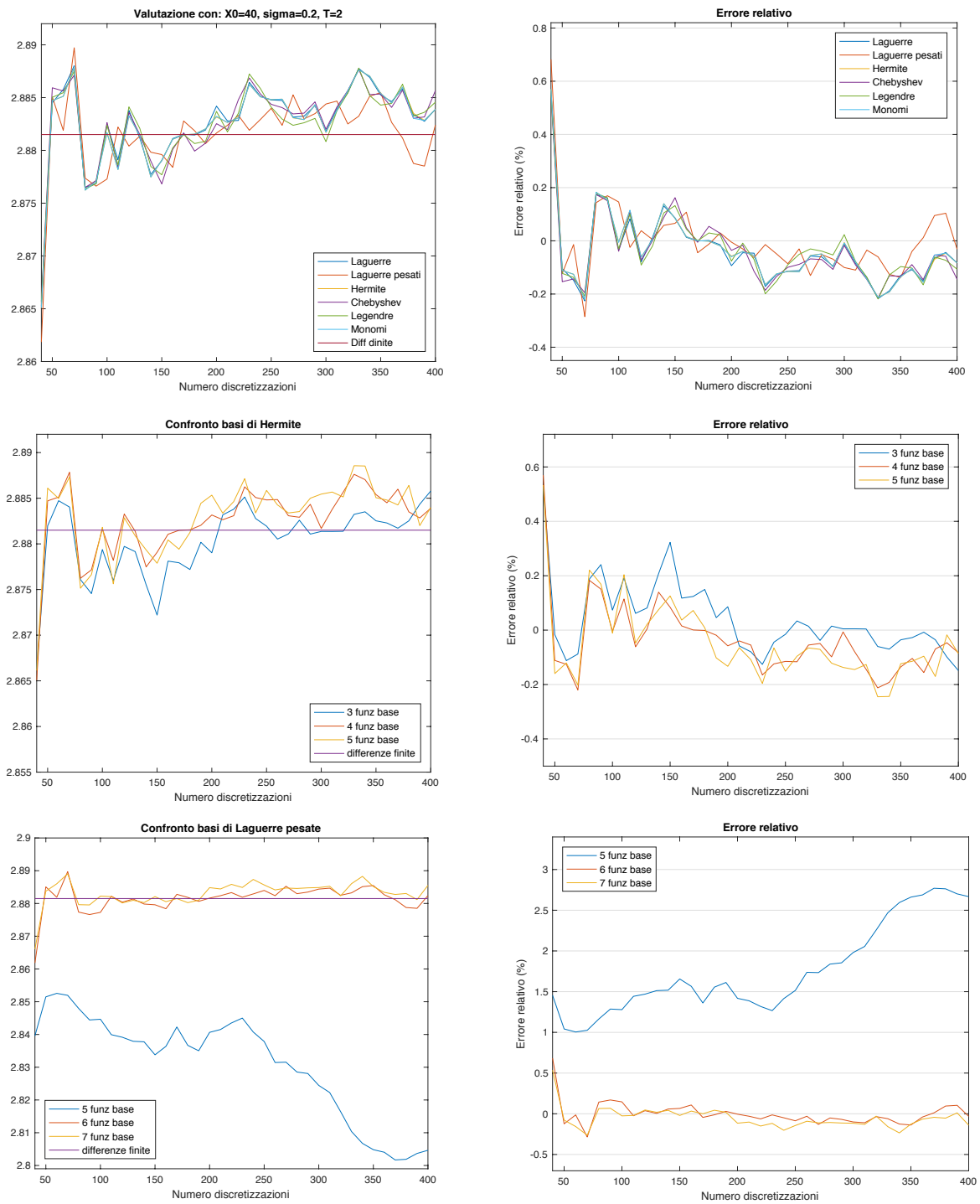


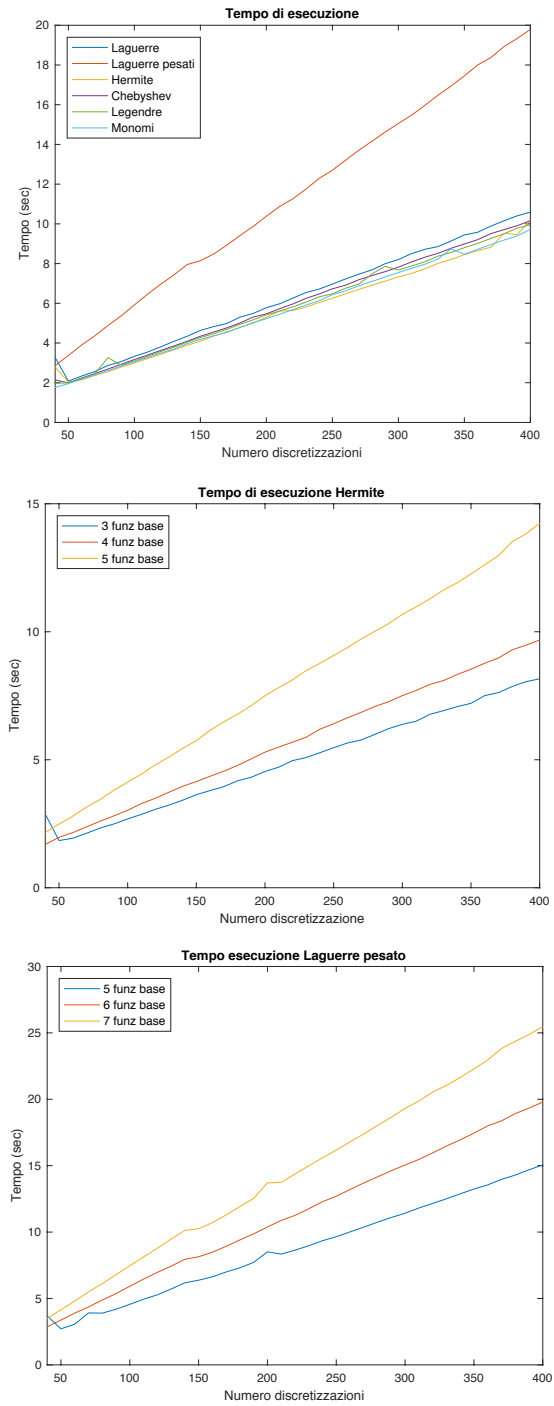
Figura 3.11: Opzione:  $X_0 = 40, r = 0.06, \sigma = 0.2$  e  $T = 2$ 

Figura 3.12: Opzione:  $X_0 = 40, r = 0.06, \sigma = 0.4$  e  $T = 1$

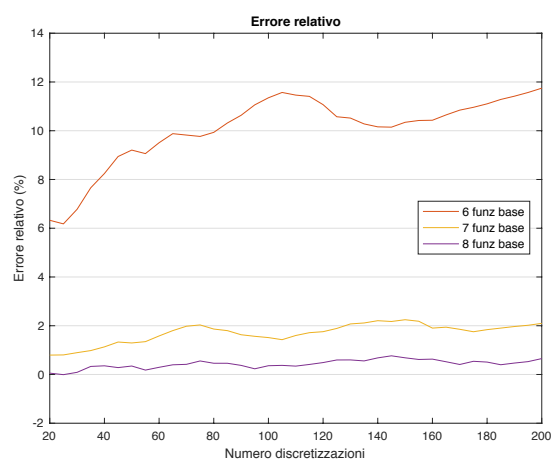
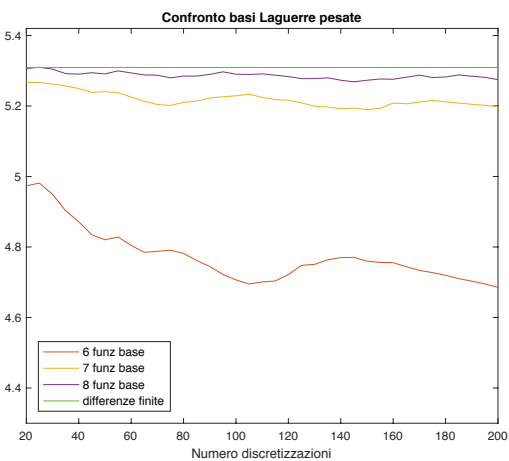
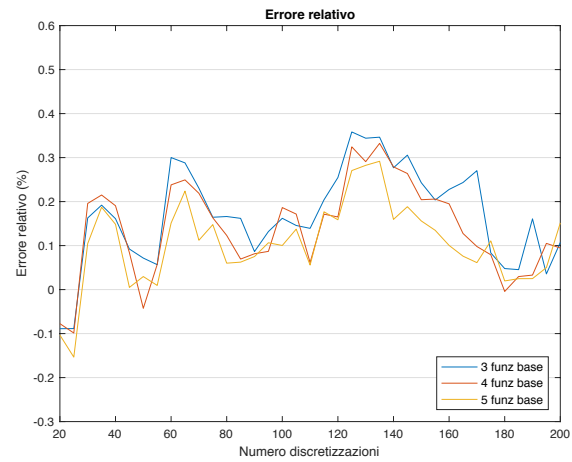
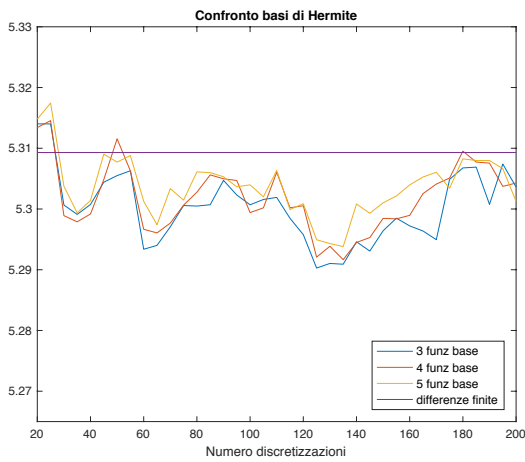
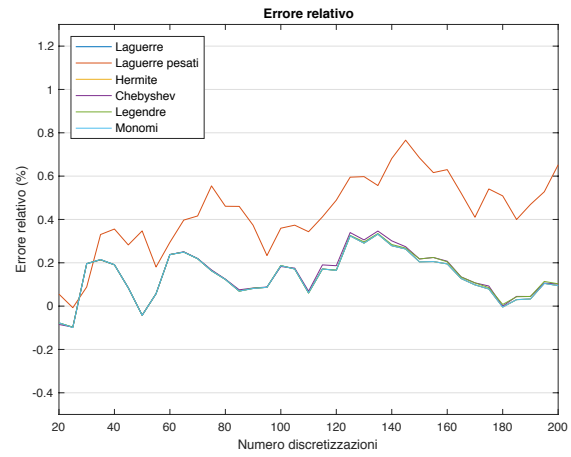
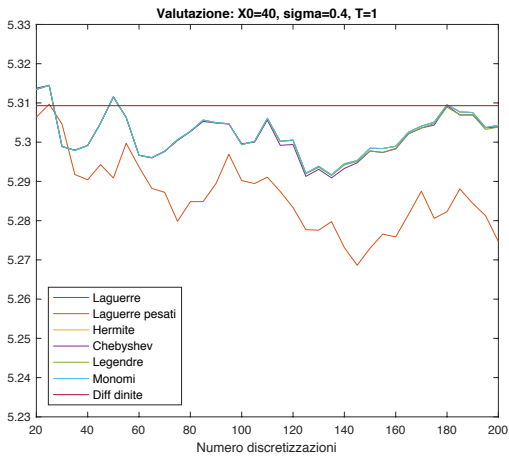


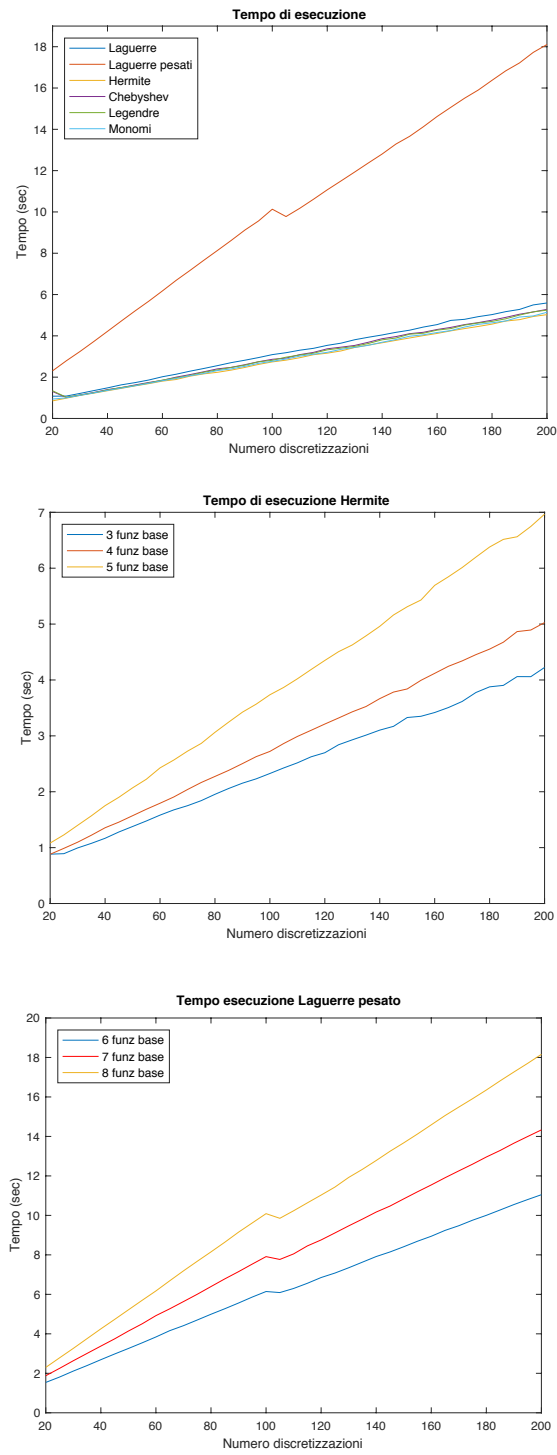
Figura 3.13: Opzione:  $X_0 = 40$ ,  $r = 0.06$ ,  $\sigma = 0.4$  e  $T = 1$ 



Figura 3.14: Opzione:  $X_0 = 44, r = 0.06, \sigma = 0.2$  e  $T = 1$

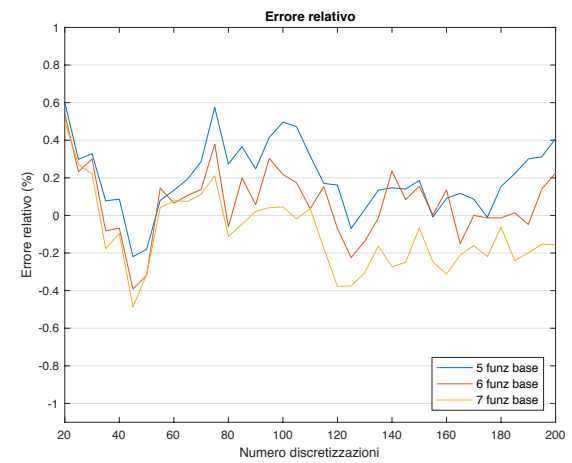
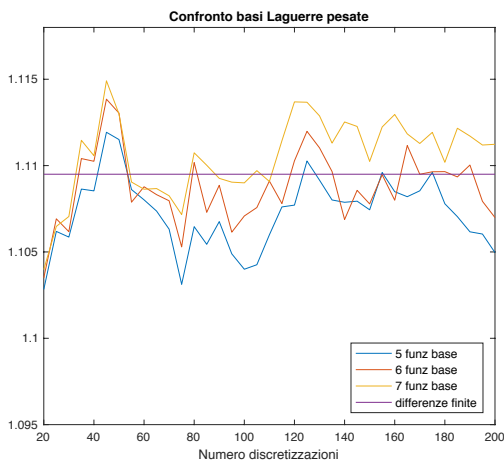
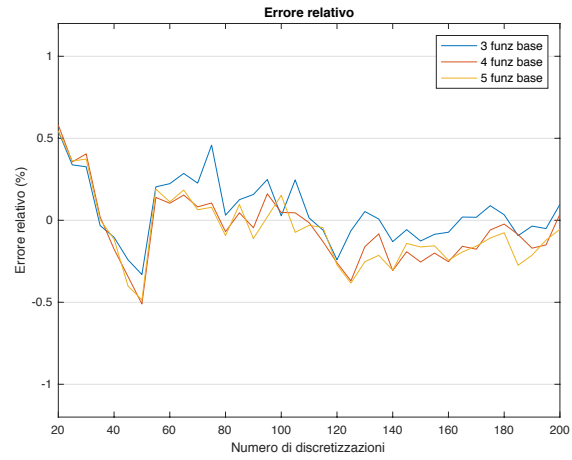
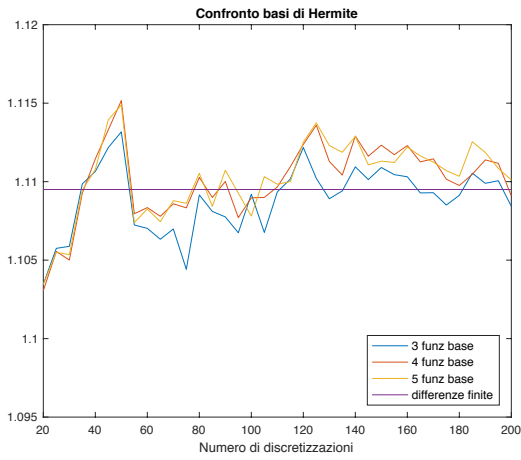
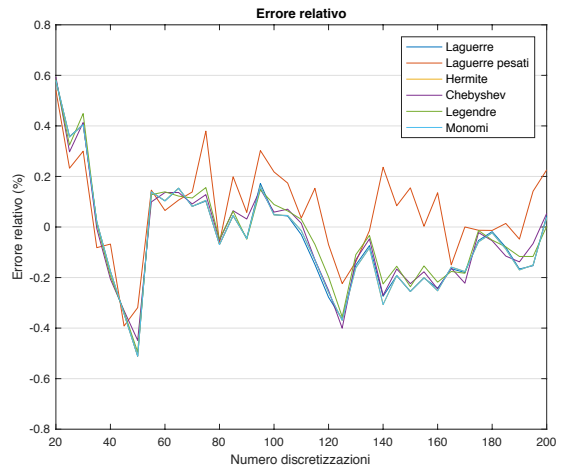
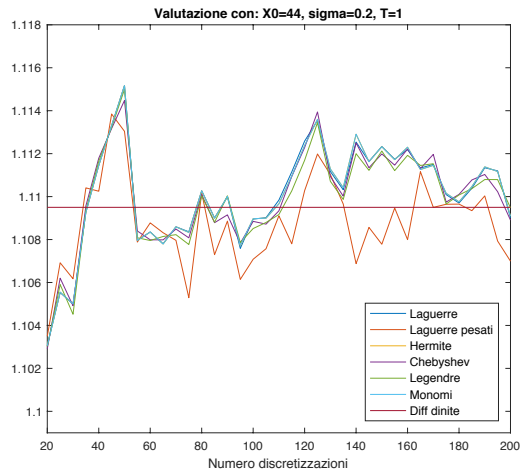


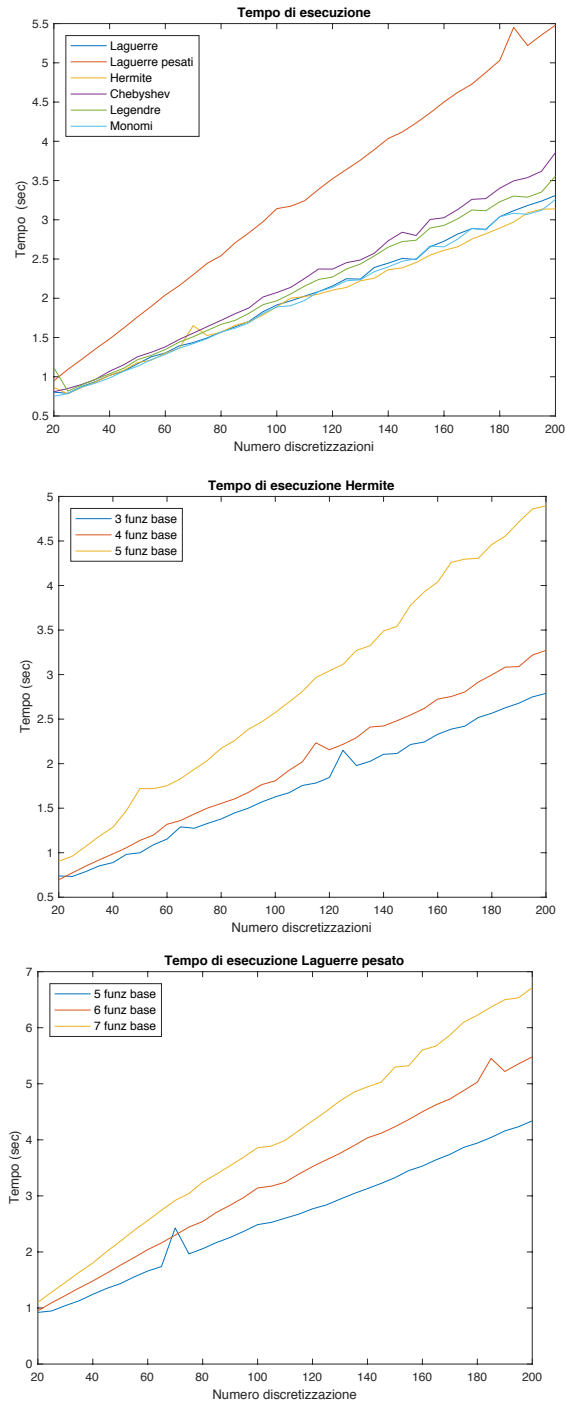
Figura 3.15: Opzione:  $X_0 = 44$ ,  $r = 0.06$ ,  $\sigma = 0.2$  e  $T = 1$ 

Figura 3.16: Opzione:  $X_0 = 44, r = 0.06, \sigma = 0.2$  e  $T = 2$

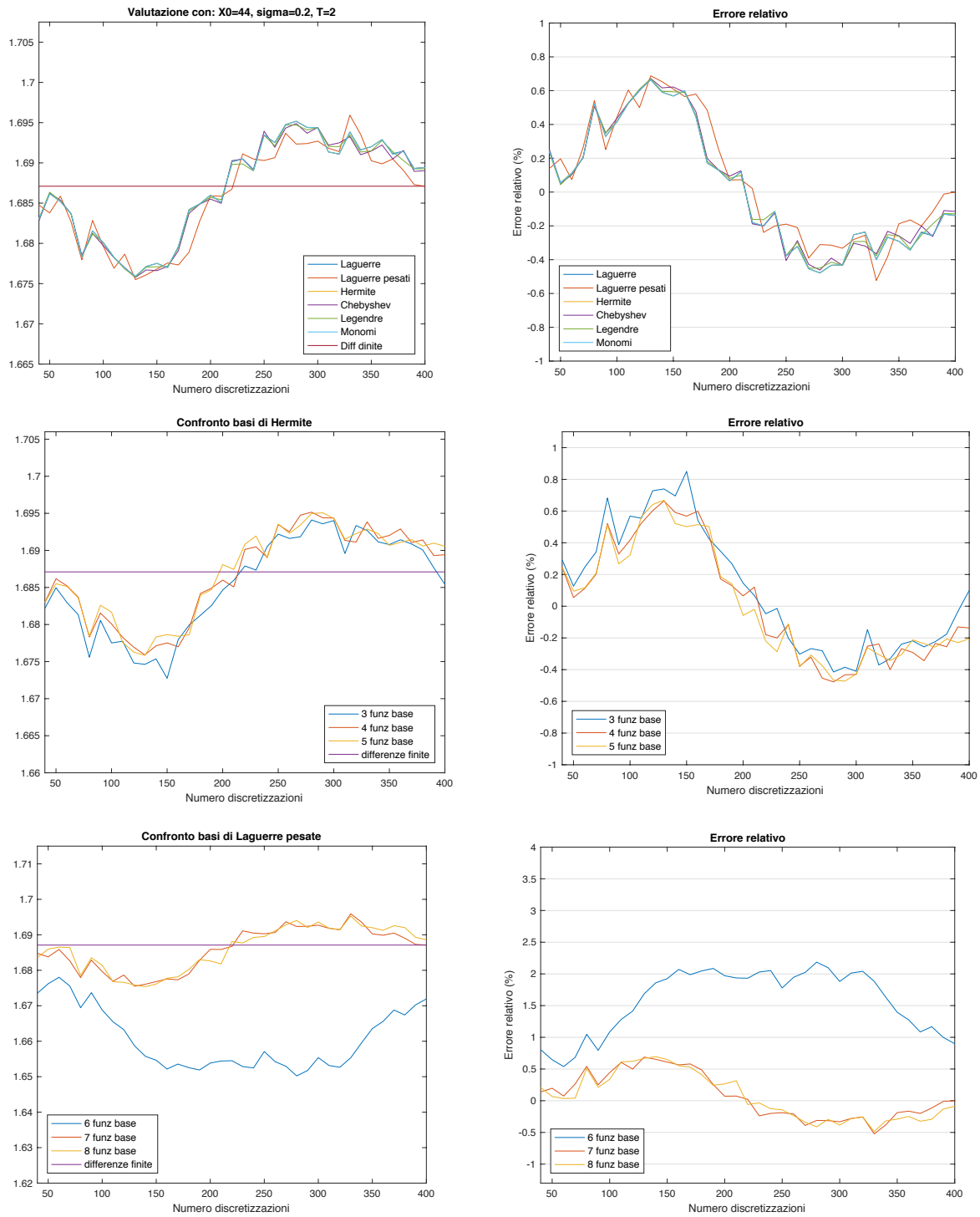


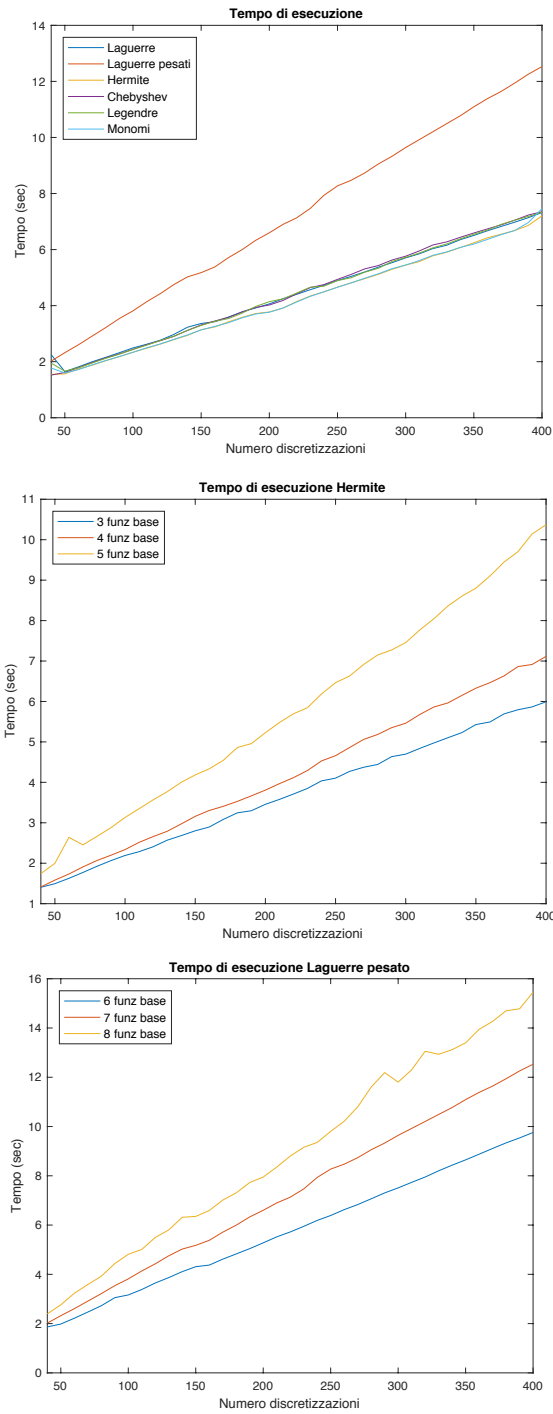
Figura 3.17: Opzione:  $X_0 = 44$ ,  $r = 0.06$ ,  $\sigma = 0.2$  e  $T = 2$ 

Figura 3.18: Opzione:  $X_0 = 44, r = 0.06, \sigma = 0.4$  e  $T = 1$

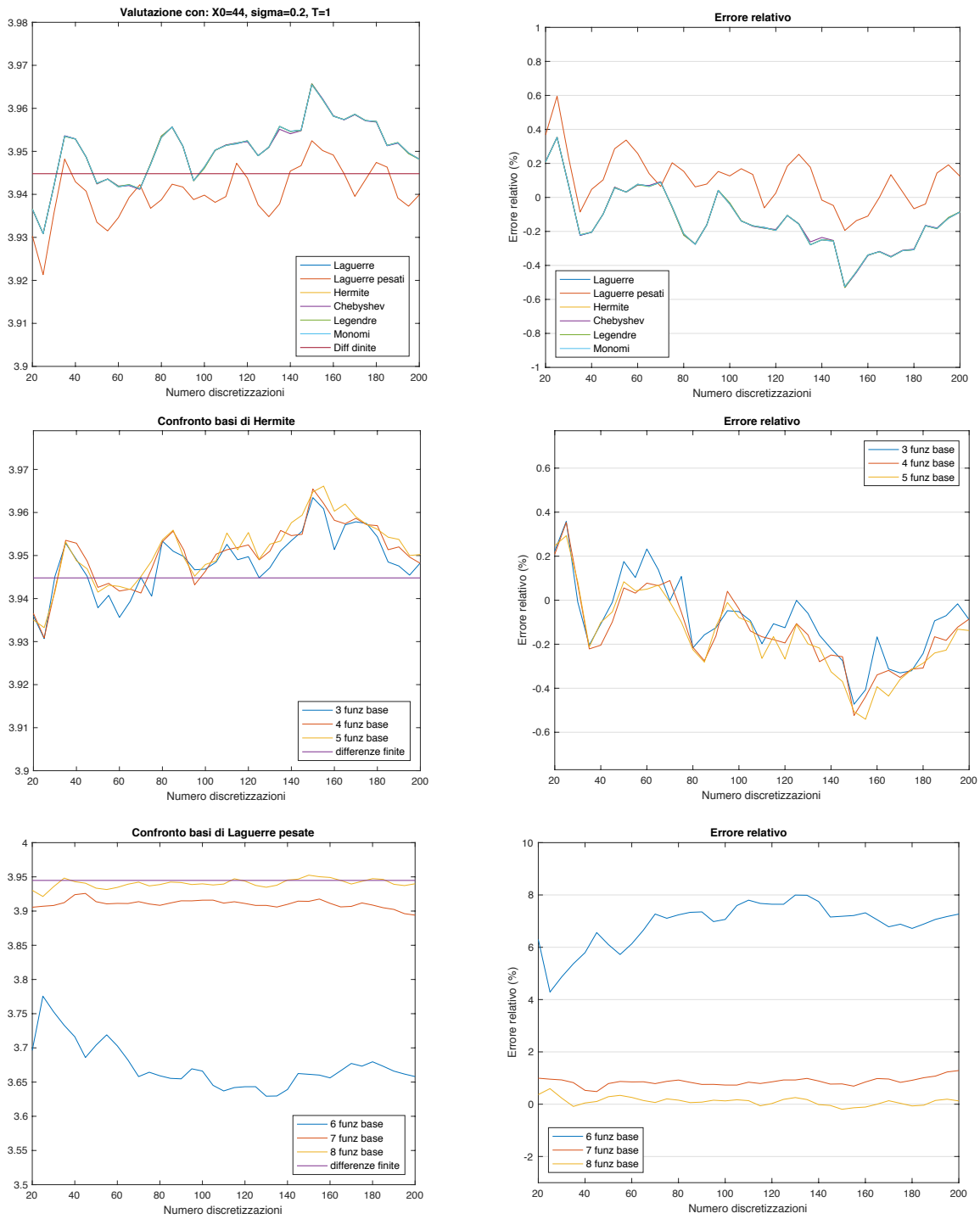
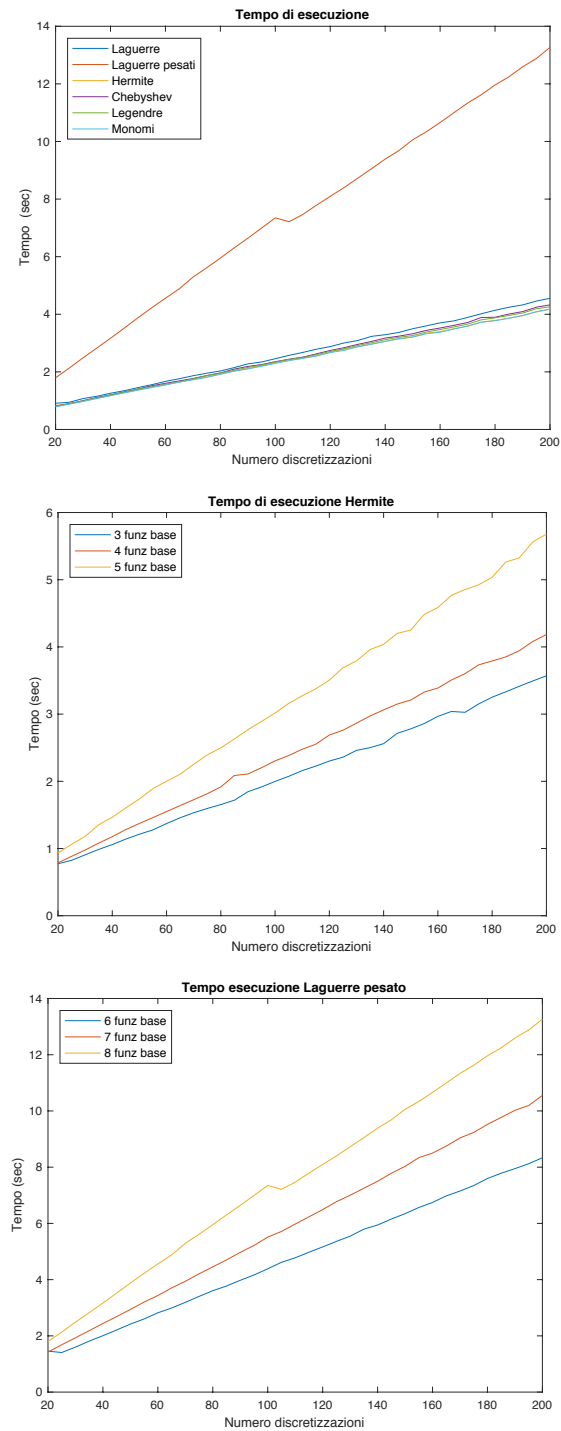


Figura 3.19: Opzione:  $X_0 = 44$ ,  $r = 0.06$ ,  $\sigma = 0.4$  e  $T = 1$ 

## Capitolo 4

# Conclusioni

Attraverso questa tesi abbiamo mostrato i metodi di regressione per la programmazione dinamica stocastica al fine di valutare le opzioni di tipo Americano attraverso la simulazione.

Utilizzando il metodo di Eulero per la modellazione, il metodo Monte Carlo per approssimare i valori attesi, e il metodo di regressione per valutare l'opzione, abbiamo ottenuto una tecnica:

- intuitiva: lo abbiamo visto approfondendo la teoria che è alla base dei metodi utilizzati;
- facile da applicare: in sezione 3.2 abbiamo visto passo passo come si applica l'algoritmo, il quale prevede un limitato numero di input oltre alle caratteristiche dell'opzione e si sviluppa in modo semplice, intuitivo e lineare;
- precisa: lo abbiamo osservato dal confronto dei grafici sull'errore relativo, in sezione 3.4, in quanto l'errore rimane al di sotto dell'1% in valore assoluto in ogni opzione analizzata, inoltre il margine di errore si può ridurre ulteriormente variando il numero delle basi, delle discretizzazioni e delle traiettorie;
- computazionalmente efficiente: possiamo affermarlo grazie ai grafici sui tempi di esecuzione, in sezione 3.4, tenendo presente che le rilevazioni sul tempo sono state fatte da un semplice computer portatile mostrando così che l'implementazione è accessibile a tutti i calcolatori.

Questi metodi sono stati testati su diversi esempi di put Americane con caratteristiche diverse per valore iniziale, volatilità e intervallo di definizione. Tuttavia le tecniche di simulazione viste sono sviluppabili anche per particolari derivati finanziari, come le opzioni callable che prevedono la possibilità di default o i mercati finanziari composti da più sottostanti.





# Appendice A

## Codici Matlab

### A.1 Algoritmo di Longstaff-Schwartz

```
% Algoritmo per la quotazione di un'opzione PUT Americana attraverso il
% metodo di Longstaff e Schwartz.

% Input:
X0=36;      % Prezzo del sottostante al tempo iniziale
r=0.06;     % Tasso di interesse costante
sigma=0.2;  % Volatilità
T=1;        % Tempo di scadenza
K=40;       % Strike
b=100000;   % Numero delle traiettorie simulate del sottostante
m=200;      % Numero degli intervalli di discretizzazione del tempo
M=5;        % Numero delle funzioni base

step=T/m;   % Passo della discretizzazione
t=0:step:T; % Vettore di lunghezza m, delle opportunità d'esercizio

sconto=exp(-r*step); % Fattore di sconto

% X matrice mxb delle traiettorie del moto Browniano geometrico
% X(i,j)= j-ma traiettoria all'i-mo tempo
X=mBgeom(b,m,step,X0,r,sigma);

% Payoff nel nodo terminale di ogni traiettoria: V, vettore di lunghezza m+1
V=max(K-X(m+1,:),0);
```

```

% Metodo di regressione
for i=m:-1:2

    % Consideriamo solo le traiettorie in the money al tempo i-mo
    itm=find(X(i,:)<K);    % Indici delle traiettorie i.t.m.
    N=length(itm);        % Numero delle traiettorie i.t.m.
    Xitm=X(i,itm);        % Valore delle traiettorie i.t.m.

    % Payoff scontati per le traiettorie i.t.m.  Vettore di lunghezza N
    Vitm=V(itm)*sconto;

    % L: matrice MxN delle funzioni di base per le traiettorie i.t.m.
    % L(p,q)= funzione di base di grado p-1 valutata nella traiettoria q-ma
    L=Basis_Laguerre(Xitm,M);

    % Coefficienti di regressione:  vettore colonna di lunghezza M
    beta=L'\ Vitm';

    % Valori di continuazione:  vettore di lunghezza N
    C=beta'*L;

    % Valori di esercizio immediato per le traiettorie i.t.m.
    E=K-Xitm;

    % Confronto:
    ex=itm(C<E);    % Indici delle traiettorie i.t.m.  dove conviene esercitare
    cont=setdiff(1:b,ex);  % Indici delle traiettorie rimanenti

    % Aggiornamento del valore dell'opzione per il tempo i:
    % - per le traiettorie dove si esercita, prendiamo il valore d'esercizio
    % - per le altre traiettorie, prendiamo il valore dell'opzione scontata
    V(ex)=E(C<E);
    V(cont)=V(cont)*sconto;

end

% Metodo Monte Carlo per il prezzo dell'opzione al tempo iniziale
V0=mean(V*sconto);

```

## A.2 Simulazione del moto Browniano geometrico

```
function [S]=mBgeom(b,m,dt,S0,mu,sigma)

% Simuliamo b traiettorie del moto Browniano geometrico S.
% Utilizziamo il metodo delle variabili antitetiche per ridurre la varianza.

% Caratteristiche del moto:
% S0: valore iniziale
% mu: drift
% sigma: volatilità
% dt: passo della discretizzazione del tempo
% m: numero degli intervalli di discretizzazione del tempo
% b: numero delle traiettorie da simulare

% Si ottiene S: matrice mxb
% S(i,j)=j-ma traiettoria all'i-mo tempo

z = randn(m,b/2);
e = exp((mu-0.5*sigma^2)*dt+sigma*sqrt(dt)*[z -z]);
S = cumprod([S0*ones(1,b); e]);

end
```

### A.3 Funzioni base per la regressione

```

function [L]=Basis_Laguerre(S,M)

%Restituisce la matrice MxM dei polinomi di LAGUERRE
%L(p,q)= polinomio di Laguerre di grado p-1 valutato nella traiettoria q-ma

if M==1
    L=[ones(size(S)); (1-S)];
elseif M==2
    L=[ones(size(S)); (1-S); 1/2*(2-4*S+S.^2)];
elseif M==3
    L=[ones(size(S)); (1-S); 1/2*(2-4*S+S.^2); 1/6*(6-18*S+9*S.^2-S.^3) ];
elseif M==4
    L=[ones(size(S)); (1-S); 1/2*(2-4*S+S.^2); 1/6*(6-18*S+9*S.^2-S.^3); ...
        1/24*(24-96*S+72*S.^2-16*S.^3+S.^4)];
elseif M==5
    L=[ones(size(S)); (1-S); 1/2*(2-4*S+S.^2); 1/6*(6-18*S+9*S.^2-S.^3); ...
        1/24*(24-96*S+72*S.^2-16*S.^3+S.^4); ...
        1/120*(120-600*S+600*S.^2-200*S.^3+25*S.^4-S.^5)];
else
    error('Numero massimo di funzioni base: 5');
end
end
end

```

```
function [L]=Basis_Hermite(S,M)

%Restituisce la matrice MxM dei polinomi di HERMITE
%L(p,q)= polinomio di Hermite di grado p-1 valutato nella traiettoria q-ma

if M==1
    L=[ones(size(S)); S];
elseif M==2
    L=[ones(size(S)); S; S.^2-1];
elseif M==3
    L=[ones(size(S)); S; S.^2-1; S.^3-3*S ];
elseif M==4
    L=[ones(size(S)); S; S.^2-1; S.^3-3*S; S.^4-6*S.^2+3];
elseif M==5
    L=[ones(size(S)); S; S.^2; S.^3-3*S; S.^4-6*S.^2+3; S.^5-10*S.^3+15*S];
else
    error('Numero massimo di funzioni base: 5');
end
end
```

```
function [L]=Basis_Chebyshev(S,M)

%Restituisce la matrice MxM dei polinomi di CHEBYSHEV
%L(p,q)= polinomio di Chebyshev di grado p-1 valutato nella traiettoria q-ma

if M==1
    L=[ones(size(S)); S];
elseif M==2
    L=[ones(size(S)); S; 2*(S.^2)-1];
elseif M==3
    L=[ones(size(S)); S; 2*(S.^2)-1; 4*(S.^3)-3*S ];
elseif M==4
    L=[ones(size(S)); S; 2*(S.^2)-1; 4*(S.^3)-3*S; 8*(S.^4)-8*(S.^2)+1 ];
elseif M==5
    L=[ones(size(S)); S; 2*(S.^2)-1; 4*(S.^3)-3*S; 8*(S.^4)-8*(S.^2)+1; ...
        16*(S.^5)-20*(S.^3)+5*S];
else
    error('Numero massimo di funzioni base: 5');
end
end
```

```
function [L]=Basis.Legendre(S,M)

%Restituisce la matrice MxM dei polinomi di LEGENDRE
%L(p,q)= polinomio di Legendre di grado p-1 valutato nella traiettoria q-ma

if M==1
    L=[ones(size(S)); S];
elseif M==2
    L=[ones(size(S)); S; 1.5*(S.^2)-0.5];
elseif M==3
    L=[ones(size(S)); S; 1.5*(S.^2)-0.5; 2.5*(S.^3)-1.5*S ];
elseif M==4
    L=[ones(size(S)); S;1.5*(S.^2)-0.5; 2.5*(S.^3)-1.5*S; ...
        0.125*(34*(S.^4)-30*(S.^2)+3)];
elseif M==5
    L=[ones(size(S)); S; 1.5*(S.^2)-0.5; 2.5*(S.^3)-1.5*S; ...
        0.125*(34*(S.^4)-30*(S.^2)+3); 1/8*(63*S.^5-70*S.^3+15*S)];
else
    error('Numero massimo di funzioni base: 5');
end
end
```

```
function [L]=Basis_monomi(S,M)

%Restituisce la matrice MxM dei polinomi di MONOMI
%L(p,q)= monomio di grado p-1 valutato nella traiettoria q-ma

if M==1
    L=[ones(size(S)); S];
elseif M==2
    L=[ones(size(S)); S; S.^2];
elseif M==3
    L=[ones(size(S)); S; S.^2; S.^3 ];
elseif M==4
    L=[ones(size(S)); S; S.^2; S.^3; S.^4];
elseif M==5
    L=[ones(size(S)); S; S.^2; S.^3; S.^4; S.^5];
else
    error('Numero massimo di funzioni base: 5');
end
end
```



# Bibliografia

- [1] Andrea Pascucci  
*PDE and Martingale Methods in Option Pricing*  
Bocconi and Springer series: Mathematics, Statistic, Finance and Economics, Vol. 2  
Springer, 2011.
- [2] Andrea Pascucci  
*Calcolo stocastico per la finanza*  
Springer, 2008.
- [3] Eric Couffignals  
*Quasi-Monte Carlo Simulations for Longstaff Schwartz Pricing of American Options*  
University of Oxford, 2010
- [4] Francis A. Longstaff, Eduardo S.Schwartz  
*Valuing American Options by Simulation: A Simple Least-Squares Approach*  
in “The Review of Financial Studies”, Vol.14, No. 1, pp. 113-147  
Spring, 2001
- [5] Milton Abramowitz, Irene A. Stegun  
*Handbook of mathematical functions: with formulas, graphs and mathematical tables*  
Government Printing Office, Cap. 22 (1964)
- [6] Karl Sigman  
*Introduction to reducing variance in Monte Carlo simulations*  
Columbia University, 2010
- [7] Paul Glasserman  
*Monte Carlo Methods in Financial Engineering*  
Springer, 2003
- [8] Robert L. Harrison  
*Introduction to Monte Carlo simulation*  
in “AIP Conference Proceedings”, Vol.1204 (2010)