

Politecnico di Milano

FACOLTÀ DI INGEGNERIA
Corso di Laurea Magistrale in Ingegneria Matematica

TESI DI LAUREA MAGISTRALE



**Pricing di opzioni path-dependent
sotto dinamica time-changed Lévy:
approccio mediante trasformata di Hilbert**

Relatore:
Prof. Daniele Marazzina

Candidato:
Diletta Ricci
Matricola 798990

Ringraziamenti

Desidero dapprima ringraziare il Professor Marazzina per avermi incoraggiato e guidato durante lo sviluppo del presente lavoro. Mi ha dato la possibilità di lavorare su una tematica interessante e stimolante. Tante sono le persone che mi sono state vicine in questi anni, ma prima di tutti ci sono loro, mamma e nonni, che mi hanno supportato, riponendo in me un'infinita fiducia, consentendomi di arrivare oggi fin qui. Ho desiderio di ringraziare con affetto anche tutti gli altri miei familiari, per l'interesse assiduo mostrato verso il mio percorso di studi, in particolare mio zio Cesare e mio zio Massimo. Un grazie di cuore alle mie amiche di vita: Gaia, Claudia, Giulia, Francesca, Isotta ed Elisa, che riescono in ogni situazione a mettermi di buon umore e hanno contribuito a rendermi la persona che sono. Un grazie speciale anche ai miei compagni di studio qui al Politecnico, con cui ho condiviso il mio cammino universitario. Non posso, infine, non esprimere tutta la mia gratitudine a Riccardo, il cui incrollabile sostegno è stato fondamentale per permettermi di raggiungere questo importante traguardo.

Milano, Dicembre 2014

D. R.

An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.

John Tukey

Per aspera ad astra.

Abstract

The goal of the thesis is the construction of accurate and efficient numerical algorithms for pricing discretely monitored path-dependent derivatives, in particular barrier, lookback and Bermudan style options, by applying the Fast Hilbert Transform method in the context of Lévy processes. These methods allow us to improve the performance in terms of computational cost obtainable through the use of Fourier Transform. The method relies on the connection between Hilbert and Fourier transforms, exploiting the efficient algorithm for the matrix-vector multiplication if Toeplitz matrices are considered. Besides the Lévy processes, it is possible to extend the Hilbert transform approach to the option pricing in time-changed Lévy framework. In this case the advantages of the method, as the exponential decay of the error with respect to the step size, are retained. Finally, these methods are suitable for credit risk applications. Specifically here the fair value of a firm under the dividend-ruin model with both embedded reflecting barrier, representing dividend distribution, and absorbing barrier, characterizing the default, is computed.

Sommario

L'obiettivo della tesi è quello di costruire algoritmi numerici accurati ed efficienti per prezzare opzioni path-dependent a monitoraggio discreto, in particolare, opzioni barriera, opzioni lookback ed opzioni Bermuda, applicando il metodo della trasformata veloce di Hilbert nel contesto dei processi di Lévy. Questo metodo consente di migliorare le prestazioni in termini di costo computazionale ottenibili mediante il solo utilizzo della trasformata di Fourier. Il metodo si appoggia ai risultati della teoria che lega le trasformate di Hilbert e Fourier, rendendo le discretizzazioni molto semplici e si avvale dell'efficiente algoritmo di Toeplitz per la moltiplicazione matrice vettore. Oltre ai processi di Lévy, è possibile estendere l'approccio con trasformata di Hilbert anche al pricing di opzioni nel contesto dei processi time-changed Lévy. In questo caso, i vantaggi del metodo, come il decadimento esponenziale dell'errore in termini del passo della trasformata, restano validi. Infine, il metodo fornisce un framework per applicazioni al mondo del rischio di credito. Nello specifico si è qui calcolato il valore di un'azienda nel modello dividend-ruin con barriera riflettente, rappresentante l'erogazione dei dividendi, e assorbente, caratterizzante il default.

Indice

1	Nozioni preliminari	1
1.1	Processi stocastici	1
1.1.1	Funzione generatrice dei momenti	3
1.1.2	Funzione caratteristica	4
1.2	Il modello di Black&Scholes	4
1.2.1	Moto Browniano	4
1.2.2	Moto Browniano geometrico	5
1.2.3	Pricing	6
1.2.4	Fatti stilizzati dei mercati finanziari	6
1.3	Modelli con salti	7
1.3.1	Processi di Poisson	7
1.3.2	Misure di salto	10
1.3.3	Costruzione di processi di Lévy	11
1.3.4	Merton	13
1.3.5	Kou	13
1.3.6	VG	14
1.3.7	NIG	14
1.4	I processi time-changed Lévy	15
1.4.1	Leverage nei processi time-changed Lévy	17
1.4.2	Il modello a volatilità stocastica di Heston	18
1.4.3	Probabilità di transizione della varianza	19
1.5	Pricing neutrale rispetto al rischio	24
2	Opzioni Europee e Barriera	27
2.1	Introduzione	28
2.2	Opzioni Europee	30
2.2.1	Procedura di pricing	32
2.2.2	Test numerici	34
2.3	Opzioni Barriera con processi di Lévy	38
2.3.1	Induzione backward nello spazio di stato	38

2.3.2	Trasformata di Hilbert	39
2.3.3	Induzione backward nello spazio di Fourier: rappresentazione con trasformata di Hilbert	41
2.3.4	Approssimazione discreta	42
2.3.5	Computazione: algoritmo FHT	45
2.3.6	Moltiplicazione Matrice-Vettore di Toeplitz via FFT	47
2.3.7	Test numerici	48
2.4	Opzioni Barriera con processi time-changed Lévy	54
2.4.1	Induzione backward	54
2.4.2	Il metodo della trasformata veloce di Hilbert	55
2.4.3	Sintesi dell'algoritmo numerico di pricing	57
2.4.4	Test numerici	58
3	Opzioni Lookback	61
3.1	Introduzione teorica	61
3.1.1	Ricorsione forward nello spazio di stato	62
3.1.2	Ricorsione forward nello spazio di Fourier	63
3.1.3	Trasformata di Hilbert	64
3.2	Floating strike lookback put	64
3.2.1	Test numerici	66
4	Opzioni Bermuda	69
4.1	Opzioni Bermuda con processi di Lévy	69
4.1.1	Induzione backward	69
4.1.2	Approssimazione discreta	71
4.1.3	Implementazione	72
4.1.4	Test numerici	73
4.2	Opzioni Bermuda con processi time-changed Lévy	77
4.2.1	Introduzione	77
4.2.2	Induzione backward	78
4.2.3	Il metodo della trasformata veloce di Hilbert	79
4.2.4	Sintesi dell'algoritmo numerico di pricing	82
4.2.5	Complessità computazionale	82
4.2.6	Test numerici	83
5	Modello dividend-ruin con dinamica time-changed Lévy	85
5.1	Formulazione del modello	85
5.1.1	Le barriere assorbente e riflettente	85
5.1.2	La funzione valore del processo valore d'azienda	87
5.1.3	Condizione di salto della funzione valore	88
5.1.4	Il calcolo tra due date successive di monitoraggio	88
5.2	Costruzione dell'algoritmo FHT	91
5.2.1	Implementazione	93
5.2.2	Sintesi dell'algoritmo numerico di pricing	95
5.2.3	Complessità computazionale	95
5.2.4	Test numerici	96
6	Conclusioni	99

A	Codici	101
A.1	Metodi di Pricing per opzioni Europee basati su FHT	101
A.1.1	PricingEuropean_HT	101
A.2	Metodi di Pricing per opzioni Barriera	102
A.2.1	PricingPDO_HT	102
A.2.2	PricingCUO_HT	104
A.2.3	PricingKO_HT	106
A.2.4	PricingPDO_HT_Heston	108
A.2.5	PricingPDO_MC_Heston	110
A.3	Metodi di Pricing per opzioni Lookback basati su FHT	111
A.3.1	PricingLookbackPut_HT	111
A.4	Metodi di Pricing per opzioni Bermuda basati su FHT	113
A.4.1	PricingBermudanPut_HT	113
A.4.2	PricingBermudanPut_HT_Heston	115
A.5	Metodi di Pricing modello Dividend Ruin	118
A.5.1	PricingDividendRuin_HT_Heston	118
A.5.2	PricingDividendRuin_MC_Heston	121
A.6	Funzioni ausiliarie	122
A.6.1	ProdottoToeplitz	122
A.6.2	P_transizione_log	122
A.6.3	CharF_CIR_Cond	123
A.6.4	Frammento_MatricePsiTilda	123
	Bibliografia	125

Elenco delle figure

1.1	Tasso di decadimento nella coda di sinistra della densità della varianza, quando ν tende a -1	21
1.2	Tasso di decadimento nella coda di sinistra della densità della varianza logaritmica, quando ν tende a -1	22
1.3	Tasso di decadimento in <i>scala logaritmica</i> nella coda di sinistra della densità della varianza logaritmica, quando ν tende a -1	23
2.1	Prezzo di un'opzione europea Call, errore e costi.	36
2.1	Prezzo di un'opzione europea Call errore e costi.	37
2.2	Pricing Up-Out Call, errore e costi.	50
2.2	Pricing Up-Out Call, errore e costi.	51
2.3	Pricing Down-Out Put, errore e costi.	52
2.3	Pricing Down-Out Put, errore e costi.	53
2.4	Prezzo Put Down-Out per diversi livelli di volatilità iniziale	59
3.1	Pricing Lookback Floating Strike Put, errore e costi.	67
3.1	Pricing Lookback Floating Strike Put, errore e costi.	68
4.1	Valore di un'opzione Bermuda put e corrispondente frontiera di esercizio anticipato.	74
4.2	Valore di un'opzione Bermuda put e corrispondente frontiera di esercizio anticipato.	75
4.3	Convergenza e costi computazionali del metodo FHT.	76
4.4	Prezzo opzione Bermuda per diversi livelli di volatilità iniziale	84
5.1	Simulazione del valore aziendale	86
5.2	Andamento del valore aziendale rispetto alla volatilità iniziale	97

Elenco delle tabelle

4.1	Test modello di Heston per opzione Bermuda	84
-----	--	----

Introduzione

Un'opzione discreta *path dependent* è un contratto finanziario il cui payoff dipende dalle realizzazioni del processo sottostante che viene monitorato a tempi discreti durante la vita del derivato. In letteratura, per prezzare tali opzioni, sono stati sviluppati diversi strumenti analitici, a partire da numerosi approcci, quali le equazioni integro-differenziali, il Monte Carlo e le trasformate di Fourier. Il problema di pricing di opzioni path dependent nei modelli di exponential Lévy e time changed Lévy costituisce ancora una sfida matematica e computazionale. Un problema che può sorgere è quello dell'accuratezza di queste formule, infatti la bontà dell'approssimazione può deteriorarsi significativamente per determinate classi di opzioni. In tempi più recenti sono apparsi in letteratura metodi di pricing particolarmente accurati e dall'alta efficienza computazionale, che si basano sulla trasformata di Hilbert.

Nel presente lavoro, vengono analizzati questi metodi, originariamente introdotti in [6–9, 18]; i metodi vengono applicati ad un numero maggiore di dinamiche stocastiche rispetto a quanto fatto negli articoli che li hanno introdotti. I metodi sono edificati sulla *trasformata di Hilbert*, utilizzando i risultati della teoria di approssimazione basata sull'*espansione Sinc* per funzioni analitiche. Grazie ad essi, le trasformate di Hilbert e Fourier possono essere discretizzate tramite regole molto semplici, avvelendosi della moltiplicazione matrice vettore di Toeplitz. Una sequenza di trasformate di Hilbert, in particolare del prodotto della trasformata di Fourier del prezzo per la funzione caratteristica del processo di Lévy, vengono valutate ricorsivamente per ogni step temporale compreso tra due date di monitoraggio. Il presupposto teorico per l'elevata accuratezza del metodo risiede nella potente teoria di approssimazione per le funzioni analitiche in uno strip orizzontale del piano complesso contenente l'asse reale. L'errore di discretizzazione per l'approssimazione della trasformata di Hilbert con passo h è $O(\exp(-c_1/h))$, per un certo $c_1 > 0$, ovvero decade esponenzialmente come $1/h$. Per un passo fissato h l'errore di troncamento della serie infinita risultante dalla discretizzazione della trasformata di Hilbert è $O(M^{1-\nu}h^{-\nu} \exp(-c_2(Mh)^\nu))$, dove M è il livello di troncamento, con $c_2 > 0$ e $\nu \in (0, 2]$ per una classe vasta di processi Lévy. Selezionando opportunamente

$h = h(M)$ l'errore totale diventa $O(M^{1/(1+\nu)} \exp(-c_3 M^{\nu/(1+\nu)}))$, per $c_3 > 0$, che decade esponenzialmente. Inoltre la risultante approssimazione discreta può essere implementata in maniera efficiente utilizzando la moltiplicazione matrice-vettore di Toeplitz, algoritmo basato sulla trasformata di Fourier con costo computazionale pari a $O(NM \log M)$, dove N è il numero di date di monitoraggio.

Le opzioni barriera e le opzioni lookback sono le due più importanti classi di opzioni path dependent, il cui payoff dipende dal massimo o dal minimo raggiunto dal sottostante in un periodo di tempo fissato. In pratica, il massimo e il minimo sono tipicamente osservati a un insieme discreto di istanti di monitoraggio. Ai fini del pricing di opzioni barriera discrete, si inizia considerando il payoff dell'opzione a scadenza T e si segue un algoritmo di *induzione backward* tra le N date di monitoraggio fino al tempo di valutazione t_0 . Ad ogni step si calcola il valore atteso dell'opzione ottenuto dallo step precedente e lo si moltiplica per la funzione indicatrice che caratterizza la barriera. Implementando queste operazioni nello spazio di Fourier e sfruttando le relazioni tra trasformata di Hilbert, trasformata di Fourier e funzioni indicatrici il tutto si riduce alla computazione in sequenza di trasformate di Hilbert.

Il lavoro di tesi si articola nel seguente modo:

Il Capitolo 1 fornisce gli strumenti matematici necessari alla comprensione del presente lavoro. Vengono ripassati i concetti di processo stocastico e le relative proprietà. La trattazione procede con il modello di Black&Scholes che ha avuto un notevole successo applicativo per la sua semplicità, nonostante abbia alla base assunzioni poco realistiche. Vengono poi introdotti i modelli con salti più comuni per la simulazione delle traiettorie del sottostante: i processi di Lévy ad attività finita di Merton e Kou e i processi ad attività infinita Variance Gamma e Normal Inverse Gaussian. In seguito vengono spiegati i processi time-changed Lévy, a tal fine viene analizzato il processo CIR, utilizzato come clock per determinare il business time del cambio di scala temporale. La scelta di questi modelli introduce alcune complicazioni, in particolare riguardo alla funzione densità della probabilità di transizione dell'activity rate che tende a divergere. Una possibile soluzione è quella di considerare la log-activity rate, come suggerito in [6], i miglioramenti legati a questa scelta vengono rilevati con delle prove numeriche. All'interno dei processi time-changed Lévy viene incasellato il modello di Heston, processo a volatilità stocastica, che verrà scelto come rappresentate della classe nelle trattazioni successive. Infine viene richiamata la nozione di risk neutral pricing, necessaria per la valutazione dei contratti finanziari.

Il Capitolo 2 presenta la tecnica di pricing basata sulla trasformata di Hilbert che sarà utilizzata nel corso della tesi. Viene preso in considerazione il caso in cui il sottostante segue un andamento descritto da un processo di Lévy, in particolare vengono considerate le dinamiche di Black&Scholes, Merton, Kou, VG e NIG. In primo luogo, si inizia con il pricing delle opzioni di tipo Europeo, in questo caso nessun tipo di induzione backward si rende necessario, viene però già introdotto il framework utilizzato per le opzioni path dependent. Si passa poi alla valutazione di opzioni di tipo Barriera: viene effettuata

un'introduzione teorica e viene presentato l'algoritmo di induzione backward nello spazio di Fourier. In questo ambito viene anche descritto l'algoritmo di moltiplicazione di una matrice Toeplitz per un vettore, con approccio FFT¹. Il capitolo comprende anche numerosi test numerici al variare del tipo di opzione e della dinamica del sottostante. Anche il caso dei processi a volatilità stocastica viene affrontato, quale generalizzazione dei processi di Lévy.

Il Capitolo 3 utilizza il metodo FHT² per il calcolo del momento esponenziale del massimo di un processo di Lévy, osservato a tempi discreti. Tale metodo viene poi applicato per la valutazione di un'opzione lookback floating strike. Il sottostante evolve qui con le stesse dinamiche di Lévy del capitolo precedente.

Il Capitolo 4 tratta il pricing di opzioni Bermuda che corrisponde a un problema di controllo ottimo stocastico nel discreto. Il problema viene implementato sempre tramite l'induzione backward dove ad ogni istante si prende il massimo tra il valore atteso condizionato, rappresentante il valore di continuazione dell'opzione, e il payoff dell'opzione che rappresenta il profitto derivante dall'immediato esercizio. In primo luogo, vengono qui considerate le stesse dinamiche di Lévy del capitolo precedente, in seguito però si considera l'estensione ai processi di tipo time-changed Lévy. In particolare, il metodo viene applicato per la valutazione di opzioni Bermuda sotto la dinamica di Heston, che può essere incorporato nella classe di questi processi.

Il Capitolo 5 applica il metodo basato sulla trasformata di Hilbert alla valutazione di un'azienda con modello dividend-ruin con barriera upper riflettente, oltre la quale avviene l'erogazione dei dividendi, e barriera lower assorbente rappresentante il default. Anche in questo caso si assume che l'azienda evolva con dinamica time-changed Lévy, in particolare il modello di Heston. L'idea di questo capitolo è quella di mostrare la versatilità del metodo, applicandolo ad un diverso tipo di payoff, con un complesso andamento del sottostante. A tutti gli effetti questo capitolo può essere considerato un'estensione del precedente, abbracciando una condizione più generale del payoff.

L'appendice A riporta i codici utilizzati per la computazione. Vengono indicate le funzioni utilizzate per l'effettivo pricing dei derivati e le funzioni ausiliarie da esse impiegate, quali ad esempio il prodotto Toeplitz e le probabilità di transizione del CIR clock.

¹Fast Fourier Transform

²Fast Hilbert Transform

Capitolo 1

Nozioni preliminari

In questo capitolo vengono presentati gli strumenti matematici che serviranno ad effettuare il pricing di diversi contratti finanziari. In particolare, vengono esposti i concetti base al fine di simulare le traiettorie del sottostante, che può essere un titolo finanziario, un indice economico, una commodity o una valuta, dal quale dipenderà l'andamento del prezzo del contratto derivato stipulato su di esso.

1.1 Processi stocastici

Prima di enunciare la nozione fondamentale di processo stocastico è necessario introdurre alcune definizioni di teoria della misura.

Definizione 1.1.1 (σ -algebra). Sia Ω un insieme astratto e sia \mathcal{F} una famiglia di sottoinsiemi delle parti di Ω , $\mathcal{F} \subseteq \mathcal{P}(\Omega)$, si dice che \mathcal{F} è una σ -algebra se verifica le seguenti proprietà:

- $\Omega \in \mathcal{F}$;
- \mathcal{F} è stabile rispetto a complementazione: se $A \in \mathcal{F} \implies A^c = \Omega - A \in \mathcal{F}$;
- \mathcal{F} è stabile rispetto a unione numerabile: $A_1, \dots, A_n \in \mathcal{F} \quad \forall n \in \mathbb{N} \implies A_1 \cup \dots \cup A_n \in \mathcal{F}$.

L'insieme Ω rappresenta tutti i possibili stati del mondo, cioè i vari andamenti che il sottostante può avere nei diversi scenari di mercato, mentre $A \in \mathcal{F}$ è un evento che può accadere a cui si associa una probabilità. La coppia (Ω, \mathcal{F}) viene chiamata spazio misurabile.

Definizione 1.1.2 (Spazio di Probabilità). Si definisce spazio di probabilità la terna $(\Omega, \mathcal{F}, \mathbb{P})$ dove

- Ω è lo spazio campionario,
- \mathcal{F} è una σ -algebra su Ω ,

- $\mathbb{P} : \mathcal{F} \mapsto [0, 1]$ è una probabilità, $\mathbb{P}(\Omega) = 1$.

Definizione 1.1.3 (Variabile aleatoria). Sia $(\Omega, \mathcal{F}, \mathbb{P})$ spazio di probabilità ed (E, \mathcal{E}) spazio misurabile, una variabile aleatoria è una funzione $X : \Omega \mapsto E$ misurabile:

$$X^{-1}(A) \in \mathcal{F} \quad \forall A \in \mathcal{E}. \quad (1.1)$$

Se $E = \mathbb{R}$ e $\mathcal{E} = \mathcal{B}(\mathbb{R})$ (σ -algebra di Borel) si dice che X è una variabile aleatoria reale. Grazie alla condizione (1.1) è possibile associare delle probabilità, usando \mathbb{P} , agli eventi della forma $\{X \in A\}$ per un qualche $A \in \mathcal{E}$, attraverso la controimmagine di X .

Definizione 1.1.4 (Distribuzione di una variabile aleatoria). Sia X una variabile aleatoria definita su $(\Omega, \mathcal{F}, \mathbb{P})$ a valori in (E, \mathcal{E}) . La sua distribuzione è la misura di probabilità p_X su (E, \mathcal{E}) definita come

$$p_X(A) = \mathbb{P}(X^{-1}(A)) \quad \forall A \in \mathcal{E}. \quad (1.2)$$

In seguito, si lavorerà con variabili aleatorie reali, cioè a valori in $(\mathbb{R}; \mathcal{B}(\mathbb{R}))$.

Definizione 1.1.5 (Valore atteso). Data una variabile aleatoria reale X definita su $(\Omega, \mathcal{F}, \mathbb{P})$, il suo valore atteso è dato da

$$\mathbb{E}(X) = \int_{\Omega} X(\omega)P(d\omega) = \int_{\mathbb{R}} xp_X(dx). \quad (1.3)$$

Si osserva che data una variabile aleatoria reale X e una funzione Borel misurabile $f : \mathbb{R} \mapsto \mathbb{R}$, è sempre possibile definire il valore atteso della variabile aleatoria $f(X)$,

$$\mathbb{E}(f(X)) = \int_{\Omega} f(X(\omega))P(d\omega) = \int_{\mathbb{R}} f(x)p_X(dx). \quad (1.4)$$

Ora si può definire la nozione di processo stocastico.

Definizione 1.1.6 (Processo Stocastico). Si definisce processo stocastico $X = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \in T}, (X_t)_{t \in T}, \mathbb{P})$ tale che:

- $(\Omega, \mathcal{F}, \mathbb{P})$ sia uno spazio di probabilità.
- $T \subseteq \mathbb{R}^+$ sia l'insieme dei tempi considerati.
- $(\mathcal{F}_t)_{t \in T}$ sia una filtrazione, ovvero una famiglia di σ -algebre crescente di sotto σ -algebre di \mathcal{F} : $\forall s, t \in T : s \leq t, \mathcal{F}_s \subseteq \mathcal{F}_t \subseteq \mathcal{F}$.
- $X_t : (\Omega, \mathcal{F}) \mapsto (E, \mathcal{E})$ sia una famiglia di variabili aleatorie a valori nello spazio misurabile (E, \mathcal{E}) , adattata alla filtrazione \mathcal{F}_t : $(X_t)_t$ misurabile rispetto a $\mathcal{F}_t \iff \mathcal{G}_t = \sigma(X_s | 0 \leq s \leq t) \subseteq \mathcal{F}_t \forall t \in T$, dove \mathcal{G}_t è la filtrazione naturale generata dal processo.
- \mathbb{P} sia una probabilità su (Ω, \mathcal{F}) .

$(\mathcal{F}_t)_{t \in T}$ rappresenta l'informazione disponibile al tempo t , dunque con il trascorrere del tempo la σ -algebra associata ad esso è sempre più grande: vi è una direzione privilegiata, senza perdita di memoria. Nel seguito si farà riferimento ad un processo stocastico indicandolo con $(X_t)_t$, omettendo gli altri argomenti.

In teoria delle probabilità un ruolo importante è ricoperto dai processi martingale.

Definizione 1.1.7 (Martingala). Un processo stocastico $M = (M_t)_{t \geq 0}$ a valori in \mathbb{R} è una martingala se:

- $M_t \in \mathbb{L}^1(\Omega, \mathcal{F}, \mathbb{P}) \quad \forall t \in T$,
- $\mathbb{E}[M_t | \mathcal{F}_s] = M_s \quad \forall 0 \leq s \leq t$.

Nello studio dei processi stocastici anche la nozione di tempo d'arresto ha notevole rilevanza.

Definizione 1.1.8 (Tempo d'arresto). Sia \mathcal{F}_t una filtrazione. Una variabile aleatoria $\tau : \Omega \mapsto T \cap \{0, \infty\}$ si dice tempo d'arresto se, per ogni $t \in T$, $\{\tau \leq t\} \in \mathcal{F}_t$. Si pone inoltre

$$\mathcal{F}_\tau = \{A \in \mathcal{F}_\infty, A \cap \{\tau \leq t\} \in \mathcal{F}_t \quad \forall t \in T\}, \quad (1.5)$$

dove $\mathcal{F}_\infty = \bigcup_t \mathcal{F}_t$.

La proposizione seguente riunisce alcune proprietà elementari dei tempi di arresto.

Proposizione 1.1. *Siano σ e τ tempi d'arresto. Allora*

- τ è \mathcal{F}_τ -misurabile.
- $\sigma \vee \tau, \sigma \wedge \tau$ sono tempi d'arresto.
- Se $\sigma \leq \tau$, $\mathcal{F}_\sigma \subset \mathcal{F}_\tau$.
- $\mathcal{F}_{\sigma \wedge \tau} = \mathcal{F}_\sigma \cap \mathcal{F}_\tau$.

1.1.1 Funzione generatrice dei momenti

Come punto di partenza si introduce la definizione di momento n -esimo di una variabile aleatoria.

Definizione 1.1.9 (Momento di ordine n). Sia X una variabile aleatoria con distribuzione f_X , si definisce momento di ordine n la quantità

$$m_n(X) = \mathbb{E}[X^n] = \int_{\mathbb{R}} x^n f_X(x) dx,$$

se l'integrale è ben definito.

I momenti possono essere anche calcolati tramite la funzione generatrice dei momenti

$$M_X(z) = \int_{\mathbb{R}} e^{zx} f_X(x) dx,$$

cioè $M_X(z)$ è la trasformata di Laplace della densità $f_X(x)$. Una volta nota la funzione generatrice dei momenti è possibile ricavare i momenti semplicemente derivando:

$$m_n(X) = \frac{d^n}{dz^n} M_X(z)|_{z=0}. \quad (1.6)$$

1.1.2 Funzione caratteristica

Un altro importante concetto è quello di funzione caratteristica del processo:

$$\Phi_X(z) = \mathbb{E}[e^{izX}] = \int_{\mathbb{R}} e^{izx} f_X(x) dx = M_X(iz).$$

$\Phi_X(z) : \mathbb{R} \mapsto \mathbb{C}$ non è altro che la trasformata di Fourier della densità $f_X(x)$ e caratterizza completamente la legge della variabile.

Teorema 1.1. *Date due variabili aleatorie X e Y si ha che:
 X e Y hanno la stessa distribuzione di probabilità $\iff \Phi_X(z) \equiv \Phi_Y(z)$.*

Grazie a questo teorema sarà possibile effettuare il pricing nello spazio di Fourier, con notevoli vantaggi computazionali.

Altre proprietà elementari di Φ_X sono:

- $|\Phi_X(z)| \leq 1 \quad \forall z \in \mathbb{R}$,
- $\Phi_X(0) = 1$,
- $\Phi_X(-z) = \overline{\Phi_X(z)} \quad \forall z \in \mathbb{R}$,
- Se $\mathbb{E}[|X|^n] \leq \infty$ per qualche $n \in \mathbb{N}$ allora

$$\mathbb{E}[X^n] = i^{-n} \frac{\partial^n}{\partial z^n} \Phi_X(z)|_{z=0}. \quad (1.7)$$

Inoltre se Φ_X è continua in zero allora si può scrivere

$$\Phi_X(z) = e^{\Psi_X(z)},$$

dove $\Psi_X(z)$ è l'esponente caratteristico di X . A seconda del tipo di processo si avrà maggiore facilità di calcolo con l'una o con l'altra rappresentazione.

1.2 Il modello di Black&Scholes

1.2.1 Moto Browniano

In questa sezione si vuole studiare il moto Browniano che è il processo più comune per la simulazione delle fluttuazioni del prezzo di mercato di un titolo.

Definizione 1.2.1 (Moto Browniano). $W = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \geq 0}, (W_t)_{t \geq 0}, \mathbb{P})$ processo stocastico è moto Browniano se:

1. $W_0 = 0$ q.c.
2. $W_t - W_s \perp \mathcal{F}_s \forall 0 \leq s \leq t$
3. $W_t - W_s \sim N(0, t - s)$

Dunque un moto Browniano W_t è caratterizzato da incrementi indipendenti, stazionari e con legge gaussiana. Si può dimostrare inoltre che il moto Browniano è una martingala.

Si ricava facilmente l'esponente caratteristico del moto Browniano, pari a

$$\Psi_t(u) = -\frac{\sigma^2 u^2}{2} \quad (1.8)$$

Aggiungendo al moto Browniano standard il termine di drift μt si ha il moto Browniano con drift:

$$X_t = \mu t + \sigma W_t,$$

il cui esponente caratteristico è

$$\Psi_t(u) = i\mu u - \frac{\sigma^2 u^2}{2}. \quad (1.9)$$

Vi sono inoltre altre proprietà principali che riguardano la regolarità delle sue traiettorie:

- le traiettorie del moto Browniano $t \mapsto W_t(\omega)$ sono q.c. non differenziabili;
- il moto Browniano è continuo e γ -holderiano per ogni $0 < \gamma < \frac{1}{2}$;
- le traiettorie di un moto Browniano $t \mapsto W_t(\omega)$ non sono a variazione finita su alcun intervallo di tempo q.c..

1.2.2 Moto Browniano geometrico

Si presenta ora il modello di Black&Scholes che utilizza il moto Browniano geometrico come elemento costitutivo per la modellizzazione dei prezzi dei titoli. La sua dinamica è espressa dalla seguente equazione differenziale stocastica:

$$\begin{cases} dS_t = \mu S_t dt + \sigma S_t dW_t \\ S_0 = s_0 \end{cases} \quad (1.10)$$

dove μ è il termine di *drift* e σ è il termine di *diffusione* del processo. Per piccoli valori di σ , le traiettorie rimarranno vicine al valore atteso, mentre per alti valori di σ ci saranno fluttuazioni più marcate. Soluzione di (1.10) è

$$S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}. \quad (1.11)$$

Questo modello di mercato ha avuto un notevole successo applicativo per la sua semplicità e per il fatto che consente di ottenere formule esplicite di valutazione di molti titoli derivati.

1.2.3 Pricing

Nell'ambito del modello di Black&Scholes è possibile ottenere in forma chiusa l'espressione del prezzo di un'opzione europea. Nel caso di opzione Call la formula è la seguente:

$$C(S_t, K, r, \sigma) = S_t N(d_1) - e^{-r(T-t)} K N(d_2), \quad (1.12)$$

dove

$$d_1 = \frac{\ln(\frac{S_t}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}},$$

$$d_2 = d_1 - \sigma\sqrt{T-t}.$$

Per una sua dimostrazione si faccia riferimento a [3].

1.2.4 Fatti stilizzati dei mercati finanziari

Nei mercati finanziari si riscontrano delle proprietà statistiche che evidenziano i limiti del modello di Black&Scholes, infatti analizzando le serie storiche finanziarie emergono delle regolarità empiriche che vengono chiamate *fatti stilizzati*. Per la trattazione si fa riferimento a [2, 5].

1. **Code grasse** La densità dei rendimenti presenta code più spesse rispetto a quelle della distribuzione Normale, con un eccesso positivo di curtosi.
2. **Volatility clustering** Fasi di elevata (moderata) volatilità tendono a confermarsi nel tempo. Sebbene i rendimenti sembrano incorrelati nel tempo i loro quadrati mostrano una correlazione positiva. Essi non sono quindi indipendentemente distribuiti nel tempo, cioè la volatilità stessa è variabile e presenta effetti di eteroschedasticità.
3. **Effetto leva** La volatilità e il rendimento del sottostante sono negativamente correlati: quando il prezzo di un'azione cade la rischiosità percepita cresce. Da qui deriva l'effetto leverage che nel modellare la varianza condizionata richiede un trattamento differente delle innovazioni a seconda del segno.
4. **Smile di volatilità** Se si ricava la volatilità del sottostante tramite la formula di Black&Scholes partendo dai prezzi delle opzioni europee quotate sul mercato si ottiene la *volatilità implicita*, ovvero si ha che $\sigma = \sigma(K, T)$.
5. **Non continuità** Le traiettorie degli asset finanziari non sono continue: presentano infatti dei salti.

6. Asimmetria I rendimenti negativi elevati in valore assoluto si manifestano più frequentemente rispetto a quanto osservato per i rendimenti positivi: si notano grandi salti negativi nei prezzi delle azioni.

Purtroppo non è possibile costruire un processo stocastico che soddisfi nello stesso tempo tutte le proprietà sopra elencate. Diverse classi di modelli sono state comunemente sviluppate, come i modelli con salti e quelli a volatilità stocastica, che riproducono in maniera più realistica i rendimenti.

1.3 Modelli con salti

Generalizzazione del modello di Black&Scholes sono i processi di Lévy, trattati ad esempio in [5]. Anche lo stesso processo di Wiener è un processo di Lévy: esso è l'unico Lévy a traiettorie continue. I processi di Lévy introducono nella dinamica del sottostante i salti in modo tale da inspessire le code della distribuzione dei log-rendimenti, portandole ad assumere una forma leptocurtica. Si osserva inoltre che i processi di Lévy riescono a riprodurre lo smile di volatilità soprattutto per brevi scadenze. Lo svantaggio principale risiede, invece, nel fatto che non riescono a catturare gli effetti di volatility clustering, obiettivo raggiunto dai modelli a volatilità stocastica.

1.3.1 Processi di Poisson

Il processo di Poisson è un esempio fondamentale di processo stocastico con traiettorie discontinue e viene utilizzato come mattone per la costruzione di processi più complessi con salti. Esso è un caso particolare di processo di conteggio

$$N_t = \sum_{n=1}^{\infty} \mathbb{I}_{\{T_n \leq t\}} = \#\{n \geq 1, T_n \leq t\}, \quad (1.13)$$

dove i $T_n, n \in \mathbb{N}$, sono una successione crescente di tempi casuali:

$$T_n : \Omega \rightarrow \mathbb{R}^+$$

e

$$\omega \mapsto T_n(\omega),$$

che conta il numero di tempi casuali (T_n) compresi tra 0 e t .

Infatti nella definizione di processo di Poisson le variabili T_n sono costruite, invece, come sommatorie parziali di una successione di variabili aleatorie esponenziali i.i.d.¹.

¹indipendenti identicamente distribuite

Variabile aleatoria esponenziale

Una variabile aleatoria τ segue una distribuzione esponenziale di parametro $\lambda > 0$ se ha una densità di probabilità della forma

$$f_\tau(t) = \lambda e^{-\lambda t} \mathbb{I}_{t \geq 0}.$$

Variabile aleatoria poissoniana

Una variabile aleatoria N a valori in \mathbb{N} ha una distribuzione di Poisson di parametro λ se

$$\mathbb{P}(N = n) = e^{-\lambda} \frac{\lambda^n}{n!} \quad \forall n \in \mathbb{N}.$$

Entrambe le variabili concorrono alla definizione di processo di Poisson nel seguente modo:

Definizione 1.3.1 (Processo di Poisson). Sia $(\tau_i)_{i \geq 1}$ una successione di variabili aleatorie indipendenti esponenziali di parametro λ e $T_n = \sum_{i=1}^n \tau_i$. Il processo $(N_t)_{t \geq 0}$ definito da

$$N_t = \sum_{n \geq 1} \mathbb{I}_{\{T_n \leq t\}} \quad (1.14)$$

è detto processo di Poisson.

La sua densità è

$$\mathbb{P}(N_t = n) = e^{-\lambda t} \frac{(\lambda t)^n}{n!} \quad \forall n \in \mathbb{N}.$$

Mentre la sua funzione caratteristica è

$$\Phi_t(u) = e^{\lambda t(e^{iu} - 1)}. \quad (1.15)$$

I tempi casuali τ_i rappresentano il tempo che intercorre tra un salto e l'altro e di conseguenza le variabili T_n sono gli istanti di salto del processo. Le traiettorie del processo di Poisson sono costanti a tratti con salti di ampiezza unitaria in corrispondenza di tempi aleatori. Vale inoltre la seguente proposizione:

Proposizione 1.2. *Sia N_t un processo di conteggio con incrementi indipendenti e stazionari. Allora N_t è un processo di Poisson.*

In generale si ha che il processo N_t non è una martingala. Per renderlo martingala si sottrae il suo valore atteso al tempo t .

$$\tilde{N}_t = N_t - \mathbb{E}[N_t] = N_t - \lambda t.$$

Il processo \tilde{N}_t è chiamato processo di Poisson compensato. In questo modo è verificata la proprietà di martingala: $\mathbb{E}[\tilde{N}_t | \tilde{N}_s] = \tilde{N}_s \quad \forall s \leq t$.

La sua funzione caratteristica è

$$\Phi_t(u) = e^{\lambda t(e^{iu} - 1 - iu)}. \quad (1.16)$$

Invece della funzione indicatrice che appare nella (1.14) si può scegliere una variabile aleatoria Y con distribuzione F_Y , indipendente dal processo N_t .

Definizione 1.3.2 (Compound Poisson). Un processo compound Poisson con intensità $\lambda > 0$ e distribuzione delle ampiezze dei salti f è un processo stocastico X_t della forma

$$X_t = \sum_{i=1}^{N_t} Y_i$$

dove le ampiezze dei salti Y_i sono i.i.d. con distribuzione f e N_t è un processo di Poisson con intensità λ , indipendente da $(Y_i)_{i \geq 1}$.

Il processo di Poisson omogeneo può essere visto come un compound Poisson con le ampiezze $Y_i \equiv 1$. Vale la seguente proprietà:

Proposizione 1.3. $(X_t)_{t \geq 0}$ è un processo compound Poisson \iff è un processo di Lévy con traiettorie costanti a tratti.

Altre definizioni necessarie per trattazione dei processi di Lévy sono le seguenti:

Definizione 1.3.3 (Funzione a variazione finita). Sia $\pi = \{a = t_0 < t_1 < \dots < t_n = b\}$ una partizione dell'intervallo reale $[a, b]$. Si definisce variazione della funzione f sulla partizione π la quantità

$$VA_\pi(f) = \sum_{i=0}^{n-1} |f(t_{i+1}) - f(t_i)|. \quad (1.17)$$

Inoltre se

$$TVA(f) = \sup_{\pi \in \Pi} VA_\pi(f) \leq \infty, \quad (1.18)$$

allora la funzione f si dice a variazione finita sull'intervallo $[a, b]$. In caso contrario, si indica come a variazione infinita.

E' necessario ritornare ora ai processi stocastici.

Definizione 1.3.4 (Processo a variazione finita). Il processo stocastico $(X_t)_t$ si dice a variazione finita se per quasi ogni $\omega \in \Omega$ le sue traiettorie sono funzioni a variazione finita. Se questo non accade il processo è a variazione infinita.

Definizione 1.3.5 (Processo cadlag). Un processo stocastico $(X_t)_t$ si definisce cadlag se per quasi ogni $\omega \in \Omega$ le sue traiettorie sono delle funzioni continue a destra e limitate a sinistra.

Il termine cadlag deriva dal francese *continue a droite et limites a gauche*. Se il processo stocastico $(X_t)_t$ è cadlag si denota con

$$X_{t-} = \lim_{s \rightarrow t-} X_s \quad (1.19)$$

il suo limite sinistro, e con

$$\Delta X_t = X_t - X_{t-} \quad (1.20)$$

il salto avvenuto al tempo t .

Definizione 1.3.6 (Processo di Lévy). Un processo stocastico cadlag $(X_t)_{t \geq 0}$ su $(\Omega, \mathcal{F}, \mathbb{P})$ tale che $X_0 = 0$ è detto processo di Lévy se:

1. Ha incrementi indipendenti: per ogni successione crescente di istanti temporali $t_0 \dots t_n$, le variabili aleatorie $X_{t_0}, X_{t_1} - X_{t_0}, \dots, X_{t_n} - X_{t_{n-1}}$ sono tra loro indipendenti.
2. Ha incrementi stazionari: la legge di $X_{t+h} - X_t$ non dipende da t .
3. Soddisfa la continuità stocastica: $\forall \epsilon > 0, \lim_{h \rightarrow 0} \mathbb{P}(|X_{t+h} - X_t| \geq \epsilon) = 0$.

L'ultima condizione comporta che la probabilità che si verifichi un salto al tempo t è nulla: le discontinuità avvengono, dunque, a tempi casuali.

Definizione 1.3.7 (Infinita divisibilità). Una distribuzione di probabilità F è detta infinitamente divisibile se esistono $n \geq 2$ variabili aleatorie i.i.d. Y_1, \dots, Y_n tali che $\sum_{i=1}^n Y_i \sim G$.

La legge dei processi di Lévy è infinitamente divisibile poichè gli incrementi dei processi di Lévy sono i.i.d. Inoltre i processi di Lévy ammettono un esponente caratteristico, cioè si può scrivere la loro funzione caratteristica come

$$\Phi(z) = e^{t\Psi(z)}.$$

Per i processi introdotti in seguito $\Psi(z)$ è noto analiticamente, quindi è possibile applicare i metodi di pricing basati su FFT.

1.3.2 Misure di salto

Definizione 1.3.8 (Misura aleatoria). Sia $(\Omega, \mathcal{F}, \mathbb{P})$ uno spazio di probabilità. Una misura aleatoria è una mappa $M : \Omega \times E \mapsto \mathbb{R}^+$ tale che:

- $\forall A \in \mathcal{E}$ la misura $\omega \mapsto M(\omega, A)$ è una variabile aleatoria.
- $A \mapsto M(\omega, A)$ è q.c. una misura su \mathcal{E} .

Se l'intensità dei salti è λ e la distribuzione delle loro ampiezze è f si può introdurre la misura $\nu(A) = \lambda f(A)$ per ogni $A \subseteq \mathbb{R}$. Tuttavia $\nu(A)$ non è una misura di probabilità poichè $\int_{\mathbb{R}} \nu(dx) = \lambda \int_{\mathbb{R}} f(dx) = \lambda$ che è diverso da 1 in generale. Ad ogni processo cadlag e in particolare ad ogni compound Poisson si associa una misura aleatoria su $\mathbb{R} \times [0, \infty)$ che descrive i salti di X , per ogni insieme misurabile $B \subset \mathbb{R} \times [0, \infty)$

$$J_X(B) = \#\{(t, X_t - X_{t-}) \in B\}.$$

Se $B = [t_1, t_2] \times A$ tale misura conta il numero di salti aventi ampiezze appartenenti ad A che accadono tra t_1 e t_2 .

Nel caso dei compound Poisson con intensità λ e distribuzione delle ampiezze di salto f si può dimostrare che J_X ha intensità $\mu(dx \times dt) = \nu(dx)dt = \lambda f(dx)dt$. Di conseguenza la misura di Lévy ν di un compound Poisson si può interpretare come il numero medio di salti nell'unità di tempo. In generale per i processi di Lévy si ha:

Definizione 1.3.9 (Misura di Lévy). Sia $(X_t)_{t \geq 0}$ un processo di Lévy su \mathbb{R} . La misura

$$\nu(A) = \mathbb{E}[\#\{t \in [0, 1] : \Delta X_t \neq 0, \Delta X_t \in A\}], \quad A \in \mathcal{B}(\mathbb{R})$$

è detta misura di Lévy di X .

Tale definizione consente di scrivere i compound Poisson nel modo seguente:

$$X_t = \sum_{s \in [0, t]} \Delta X_s = \int_{[0, t] \times \mathbb{R}} x J_x(ds \times dx). \quad (1.21)$$

Il processo si può riscrivere come somma dei suoi salti; grazie al fatto che per un compound Poisson il numero di salti è q.c. finito in ogni intervallo limitato l'integrale in (1.21) è ben definito.

1.3.3 Costruzione di processi di Lévy

Vi sono due categorie di modelli di Lévy: attività finita ed attività infinita. Nella prima il numero di salti, anche di piccola ampiezza, è finito. La seconda categoria include, invece, processi con un numero infinito di salti di piccola ampiezza in ogni intervallo temporale. Questi modelli non prevedono una componente diffusiva dato che la dinamica dei salti è già significativa.

Definizione 1.3.10 (Misura di Radon). Sia $E \in \mathbb{R}$, una misura di Radon su $(E, \mathcal{B}(E))$ è una misura μ tale che per ogni insieme compatto misurabile $A \in \mathcal{B}(E)$ si ha che $\mu(A) < \infty$.

Teorema 1.2 (Decomposizione di Lévy-Ito). Sia X_t un processo di Lévy e ν la sua misura di Lévy \implies

1. ν è una misura di Radon su $\mathbb{R} \setminus \{0\}$.
2. $\int_{|x| \geq 1} \nu(dx) < +\infty$.
3. $\int_{|x| \leq 1} |x|^2 \nu(dx) < +\infty$.
4. $X_t = X_t^c + X_t^l + \lim_{\epsilon \rightarrow 0} \tilde{X}_t^\epsilon$.

Dove

$$X_t^c = \gamma t + \sigma W_t, \quad (1.22)$$

$$X_t^l = \int_{|x| \geq 1, s \in [0, t]} x J_x(ds \times dx), \quad (1.23)$$

$$\begin{aligned} \tilde{X}_t^\epsilon &= \int_{\epsilon \leq |x| \leq 1, s \in [0, t]} x [J_x(ds \times dx) - \nu(dx) ds] \\ &= \int_{\epsilon \leq |x| \leq 1, s \in [0, t]} x \tilde{J}_x(ds \times dx). \end{aligned} \quad (1.24)$$

In questa decomposizione X_t^c rappresenta la parte continua del processo, X_t^l e \tilde{X}_t^ϵ la componente discontinua, in particolare X_t^l quella dei grandi salti e \tilde{X}_t^ϵ quella dei

piccoli salti, individuati come quelli di ampiezza minore di 1. La misura $\nu(B)$ è finita per ogni insieme compatto tale che $0 \notin B$, per non contraddire l'ipotesi cadlag. Non si ha comunque che la misura $\nu(B)$ è finita: il processo X_t potrebbe avere infiniti salti di ampiezza infinitesima. L'ultima uguaglianza in (1.24) è la versione compensata dell'integrale, si tratta di una martingala che permette di ottenere, insieme alla condizione di intergrabilità 3. su ν , la convergenza quasi certa per ogni t di \tilde{X}_t^ϵ .

Per definire univocamente un processo di Lévy si può specificare la tripletta caratteristica (γ, σ, ν) . Esiste un legame tra la tripletta caratteristica e l'esponente caratteristico fornito dal seguente

Teorema 1.3 (Formula di Lévy-Khintchine). *Sia X_t un processo di Lévy con tripletta caratteristica $(\gamma, \sigma, \nu) \implies$ l'esponente caratteristico è*

$$\Psi_X(z) = -\frac{1}{2}\sigma^2 z^2 + i\gamma z + \int_{\mathbb{R}} \left(e^{izx} - 1 - izx\mathbb{I}_{|x|\leq 1} \right) \nu(dx) \quad (1.25)$$

Modelli Jump-diffusion Un processo di Lévy del tipo Jump-diffusion ha la forma seguente:

$$X_t = \gamma t + \sigma W_t + \sum_{i=1}^{N_t} Y_i, \quad (1.26)$$

dove $(N_t)_{t \geq 0}$ è un processo di conteggio di Poisson e Y_i sono le jump sizes. Il modello è completamente specificato una volta assegnata la distribuzione della jump sizes $f(x)$. In questa categoria di modelli i più utilizzati sono quello di Merton e Kou.

Subordinazione browniana Una tecnica ulteriore per la costruzione di processi di Lévy consiste nella subordinazione del moto Browniano con drift $X_t = \theta t + \sigma W_t$ attraverso un processo $(T_t)_{t \geq 0}$ detto subordinatore. Il processo risultante $Y_t = X_{T_t} = \theta T_t + \sigma W_{T_t}$ è ancora un moto Browniano, ma osservato nella nuova scala temporale stocastica espressa da T_t . L'interpretazione finanziaria è quella di *business time*: nei momenti di maggior attività il tempo viene accelerato, viceversa viene rallentato al diminuire dell'attività. Il subordinatore scelto, $(T_t)_t$, è un processo α -stabile, con densità

$$\rho(x) = \frac{c e^{-\lambda x}}{x^{1+\alpha}} \mathcal{I}_{x>0}. \quad (1.27)$$

Si tratta di un processo di Lévy a tre parametri con salti soltanto positivi, dove $c > 0$, $\lambda > 0$ e $1 > \alpha \geq 0$. Il parametro c controlla la scala del processo, ovvero varia l'intensità dei salti di ogni ampiezza simultaneamente, λ seleziona il tasso di decadimento dei grandi salti e α determina il peso relativo dei piccoli salti nelle traiettorie del processo.

Spesso il processo viene riparametrizzato come

$$\rho(x) = \frac{1}{\Gamma(1-\alpha)} \left(\frac{1-\alpha}{\kappa} \right)^{1-\alpha} \frac{e^{-(1-\alpha)x/\kappa}}{x^{1+\alpha}} \quad (1.28)$$

Il parametro κ rappresenta la varianza del subordinatore al tempo 1 e di fatto è una misura della casualità del cambio di scala temporale, nel caso in cui sia $\kappa = 0$ si ha una funzione deterministica.

Effettuando un cambio di scala temporale ad un Moto Browniano (di volatilità σ e drift θ) si ottiene un processo *normal tempered stable*.

In generale il suo esponente caratteristico è del tipo

$$\Psi(\xi) = \frac{1 - \alpha}{\kappa\alpha} \left\{ 1 - \left(1 + \frac{\kappa(\xi^2\sigma^2/2 - i\theta\xi)}{1 - \alpha} \right)^2 \right\}. \quad (1.29)$$

Per $\alpha = 0$ e $\alpha = \frac{1}{2}$ la densità di probabilità di un subordinatore α -stabile è nota in forma chiusa. Di conseguenza, i processi corrispondenti VG e NIG sono stati ampiamente utilizzati in letteratura, grazie al fatto di essere più trattabili dal punto di vista matematico e di più facile simulazione.

Dopo aver descritto le proprietà dei processi di Lévy, vengono presentati quelli più comunemente utilizzati in letteratura.

1.3.4 Merton

Il modello Nel modello ad attività finita di Merton i salti del log price X_t hanno una distribuzione Gaussiana, $Y_i \sim N(\mu, \delta^2)$. I parametri che entrano in gioco sono 4

1. σ volatilità della componente diffusiva,
2. λ intensità dei salti,
3. μ media della jump size,
4. δ deviazione standard della jump size.

Esponente caratteristico L'esponente caratteristico del processo X_t è pari a:

$$\Psi_t(\xi) = -\frac{\sigma^2\xi^2}{2} + \lambda \left(e^{-\frac{\delta^2\xi^2}{2} + i\mu\xi} - 1 \right). \quad (1.30)$$

1.3.5 Kou

Il modello Nel modello ad attività finita di Kou i salti del log price X_t hanno una distribuzione esponenziale asimmetrica, con una densità della forma:

$$f(dx) = \left[p\lambda_+ e^{-\lambda_+ x} \mathbb{I}_{x>0} + (1-p)\lambda_- e^{-\lambda_- |x|} \mathbb{I}_{x<0} \right] dx,$$

dove $\lambda_+ > 0$ e $\lambda_- > 0$, il loro inverso rappresenta la media di ampiezza di salto, rispettivamente positivo e negativo, e $p \in [0, 1]$ rappresenta la probabilità che il salto possa essere positivo.

Esponente caratteristico L'esponente caratteristico del processo X_t è pari a:

$$\Psi_t(\xi) = -\frac{\sigma^2 \xi^2}{2} + i\xi \lambda \left(\frac{p}{\lambda_+ - i\xi} - \frac{1-p}{\lambda_- + i\xi} \right). \quad (1.31)$$

1.3.6 VG

Il modello Questo processo ad attività infinita si ottiene subordinando un moto Browniano con un *normal tempered stable process* di parametro $\alpha = 0$. Sostituendo nella (1.28) si ottiene

$$\rho(x) = \frac{1}{\sqrt{2\pi\kappa}} \frac{e^{-\frac{x}{2\kappa}}}{x^{3/2}}.$$

Esponente caratteristico L'esponente caratteristico del processo X_t è pari a:

$$\Psi_t(\xi) = -\frac{1}{k} \log \left\{ 1 + \frac{\xi^2 \sigma^2 k}{2} - i\theta \xi k \right\}. \quad (1.32)$$

1.3.7 NIG

Il modello Questo processo ad attività infinita si ottiene subordinando un moto Browniano con un *normal tempered stable process* di parametro $\alpha = 0.5$. Sostituendo nella (1.28) si ottiene

$$\rho(x) = \frac{1}{\kappa} \frac{e^{-x/\kappa}}{x}.$$

Esponente caratteristico L'esponente caratteristico del processo X_t è pari a:

$$\Psi_t(u) = -\frac{1}{k} \log \left\{ 1 + \frac{u^2 \sigma^2 k}{2} - i\theta uk \right\}. \quad (1.33)$$

Tuttavia la densità può essere parametrizzata anche nel seguente modo:

$$\rho(x) = \frac{\delta \alpha}{\pi|x|} e^{\beta x} K_1(\alpha|x|).$$

Esponente caratteristico Pertanto l'esponente caratteristico del processo X_t diventa:

$$\Psi_t(\xi) = \delta(\sqrt{\alpha^2 - (\beta + i\xi)^2} - \sqrt{\alpha^2 - \beta^2}). \quad (1.34)$$

1.4 I processi time-changed Lévy

Sia X_t un processo di Lévy con filtrazione \mathcal{F}_t e funzione caratteristica data dalla formula di Lévy-Khintchine (1.25). Nel contesto dei processi time-changed Lévy con X_t ci si riferisce al processo base. Si suppone che T_t sia un processo non negativo, non decrescente, continuo a destra e con limite a sinistra tale che per ogni t la variabile aleatoria T_t sia un tempo d'arresto rispetto a \mathcal{F}_t . La famiglia di tempi d'arresto T_t definisce in corrispondenza un cambio di tempo aleatorio e il processo risultante

$$M_t = X_{T_t}$$

è chiamato processo time-changed Lévy. Ci sono differenti metodi per scegliere un cambio di tempo appropriato a seconda del modello di rendimento del titolo finanziario. I due approcci più popolari sono i subordinatori e il cambio di tempo assolutamente continuo.

I subordinatori sono processi di Lévy non decrescenti, processi di salto puro con possibilità di attività infinita più una componente drift deterministica. Anche i processi di Lévy VG e NIG, rispettivamente introdotti nelle sezioni 1.3.6 e 1.3.7, possono essere generati a partire dal moto Browniano modificando il tempo tramite un subordinatore indipendente, Sezione 1.3.3.

Altri importanti tipi di cambio di tempo sono dati dal cambio di tempo assolutamente continuo della forma $T_t = \int_0^t v_s ds$, dove v_t è il tasso di attività (business) istantaneo. Intuitivamente, si può vedere t come la data del calendario e T_t come il business time al tempo t . Una giornata con attività di business frenetica, catturata da un più alto tasso di attività, implicherà una volatilità maggiore. Mentre al processo del tasso di attività istantaneo v_t è permesso avere salti, T_t rimane continuo in accordo con la sua definizione. Si noti che i salti in v_t sono richiesti non negativi in modo da garantire la proprietà di non decrescenza di T_t . Il principale vantaggio della scelta di un cambio di tempo continuo è che permette di modellizzare i ritorni dei titoli finanziari in maniera semplice.

Il processo del tasso di attività come processo CIR

Ci sono diverse scelte di processo stocastico per modellizzare il processo di attività, nell'elaborato si confinerà la scelta al processo CIR. Un processo v_t è un processo CIR se la sua dinamica è governata dalla seguente EDS ²

$$dv_t = \lambda(\bar{v} - v_t)dt + \eta\sqrt{v_t}dW_t^v. \quad (1.35)$$

Si noti che la singolarità del coefficiente di diffusione nell'origine impedisce un valore negativo a v_t una volta che v_t sia inzializzato a un valore positivo. In realtà, la presenza della radice di v_t nel termine di diffusione preclude valori negativi a v_t .

²equazione differenziale stocastica

Il modello CIR presenta nel termine di drift la proprietà di *mean reversion* e il parametro \bar{v} ha l'usuale interpretazione di *mean reversion level*. Il corrispondente *continuous CIR clock* con v_t come processo del tasso di attività è dato dal seguente integrale di cambio di tempo

$$T_t = \int_0^t v_s ds. \quad (1.36)$$

La condizione di Feller

Quando vale la condizione di Feller, $2\lambda\bar{v} \geq \eta^2$, l'origine diventa inaccessibile al processo CIR v_t , cioè $\mathbb{P}\{v_t = 0\} = 0$ se $v_0 \neq 0$. Questo garantisce che v_t rimanga sempre strettamente positivo. Sfortunatamente, i parametri del modello CIR ottenuti dalla calibrazione dei dati di mercato non soddisfano la condizione di Feller. Quando la condizione di Feller cade, la distribuzione cumulativa della varianza modellizzata dal processo CIR può mostrare un comportamento singolare nell'intorno dell'origine. L'origine è accessibile a v_t e fortemente riflettente. Nello specifico, la densità della varianza v_t è estremamente veloce nella coda di sinistra. Il fenomeno può dar luogo a errori significativi nei metodi di pricing di opzioni basati su integrazione, dove è adottato un troncamento finito del range di integrazione. Per risolvere questa difficoltà Fang e Oosterlee in [6] propongono di trasformare la funzione densità dal dominio della *varianza* al dominio della *log-varianza* definendo $\gamma_t = \ln v_t$. Siccome un valore negativo per v_t è precluso dal processo CIR il tasso della log-attività γ_t è ben definito.

Proprietà del processo CIR

Considerato il processo CIR v_t governato dalla (1.35), la distribuzione di v_t dato v_s , con $t > s$, è una distribuzione chi-quadro non centrata. Dati i seguenti parametri

$$\nu = \frac{2\lambda\bar{v}}{\eta^2} - 1 \quad \text{e} \quad \zeta = \frac{2\lambda}{[1 - e^{-\lambda(t-s)}]\eta^2},$$

la legge di transizione di v_t può essere espressa come:

$$v_t = \frac{1}{2\zeta} \chi_{2(\nu+1)}^2 [2\zeta e^{-\lambda(t-s)} v_s], \quad t > s \quad (1.37)$$

dove $\chi_{2(\nu+1)}^2(m)$ denota la variabile aleatoria chi-quadro con $2(\nu+1)$ gradi di libertà e di parametro non centrato m . Mentre la probabilità di v_t dato v_s , $s < t$, è data da

$$p(v_t|v_s) = \zeta e^{-\zeta[v_s e^{-\lambda(t-s)} + v_t]} \left[\frac{e^{\lambda(t-s)} v_t}{v_s} \right]^{\frac{\nu}{2}} I_\nu \left(2\zeta e^{-\frac{1}{2}\lambda(t-s)} \sqrt{v_s v_t} \right), \quad s < t \quad (1.38)$$

dove $I_\nu(\cdot)$ è la funzione modificata di Bessel del primo tipo di ordine ν . Dunque la condizione di Feller è equivalente a $\nu \geq 0$. Nel seguito si spiega come la trasformazione

dal dominio della *varianza* al dominio della *log-varianza* risolve il comportamento critico quando i parametri del modello CIR, ottenuti dalla calibrazione dei dati reali di mercato, non soddisfano la condizione di Feller. Sia $\gamma_t = \ln v_t$, la probabilità condizionata del processo del tasso della log-attività γ_t è la seguente

$$p_\gamma(\gamma_t|\gamma_s) = \zeta e^{-\zeta[e^{\gamma_s} e^{-\lambda(t-s)} + e^{\gamma_t}]} \left[e^{\gamma_t - \gamma_s} e^{\lambda(t-s)} \right]^{\frac{\nu}{2}} e^{\gamma_t} I_\nu \left(2\zeta e^{-\frac{1}{2}\lambda(t-s)} \sqrt{e^{\gamma_t} e^{\gamma_s}} \right). \quad (1.39)$$

La presenza del termine e^{γ_t} compensa il termine $(\cdot)^{\frac{\nu}{2}}$, cosicchè la densità condizionata del tasso della log-attività converge a zero quando $\gamma_t \rightarrow -\infty$. Confrontata alla densità condizionata del tasso di attività la nuova densità condizionata mostra due principali vantaggi. In primo luogo, la coda di sinistra decade a zero rapidamente invece di crescere significativamente, sebbene il tasso di decadenza possa decrescere quando ν si avvicina a -1. In secondo luogo, la nuova densità condizionata al variare di diversi parametri è più simmetrica. L'espressione in forma chiusa per la funzione caratteristica condizionata è

$$\begin{aligned} \Phi(\xi; \gamma_t, \gamma_s) &= \mathbb{E} \left[e^{i\xi \int_s^t v_u du} | \gamma_t, \gamma_s \right] \\ &= \frac{I_\nu \left(\sqrt{e^{\gamma_t} e^{\gamma_s}} \frac{4\tilde{\gamma}(\xi) e^{-\frac{1}{2}\tilde{\gamma}(\xi)(t-s)}}{\eta^2 [1 - e^{-\tilde{\lambda}(\xi)(t-s)}]} \right)}{I_\nu \left(\sqrt{e^{\gamma_t} e^{\gamma_s}} \frac{4\lambda e^{-\frac{1}{2}\lambda(t-s)}}{\eta^2 [1 - e^{-\lambda(t-s)}]} \right)} \frac{\tilde{\gamma}(\xi) e^{-\frac{1}{2}[\tilde{\gamma}(\xi) - \lambda](t-s)} [1 - e^{-\lambda(t-s)}]}{\lambda [1 - e^{-\tilde{\lambda}(\xi)(t-s)}]} \\ &\quad \exp \left(\frac{e^{\gamma_t + \gamma_s}}{\eta^2} \left\{ \frac{\lambda [1 + e^{-\lambda(t-s)}]}{1 - e^{-\lambda(t-s)}} - \frac{\tilde{\gamma}(\xi) [1 + e^{-\tilde{\gamma}(\xi)(t-s)}]}{[1 - e^{-\tilde{\gamma}(\xi)(t-s)}]} \right\} \right). \end{aligned} \quad (1.40)$$

dove $\tilde{\gamma}(\xi) = \sqrt{\lambda^2 - 2i\eta^2\xi}$.

1.4.1 Leverage nei processi time-changed Lévy

Al fine di poter successivamente introdurre il modello di Heston, che verrà spiegato nella prossima sezione, nella classe dei processi time-changed Lévy, è necessario introdurre una generalizzazione rispetto a quanto detto finora. Ciò consiste nell'aggiunta del coefficiente di leverage, rappresentato da un termine cv_t , dove

$$L_t = X_{T_t} + cv_t. \quad (1.41)$$

La funzione generatrice dei momenti condizionata di $L_t - L_s$ ammette la seguente struttura

$$\begin{aligned} \mathbb{E}[e^{\omega(L_t - L_s)} | \mathcal{F}_s, \gamma_t] &= e^{\omega c(v_t - v_s)} \mathbb{E}[e^{\omega(X_{T_t} - X_{T_s})} | \mathcal{F}_s, \gamma_t] \\ &= e^{\omega c(v_t - v_s)} \mathbb{E} \left[\mathbb{E}[e^{\omega(X_{T_t} - X_{T_s})} | T_t - T_s, \mathcal{F}_s, \gamma_t] | \mathcal{F}_s, \gamma_t \right] \\ &= e^{\omega c(e^{\gamma_t - \gamma_s})} \mathbb{E} \left[e^{-\psi_X(-i\omega)(T_t - T_s)} | \mathcal{F}_s, \gamma_t \right] \\ &= e^{\omega c(e^{\gamma_t - \gamma_s})} \Phi(i\psi_X(-i\omega); \gamma_t, \gamma_s), \end{aligned} \quad (1.42)$$

dove $\Phi(\xi; \gamma_t, \gamma_s)$ è dato dalla formula (1.40).

1.4.2 Il modello a volatilità stocastica di Heston

Viene ora introdotto il modello di Heston, in cui non compaiono più i salti, ma una volatilità del sottostante con dinamica stocastica. Il principale vantaggio di modelli a volatilità stocastica rispetto ai modelli con salti risiede nella capacità di riprodurre gli smile osservati sul mercato: mentre i processi di Lévy funzionano bene per brevi scadenze, quelli a volatilità stocastica performano meglio per lunghe scadenze.

Il modello Nel modello di Heston compaiono i processi stocastici di prezzo e varianza, rispettivamente S_t e v_t , correlati tramite il coefficiente di correlazione ρ . Il sistema di equazioni differenziali stocastiche del modello è il seguente

$$\begin{aligned} dS_t &= rS_t dt + S_t \sqrt{v_t} dW_t, \\ dv_t &= \lambda(\bar{v} - v_t) dt + \eta \sqrt{v_t} dW_t^v. \end{aligned} \quad (1.43)$$

dove $\mathbb{E}[dW_t dW_t^v] = \rho dt$. Il parametro λ rappresenta la velocità di mean reversion della varianza, \bar{v} è la varianza di lungo periodo, ρ è la correlazione tra i moti browniani W e W^v che guidano i processi, η è la vol-of-vol³, S_0 è il prezzo spot e v_0 la varianza spot.

La dinamica del logaritmo del prezzo $Y_t = \ln \frac{S_t}{S_0}$ è

$$dY_t = \left(r - \frac{1}{2} v_t \right) dt + \sqrt{v_t} dW(t).$$

Funzione caratteristica La funzione caratteristica del processo Y_t è nota in forma chiusa ed è pari a:

$$\Phi_t(u, T) = e^{C(T-t, u) + D(T-t, u)v + i u x} \quad (1.44)$$

dove $v = v_t$ è il valore al tempo t della volatilità, $y = Y_t$ è il valore al tempo t del processo e

$$\begin{aligned} C(t, u) &= \frac{\lambda \bar{v}}{\eta^2} \left\{ (\lambda - \rho \eta u i - D)(T - t) - 2 \ln \left[\frac{G e^{-D(T-t)}}{G - 1} - 1 \right] \right\}, \\ D(t, u) &= \frac{\lambda - \rho \eta u i - D}{\eta^2} \left[\frac{1 - e^{-D(T-t)}}{1 - G e^{-D(T-t)}} \right], \\ G &= \frac{\lambda - \rho \eta u i - D}{\lambda - \rho \eta u i + D}, \\ D &= \sqrt{(\lambda - \rho \eta u i)^2 + u(i + u)\eta^2}. \end{aligned}$$

³volatilità della volatilità

Heston come time changed Lévy con leverage Come detto precedentemente, all'interno dei processi time-changed Lévy mediante CIR, con leverage (1.41), si può includere il modello di Heston a volatilità stocastica (1.43). Infatti il processo log-asset return Y_t , come dimostrato in [18], può essere rappresentato alternativamente come segue

$$Y_t = \ln \frac{S_t}{S_0} = rt + \tilde{W}_{T_t} - \frac{1}{2}T_t, \quad (1.45)$$

dove $d\tilde{W}_{T_t}$ e $\sqrt{v_t}dW_t$ sono uguali in distribuzione e T_t è dato da (1.36). Fortunatamente, il log-asset return condizionato a v_t si può esprimere in termini di X_{T_t} come

$$Y_t|v_t = \left(r - \frac{\rho\lambda\bar{v}}{\eta} \right) t + X_{T_t} + \frac{\rho}{\eta}(v_t - v_0), \quad (1.46)$$

dove $X_t \sim N\left(\left(\frac{\rho\lambda}{\eta} - \frac{1}{2}\right)t, (1 - \rho^2)t\right)$. Il corrispondente esponente caratteristico è dato da

$$\psi_X(\xi) = \left(\frac{1}{2} - \frac{\rho\lambda}{\eta}\right) i\xi + \frac{1}{2}(1 - \rho^2)\xi^2. \quad (1.47)$$

La funzione generatrice dei momenti condizionata della differenza del log-asset return è definita da

$$\Psi(\omega; \gamma_t, \gamma_s) = \mathbb{E}[e^{\omega(Y_t - Y_s)} | \mathcal{F}_s, \gamma_t]. \quad (1.48)$$

Combinando dunque $Y_t|v_t$ e $\psi_X(\xi)$ con la classe dei processi time-changed Lévy con leverage si ottiene la rappresentazione analitica di $\Psi(\omega; \gamma_t, \gamma_s)$,

$$\Psi(\omega; \gamma_t, \gamma_s) = e^{\omega\left\{r(t-s) + \frac{\rho}{\eta}[e^{\gamma_t} - e^{\gamma_s} - \lambda\bar{v}(t-s)]\right\}} \Phi\left(-i\omega\left(\frac{\rho\lambda}{\eta} - \frac{1}{2}\right) - \frac{1}{2}i\omega^2(1 - \rho^2); \gamma_t, \gamma_s\right). \quad (1.49)$$

1.4.3 Probabilità di transizione della varianza

I problemi nella coda di sinistra

La varianza nel modello di Heston è governata dalla distribuzione chi-quadro non centrata, equazione (1.37). Per alcune combinazioni dei parametri, la densità della varianza cresce vertiginosamente nella coda di sinistra: quando la varianza tende a zero la densità tende a infinito. Si vuole quindi determinare l'insieme dei parametri per i quali questo comportamento singolare non accade, viceversa il problema viene contrastato, come già detto, grazie alla trasformazione dal dominio della varianza al dominio della varianza logaritmica.

Come primo passo per comprendere il comportamento singolare della varianza si esegue una serie di simulazioni numeriche per evidenziare la singolarità nella coda di sinistra. I risultati che si otterranno possono essere utilizzati come regola per determinare per quali valori la densità della varianza è caratterizzata da valori estremamente grandi.

Risultato *Sebbene ognuno dei parametri λ , \bar{v} ed η in (1.35) svolga un ruolo indipendente nella determinazione della forma e della magnitudine della densità della varianza, il tasso di decadimento nella coda di sinistra può essere ben caratterizzato dal valore di ν , il cui intervallo di definizione è $[-1, \infty)$. Basandosi sulla non negatività di λ , \bar{v} ed η , il problema sorge quando $\nu \in [-1, 0]$, che è in relazione diretta con la condizione di Feller.*

Gli esperimenti che supportano questa intuizione sono i seguenti: i valori di \bar{v} ed η sono scelti aleatoriamente dall'intervallo $[0, 1]$ (si considera ragionevole questo intervallo sia per \bar{v} , che per η), mentre λ è dato da $(1 + \nu)\eta^2/(2\bar{v})$. I risultati sono riportati in [Figura 1.1](#). Come mostrato, il valore di ν determina il tasso di decadimento nella coda di sinistra della densità della varianza, mentre la coda di destra decade a zero rapidamente a prescindere da ν . Per $\nu \gg 0$ la densità tende a zero in entrambe le code. Per ν piccolo e vicino a zero la coda di sinistra decade più lentamente. Per $\nu = 0$, la coda di sinistra rimane pressochè costante, mentre per $\nu \in [-1, 0]$ cresce drasticamente. Il fatto che ν determini il decadimento nella coda di sinistra può essere compreso osservando attentamente la (1.38). Quando ν cambia segno, entrambe le funzioni $(\cdot)^{\nu/2}$ e I_ν cambiano andamento nell'intorno dell'origine: da monotone crescenti diventano monotone decrescenti.

Trasformazione in varianza logaritmica

Basandosi su quanto spiegato prima, in [6] è stata proposta una soluzione strategica al problema nella coda di sinistra: si trasforma il problema dal dominio della varianza al dominio della varianza logaritmica. Grazie a un cambio di variabili, la densità del processo della varianza logaritmica diventa (1.39). Questo cambio fa apparire il termine $e^{\gamma t}$, che per $\nu \in [-1, 0]$ compensa il termine $(\cdot)^{\nu/2}$, in questo modo la densità converge a zero quanto $\gamma t \rightarrow -\infty$. In [Figura 1.2](#) emerge come la densità della varianza logaritmica per differenti insiemi di parametri sia più simmetrica di quella di [Figura 1.1](#).

Inoltre è illustrato in scala logaritmica, [Figura 1.3](#), che per $\nu \in [-1, 0]$ la densità nella coda di sinistra non cresce significativamente in valore. Anzi le code decadono a zero rapidamente quando $\gamma t \rightarrow -\infty$, sebbene il tasso di decadimento decresca quando ν si avvicina a -1. In [Figura 1.2](#) e in [Figura 1.3](#) si sono solamente mostrati i casi problematici di [Figura 1.1](#); naturalmente anche nel dominio della varianza logaritmica la coda di sinistra nei casi $\nu > 0$ decade velocemente a zero.

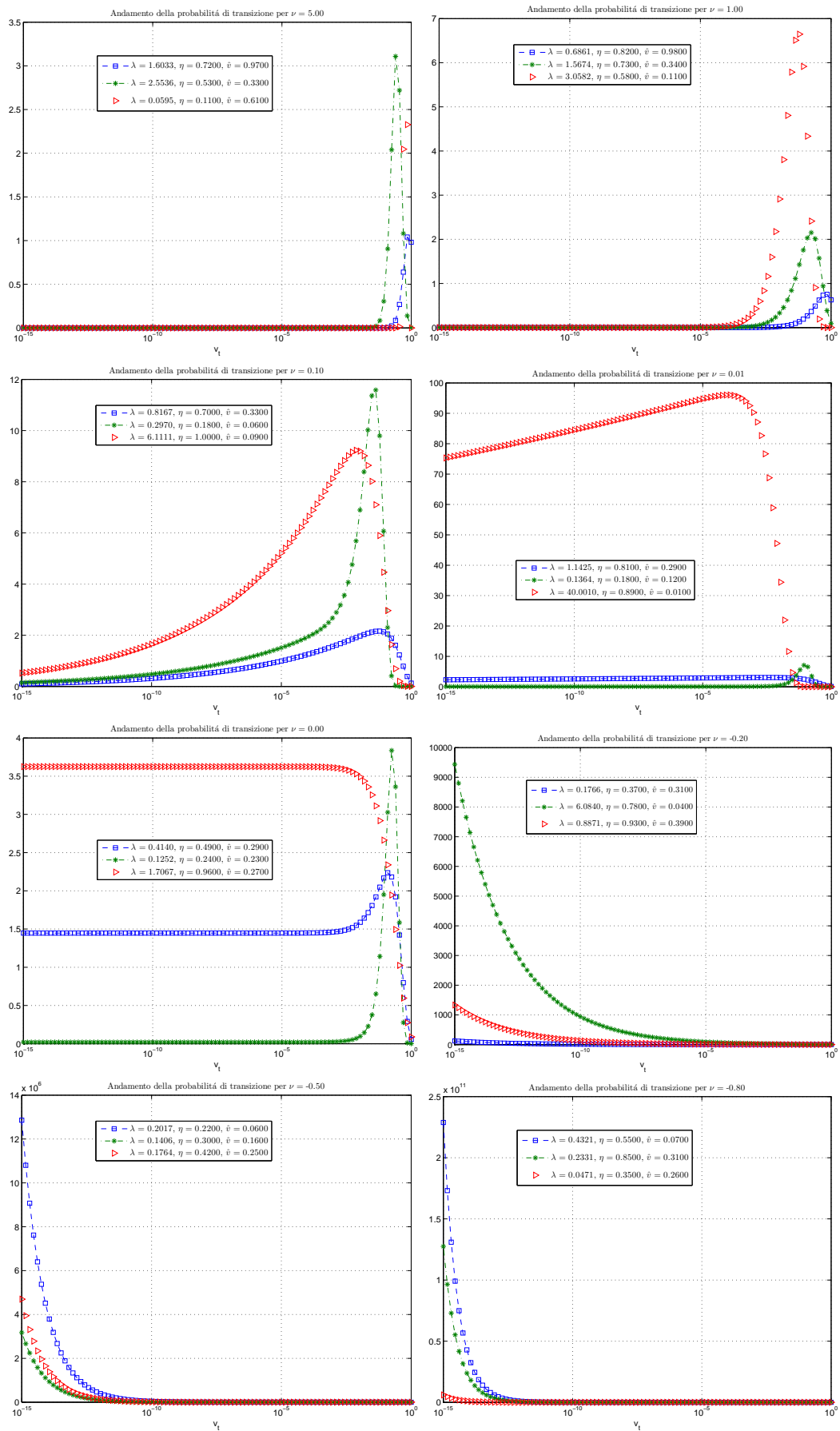


Figura 1.1: Tasso di decadimento nella coda di sinistra della densità della varianza, quando ν tende a -1

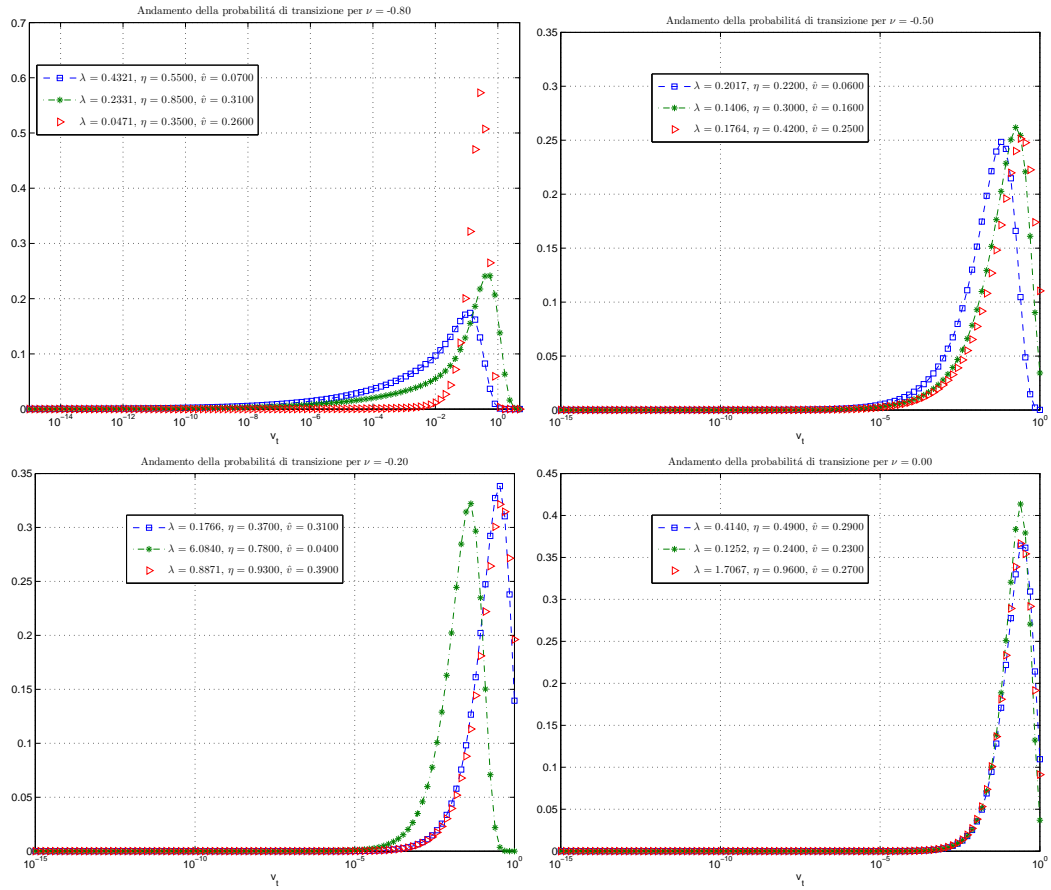


Figura 1.2: Tasso di decadimento nella coda di sinistra della densità della varianza logaritmica, quando ν tende a -1

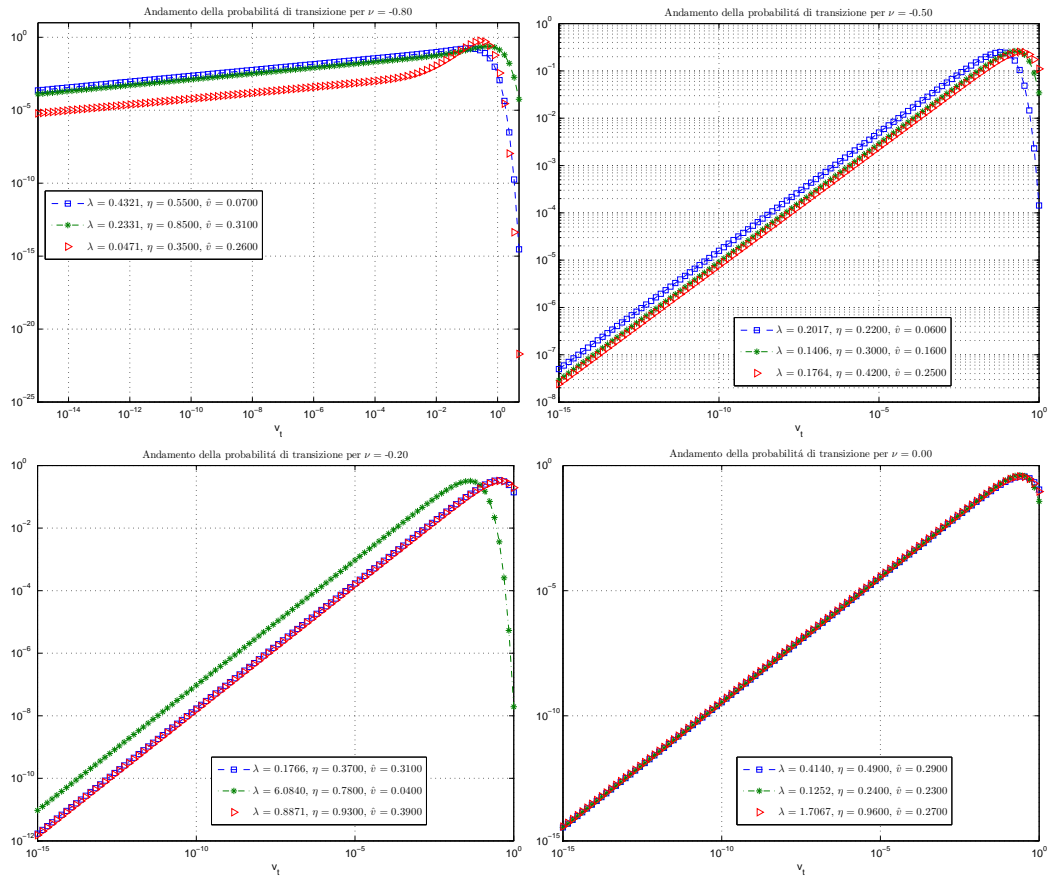


Figura 1.3: Tasso di decadimento in scala logaritmica nella coda di sinistra della densità della varianza logaritmica, quando ν tende a -1

1.5 Pricing neutrale rispetto al rischio

L'obiettivo dell'elaborato è quello di fare pricing, a tal fine occorre introdurre il concetto di assenza di arbitraggio. Verranno in seguito presentati i due teoremi fondamentali dell'asset pricing e verrà ricavata la condizione di non arbitraggio per i processi di Lévy. I concetti di seguito introdotti servono per poter di fatto utilizzare i modelli precedentemente introdotti. Per una trattazione completa si può fare riferimento a [3].

Si consideri un mercato finanziario consistente solamente in due asset:

- Un asset non rischioso, con drift pari al tasso risk free: B_t t.c. $dB_t = rB_t dt$ e $B_0 = 1$. Questo processo rappresenta la dinamica del risparmio detenuto in un conto bancario che capitalizza al tasso r .
- Un asset rischioso, il cui prezzo è descritto dal processo stocastico S_t .

Definizione 1.5.1 (Misura di martingala equivalente). Una misura di probabilità \mathbb{Q} su \mathcal{F} è detta misura di martingala equivalente per il modello (B, S) se è equivalente alla misura \mathbb{P} e il processo $\hat{S}_t = \frac{S_t}{B_t}$ è una martingala sotto la misura \mathbb{Q} .

Definizione 1.5.2 (Processo valore di un portafoglio). Dato un portafoglio, ovvero un'allocazione della ricchezza nelle due asset class disponibili, rappresentato da un processo $\xi = (\xi^B, \xi^S)$, il suo processo di valore si scrive

$$V_t^\xi = \xi_t^B B_t + \xi_t^S S_t.$$

Definizione 1.5.3 (Portafoglio autofinanziante). Un portafoglio $\xi = (\xi^B, \xi^S)$ è detto autofinanziante se vale che

$$dV_t^\xi = \xi_t^B dB_t + \xi_t^S dS_t. \quad (1.50)$$

La condizione (1.50) formalizza l'ipotesi che l'investitore non aggiunge nè sottrae capitale al portafoglio: le variazioni nel valore del portafoglio autofinanziante sono proporzionali alle variazioni dei due asset del modello.

Definizione 1.5.4 (Arbitraggio). Un portafoglio $\xi = (\xi^B, \xi^S)$ autofinanziante è detto di arbitraggio se il processo di valore V^ξ verifica:

- $V_0^\xi = 0$.
- $\mathbb{P}(V_T^\xi \geq 0) = 1$.
- $\mathbb{P}(V_T^\xi > 0) > 0$.

La possibilità di arbitraggio, o *free lunch*, è equivalente alla possibilità di ottenere un guadagno senza richiedere alcun esborso iniziale; cioè non si è sottoposti ad alcun rischio di realizzare perdite. Un modello di mercato si dice privo di arbitraggio se non esistono portafogli in grado di realizzare arbitraggi.

Teorema 1.4 (Primo teorema fondamentale dell'asset pricing). *Assumendo che il processo S sia limitato, allora esiste una misura di martingala \mathbb{Q} per il modello $(B, S) \iff$ tale modello soddisfa la condizione NFLVR (No Free Lunch With Vanishing Risk)*

La condizione NFLVR è una generalizzazione della condizione di assenza di arbitraggio, in pratica è possibile individuare una successione di portafogli autofinanzianti che approssimano quello di arbitraggio e per le quali il rischio tende a 0. L'importanza del teorema risiede nel fatto che l'esistenza di \mathbb{Q} permette di ottenere una formula di pricing neutrale rispetto al rischio.

Teorema 1.5 (Formula di pricing neutrale rispetto al rischio). *Dato il modello di mercato (B, S) ed un payoff \mathcal{X}_T si ha che il processo di prezzo $V_{\mathcal{X}}(t)$ che soddisfa l'assenza di arbitraggio è dato da*

$$V_{\mathcal{X}}(t) = e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}}[\mathcal{X}_T | \mathcal{F}_t]. \quad (1.51)$$

Si noti che $\mathbb{E}^{\mathbb{Q}}[\cdot | \mathcal{F}_t]$ è l'operatore di valore atteso rispetto alla misura \mathbb{Q} , condizionato alla filtrazione \mathcal{F}_t . Questa formula è di fondamentale importanza e sarà utilizzata per il pricing.

Un'ulteriore nozione è quella di completezza del mercato. Un modello di mercato si dice completo se per ogni payoff X esiste un portafoglio ξ che lo replica.

Teorema 1.6 (Secondo teorema fondamentale dell'asset pricing). *Il modello di mercato (B, S) è completo \iff esiste un'unica misura di martingala equivalente \mathbb{Q} .*

I modelli introdotti in precedenza, con esclusione del GBM, corrispondono a mercati incompleti. A causa dell'assenza di un'unica misura di martingala il prezzo di un derivato non è univoco, ma vi è un range di possibili prezzi, i quali soddisfano tutti la condizione di non arbitraggio. Dalla (1.51) si ha però che in ogni mercato privo di arbitraggio è possibile rappresentare i prezzi dei derivati come valori attesi condizionati e scontati rispetto ad una misura di martingala \mathbb{Q} . Di conseguenza sorge il problema di determinare una misura \mathbb{Q} in modo che il processo di prezzo scontato ($e^{-rt}S_t$) sia una martingala. In particolare, per i modelli di Lévy, vale il seguente:

Teorema 1.7. *Sia $(X_t)_{t \geq 0}$ un processo di Lévy con tripletta (γ, σ^2, ν) tale che*

$$\int_{|y| \geq 1} e^y \nu(dy) < +\infty.$$

Il processo e^{X_t} è una martingala \iff

$$\Psi(-i) = \gamma + \frac{\sigma^2}{2} + \int_{-\infty}^{+\infty} (e^z - 1 - z\mathbb{I}_{|z| < 1}) = 0. \quad (1.52)$$

Per una dimostrazione si faccia riferimento a [5]. La condizione (1.52) risulta equivalente a richiedere

$$\mathbb{E}[X_t] < \infty \quad \forall t \geq 0, \quad (1.53)$$

ciò si impone che il processo S abbia valore atteso finito per ogni t .

Per quanto riguarda Heston, la dinamica neutrale rispetto al rischio è già stata riportata nella sezione [1.4.2](#).

Capitolo 2

Opzioni Europee e Barriera

In questo capitolo viene presentato il metodo di pricing basato sulla trasformata di Hilbert applicato ad opzioni *path dependent* a monitoraggio discreto, sia con barriera singola che con barriera doppia, sotto l'ipotesi che il sottostante evolva come un processo di Lévy. Come punto di partenza, il metodo sarà utilizzato per la valutazione di opzioni di tipo europeo.

I vantaggi del metodo che verrà descritto sono molteplici:

- Il metodo è applicabile ad ogni processo di Lévy (jump-diffusion o di puro salto ad attività infinita) e ad ogni funzione payoff (soggetta alla condizione di integrabilità).
- Il metodo è applicabile sia ad opzioni a barriera singola che a barriera doppia.
- Il metodo è applicabile anche a contratti con date di monitoraggio della barriera non equispaziate, con barriere non costanti e con data di valutazione compresa tra due istanti di monitoraggio.
- Il metodo ha un errore di discretizzazione che decade esponenzialmente in $1/h$, dove h è il passo di discretizzazione usato per calcolare la trasformata di Hilbert: l'errore di discretizzazione decade come $O(\exp(-C_1/h))$. Se la funzione caratteristica del processo di Lévy decade come $\exp(-tc|\xi|^\nu)$ per $|\xi| \rightarrow \infty$, allora si può provare che l'errore di troncamento che tronca la somma infinita approssimante la trasformata di Hilbert dopo M termini è $O(h^{-1}(Mh)^{1-\nu} \exp(-\Delta c(Mh)^\nu))$ per le opzioni a barriera singola e $O((Mh)^{1-\nu} \exp(-\Delta c(Mh)^\nu))$ per le opzioni a barriera doppia, dove Δ è l'intervallo di monitoraggio della barriera. Selezionando appropriatamente $h = h(M)$, l'errore totale diventa $O(M^{1/1+\nu} \exp(-CM^{\frac{\nu}{1+\nu}}))$ per le opzioni a barriera singola e $O(\max(1, M^{(1-\nu)/(1+\nu)}) \exp(-CM^{\frac{\nu}{1+\nu}}))$ per le opzioni a barriera doppia.
- Lo sforzo computazionale complessivo è di $O(NM \log_2 M)$, lineare nel numero di date di monitoraggio N e pari a $M \log_2 M$ nel numero di punti M necessari per il calcolo della trasformata di Hilbert.

Per tutta la trattazione svolta in questo capitolo, sia riguardo alle opzioni europee che alle barriera, e per le dimostrazioni dei teoremi presentati si è fatto riferimento a [9].

2.1 Introduzione

Prima di iniziare servono alcuni risultati sui processi di Lévy.

Definizione 2.1.1. Sia X un processo di Lévy con tripletta (μ, σ^2, Π) si definisce l'insieme \mathcal{I}_X come

$$\mathcal{I}_X := \left\{ \alpha \in \mathbb{R} : \int_{\{|x|>1\}} e^{-\alpha x} \Pi(dx) < \infty \right\}. \quad (2.1)$$

Grazie al teorema di Sato, $\alpha \in \mathcal{I}_X$ se e solo se $\mathbb{E}[e^{-\alpha X_t}] < \infty$ per qualche $t > 0$ o, equivalentemente, per ogni $t > 0$. \mathcal{I}_X è un intervallo finito o infinito, contenente l'origine con estremi λ_- e λ_+ , $-\infty \leq \lambda_- \leq 0 \leq \lambda_+ \leq \infty$. Gli estremi possono appartenere o non appartenere ad \mathcal{I}_X ed è possibile che $\lambda_- = \lambda_+ = 0$, nel qual caso $\mathcal{I}_X = 0$.

Definizione 2.1.2. Supponendo che almeno uno dei due estremi di \mathcal{I}_X sia non zero, si definisce l'insieme \mathcal{S}_X come

$$\mathcal{S}_X := \{z \in \mathbb{C} : \text{Im}(z) \in (\lambda_-, \lambda_+)\}. \quad (2.2)$$

Si ricava immediatamente che l'esponente caratteristico $\Psi(z)$, come funzione della variabile complessa z , è analitico nello strip \mathcal{S}_X . Se si modella lo spot price $\{S_t, t \geq 0\}$ come esponenziale di un processo di Lévy $\{X_t, t \geq 0\}$, allora $\mathbb{E}[S_t^{-\alpha}] < \infty$ per ogni $\alpha \in \mathcal{I}_X$ e per ogni $t > 0$. In particolare, per assicurarsi che il titolo stesso sia prezzabile, si deve avere $\mathbb{E}[e^{X_t}] < \infty$ per ogni $t > 0$ e dunque dovrà essere $\lambda_- \leq -1$. Di conseguenza, nell'elaborato si restringe l'attenzione ai processi di Lévy con $[-1, 0] \in \mathcal{I}_X$.

Consideriamo ora la famiglia di operatori indicizzati da $t \geq 0$:

$$P_t f(x) = \mathbb{E}_x[f(X_t)] = \int_{\mathbb{R}} f(x+y) P_t(dy), \quad (2.3)$$

dove $P_t(dy)$ è la misura di transizione di X_t con punto iniziale nell'origine. La famiglia di operatori $\{P_t, t \geq 0\}$ definisce un semigruppato markoviano su ogni spazio $L^p(\mathbb{R})$ con $1 \leq p \leq \infty$. In particolare, si considera il semigruppato $\{P_t, t \geq 0\}$ su $L^1(\mathbb{R})$.

Si definisce la trasformata di Fourier di una funzione $f \in L^1(\mathbb{R})$ come:

Definizione 2.1.3 (Trasformata di Fourier).

$$\hat{f}(\xi) = \mathcal{F}f(\xi) = \int_{\mathbb{R}} e^{i\xi x} f(x) dx, \quad \xi \in \mathbb{R}. \quad (2.4)$$

Quindi per ogni $f \in L^1(\mathbb{R})$ e per ogni $\xi \in \mathbb{R}$ si ha

$$\mathcal{F}(P_t f)(\xi) = \phi_t(-\xi) \hat{f}(\xi), \quad t \geq 0. \quad (2.5)$$

Dunque per invertire la trasformata di Fourier in (2.5) per qualche $t > 0$, deve valere

$$\int_{\mathbb{R}} |\phi_t(-\xi) \hat{f}(\xi)| d\xi < \infty. \quad (2.6)$$

Siccome \hat{f} è limitata, per invertire la trasformata di Fourier (2.5) per ogni $f \in L^1(\mathbb{R})$ e per ogni $t > 0$, è sufficiente che

$$\int_{\mathbb{R}} |\phi_t(\xi)| d\xi = \int_{\mathbb{R}} e^{-t \operatorname{Re} \Psi(\xi)} d\xi < \infty, \quad \text{per ogni } t > 0. \quad (2.7)$$

In particolare, (2.7) vale se la funzione caratteristica ha il seguente andamento, con $c, k > 0$ e $\nu \in (0, 2]$:

$$|\phi_t(\xi)| = e^{-t \operatorname{Re} \Psi(\xi)} \leq k e^{-tc|\xi|^\nu}. \quad (2.8)$$

Assumendo che (2.7) valga, si può invertire la trasformata di Fourier in (2.5) per ogni $f \in L^1(\mathbb{R})$ e per ogni $t > 0$ e riscrivere $P_t f(x)$ come

$$P_t f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-i\xi x} \phi_t(-\xi) \hat{f}(\xi) d\xi. \quad (2.9)$$

L'equazione (2.5) è ovviamente soddisfatta se il processo di Lévy ha una componente diffusiva con $\sigma > 0$. Per un processo di puro salto, invece, la verifica di (2.7) dipende dal comportamento della misura di Lévy nell'intorno dell'origine. Consideriamo un processo di Lévy puro. Si supponga che Π soddisfi

$$\liminf_{r \rightarrow 0} \frac{\int_{[-r, r]} x^2 \Pi(dx)}{r^{2-\nu}} > 0 \quad (2.10)$$

per qualche $0 < \nu < 2$. Allora per ogni $t > 0$ la misura di transizione $P_t(dy)$ ha densità $p_t(y)$ di classe $C^\infty(\mathbb{R})$ e tutte le derivate della densità tendono a zero quando $|x| \rightarrow \infty$. In questo caso, per la funzione caratteristica valgono le stime (2.8) e (2.7). Per i processi di Lévy per cui (2.7) non è soddisfatta, la rappresentazione di Fourier (2.9) vale per quelle $f \in L^1(\mathbb{R})$ e $t > 0$ per cui vale (2.6). In particolare, essa vale per ogni $t > 0$ e $f \in L^1(\mathbb{R})$ per le quali $\hat{f} \in L^1(\mathbb{R}, \mathbb{C})$.

Il teorema seguente costituisce il mattone per il pricing di opzioni attraverso l'approccio con trasformata di Fourier.

Teorema 2.1. *Sia X un processo di Lévy tale che almeno uno dei due estremi $\{\lambda_-, \lambda_+\}$ di I_X sia diverso da zero.*

(i) *Per ogni $\alpha \in (\lambda_-, \lambda_+)$, $\{P_t, t \geq 0\}$ è un semigruppato fortemente continuo su $L^1_\alpha(\mathbb{R})$, definizione in (2.18), ed $\{e^{t\Psi(i\alpha)} P_t, t \geq 0\}$ è una contrazione fortemente continua su $L^1_\alpha(\mathbb{R})$.*

(ii) *Se $f \in L^1_\alpha(\mathbb{R})$ per qualche $\alpha \in (\lambda_-, \lambda_+)$ è tale che per qualche $t > 0$*

$$\int_{\mathbb{R}} |\phi_t(-\xi + i\alpha) \hat{f}_\alpha(\xi)| d\xi < \infty, \quad (2.11)$$

(se la condizione (2.11) è soddisfatta per qualche $t > 0$, allora è soddisfatta per tutti gli $s \geq t$) dove $\hat{f}_\alpha(\xi)$ è la trasformata di Fourier di $f_\alpha(x) := e^{\alpha x} f(x) \in L^1(\mathbb{R})$, allora l'operatore P_t ha la seguente rappresentazione di Fourier:

$$P_t f(x) = \frac{1}{2\pi} e^{-\alpha x - t\Psi(i\alpha)} \int_{\mathbb{R}} e^{-i\xi x} \phi_t^{(\alpha)}(-\xi) \hat{f}_\alpha(\xi) d\xi, \quad (2.12)$$

dove

$$\phi_t^{(\alpha)}(\xi) = \frac{\phi_t(\xi + i\alpha)}{\phi_t(i\alpha)} = e^{-t\Psi^{(\alpha)}(\xi)}, \quad \Psi^{(\alpha)}(\xi) = \Psi(\xi + i\alpha) - \Psi(i\alpha) \quad (2.13)$$

sono le funzioni caratteristiche e l'esponente caratteristico della trasformata di Esscher del processo di Lévy $X^{(\alpha)}$ con tripletta $(\mu^{(\alpha)}, \sigma^2, \Pi^{(\alpha)})$ con

$$\mu^{(\alpha)} = \mu - \sigma^2 \alpha + \int_{[-1,1]} x(e^{-\alpha x} - 1) \Pi(dx), \quad \Pi^{(\alpha)}(dx) = e^{-\alpha x} \Pi(dx). \quad (2.14)$$

Per una sua dimostrazione si faccia riferimento a [9]. La sua interpretazione probabilistica corrisponde alla trasformata di Esscher della misura di probabilità, la cui definizione, viene per completezza riportata ora:

Trasformata di Esscher

Spesso per avere l'integrabilità del payoff di un contratto è conveniente fare un cambio di misura di un processo di Lévy. Si definisce una nuova misura di probabilità \mathbb{P}^α grazie al teorema di Radon-Nikodym

$$\frac{d\mathbb{P}^\alpha}{d\mathbb{P}} \Big|_{\mathcal{F}_t} = Z_t. \quad (2.15)$$

Questa misura è chiamata trasformata di Esscher o *exponential tilting*. Un processo di Lévy che ha subito la trasformata di Esscher è ancora un processo di Lévy. La funzione caratteristica di X_t sotto la nuova misura \mathbb{P}^α è data da (2.13)

$$\phi_t^{(\alpha)} = \mathbb{E}^\alpha[e^{i\xi X_t}] = \mathbb{E}[e^{i\xi X_t} Z_t] = \frac{1}{\phi_t(i\alpha)} \mathbb{E}[e^{i(\xi+i\alpha)X_t}] = \frac{\phi_t(\xi + i\alpha)}{\phi_t(i\alpha)}. \quad (2.16)$$

Qui \mathbb{E}^α indica il valore atteso sotto la misura \mathbb{P}^α . In più, per ogni $0 \leq s < t$ e $Y_t \in \mathcal{F}_t$

$$\mathbb{E}^\alpha[Y_t | \mathcal{F}_s] = \mathbb{E}[Y_t Z_t / Z_s | \mathcal{F}_s]. \quad (2.17)$$

2.2 Opzioni Europee

Si consideri, come esempio illustrativo, un'opzione europea con payoff non negativo $F(S_T)$ al tempo $T > 0$. Si denoti con $V(S, t)$ il prezzo dell'opzione al tempo t , $0 \leq t \leq T$, quando il prezzo del sottostante è $S_t = S > 0$. Applicando la formula neutrale al rischio si ha $V(S, t) = e^{-r\tau} \mathbb{E}_{t,S}[F(S_T)]$, dove $\tau = T - t$ è il tempo

mancante alla scadenza e $\mathbb{E}_{t,S}[\cdot] = \mathbb{E}[\cdot | S_t = S]$. Si introduca $f(x) := F(Ke^x)$ e $v(x, t) := e^{r\tau}V(Ke^x, t)$ (prezzo forward calcolato nella variabile $x = \ln \frac{S}{K}$). In generale, la valutazione prevede funzioni payoff che non sono in $L^1(\mathbb{R})$. Per ovviare a questo problema, si introduce un nuovo spazio con la sua norma associata:

$$L_\alpha^1(\mathbb{R}) := L^1(\mathbb{R}, e^{\alpha x} dx), \quad (2.18)$$

$$\|f\|_{L_\alpha^1(\mathbb{R})} = \int_{\mathbb{R}} |f(x)| e^{\alpha x} dx \text{ con } \alpha \in \mathbb{R}. \quad (2.19)$$

Grazie a questo spazio, il payoff delle opzioni tipicamente appartiene ad $L_\alpha^1(\mathbb{R})$ per qualche $\alpha \in \mathbb{R}$. Per le opzioni call, $f(x) = K(e^x - 1)^+ \in L_\alpha^1(\mathbb{R})$ per ogni $\alpha < -1$. Mentre per le opzioni put, $f(x) = K(1 - e^x)^+ \in L_\alpha^1(\mathbb{R})$, per ogni $\alpha > 0$.

Definizione 2.2.1. Generalmente, si definisce per un payoff $f(x)$ l'insieme

$$\mathcal{I}_f := \{\alpha \in \mathbb{R} : f \in L_\alpha^1(\mathbb{R})\}. \quad (2.20)$$

Per le opzioni vanilla calls si ha, quindi, $\mathcal{I}_{vanillacall} = (-\infty, -1)$, mentre per le opzioni vanilla puts $\mathcal{I}_{vanillaput} = (0, \infty)$.

Si assume inoltre che il payoff dell'opzione sia tale che $\mathcal{I}_f \cap (\lambda_-, \lambda_+)$ sia non vuoto. La relazione di pricing diventa $v(x, t) = P_\tau f(x)$, dove l'operatore di valore atteso in (2.3) ha la rappresentazione di Fourier di (2.12) se vale la (2.11). Il prezzo è infine ottenuto da $V(S, t) = e^{-r\tau}v(\ln \frac{S}{K}, t)$.

- Per le opzioni vanilla calls, $F(S) = (S - K)^+$, $f(x) = K(e^x - 1)^+ \in L_\alpha^1(\mathbb{R})$, con $\alpha < -1$ ($\mathcal{I}_{vanillacall} = (-\infty, -1)$), $f_\alpha(x) = e^{\alpha x} f(x) = K(e^{(\alpha+1)x} - e^{\alpha x})^+ \in L_\alpha^1(\mathbb{R})$ e

$$\hat{f}_\alpha(\xi) = \frac{K}{(\xi - i\alpha)(\xi - i(\alpha + 1))}. \quad (2.21)$$

Inoltre si ha che $\hat{f}_\alpha(\xi) \in L^1(\mathbb{R}, \mathbb{C})$ per ogni $\alpha < -1$, la condizione (2.11) è automaticamente soddisfatta per tutti i processi di Lévy con $\lambda_- < -1$ e la rappresentazione di Fourier (2.12) vale per qualsiasi payoff vanilla call con $\alpha \in (\lambda_-, 1)$.

- Per le opzioni vanilla puts, $F(S) = (K - S)^+$, $f(x) = K(1 - e^x)^+ \in L_\alpha^1(\mathbb{R})$, con $\alpha > 0$ ($\mathcal{I}_{vanillaput} = (0, \infty)$), $f_\alpha(x) = e^{\alpha x} f(x) = K(e^{\alpha x} - e^{(\alpha+1)x})^+ \in L_\alpha^1(\mathbb{R})$ e $\hat{f}_\alpha(\xi)$ è dato sempre dalla formula (2.21), ma ora $\alpha > 0$. Allo stesso modo $\hat{f}_\alpha(\xi) \in L^1(\mathbb{R}, \mathbb{C})$ per ogni $\alpha > 0$, la condizione (2.11) è automaticamente soddisfatta per tutti i processi di Lévy con $\lambda_+ > 0$ e la rappresentazione di Fourier (2.12) vale per le opzioni vanilla puts con $\alpha \in (0, \lambda_+)$. Per prezzare sia le vanilla calls che puts con la rappresentazione di Fourier (2.12), è necessario che $\lambda_- < -1$ e $\lambda_+ > 0$. In seguito si assumerà sempre che i processi di Lévy soddisfino questa assunzione. Per ogni fissato $\alpha \in \mathbb{R}$, si noti che la funzione $\hat{f}_\alpha(z)$ della variabile complessa z in (2.21) ha due poli in $z = i\alpha$ e in $z = i(\alpha + 1)$.

- Per le opzioni con barriera controllata solamente a scadenza, per le up-out Call $F(S) = (S - K)^+ \mathbb{I}\{S < U\}$ e per le down-out put $F(S) = (K - S)^+ \mathbb{I}\{S > L\}$, dove $L < K < U$. In questo caso le funzioni $f(x) = K(e^x - 1)^+ \mathbb{I}\{x < u\}$, $u = \ln \frac{U}{K} > 0$, e $f(x) = K(1 - e^x)^+ \mathbb{I}\{x > l\}$, $l = \ln \frac{L}{K} < 0$, hanno supporto compatto negli intervalli $[0, u]$ e $[l, 0]$ rispettivamente e quindi $f \in L^1_\alpha(\mathbb{R})$, per ogni $\alpha \in \mathbb{R}$ ($\mathcal{I}_f = \mathbb{R}$). In particolare, si pone $\alpha = 0$, ma è conveniente mantenere la dipendenza da α esplicitamente per avere flessibilità di selezionare α al fine di ottimizzare la convergenza dell'algoritmo. La trasformata di Fourier di f_α è

$$\hat{f}_\alpha(\xi) = K \left(\frac{(1 - e^{(i\xi + \alpha)b})}{(i\xi + \alpha)} - \frac{(1 - e^{(i\xi + \alpha + 1)b})}{(i\xi + \alpha + 1)} \right), \quad (2.22)$$

dove $b = u > 0$ per la barriera Call e $b = l < 0$ per la barriera Put. Si noti che \hat{f}_α decade come $|\xi|^{-1}$ per $|\xi| \rightarrow \infty$ e $\hat{f}_\alpha(\xi) \notin L^1(\mathbb{R}, \mathbb{C})$. La rappresentazione di Fourier (2.12) per le opzioni barriera call e barriera put vale solamente per quei processi di Lévy e quelle scadenze tali che sia verificata la (2.11). In particolare, essa vale quando la condizione (2.7) è soddisfatta. Si noti che per la funzione $\hat{f}_\alpha(z)$ di variabile complessa z in (2.22), le singolarità $z = i\alpha$ e $z = i(1 + \alpha)$ sono rimosibili e che quindi $\hat{f}_\alpha(z)$ è una funzione intera in z .

2.2.1 Procedura di pricing

Si analizzano ora, nello specifico, i passaggi necessari per la valutazione. Innanzitutto il sottostante viene modellizzato nel seguente modo:

$$S_t = K e^{X_t},$$

dove $X_0 = 0$.

Sia $V(S, t)$ il prezzo al tempo iniziale t . Vale che:

$$V(S, t) = e^{-r(T-t)} E_{t,S}(F(S_T)).$$

Eseguendo il cambio di variabile $x = \ln(\frac{S}{K})$, per non abusare della notazione:

$$f(x) = F(K e^x).$$

La funzione di pricing andrà calcolata con:

$$v(x, t) = e^{r(T-t)} V(K e^x, t).$$

Si sa che la $v(x, t)$ si calcola come:

$$v(x, t) = \frac{1}{2\pi} e^{-\alpha x - t\Psi(i\alpha)} \int_{\mathbb{R}} e^{-i\xi x} \Phi_t^{(\alpha)}(-\xi) \hat{f}_\alpha(\xi) d\xi, \quad (2.23)$$

dove:

$$e^{-t\Psi(i\alpha)} = \Phi(i\alpha),$$

$$\Phi_t^{(\alpha)}(\xi) = \frac{\Phi_t(\xi + i\alpha)}{\Phi_t(i\alpha)}.$$

Infine si trova il prezzo come:

$$V(S, t) = e^{-r(T-t)} v\left(\ln\left(\frac{S}{K}\right), t\right).$$

Payoffs della Call

Funzione payoff a scadenza:

$$F(S) = (S - K)^+.$$

Si trasforma la funzione nello spazio in cui $x = \ln\left(\frac{S}{K}\right)$:

$$f(x) = K(e^x - 1)^+.$$

Il payoff viene temperato con un coefficiente $\alpha < -1$ per farlo appartenere alla classe $L_\alpha^1(\mathbb{R})$:

$$f(x) = K(e^{(\alpha+1)x} - e^{\alpha x})^+.$$

Dalla (2.21) si conosce la trasformata di Fourier di $f_\alpha(x)$, $\hat{f}_\alpha(x)(\xi)$.

Inversione della trasformata con FFT

Occorre ora calcolare l'integrale della (2.23), l'obiettivo é ottenere il valore non per un solo x ma per piú contemporaneamente. La (2.23) si può vedere come:

$$f(x) = \int_{\mathbb{R}} e^{-i\xi x} g(\xi) d\xi.$$

Su ξ si ha la seguente griglia:

$$\begin{cases} \xi_k = hk \\ k = -M, \dots, M \end{cases}$$

In totale la griglia contiene $2M + 1$ punti. Discretizzando l'integrale si ottiene:

$$f(x) = h \sum_{k=-M}^M w_k e^{-i\xi_k x} g(\xi_k).$$

Tuttavia si sa che nella FFT gli indici partono da zero, si rende quindi necessaria una traslazione:

$$f(x) = h \sum_{k=0}^{2M} w_{k-M} e^{-i\xi_{k-M} x} g(\xi_{k-M}).$$

Si rinomina ora per semplicità i vettori: $W_k = w_{k-M}$ e $\Xi_k = \xi_{k-M}$. Ora nei nuovi vettori gli indici sono $k = 0 : 2M$, di questo fatto si deve tenere conto quando si

introduce una griglia su x , infatti gli indici devono essere gli stessi. La griglia è scelta in modo da soddisfare la relazione di Nyquist, inoltre si vuole avere lo 0 compreso poichè $x = \ln(\frac{S}{K})$ può benissimo essere negativo:

$$\begin{cases} x_j = -M\eta + \eta j = -M\eta + \eta(0 : 2M) = \eta(-M : M) \\ j = 0, \dots, 2M \\ \eta h = \frac{2\pi}{2M+1} \end{cases}$$

Si ottiene quindi:

$$f(x_j) = h \sum_{k=0}^{2M} W_k e^{-i\Xi_k x_j} g(\Xi_k).$$

Per semplicità si riportano i calcoli del solo esponente:

$$\Xi_k x_j = (\eta j - M\eta) h k = \eta j h k - M\eta h k = \frac{2\pi}{2M+1} j k - \frac{2\pi}{2M+1} M k.$$

Moltiplicando per $-i$ si ottiene tutto l'esponente:

$$-i \frac{2\pi}{2M+1} j k + i \frac{2\pi}{2M+1} M k.$$

Poiché il primo termine è messo già dalla funzione FFT in Matlab, occorre mettere solo il secondo, *corr* nel codice:

$$corr = e^{i \frac{2\pi}{2M+1} M k}.$$

Si noti che per il pricing di opzioni Europee non è stato necessario distinguere il caso di sottostante guidato da un processo di Lévy con o senza time change. Questo accade poiché non si deve monitorare l'opzione se non alla maturity, per cui è sufficiente conoscere la funzione caratteristica del processo. Per le opzioni path-dependent che verranno analizzate in seguito questa affermazione non rimane più valida, verranno di conseguenza presentati separatamente i due casi, dove uno rappresenta l'estensione del precedente.

2.2.2 Test numerici

In [Figura 2.1](#) vengono riportati il prezzo e l'andamento dell'errore numerico per opzioni call. Per generare il grafico che rappresenta la convergenza si è ricavato il prezzo dell'opzione usando diversi livelli di troncamento M e lo si è confrontato con il prezzo ottenuto utilizzando la griglia più fitta. Il grafico del prezzo riporta solamente i valori ottenuti con la più fitta delle discretizzazioni utilizzate.

Le opzioni hanno strike price $K = 100$ e maturity $T = 1$, il tasso risk free è $r = 0.05$, mentre il sottostante ha dividend yield $q = 0.02$ ed è scelto con valore iniziale S_0 compreso tra 80 e 120, con passo di 10. Per quanto riguarda i modelli i parametri

utilizzati sono per Black&Scholes $\sigma = 0.2$, per Merton $\sigma = 0.1$, $\lambda = 3$, $\mu = -0.05$, $\delta = 0.086$, per Kou $\sigma = 0.1$, $\lambda = 3$, $p = 0.3$, $\lambda^+ = 40$, $\lambda^- = 12$, per VG $\sigma = 0.1$, $\nu = 0.01$, $\theta = -0.01$, per NIG $\alpha = 15$, $\beta = -5$, $\delta = 0.5$ e per Heston $v_0 = 0.0348$, $\bar{v} = 0.0348$, $\lambda = 1.15$, $\eta = 0.39$ e $\rho = -0.64$. Si ricorda che il significato dei parametri dei processi è dettagliato nella [Sezione 1.2](#) per il modello di Black&Scholes e nelle [Sottosezioni 1.3.4](#), [1.3.5](#), [1.3.6](#), [1.3.7](#) e [1.4.2](#), rispettivamente per i modelli di Merton, Kou, VG, NIG ed Heston. Dai grafici si nota l'andamento esponenziale dell'errore di discretizzazione.

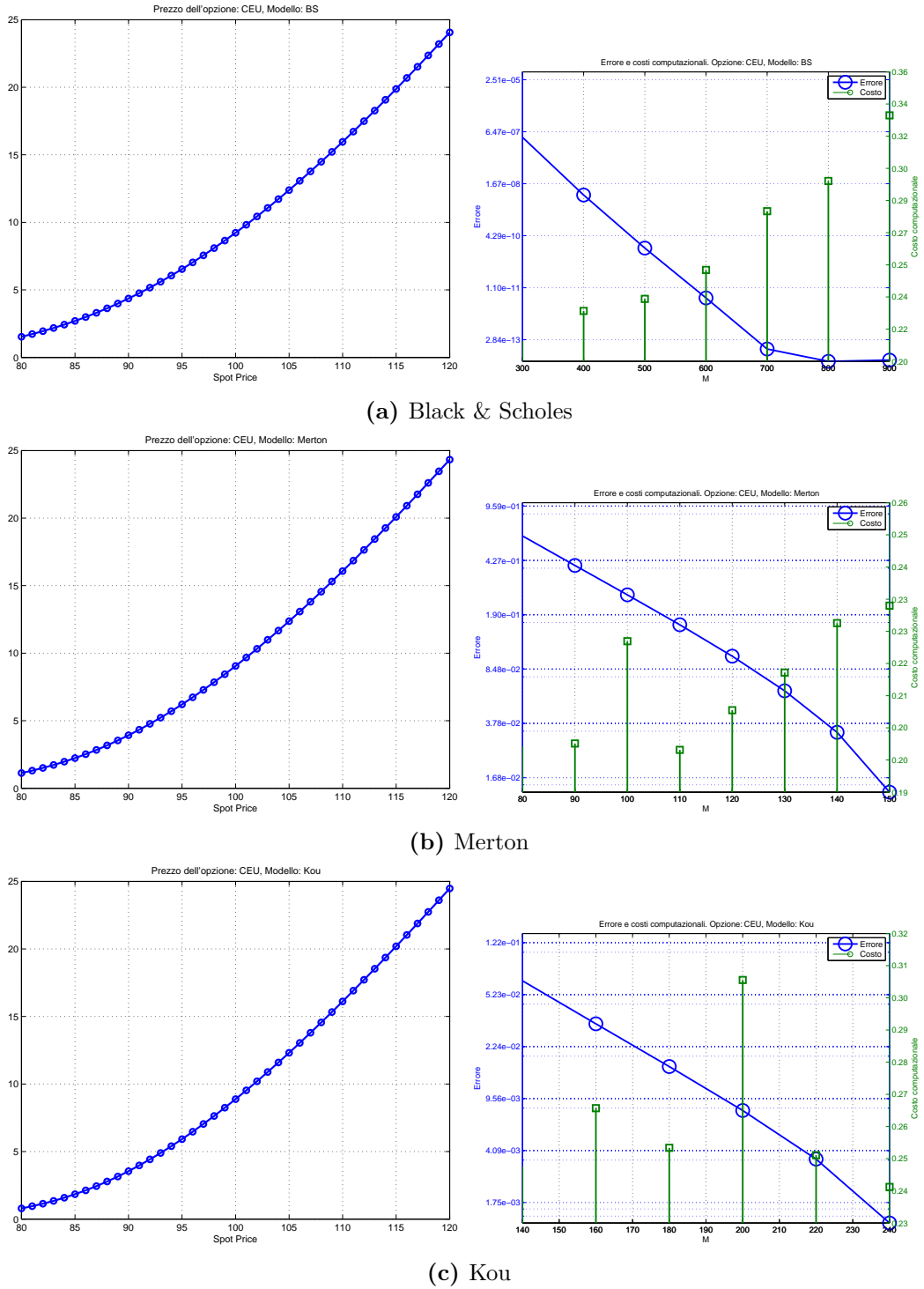
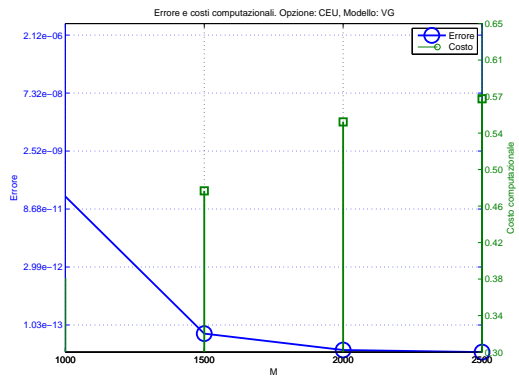
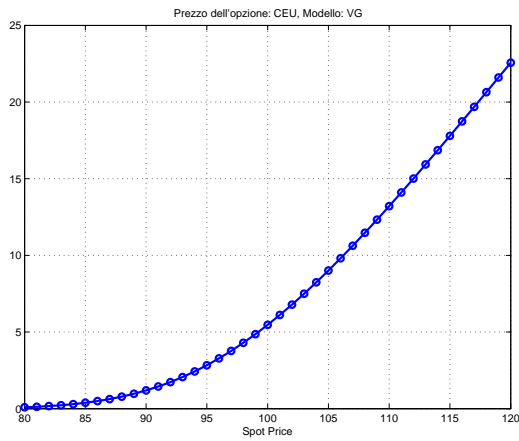
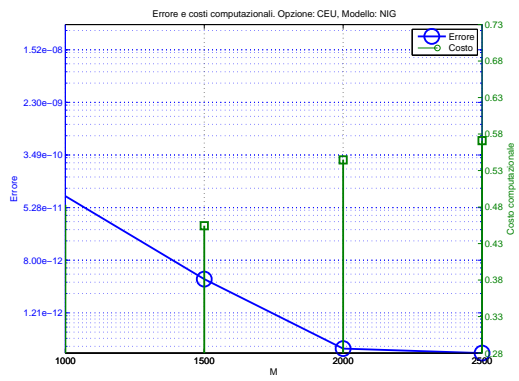
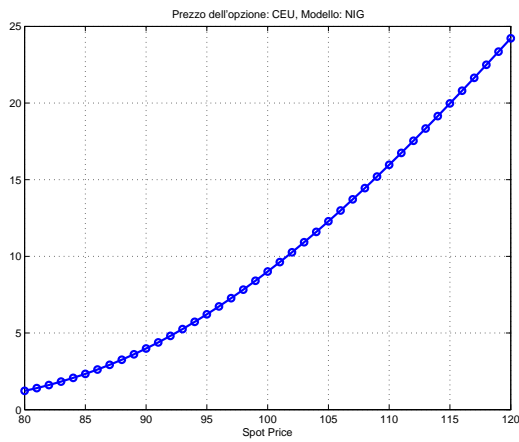


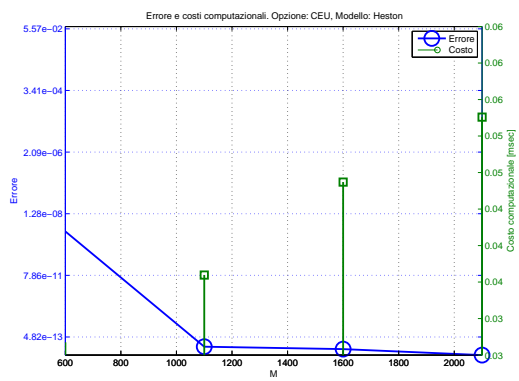
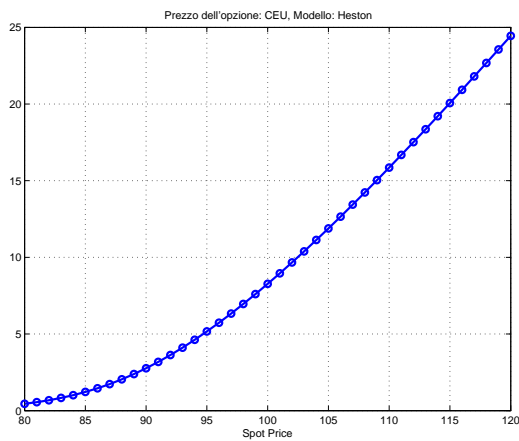
Figura 2.1: Prezzo di un'opzione europea Call, corrispondente errore e costi computazionali al variare dei modelli per il sottostante.



(d) VG



(e) NiG



(f) Heston

Figura 2.1: Prezzo di un'opzione europea Call, corrispondente errore e costi computazionali al variare dei modelli per il sottostante.

2.3 Opzioni Barriera con processi di Lévy

Dopo aver prezzato diversi tipi di opzioni europee, con tutti i possibili modelli per il sottostante, si passa ora al pricing di derivati path dependent. In primo luogo viene affrontata la determinazione del prezzo delle opzioni barriera. A questo scopo è necessario introdurre il concetto di trasformata di Hilbert, la quale si rivelerà molto utile, grazie alle proprietà che la legano alla trasformata di Fourier, per ridurre i costi computazionali. Una sequenza di trasformate viene valutata durante una procedura di induzione backward sugli istanti temporali.

2.3.1 Induzione backward nello spazio di stato

Si vogliono ora considerare le opzioni a monitoraggio discreto down-and-out, up-and-out e a barriera doppia. Un'opzione down-and-out a monitoraggio discreto è specificata dalla maturity $T > 0$, la barriera lower $L > 0$ e il payoff a scadenza $F(S) \geq 0$ per $S > L$ ($F(S) \equiv 0$ per $S \leq L$). Mentre un'opzione up-and-out a monitoraggio discreto è specificata dalla maturity $T > 0$, la barriera upper $U > 0$ e il payoff a scadenza $F(S) \geq 0$ per $S < U$ ($F(S) \equiv 0$ per $S \geq U$). Infine per un'opzione con doppia barriera knock-out a monitoraggio discreto si deve specificare la maturity $T > 0$, le barriere lower e upper L e U ed il payoff è $F(S) \geq 0$ per $S \in (L, U)$ ($F(S) \equiv 0$ per $S \notin (L, U)$). Per questi payoff gli intervalli \mathcal{I}_f e le formule esplicite per la trasformata di Fourier \hat{f}_α sono dati nella [Sezione 2.2](#). Inoltre devono essere specificate le date di monitoraggio della barriera $0 < t_1 < t < \dots < t_N = T$. In generale, l'intervallo temporale tra una data di monitoraggio e la successiva $\{\Delta_j = t_{j+1} - t_j, j = 1, 2, \dots, N-1\}$ non è uniforme. Ma per semplificare la notazione e senza perdita di generalità si considereranno istanti di monitoraggio equispaziati $\Delta_j = \Delta = \frac{T}{N}$, dunque le date di monitoraggio diventeranno $t_j = j\Delta, j = 1, 2, \dots, N$. Si assume che il sottostante segua un processo stocastico $S_t = Ke^{X_t}$, dove X è un processo di Lévy con $\lambda_- < -1, \lambda_+ > 0$ con punto iniziale $x_0 = \ln \frac{S_0}{K}$, dove K è lo strike price ed S_0 è il prezzo iniziale dell'asset. Si introduce la stessa notazione di [Sezione 2.2](#): $V(S, t)$ è la funzione valore (prezzo dell'opzione) al tempo $t \in [0, T]$ e $v(x, t) = e^{r(T-t)}V(Ke^x, t)$ è la funzione valore forward nella variabile $x = \ln \frac{S}{K}$. Si denota invece $v^j(x) := v(x, t_j), j = 0, 1, 2, \dots, N$ come funzione valore forward nelle date $t_j = j\Delta$, dove $t_0 = 0$ è l'istante in cui si valuta l'opzione. Generalmente, la data di valutazione non coincide necessariamente con una delle date di monitoraggio, ma il metodo sviluppato sarà comunque valido per qualsiasi data arbitraria di pricing. Si assume che $t_0 = 0$, senza perdita di generalità. In questo modo si può valutare l'opzione barriera come

$$v^0(x) = \mathbb{E}_x \left[f(X_{N\Delta}) \prod_{j=1}^N \mathbb{I}_I(X_{j\Delta}) \right], \quad (2.24)$$

dove $\mathbb{I}_I(x)$ è la funzione indicatrice dell'intervallo $I = (l, \infty)$ per le opzioni down-and-out, $(-\infty, u)$ per le opzioni up-and-out e (l, u) per le opzioni barriera. L'opzione si annulla se $X_t = \ln(S_t/K)$ non appartiene a I per qualche data di monitoraggio t_j .

Usando la proprietà di Markov e la proprietà di omogeneità nel tempo dei processi di Lévy, l'opzione barriera discreta si valuta con il seguente algoritmo di induzione backward:

$$\begin{aligned} v^N(x) &= f(x), \\ v^{j-1}(x) &= \mathbb{I}_I(x) P_\Delta v^j(x), \quad j = N, N-1, \dots, 2, \\ v^0(x) &= P_\Delta v^1(x), \\ &\text{e} \\ V(S, 0) &= e^{-rT} v^0(\ln(S/K)), \end{aligned}$$

dove $P_t g(x) = \mathbb{E}_x[g(X_t)]$ è il semigruppato markoviano di X . Sia X un processo di Lévy con $\lambda_- < -1$ e $\lambda_+ > 0$ e fissato $\alpha \in (\lambda_-, \lambda_+)$ tale che il payoff $f(x) \in L_\alpha^1(\mathbb{R})$ (si assuma che $\mathcal{I}_f \cap (\lambda_-, \lambda_+)$ sia non vuoto).

In accordo con la trasformata di Esscher, necessaria poiché il payoff non è una funzione integrabile, si ottiene:

$$v_\alpha^N(x) = f_\alpha(x) = e^{\alpha x} f(x), \quad (2.25)$$

$$v_\alpha^{j-1}(x) = e^{-\Delta\Psi(i\alpha)} \mathbb{I}_I(x) P_\Delta^{(\alpha)} v_\alpha^j(x), \quad j = N, N-1, \dots, 2, \quad (2.26)$$

$$v_\alpha^0(x) = e^{-\Delta\Psi(i\alpha)} P_\Delta^{(\alpha)} v_\alpha^1(x), \quad (2.27)$$

$$\begin{aligned} &\text{e} \\ V(S, 0) &= e^{-rT} \left(\frac{S}{K} \right)^{-\alpha} v_\alpha^0(\ln(S/K)). \end{aligned} \quad (2.28)$$

Inoltre ricordando che $v_\alpha^j(x) = e^{\alpha x} v^j(x) \in L^1(\mathbb{R})$ si ha la seguente stima

$$\|v_\alpha^j\|_{L^1(\mathbb{R})} \leq e^{-\Delta(N-j)\Psi(i\alpha)} \|f\|_{L_\alpha^1(\mathbb{R})}. \quad (2.29)$$

per ogni $j = 1, \dots, N$. Quindi si può selezionare lo stesso α per ogni j . Si procede ora con l'implementazione della ricorsione backward nello spazio di Fourier.

2.3.2 Trasformata di Hilbert

La trasformata di Hilbert si rivela essere molto utile per l'efficienza del metodo, consentendo di evitare il passaggio dell'antitrasformazione per controllare la barriera ad ogni passo temporale. Prima di procedere è necessario ricordare alcuni fatti circa la trasformata di Hilbert.

Definizione 2.3.1. Per ogni $f \in L^p(\mathbb{R})$, $1 \leq p < \infty$, la trasformata di Hilbert è ben definita dal valore principale secondo Cauchy del seguente integrale:

$$\mathcal{H}f(x) = \frac{1}{\pi} PV \int_{\mathbb{R}} \frac{f(y)}{x-y} dy. \quad (2.30)$$

Per $1 < p < \infty$ e $g \in L^p(\mathbb{R})$, vale la relazione di parità tra la funzione primitiva $f(x)$ e la sua trasformata $g(x)$

$$g(x) = \frac{1}{\pi} PV \int_{\mathbb{R}} \frac{f(y)}{x-y} dy, \quad f(x) = -\frac{1}{\pi} PV \int_{\mathbb{R}} \frac{g(y)}{x-y} dy. \quad (2.31)$$

Inoltre, quando $1 < p < \infty$ la trasformata di Hilbert è un operatore limitato su $L^p(\mathbb{R})$ con

$$\|\mathcal{H}f\|_{L^p(\mathbb{R})} \leq C_p \|f\|_{L^p(\mathbb{R})}, \quad (2.32)$$

dove $C_p = \tan(\pi/2p)$ se $1 < p \leq 2$ e $C_p = \cot(\pi/2p)$ se $2 \leq p < \infty$. Qui si considereranno trasformate di Hilbert di funzioni a valori complessi $f \in L^p(\mathbb{R}, \mathbb{C})$. Si ricordi che per ogni $f \in L^p(\mathbb{R})$, $1 \leq p < \infty$, la sua trasformata di Fourier $\hat{f} = \mathcal{F}f \in L^q(\mathbb{R}, \mathbb{C})$, con $1/p + 1/q = 1$.

Legame con la trasformata di Fourier La trasformata di Hilbert è in stretta relazione con la trasformata di Fourier; vale la seguente relazione per ogni $f \in L^p(\mathbb{R})$ con $1 < p < \infty$ o con $p = 1$ se in aggiunta si ha $\hat{f} \in L^1(\mathbb{R}, \mathbb{C})$:

$$\mathcal{F}(\text{sgn} \cdot f)(\xi) = i \mathcal{H}\hat{f}(\xi), \quad (2.33)$$

dove $\text{sgn}(x)$ è la funzione segno e $\mathcal{H}\hat{f} \in L^q(\mathbb{R}, \mathbb{C})$, $1/p + 1/q = 1$.

Scrivendo

$$\mathbb{I}_{(0,\infty)} = \frac{1}{2}[1 + \text{sgn}(x)], \quad (2.34)$$

si ottiene la seguente relazione:

$$\mathcal{F}(\mathbb{I}_{(0,\infty)} \cdot f)(\xi) = \frac{1}{2}\hat{f}(\xi) + \frac{i}{2}\mathcal{H}\hat{f}(\xi). \quad (2.35)$$

Denotando l'operatore di traslazione con \mathcal{T}_a , $(\mathcal{T}_a f)(x) = f(x-a)$.

Allora

$$\mathbb{I}_{(l,\infty)} = \mathcal{T}_l \mathbb{I}_{(0,\infty)} = \frac{1}{2}(1 + \mathcal{T}_l \text{sgn}) \quad (2.36)$$

e

$$\mathbb{I}_{(l,\infty)} \cdot f = \frac{1}{2}f + \frac{1}{2}f \mathcal{T}_l \text{sgn} = \frac{1}{2}f + \frac{1}{2}\mathcal{T}_l(\text{sgn} \cdot \mathcal{T}_{-l}f). \quad (2.37)$$

Applicando la trasformata di Fourier ad entrambi i lati e usando (2.33), si ottiene:

$$\mathcal{F}(\mathbb{I}_{(l,\infty)} \cdot f)(\xi) = \frac{1}{2}\hat{f}(\xi) + \frac{1}{2}\mathcal{F}(\mathcal{T}_l(\text{sgn} \cdot \mathcal{T}_{-l}f))(\xi) = \frac{1}{2}\hat{f}(\xi) + \frac{i}{2}e^{i\xi l} \mathcal{H}(e^{-i\eta l} \hat{f}(\eta))(\xi). \quad (2.38)$$

Notando che $\mathbb{I}_{(-\infty, u)} = 1 - \mathbb{I}_{[u, \infty)}$, si ricava la seguente:

$$\mathcal{F}(\mathbb{I}_{(-\infty, u)} \cdot f)(\xi) = \frac{1}{2}\hat{f}(\xi) - \frac{i}{2}e^{i\xi u} \mathcal{H}(e^{-i\eta u} \hat{f}(\eta))(\xi). \quad (2.39)$$

Infine, riscrivendo $\mathbb{I}_{(l,u)}$ come $\mathbb{I}_{(l,\infty)} - \mathbb{I}_{(-\infty,u)}$, si ha per il caso di intervallo finito:

$$\begin{aligned} \mathcal{F}(\mathbb{I}_{(l,u)} \cdot f)(\xi) &= \frac{i}{2} e^{i\xi l} \mathcal{H}(e^{-i\eta l} \hat{f}(\eta))(\xi) - \frac{i}{2} e^{i\xi u} \mathcal{H}(e^{-i\eta u} \hat{f}(\eta))(\xi) \\ &= \int_{-\infty}^{\infty} \hat{f}(\eta) \frac{e^{i u(\xi-\eta)} - e^{i l(\xi-\eta)}}{2\pi i(\xi-\eta)} d\eta \\ &= \int_{-\infty}^{\infty} \hat{f}(\eta) e^{i(\xi-\eta)(l+u)/2} \frac{\sin((\xi-\eta)(u-l)/2)}{\pi(\xi-\eta)} d\eta, \end{aligned} \quad (2.40)$$

dove PV può essere tolto in quanto $\lim_{x \rightarrow 0} (\sin x)/x = 1$.

2.3.3 Induzione backward nello spazio di Fourier: rappresentazione con trasformata di Hilbert

Ora si può formulare l'induzione backward nello spazio di Fourier, come dettagliata in [9].

Teorema 2.2. *Sia $\Delta > 0$ l'intervallo di monitoraggio della barriera. Sia X un processo di Lévy con funzione caratteristica $\phi_t(\xi)$ con $\lambda_- < -1$ e $\lambda_+ > 0$. Fissato $\alpha \in (\lambda_-, \lambda_+)$ tale che il payoff dell'opzione $f \in L^1_{\alpha}(\mathbb{R})$ (si assume che $\mathcal{I}_f \cap (\lambda_-, \lambda_+)$ sia non vuoto) e assumendo che il payoff soddisfi (2.11) per $t = \Delta$. Assumendo inoltre che esista un tempo $\mathcal{X} > 0$ tale che*

$$\|\phi_{\mathcal{X}}(\cdot + i\alpha)\|_{L^1} \equiv \int_{\mathbb{R}} |\phi_{\mathcal{X}}(\xi + i\alpha)| d\xi < \infty \quad (2.41)$$

(se la condizione (2.41) è soddisfatta per qualche $\mathcal{X} > 0$, allora è soddisfatta per tutti $t \geq \mathcal{X}$). Sia $\hat{f}_{\alpha}(\xi)$ la trasformata di Fourier di $e^{\alpha x} f(x)$. Siano $\hat{v}_{\alpha}^j(\xi)$, $j = 1, \dots, N$ le trasformate di Fourier di $v_{\alpha}^j(x) = e^{\alpha x} v^j(x)$. Allora l'induzione a ritroso (2.26) nel dominio di Fourier, partendo da $\hat{v}_{\alpha}^N(\xi) = \hat{f}_{\alpha}(\xi)$, diventa:

(i) *Opzione Down-and-Out*

$$\hat{v}_{\alpha}^{j-1}(\xi) = \frac{1}{2} e^{-\Delta\Psi(i\alpha)} \left[\phi_{\Delta}^{(\alpha)}(-\xi) \hat{v}_{\alpha}^j(\xi) + i e^{i\xi l} \mathcal{H} \left(e^{-i\eta l} \phi_{\Delta}^{(\alpha)}(-\eta) \hat{v}_{\alpha}^j(\eta) \right) (\xi) \right], \quad (2.42)$$

(ii) *Opzione Up-and-Out*

$$\hat{v}_{\alpha}^{j-1}(\xi) = \frac{1}{2} e^{-\Delta\Psi(i\alpha)} \left[\phi_{\Delta}^{(\alpha)}(-\xi) \hat{v}_{\alpha}^j(\xi) - i e^{i\xi u} \mathcal{H} \left(e^{-i\eta u} \phi_{\Delta}^{(\alpha)}(-\eta) \hat{v}_{\alpha}^j(\eta) \right) (\xi) \right], \quad (2.43)$$

(iii) *Opzione a barriera doppia*

$$\hat{v}_{\alpha}^{j-1}(\xi) = e^{-\Delta\Psi(i\alpha)} \int_{\mathbb{R}} \phi_{\Delta}^{(\alpha)}(-\eta) \hat{v}_{\alpha}^j(\xi) e^{i(\xi-\eta)(l+u)/2} \frac{\sin((\xi-\eta)(u-l)/2)}{\pi(\xi-\eta)} d\eta. \quad (2.44)$$

Allo step finale $v_{\alpha}^0(x)$ è calcolato tramite la rappresentazione di Fourier di $P_{\Delta}^{(\alpha)} v_{\alpha}^1(x)$:

$$v_{\alpha}^0(x) = \frac{1}{2\pi} e^{-\Delta\Psi(i\alpha)} \int_{\mathbb{R}} e^{-ix\xi} \phi_{\Delta}^{(\alpha)}(-\xi) \hat{v}_{\alpha}^1(\xi) d\xi, \quad (2.45)$$

a partire da $\hat{v}_{\alpha}^1(\xi)$ ottenuta al penultimo step. Il prezzo è infine ricavato dalla (2.28).

Per la sua dimostrazione si faccia riferimento a [9].

2.3.4 Aproxsimazione discreta

Le trasformate di Hilbert in (2.42) (2.43), la convoluzione in (2.44) e l'inversione finale di Fourier in (2.45) devono essere valutate numericamente. L'approssimazione di esse è ricavata in [9]. Riassumendo brevemente, si introducono le funzioni appartenenti a $W(\pi/h)$, cioè le funzioni intere di tipo esponenziale: analitiche nell'intero piano complesso tali che $\int_{\mathbb{R}} |f(x)|^2 dx < \infty$ e tali che per ogni $z \in \mathbb{C}$ la funzione ha un andamento esponenziale $|f(z)| \leq Ce^{\pi|z|/h}$. Lo spazio $W(\pi/h)$ è chiamato spazio di Wiener delle funzioni intere di tipo esponenziale. Per queste funzioni si ha che esiste una sequenza ortonormale $\{h^{-1/2}S(k, h)(z)\}_{k=-\infty}^{\infty}$ in $W(\pi/h)$ tale che ogni $f \in W(\pi/h)$ ha l'espansione

$$f(z) = \sum_{k=-\infty}^{\infty} f(kh)S(k, h)(z) = h \sum_{k=-\infty}^{\infty} f(kh) \frac{\sin[\pi(z - kh)/h]}{\pi(z - kh)}. \quad (2.46)$$

Questa espansione è chiamata *rappresentazione Sinc* ed è una formula di quadratura esatta per le funzioni intere di tipo esponenziale. In modo simile, per le funzioni di questo spazio la trasformata di Hilbert può essere valutata esattamente in $W(\pi/h)$

$$\mathcal{H}f(x) = \frac{1}{\pi} P.V. \int_{-\infty}^{\infty} \frac{f(y)}{x - y} dy = \sum_{k=-\infty}^{\infty} f(kh) \frac{1 - \cos[\pi(x - kh)/h]}{\pi(x - kh)/h}. \quad (2.47)$$

Sebbene l'espansione Sinc sia esatta solo per le funzioni appartenenti a $W(\pi/h)$, essa costituisce il mattone della teoria di approssimazione per funzioni analitiche anche solo in uno strip contenente l'asse reale. Sia $d > 0$ e $\mathcal{D}_d = \{z \in \mathbb{C} : |\Im(z)| < d\}$, si denota con $H^1(\mathcal{D}_d)$ lo spazio di Hardy di funzioni analitiche in uno strip \mathcal{D}_d contenente l'asse reale e tali che $\int_{-d}^d f(x + iy) dy \rightarrow 0$ quando $x \rightarrow \infty$. Per queste funzioni l'espansione di f è data da (2.46), mentre l'espansione di $\mathcal{H}f$ è data da (2.47). Tuttavia tali espansioni sono approssimazioni: la formula non è esatta ma vi è un errore che decade esponenzialmente in d/h .

Applicando i risultati dell'approssimazione Sinc si ottengono dunque i seguenti risultati.

$$\begin{aligned} \hat{v}_{\alpha, h, M}^{j-1}(\xi) &= \frac{1}{2} e^{-\Delta\Psi(i\alpha)} (\phi_{\Delta}^{(\alpha)}(-\xi) \hat{v}_{\alpha}^j(\xi) + \dots \\ &+ i\theta e^{i\xi b} \sum_{m=-M}^M e^{-imhb} \phi_{\Delta}^{(\alpha)}(-mh) \hat{v}_{\alpha, h, M}^j(mh) \frac{1 - \cos[\pi(\xi - mh)/h]}{\pi(\xi - mh)/h}), \quad \xi \in \mathbb{R}. \end{aligned} \quad (2.48)$$

La (2.48) è valida sia per le opzioni down-out con $b = l, \theta = 1$ che per le opzioni up-out con $b = u, \theta = -1$.

$$\begin{aligned} \hat{v}_{\alpha, h, M}^{j-1}(\xi) &= e^{-\Delta\Psi(i\alpha)} \dots \\ &\cdot h \sum_{m=-M}^M e^{i(\xi - mh)(u+l)/2} \phi_{\Delta}^{(\alpha)}(-mh) \hat{v}_{\alpha, h, M}^j(mh) \frac{\sin[(\xi - mh)(u - l)/2]}{\pi(\xi - mh)}, \quad \xi \in \mathbb{R}, \end{aligned} \quad (2.49)$$

$$(2.50)$$

La (2.49) vale per le opzioni a barriera doppia.

$$v_{\alpha,h,M}^0(\xi) = \frac{1}{2\pi} e^{-\Delta\Psi(i\alpha)} h \sum_{m=-M}^M e^{-ixmh} \phi_{\Delta}^{(\alpha)}(-mh) \hat{v}_{\alpha,h,M}^1(mh), \quad x \in \mathbb{R}. \quad (2.51)$$

La (2.51) è la discretizzazione di (2.45).

Per quanto riguarda l'errore di discretizzazione valgono i seguenti teoremi dimostrati in [9].

Teorema 2.3. *Si assuma che la funzione caratteristica $\phi_t(\xi) = e^{-t\Psi(\xi)}$ soddisfi la stima (2.8) per ogni ξ reale per qualche $\nu \in (0, 2]$, $c > 0$, $k > 0$, e $\|e^{-\Delta\Psi(i\alpha)}\|_{L^1} < \infty$. Si supponga inoltre che $\hat{v}_{\alpha}(\xi)$ sia analitica e limitata in \mathcal{D}_d . Si ha che ponendo*

$$h = \left(\frac{\pi d}{\Delta c} \right)^{\frac{1}{1+\nu}} M^{-\frac{\nu}{1+\nu}}. \quad (2.52)$$

Allora esiste un $C > 0$ indipendente da M tale che

$$\|\hat{v}_{\alpha,h,M} - \hat{v}\|_{L^\infty} \leq CM^{\frac{1}{1+\nu}} \exp\left(-(\Delta c)^{\frac{1}{1+\nu}} (\pi d M)^{\frac{\nu}{1+\nu}}\right). \quad (2.53)$$

Il Teorema 2.3 vale per le opzioni a barriera singola. Mentre per le opzioni a barriera doppia vale il seguente

Teorema 2.4. *Sotto le stesse assunzioni del Teorema 2.3, ponendo*

$$h = \left(\frac{2\pi d}{\Delta c} \right)^{\frac{1}{1+\nu}} M^{-\frac{\nu}{1+\nu}}. \quad (2.54)$$

Allora esiste un $C > 0$ indipendente da M tale che

$$\|\hat{v}_{\alpha,h,M} - \hat{v}\|_{L^\infty} \leq C \max(1, M^{\frac{1-\nu}{1+\nu}}) \exp\left(-(\Delta c)^{\frac{1}{1+\nu}} (2\pi d M)^{\frac{\nu}{1+\nu}}\right). \quad (2.55)$$

Teorema 2.5. *Sotto le stesse assunzioni del Teorema 2.3, per i contratti a barriera singola, ponendo h in accordo a (2.52), allora esiste un $C > 0$ indipendente da M tale che*

$$|v_{\alpha,h,M}^0 - v| \leq C \max(1, M^{\frac{1-\nu}{1+\nu}}) \exp\left(-(\Delta c)^{\frac{1}{1+\nu}} (2\pi d M)^{\frac{\nu}{1+\nu}}\right). \quad (2.56)$$

Mentre per i contratti a barriera doppia, ponendo h in accordo con (2.54), esiste un $D > 0$ indipendente da M tale che

$$|v_{\alpha,h,M}^0 - v| \leq D \max(1, M^{\frac{1-\nu}{1+\nu}}) \exp\left(-(\Delta c)^{\frac{1}{1+\nu}} (2\pi d M)^{\frac{\nu}{1+\nu}}\right). \quad (2.57)$$

La stima (2.8) non vale per il processo VG: la sua funzione caratteristica ha un andamento polinomiale $|\phi_t(\xi)| \leq k|\xi|^{-2t/\nu}$. I Teoremi 2.3, 2.4, 2.5 si possono estendere a mostrare che l'errore di discretizzazione è ancora $O(\exp(-c/h))$. Tuttavia l'errore di troncamento decade solo polinomialmente a causa dell'andamento solo polinomiale della funzione caratteristica.

Teorema 2.6. *Si supponga che $\Delta > \nu/2$ nel modello VG. Allora esistono $A, \mathcal{A}, B, \mathcal{B}, C, \mathcal{C} > 0$ indipendenti da h e M tale che per i contratti a barriera singola*

$$\|\hat{v}_{\alpha,h,M} - \hat{v}\|_{L^\infty} \leq A \frac{e^{-\pi d/h}}{\pi d(1 - e^{-\pi d/h})} + \mathcal{A}h^{-1}(Mh)^{-(2\Delta/\nu-1)}. \quad (2.58)$$

Mentre per i contratti a barriera doppia

$$\|\hat{v}_{\alpha,h,M} - \hat{v}\|_{L^\infty} \leq B \frac{e^{-2\pi d/h}}{(1 - e^{-2\pi d/h})} + \mathcal{B}(Mh)^{-(2\Delta/\nu-1)}. \quad (2.59)$$

Infine l'errore allo step finale è

$$|v_{\alpha,h,M}^0 - v| \leq C \frac{e^{-2\pi d/h}}{(1 - e^{-2\pi d/h})} + \mathcal{C}(Mh)^{-(2\Delta/\nu-1)}. \quad (2.60)$$

Nel caso particolare in cui il sottostante abbia dinamica modellizzata con Black&Scholes valgono i seguenti teoremi dimostrati in [9].

Teorema 2.7. *Per i contratti a barriera singola ponendo*

$$h = \left(\frac{\pi^2}{\Delta\sigma^4 T} \right)^{\frac{1}{4}} M^{-\frac{1}{2}} \quad (2.61)$$

Allora esistono $C > 0$ e $D > 0$ indipendenti da M tale che

$$\|\hat{v}_{\alpha,h,M} - \hat{v}\|_{L^\infty} \leq C \exp\left(-\sqrt{\frac{\pi^2\Delta}{4T}}M\right). \quad (2.62)$$

$$|v_{\alpha,h,M}^0 - v| \leq D \exp\left(-\sqrt{\frac{\pi^2\Delta}{4T}}M\right). \quad (2.63)$$

e ponendo per i contratti a barriera doppia

$$h = \left(\frac{4\pi^2}{\Delta\sigma^4 T} \right)^{\frac{1}{4}} M^{-\frac{1}{2}} \quad (2.64)$$

Allora esistono $C > 0$ e $D > 0$ indipendenti da M tale che

$$\|\hat{v}_{\alpha,h,M} - \hat{v}\|_{L^\infty} \leq C \exp\left(-\sqrt{\frac{\pi^2\Delta}{T}}M\right). \quad (2.65)$$

$$|v_{\alpha,h,M}^0 - v| \leq D \exp\left(-\sqrt{\frac{\pi^2\Delta}{T}}M\right). \quad (2.66)$$

2.3.5 Computazione: algoritmo FHT

Il processo di Lévy del sottostante è il processo trasformato di Esscher $X^{(\alpha)}$ con un appropriato $\alpha \in (\lambda_-, \lambda_+)$ tale che l'opzione payoff $f \in L^1_\alpha(\mathbb{R})$. Ricordando che $\phi_t(z)$ è analitica nello strip $\{z \in \mathbb{C} : \Im(z) \in (\lambda_-, \lambda_+)\}$ contenente l'asse reale. Allora $\phi_t^{(\alpha)}(-z) = \phi_t(-z + i\alpha)e^{t\Psi(i\alpha)}$ con $\alpha \in (\lambda_-, \lambda_+)$ è analitica nello strip $\{z \in \mathbb{C} : \Im(z) \in (\alpha - \lambda_+, \alpha - \lambda_-)\}$ contenente l'asse reale. Le trasformate di Fourier $\hat{f}_\alpha(z)$ dei payoff considerati sono analitiche in uno strip contenente l'asse reale. Perciò la funzione $\phi_\Delta^{(\alpha)}(-z)\hat{f}_\alpha(z)$ è anch'essa analitica in uno strip contenente l'asse reale. Nello specifico, per il payoff vanilla call ($\alpha \in (\lambda_-, -1)$), per il payoff vanilla put ($\alpha \in (\lambda_-, 0)$), gli strip di analiticità per le funzioni $\phi_\Delta^{(\alpha)}(-z)\hat{f}_\alpha(z)$ sono:

- $\mathcal{S}_{\text{vanilla call}} = \{z \in \mathbb{C} : \Im(z) \in (\alpha + 1, \alpha - \lambda_-)\}$,
- $\mathcal{S}_{\text{vanilla put}} = \{z \in \mathbb{C} : \Im(z) \in (\alpha - \lambda_+, \alpha)\}$,
- $\mathcal{S}_{\text{barriera call}} = \mathcal{S}_{\text{barriera put}} = \{z \in \mathbb{C} : \Im(z) \in (\alpha - \lambda_+, \alpha - \lambda_-)\}$.

Generalmente, dato un processo di Lévy X , un payoff f e $\alpha \in (\lambda_-, \lambda_+)$ tale che $f \in L^1_\alpha(\mathbb{R})$, si denota lo strip di analiticità di $\phi_\Delta^{(\alpha)}(-z)\hat{f}_\alpha(z)$ con $S_{X,f}^{(\alpha)} = \{z \in \mathbb{C} : \Im(z) \in (d_-, d_+)\}$. Si seleziona $\alpha \in (\lambda_-, \lambda_+)$ in modo che $S_{X,f}^{(\alpha)}$ sia simmetrica intorno all'asse reale, $-d_- = d_+ = d$. Questo dà $\alpha = (\lambda_- - 1)/2$ e $d = -(\lambda_- + 1)/2$ per vanilla call, $\alpha = \lambda_+/2$ e $d = \lambda_+/2$ per vanilla put, $\alpha = (\lambda_+ + \lambda_-)/2$ e $d = (\lambda_+ - \lambda_-)/2$ per il payoff troncato put e call. Quindi la funzione $\phi_\Delta^{(\alpha)}(-z)\hat{f}_\alpha(z)$ è analitica in $\mathcal{D}_d := \{z \in \mathbb{C} : |\Im(z)| \leq d\}$. Inoltre, se $g(z)$ è analitica in \mathcal{D}_d , allora la sua trasformata di Hilbert $(\mathcal{H}g)(\xi)$ può essere analiticamente continua dall'asse reale allo strip \mathcal{D}_d . Da qui si ha anche che tutte le $\hat{v}_\alpha(z)$ sono analitiche in \mathcal{D}_d . In aggiunta anche le $\hat{v}_\alpha(z)$ sono limitate in \mathcal{D}_d . Discretizzando anche la variabile ξ la ricorsione per le opzioni a singola barriera diventa:

$$\hat{v}_{\alpha,M}^N(kh) = \hat{f}_\alpha(kh) \quad \text{per } k = -M, \dots, M, \quad (2.67)$$

$$\begin{aligned} \hat{v}_{\alpha,M}^{j-1}(kh) &= \frac{1}{2}e^{-\Delta\Psi(i\alpha)}\phi_\Delta^{(\alpha)}(-kh)\hat{v}_{\alpha,M}^j(kh) + \dots \\ &+ \frac{i\theta}{2\pi}e^{-\Delta\Psi(i\alpha)} \sum_{m=-M, m \neq M}^M e^{ikhb}\phi_\Delta^{(\alpha)}(-mh)\hat{v}_{\alpha,M}^j(mh)\frac{1 - (-1)^{k-m}}{k-m} \end{aligned} \quad (2.68)$$

per $k = -M, \dots, M$

per $j = N, N-1, \dots, 2$ ($\theta = 1$ e $b = l$ per le opzioni Down-and-Out e $\theta = -1$ e $b = u$ per le opzioni Up-and-Out), e

$$v_{\alpha,M}^0(x) = \frac{1}{2\pi}e^{-\Delta\Psi(i\alpha)} \sum_{m=-M}^M e^{-imhx}\phi_\Delta^{(\alpha)}(-mh)\hat{v}_{\alpha,M}^1(mh)h. \quad (2.69)$$

L'approssimazione $V_M(S, 0)$ del prezzo dell'opzione $V(S, 0)$ al tempo zero quando lo spot price è S si ottiene tramite (2.28) da $v_{\alpha,M}^0(x)$ con $x = \ln(S/K)$. Per ottenere la trasformata discreta di Hilbert dall'approssimazione Sinc

$$\mathcal{H}_{h,M}f(kh) = \frac{1}{\pi} \sum_{m=-M, m \neq k}^M f(mh)\frac{1 - (-1)^{k-m}}{k-m} \quad (2.70)$$

si noti che

$$\frac{1 - \cos(\pi(kh - mh)/h)}{\pi(kh - mh)/h} = \begin{cases} \frac{1 - (-1)^{k-m}}{k-m}, & m \neq k \\ 0 & m = k \end{cases} \quad (2.71)$$

Per le opzioni a doppia barriera, l'induzione backward (2.68) è sostituita con

$$\begin{aligned} \hat{v}_{\alpha, M}^{j-1}(kh) &= e^{-\Delta\Psi(i\alpha)} e^{ikh(u+l)/2} \sum_{m=-M, m \neq M}^M e^{-imh(u+l)/2} \phi_{\Delta}^{(\alpha)}(-mh) \hat{v}_{\alpha, M}^j(mh) \dots \\ &\frac{\sin(h(k-m)(u-l)/2)}{\pi(k-m)} + e^{-\Delta\Psi(i\alpha)} \frac{(u-l)}{2\pi} \phi_{\Delta}^{(\alpha)}(-kh) \hat{v}_{\alpha, M}^j(kh)h, \quad (2.72) \\ &\text{per } k = -M, \dots, M, \text{ e } j = N, N-1, \dots, 2. \end{aligned}$$

Per prezzare le opzioni a singola barriera, ad ogni step temporale si deve valutare la trasformata discreta di Hilbert in (2.68) di un vettore complesso. Mentre per prezzare le opzioni a barriera doppia, ad ogni step temporale si deve valutare la convoluzione in (2.72). In entrambi i casi, la computazione comporta una moltiplicazione matrice vettore di Toeplitz. Una matrice T è detta matrice di Toeplitz se $T_{n,m} = T_{n+1,m+1}$ dove $T_{n,m}$ denota l' $(n, m)^{th}$ elemento. Il vantaggio è che il calcolo della moltiplicazione matrice-vettore di Toeplitz può essere incorporato in una matrice circolante. Mentre il costo computazionale della usuale moltiplicazione matrice-vettore è $O(M^2)$, la moltiplicazione di una matrice circolante per un vettore può essere implementato usando FHT con costo di $O(M \log_2 M)$. Per le opzioni a barriera singola la matrice di Toeplitz ha la seguente struttura:

$$T_{k,m} = \begin{cases} \frac{1 - (-1)^{k-m}}{k-m}, & m \neq k \\ 0 & m = k \end{cases} \quad (2.73)$$

e per le opzioni a barriera doppia:

$$T_{k,m} = \begin{cases} \frac{\sin(h(k-m)(u-l)/2)}{\pi(k-m)}, & m \neq k \\ \frac{(u-l)h}{2\pi} & m = k \end{cases} \quad (2.74)$$

Ad ogni passo temporale la moltiplicazione matrice-vettore di Toeplitz impiega $O(M \log_2 M)$ operazioni usando FFT. Il calcolo di (2.69) per un valore $x = \ln(S/K)$ è dato in $O(M)$ operazioni. Se si desidera ottenere i prezzi simultaneamente in corrispondenza di più spot price si può utilizzare FFT, compiendo lo step finale in $O(M \log_2 M)$ operazioni. In conclusione, l'algoritmo impiega $O(NM \log_2 M)$. Riassumendo i passi si procede con

1. **Preparazione** Dato il payoff $f(x)$ ($x = \ln(S/K)$) e dato un processo di Lévy X , si seleziona $\alpha \in (\lambda_-, \lambda_+)$ tale che $f \in L_{\alpha}^1(\mathbb{R})$ e $-d_- = d_+ = d$ e si calcola la trasformata di Fourier $\hat{f}_{\alpha}(\xi) = (\mathcal{F}f\alpha)(\xi)$. Si fissa un intero $M > 0$ e si seleziona un passo di discretizzazione appropriato $h = h(M)$. Si prepara infine $\{\phi_{\Delta}^{(\alpha)}(-mh)\}_{m=-M}^M$ e $\{\hat{v}_{\alpha}^N(mh) = \hat{f}_{\alpha}(mh)\}_{m=-M}^M$.

2. **Induzione backward** Pero ogni $j = N, N-1, \dots, 2$, per le opzioni a barriera singola si ricava il vettore $\{\hat{v}_{\alpha, M}^{j-1}(mh)\}_{m=-M}^M$ da $\{\hat{v}_{\alpha, M}^j(mh)\}_{m=-M}^M$ tramite la (2.68) calcolando la FHT via matrice Toeplitz (per le opzioni a barriera doppia tramite la (2.72)).
3. **Inversione finale di Fourier** Si calcola l'inversione finale (2.69).

2.3.6 Moltiplicazione Matrice-Vettore di Toeplitz via FFT

Come già annunciato precedentemente, al fine di rendere il costo computazionale di ogni step dell'algoritmo dell'ordine di $O(M \log M)$ invece che il consueto $O(M^2)$ per una moltiplicazione classica matrice-vettore, è opportuno sfruttare la struttura particolare della matrice di Toeplitz. Infatti è possibile effettuare la moltiplicazione per vettore servendosi dell'algoritmo FFT per abbattere significativamente i tempi macchina.

La trasformata discreta di Fourier (DFT) $(\hat{f}_n)_{n=0}^{N-1}$ di un vettore $(f_n)_{n=0}^{N-1}$ è definita come segue

$$\hat{f}_k = \sum_{n=0}^{N-1} e^{-\frac{2\pi i nk}{N}} f_n, \quad k = 0, 1, \dots, N-1. \quad (2.75)$$

Sia w la radice N -esima dell'unità, $w = e^{-\frac{2\pi i}{N}}$ e si introduca la matrice \mathbb{F} :

$$\mathbb{F} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w^1 & w^2 & \dots & w^{N-1} \\ 1 & w^2 & w^4 & \dots & w^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \dots & w^{(N-1)(N-1)} \end{pmatrix}. \quad (2.76)$$

Dunque la DFT può essere scritta nella forma: $\hat{f} = \mathbb{F}f$. La trasformata inversa discreta di Fourier (IDFT) è

$$f_n = \frac{1}{N} \sum_{k=0}^{N-1} e^{\frac{2\pi i nk}{N}} \hat{f}_k, \quad n = 0, 1, \dots, N-1, \quad (2.77)$$

o nella forma matriciale, $f = \mathbb{F}^{-1}\hat{f}$, dove la matrice della trasformata inversa di DFT è $\mathbb{F}^{-1} = \frac{1}{N}\mathbb{F}^H$ e $F^H = \bar{F}^T$ è la hermitiana di \mathbb{F} . L'algoritmo FHT è efficiente per calcolare DFT (e IDFT) in $O(N \log_2 N)$ in confronto a N^2 richiesto dalla moltiplicazione standard matrice-vettore. Una matrice $N \times N$ si dice *circolante* se ha la seguente forma:

$$C = \begin{pmatrix} c_0 & c_{N-1} & \dots & c_1 \\ c_1 & c_0 & \dots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ c_{N-1} & c_{N-2} & \dots & c_0 \end{pmatrix}. \quad (2.78)$$

Essa è completamente specificata dalla prima colonna $c = (c_0, \dots, c_{N-1})^T$ ed ogni colonna successiva è ottenuta facendo uno shift in basso della precedente. Una matrice circolante è diagonalizzabile da una matrice DFT ($C = \mathbb{F}^{-1}\Delta\mathbb{F}$, dove Δ è la matrice diagonale contenente gli autovalori di C : $\Delta = \text{diag}(\mathbb{F}_c)$). Questa

fattorizzazione può essere utilizzata per implementare la moltiplicazione matrice-vettore. Sia $x = (x_n)_{n=0}^{N-1}$ un vettore N -dimensionale e $C = (C_{n,m})_{n,m=0}^{N-1}$ una matrice circolante $N \times N$. Allora

$$Cx = \mathbb{F}^{-1} \Delta \mathbb{F} x = \mathbb{F}^{-1} (\mathbb{F} c \circ \mathbb{F} x), \quad (2.79)$$

dove \circ denota la moltiplicazione elemento per elemento di Hadamard. Questo può essere calcolato efficientemente usando la FFT, calcolando prima le due trasformate di Fourier $\mathbb{F}c$ e $\mathbb{F}x$ e calcolando dopo la IDFT $\mathbb{F}^{-1}(\mathbb{F}c \circ \mathbb{F}x)$. Se la moltiplicazione matrice-vettore è effettuata ripetutamente con la stessa matrice circolante e vettori differenti, la DFT $\mathbb{F}c$ deve essere calcolata una sola volta. Una matrice $M \times M$ si dice matrice di Toeplitz T se ha valori costanti lungo ogni diagonale ed ha la seguente forma:

$$T = \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{(M-1)} \\ t_1 & t_0 & \cdots & t_{M-2} \\ \vdots & \vdots & \vdots & \vdots \\ t_{M-1} & t_{M-2} & \cdots & t_0 \end{pmatrix}. \quad (2.80)$$

Essa è completamente specificata dalla prima riga e dalla prima colonna. Una matrice di Toeplitz T $M \times M$ può essere incorporata in una matrice circolante C $N \times N$ con la prima colonna $c = (t_0, \dots, t_{M-1}, 0, \dots, 0, t_{(M-1)}, t_{-1})^T$. Qui $N = 2^l$ è la più piccola potenza di due tale che $N \geq 2M - 1$. Si noti che il vettore c ha $N - (2M - 1)$ zeri. La moltiplicazione matrice-vettore può essere quindi scritta come

$$(Tx)_k = (Cx^*)_k = (\mathbb{F}^{-1}(\mathbb{F}c \circ \mathbb{F}x^*))_k, \quad k = 0, 1, \dots, M - 1 \quad (2.81)$$

dove il vettore N -dimensionale x^* è un'estensione del vettore originale M -dimensionale x ottenuto aggiungendo $N - M$ zeri a x . Il problema è ora ridotto alla moltiplicazione matrice circolante-vettore, che può essere calcolato efficientemente usando la FFT come descritto in precedenza (si sceglie N come potenza di due al fine di usare la FFT di radice 2).

2.3.7 Test numerici

Vengono ora considerate un'opzione Put Down-Out con barriera lower $L = 80$ e un'opzione Call Up-Out con barriera upper $U = 120$. Per ognuna il monitoraggio è discreto con $T = 1$, $N = 252$. Il tasso di interesse privo di rischio e lo dividend-yield sono rispettivamente $r = 5\%$ e $q = 2\%$. Per quanto riguarda i modelli i parametri utilizzati sono per Black&Scholes $\sigma = 0.2$, per Merton $\sigma = 0.1$, $\lambda = 3$, $\mu = -0.05$, $\delta = 0.086$, per Kou $\sigma = 0.1$, $\lambda = 3$, $p = 0.3$, $\lambda^+ = 40$, $\lambda^- = 12$, per VG $\sigma = 0.1$, $\nu = 0.01$, $\theta = -0.01$, per NIG $\alpha = 15$, $\beta = -5$, $\delta = 0.5$.

Per generare il grafico che rappresenta la convergenza si è ricavato il prezzo dell'opzione usando diversi livelli di troncamento M e lo si è confrontato con il prezzo ottenuto utilizzando la griglia più fitta. Il grafico del prezzo riporta solamente i valori ottenuti con la più fitta delle discretizzazioni utilizzate. Si noti che nei grafici l'asse delle ascisse è scelto di volta in volta coerentemente con i risultati teorici presentati in 2.3.4. I grafici confermano l'andamento esponenziale dell'errore di troncamento come predetto dalla teoria.

In [Figura 2.2](#) viene riportato il prezzo per l'opzione Call Up-Out. In [Figura 2.3](#) il prezzo per l'opzione Put Down-Out.

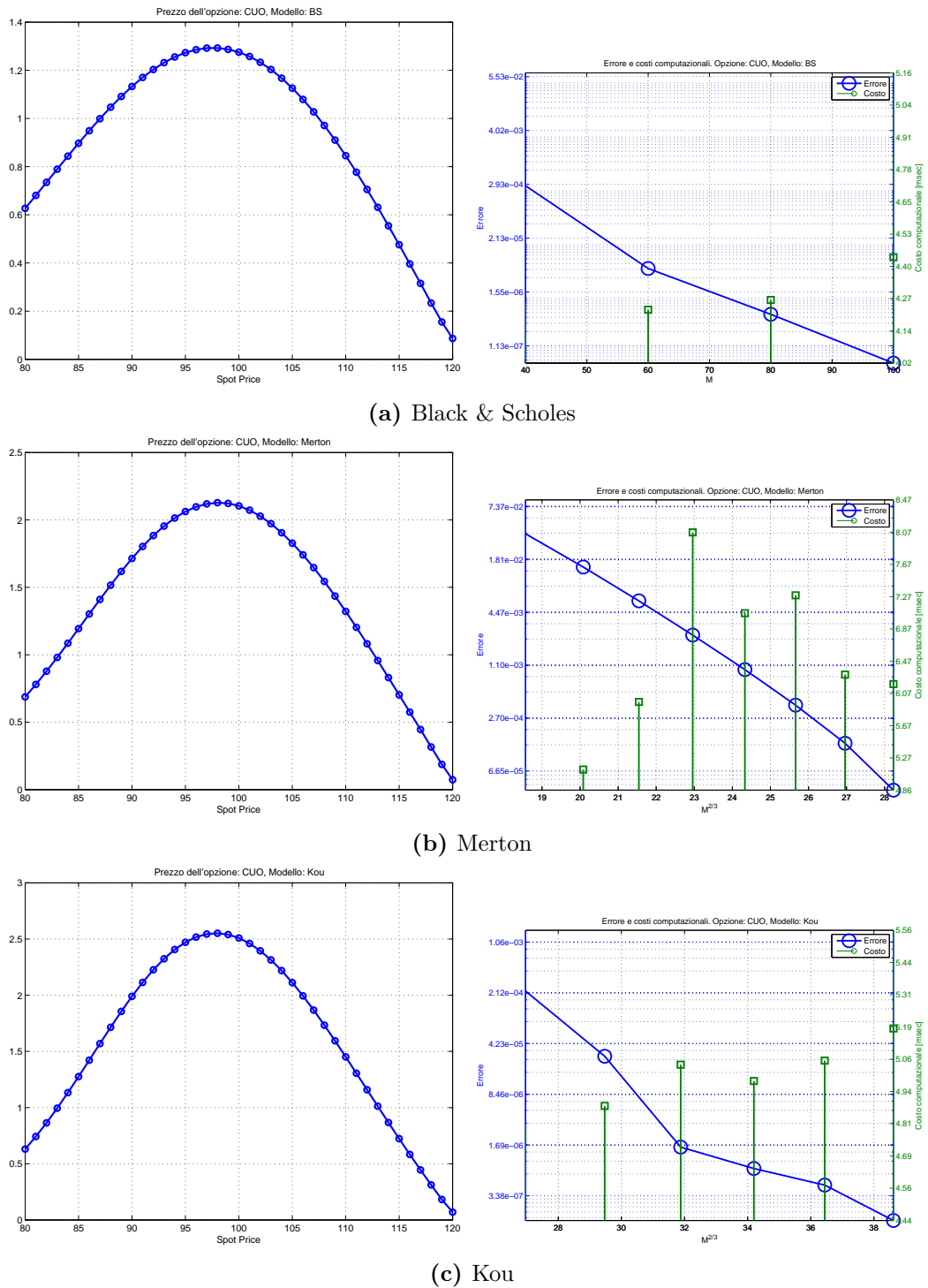
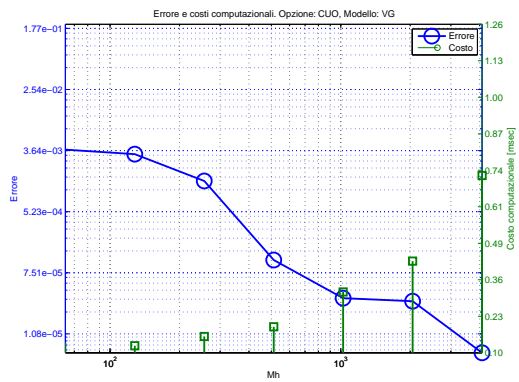
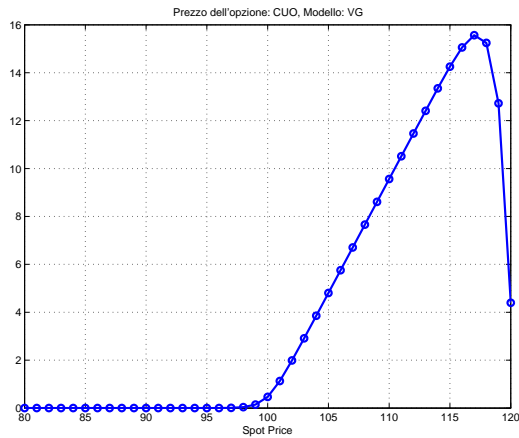
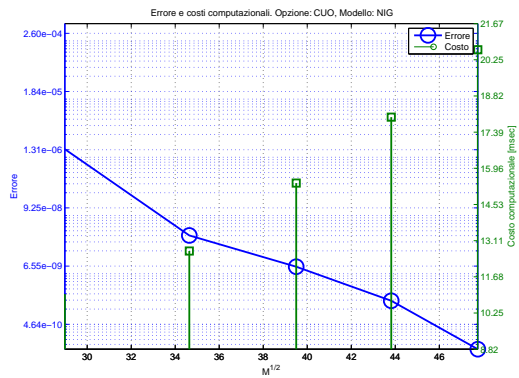
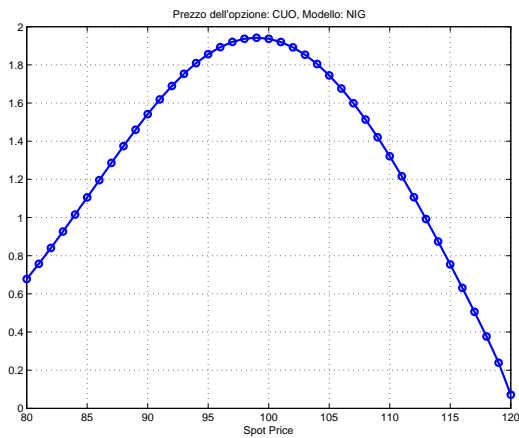


Figura 2.2: Pricing Up-Out Call, corrispondente errore e costi computazionali al variare del modello utilizzato.

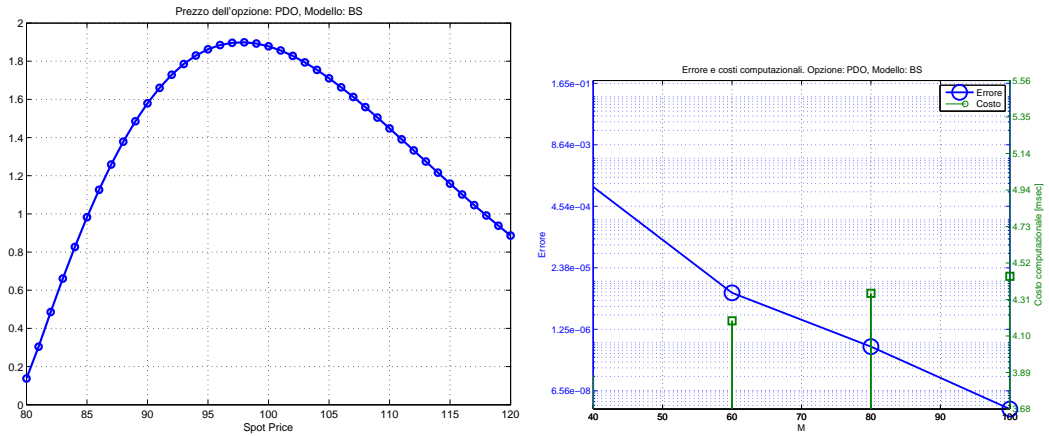


(d) VG

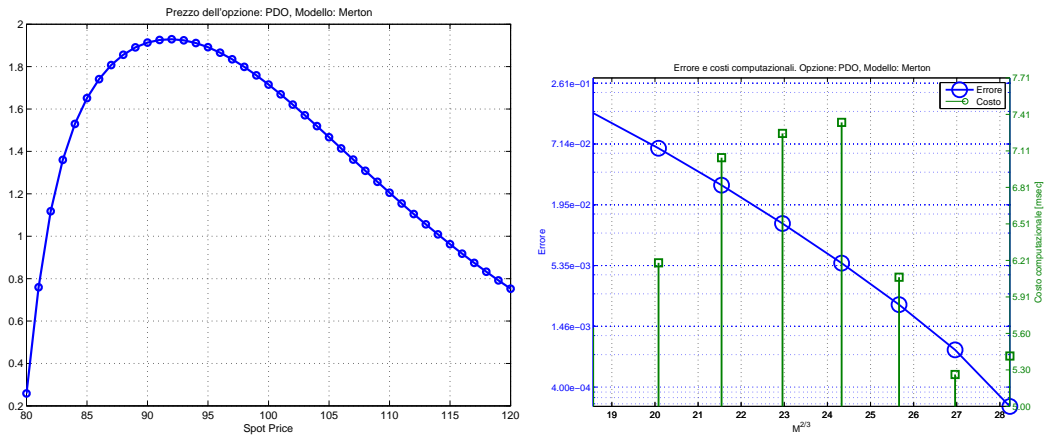


(e) NIG

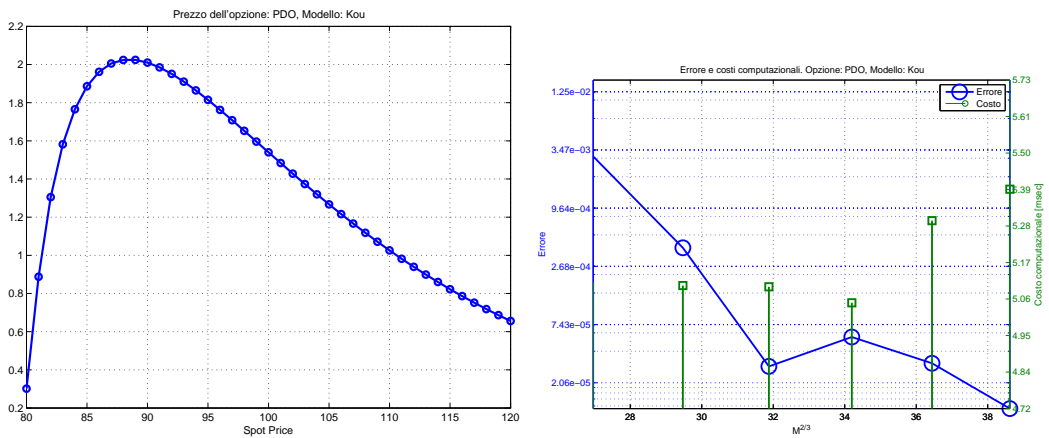
Figura 2.2: Pricing Up-Out Call, corrispondente errore e costi computazionali al variare del modello utilizzato.



(a) Black & Scholes

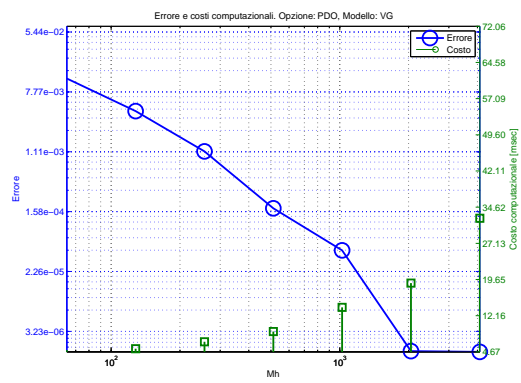
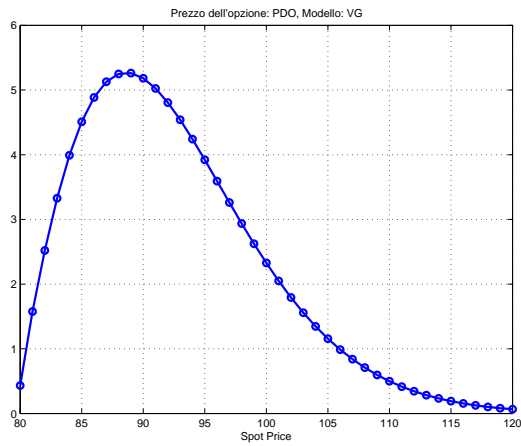


(b) Merton

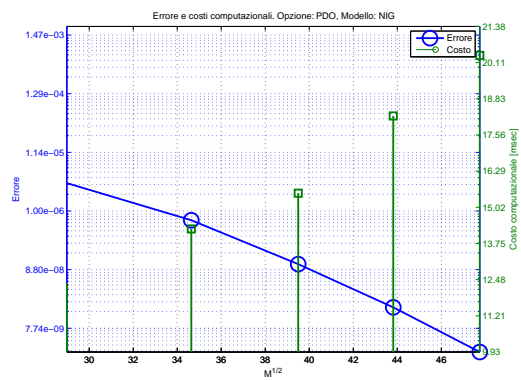
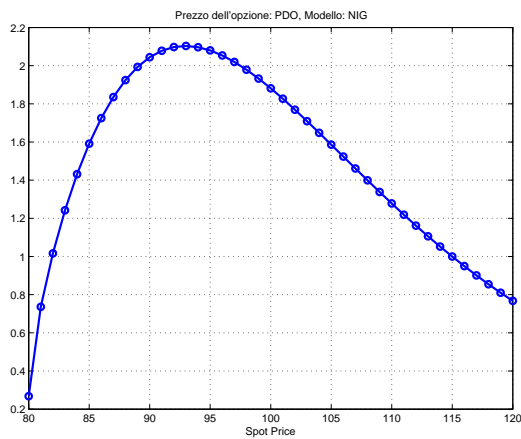


(c) Kou

Figura 2.3: Pricing Down-Out Put, corrispondente errore e costi computazionali al variare del modello utilizzato.



(d) VG



(e) NIG

Figura 2.3: Pricing Down-Out Put, corrispondente errore e costi computazionali al variare del modello utilizzato.

2.4 Opzioni Barriera con processi time-changed Lévy

In questa sezione, si estende quanto visto precedentemente al caso di dinamica sostenuta da processi time-changed Lévy, in particolare la trattazione sviluppata sarà poi applicata al modello di Heston. Si dedurrà che in questo caso l'algoritmo FHT è una generalizzazione dell'algoritmo precedente, in quanto è necessario gestire contemporaneamente un ulteriore processo, ovvero il CIR che sottostà all'andamento stocastico della volatilità. Pertanto il problema diventa bidimensionale invece che unidimensionale come nel caso precedente. L'algoritmo di pricing è sviluppato applicando il metodo FHT nella dimensione del log-asset return e adottando una formula di quadratura per l'integrazione numerica del tasso di attività logaritmico del cambio stocastico di scala temporale. Il focus della sezione sarà sul pricing di un'opzione barriera Put Down-Out, ma un approccio simile si può applicare anche per le opzioni barriera Call Up-Out e Knock-Out.

2.4.1 Induzione backward

Si consideri il processo log-asset return normalizzato al prezzo di esercizio K sotto l'ipotesi di processo time-changed Lévy X_{T_t} , (1.45).

Per un'opzione Down-Out Put, il payoff a scadenza è dato da $G(Y_T) = \tilde{G}(S_T) = K(1 - e^{Y_T})^+ \mathbb{I}_{(Y_T > L)}$. Date le N date di monitoraggio, $\mathcal{T} = \{t_k | k = 1, \dots, N\}$ con $t_N = T$, in corrispondenza di ognuna si controlla che il sottostante non sia disceso al di sotto della barriera L . La formula di pricing dell'opzione barriera si esprime quindi come

$$\tilde{V}(S_{t_k}, \gamma_{t_k}, t_k) = e^{-r\Delta} \mathbb{E}_{t_k} [\tilde{V}(S_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1})] \mathbb{I}_{(S_{t_k} > L)}, \quad (2.82)$$

dove S_{t_k} è il valore del sottostante al tempo t_k , γ_{t_k} è la varianza logaritmica al tempo t_k e \mathbb{E}_{t_k} denota il valore atteso condizionato a S_{t_k} e γ_{t_k} . Utilizzando la (1.45) e la tower property si ha

$$\begin{aligned} V(Y_{t_k}, \gamma_{t_k}, t_k) &= e^{-r\Delta} \mathbb{E} [\mathbb{E}[V(Y_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1}) | \mathcal{F}_{t_k}, \gamma_{t_{k+1}}] | \mathcal{F}_{t_k}] \\ &\approx e^{-r\Delta} \sum_{j=1}^J \omega_j p_\gamma(\zeta_j | \gamma_{t_k}) \mathbb{E}[V(Y_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1}) | \mathcal{F}_{t_k}, \gamma_{t_{k+1}} = \zeta_j], \end{aligned} \quad (2.83)$$

dove ζ_j , con $j = 1, 2, \dots, J$ sono i nodi della formula di quadratura per il tasso di attività logaritmico. Smorzando la funzione valore, si definisce $V_\alpha(Y_t, \gamma_t, t) = e^{\alpha Y_t} V(Y_t, \gamma_t, t)$. Il valore atteso interno si scrive in termini di trasformata di Fourier inversa come

$$\begin{aligned} &\mathbb{E}[V(Y_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1}) | \mathcal{F}_{t_k}, \gamma_{t_{k+1}} = \zeta_j] \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-\omega Y_{t_k}} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \Psi(-\omega; \zeta_j, \gamma_{t_k}) d\beta, \end{aligned} \quad (2.84)$$

dove $\omega = \alpha + i\beta$ e $\hat{V}_\alpha(\beta, \gamma_t, t)$ denota la trasformata di Fourier generalizzata di $V(Y_t, \gamma_t, t)$.

Combinando (2.83) e (2.84) e ripercorrendo passaggi simili a quelli presentati in [18], si ottiene la seguente formula di approssimazione

$$V_\alpha(Y_{t_k}, \gamma_{t_k}, t_k) \approx \mathbb{I}_{(l, \infty)} \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-i\beta Y_{t_k}} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\omega; \zeta_j, \gamma_{t_k}) d\beta, \quad (2.85)$$

dove $\tilde{\Psi}(\omega; \zeta_j, \gamma_{t_k}) = \Psi(\omega; \zeta_j, \gamma_{t_k}) p_\gamma(\zeta_j | \gamma_{t_k})$ e $l = \log \frac{L}{K}$.

Algoritmo trasformata di Fourier

Date N date di monitoraggio $\mathcal{T} = \{t_k | k = 1, 2, \dots, N\}$ con $t_N = T$, la procedura di induzione backward per il pricing di un'opzione Put Down-Out si riassume nel seguente schema

1. Inizializzazione della funzione valore a partire dal payoff a scadenza

$$V_\alpha(Y_{t_N}, \gamma_{t_N}, t_N) = G_\alpha(Y_{t_N}). \quad (2.86)$$

2. Il valore dell'opzione al tempo t_k si approssima come

$$V_\alpha(Y_{t_k}, \gamma_{t_k}, t_k) \approx \mathbb{I}_{(l, \infty)} \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-i\beta Y_{t_k}} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \gamma_{t_k}) d\beta. \quad (2.87)$$

La procedura sopra illustra l'implementazione backward usando il metodo basato sulla trasformata di Fourier, dove sono compiuti $N - 1$ passi della trasformata di Fourier inversa e $N - 1$ passi della trasformata di Fourier. Tuttavia si può riformulare l'induzione a ritroso sfruttando le eleganti proprietà del metodo FHT.

2.4.2 Il metodo della trasformata veloce di Hilbert

L'induzione a ritroso nel dominio di Fourier procede come segue. Si inizia con $k = N$, dove

$$\hat{V}_\alpha(\beta, \zeta_j, t_N) = \hat{G}_\alpha(\beta), \quad j = 1, 2, \dots, J. \quad (2.88)$$

L'approssimazione numerica di $\hat{V}_\alpha(\beta, \zeta_p, t_k)$ è data in termini della trasformata di Hilbert

$$\begin{aligned} \hat{V}_\alpha(\beta, \zeta_p, t_k) \approx & \frac{e^{-r\Delta}}{2} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \zeta_p) \\ & + \frac{i e^{-r\Delta}}{2} e^{i\beta l} \mathcal{H} \left(e^{-i\eta l} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\eta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\eta; \zeta_j, \gamma_{t_k}) \right) (\beta), \quad (2.89) \end{aligned}$$

per $p = 1, 2, \dots, J$. L'approssimazione numerica del valore dell'opzione barriera al tempo iniziale è data da

$$V(Y_{t_0}, \zeta_p, t_0) \approx \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-(\alpha+i\beta)Y_{t_0}} \hat{V}_\alpha(\beta, \zeta_j, t_1) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \zeta_p) d\beta. \quad (2.90)$$

Si noti che la trasformata di Fourier di $\int_{-\infty}^{\infty} e^{-i\beta Y_{t_k}} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \gamma_{t_k}) d\beta$ è il prodotto di $\hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \Psi(-\alpha - i\beta; \zeta_j, \gamma_{t_k})$.

Applicando la trasformata di Fourier a entrambi i lati di (2.87) e usando la (2.38) si ottiene

$$\begin{aligned} \hat{V}_\alpha(\beta, \gamma_{t_k}, t_k) &\approx \frac{e^{-r\Delta}}{2} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \gamma_{t_k}) \\ &+ \frac{ie^{-r\Delta}}{2} e^{i\beta l} \mathcal{H} \left(e^{-i\eta l} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\eta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\eta; \zeta_j, \gamma_{t_k}) \right) (\beta). \end{aligned} \quad (2.91)$$

Ponendo $\gamma_{t_k} = \zeta_p$, $p = 1, 2, \dots, J$, si ottiene la (2.88). Infine, siccome la trasformata inversa di Fourier di $e^{-r\Delta} \mathbb{E}[V(Y_{t_1}, \zeta_j, t_1) | \mathcal{F}_{t_0}]$ fornisce il valore al tempo t_0 , (2.87), si ottiene il valore dell'opzione barriera all'istante iniziale come mostrato in (2.90).

Approssimazione discreta

Nell'algoritmo FHT per il pricing dell'opzione barriera si deve valutare la trasformata di Hilbert della seguente forma

$$Rg(\beta) = \mathcal{H} \left(e^{-i\eta x} g(\eta, \zeta_j) \tilde{\Psi}(-\alpha - i\eta; \zeta_j, \zeta_p) \right) (\beta), \quad (2.92)$$

che può essere valutata tramite l'approssimazione Sinc (2.47) come segue

$$R_{h,M}g(\beta) = \sum_{m=-M}^M e^{-imhx} g(mh, \zeta_j) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) \frac{1 - \cos \frac{\pi(\beta - mh)}{h}}{\frac{\pi(\beta - mh)}{h}}. \quad (2.93)$$

Qui $\beta = -Mh, \dots, Mh$, h è lo step ed M è il livello di troncamento. L'approssimazione tramite somma trapezoidale è altamente accurata e presenta un errore di discretizzazione che decade esponenzialmente. Anche in questo caso la matrice di Toeplitz è data dalla (2.73).

Basandosi su (2.89) e (2.93) e sostituendo gli elementi della matrice di Toeplitz definiti in (2.73), si ottiene

$$\hat{V}_\alpha(nh, \zeta_p, t_k) = \frac{e^{-r\Delta}}{2} \sum_{j=1}^J \omega_j \hat{V}_\alpha(nh, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i nh; \zeta_j, \zeta_p) \quad (2.94)$$

$$+ \frac{i e^{-r\Delta} e^{i n h l}}{2} \sum_{\substack{m=-M \\ m \neq n}}^M e^{-i m h l} \sum_{j=1}^J \omega_j \hat{V}_\alpha(mh, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i mh; \zeta_j, \zeta_p) \frac{1 - (-1)^{n-m}}{\pi(n-m)}, \quad (2.95)$$

dove $n = -M, \dots, M$. Ora si approssima il valore iniziale dell'opzione usando la formula del trapezio come segue

$$V(Y_{t_0}, \zeta_p, t_0) \approx \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \sum_{m=-M}^M e^{-(\alpha+i mh)Y_{t_0}} \hat{V}_\alpha(mh, \zeta_j, t_1) \tilde{\Psi}(-\alpha - i mh; \zeta_j, \zeta_p) h. \quad (2.96)$$

Il calcolo dell'induzione backward nel dominio di Fourier può essere riformulato con la notazione matrice-vettore di più facile lettura. Si costruisce la seguente moltiplicazione matrice-vettore

$$\begin{aligned} \mathbf{k}_p(t_k) &= [\hat{V}_\alpha(t_{k+1}) \cdot \tilde{\Psi}(\zeta_p)] \mathbf{w}, \quad p = 1, 2, \dots, J; \\ B(t_k) &= [\mathbf{k}_1(t_k), \mathbf{k}_2(t_k), \dots, \mathbf{k}_J(t_k)], \end{aligned}$$

dove $\mathbf{k}_p(t_k)$ è un vettore colonna di dimensione $2M + 1$, $B(t_k)$ è una matrice $(2M + 1) \times J$ e l'operatore \cdot denota il prodotto elemento per elemento. Siano H, F e \hat{G} matrici $(2M + 1) \times J$, i cui elementi sono rispettivamente $H_{m,p} = e^{-i m h l}$, $F_{m,p} = e^{i m h l}$. In accordo con la (2.94) si può riscrivere l'induzione backward nel dominio di Fourier nella seguente forma matrice-vettore

$$\hat{V}_\alpha(t_k) = \frac{e^{-r\Delta}}{2} B(t_k) + \frac{e^{-r\Delta} i}{2} [T(B(t_k) \cdot H(t_k))] \cdot F(t_k). \quad (2.97)$$

Grazie alla struttura della matrice di Toeplitz l'algoritmo FFT può essere applicato raggiungendo una complessità di $O(M \log_2 M)$ nella dimensione del tasso di attività logaritmico.

2.4.3 Sintesi dell'algoritmo numerico di pricing

Step1: Preparazione

Calcolare $\hat{V}_\alpha(t_N)$ usando la formula analitica (2.88) per $k = N$;
 Preparare la matrice $\tilde{\Psi}(\zeta_p)$ per $p = 1, 2, \dots, J$;
 Calcolare la prima riga e colonna della matrice di Toeplitz.
 Costruire le matrici H, F e \hat{G} ;

Step2: Induzione backward

Calcolare $\mathbf{k}_p(t_k) = [\hat{V}_\alpha(t_{k+1}) \cdot \tilde{\Psi}(\zeta_p)]\mathbf{w}$, per $p = 1, 2, \dots, J$ e aggiornare $B(t_k)$;
 Calcolare $\tilde{V}_\alpha(t_k)$ tramite (4.28) usando l'algoritmo FFT per $k = N - 1, \dots, 1$.

Step3: Valore iniziale dell'opzione

Calcolare il valore iniziale dell'opzione tramite (2.96). Utilizzare infine un'interpolazione spline per ottenere $V(Y_{t_0}, \gamma_{t_0}, t_0)$.

2.4.4 Test numerici

In questa sezione si utilizza l'algoritmo precedentemente introdotto per ricavare il prezzo di un'opzione barriera Put Down-Out sotto l'ipotesi di processo di Heston a volatilità stocastica, scelto come rappresentante della classe di modelli time-changed Lévy. L'opzione scelta ha strike price $K = 100$, maturity $T = 1$, date di monitoraggio $N = 12$ e il tasso risk free è $r = 0.05$, mentre il sottostante è scelto con valore iniziale S_0 compreso tra 80 e 120, con passo di 5, mentre la barriera down è sempre pari a $L = 80$. I parametri utilizzati per la dinamica Heston del sottostante sono $v_0 = 0.08, 0.15, 0.22$, $\bar{v} = 0.16$, $\lambda = 5$, $\eta = 0.9$ e $\rho = 0.1$, si faccia riferimento alla Sezione 1.4.2 per il loro significato finanziario, mentre il livello di troncamento per la discretizzazione della trasformata di Hilbert è $M = 2^6$.

In Figura 2.4 viene riportato l'andamento dei prezzi ottenuto per diversi livelli della volatilità iniziale, al variare di S_0 . Come ci si aspetta il valore dell'opzione diminuisce sia al crescere di S_0 che all'avvicinarsi di S_0 alla barriera L .

Il prezzo trovato con i precedenti parametri e $v_0 = 0.0625$, $M = 2^5$ è di 1.1622 mediante l'algoritmo FHT e di 1.1580 con un intervallo di confidenza al 95% di ampiezza 0.0432 mediante il metodo Monte Carlo, che può essere considerato il benchmark, con 10^5 simulazioni.

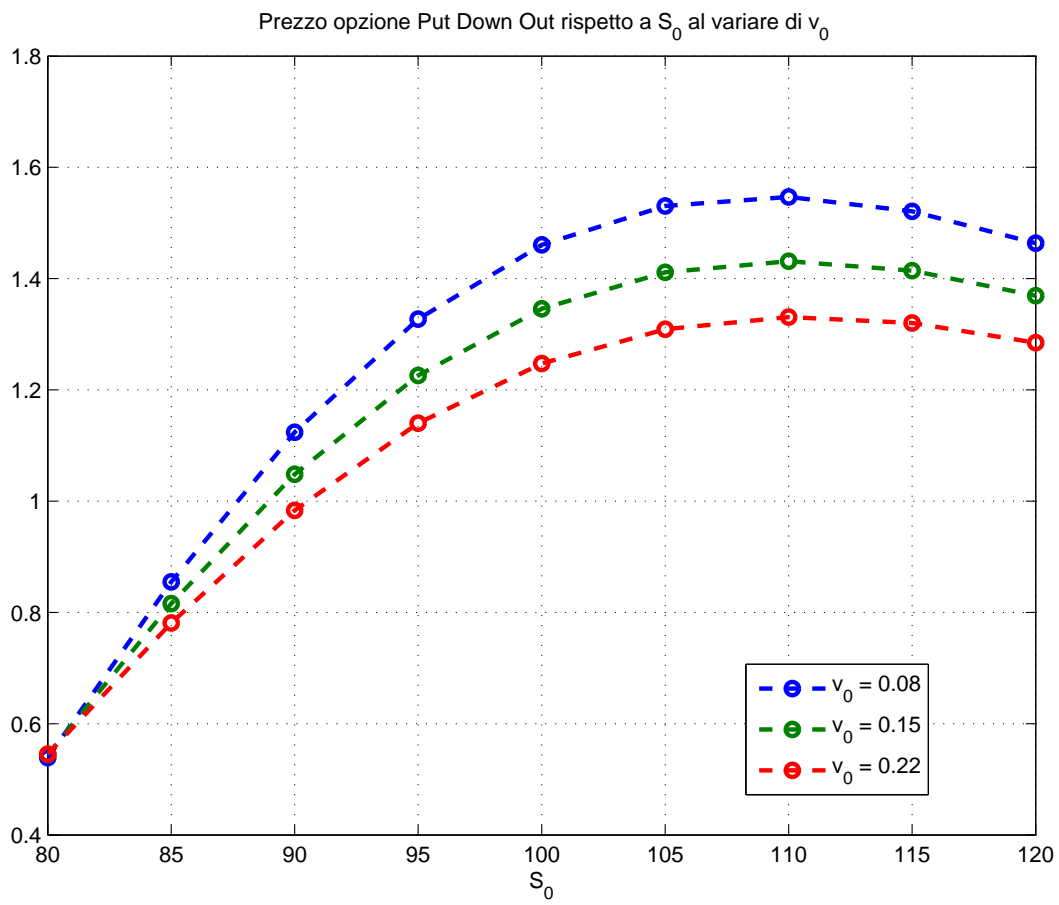


Figura 2.4: Prezzo Put Down-Out per diversi livelli di volatilità iniziale

Capitolo 3

Opzioni Lookback

In questo capitolo ci si occupa di un'altra classe di opzioni path dependent, in particolare della valutazione di opzioni lookback, per tutta la trattazione svolta e per le dimostrazioni dei teoremi presentati si è fatto riferimento a [8]. Il payoff di questo tipo di opzioni dipende dal massimo o dal minimo raggiunto dal sottostante durante tutta la vita del derivato, monitorati su un insieme discreto di date. Il problema che viene affrontato è quello della determinazione dei momenti esponenziali della densità del massimo, i quali possono poi essere utilizzati per un pricing efficiente. Questo problema viene trattato soltanto nel caso dei processi di Lévy, in quanto si complica notevolmente qualora si consideri l'effetto del time-change.

3.1 Introduzione teorica

Sia X un processo di Lévy e $\mathcal{T} = \{0, \Delta, \dots, N\Delta = T\}$ l'insieme dei tempi di monitoraggio. Si denoti $X_{(j)} = X_{j\Delta}$ per $0 \leq j \leq N$. Sia \mathfrak{M}_N il massimo osservato del processo di Lévy:

$$\mathfrak{M}_N = \max_{0 \leq j \leq N} X_{(j)}. \quad (3.1)$$

Si è interessati a valutare il momento dell'esponenziale di \mathfrak{M}_N :

$$\mathcal{L}(\mathfrak{M}_N)(s) = \mathbb{E}[e^{s\mathfrak{M}_N}]. \quad (3.2)$$

Per ogni $s \in (-\lambda_+, -\lambda_-)$, si definisce

$$Z_t = e^{sX_t + t\psi(-is)}. \quad (3.3)$$

Allora Z_t è una martingala positiva di media 1. In particolare, essa definisce una misura di probabilità equivalente \mathbb{P}^* tramite il teorema di Radon-Nikodym. Sotto questa nuova misura \mathbb{P}^* , X è un processo di Lévy con funzione caratteristica

$$\phi_t^* = \frac{\phi_t(\xi - is)}{\phi(-is)}, \quad \xi \in \mathbb{R}. \quad (3.4)$$

Segue che

$$\begin{aligned}\mathcal{L}(\mathfrak{M}_N)(s) &= \mathbb{E}[e^{s(\mathfrak{M}_N - X_{(N)}) + sX_{(N)}}] \\ &= e^{-T\psi(-is)} \mathbb{E}[Z_T e^{s(\mathfrak{M}_N - X_{(N)})}] \\ &= \phi_T(-is) \mathbb{E}^*[e^{s(\mathfrak{M}_N - X_{(N)})}].\end{aligned}$$

dove \mathbb{E}^* denota l'operatore valore atteso sotto la misura \mathbb{P}^* .

3.1.1 Ricorsione forward nello spazio di stato

Si consideri il massimo discreto $\mathfrak{M}_j = \max_{0 \leq k \leq j} X_{(k)}$ osservato entro il tempo $t_j = j\Delta$, $0 \leq j \leq N$. Si suppone che $\phi_\Delta^* \in L^1(\mathbb{R})$ in modo che la probabilità di transizione $p_\Delta^*(x)$ del processo di Lévy X sotto \mathbb{P}^* sia continua e limitata. Per ogni j , la variabile aleatoria non negativa $\mathfrak{M}_j - X_{(j)}$ ha una densità $f_j(x)$ continua su $(0, \infty)$ e una massa di probabilità in zero $c_0 = \mathbb{P}^*(\mathfrak{M}_0 - X_{(0)} = 0) = 1$. Allora la distribuzione di $\mathfrak{M}_N - X_{(N)}$ si trova ricorsivamente:

$$\begin{aligned}\mathfrak{M}_j - X_{(j)} &= \max(X_{(j)}, \mathfrak{M}_{j-1}) - X_{(j)} \\ &= \max(0, \mathfrak{M}_{j-1} - X_{(j)}) \\ &= \max(0, \mathfrak{M}_{j-1} - X_{(j-1)} - (X_{(j)} - X_{(j-1)})).\end{aligned}\tag{3.5}$$

Si inizia con $f_0 = 0$, $c_0 = 1$. Conoscendo f_{j-1} e c_{j-1} , si ottiene la densità di $\mathfrak{M}_{j-1} - X_{(j)}$ per convoluzione:

$$c_{j-1}p_\Delta^*(-x) + \int_0^\infty f_{j-1}(y)p_\Delta^*(y-x)dy = c_{j-1}p_\Delta^*(-x) + \int_{\mathbb{R}} f_{j-1}(y)p_\Delta^*(y-x)dy.\tag{3.6}$$

L'uguaglianza vale poichè $f_{j-1}(x) = 0$ per $x \leq 0$. La densità e la massa di probabilità associata diventano quindi:

$$f_j(x) = \mathbb{I}_{(0, \infty)}(x) \left(c_{j-1}p_\Delta^*(-x) + \int_{\mathbb{R}} f_{j-1}(y)p_\Delta^*(y-x)dy \right), \quad c_j = 1 - \int_{\mathbb{R}} f_j(x)dx\tag{3.7}$$

per $j = 1, 2, \dots, N$. Avendo calcolato la distribuzione di $\mathfrak{M}_N - X_{(N)}$, il momento esponenziale diventa:

$$\mathbb{E}[e^{s\mathfrak{M}_N}] = \phi_T(-is) \mathbb{E}^*[e^{s(\mathfrak{M}_N - X_{(N)})}] = \phi_T(-is) \left(c_N + \int_{\mathbb{R}} e^{sx} f_N(x)dx \right).\tag{3.8}$$

Tuttavia la computazione dell'algoritmo nello spazio degli stati è molto costosa. Fortunatamente, si può considerare il problema nello spazio di Fourier. L'algoritmo di ricorsione forward che sarà presentato richiede solamente la conoscenza della funzione caratteristica del processo di Lévy e comporta il calcolo di una sola trasformata di Hilbert ad ogni passo. Prima di riscrivere l'algoritmo nello spazio di Fourier, lo si riformula in una maniera più conveniente per la computazione del momento

esponenziale. Per ogni $s \in (-\lambda_+, -\lambda_-)$ e $0 \leq j \leq N$, si definisce $g_j(x) = e^{sx} f_j(x)$. Allora

$$g_j(x) = \mathbb{I}_{(0,\infty)}(x) \left(c_{j-1} e^{sx} p_\Delta^*(-x) + \int_{\mathbb{R}} e^{sx-sy} g_{j-1}(y) p_\Delta^*(y-x) dy \right). \quad (3.9)$$

Si noti che $g_0 = 0 \in L^1(\mathbb{R})$ e $e^{sx} p_\Delta^*(-x) \in L^1(\mathbb{R})$ siccome

$$\int_{\mathbb{R}} e^{sx} p_\Delta^*(-x) dx = \int_{\mathbb{R}} e^{-sx} p_\Delta^*(x) dx = \phi_\Delta^*(is) < \infty. \quad (3.10)$$

Siccome la convoluzione di due funzioni in $L^1(\mathbb{R})$ è ancora in $L^1(\mathbb{R})$ per induzione si ha che $g_j(x) \in L^1(\mathbb{R})$ per ogni $0 \leq j \leq N$. Allora si ottiene la seguente espressione:

$$\mathbb{E}^*[e^{s(\mathfrak{M}_N - X_{(N)})}] = c_N + \int_{\mathbb{R}} e^{sx} f_N(x) dx = c_N + \int_{\mathbb{R}} g_N(x) dx. \quad (3.11)$$

Di conseguenza, il momento esponenziale $\mathcal{L}(\mathfrak{M}_N)(s)$ può essere calcolato usando la ricorsione che coinvolge f e g . Riassumendo per $0 \leq j \leq N$:

$$c_0 = 1, \quad f_0 = 0, \quad g_0 = 0, \quad (3.12a)$$

$$f_j(x) = \mathbb{I}_{(0,\infty)}(x) \left(c_{j-1} p_\Delta^*(-x) + \int_{\mathbb{R}} f_{j-1}(y) p_\Delta^*(y-x) dy \right), \quad (3.12b)$$

$$g_j(x) = \mathbb{I}_{(0,\infty)}(x) \left(c_{j-1} e^{sx} p_\Delta^*(-x) + \int_{\mathbb{R}} e^{sx-sy} g_{j-1}(y) p_\Delta^*(y-x) dy \right), \quad (3.12c)$$

$$c_j = 1 - \int_{\mathbb{R}} f_j(x) dx, \quad (3.12d)$$

$$\mathcal{L}(\mathfrak{M}_N)(s) = \phi_T(-is) \left(c_N + \int_{\mathbb{R}} g_N(x) dx \right). \quad (3.12e)$$

3.1.2 Ricorsione forward nello spazio di Fourier

Si consideri ora l'algoritmo (3.12) nello spazio di Fourier. Indicando con \hat{f}_j e \hat{g}_j le trasformate di Fourier rispettivamente di f_j e g_j , si ottiene per $0 \leq j \leq N$:

$$c_0 = 1, \quad \hat{f}_0 = 0, \quad \hat{g}_0 = 0, \quad (3.13a)$$

$$\hat{f}_j(\xi) = \frac{1}{2} \phi_\Delta^*(-\xi) (c_{j-1} + \hat{f}_{j-1}(\xi)) + \frac{i}{2} \mathcal{H} \left(\phi_\Delta^*(-\eta) (c_{j-1} + \hat{f}_{j-1}(\eta)) \right) (\xi), \quad (3.13b)$$

$$\hat{g}_j(x) = \frac{1}{2} \phi_\Delta^*(-\xi + is) (c_{j-1} + \hat{g}_{j-1}(\xi)) + \frac{i}{2} \mathcal{H} \left(\phi_\Delta^*(-\eta + is) (c_{j-1} + \hat{g}_{j-1}(\eta)) \right) (\xi), \quad (3.13c)$$

$$c_j = 1 - \hat{f}_j(0), \quad (3.13d)$$

$$\mathcal{L}(\mathfrak{M}_N)(s) = \phi_T(-is) (c_N + \hat{g}_N(0)). \quad (3.13e)$$

3.1.3 Trasformata di Hilbert

Si discretizza ora l'algoritmo (3.13a), fissato il passo $h > 0$ e il livello di troncamento M , $-M \leq k \leq M$, si ha:

$$c_{0,h,M} = 1, \quad \hat{f}_{0,h,M} = 0, \quad \hat{g}_{0,h,M} = 0, \quad (3.14a)$$

$$\begin{aligned} \hat{f}_{j,h,M}(kh) &= \frac{1}{2} \phi_{\Delta}^*(-kh)(c_{j-1} + \hat{f}_{j-1,h,M}(kh)) + \dots \\ &+ \frac{i}{2} \sum_{m=-M, m \neq k}^M \phi_{\Delta}^*(-mh)(c_{j-1} + \hat{f}_{j-1,h,M}(mh)) \frac{1 - (-1)^{k-m}}{\pi(k-m)}, \end{aligned} \quad (3.14b)$$

$$\begin{aligned} \hat{g}_{j,h,M}(kh) &= \frac{1}{2} \phi_{\Delta}^*(-kh + i s)(c_{j-1} + \hat{g}_{j-1,h,M}(kh)) + \dots \\ &+ \frac{i}{2} \sum_{m=-M, m \neq k}^M \phi_{\Delta}^*(-mh + i s)(c_{j-1} + \hat{g}_{j-1,h,M}(mh)) \frac{1 - (-1)^{k-m}}{\pi(k-m)}, \end{aligned} \quad (3.14c)$$

$$c_{j,h,M} = 1 - \hat{f}_{j,h,M}(0), \quad (3.14d)$$

$$\mathcal{L}_{h,M}(\mathfrak{M}_N)(s) = \phi_T(-i s)(c_N + \hat{g}_{N,h,M}(0)). \quad (3.14e)$$

L'algoritmo (3.14) può essere efficientemente implementato usando l'algoritmo FFT. La sommatoria in (3.14b) e in (3.14c) corrisponde a una moltiplicazione matrice-vettore di Toeplitz, che ha un costo di $O(M \log(M))$ invece di $O(M^2)$. Lo sforzo complessivo risulta $O(NM \log(M))$.

3.2 Floating strike lookback put

Un'opzione lookback floating strike put ha il seguente payoff:

$$\max_{t \in T} S_t - S_T. \quad (3.15)$$

Essa consente al possessore di vendere l'asset sottostante al prezzo massimo. Sia $\mathcal{T} = \{0, \Delta, 2\Delta, \dots, N\Delta = T\}$ l'insieme dei tempi di osservazione, con Δ intervallo temporale tra due tempi di monitoraggio consecutivi. Il prezzo dell'opzione lookback put al tempo zero è data da:

$$\begin{aligned} V &= e^{-rT} \mathbb{E} \left[\max_{0 \leq j \leq N} S_{j\Delta} - S_T \right] = S_0 e^{-rT} \mathbb{E}[e^{\mathfrak{M}_N}] \\ &= e^{-qT} S_0 = S_0 e^{-rT} \mathcal{L}(\mathfrak{M}_N)(1) - e^{-qT} S_0, \end{aligned}$$

dove il momento $\mathcal{L}(\mathfrak{M}_N)$ con $s = 1$ può essere calcolato come descritto nella [Sezione 3.1](#). Si supponga che il tempo di pricing corrente sia zero, il massimo prezzo del sottostante osservato prima del tempo corrente è S_{max} , il massimo seguente osservato sarà per $t > 0$ e il tempo rimanente alla scadenza è $T = t + N\Delta$. Il payoff a scadenza

T è dato dal $\max(S_{max}, \max_{0 \leq j \leq N} S_{t+j\Delta}) - S_T$. Sia $X_{max} = \ln(S_{max}/S_0)$. Per $0 \leq j \leq N$, si definisce $X_{(j)} := X_{t+j\Delta}$ e

$$\mathfrak{M}_j := \max(X_{max}, \max_{0 \leq k \leq j} X_{(k)}). \quad (3.16)$$

In questo caso

$$\mathfrak{M}_0 - X_{(0)} := \max(0, X_{max} - X_t). \quad (3.17)$$

Questo è in contrasto con $\mathfrak{M}_0 - X_0 = 0$ nel caso considerato nella [Sezione 3.1](#). Comunque la relazione (3.5) vale ancora. Di conseguenza, per calcolare $\mathcal{L}(\mathfrak{M}_N(s))$, bisogna modificare il primo step della ricorsione forward in (3.12). Per $0 \leq j \leq N$, si definiscono c_j , $f_j(x)$ e $g_j(x) = e^{sx} f_j(x)$ come prima in modo tale che la densità di $\mathfrak{M}_j - X_{(j)}$ su $(0, \infty)$ sotto \mathbb{P}^* sia data da $f_j(x)$ e $c_j = \mathbb{P}^*(\mathfrak{M}_j - X_{(j)} = 0)$. Allora

$$f_0(x) = \mathcal{I}_{(0, \infty)}(x) \cdot p_t^*(X_{max} - x), \quad (3.18)$$

dove p_t^* è la probabilità di transizione di X sotto \mathbb{P}^* . Si noti che la trasformata di Fourier di $p_t^*(X_{max} - x)$ è data da $e^{i\xi X_{max}} \phi_t^*(-\xi)$, e che $e^{sx} p_t^*(X_{max} - x)$ è data da $e^{X_{max}(i\xi+s)} \phi_t^*(-\xi + is)$, dove ϕ_t^* è definito in (3.4). Si arriva quindi al seguente algoritmo per calcolare $\mathcal{L}(\mathfrak{M}_N)(s)$ con il massimo predeterminato X_{max} e $N + 1$ tempi di osservazione rimanenti $t + j\Delta$, $t > 0$, $j = 0, 1, \dots, N$.

Si inizia con

$$\hat{f}_0(\xi) = \frac{1}{2} e^{i\xi X_{max}} \phi_t^*(-\xi) + \frac{i}{2} \mathcal{H}(e^{i\eta X_{max}} \phi_t^*(-\eta))(\xi), \quad \xi \in \mathbb{R}, \quad (3.19)$$

$$\hat{g}_0(\xi) = \frac{1}{2} e^{X_{max}(i\xi+s)} \phi_t^*(-\xi + is) + \frac{i}{2} \mathcal{H}(e^{X_{max}(i\eta+s)} \phi_t^*(-\eta + is))(\xi), \quad \xi \in \mathbb{R}, \quad (3.20)$$

$$c_0 = 1 - \hat{f}_0(0). \quad (3.21)$$

Per $j = 1, 2, \dots, N$, si calcola \hat{f}_j , c_j , \hat{g}_j e $\mathcal{L}(\mathfrak{M}_N)(s)$ in accordo con (3.14). Quando il tempo di pricing è anche un tempo massimo $\mathfrak{M}_0 = \max(X_{max}, 0)$ è costante, non necessariamente zero. Perciò, la distribuzione di $\mathfrak{M}_0 - X_0$ ha una massa di probabilità in nel punto \mathfrak{M}_0 . Allora per $j = 1$, (3.5) diventa

$$\mathfrak{M}_1 - X_{(1)} = \max(0, \mathfrak{M}_0 - X_{(1)}). \quad (3.22)$$

Conseguentemente, si ha

$$f_1(x) = \mathcal{I}_{(0, \infty)}(x) \cdot p_\Delta^*(\mathfrak{M}_0 - x). \quad (3.23)$$

Si ha quindi il seguente algoritmo per calcolare il momento esponenziale di $\mathcal{L}(\mathfrak{M}_N)(s)$:
Si inizia con

$$\hat{f}_1(\xi) = \frac{1}{2} e^{i\xi \mathfrak{M}_0} \phi_\Delta^*(-\xi) + \frac{i}{2} \mathcal{H}(e^{i\eta \mathfrak{M}_0} \phi_\Delta^*(-\eta))(\xi), \quad \xi \in \mathbb{R}, \quad (3.24)$$

$$\hat{g}_1(\xi) = \frac{1}{2} e^{\mathfrak{M}_0(i\xi+s)} \phi_\Delta^*(-\xi + is) + \frac{i}{2} \mathcal{H}(e^{\mathfrak{M}_0(i\eta+s)} \phi_\Delta^*(-\eta + is))(\xi), \quad \xi \in \mathbb{R}, \quad (3.25)$$

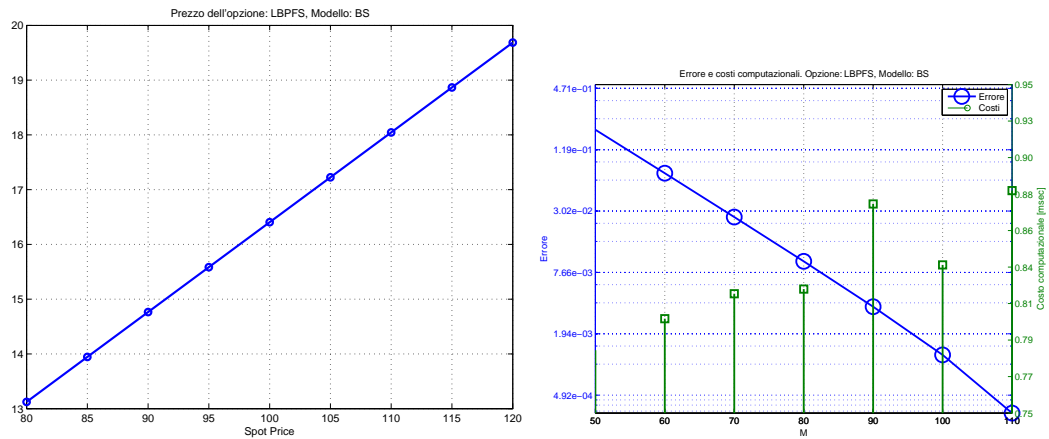
$$c_1 = 1 - \hat{f}_1(0), \quad (3.26)$$

dove $\mathfrak{M}_0 = \max(X_{max}, 0)$. Per $j = 2, 3, \dots, N$, si calcola $\hat{f}_j, c_j, \hat{g}_j$ e $\mathcal{L}(\mathfrak{M}_N)(s)$ in accordo con (3.14).

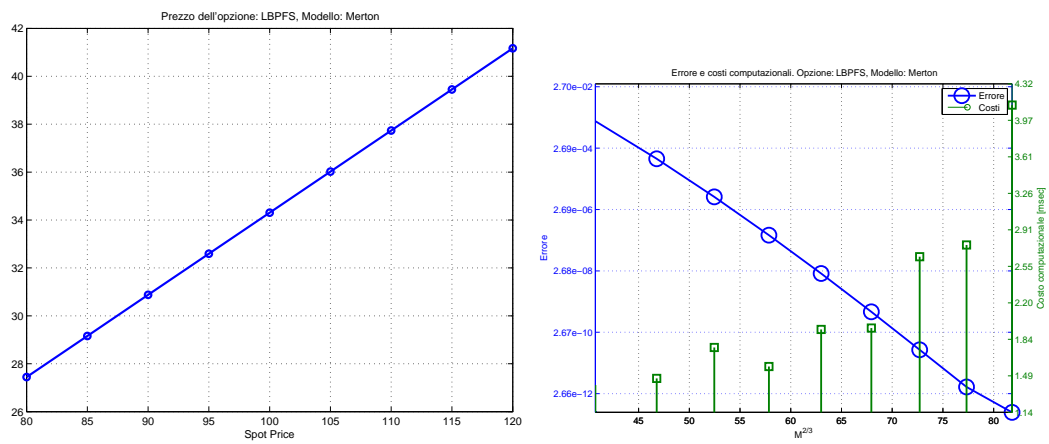
3.2.1 Test numerici

In [Figura 3.1](#) viene riportato il prezzo per un'opzione Put lookback floating strike. L'opzione ha i seguenti parametri: $S_0 = 80 : 120$, con passo di 5, $T = 1$, $r = 0.05$, $q = 0.02$, $N = 252$. Per quanto riguarda i modelli i parametri utilizzati sono per Black&Scholes $\sigma = 0.2$, per Merton $\sigma = 0.3$, $\lambda = 3$, $\mu = -0.05$, $\delta = 0.086$, per Kou $\sigma = 0.1$, $\lambda = 4$, $p = 0.4$, $\lambda^+ = 20$, $\lambda^- = 15$, per VG $\sigma = 0.1$, $\nu = 0.01$, $\theta = -0.01$, per NIG $\alpha = 15$, $\beta = -5$, $\delta = 0.5$.

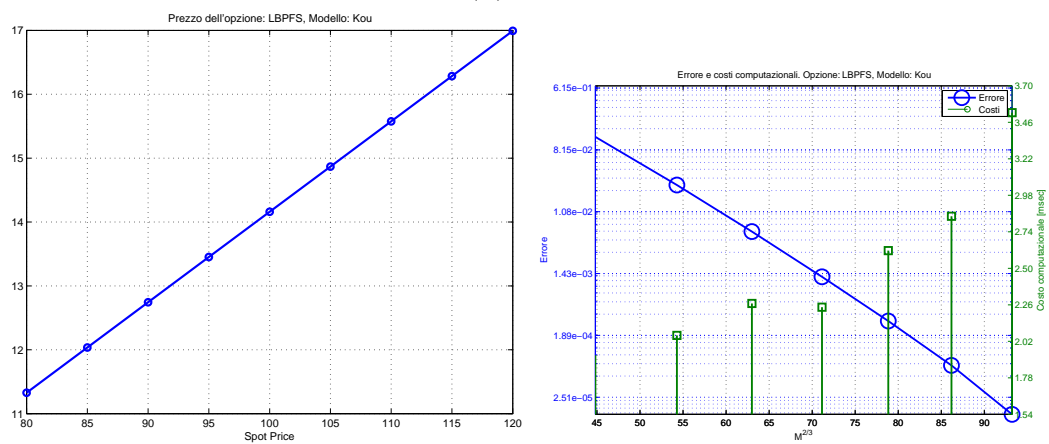
Per generare il grafico che rappresenta la convergenza si è ricavato il prezzo dell'opzione usando diversi livelli di troncamento M e lo si è confrontato con il prezzo ottenuto utilizzando la griglia più fitta. Il grafico del prezzo riporta solamente i valori ottenuti con la più fitta delle discretizzazioni utilizzate. I grafici mostrano che i risultati teorici di [Sezione 2.3.4](#) trovano accordo con l'andamento empirico dell'errore.



(a) Black & Scholes

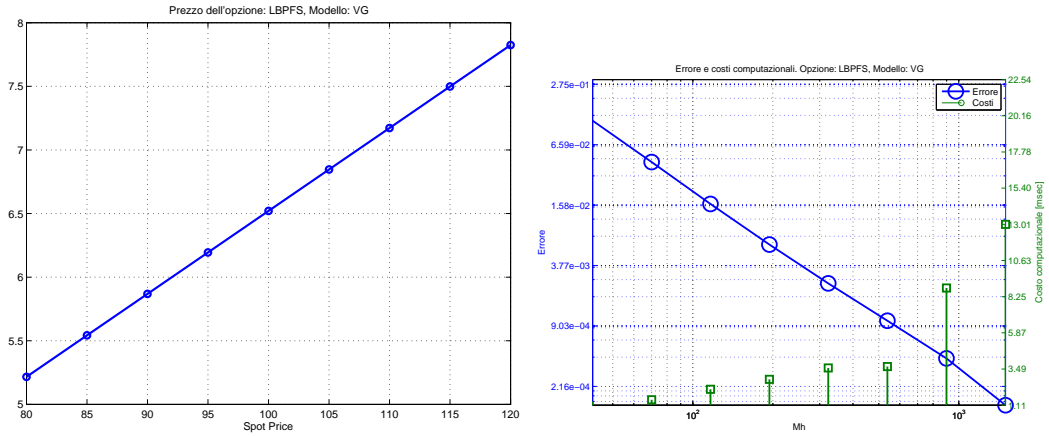


(b) Merton

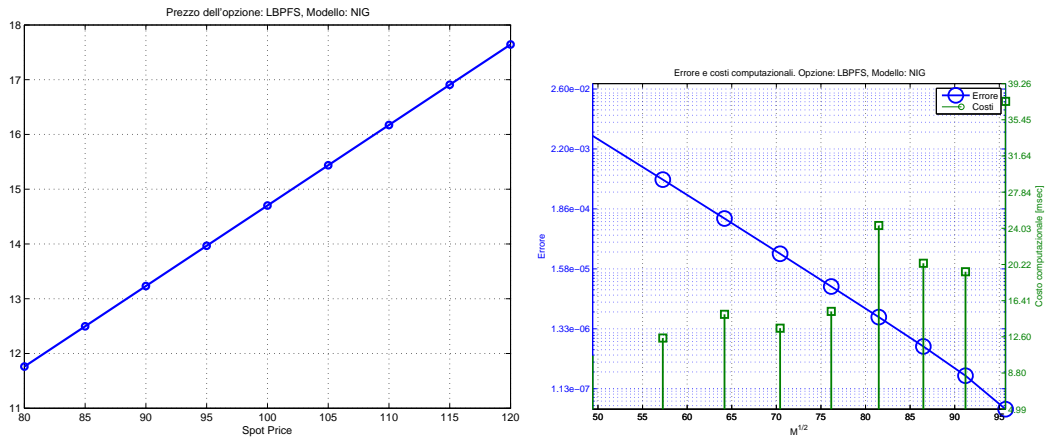


(c) Kou

Figura 3.1: Pricing Floating Strike Put, corrispondente errore e costi computazionali al variare del modello utilizzato.



(d) VG



(e) NIG

Figura 3.1: Pricing Floating Strike Put, corrispondente errore e costi computazionali al variare del modello utilizzato.

Capitolo 4

Opzioni Bermuda

In questo capitolo vengono considerate le opzioni Bermuda, quale ulteriore rappresentante della classe delle opzioni path-dependent. Questo tipo di opzioni può essere esercitato, a discrezione del possessore, non soltanto a scadenza, ma anche a predeterminate scadenze durante la vita dell'opzione. Sotto questo aspetto le Bermuda sono simili alle opzioni Americane, queste ultime però possono essere esercitate in ogni momento antecedente alla loro scadenza. Nella prima parte del capitolo le opzioni verranno prezzate considerando tutti e soli i modelli di sottostante di Lévy, per questa parte e per le dimostrazioni dei teoremi presentati si faccia riferimento a [7]. Nella seconda parte, invece, il metodo viene esteso ai più complessi modelli time changed Lévy con CIR clock, la trattazione di questi processi prende spunto invece da [18].

4.1 Opzioni Bermuda con processi di Lévy

Il focus di questa sezione è la valutazione di un'opzione Bermuda put mediante applicazione del metodo della trasformata di Hilbert. Il sottostante all'opzione viene sempre modellizzato come un processo di Lévy. Si supponga che il tempo corrente sia $t_0 = 0$ e che gli istanti di monitoraggio dell'opzione siano $\mathcal{T} = \{t_0, t_1, \dots, t_N\}$, dove $t_N = T$ è la scadenza dell'opzione. Senza perdita di generalità si assume che l'intervallo di tempo tra due date consecutive sia uniforme, cioè $t_j = j\Delta$, $j = 0, 1, \dots, N$. L'opzione Bermuda può essere esercitata in uno qualsiasi degli elementi di \mathcal{T} . Quando l'opzione viene esercitata al generico $t_n \in \mathcal{T}$, il payoff è $G(S_{t_n})$, dove $G(S) = (K - S)^+$ per una put, con K strike price dell'opzione.

4.1.1 Induzione backward

La valutazione di un'opzione Bermuda comporta la risoluzione di un problema di controllo ottimo stocastico. La valutazione al tempo 0 dell'opzione è data da

$$V^0(S_0) = \sup_{\tau} \mathbb{E}[e^{-r\tau} G(S_{\tau})], \quad (4.1)$$

dove il sup è valutato sull'insieme dei tempi d'arresto che prendono valore in \mathcal{T} . Il problema viene risolto mediante il seguente algoritmo di induzione a ritroso:

$$V^N(S) = G(S), \quad (4.2a)$$

$$V^j(S) = \max(G(S), e^{-r\tau} \mathbb{E}_{j\Delta, S}[V^{j+1}(S_{(j+1)\Delta})]), \quad 0 \leq j \leq N-1, \quad (4.2b)$$

dove $\mathbb{E}_{j\Delta, S}$ denota il valore atteso condizionato a $S_{j\Delta} = S$.

Come punto di partenza si mostrerà che per $0 \leq j \leq N-1$, esiste un unico valore $0 < S_j^* < K$ tale che $G(S)$ e $e^{-r\Delta} \mathbb{E}_{j\Delta, S}[V^{j+1}(S_{(j+1)\Delta})]$ si eguagliano. A sinistra di S_j^* , $V^j(S) = G(S)$, a destra $V^j(S)$ è data dal valore atteso scontato. Precisamente, quando il sottostante al tempo $j\Delta$ è sotto S_j^* è ottimale l'esercizio anticipato. Viceversa se l'asset è sopra S_j^* è ottimale mantenere l'opzione. L'esistenza di un unico $S_j^* \in (0, K)$ è fondamentale per sviluppare l'approccio con trasformata di Hilbert. Per $j = N$, si pone $S_j^* = K$, perciò a scadenza si esercita solamente se l'opzione è in the money. La collezione di $\{S_j^*, 0 \leq j \leq N\}$ è chiamata frontiera di esercizio anticipato. In pratica, oltre a determinare il valore dell'opzione, si vuole anche trovare la frontiera di esercizio anticipato.

Teorema 4.1. *Per ogni $0 \leq j \leq N-1$, esiste un unico $0 < S_j^* < K$ tale che V^j definito in (4.2a) soddisfa*

$$V^j(S) = G(S) \cdot \mathbb{I}_{(0, S_j^*]}(S) + e^{-r^*\Delta} \mathbb{E}_{j\Delta, S}[V^{j+1}(S_{(j+1)\Delta})] \cdot \mathbb{I}_{(S_j^*, \infty)}(S). \quad (4.3)$$

Per una sua dimostrazione si faccia riferimento a [7].

Si procede ora con un cambio di variabile $x = \ln(S/K)$ e con il processo $X_t = \ln(S_t/K)$. Si denoti con $f^0(x) = V^0(Ke^x)$ il valore dell'opzione al tempo 0 nella nuova variabile di stato x . Si indica $g(x) = G(Ke^x)$ e $x_j^* = \ln(S_j^*/K) < 0$, $0 \leq j \leq N-1$. Allora (4.2a) e il Teorema 4.1 portano alla risoluzione del seguente algoritmo backward:

$$f^N(x) = g(x) \quad (4.4a)$$

per ogni $0 \leq j \leq N-1$ risolve

$$g(x) = e^{-r\Delta} \mathbb{E}_{j\Delta, x}[f^{j+1}(X_{(j+1)\Delta})], \quad (4.4b)$$

$$f^j(x) = g(x) \cdot \mathbb{I}_{(-\infty, x_j^*]}(x) + e^{-r\Delta} \mathbb{E}_{j\Delta, x}[f^{j+1}(X_{(j+1)\Delta})] \cdot \mathbb{I}_{(x_j^*, \infty)}(x). \quad (4.4c)$$

qui $\mathbb{E}_{t, x}$ indica il valore atteso condizionato a $X_t = x$. Il valore dell'opzione al tempo 0 è trovato con $V^0(S_0) = f^0(\ln(S_0/K))$. L'algoritmo sarà implementato usando un metodo basato sulla trasformata di Fourier.

Per garantire che la trasformata di Fourier sia ben definita, si deve introdurre il fattore di decadimento esponenziale. Per $\alpha \in \mathbb{R}$, si definiscono $f_\alpha^j(x) = e^{\alpha x} f^j(x)$ e $g_\alpha(x) = e^{\alpha x} g(x)$. Come enunciato dal teorema seguente, per un appropriato α , $f_\alpha^j \in L^1(\mathbb{R})$ per ogni $0 \leq j \leq N$ cosicchè la trasformata di Fourier è ben definita.

Teorema 4.2. Sia $\alpha \in \mathcal{I}_X$ tale che $g_\alpha(x) \in L^1(\mathbb{R})$. Allora $f_\alpha^j(x) \in L^1(\mathbb{R})$ per ogni $0 \leq j \leq N$ e risolve il seguente algoritmo backward

$$f_\alpha^N(x) = g_\alpha(x) \quad (4.5a)$$

per ogni $0 \leq j \leq N - 1$ risolve

$$g_\alpha(x) = e^{-r\Delta} \phi_\Delta(i\alpha) \mathbb{E}_{j\Delta, x}^\alpha [f_\alpha^{j+1}(X_{(j+1)\Delta})], \quad (4.5b)$$

$$f_\alpha^j(x) = g_\alpha(x) \cdot \mathbb{I}_{(-\infty, x_j^*]}(x) + e^{-r\Delta} \phi_\Delta(i\alpha) \mathbb{E}_{j\Delta, x}^\alpha [f_\alpha^{j+1}(X_{(j+1)\Delta})] \cdot \mathbb{I}_{(x_j^*, \infty)}(x), \quad (4.5c)$$

dove \mathbb{E}^α denota il valore atteso sotto la misura \mathbb{P}^α , che è definita come

$$\frac{d\mathbb{P}^\alpha}{d\mathbb{P}} \Big|_{\mathcal{F}_t} = Z_t \quad (4.6)$$

con

$$Z_t = \frac{e^{-\alpha X_t}}{\phi_t(i\alpha)}. \quad (4.7)$$

Passando allo spazio di Fourier si ha il seguente

Teorema 4.3. L'induzione a ritroso (4.5) si risolve come sotto

$$\hat{f}_\alpha^N(x) = \hat{g}_\alpha(x), \quad (4.8a)$$

per ogni $0 \leq j \leq N - 1$, x_j^* risolve

$$\frac{1}{2\pi} e^{-r\Delta} \int_{\mathbb{R}} e^{-i\xi x} \hat{f}_\alpha^{j+1}(\xi) \phi_\Delta(-\xi + i\alpha) d\xi - g_\alpha(x) = 0, \quad (4.8b)$$

$$\begin{aligned} \hat{f}_\alpha^j(\xi) &= \mathcal{F}(g_\alpha \cdot \mathbb{I}_{(-\infty, x_j^*]})(\xi) + e^{-r\Delta} \left(\frac{1}{2} \hat{f}_\alpha^{j+1}(\xi) \phi_\Delta(-\xi + i\alpha) + \dots \right. \\ &\quad \left. + \frac{i}{2} e^{i\xi x_j^*} \mathcal{H}(e^{-i\eta x_j^*} \hat{f}_\alpha^{j+1}(\eta) \phi_\Delta(-\eta + i\alpha))(\xi) \right), \end{aligned} \quad (4.8c)$$

$$f_\alpha^0(x) = \max \left(g_\alpha(x), \frac{1}{2\pi} e^{-r\Delta} \int_{\mathbb{R}} e^{-i\xi x} \hat{f}_\alpha^1(\xi) \phi_\Delta(-\xi + i\alpha) d\xi \right). \quad (4.8d)$$

Nell'ultimo step si calcola f_α^0 da \hat{f}_α^1 tramite la trasformata inversa di Fourier.

4.1.2 Approssimazione discreta

Metodo Sinc

Nell'induzione backward (4.8) si devono valutare le trasformate di Fourier (4.8b) e (4.8d) e la trasformata di Hilbert in (4.8c). Nello specifico f viene approssimata da

$$f_{h,M}(x) = \frac{1}{2\pi} \sum_{m=-M}^M e^{-ixmh} \hat{f}(mh)h, \quad (4.9)$$

dove h è il passo della griglia di Fourier ed M è il livello di troncamento. L'errore di discretizzazione si può dimostrare che decade esponenzialmente.

Inoltre si deve valutare anche

$$\hat{g}(\xi) = \mathcal{H}(e^{-i\eta x} \hat{f}(\eta))(\xi). \quad (4.10)$$

In modo simile (4.10) si discretizza come

$$\hat{g}_{h,M}(\xi) = \sum_{m=-M}^M e^{-ixmh} \hat{f}(mh) \frac{1 - \cos(\pi(\xi - mh)/h)}{\pi(\xi - mh)/h}. \quad (4.11)$$

Anche in questo caso che l'errore di discretizzazione decade esponenzialmente.

4.1.3 Implementazione

Per un'opzione Bermuda put $g_\alpha(x) = Ke^{\alpha x}(1 - e^x)^+ \in L^1(\mathbb{R})$ per ogni $\alpha > 0$. In accordo con il Teorema 4.2, il fattore di smorzamento deve essere $\alpha \in (0, \lambda_+)$. L'algoritmo (4.8) diventa

Step1

Si inizia con (4.8a):

$$\hat{f}_\alpha^N(\xi) = \frac{K}{(\alpha + i\xi)(\alpha + 1 + i\xi)}, \quad \xi = -Mh, \dots, Mh. \quad (4.12)$$

Step2

Si risolve (4.8b) usando il metodo di Newton-Raphson. Come guess iniziale si prende $x_N^* = 0$. Quando si risolve per trovare x_j^* , $j = N-2, \dots, 1$, si inizia con guess iniziale x_{j+1}^* . Indicando la generica guess iniziale con \tilde{x} , si valuta il membro di sinistra di (4.8b) e la sua derivata rispetto a x nel punto \tilde{x} usando la formula dei trapezi con passo h e livello di troncamento M , ottenendo

$$f(\tilde{x}) = \frac{1}{2\pi} \sum_{m=-M}^M e^{-ixmh} \hat{f}_\alpha^{j+1}(mh) \phi_\Delta(-mh + i\alpha)h - g_\alpha(\tilde{x}),$$

$$f'(\tilde{x}) = \frac{-i}{2\pi} \sum_{m=-M}^M e^{-ixmh} mh \hat{f}_\alpha^{j+1}(mh) \phi_\Delta(-mh + i\alpha)h - g'_\alpha(\tilde{x}),$$

dove $g'_\alpha(x) = Ke^{\alpha x}(\alpha - (1 + \alpha)e^x)$. Sostituendo poi \tilde{x} con $\tilde{x} - f(\tilde{x})/f'(\tilde{x})$ e ripetendo finantochè $|f(\tilde{x})/f'(\tilde{x})| > \epsilon_R$ con ϵ_R tolleranza fissata.

Step3

Il primo termine di (4.8c) è dato da

$$\mathcal{F}(g_\alpha \cdot \mathbb{I}_{(-\infty, x_j^*]})(\xi) = K e^{(i\xi + \alpha)x_j^*} \left(\frac{1}{i\xi + \alpha} - \frac{e^{x_j^*}}{i\xi + \alpha + 1} \right). \quad (4.13)$$

Usando la (4.11), si calcola $\hat{f}_\alpha^j(\xi)$ come:

$$\begin{aligned} \hat{f}_\alpha^j(\xi) &= \mathcal{F}(g_\alpha \cdot \mathbb{I}_{(-\infty, x_j^*]})(\xi) + e^{-r\Delta} \left(\frac{1}{2} \hat{f}_\alpha^{j+1}(\xi) \phi_\Delta(-\xi + i\alpha) + \dots \right. \\ &\quad \left. + \frac{i}{2} e^{i\xi x_j^*} \sum_{M=-m}^M e^{-ix_j^* mh} \hat{f}_\alpha^{j+1}(mh) \phi_\Delta(-mh + i\alpha) \frac{1 - \cos(\pi(\xi - mh)/h)}{\pi(\xi - mh)/h} \right), \end{aligned} \quad (4.14)$$

con $\xi = -Mh, \dots, Mh$.

Si ripete quindi lo **Step2** e lo **Step3** per $j = N - 1, \dots, 1$.

Step4

La trasformata inversa di Fourier in (4.8d) si approssima usando la formula dei trapezi:

$$f_\alpha^0(x) = \max \left(g_\alpha(x), \frac{1}{2\pi} e^{-r\Delta} \sum_{m=-M}^M e^{-ixmh} \hat{f}_\alpha^1(mh) \phi_\Delta(-mh + i\alpha)h \right). \quad (4.15)$$

Fine

Il costo computazionale è racchiuso nello **Step3**, dove si deve valutare $\hat{f}_\alpha^j(\xi)$ per $\xi = -Mh, \dots, Mh$. Se implementata direttamente, la moltiplicazione matrice vettore ha un costo computazionale pari a $O(M^2)$, ridotto a $O(M \log(M))$ se viene utilizzato l'algoritmo precedentemente introdotto basato sulla trasformata di Fourier.

4.1.4 Test numerici

In questa sezione si illustrano le performance del metodo. Ai fini dell'analisi si è considerato il pricing di un'opzione Bermuda put al variare dei modelli introdotti nel [Capitolo 1](#). Per quanto riguarda i parametri del modello di Black&Scholes si è scelto $\sigma = 0.2$; per il modello di Merton $\sigma = 0.1$, $\lambda = 3$, $\mu = -0.05$, $\delta = 0.086$; per il modello Kou $\sigma = 0.1$, $\lambda = 3$, $p = 0.3$, $\lambda_p = 20$, $\lambda_m = 20$; per il modello VG $\sigma = 0.1$, $\nu = 0.1$, $\theta = -0.2$; per il modello NIG $\alpha = 15$, $\beta = -5$, $\delta = 0.5$. La maturity è $T = 1$ e lo strike price è $K = 100$. Il tasso privo di rischio $r = 5\%$ e lo dividend yield $q = 2\%$. L'opzione è monitorata per $N = 252$ e $\Delta = 1/252$. Si seleziona $\alpha = 5$ in tal modo la convergenza esponenziale è massimizzata. In [Figura 4.2](#) sono riportati i grafici della valutazione dell'opzione al variare del modello e la frontiera di esercizio anticipato, mentre in [Figura 4.3](#) sono riportati gli andamenti dell'errore

di discretizzazione dove il prezzo ottenuto con la griglia più fitta è di volta in volta preso come riferimento.

Anche in questo caso l'andamento teorico ricavato in [7] è in accordo con quanto empiricamente osservabile dai grafici seguenti.

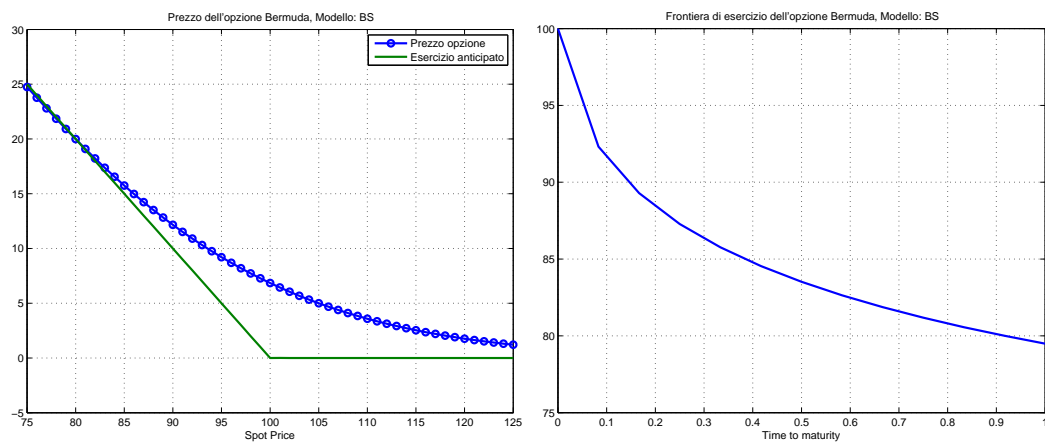


Figura 4.1: Valore di un'opzione Bermuda put e corrispondente frontiera di esercizio anticipato.

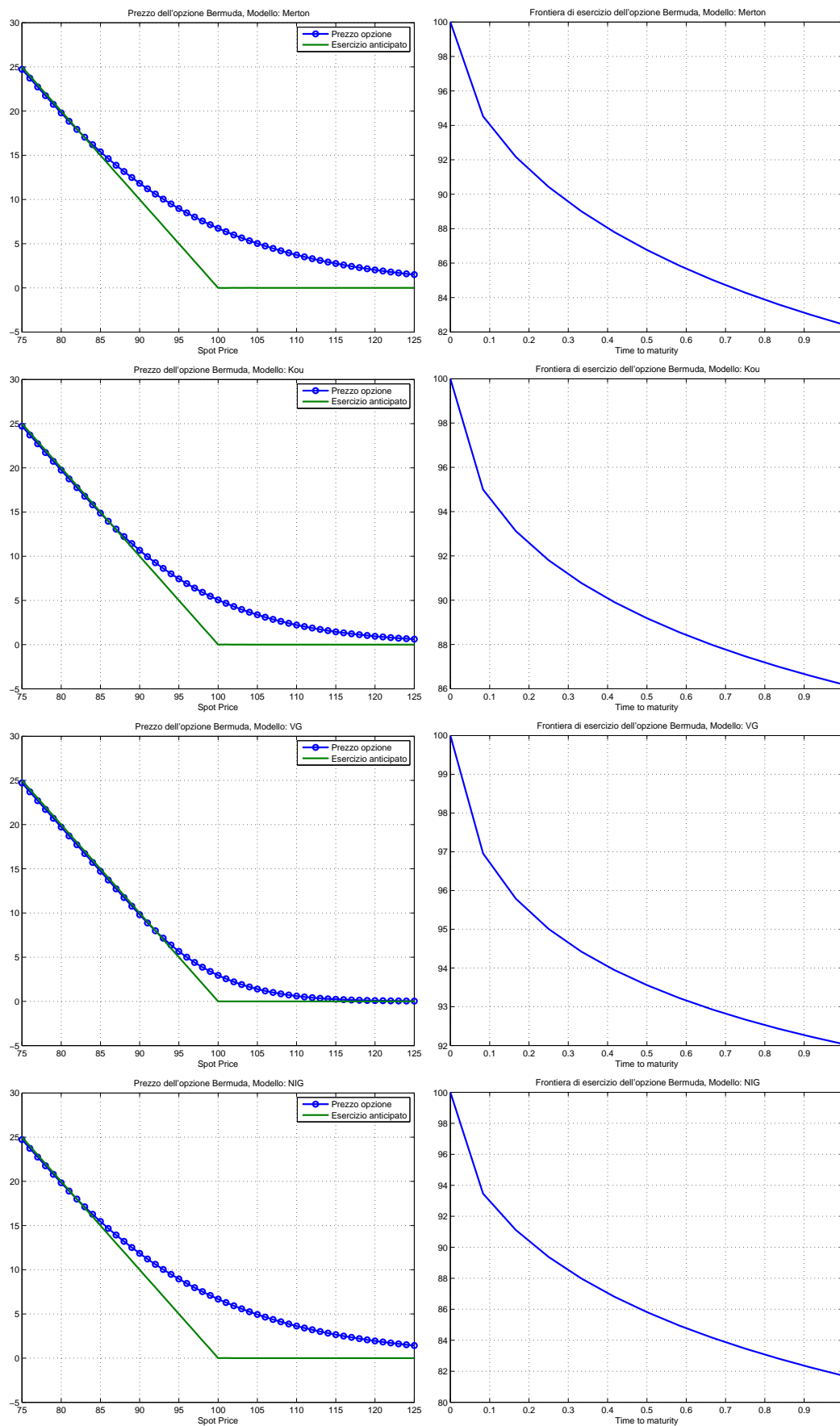


Figura 4.2: Valore di un'opzione Bermuda put e corrispondente frontiera di esercizio anticipato.

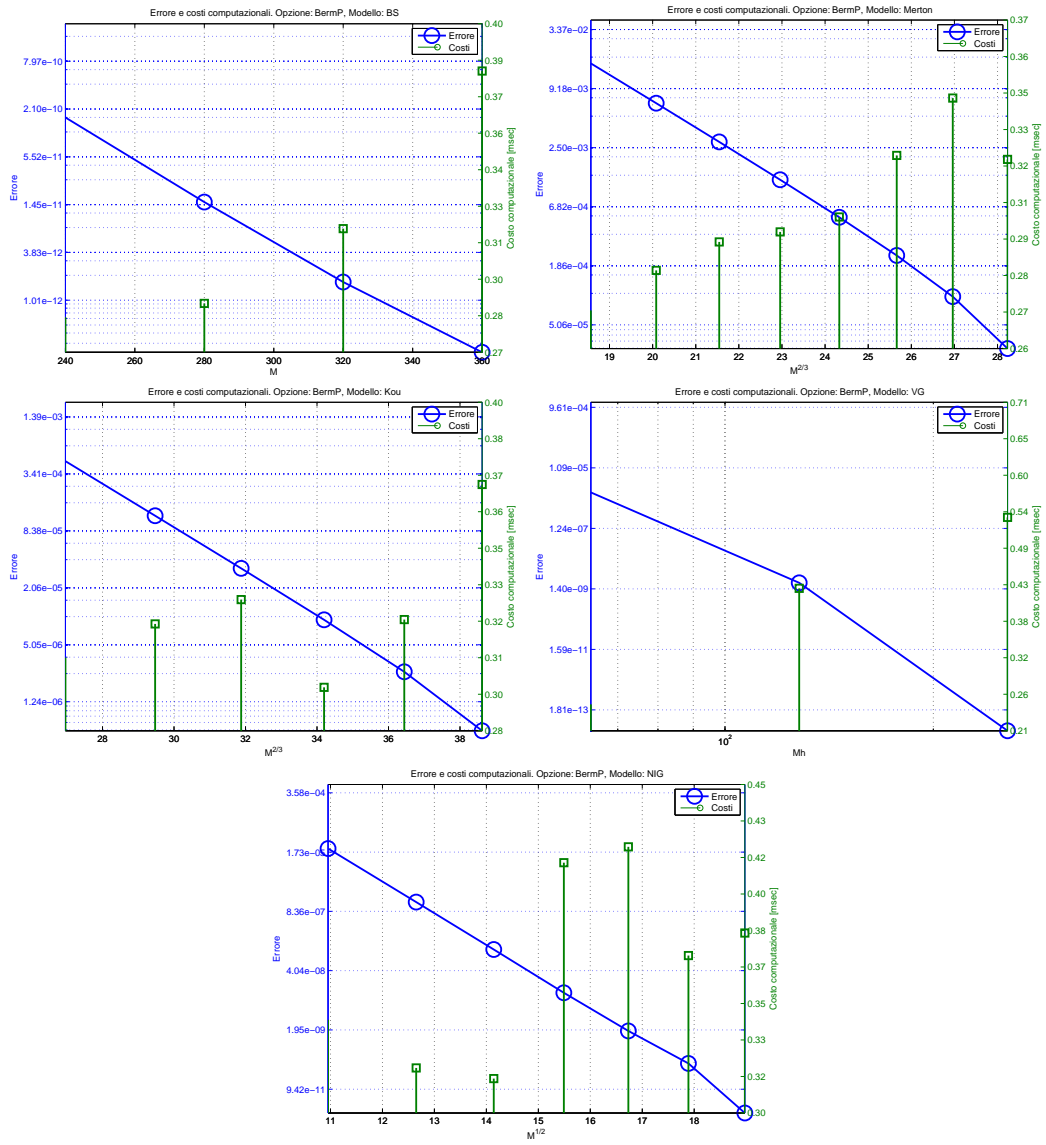


Figura 4.3: Convergenza e costi computazionali del metodo FHT.

4.2 Opzioni Bermuda con processi time-changed Lévy

In questa sezione, si estende, sempre per le opzioni Bermuda, quanto visto nella sezione precedente al caso di dinamica dettata da processi time-changed Lévy.

4.2.1 Introduzione

L'algoritmo di pricing è sviluppato applicando il metodo FHT nella dimensione del log-asset return e adottando una formula di quadratura per l'integrazione numerica del tasso di attività logaritmico del cambio stocastico di scala temporale. Si supponga che il tempo iniziale sia $t_0 = 0$ e l'insieme dei tempi di monitoraggio sia $\mathcal{T} = \{t_1, t_2, \dots, t_N\}$, dove $t_N = T$ è la scadenza. Assumendo uno step temporale uniforme Δ , si pone $t_k = k\Delta$, con $k = 0, 1, \dots, N$. L'opzione Bermuda, per sua definizione, può essere esercitata per ogni $t_k \in \mathcal{T}$ con il payoff di esercizio $\tilde{G}(S_{t_k})$. Si noti che $\tilde{G}(S) = (S - K)^+$ è il payoff di una call europea, mentre $\tilde{G}(S) = (K - S)^+$ è il payoff di una put, con K prezzo di esercizio. Qui ci si focalizzerà sul pricing di opzioni Bermuda put, ma un approccio simile si può applicare anche per le opzioni Bermuda call. Dal pricing di opzioni Bermuda si può derivare anche il pricing di opzioni Americane facendo il limite che tende a zero del tempo tra due date di monitoraggio successive, cioè $\Delta \rightarrow 0$.

Determinazione dei prezzi critici

Il problema di arresto ottimo per un'opzione Bermuda put è caratterizzato dalla determinazione dei prezzi critici, in tutte le date di monitoraggio, sotto i quali è ottimo esercitare l'opzione. Il principale step nell'algoritmo FHT è la determinazione del prezzo critico $S_k^*(\gamma_{t_k})$ alla data di monitoraggio t_k e a livello γ_{t_k} del tasso di attività logaritmico, con $k = 1, 2, \dots, N$. Si assume, inoltre, che l'asset sottostante non paghi dividendi. È noto che il prezzo critico $S_N^*(\gamma_{t_N})$, a scadenza t_N , è K . I prezzi critici $S_k^*(\gamma_{t_k})$, con $k = N - 1, N - 2, \dots, 1$, vengono determinati successivamente utilizzando l'induzione backward. Prima di tutto, si vuole stabilire l'esistenza di un unico valore $S_k^*(\gamma_{t_k})$, tale che $0 < S_k^*(\gamma_{t_k}) < K$ per $k = N - 1, N - 2, \dots, 1$, per il seguente teorema si faccia riferimento a [18].

Proposizione 4.1. *Assumendo che $r \geq 0$ per $k = 1, 2, \dots, N - 1$ allora esiste un unico prezzo critico $S_k^*(\gamma_{t_k})$, dove $0 < S_k^*(\gamma_{t_k}) < K$, tale che il valore dell'opzione Bermuda put $\tilde{V}(S_{t_k}, \gamma_{t_k}, t_k)$ soddisfa*

$$\tilde{V}(S_{t_k}, \gamma_{t_k}, t_k) = \tilde{G}(S_{t_k})\mathbb{I}_{\{0, S_k^*(\gamma_{t_k})\}} + e^{-r\Delta}\mathbb{E}_{t_k}[\tilde{V}(S_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1})]\mathbb{I}_{\{S_k^*(\gamma_{t_k}), \infty\}}, \quad (4.16)$$

dove \mathbb{E}_{t_k} denota il valore atteso condizionato rispetto a S_{t_k} e γ_{t_k} . Per la dimostrazione di questa proposizione si faccia riferimento a [18]. All'istante di monitoraggio t_k , per ogni dato livello γ_{t_k} del tasso di attività logaritmico, è ottimo l'esercizio dell'opzione Bermuda put quando S_{t_k} è nella regione $(0, S_k^*(\gamma_{t_k})]$, mentre l'opzione continua a vivere quando S_{t_k} è nella regione di continuazione $(S_k^*(\gamma_{t_k}), \infty)$. La regione di

esercizio e la regione di continuazione sono quindi separate dal prezzo critico $S_k^*(\gamma_{t_k})$ e il valore di continuazione $\tilde{C}(S_{t_k}, \gamma_{t_k}, t_k)$ è dato da $e^{-r\Delta} \mathbb{E}_{t_k}[\tilde{V}(S_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1})]$.

4.2.2 Induzione backward

Si consideri il processo log-asset return normalizzato al prezzo di esercizio K sotto l'ipotesi di processo time-changed Lévy X_{T_t} , (1.45).

Per un'opzione Bermuda put, il payoff di esercizio a una data di monitoraggio t_k è dato da $G(Y_{t_k}) = \tilde{G}(S_{t_k}) = K(1 - e^{Y_{t_k}})^+$. Si scrive il valore dell'opzione Bermuda put e il valore di continuazione rispettivamente come $V(Y_t, \gamma_t, t) = \tilde{V}(Y_t, \gamma_t, t)$ e $C(Y_t, \gamma_t, t) = \tilde{C}(S_t, \gamma_t, t)$. A una data di monitoraggio il prezzo dell'opzione Bermuda put è data dal massimo tra il valore di continuazione e il payoff di esercizio. Date N date di monitoraggio, $\mathcal{T} = \{t_k | k = 1, \dots, N\}$ con $t_n = T$, la formula di pricing dell'opzione Bermuda put si esprime come

$$V(Y_t, \gamma_t, t) = \begin{cases} G(Y_{t_N}) & \text{per } k = N \\ \max\{C(Y_{t_k}, \gamma_{t_k}, t_k), G(Y_{t_k})\} & \text{per } k = 1, 2, \dots, N - 1. \\ C(Y_{t_0}, \gamma_{t_0}, t_0) & \text{per } k = 0 \end{cases} \quad (4.17)$$

Valore di continuazione

Rispetto al caso delle opzioni Barriera occorre a questo punto introdurre un ulteriore passaggio, ovvero il calcolo del valore di continuazione $x_k^*(\gamma_{t_k})$. Si esprime il valore di continuazione $C(Y_{t_k}, \gamma_{t_k}, t_k)$ come doppio valore atteso e si valuta il valore atteso nella dimensione del tasso di attività logaritmico usando un'integrazione numerica. Usando la tower property e un'appropriata formula di quadratura, il valore di continuazione a una data di monitoraggio t_k è data da

$$\begin{aligned} C(Y_{t_k}, \gamma_{t_k}, t_k) &= e^{-r\Delta} \mathbb{E} [\mathbb{E}[V(Y_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1}) | \mathcal{F}_{t_k}, \gamma_{t_{k+1}}] | \mathcal{F}_{t_k}] \\ &\approx e^{-r\Delta} \sum_{j=1}^J \omega_j p_\gamma(\zeta_j | \gamma_{t_k}) \mathbb{E}[V(Y_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1}) | \mathcal{F}_{t_k}, \gamma_{t_{k+1}} = \zeta_j]. \end{aligned} \quad (4.18)$$

dove ζ_j , con $j = 1, 2, \dots, J$ sono i nodi della formula di quadratura per il tasso di attività logaritmico. Applicando l'usuale procedura di smorzamento alle funzioni valori, si definisce $C_\alpha(Y_t, \gamma_t, t) = e^{\alpha Y_t} C(Y_t, \gamma_t, t)$, $V_\alpha(Y_t, \gamma_t, t) = e^{\alpha Y_t} V(Y_t, \gamma_t, t)$ e $G_\alpha(Y_t) = e^{\alpha Y_t} G(Y_t)$. Si scrive il valore atteso interno in termini di trasformata di Fourier inversa

$$\begin{aligned} &\mathbb{E}[V(Y_{t_{k+1}}, \gamma_{t_{k+1}}, t_{k+1}) | \mathcal{F}_{t_k}, \gamma_{t_{k+1}} = \zeta_j] \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-\omega Y_{t_k}} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \Psi(-\omega; \zeta_j, \gamma_{t_k}) d\beta, \end{aligned} \quad (4.19)$$

dove $\omega = \alpha + i\beta$ e $\hat{V}_\alpha(\beta, \gamma_t, t)$ denota la trasformata di Fourier generalizzata di $V(Y_t, \gamma_t, t)$.

Combinando (4.18) e (4.19) si ottiene la seguente formula di approssimazione

$$C_\alpha(Y_{t_k}, \gamma_{t_k}, t_k) \approx \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-i\beta Y_{t_k}} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\omega; \zeta_j, \gamma_{t_k}) d\beta, \quad (4.20)$$

dove $\tilde{\Psi}(\omega; \zeta_j, \gamma_{t_k}) = \Psi(\omega; \zeta_j, \gamma_{t_k}) p_\gamma(\zeta_j | \gamma_{t_k})$.

Algoritmo trasformata di Fourier

Questa sezione ripercorre, con minime differenze, dovute essenzialmente all'introduzione del calcolo del valore di continuazione, quanto fatto nell'analoga sezione riferita alle opzioni Barriera. Date N date di monitoraggio $\mathcal{T} = \{t_k | k = 1, 2, \dots, N\}$ con $t_N = T$, la procedura di induzione backward per il pricing di un'opzione bermuda put si riassume nel seguente schema

1. Inizializzazione della funzione valore a partire dal payoff a scadenza

$$V_\alpha(Y_{t_N}, \gamma_{t_N}, t_N) = G_\alpha(Y_{t_N}). \quad (4.21)$$

2. Determinazione dei prezzi critici e calcolo del valore dell'opzione con induzione backward. Sia $x_k^*(\gamma_{t_k}) = \ln S_k^*(\gamma_{t_k})$, ai prezzi critici $x_k^*(\gamma_{t_k})$ si ha l'uguaglianza del payoff di esercizio e del valore di continuazione. Per $k = N-1, N-2, \dots, 1$, $x_k^*(\gamma_{t_k})$ risolve l'equazione algebrica

$$G_\alpha(x) = \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-i\beta x} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \gamma_{t_k}) d\beta. \quad (4.22)$$

Il valore dell'opzione al tempo t_k si approssima come

$$\begin{aligned} V_\alpha(Y_{t_k}, \gamma_{t_k}, t_k) &\approx \mathbb{I}_{(-\infty, x_k^*(\gamma_{t_k})]} G_\alpha(Y_{t_k}) \\ &+ \mathbb{I}_{(x_k^*(\gamma_{t_k}), \infty)} \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-i\beta Y_{t_k}} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \gamma_{t_k}) d\beta. \end{aligned} \quad (4.23)$$

4.2.3 Il metodo della trasformata veloce di Hilbert

Anche in questo caso le differenze con l'analoga procedura per le opzioni Barriera sono minime, per completezza viene comunque riportata la derivazione dell'algoritmo che verrà implementato. L'induzione a ritroso nel dominio di Fourier procede come segue. Si inizia con $k = N$, dove

$$\hat{V}_\alpha(\beta, \zeta_j, t_N) = \hat{G}_\alpha(\beta), \quad j = 1, 2, \dots, J. \quad (4.24)$$

Per $k = N - 1, N - 2, \dots, 1$ si deve applicare un metodo efficiente per trovare le radici, come per esempio il metodo di Newton-Raphson, al fine di ottenere gli $x_{k,p}^*$ alla (4.22) in ognuno dei nodi del tasso di attività logaritmico $\gamma_{t_k} = \zeta_p$, con $p = 1, 2, \dots, J$. L'approssimazione numerica di $\hat{V}_\alpha(\beta, \zeta_p, t_k)$ è data in termini della trasformata di Hilbert

$$\begin{aligned} V_\alpha(\beta, \zeta_p, t_k) &\approx \mathcal{F} \left(G_\alpha(Y_{t_k}) \mathbb{I}_{(-\infty, x_{k,p}^*]} \right) (\beta) \\ &+ \frac{e^{-r\Delta}}{2} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \zeta_p) \\ &+ \frac{i e^{-r\Delta}}{2} e^{i\beta x_{k,p}^*} \mathcal{H} \left(e^{-i\eta x_{k,p}^*} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\eta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\eta; \zeta_j, \gamma_{t_k}) \right) (\beta), \end{aligned} \quad (4.25)$$

per $p = 1, 2, \dots, J$. L'approssimazione numerica del valore dell'opzione Bermuda put al tempo iniziale è data da

$$V(Y_{t_0}, \zeta_p, t_0) \approx \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-(\alpha+i\beta)Y_{t_0}} \hat{V}_\alpha(\beta, \zeta_j, t_1) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \zeta_p) d\beta. \quad (4.26)$$

La dimostrazione dell'induzione backward è la seguente. Prima di tutto è necessario determinare i prezzi critici. Ponendo $\gamma_{t_k} = \zeta_p$, con $p = 1, 2, \dots, J$, si utilizza la (4.22) per ottenere i prezzi critici. Si noti che la trasformata di Fourier di $\int_{-\infty}^{\infty} e^{-i\beta Y_{t_k}} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \gamma_{t_k}) d\beta$ è il prodotto di $\hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \gamma_{t_k})$. Applicando la trasformata di Fourier a entrambi i lati di (4.23) si ottiene:

$$\begin{aligned} \hat{V}_\alpha(\beta, \gamma_{t_k}, t_k) &\approx \mathcal{F} \left(G_\alpha(Y_{t_k}) \mathbb{I}_{(-\infty, x_k^*(\gamma_{t_k})]} \right) (\beta) \\ &+ \frac{e^{-r\Delta}}{2} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \gamma_{t_k}) \\ &+ \frac{i e^{-r\Delta}}{2} e^{i\beta x_k^*(\gamma_{t_k})} \mathcal{H} \left(e^{-i\eta x_k^*(\gamma_{t_k})} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\eta, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - i\eta; \zeta_j, \gamma_{t_k}) \right) (\beta). \end{aligned} \quad (4.27)$$

Ponendo $\gamma_{t_k} = \zeta_p$, $p = 1, 2, \dots, J$, si ottiene la (4.24). Infine, siccome la trasformata inversa di Fourier di $e^{-r\Delta} \mathbb{E}[V(Y_{t_1}, \zeta_j, t_1) | \mathcal{F}_{t_0}]$ fornisce il valore di continuazione al tempo t_0 [(4.20)], si ottiene il valore dell'opzione Bermuda put all'istante iniziale come mostrato in (4.26).

Approssimazione discreta

Ripercorrendo passaggi analoghi a quanto visto nella Sezione 2.4 sulle opzioni Barriera, si perviene alla seguente relazione:

$$\begin{aligned} \mathbf{k}_p(t_k) &= [\hat{V}_\alpha(t_{k+1}) \cdot \tilde{\Psi}(\zeta_p)] \mathbf{w}, \quad p = 1, 2, \dots, J; \\ B(t_k) &= [\mathbf{k}_1(t_k), \mathbf{k}_2(t_k), \dots, \mathbf{k}_J(t_k)], \end{aligned}$$

dove $\mathbf{k}_p(t_k)$ è un vettore colonna di dimensione $2M + 1$, $B(t_k)$ è una matrice $(2M + 1) \times J$ e l'operatore \cdot denota il prodotto elemento per elemento. Siano H, F e \hat{G} matrici $(2M + 1) \times J$, i cui elementi sono rispettivamente $H_{m,p}(t_k) = e^{-imhx_{k,p}^*}$, $F_{m,p}(t_k) = e^{imhx_{k,p}^*}$ e $\hat{G}_{m,p}(t_k) = \mathcal{F}\left(G_\alpha(Y_{t_k}, t_k)\mathbb{I}_{(-\infty, x_{k,p}^*]}\right)(mh)$. Si noti che in questo caso, rispetto alle opzioni Barriera, le matrici appena introdotte non sono più costanti e vanno quindi ricalcolate ad ogni iterazione. Scrivendo nuovamente l'induzione backward nel dominio di Fourier nella forma matrice/vettore si ottiene:

$$\hat{V}_\alpha(t_k) = \hat{G}(t_k) + \frac{e^{-r\Delta}}{2}B(t_k) + \frac{e^{-r\Delta}i}{2}[T(B(t_k) \cdot H(t_k))] \cdot F(t_k). \quad (4.28)$$

Grazie alla struttura della matrice di Toeplitz l'algoritmo FFT può essere applicato raggiungendo una complessità di $O(M \log_2 M)$ nella dimensione del tasso di attività logaritmico.

La procedura per trovare le radici e ottenere i prezzi critici

Come già detto l'algoritmo FHT per il pricing di un'opzione Bermuda put comporta la determinazione dei prezzi critici, questo particolare rappresenta una differenza rispetto al caso delle opzioni Barriera. Si adotta quindi un metodo per trovare le radici, come ad esempio il metodo di Newton-Raphson, per risolvere l'equazione (4.22) per i vari valori di γ_{t_k} che sono posti uguali a ζ_p , con $p = 1, 2, \dots, J$. Per esempio, si possono prendere come guess iniziali $x_{k,p}^* = x_{k+1,p}^*$ o $x_{k,p}^* = x_{k,p-1}^*$. Si supponga che $f(x) = 0$ sia l'equazione di cui bisogna trovare le radici. Se l'iterazione corrente è \tilde{x} , il metodo di Newton-Raphson determina l'iterazione successiva rimpiazzando \tilde{x} con $\tilde{x} - \frac{f(\tilde{x})}{f'(\tilde{x})}$. In questo caso, si approssima $f(x)$ e $f'(x)$ usando la formula del trapezio con step h e livello di troncamento M , come segue

$$f(x) = \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \sum_{m=-M}^M e^{-imhx} \hat{V}_\alpha(mh, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) h - G_\alpha(x),$$

$$f'(x) = -\frac{e^{-r\Delta}i}{2\pi} \sum_{j=1}^J \omega_j \sum_{m=-M}^M e^{-imhx} mh \hat{V}_\alpha(mh, \zeta_j, t_{k+1}) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) h - G'_\alpha(x),$$

dove $G'_\alpha(x) = Ke^{\alpha x}[\alpha - (1 + \alpha)e^x]$. La procedura per trovare le radici si ripete fintantochè $|\frac{f(\tilde{x})}{f'(\tilde{x})}| < \epsilon$, dove ϵ è la tolleranza arbitrariamente piccola.

Trasformata di Fourier generalizzata del payoff condizionatamente all'esercizio anticipato

Per un'opzione Bermuda put, il payoff di esercizio è dato da $G_\alpha(Y_{t_k}) = Ke^{\alpha Y_{t_k}}(1 - e^{Y_{t_k}})^+ \in L^1(\mathbb{R})$, per ogni $\alpha > 0$, $k = 1, 2, \dots, N$. Ricordando che $\alpha \in \mathcal{L}_X$ e che $\alpha > 0$, si ha $\alpha \in (0, \lambda_+)$. Si calcola quindi la trasformata di Fourier generalizzata del prodotto del payoff di esercizio e della funzione indicatrice tramite la formula

della trasformata di Fourier

$$\mathcal{F}\left(G_\alpha(Y_{t_k}, t_k)\mathbb{I}_{(-\infty, x_{k,p}^*]}\right)(\beta) = \begin{cases} \frac{K}{(\alpha + i\beta)(\alpha + i\beta + 1)} & k = N \\ K\left(\frac{1}{(\alpha + i\beta)} - \frac{e^{x_{k,p}^*}}{(\alpha + i\beta + 1)}\right) & k = N - 1, \dots, 1 \end{cases} \quad (4.29)$$

dove $\beta = -Mh, \dots, Mh$.

La versione finale dell'algoritmo FHT è riassunta nel seguente schema.

4.2.4 Sintesi dell'algoritmo numerico di pricing

Step1: Preparazione

Calcolare $\hat{V}_\alpha(t_N)$ usando la formula analitica (4.29) per $k = N$;

Preparare la matrice $\tilde{\Psi}(\zeta_p)$ per $p = 1, 2, \dots, J$;

Calcolare la prima riga e colonna della matrice di Toeplitz.

Step2: Induzione backward

Determinare i prezzi critici tramite il metodo di Newton-Raphson per trovare le radici;

Aggiornare $H(t_k)$, $F(t_k)$ e $\hat{G}(t_k)$;

Calcolare $\mathbf{k}_p(t_k) = [\hat{V}_\alpha(t_{k+1}) \cdot \tilde{\Psi}(\zeta_p)]\mathbf{w}$, per $p = 1, 2, \dots, J$ e aggiornare $B(t_k)$;

Calcolare $\hat{V}_\alpha(t_k)$ tramite (4.28) usando l'algoritmo FFT per $k = N - 1, \dots, 1$.

Step3: Valore iniziale dell'opzione

Calcolare il valore iniziale dell'opzione tramite (2.90). Utilizzare infine un'interpolazione spline per ottenere $V(Y_{t_0}, \gamma_{t_0}, t_0)$.

4.2.5 Complessità computazionale

Si è adottata la formula di quadratura dei trapezi nella dimensione del tasso di attività logaritmico. Lo sforzo computazionale nella preparazione di nodi non equidistanti è dominata dal calcolo delle matrici $\tilde{\Psi}(\zeta_p)$ per $p = 1, 2, \dots, J$, che richiede $O(\mathcal{A}MJ^2)$ operazioni. Si determinano poi i prezzi critici nel loop principale dell'algoritmo con il metodo di Newton-Raphson. I test numerici mostrano che sono sufficienti 3 o 4 iterazioni per giungere ad una soluzione accurata entro un limite di tolleranza buono. Il costo computazionale del precedente step è di $O(MJ^2)$. Infatti, la computazione del loop principale è dominata dalla valutazione numerica degli elementi di $[T(B(t_k) \cdot H(t_k))] \cdot I(t_k)$. Siccome il calcolo delle componenti del vettore $\mathbf{k}_p(t_k)$ costa $O(MJ)$ operazioni, questo comporta una complessità di $O(MJ^2)$ per il calcolo degli elementi della matrice $B(t_k)$. Dunque, il costo computazionale

complessivo per $[T(B(t_k) \cdot H(t_k))] \cdot I(t_k)$ è di $O(M \log_2 MJ^2)$, che è ottenuto grazie alla struttura speciale della matrice di Toeplitz T e all'uso della trasformata veloce di Fourier. Come risultato, la complessità computazionale totale di FHT è di $O(\max[\mathcal{A}, N \log_2 M]MJ^2)$.

Osservazioni

1. L'algoritmo FHT può essere utilizzato per ottenere simultaneamente i valori delle opzioni barriera per vari valori iniziali di S_0 senza costi addizionali. Anche delta e gamma dell'opzione si ricavano essenzialmente senza ulteriori sforzi. Specificatamente, l'ultimo step di (4.26) si modifica parzialmente per il calcolo di delta in

$$\begin{aligned} & \frac{\partial V(Y_{t_0}, \zeta_p, t_0)}{\partial S_0} \\ & \approx -\frac{e^{-r\Delta}}{2\pi S_0} \sum_{j=1}^J \omega_j \sum_{m=-M}^M (\alpha + imh) e^{-(\alpha+imh)Y_{t_0}} \hat{V}_\alpha(mh, \zeta_j, t_1) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) h. \end{aligned}$$

2. Nell'algoritmo FHT la stessa matrice di Toeplitz può essere utilizzata da un passo temporale all'altro. Solamente due runs della trasformata veloce di Fourier sono richiesti per ogni passo temporale, mostrando un errore di pricing che decade esponenzialmente.

4.2.6 Test numerici

In questa sezione si vuole mostrare la performance dell'algoritmo FHT per il pricing di un'opzione barriera sotto l'ipotesi di processo time-changed Lévy: come rappresentante di questa classe di modelli viene scelto il processo di Heston a volatilità stocastica. L'opzione scelta ha $K = 10$, $T = 0.25$, $r = 0.1$, $N = 60$, i parametri utilizzati per la dinamica Heston del sottostante sono $\lambda = 5$, $\eta = 0.9$, $\bar{v} = 0.16$, $v_0 = 0.0625$, $\rho = 0.1$, mentre il livello di troncamento per la discretizzazione della trasformata di Hilbert è $M = 2^6$.

In [Figura 4.4](#) viene riportato l'andamento dei prezzi ottenuto per diversi livelli della volatilità iniziale, con $S_0 = 8 : 0.5 : 12$. Come per i casi precedenti si nota un accordo tra quanto previsto dalla teoria e i risultati empirici.

In [Tabella 4.1](#) vengono confrontati i prezzi di opzioni Bermuda con $S_0 = 8$ e date di monitoraggio $N = 10, 20, 40$ ottenuti mediante l'applicazione dell'algoritmo FHT con i prezzi benchmark riportati in [\[18\]](#). Si noti come le performance numeriche di FHT dipendano dal comportamento della coda della funzione generatrice dei momenti condizionata $\Psi_\Delta(-\alpha - i\beta; \xi_j, \xi_p)$. Per piccoli valori di Δ essa decade lentamente, dunque un valore elevato di M è necessario per raggiungere un certo livello di tolleranza nell'errore di troncamento. Perciò il pricing di opzioni Bermuda con un'alta frequenza di monitoraggio è computazionalmente più impegnativo.

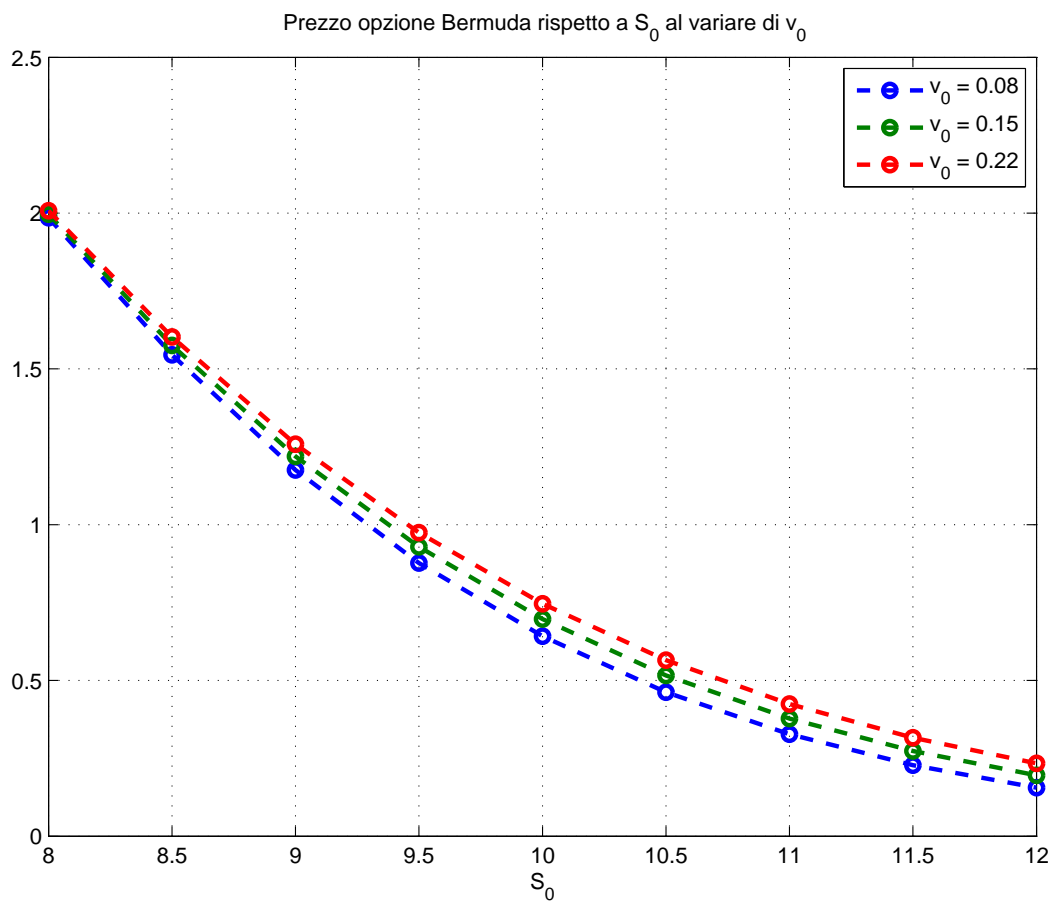


Figura 4.4: Prezzo opzione Bermuda per diversi livelli di volatilità iniziale

N/S_0	8
10	1.9820 (1.9820)
20	1.9934 (1.9905)
40	2.0054 (1.9949)

Tabella 4.1: Confronto dei risultati numerici per il valore di un'opzione Bermuda ottenuti mediante l'algoritmo FHT con quelli benchmark (mostrati in parentesi).

Capitolo 5

Modello dividend-ruin con dinamica time-changed Lévy

In questo capitolo consideriamo il problema classico in cui un'azienda deve erogare i dividendi in accordo con la strategia *dividend-barrier*, cioè il surplus di valore dell'azienda al di sopra di una certa barriera B viene pagato agli azionisti. Inoltre l'azienda si considera fallita quando il valore dei suoi asset scende al di sotto di una soglia L . Nel modello *dividend-ruin* il valore degli asset dell'azienda è un processo stocastico che è modellizzato come time-changed Lévy ristretto con una barriera assorbente inferiore e una barriera riflettente superiore. La barriera assorbente rappresenta il default dell'azienda quando il valore degli asset raggiungono la soglia di default L . La barriera riflettente modella, invece, la politica *dividend-barrier*. Il modello è a tempo finito e il monitoraggio delle due barriere è discreto. L'obiettivo è il pricing del modello dividend-ruin con dinamica time-changed Lévy utilizzando la trasformata veloce di Hilbert (FHT).

Per tutta la trattazione svolta in questo capitolo e per le dimostrazioni dei teoremi presentati si è fatto riferimento a [18].

Sia $t_0 = 0$ il tempo iniziale e $\mathcal{T} = \{t_0, t_1, \dots, t_N\}$, dove $t_N = T$ è la scadenza, l'insieme delle date di monitoraggio. Sebbene gli intervalli temporali tra due date successive di monitoraggio non siano in generale costanti, senza perdita di generalità, si può assumere un passo temporale Δ costante per semplificare il modello. Dunque si può scrivere $t_k = k\Delta$, con $k = 0, 1, \dots, N$.

5.1 Formulazione del modello

5.1.1 Le barriere assorbente e riflettente

Il modello descrive il logaritmo del valore degli asset dell'azienda non ristretto, A_t , come un processo time-changed Lévy. Sotto una misura neutrale al rischio \mathbb{Q} si

assume che A_t segua la seguente dinamica

$$A_t = e^{Y_t} = e^{rt} e^{X_{T_t}}, \quad (5.1)$$

dove X_{T_t} è il processo time-changed Lévy, inoltre si pone $Y_0 = \ln A_0$, con A_0 valore degli asset dell'azienda al tempo t_0 . La barriera assorbente L si può vedere come la barriera knock-out del processo ed è monitorata nei tempi di \mathcal{T} . D'altro canto l'azienda paga i dividendi agli azionisti ogni volta che il processo del valore degli asset supera la barriera B a una data di monitoraggio. In base a questa politica di dividendi, il valore dell'azienda ristretto non può mai oltrepassare la barriera riflettente B nelle date di monitoraggio. Dunque il processo ristretto presenta sia le caratteristiche di lookback discreta che quelle di barriera.

Siano \underline{A}_0^t e \bar{A}_0^t rispettivamente il minimo e il massimo valore realizzati dal processo nell'intervallo $[0, t]$ a monitoraggio discreto. Sia $j = \max\{i : t_i \leq t\}$, si ha

$$\underline{A}_0^t = \min\{A_{t_0}, A_{t_1}, \dots, A_{t_j}\} \quad \bar{A}_0^t = \max\{A_{t_0}, A_{t_1}, \dots, A_{t_j}\}. \quad (5.2)$$

Nel periodo finito $[0, t]$, condizionatamente alla sopravvivenza e all'adozione della strategia dividend-barrier, la frazione di asset dell'azienda rimanente è data dal $\min\left(1, \frac{B}{\bar{A}_0^t}\right)$. In accordo, è conveniente definire il valore degli asset dell'azienda non-ruined \hat{A}_t al tempo t come

$$\hat{A}_t = A_t \min\left(1, \frac{B}{\bar{A}_0^t}\right). \quad (5.3)$$

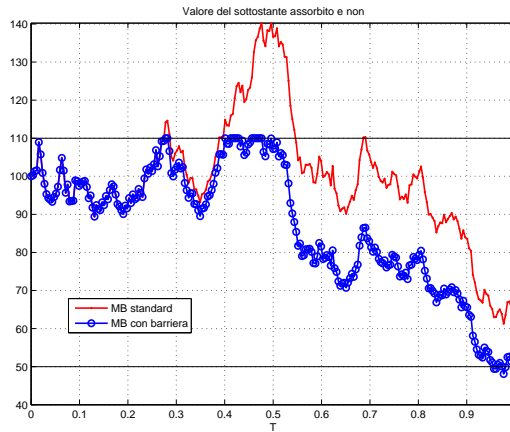


Figura 5.1: Simulazione del valore aziendale

Si noti che \bar{A}_0^t può presentare un salto in una data di monitoraggio quando il processo realizza un nuovo massimo, lo stesso accade per \hat{A}_t . Si definisce ora il valore minimo di \hat{A}_t sull'intervallo $[T_1, T_2]$ come

$$\hat{\underline{A}}_{T_1}^{T_2} = \min_{T_1 \leq t_j \leq T_2} (\hat{A}_{t_j}). \quad (5.4)$$

Si denoti ora con \tilde{A}_t il valore degli asset dell'azienda ristretta con le due barriere riflettente ed assorbente a monitoraggio discreto. Il valore a scadenza T è dato da

$$\tilde{A}_T = \hat{A}_T \mathbb{I}_{\{\hat{A}_0^T > L\}}, \quad (5.5)$$

dove l'evento $\{\hat{A}_0^T \leq L\}$ cattura il verificarsi del default quando il valore dell'azienda non-ruined \hat{A}_t cade al di sotto di L in un istante di monitoraggio nell'intervallo $[0, T]$.

5.1.2 La funzione valore del processo valore d'azienda

L'obiettivo è calcolare al tempo t la funzione valore del valore degli asset dell'azienda, con monitoraggio discreto, ristretto alle due barriere. Per fare il pricing si considera la dinamica time-changed Lévy, v_t denota il tasso di attività (business) del cambio di tempo stocastico che segue il processo CIR definito nella (1.35). Si procede con il cambio di variabile $\gamma_t = \ln v_t$, dove γ_t è il tasso della log-attività. Sia $V_{in}(A_t, \gamma_t, t; \hat{A}_0^T, \bar{A}_0^t)$ la funzione valore in-progress al tempo t che dipende dalle variabili di stato A_t e γ_t e dalle variabili di stato path dependent \hat{A}_0^T e \bar{A}_0^t . Applicando la valutazione neutrale al rischio, V_{in} è il valore atteso scontato del valore ristretto dell'azienda a scadenza T .

$$V_{in}(A_t, \gamma_t, t; \hat{A}_0^T, \bar{A}_0^t) = \mathbb{E}[e^{-r(T-t)} \tilde{A}_T]. \quad (5.6)$$

Si definisce poi $V_0(A_t, \gamma_t, t)$ come funzione valore dello stato iniziale corrispondente allo stato in cui A_t non ha mai raggiunto nè la barriera assorbente nè la barriera riflettente nelle date di monitoraggio $\mathcal{T}_t = \{t_0, \dots, t_j\}$, dove $j = \max\{i : t_i \leq t\}$. Si noti come la funzione valore dello stato iniziale $V_0(A_t, \gamma_t, t)$ non dipenda dalle variabili path-dependent \hat{A}_0^T e \bar{A}_0^t . In [13] gli autori hanno dimostrato la seguente relazione tra V_{in} e V_0 :

$$V_{in}(A_t, \gamma_t, t; \hat{A}_0^T, \bar{A}_0^t) = \begin{cases} V_0\left(\frac{B}{\bar{A}_0^t} A_t, \gamma_t, t\right) & \text{se } \{\hat{A}_0^T > L\} \text{ e } \bar{A}_0^t > B \\ V_0(A_t, \gamma_t, t) & \text{se } \{\hat{A}_0^T > L\} \text{ e } \bar{A}_0^t \leq B \\ 0 & \text{se } \{\hat{A}_0^T \leq L\} \end{cases} \quad (5.7)$$

(5.7) si può riscrivere come segue

$$V_{in}(A_t, \gamma_t, t; \hat{A}_0^T, \bar{A}_0^t) = V_0(\hat{A}_t, \gamma_t, t) \mathbb{I}_{\{\hat{A}_0^T > L\}}. \quad (5.8)$$

Tuttavia è più conveniente utilizzare $Z_t = \ln \hat{A}_t$ come variabile di stato e riscrivere $V(Z_t, \gamma_t, t) = V_0(\hat{A}_t, \gamma_t, t)$, ponendo $l = \ln L$ e $b = \ln B$. A questo punto si sviluppa l'algoritmo FHT per calcolare $V(Z_t, \gamma_t, t)$.

5.1.3 Condizione di salto della funzione valore

Siano t_k^- e t_k^+ gli istanti di tempo immediatamente prima e dopo una data di monitoraggio t_k con $k = 1, 2, \dots, N$. Siccome non viene dato nessun cash flow agli azionisti in una data di monitoraggio, la funzione valore appena prima e appena dopo di essa deve rimanere la stessa. Supponiamo che $Z_{t_k^-} > b$, allora $Z_{t_k^+}$ sarà posta uguale a b . Questa condizione di salto in t_k modella la frazione del valore di azienda che viene persa a causa del payout dei dividendi e il valore degli asset si riduce, di conseguenza, a B , in accordo con la politica dividend-barrier. D'altro canto, Z_t rimane continua in t_k quando $Z_{t_k^-} \leq b$. Il default è modellizzato, invece, da $\mathbb{I}_{\{l < Z_{t_k^-} < b\}}$ in modo tale che la funzione valore diventi zero quando $Z_{t_k^-} < l$. Dunque la condizione di salto della funzione valore diventa

$$V(Z_{t_k^-}, \gamma_{t_k}, t_k^-) = V(Z_{t_k^+}, \gamma_{t_k}, t_k^+) \mathbb{I}_{\{l < Z_{t_k^-} < b\}} + V(b, \gamma_{t_k}, t_k^+) \mathbb{I}_{\{Z_{t_k^-} > b\}}. \quad (5.9)$$

Il calcolo a ritroso si inizia nell'istante appena prima della scadenza t_N e la condizione finale della funzione valore è

$$V(Z_{t_N^-}, \gamma_{t_N}, t_N^-) = e^{Z_{t_N^-}} \mathbb{I}_{\{l < Z_{t_N^-} < b\}} + B \mathbb{I}_{\{Z_{t_N^-} > b\}}. \quad (5.10)$$

5.1.4 Il calcolo tra due date successive di monitoraggio

Partendo dalla formula di pricing neutrale rispetto al rischio

$$V(Z_{t_k^+}, \gamma_{t_k}, t_k^+) = e^{-r\Delta} \mathbb{E}[V(Z_{t_{k+1}^-}, \gamma_{t_{k+1}}, t_{k+1}^-) | \mathcal{F}_{t_k^+}], \quad (5.11)$$

dove $\Delta = t_{k+1} - t_k$ e applicando la *tower property*, condizionando rispetto al tasso della log-attività $\gamma_{t_{k+1}}$ al tempo t_{k+1} , si ottiene

$$V(Z_{t_k^+}, \gamma_{t_k}, t_k^+) = e^{-r\Delta} \mathbb{E} \left[\mathbb{E}[V(Z_{t_{k+1}^-}, \gamma_{t_{k+1}}, t_{k+1}^-) | \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}}] | \mathcal{F}_{t_k^+} \right]. \quad (5.12)$$

Il calcolo della funzione valore tra due successive date di monitoraggio può essere espresso, quindi, come doppio valore atteso. Il valore atteso esterno comporta un'integrazione sulla densità $p_\gamma(\gamma_{t_{k+1}} | \gamma_{t_k})$ che è nota in forma chiusa, (1.39), se il processo del tasso di attività v_t segue la dinamica CIR. In sintesi, per calcolare il doppio integrale si applica un'interpolazione basata sulla regola di quadratura per la valutazione numerica dell'integrale esterno e il metodo della trasformata di Fourier per il valore atteso interno. A cavallo di una data di monitoraggio il metodo della trasformata di Hilbert, data la presenza della barriera nel payoff, diventa di grande vantaggio.

Formula di quadratura numerica

Come primo passo si applica una formula di quadratura con J punti (come la formula di Gauss-Legendre o composita del trapezio) per calcolare il valore atteso esterno. Discretizzando lungo la dimensione di $\gamma_{t_{k+1}}$ nei nodi ζ_j , con $j = 1, 2, \dots, J$ si ha

$$V(Z_{t_k^+}, \gamma_{t_k}, t_k^+) = e^{-r\Delta} \sum_{j=1}^J \omega_j p_\gamma(\zeta_j | \gamma_{t_k}) \mathbb{E} \left[V(Z_{t_{k+1}^-}, \gamma_{t_{k+1}}, t_{k+1}^-) \middle| \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j \right], \quad (5.13)$$

dove gli ω_j sono i pesi di quadratura nei nodi ζ_j , con $j = 1, 2, \dots, J$.

Il metodo CONV

Al fine di calcolare il valore atteso interno si adotta il metodo *CONV* che è comunemente usato negli algoritmi di pricing delle opzioni basati su trasformata di Fourier. Tipicamente in questi algoritmi, per garantire che la trasformata di Fourier sia ben definita, è necessario introdurre un fattore esponenziale di smorzamento. Sia $\omega = \alpha + i\beta$, dove α è costante, fissati $\gamma_{t_{k+1}} = \zeta_j$ e $Z_{t_{k+1}^-} = x$, si definisce $V_\alpha(x, \zeta_j, t_{k+1}^-) = e^{\alpha x} V(x, \zeta_j, t_{k+1}^-)$, dove $e^{\alpha x}$ è il fattore di smorzamento della funzione valore $V(x, \zeta_j, t_{k+1}^-)$ nei nodi ζ_j . Il parametro α è scelto per assicurare l'esistenza della trasformata generalizzata di Fourier di $V(x, \zeta_j, t_{k+1}^-)$ che è

$$\hat{V}_\alpha(\beta, \zeta_j, t_{k+1}^-) = \int_{-\infty}^{\infty} e^{\omega x} V(x, \zeta_j, t_{k+1}^-) dx = \int_{-\infty}^{\infty} e^{i\beta x} V_\alpha(x, \zeta_j, t_{k+1}^-) dx. \quad (5.14)$$

Grazie alla relazione di Parseval nella trasformata di Fourier si riscrive l'integrale interno come

$$\begin{aligned} & \mathbb{E} \left[V(Z_{t_{k+1}^-}, \gamma_{t_{k+1}}, t_{k+1}^-) \middle| \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j \right] \\ &= \int_{-\infty}^{\infty} V(x, \zeta_j, t_{k+1}^-) p(x | \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j) dx \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}^-) \check{p}(\omega | \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j) d\beta. \end{aligned} \quad (5.15)$$

dove $\check{p}(\omega | \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j) = \mathbb{E}[e^{-\omega x} | \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j]$ è la trasformata generalizzata inversa di Fourier della densità condizionata $p(x | \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j)$ di $Z_{t_{k+1}^-}$. Nel periodo (t_k^+, t_{k+1}^-) non c'è payout del dividendo e non avviene il default quindi si ha che $Z_{t_{k+1}^-} - Z_{t_k^+} = Y_{t_{k+1}} - Y_{t_k}$. Possiamo riscrivere ora $\check{p}(\omega | \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j)$ nella seguente rappresentazione

$$\check{p}(\omega | \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j) = e^{-\omega Z_{t_k^+}} \mathbb{E} \left[e^{-\omega(Y_{t_{k+1}} - Y_{t_k})} \middle| \mathcal{F}_{t_k^+}, \gamma_{t_{k+1}} = \zeta_j \right]. \quad (5.16)$$

Posto $\Psi(\omega; \gamma_t, \gamma_s) = \mathbb{E} \left[e^{\omega(Y_t - Y_s)} \middle| \mathcal{F}_s, \gamma_t \right]$, si cerca di legare $\Psi(\omega; \gamma_t, \gamma_s)$ con la funzione caratteristica $\Phi_X(\cdot)$ di X_t .

$$\begin{aligned} \Psi(\omega; \gamma_t, \gamma_s) &= \mathbb{E} \left[\mathbb{E} [e^{\omega(Y_t - Y_s)} | \mathcal{F}_s, \gamma_t, T_t - T_s] \middle| \mathcal{F}_s, \gamma_t \right] \\ &= \mathbb{E} \left[\mathbb{E} [e^{\omega(Y_t - Y_s)} | T_t - T_s] \middle| \mathcal{F}_s, \gamma_t \right] \\ &= e^{\omega r(t-s)} \mathbb{E} [e^{-\psi_X(-i\omega)(T_t - T_s)} | \mathcal{F}_s, \gamma_t] \\ &= e^{\omega r(t-s)} \Phi(i\psi_X(-i\omega); \gamma_t \gamma_s). \end{aligned} \quad (5.17)$$

Combinando (5.15), (5.16) e (5.17) si può esprimere il valore atteso interno nel punto $\gamma_{t_{k+1}} = \zeta_j$ come la trasformata di Fourier inversa:

$$\mathbb{E} \left[V(Z_{t_{k+1}}^-, \gamma_{t_{k+1}}, t_{k+1}^-) \middle| \mathcal{F}_{t_k}^+, \gamma_{t_{k+1}} \right] = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-\omega Z_{t_k}^+} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}^-) \Psi(-\omega; \zeta_j, \gamma_{t_k}) d\beta, \quad (5.18)$$

dove si è posto $\gamma_{t_{k+1}} = \zeta_j$ nella funzione generatrice dei momenti condizionata $\Psi(\omega; \gamma_{t_{k+1}}, \gamma_{t_k})$.

La procedura di calcolo

Date $N + 1$ date di monitoraggio, dove $\mathcal{T} = \{t_k | k = 0, 1, \dots, N\}$ con $t_N = T$, i passi associati con la valutazione numerica del doppio valore atteso sono i seguenti:

1. La procedura di induzione backward parte con la condizione finale specificata nella (5.10)
2. Il calcolo tra due date di monitoraggio si basa sulla condizione di salto (5.9), sulla formula di quadratura (5.13) e sull'algoritmo CONV (5.18). Per $k = N - 1, N - 2, \dots, 1$, l'approssimazione numerica di $V(Z_{t_k}^-, \gamma_{t_k}, t_k^-)$ è data da

$$\begin{aligned} V(Z_{t_k}^-, \gamma_{t_k}, t_k^-) &\approx \\ &\approx \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j p_\gamma(\zeta_j | \gamma_{t_k}) [\mathbb{I}_{\{l < Z_{t_k}^- < b\}} \int_{-\infty}^{\infty} e^{-\omega Z_{t_k}^+} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}^-) \Psi(-\omega; \zeta_j, \gamma_{t_k}) d\beta \\ &\quad + \mathbb{I}_{\{Z_{t_k}^- \geq b\}} \int_{-\infty}^{\infty} e^{-\omega b} \hat{V}_\alpha(\beta, \zeta_j, t_{k+1}^-) \Psi(-\omega; \zeta_j, \gamma_{t_k}) d\beta]. \end{aligned} \quad (5.19)$$

Siccome si assume $Z_{t_0} \in (l, b]$, lo schema di approssimazione numerica per $k = 0$ diventa

$$V(Z_{t_0}, \gamma_{t_0}, t_0) \approx \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j p_\gamma(\zeta_j | \gamma_{t_0}) \int_{-\infty}^{\infty} e^{-\omega Z_{t_0}} \hat{V}_\alpha(\beta, \zeta_j, t_1^-) \Psi(-\omega; \zeta_j, \gamma_{t_0}) d\beta. \quad (5.20)$$

La valutazione degli integrali (5.19) e (5.20) si fa mediante il metodo FFT. Nell'usuale procedura FFT il check della condizione di knock-out e la politica dividend-barrier

deve essere effettuato nel dominio reale. Dunque, sarebbe necessario antitrasformare ad ogni passo temporale nell' induzione a ritroso per recuperare la funzione valore nel dominio reale. Si rivela perciò vantaggioso l' uso della trasformata veloce di Hilbert, FHT, che evita il recupero della funzione valore ad ogni passo temporale. Come già più volte spiegato, il guadagno in termini di efficienza computazionale è enorme perchè la barriera è naturalmente incorporata nella procedura FHT.

5.2 Costruzione dell' algoritmo FHT

Come già detto, l' ingrediente principale del metodo FHT è che la moltiplicazione di una funzione per una funzione indicatrice, associata con la caratteristica di barriera, nel dominio reale corrisponde a eseguire la trasformata di Hilbert nel dominio di Fourier. Di conseguenza, invece di calcolare $N - 1$ volte la trasformata di Fourier e la sua inversa, come mostrato in (5.19), il metodo FHT richiede solamente $N - 1$ passi di computazione.

La procedura di induzione backward nel dominio di Fourier usando l' algoritmo trasformata di Hilbert viene riformulata come segue:

1. Per $k = N$, si ha la stessa condizione finale specificata in (5.10). Si calcola inoltre la trasformata di Fourier generalizzata $\hat{V}_\alpha(\beta, \zeta_j, t_N^-)$, dove si pone $\gamma_{t_N} = \zeta_j, j = 1, 2, \dots, N$.
2. Per $k = N - 1, N - 2, \dots, 1$, si ha la seguente approssimazione numerica

$$\begin{aligned} \hat{V}_\alpha(\beta, \zeta_p, t_k^-) &\approx \\ &\approx e^{-r\Delta} \int_{-\infty}^{\infty} \sum_{j=1}^J \omega_j \hat{V}_\alpha(\eta, \zeta_j, t_{k+1}^-) \tilde{\Psi}(-\alpha - i\eta; \zeta_j, \zeta_p) e^{i \frac{(\beta-\eta)(t+b)}{2}} \frac{\sin \frac{(\beta-\eta)(b-t)}{2}}{\pi(\beta-\eta)} d\eta \\ &\quad - \frac{e^{-r\Delta} e^{i\beta b}}{2\pi(\alpha + i\beta)} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-i\eta b} \hat{V}_\alpha(\eta, \zeta_j, t_{k+1}^-) \tilde{\Psi}(-\alpha - i\eta; \zeta_j, \zeta_p) d\eta, \end{aligned} \quad (5.21)$$

per $p = 1, 2, \dots, J$. Nella funzione generatrice dei momenti condizionata, $\tilde{\Psi}(\omega; \gamma_{t_k}, \gamma_{t_{k+1}})$, si è posto $\gamma_{t_{k+1}} = \zeta_j$ e $\gamma_{t_k} = \zeta_p$ e si è riscritto $\tilde{\Psi}(\omega; \zeta_j, \zeta_p)$ come $\Psi(\omega; \zeta_j, \zeta_p) p_\gamma(\zeta_j, \zeta_p)$.

3. Per $k = 0$, l' approssimazione numerica della funzione valore al tempo t_0 è ottenuta come

$$V(Z_{t_0}, \zeta_p, t_0) \approx \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \int_{-\infty}^{\infty} e^{-(\alpha+i\beta)Z_{t_0}} \hat{V}_\alpha(\beta, \zeta_j, t_1^-) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \zeta_p) d\beta. \quad (5.22)$$

L' algoritmo FHT deriva da (5.9) e (5.10) mediante applicazione della trasformata generalizzata di Fourier a entrambi i membri di (5.19), che incorpora le funzioni indicatrici che modellizzano le caratteristiche dividend-ruin. Prima di tutto, è necessario moltiplicare entrambi i lati della (5.19) per il fattore di smorzamento

esponenziale $e^{\alpha Z_{t_k}^-}$, in modo tale che le trasformate di Fourier siano ben definite. Dal momento che sia la trasformata di Fourier inversa che la funzione indicatrice $\mathbb{I}_{\{l < Z_{t_k}^- < b\}}$ includono la variabile di stato $Z_{t_k}^-$ per il primo termine a destra si fa uso della trasformata di Hilbert al fine di trattare con la trasformata di Fourier del prodotto di una funzione per una funzione indicatrice. A causa della non esistenza della trasformata di Fourier della variabile di stato $Z_{t_k}^-$ del secondo termine, si calcola la sua usuale trasformata di Fourier.

Il passo seguente prevede l'utilizzo di una formula di quadratura, lungo la dimensione del tasso della log-attività, con nodi $\gamma_{t_{k+1}} = \zeta_j$ dove $j = 1, 2, \dots, N$. Ci si serve dello stesso insieme di nodi per tutti i passi temporali.

Nell'ultimo step si calcola $V(Z_{t_0}, \zeta_p, t_0)$ mediante la formula della trasformata inversa di Fourier.

La funzione $\Psi(\omega; \zeta_j, \zeta_p)$ è l'input che caratterizza il processo time-changed Lévy. Combinando (1.39), (1.40) e (5.17), la funzione di Bessel presente nel termine $p_\gamma(\zeta_j | \zeta_p)$ si cancella con quella al denominatore del termine $\Psi(\omega; \zeta_j, \zeta_p)$, lasciando solamente quella di $\tilde{\Psi}(\omega; \zeta_j, \zeta_p)$. Ciò comporta il calcolo della funzione modificata di Bessel del primo tipo che cresce considerevolmente quando $q \rightarrow -1$ o $\omega \rightarrow \infty$.

Per avere un minore costo computazionale si può implementare la funzione di Bessel riscalata. In secondo luogo, alcuni termini presenti in $\tilde{\Psi}(\omega; \zeta_j, \zeta_p)$ potrebbero diventare molto grandi quando un certo insieme di parametri viene scelto, quindi si deve cercare di semplificare queste moltiplicazioni nel calcolo del valore numerico di $\tilde{\Psi}(\omega; \zeta_j, \zeta_p)$.

Nell'ultimo step si usa, invece, un'interpolazione spline per ottenere il valore $V(Z_{t_0}, \gamma_{t_0}, t_0)$ da $V(Z_{t_0}, \zeta_p, t_0)$, oppure si sceglie un layout di griglia tale che γ_{t_0} stia esattamente sulla griglia.

Approssimazione discreta

Si consideri la trasformata inversa di Fourier di g

$$Pg(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\beta x} g(\beta, \zeta_j) \tilde{\Psi}(-\alpha - i\beta; \zeta_j, \zeta_p) d\beta, \quad (5.23)$$

e si valuti l'integrale di convoluzione

$$Qg(\beta) = \int_{-\infty}^{\infty} g(\eta, \zeta_j) \tilde{\Psi}(-\alpha - i\eta; \zeta_j, \zeta_p) e^{i\frac{(\beta-\eta)(t+b)}{2}} \frac{\sin \frac{(\beta-\eta)(b-l)}{2}}{\pi(\beta-\eta)} d\eta. \quad (5.24)$$

La trasformata inversa di Fourier può essere valutata numericamente come

$$P_{h,M}g(x) = \frac{1}{2\pi} \sum_{m=-M}^M e^{-ixmh} g(mh, \zeta_j) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) h, \quad (5.25)$$

mentre la valutazione numerica di $Qg(\beta)$ può essere effettuata con il seguente schema di discretizzazione con $\beta = nh$

$$Q_{h,M}g(nh) = \sum_{m=-M}^M g(mh, \zeta_j) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) e^{i \frac{h(n-m)(l+b)}{2}} \frac{\sin \frac{h(n-m)(l+b)}{2}}{\pi h(n-m)} h, \quad (5.26)$$

dove $n = -M, \dots, M$, h è il passo e il dominio infinito viene troncato al dominio finito $[-Mh, Mh]$, in cui M si riferisce al livello di troncamento. La più semplice approssimazione trapezoidale del termine somma raggiunge una buona accuratezza con errore di discretizzazione che decade esponenzialmente.

Anche in questo caso compare una moltiplicazione matrice-vettore che coinvolge la matrice di Toeplitz T , in particolare quella già vista per il caso di barriera doppia:

$$T_{n,m} = \begin{cases} \frac{\sin \frac{h(n-m) \ln \frac{B}{L}}{2}}{\pi(n-m)} & \text{se } m \neq n \\ \frac{h \ln \frac{B}{L}}{2\pi} & \text{se } m = n \end{cases} \quad (5.27)$$

5.2.1 Implementazione

Prima di iniziare, è necessario scegliere un appropriato α al fine di garantire l'esistenza della trasformata di Hilbert generalizzata. Si noti che $V(Z_{t_N^-}, \gamma_{t_N^-}, t_N^-) = e^{Z_{t_N^-}} \mathbb{I}\{l < Z_{t_N^-} \leq b\} + B \mathbb{I}\{Z_{t_N^-} > b\} \in L^1(\mathbb{R})$ per ogni $\alpha < 0$. Ricordando anche che $\alpha \in \mathcal{L}_X$, si ha $\alpha \in (\lambda_-, 0)$.

Occorre calcolare la trasformata generalizzata di Fourier del payoff a scadenza usando la formula analitica

$$\hat{V}_\alpha(\beta, \zeta_j, t_N^-) = \frac{B e^{(\alpha+i\beta)b} - L e^{(\alpha+i\beta)l}}{\alpha + i\beta + 1} - \frac{B e^{(\alpha+i\beta)b}}{\alpha + i\beta}, \quad \beta = -Mh, \dots, Mh. \quad (5.28)$$

Preparare la matrice $\tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p)$ per $p = 1, 2, \dots, J$.

Basandosi sulle equazioni (5.21) e (5.26) e scambiando l'ordine delle sommatorie, si calcola $\hat{V}_\alpha(\beta, \zeta_p, t_k^-)$ come segue

$$\begin{aligned} \hat{V}_\alpha(\beta, \zeta_p, t_k^-) &\approx \\ &\approx e^{-r\Delta} e^{\frac{i\beta(l+b)}{2}} \sum_{m=-M}^M e^{\frac{-imh(l+b)}{2}} \sum_{j=1}^J \omega_j \hat{V}_\alpha(mh, \zeta_j, t_{k+1}^-) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) \frac{\sin \frac{(\beta-mh)(b-l)}{2}}{\pi(\beta-mh)} h \\ &\quad - \frac{e^{-r\Delta} e^{i\beta b}}{2\pi(\alpha+i\beta)} \sum_{m=-M}^M e^{-imhb} \sum_{j=1}^J \omega_j \hat{V}_\alpha(mh, \zeta_j, t_{k+1}^-) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) h, \end{aligned} \quad (5.29)$$

per $\beta = -Mh, \dots, Mh$. Si ripete poi *Step2* per $k = N - 1, \dots, 1$.

Si approssima la trasformata di Fourier in (5.22) con una formula di quadratura per trovare la funzione valore al tempo t_0 come segue

$$V(Z_{t_0}, \zeta_p, t_0) \approx \frac{e^{-r\Delta}}{2\pi} \sum_{j=1}^J \omega_j \sum_{m=-M}^M e^{-(\alpha+imh)Z_{t_0}} \hat{V}_\alpha(mh, \zeta_j, t_1^-) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) h. \quad (5.30)$$

Infine si utilizza un'interpolazione spline per ottenere $V(Z_{t_0}, \gamma_{t_0}, t_0)$ dalla griglia di funzioni valore $V(Z_{t_0}, \zeta_p, t_0)$ per $p = 1, 2, \dots, J$.

Osservazioni

1. Siccome il valore iniziale degli asset dell'azienda A_0 è sostituito nella formula di pricing solamente nello step finale dell'algoritmo, l'algoritmo FHT può essere utilizzato per trovare la funzione valore per più valori di A_0 simultaneamente, senza ulteriori costi computazionali.
2. Come gli altri metodi di pricing di Fourier, l'algoritmo FHT può calcolare la delta e la gamma della funzione valore con facilità.

Si scrive ora $\hat{V}_\alpha(nh, \zeta_p, t_k^-)$ come la funzione $\hat{V}_\alpha(\beta, \zeta_p, t_k^-)$ valutata in $\beta = nh$. Al fine di maneggiare con le barriere lower e upper, la (5.29) diventa

$$\begin{aligned} \hat{V}_\alpha(nh, \zeta_p, t_k^-) &\approx \\ &\approx e^{-r\Delta} e^{\frac{inh(t+b)}{2}} \sum_{\substack{m=-M \\ m \neq n}}^M e^{-\frac{imh(t+b)}{2}} \sum_{j=1}^J \omega_j \hat{V}_\alpha(mh, \zeta_j, t_{k+1}^-) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) \frac{\sin \frac{h(n-m) \ln \frac{B}{L}}{2}}{\pi(n-m)} \\ &+ \sum_{j=1}^J \omega_j \hat{V}_\alpha(nh, \zeta_j, t_{k+1}^-) \tilde{\Psi}(-\alpha - in h; \zeta_j, \zeta_p) \frac{h \ln \frac{B}{L}}{2\pi} \\ &- \frac{e^{-r\Delta} e^{inhb}}{2\pi(\alpha + in h)} \sum_{m=-M}^M e^{-imhb} \sum_{j=1}^J \omega_j \hat{V}_\alpha(mh, \zeta_j, t_{k+1}^-) \tilde{\Psi}(-\alpha - imh; \zeta_j, \zeta_p) h, \end{aligned} \quad (5.31)$$

con $n = -M, \dots, M$. Siccome (5.31) ad ogni passo temporale utilizza una moltiplicazione matrice-vettore di Toeplitz, l'induzione backward nel dominio di Fourier dello *Step2* può essere riformulata mediante una notazione più semplice matrice-vettore. Introduciamo la seguente notazione: $\hat{V}_\alpha(t_{k+1})$ è una matrice di dimensioni $(2M+1) \times J$ con elementi $\hat{V}_\alpha(mh, \zeta_j, t_{k+1})$, $\tilde{\Psi}(\zeta_p)$ è una matrice di dimensioni $(2M+1) \times J$ con elementi $\tilde{\Psi}(-\alpha - imh, \zeta_j, \zeta_p)$ per $p = 1, 2, \dots, J$, \mathbf{w} è un vettore colonna contenente i pesi di quadratura. Si costruisce quindi la moltiplicazione matrice-vettore

$$\begin{aligned} \mathbf{k}_p(t_k) &= [\hat{V}_\alpha(t_{k+1}) \cdot \tilde{\Psi}(\zeta_p)] \mathbf{w}, \quad p = 1, 2, \dots, J; \\ B(t_k) &= [\mathbf{k}_1(t_k), \mathbf{k}_2(t_k), \dots, \mathbf{k}_J(t_k)], \end{aligned}$$

dove $\mathbf{k}_p(t_k)$ è un vettore colonna di dimensioni $2M+1$, $B(t_k)$ è una matrice $(2M+1) \times J$ e l'operatore \cdot denota il prodotto elemento per elemento. Siano H ed

F matrici $(2M + 1) \times J$ i cui elementi sono $H_{m,p} = e^{-\frac{imh(t+b)}{2}}$ ed $F_{m,p} = e^{\frac{imh(t+b)}{2}}$, siano $\boldsymbol{\xi}$ ed \mathbf{s} vettori colonna composti con elementi $\boldsymbol{\xi}_m = e^{-imhb}h$ e $\mathbf{s}_m = \frac{e^{imhb}}{\alpha + imh}$. In accordo con (5.31), possiamo riscrivere l'induzione backward nel dominio di Fourier nella seguente forma matrice-vettore

$$\hat{V}_\alpha(t_k) = e^{-r\Delta} [T(B(t_k) \cdot H)] \cdot F - \frac{e^{-r\Delta}}{2\pi} \mathbf{s} [\boldsymbol{\xi}^T B(t_k)], \quad (5.32)$$

dove T è la matrice di Toeplitz definita in (5.27). Il vantaggio nella moltiplicazione matrice-vettore è enorme con complessità $O(M \log_2 M)$ nella dimensione log asset return. In aggiunta, la stessa matrice di Toeplitz è usata ad ogni step e solamente due runs di FFT sono richieste ad ogni passo temporale. Infatti è necessario aggiornare ad ogni step solamente $B(t_k)$ per ottenere $\hat{V}_\alpha(t_k)$.

La versione dell'algoritmo FHT è riassunta nella tabella seguente.

5.2.2 Sintesi dell'algoritmo numerico di pricing

Step1: Preparazione

Calcolare $\hat{V}_\alpha(t_N)$ usando la formula analitica (5.28);
 Preparare la matrice $\tilde{\Psi}(\zeta_p)$ per $p = 1, 2, \dots, J$;
 Preparare le matrici H , F e i vettori $\boldsymbol{\xi}$, \mathbf{s} ;
 Calcolare la prima riga e colonna della matrice di Toeplitz.

Step2: Induzione backward

Calcolare $\mathbf{k}_p(t_k) = [\hat{V}_\alpha(t_{k+1}) \cdot \tilde{\Psi}(\zeta_p)] \mathbf{w}$, per $p = 1, 2, \dots, J$ e aggiornare $B(t_k)$;
 Calcolare $\hat{V}_\alpha(t_k)$ tramite (5.32) usando l'algoritmo FFT per $k = N - 1, \dots, 1$.

Step3: Valore degli asset dell'azienda in t_0

Calcolare il valore degli asset dell'azienda tramite (5.30). Utilizzare un'interpolazione spline per ottenere $V(Z_{t_0}, \gamma_{t_0}, t_0)$.

5.2.3 Complessità computazionale

Si adotta la formula di quadratura dei trapezi nella dimensione del tasso della log-attività. Lo sforzo computazionale nella preparazione con nodi non equidistanti è dominata dal calcolo delle matrici $\tilde{\Psi}(\zeta_p)$ per $p = 1, 2, \dots, J$, che comportano la valutazione numerica di funzioni modificate di Bessel. Si ricordi che il calcolo di una funzione modificata di Bessel costa significativamente di più di una semplice moltiplicazione. Più precisamente, si supponga che il calcolo di una funzione di Bessel costi \mathcal{A} volte il numero di operazioni necessarie per una moltiplicazione, le

matrici $\tilde{\Psi}(\zeta_p)$ per $p = 1, 2, \dots, J$ richiederebbero $O(AMJ^2)$ operazioni per il calcolo di tutti gli elementi.

Per quanto riguarda il calcolo nel loop principale dell'algoritmo è dominato dal termine $[T(B(t_k) \cdot H)] \cdot F$. Siccome il calcolo del vettore $\mathbf{k}_p(t_k)$ costa $O(MJ)$ operazioni, ciò porta a una complessità di $O(MJ^2)$ per il calcolo della matrice $B(t_k)$. Il prodotto $\mathbf{s}[\boldsymbol{\xi}^T B(t_k)]$ costa $O(M \log_2 MJ^2)$, grazie all'utilizzo della matrice di Toeplitz T e all'algoritmo FHT. La complessità è inferiore se paragonata a quella del calcolo diretto di $O(M^2 J^2)$. Come risultato, la complessità totale di FHT è $O(\max[\mathcal{A}, N \log_2 M] MJ^2)$. Riassumendo, l'essenza di FHT è che il calcolo numerico rimane nel dominio di Fourier ed è richiesta solamente un'inversione di Fourier nello step finale. FFT richiede invece il doppio di FHT. Questo deriva dal fatto che è necessario trovare la trasformata di Fourier e la sua inversa ad ogni istante temporale dell'induzione a ritroso. Inoltre, è ben noto che FFT deve rispettare una restrizione sullo step, dove $\Delta_x \Delta_\beta = \frac{2\pi}{M}$. In aggiunta, la formula di quadratura trapezoidale usata nel calcolo della trasformata di Fourier del valore dell'opzione è accurata al secondo ordine nella discretizzazione nella dimensione del log-asset return. In [9] hanno mostrato che se si adotta FHT, la formula di quadratura trapezoidale usata nel calcolo della trasformata di Fourier dell'inversa (5.23) e nell'integrale di convoluzione (5.26) ha un errore che decresce esponenzialmente grazie all'analiticità dell'integranda in un appropriato strip del piano complesso.

5.2.4 Test numerici

In questa sezione si utilizza l'algoritmo FHT per il pricing del modello dividend-ruin sotto l'ipotesi di processo time-changed Lévy: il modello di Heston a volatilità stocastica.

Si presentano qui i risultati numerici per trovare la funzione valore degli asset dell'azienda con modello dividend-ruin dove il processo valore dell'azienda segue un processo time-changed Lévy. I parametri utilizzati nelle simulazioni sono $A_0 = 10$, $T = 1$, $B = 13$, $L = 7$, $r = 0.04$ e $N = 10$. La dinamica Heston del sottostante ha parametri $\lambda = 1.15$, $\eta = 0.39$, $\bar{v} = 0.0348$, $v_0 = 0.0348$ e $\rho = -0.64$, mentre il livello di troncamento per la discretizzazione della trasformata di Hilbert è $M = 2^6$.

Ricordando che $\nu = \frac{2\lambda\bar{v}}{\eta^2} - 1$ e che la valenza della condizione di Feller è equivalente a $\nu \geq 0$, con i parametri precedenti si ottiene $\nu = -0.47 < 0$ e dunque la condizione di Feller non è soddisfatta. In questo caso, quindi, la coda di sinistra della densità del tasso di attività cresce estremamente veloce in valore. Grazie alla trasformazione dal dominio del tasso di attività al dominio del tasso di attività logaritmica, si riescono ad ottenere dei risultati numerici molto accurati usando l'algoritmo della trasformata veloce di Hilbert.

In Figura 5.2 è riportato il grafico della valutazione dell'azienda al variare della volatilità iniziale. Come ci si aspetta il prezzo si riduce all'aumentare della volatilità in quanto volatilità elevata comporta una maggiore probabilità di fallimento, ma non è bilanciata allo stesso tempo da possibili innalzamenti del valore della società poichè vi è un upperbound riflettente.

Il prezzo trovato con i precedenti parametri è di 9.5313 mediante l'algoritmo FHT e di 9.5335 con un intervallo di confidenza al 95% di ampiezza 0.0584 mediante il metodo Monte Carlo, che può essere considerato il benchmark, con 10^4 simulazioni.

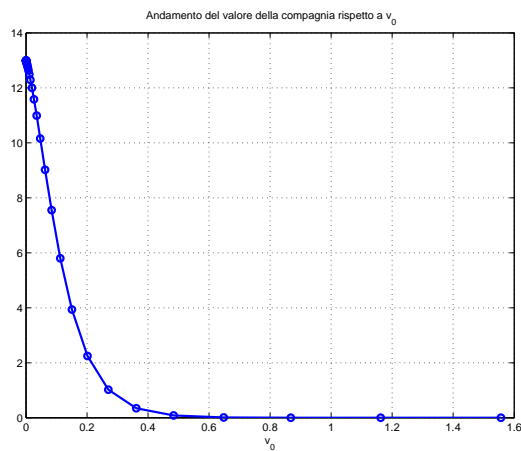


Figura 5.2: Andamento del valore aziendale rispetto alla volatilità iniziale

Capitolo 6

Conclusioni

Si è sviluppato e applicato l'algoritmo FHT per il pricing di diversi tipi di contratti derivati, sia Europei che path dependent. Questo metodo si è rivelato efficiente e robusto, capace di assecondare numerosi differenti modelli per la dinamica del sottostante. Dapprima si sono considerati i processi di Lévy, poi il modello a volatilità stocastica di Heston, incasellato nella classe dei processi time-changed Lévy. Quest'ultima è una generalizzazione non banale dell'algoritmo FHT, precedentemente sviluppato, per il pricing sotto i processi di Lévy unidimensionali.

Si è anche considerato il modello dividend-ruin con default e barriera per elargire il dividendo, come rappresentante della classe dei problemi sul rischio di credito, a testimoniare la flessibilità e la vastità delle possibili applicazioni del metodo. Il modello dividend-ruin a tempo finito ricorda un'opzione path dependent con caratteristiche sia di lookback che di barriera, il problema di pricing è dunque considerato più difficile rispetto ad una semplice opzione barriera. Si osserva un'elevata accuratezza numerica anche nello scenario dove la condizione di Feller nel processo CIR del tasso di attività non è soddisfatta.

In conclusione il metodo FHT è risultato molto adatto alla risoluzione dei problemi di pricing, non solo in ambito equity ma anche in caso di applicazioni creditizie.

Appendice A

Codici

In questa appendice vengono riportati i principali codici utilizzati per la realizzazione della tesi.

A.1 Metodi di Pricing per opzioni Europee basati su FHT

A.1.1 PricingEuropean_HT

Questo script consente di prezzare un'opzione Europea Call e Put mediante la tecnica della trasformata di Hilbert.

```
%% Pricing Opzione Europea
```

```
function PrezzoFFT = PricingEuropean_HT(S0, T, K, r, q, alpha,
    modello, params, M)
```

```
% ** Opzione Considerata
```

```
% Parametri opzione
```

```
Delta = T/1; % Passo temporale Europea
```

```
x = log(S0/K); % Trasformazione log-strike
```

```
% ** Modello Considerato
```

```
Frammento_LetturaParametri;
```

```
% ** Ulteriori Parametri
```

```
% Troncamento
```

```
dd = 20;
```

```
nu = 2;
```

```
h = ( (pi*dd)/(T*c) )^(1/(1+nu)) * ...
```

```
    M^(-nu/(1+nu)); % Passo griglia
```

```
% Griglia
```

```
k = (-M:M-1)'; % Indici della griglia spazio Fourier
```

```
MM = length(k); % Numero elementi griglia Fourier
```

```
xi = k*h; % Griglia spazio Fourier
```

```
% xi(M+1) = 1e-20; % Per non avere errori numerici rimuovo lo zero
```

```

% ** Inizializzazione del ciclo (per le europee non serve)
f_capp_alpha_EURO = -K ./ ( (xi - 1i*alpha) .* (xi - 1i*(alpha+1)) );
g = CharFunc(-xi + 1i*alpha) / CharFunc(1i*alpha) ...
    .* f_capp_alpha_EURO;

% ** Inversione finale mediante FFT
w = ones(size(xi)); w(1) = 0.5; w(end) = 0.5; % pesi

j = (0:2*M-1)'; % Indici nello spazio reale
k = j; % Traslo indici spazio Fourier
eta = 2*pi / (h*M); % Passo griglia spazio reale
xx = -M*eta+j*eta; % Punti griglia spazio reale

corr = exp( 2*M/M*M*1i*pi*k ); % Correzione da applicare alla g

z = h * fft( corr .* w .* g );
z = abs(z);

v = 0.5/pi*exp(-alpha*xx-T*expcaratt(1i*alpha)).*z;

PrezzoFFT = exp(-r*T)*interp1(xx, v, x, 'spline');

end

```

A.2 Metodi di Pricing per opzioni Barriera

A.2.1 PricingPDO_HT

Questo script consente di prezzare un'opzione Put Down Out mediante la tecnica della trasformata di Hilbert.

```

%% Pricing Opzione Barriera Put Down-Out

function PrezzoFFT = PricingPDO_HT(S0, T, N, K, L, r, q, alpha,
    modello, params, M)

% ** Opzione Considerata
% Parametri opzione
l = log(L/K);
x = log(S0/K); % Trasformazione log-strike
Delta = T/N;
% Alpha
b = 1; theta = 1; % PutDownOut

% ** Modello Considerato
Frammento_LetturaParametri;

% ** Ulteriori Parametri
% Troncamento
if strcmp(modello, 'Merton')
    dd = 20;
    nu = 2;

```

```

    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu));      % Passo griglia
end
if strcmp(modello, 'Kou')
    dd = 26;
    nu = 2;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu));      % Passo griglia
end
if strcmp(modello, 'VG')
    h = 1;
end
if strcmp(modello, 'NIG')
    dd = 9.5;
    nu = 1;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu));      % Passo griglia
end
if strcmp(modello, 'BS')
    h = ((pi^2)/(Delta*params^4*T))^0.25*M^(-0.5); % Specifico per BS
end

% Griglia
k = (-M:M-1)';          % Indici della griglia spazio Fourier
MM = length(k);        % Numero elementi griglia Fourier
xi = k*h;               % Griglia spazio Fourier
% xi(M+1) = 1e-20;      % Per non avere errori numerici rimuovo lo zero

% ** Inizializzazione del ciclo
f_capp_alpha_DO = K .* ...
    ( (1 - exp((1i*xi+alpha)*b)) ./ (1i*xi + alpha) - ...
    (1 - exp((1i*xi+alpha+1)*b)) ./ (1i*xi + alpha+1) );
v_capp = f_capp_alpha_DO;

% ** Costruzione della matrice di Toeplitz
% Prima Colonna
index = (1:MM)';
% index = (-M:M)';
C1 = ( 1 - (-1).^(index - 1) )./(pi*(index - 1));
C1(isnan(C1)) = 0;
% Prima Riga
R1 = ( 1 - (-1).^(1 - index) )./(pi*(1 - index));
R1 = R1';
R1(isnan(R1)) = 0;

% ** Ciclo temporale
for j = N:-1:2
    % Funzione che deve subire la trasformata di Hilbert
    f_da_trasf = exp(-1i*xi*b) .* ...
        CharFunc(-xi+1i*alpha) / CharFunc(1i*alpha) .* ...
        v_capp;
    % Trasformata di Hilbert come prodotto di matrice Toeplitz
    trasfH = ProdottoToeplitz(C1, R1, f_da_trasf);
    % Calcolo della v_capp(j-1)
    v_capp = 0.5*exp(-Delta*expcaratt(1i*alpha)) .* ...
        CharFunc(-xi+1i*alpha)/CharFunc(1i*alpha) .* ...
        v_capp ...
        + ...

```

```

0.5*1i*theta.*exp(-Delta*expcaratt(1i*alpha)) .* ...
exp(1i*xi*b) .* trasfH;
end

% Calcolo della funzione integranda
g = CharFunc(-xi + 1i*alpha) / CharFunc(1i*alpha) ...
.* v_capp;

% ** Inversione finale mediante FFT
w = ones(size(xi)); w(1) = 0.5; w(end) = 0.5; % pesi

j = (0:2*M-1)'; % Indici nello spazio reale
k = j; % Traslo indici spazio Fourier
eta = 2*pi/(h*M); % Passo griglia spazio reale
xx = -M*eta+j*eta; % Punti griglia spazio reale

corr = exp(2*M/M*1i*pi*k); % Correzione da applicare alla g

z = h * fft(corr .* w .* g);
z = abs(z);

v = 0.5/pi*exp(-Delta*expcaratt(1i*alpha))*z;

PrezzoFFT = exp(-r*T)*(K./S0).^alpha.*interp1(xx, v, x, 'spline');
end

```

A.2.2 PricingCUO_HT

Questo script consente di prezzare un'opzione Call Up Out mediante la tecnica della trasformata di Hilbert.

%% Pricing Opzione Barriera Call Up-Out

```
function PrezzoFFT = PricingCUO_HT(S0, T, N, K, U, r, q, alpha,
    modello, params, M)
```

```

% ** Opzione Considerata
% Parametri opzione
u = log(U/K);
x = log(S0/K); % Trasformazione log-strike
Delta = T/N;
% Alpha
b = u; theta = -1; % CallUpOut

% ** Modello Considerato
Frammento_LetturaParametri;

% ** Ulteriori Parametri
% Troncamento
if strcmp(modello, 'Merton')
    dd = 20;
    nu = 2;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu)); % Passo griglia
end

```

```

if strcmp(modello, 'Kou')
    dd = 26;
    nu = 2;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'VG')
    dd = 9.5;
    h = 2; %Passo griglia
    N = 4;
end
if strcmp(modello, 'NIG')
    dd = 9.5;
    nu = 1;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'BS')
    h = ((pi^2)/(Delta*params^4*T))^0.25*M^(-0.5); % Specifico per BS
end

% Griglia
k = (-M:M-1)'; % Indici della griglia spazio Fourier
MM = length(k); % Numero elementi griglia Fourier
xi = k*h; % Griglia spazio Fourier
% xi(M+1) = 1e-20; % Per non avere errori numerici rimuovo lo zero

% ** Inizializzazione del ciclo
f_capp_alpha_UO = K .* ...
    ( (1 - exp((1i*xi+alpha)*b)) ./ (1i*xi + alpha) - ...
    (1 - exp((1i*xi+alpha+1)*b)) ./ (1i*xi + alpha+1) );
v_capp = f_capp_alpha_UO;

% ** Costruzione della matrice di Toeplitz
% Prima Colonna
index = (1:MM)';
% index = (-M:M)';
C1 = ( 1 - (-1).^(index - 1) )./(pi*(index - 1));
C1(isnan(C1)) = 0;
% Prima Riga
R1 = ( 1 - (-1).^(1 - index) )./(pi*(1 - index));
R1 = R1';
R1(isnan(R1)) = 0;

% ** Ciclo temporale
for j = N:-1:2
    % Funzione che deve subire la trasformata di Hilbert
    f_da_trasf = exp(-1i*xi*b) .* ...
        CharFunc(-xi+1i*alpha) / CharFunc(1i*alpha) .* ...
        v_capp;
    % Trasformata di Hilbert come prodotto di matrice Toeplitz
    trasfH = ProdottoToeplitz(C1, R1, f_da_trasf);
    % Calcolo della v_capp(j-1)
    v_capp = 0.5*exp(-Delta*expcaratt(1i*alpha)) .* ...
        CharFunc(-xi+1i*alpha)/CharFunc(1i*alpha) .* ...
        v_capp ...
        + ...
        0.5*1i*theta.*exp(-Delta*expcaratt(1i*alpha)) .* ...

```

```

        exp(1i*xi*b) .* trasfH;
end

% Calcolo della funzione integranda
g = CharFunc(-xi + 1i*alpha) / CharFunc(1i*alpha) ...
    .* v_capp;

% ** Inversione finale mediante FFT
w = ones(size(xi)); w(1) = 0.5; w(end) = 0.5; % pesi

j = (0:2*M-1)'; % Indici nello spazio reale
k = j; % Traslo indici spazio Fourier
eta = 2*pi / (h*M); % Passo griglia spazio reale
xx = -M*eta+j*eta; % Punti griglia spazio reale

corr = exp( 2*M/M*1i*pi*k ); % Correzione da applicare alla g

z = h * fft( corr .* w .* g);
z = abs(z);

v = 0.5/pi*exp(-Delta*expcaratt(1i*alpha))*z;

PrezzoFFT = exp(-r*T)*(K./S0).^alpha.*interp1(xx, v, x, 'spline');

end

```

A.2.3 PricingKO_HT

Questo script consente di prezzare un'opzione Knock Out mediante la tecnica della trasformata di Hilbert.

%% Pricing Opzione Barriera Put Knock-Out

```

function PrezzoFFT = PricingKO_HT(S0, T, N, K, U, L, r, q, alpha,
    modello, params, M)

% ** Opzione Considerata
% Parametri opzione
l = log(L/K);
u = log(U/K);
x = log(S0/K); % Trasformazione log-strike
Delta = T/N;
% Alpha
b = 1;

% ** Modello Considerato
Frammento_LetturaParametri;

% ** Ulteriori Parametri
% Troncamento
if strcmp(modello, 'Merton')
    dd = 20;
    nu = 2;
    h = ( (2*pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
        M^(-nu/(1+nu)); % Passo griglia
end

```

```

if strcmp(modello, 'Kou')
    dd = 26;
    nu = 2;
    h = ( (2*pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'VG')
    dd = 9.5;
    h = 2; % Passo griglia
    N = 4;
end
if strcmp(modello, 'NIG')
    dd = 9.5;
    nu = 1;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'BS')
    h = ((4*pi^2)/(Delta*params^4*T))^0.25*M^(-0.5); % Specifico per BS
end

% Griglia
k = (-M:M-1)'; % Indici della griglia spazio Fourier
MM = length(k); % Numero elementi griglia Fourier
xi = k*h; % Griglia spazio Fourier
% xi(M+1) = 1e-20; % Per non avere errori numerici rimuovo lo
zero

% ** Inizializzazione del ciclo
f_capp_alpha_DO = K .* ...
( (1 - exp((1i*xi+alpha)*b)) ./ (1i*xi + alpha) - ...
(1 - exp((1i*xi+alpha+1)*b)) ./ (1i*xi + alpha+1) );
v_capp = f_capp_alpha_DO;

% ** Costruzione della matrice di Toeplitz
% Prima Colonna
index = (1:MM)';
C1 = sin( h*(index - 1)*(u-1)/2) ./ (pi*(index-1));
C1(1) = (u-1)*h/(2*pi);
% Prima Riga
R1 = sin( h*(1 - index)*(u-1)/2) ./ (pi*(1-index));
R1 = R1';
R1(1) = (u-1)*h/(2*pi);

% ** Ciclo temporale
for j = N:-1:2
    % Funzione che deve subire la trasformata di Hilbert
    f_da_trasf = exp(-1i*xi*(u+1)/2) .* ...
    CharFunc(-xi+1i*alpha) / CharFunc(1i*alpha) .* ...
    v_capp;
    % Trasformata di Hilbert come prodotto di matrice Toeplitz
    trasfH = ProdottoToeplitz(C1, R1, f_da_trasf);
    % Calcolo della v_capp(j-1)
    v_capp = exp(-Delta*expcaratt(1i*alpha)) .* ...
    exp(1i*xi*(u+1)/2) .* trasfH;
end

% Calcolo della funzione integranda

```

```

g = CharFunc(-xi + 1i*alpha) / CharFunc(1i*alpha) ...
    .* v_capp;

% ** Inversione finale mediante FFT
w = ones(size(xi)); w(1) = 0.5; w(end) = 0.5; % pesi

j = (0:2*M-1)'; % Indici nello spazio reale
k = j; % Traslo indici spazio Fourier
eta = 2*pi / (h*MM); % Passo griglia spazio reale
xx = -M*eta+j*eta; % Punti griglia spazio reale

corr = exp( 2*M/MM*1i*pi*k ); % Correzione da applicare alla g

z = h * fft( corr .* w .* g);
z = abs(z);

v = 0.5/pi*exp(-Delta*expcaratt(1i*alpha))*z;

PrezzoFFT = exp(-r*T)*(K./S0).^alpha.*interp1(xx, v, x, 'spline');

end

```

A.2.4 PricingPDO_HT_Heston

Questo script consente di prezzare un'opzione Put Down Out mediante la tecnica della trasformata di Hilbert, nel caso di processi Time changed Lévy.

% Pricing Opzione Barriera PDO under time change Levy models con FHT

```

function Prezzo = PricingPDO_HT_Heston(S0, T, N, K, r, D, ...
    lambda, eta, v_segnato, v0, rho, M, J)

log_barr = ones(1, J) * log(D/K);
MM = 2*M+1; % Numero di elementi Fourier

n_s = length(S0);
Y0 = log(S0/K); % Valore iniziale in log-strike

nu = 2 * lambda * v_segnato / eta^2 - 1;
if (nu < 0)
    fprintf('nu=%1.2f<0, Feller non soddisfatta\n', nu);
end

Delta = T/N; % Tempo tra due istanti di monitoraggio
alpha = 5; % Parametro di dampening

% Troncamento griglia varianza logaritmica
ln_Evt = log( v0*exp(-lambda*T) + v_segnato*(1-exp(-lambda*T)) );
left = ln_Evt - 5 / (1+nu);
right = ln_Evt + 2 / (1+nu);
[left, right];
TOL_L = 1e-8;
TOL_R = 1e-3;

while P_transizione_log(left, log(v0), T, 0, lambda, eta, v_segnato) >
    TOL_L

```



```

    left = left - 0.01;
end
while P_transizione_log(right, log(v0), T, 0, lambda, eta, v_segnato)
    > TOL_R
    right = right + 0.01;
end
[left, right];
zeta_j = linspace(left, right, J)'; % Griglia log-activity rate

delta_j = zeta_j(2)-zeta_j(1); % passo log-activity
v_j = exp(zeta_j);

% ** Griglia activity-rate
% Griglia spazio di Fourier
h = 0.1339;
m = (-M:M)'; % Indici griglia spazio Fourier
beta = m*h; % Vettore colonna 2M+1 x 1

% ----- PREPARATION
% ** Inizializzazione Algoritmo
temp = alpha + 1i*beta; % variabile ausiliaria
Val = K ./ (temp .* (temp+1));
V_alpha_capp = repmat(Val, 1, J); % repmat per J volte, (2M+1)*J

% ** Calcolo della matrice PsiTilda
% Viene generata una variabile PSI_tilda, dimensione 2M+1 x J x J
Frammento_MatricePsiTilda;

% ** Costruzione della matrice di Toeplitz, formula (4.9b)
% Prima Colonna
index = (1:MM)';
C1 = ( 1 - (-1).^(index - 1))./(pi*(index - 1));
C1(isnan(C1)) = 0;
% Prima Riga
R1 = ( 1 - (-1).^(1 - index))./(pi*(1 - index));
R1 = R1';
R1(isnan(R1)) = 0;
% Costruzione Matrice
Toep = toeplitz(C1, R1);

% ----- INDUCTION
% Vettore dei pesi di integrazione
w = ones(J, 1)*delta_j; w(1) = 0.5*delta_j; w(end) = 0.5*delta_j;

% ** Ciclo temporale
for k = N-1:-1:1
    % -----
    % ** Aggiornamento di H, F, G_capp
    Hmat = exp(-1i .* m .* h * log_barr); %(2M+1)*J

    Fmat = 1./Hmat; %(2M+1)*J

    % ** Aggiornamento della matrice B = B(t_k)
    % B_k ha fissato il tempo t_k, sulle colonne ha i diversi gamma_p
    B_k = zeros(MM, J);
    for ind_p = 1:J
        k_p = ( V_alpha_capp .* PSI_tilda(:, :, ind_p) ) * w; %
        (2M+1) x 1
    end
end

```

```

        B_k(:, ind_p) = k_p; %
(2M+1) x J
    end

    % ** Calcolo del valore allo step precedente
    V_alpha_capp = exp(-r*Delta)/2 * B_k + ...
        1i*exp(-r*Delta)/2 * (Toep*(B_k.*Hmat)).*Fmat;
end

%%
% ** Inversione finale
% V = V(gamma_0, S_0), dove le gamma_0 sono sulla griglia
V = zeros(J, n_s);
for indp = 1:J
    % Ciclo sugli S_0
    for inds = 1:n_s
        V(indp, inds) = exp(-r*Delta)/(2*pi) * ...
            sum(w'.* ... % Somma sui j
            sum( ... % Somma sugli m
            exp(-(alpha+1i*repmat(beta, 1,
            J))*Y0(inds)).*V_alpha_capp.*PSI_tilda(:, :, indp)*h));
    end
end

% Interpolazione fatta sulla varianza, gamma_0 = log(v_0)
V = real(V);
Prezzo = interp1(zeta_j, V, log(v0), 'spline');

end

```

A.2.5 PricingPDO_MC_Heston

Questo script consente di prezzare un'opzione Put Down Out con dinamica di Heston, usando la tecnica dell'integrazione Monte Carlo.

```

clc;
clear all;
close all;
format compact;

%% Pricing Opzione Barriera PDO under time change Levy models con MC

% Parametri
S0 = 100; % Spot price
K = 100; % Strike
T = 1; % Time to Maturity
r = 0.05; % Tasso di Interesse
Nsim = 1e5; % Numero di simulazioni
Nsteps = 12; % Numero date di monitoraggio
D = 80; % Barriera Down
flag = -1; % flag = 1:Call, flag = -1:Put

% >> Modello Heston
V0 = 0.0625;
rho = 0.1;

```

```

eta = 0.9;
lambda = 5;
v_segnato = 0.16;

%% 1 — Monte Carlo Classico

tic

% >> Simulazione del Sottostante
S = AssetHeston(S0, r, V0, rho, lambda, v_segnato, eta, T, Nsim,
    Nsteps);
ST = S(:, end);

% >> Calcolo del Payoff Scontato
DiscPayoff = zeros(Nsim, 1);
for i = 1:Nsim
    DiscPayoff(i) = max(flag*(ST(i)-K), 0)*exp(-r*T); %CallPut
    DiscPayoff(i) = DiscPayoff(i) * (min(S(i, :)) > D);
end

% >> Stima del prezzo
[Prezzo, ~, IC] = normfit(DiscPayoff); %IC al 95%
toc

Prezzo
IC
ampiezza = IC(2)-IC(1)

```

A.3 Metodi di Pricing per opzioni Lookback basati su FHT

A.3.1 PricingLookbackPut_HT

Questo script consente di prezzare un'opzione Lookback mediante la tecnica della trasformata di Hilbert.

```

%% Pricing Opzione Lookback

function PrezzoFFT = PricingLookbackPut_HT(S0, T, N, r, q, s, modello,
    params, M)

% ** Opzione Considerata
% Parametri opzione
Delta = T/N; % Passo temporale Europea

% ** Modello Considerato
Frammento_LetturaParametri_LB;

%% ** Controllo Parametri
Frammento_ControlloParametri_LB;

% ** Ulteriori Parametri
% Troncamento
if strcmp(modello, 'Merton')

```

```

    dd = 20;
    nu = 2;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'Kou')
    dd = 26;
    nu = 2;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'VG')
    h = 2;
end
if strcmp(modello, 'NIG')
    dd = 9.5;
    nu = 1;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
    M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'BS')
    h = ((pi^2)/(Delta*params^4*T))^0.25*M^(-0.5); % Specifico per BS
end

% Griglia
k = (-M:M-1)'; % Indici della griglia spazio Fourier
MM = length(k); % Numero elementi griglia Fourier
xi = k*h; % Griglia spazio Fourier

% ** Costruzione della matrice di Toeplitz
% Prima Colonna
index = (1:MM)';
C1 = ( 1 - (-1).^(index - 1) )./(pi*(index - 1));
C1(isnan(C1)) = 0;
% Prima Riga
R1 = ( 1 - (-1).^(1 - index) )./(pi*(1 - index));
R1 = R1';
R1(isnan(R1)) = 0;

% ** Inizializzazione
c = 1;
f = zeros(size(xi));
g = zeros(size(xi));

% ** Ciclo
for j = 1:N
    % ** Aggiorno f
    vect1 = CharFunc(-xi).*(c + f);
    f = 0.5*vect1 + ...
        1i*0.5*ProdottoToeplitz(C1, R1, vect1);
    % ** Aggiorno g
    vect2 = CharFunc(-xi + 1i*s).*(c + g);
    g = 0.5*vect2 + ...
        1i*0.5*ProdottoToeplitz(C1, R1, vect2);
    % ** Aggiorno c
    c = 1 - f(M+1); % Ci va f(0), centro della griglia
end

```

```

% ** Formula finale
L = CharFuncT(-1i*s) .* (c + g(M+1));
L = real(L);

% ** Formula di pricing
% Floating Strike Lookback Put
PrezzoFFT = S0*exp(-r*T)*L - exp(-q*T)*S0;

end

```

A.4 Metodi di Pricing per opzioni Bermuda basati su FHT

A.4.1 PricingBermudanPut_HT

Questo script consente di prezzare un'opzione Bermuda mediante la tecnica della trasformata di Hilbert.

```

%% Pricing Opzione Bermuda con FHT

function [PrezzoFFT, Esercizio, x_star_v] = PricingBermudanPut_HT(S0,
    T, N, K, r, q, alpha, modello, params, M)

% ** Opzione Considerata
% Parametri opzione
x = log(S0/K); % Trasformazione log-strike
Delta = T/N;

% ** Modello Considerato
LetturaParamModello;

% ** Ulteriori Parametri
% Troncamento
if strcmp(modello, 'Merton')
    dd = 20;
    nu = 2;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
        M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'Kou')
    dd = 26;
    nu = 2;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
        M^(-nu/(1+nu)); % Passo griglia
end
if strcmp(modello, 'VG')
    h = 1;
end
if strcmp(modello, 'NIG')
    dd = 5;
    nu = 1;
    h = ( (pi*dd)/(Delta*c) )^(1/(1+nu)) * ...
        M^(-nu/(1+nu)); % Passo griglia
end
end

```

```

if strcmp(modello, 'BS')
    h = ((pi^2)/(Delta*params^4*T))^0.25*M^(-0.5); % Specifico per BS
end

eps_NR = 1e-4; % Epsilon Newton-Rapson
% Griglia
k = (-M:M-1)'; % Indici della griglia spazio Fourier
MM = length(k); % Numero elementi griglia Fourier
xi = k*h; % Griglia spazio Fourier
% xi(M+1) = 1e-20; % Per non avere errori numerici rimuovo lo zero

% ** Inizializzazione del ciclo (per le europee niente ciclo!)
f_capp_alpha = K ./ ( (alpha + 1i*xi) .* (alpha + 1 + 1i*xi) );

% ** Costruzione della matrice di Toeplitz
% Prima Colonna
index = (1:MM)';
C1 = ( 1 - (-1).^(index - 1) )./(pi*(index - 1));
C1(isnan(C1)) = 0;
% Prima Riga
R1 = ( 1 - (-1).^(1 - index) )./(pi*(1 - index));
R1 = R1';
R1(isnan(R1)) = 0;

% ** Ciclo temporale
% Salvo dati su ogni iterazione
x_star_v = zeros(N-1, 1); % Punti di esercizio
iter_v = zeros(N-1, 1); % Numero iterazioni
err_v = zeros(N-1, 1); % Errore all'arresto
x_star = 0; % Ciclo inizializzato con x_star = 0
MAXITER = 200; % Massimo numero di iterazioni
for j = N-1:-1:0
    % ** Ricerca di x_capp col metodo di Newton-Raphson
    err = 1e2; % Inizializzo l'errore a un valore elevato
    iter = 0; % Inizializzo a 0 le iterazioni di
    Newton-Raphson
    while (err > eps_NR) && (iter < MAXITER)
        iter = iter + 1;
        g = K*exp(alpha*x_star)*(1-exp(x_star))*(x_star < 0);
        g_der = K*exp(alpha*x_star)*(alpha-(1+alpha)*exp(x_star));
        f = h/(2*pi)*sum( exp(-1i*x_star*xi) .* ...
            f_capp_alpha .* CharFunc(-xi + 1i*alpha)) ...
            - g;
        f_der = -1i*h/(2*pi)*sum( exp(-1i*x_star*xi) .* ...
            xi .* f_capp_alpha .* CharFunc(-xi + 1i*alpha)) ...
            - g_der;
        step = f/f_der;
        x_star = real(x_star - step); % Rimuovo la piccola parte
        immaginaria
        err = abs(step);
    end
    iter_v(N-j+1) = iter;
    err_v(N-j+1) = err;
    x_star_v(N-j+1) = x_star;

% ** Calcolo di f_capp al tempo precedente

```

```

f_da_trasf = exp(-1i*x_star*xi).*f_capp_alpha.*CharFunc(-xi +
1i*alpha);
trasfH = ProdottoToeplitz(C1, R1, f_da_trasf);
f_capp_alpha = K*exp((1i*xi + alpha)*x_star) .* ...
( 1./(1i*xi + alpha) - exp(x_star)./(1i*xi + alpha + 1) ) ...
+ ...
exp(-r*Delta) *...
(0.5*f_capp_alpha.*CharFunc(-xi + 1i*alpha) + ...
0.5*1i*exp(1i*xi*x_star).*trasfH);
end

% ** Inversione finale mediante FFT
integranda = CharFunc(-xi + 1i*alpha) ...
.* f_capp_alpha;

w = ones(size(xi)); w(1) = 0.5; w(end) = 0.5; % pesi

j = (0:2*M-1)'; % Indici nello spazio reale
k = j; % Traslo indici spazio Fourier
eta = 2*pi / (h*MM); % Passo griglia spazio reale
xx = -M*eta+j*eta; % Punti griglia spazio reale

corr = exp( 2*M/MM*1i*pi*k ); % Correzione da applicare alla g

z = h * fft( corr .* w .* integranda);
z = abs(z);

gg = K*exp(alpha*xx).*max(1-exp(xx), 0);
v = 0.5/pi*exp(-r*Delta)*z;

PrezzoFFT = interp1(xx, v.*exp(-alpha*xx), x, 'spline');
Esercizio = interp1(xx, gg.*exp(-alpha*xx), x, 'spline');
x_star_v = x_star_v';
end

```

A.4.2 PricingBermudanPut_HT_Heston

Questo script consente di prezzare un'opzione Bermuda mediante la tecnica della trasformata di Hilbert, nel caso di processi Time changed Lévy.

```

% Pricing Opzione Bermuda under time change Levy models con FHT

function Prezzo = PricingBermudanPut_HT_Heston(S0, T, N, K, r, ...
lambda, eta, v_segnato, v0, rho, M, J)

MM = 2*M+1; % Numero di elementi Fourier

n_s = length(S0);
Y0 = log(S0/K); % Valore iniziale in log-strike

eps_NR = 1e-4; % Epsilon Newton-Rapson

nu = 2 * lambda * v_segnato / eta^2 - 1;
if (nu < 0)
    fprintf('nu=%1.2f<0, Feller non soddisfatta\n', nu);
end
end

```

```

Delta = T/N;          % Tempo tra due istanti di monitoraggio
alpha = 5;           % Parametro di dampening

% Troncamento griglia varianza logaritmica
ln_Evt = log( v0*exp(-lambda*T) + v_segnato*(1-exp(-lambda*T)) );
left = ln_Evt - 5 / (1+nu);
right = ln_Evt + 2 / (1+nu);
[left , right];
TOL_L = 1e-8;
TOL_R = 1e-3;

while P_transizione_log(left , log(v0), T, 0, lambda, eta , v_segnato) >
    TOL_L
    left = left - 0.01;
end
while P_transizione_log(right , log(v0), T, 0, lambda, eta , v_segnato)
    > TOL_R
    right = right + 0.01;
end
[left , right];
zeta_j = linspace(left , right , J)'; % Griglia log-activity rate

delta_j = zeta_j(2)-zeta_j(1);      % passo log-activity
v_j = exp(zeta_j);

% ** Griglia activity-rate
% Griglia spazio di Fourier
h = 0.1339;
m = (-M:M)';                       % Indici griglia spazio Fourier
beta = m*h;                         % Vettore colonna 2M+1 x 1

% ----- PREPARATION
% ** Inizializzazione Algoritmo
temp = alpha + li*beta;             % variabile ausiliaria
Val = K ./ (temp .* (temp+1));
V_alpha_capp = repmat(Val, 1, J);  % repmat per J volte , (2M+1)*J

% ** Calcolo della matrice PsiTilda
% Viene generata una variabile PSI_tilda , dimensione 2M+1 x J x J
Frammento_MatricePsiTilda;

% ** Costruzione della matrice di Toeplitz , formula (4.9b)
% Prima Colonna
index = (1:MM)';
C1 = ( 1 - (-1).^ (index - 1) )./( pi*(index - 1) );
C1(isnan(C1)) = 0;
% Prima Riga
R1 = ( 1 - (-1).^ (1 - index) )./( pi*(1 - index) );
R1 = R1';
R1(isnan(R1)) = 0;
% Costruzione Matrice
Toep = toeplitz(C1, R1);

% ----- INDUCTION
% Vettore dei pesi di integrazione
w = ones(J, 1)*delta_j; w(1) = 0.5*delta_j; w(end) = 0.5*delta_j;

```



```

% ** Ciclo temporale
% Salvo dati su ogni iterazione
x_star_v = zeros(N-1, J); % Punti di esercizio
iter_v = zeros(N-1, J); % Numero iterazioni
err_v = zeros(N-1, J); % Errore all'arresto

x_star_gam = zeros(1, J); % Ciclo inizializzato con x_star = 0
MAXITER = 200; % Massimo numero di iterazioni
for k = N-1:-1:1
    % ----- VETTORE x*
    % x = x(gamma_tk, tk) --- dimensione pari a gamma_tk, vettore, t_k
    % fissato
    % ** Ciclo sulla varianza logaritmica gamma_tk e calcolo x*(:, tk)
    for p = 1:J
        % ** Ricerca di x_capp col metodo di Newton-Raphson
        err = 1e2; % Inizializzo l'errore a un valore
        elevato
        iter = 0; % Inizializzo a 0 le iterazioni di
        Newton-Raphson
        x_star = x_star_gam(p); % Inizializzo cosi: x*_k,p = x*_k+1,p
        while (err > eps_NR) && (iter < MAXITER)
            iter = iter + 1;
            % Funzioni g, g', f, f'
            g = K*exp(alpha*x_star)*(1-exp(x_star))*(x_star < 0);
            g_der = K*exp(alpha*x_star)*(alpha-(1+alpha)*exp(x_star));
            f = exp(-r*Delta)*h/(2*pi)*sum(w'...
                .*sum( repmat(exp(-1i*x_star*beta), 1, J)...
                    .* V_alpha_capp .* shiftdim(PSI_tilda(:, :, p)) ) ) ...
                - g;
            f_der = -1i*exp(-r*Delta)*h/(2*pi)*sum(w'...
                .*sum( repmat(exp(-1i*x_star*beta), 1, J).* ...
                    repmat(beta, 1, J) .* V_alpha_capp .*
            shiftdim(PSI_tilda(:, :, p)) ) ) ...
                - g_der;
            % Passo dell'iterazione
            step = f/f_der;
            x_star = real(x_star - step); % Rimuovo la piccola parte
            immaginaria
            err = abs(step);
        end
        iter_v(N-k+1, p) = iter;
        err_v(N-k+1, p) = err;
        x_star_v(N-k+1, p) = x_star;
    end

    % Estraggo la riga corrispondente al tempo attuale
    x_star_gam = x_star_v(N-k+1, :);

    % -----
    % ** Aggiornamento di H, F, G_capp
    Hmat = exp(-1i .* m .* h .* x_star_gam); %(2M+1)*J

    Fmat = 1./Hmat; %(2M+1)*J

    G_capp = K * exp( temp*x_star_gam) .* ...
        ( repmat( 1./temp, [1, length(x_star_gam)] ) - ...
        (1./(temp+1))*exp(x_star_gam) ...
        );

```

```

% ** Aggiornamento della matrice B = B(t_k)
% B_k ha fissato il tempo t_k, sulle colonne ha i diversi gamma_p
B_k = zeros(MM, J);
for ind_p = 1:J
    k_p = ( V_alpha_capp .* PSI_tilda(:, :, ind_p) ) * w; %
(2M+1) x 1
    B_k(:, ind_p) = k_p; %
(2M+1) x J
end

% ** Calcolo del valore allo step precedente
V_alpha_capp = G_capp + ...
                exp(-r*Delta)/2 * B_k + ...
                1i*exp(-r*Delta)/2 * (Toep*(B_k.*Hmat)).*Fmat;
end

%%
% ** Inversione finale
% V = V(gamma_0, S_0), dove le gamma_0 sono sulla griglia
V = zeros(J, n_s);
for indp = 1:J
    % Ciclo sugli S_0
    for inds = 1:n_s
        V(indp, inds) = exp(-r*Delta)/(2*pi) * ...
            sum(w'.* ... % Somma sui j
            sum( ... % Somma sugli m
            exp(-(alpha+1i*repmat(beta, 1, J))*Y0(inds)).*...
            V_alpha_capp.*PSI_tilda(:, :, indp)*h));
    end
end

% Interpolazione fatta sulla varianza, gamma_0 = log(v_0)
V = real(V);
Prezzo = interp1(zeta_j, V, log(v0), 'spline');
end

```

A.5 Metodi di Pricing modello Dividend Ruin

A.5.1 PricingDividendRuin_HT_Heston

Questo script consente di prezzare secondo il modello dividend-ruin mediante la tecnica della trasformata di Hilbert, nel caso di processi Time changed Lévy.

% Pricing dividend-ruin firm asset value model under time change Levy

```

function Prezzo = PricingDividendRuin_HT_Heston(A0, T, r, L, B,
        lambda, eta, v_segnato, v0, rho, J, M, N, alpha)

MM = 2*M+1; % Numero di elementi Fourier

n_s = length(A0);
Z0 = log(A0);

```

```

b = log(B);
l = log(L);

% ** Modello Considerato
nu = 2 * lambda * v_segnato / eta^2 - 1;
if (nu < 0)
    fprintf('nu=%1.2f<0, Feller non soddisfatta\n', nu);
end

% ** Ulteriori Parametri
Delta = T/N; % Tempo tra due istanti di monitoraggio

% Troncamento griglia varianza logaritmica
ln_Evt = log( v0*exp(-lambda*T) + v_segnato*(1-exp(-lambda*T)) );
left = ln_Evt - 5 / (1+nu);
right = ln_Evt + 2 / (1+nu);
[left , right];
TOL_L = 1e-8;
TOL_R = 1e-3;

while P_transizione_log(left , log(v0), T, 0, lambda, eta , v_segnato) >
    TOL_L
    left = left - 0.01;
end
while P_transizione_log(right , log(v0), T, 0, lambda, eta , v_segnato)
    > TOL_R
    right = right + 0.01;
end
[left , right];
zeta_j = linspace(left , right , J)'; % Griglia log-activity rate

delta_j = zeta_j(2)-zeta_j(1); % passo log-activity
v_j = exp(zeta_j);

% Griglia spazio di Fourier
h = 0.1337;
m = (-M:M)'; % Indici griglia spazio Fourier
beta = m*h; % Vettore colonna 2M+1 x 1

% ** Inizializzazione Algoritmo
temp = alpha + 1i*beta; % variabile ausiliaria
Val = (B*exp(temp*b) - L*exp(temp*l)) ./ (temp + 1) ...
    - B*exp(temp*b) ./ temp;
V_alpha_capp = repmat(Val, 1, J); % repmat per J volte , (2M+1)*J

% ** Costruzione della matrice di Toeplitz
% Prima Colonna
index = (1:MM)';
C1 = sin( h*(index-1)*log(B/L)/2 ) ./ (pi*(index-1));
C1(1) = h*log(B/L)/(2*pi);
% Prima Riga
R1 = sin( h*(1-index)*log(B/L)/2 ) ./ (pi*(1-index));
R1 = R1';
R1(1) = h*log(B/L)/(2*pi);
% Costruzione Matrice
Toep = toeplitz(C1, R1);

```

```

% ** Costruzione delle matrici e vettori ausiliari per il ciclo
% Matrici H, F
Hmat = exp(-1i .* m .* h .* (b + 1) ./ 2);
Hmat = repmat(Hmat, 1, J); %(2M+1)*J
Fmat = 1./Hmat;           %(2M+1)*J
% Vettori xi, s
xivect = exp(-1i .* m .* h .* b) .* h;
svect = exp(1i .* m .* h .* b) ./ (alpha + 1i .* m .* h);
% Vettore dei pesi di integrazione
w = ones(J, 1)*delta_j; w(1) = 0.5*delta_j; w(end) = 0.5*delta_j;

% ** Calcolo della matrice PsiTilda
% Viene generata una variabile PSI_tilda, dimensione 2M+1 x J x J
Frammento_MatricePsiTilda;

% ** Ciclo esterno sul tempo, induzione backward
for k = N-1:-1:1
% ** Aggiornamento della matrice B = B(t_k)
B_k = zeros(MM, J);
for ind_p = 1:J
k_p = ( V_alpha_capp .* PSI_tilda(:, :, ind_p) ) * w; %
(2M+1) x 1
B_k(:, ind_p) = k_p; %
(2M+1) x J
end
% ** Calcolo del valore allo step precedente
V_alpha_capp = exp(-r*Delta) * ( Toep*(B_k.*Hmat) ).*Fmat ...
- ...
exp(-r*Delta)/(2*pi) * svect * (xivect.'*B_k);
end

%%
% ** Inversione finale
% V = V(gamma_0, S_0), dove le gamma_0 sono sulla griglia
V = zeros(J, n_s);
for indp = 1:J
% Ciclo sugli S_0
for inds = 1:n_s
V(indp, inds) = exp(-r*Delta)/(2*pi) * ...
sum(w'.* ... % Somma sui j
sum( ... % Somma sugli m
exp(-(alpha+1i*repmat(beta, 1, J))*Z0(inds)).*...
V_alpha_capp.*PSI_tilda(:, :, indp)*h));
end
end

% Interpolazione fatta sulla varianza, gamma_0 = log(v_0)
V = real(V);
Prezzo = interp1(zeta_j, V, log(v0), 'spline');

end

```

A.5.2 PricingDividendRuin_MC_Heston

Questo script consente di prezzare secondo il modello dividend-ruin con dinamica di Heston, usando la tecnica dell'integrazione Monte Carlo.

```

clc;
clear all;
close all;
format short;
format compact;

%% Pricing Modello Dividend Ruin under time change Levy models con MC

% Parametri opzione
S0 = 10;
r = 0.04;
T = 1;
L = 7;           % Barriera assorbente
B = 13;         % Barriera riflettente

% Parametri MC
Nsteps = 10;    % Date di monitoraggio
Nsim = 1e4;     % Numero di simulazioni

% Parametri modello
% Heston
V0 = 0.0348;
rho = -0.64;
eta = 0.39;
lambda = 1.15;
v_segnato = 0.0348;

%% Computazione
tic

% Simulazione del sottostante
S = AssetHeston(S0, r, V0, rho, lambda, v_segnato, eta, T, Nsim,
    Nsteps);

% Calcolo del minimo e massimo mobile
% Smin(t) = min S(i), i <= t
% Smax(t) = max S(i), i <= t
S_min_mobile = zeros(Nsim, Nsteps);
S_max_mobile = zeros(Nsim, Nsteps);
for i = 1:Nsteps+1
    S_min_mobile(:, i) = min( S(:, 1:i), [], 2);
    S_max_mobile(:, i) = max( S(:, 1:i), [], 2);
end

% S cappuccio, incorpora i dividendi
S_capp = S .* min(1, B./S_max_mobile);

% S tilda, definito solo a scadenza, contempla la ruin
S_tilda = S_capp(:, end) .* (min(S_capp, [], 2) > L);

% Prezzo col MC
discpayoff = exp(-r*T)*S_tilda;

```

```
[V, ~, IC] = normfit(discpayoff);
toc
```

```
%% Risultati
```

```
V
ampiezza = (IC(2) - IC(1))/2
```

A.6 Funzioni ausiliarie

A.6.1 ProdottoToeplitz

Funzione che esegue la moltiplicazione di una matrice di Toeplitz per un vettore utilizzando la tecnica FFT.

```
function Prod = ProdottoToeplitz(C1, R1, vect)

% Rendo C1 vettore colonna e R1 vettore riga se non lo sono gia
[m, n] = size(C1);
if (n > m)
    C1 = C1';
end
[m, n] = size(R1);
if (m > n)
    R1 = R1';
end

% Moltiplicazione
n = length(C1);
aux = [C1; 0; fliplr(R1(2:end)).'];
p = ifft(fft(aux).*fft([vect; zeros(n,1)]));
Prod = p(1:n);

end
```

A.6.2 P_transizione_log

Questo script contiene la probabilità di transizione della varianza logaritmica.

```
function Val = P_transizione_log(gt, gs, t, s, lambda, eta, v_segnato)
% P
% probabilita di transizione da gammas a gammat, dove gamma e log
% varianza,
% in modo da soddisfare Feller

nu = 2 * lambda * v_segnato / eta^2 - 1;
zeta = 2*lambda ./ ( (1-exp(-lambda*(t-s))) * eta^2 );

argB = 2*zeta.*exp(-0.5*lambda*(t-s)).*sqrt(exp(gt).*exp(gs));

Val = zeta ...
    .* exp(-zeta .* ( exp(gs - lambda*(t-s)) + exp(gt) ) ) ...
    .* (exp(gt-gs+lambda*(t-s))).^(nu/2) ...
```

```

.* exp(gt) ...
.* ...
    besseli(nu, argB);
end

```

A.6.3 CharF_CIR_Cond

Questo script contiene la funzione caratteristica condizionata del processo CIR.

```

function PHI = CharF_CIR_Cond(xi, gt, gs, t, s, lambda, eta, v_segnato)
%Funzione generatrice dei momenti condizionata della varianza
%logaritmica

nu = 2 * lambda * v_segnato / eta^2 - 1;

gamma_tilda = @(xi) sqrt(lambda^2 - 2*1i*eta^2*xi);

argBnum = sqrt( exp(gt + gs) ) .* ...
    4*gamma_tilda(xi) .* exp(-0.5*gamma_tilda(xi)*(t-s)) ./ ...
    (eta^2.*(1-exp(-gamma_tilda(xi)*(t-s))));

argBden = sqrt( exp(gt + gs) ) .* ...
    4*lambda.*exp(-0.5 * lambda.*(t-s)) ./ ...
    (eta^2.*(1-exp(-lambda.*(t-s))));

PHI = besseli(nu, argBnum) ./ ...
    besseli(nu, argBden) ...
    .* ...
    gamma_tilda(xi) .* exp(-0.5*(gamma_tilda(xi)-lambda).(t-s)) * ...
    (1-exp(-lambda*(t-s))) ./ ...
    (lambda*(1-exp(-gamma_tilda(xi).(t-s)))) ...
    .* ...
    exp( (exp(gs)+exp(gt))./eta^2 .* ...
    ( (lambda*(1+exp(-lambda.*(t-s)))/(1-exp(-lambda.*(t-s)))) - ...
    gamma_tilda(xi) .* (1+exp(-gamma_tilda(xi).(t-s)))./...
    (1-exp(-gamma_tilda(xi).(t-s))));

end

```

A.6.4 Frammento_MatricePsiTilda

Questo script contiene la funzione generatrice dei momenti condizionata del processo CIR.

```

% ** Esponente caratteristico processo ausiliario X
CharExpAux = @(x) (0.5 - rho*lambda/eta)*1i*x + ...
    0.5*(1 - rho^2)*x.^2;

% ** Probabilita di transizione log activity del CIR
Ptrans = zeros(J, J);
for ind = 1:J
    Ptrans(ind, :) = P_transizione_log(zeta_j(ind), zeta_j, Delta, 0,
    lambda, eta, v_segnato);
end

```

```

% ** Matrice PSI, funzione generatrice dei momenti condizionata
% della differenza dei rendimenti logaritmici di Heston
PSI = zeros(MM, J, J);
for indp = 1:J
    % creo matrice PSI(p) con p fissato.
    parziale = zeros(MM, J);
    for indbeta = 1:MM
        omega = -alpha - 1i*beta(indbeta);
        % Dove calcolare la PHI per avere la PSI.
        arg = 1i*CharExpAux(-1i*omega);
        parziale(indbeta, :) = ...
            exp( omega*( r*(Delta-0) + ...
                rho/eta*(
exp(zeta_j(indp)) - exp(zeta_j) - lambda*v_segnato*(Delta-0) ) ) ) .*
            ...
            CharF_CIR_Cond(arg, zeta_j(indp), zeta_j, Delta, 0,
lambda, eta, v_segnato);
    end
    % Introduco la matrice appena calcolata nella collezione di
    % matrici.
    PSI(:, :, indp) = parziale;
end

% ** Matrice PSI_TILDA
PSI_tilda = zeros(MM, J, J);
for indbeta = 1:MM
    PSI_tilda(indbeta, :, :) = ...
        shiftdim(PSI(indbeta, :, :)) .* Ptrans;
end

```


Bibliografia

- [1] P. Baldi. *Equazioni differenziali stocastiche e applicazioni*. Quaderni dell'Unione Matematica Italiana. Pitagora, 2000. ISBN: 9788837112110. URL: <http://books.google.it/books?id=qkHMPQAACAAJ>.
- [2] E. Barucci. *Ingegneria finanziaria. Un'introduzione quantitativa*. I Manuali. EGEA, 2009. ISBN: 9788823820951. URL: <http://books.google.it/books?id=fWeEPgAACAAJ> (cit. a p. 6).
- [3] T. Björk. *Arbitrage Theory in Continuous Time*. Oxford Finance Series. Oxford University Press, Incorporated, 2009. ISBN: 9780199574742. URL: <http://books.google.it/books?id=Z9rHMgEACAAJ> (cit. alle pp. 6, 24).
- [4] Peter Carr e Dilip B. Madan. «Option Valuation Using the Fast Fourier Transform». In: *Journal of computational finance* 2 (1999), pp. 61–73.
- [5] R. Cont e P. Tankov. *Financial Modelling with Jump Processes*. Chapman & Hall/Crc Financial Mathematics Series. Chapman & Hall/CRC, 2012. ISBN: 9781420082197. URL: <http://books.google.it/books?id=-fZtKgAACAAJ> (cit. alle pp. 6, 7, 25).
- [6] F. Fang e C. W. Oosterlee. «A Fourier-Based Valuation Method For Bermudan And Barrier Options Under Heston's Model». en. In: *SIAM Journal on Financial Mathematics* 2 (2011), pp. 439–463. URL: <http://oai.cwi.nl/oai/asset/18532/18532D.pdf> (cit. alle pp. xvii, xviii, 16, 20).
- [7] Liming Feng e Xiong Lin. «Pricing Bermudan Options in Lévy Process Models». In: *SIAM J. Financial Math.* 4.1 (2013), pp. 474–493 (cit. alle pp. xvii, 69, 70, 74).
- [8] Liming Feng e Vadim Linetsky. «Computing exponential moments of the discrete maximum of a Lévy process and lookback options». In: *Finance and Stochastics* 13.4 (2009), pp. 501–529. DOI: [10.1007/s00780-009-0096-x](https://doi.org/10.1007/s00780-009-0096-x). URL: <http://dx.doi.org/10.1007/s00780-009-0096-x> (cit. alle pp. xvii, 61).

- [9] Liming Feng e Vadim Linetsky. «Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Hilbert transform approach». In: *Mathematical Finance* 18.3 (2008), pp. 337–384. URL: <http://EconPapers.repec.org/RePEc:bla:mathfi:v:18:y:2008:i:3:p:337-384> (cit. alle pp. xvii, 28, 30, 41–44, 96).
- [10] E.R. Gianin e C. Sgarra. *Esercizi di finanza matematica*. Collana Unitext. Springer, 2007. ISBN: 9788847006102. URL: <http://books.google.it/books?id=Z209o006jZcC>.
- [11] J. Jacod e P.E. Protter. *Probability Essentials*. Universitext (1979). Springer Berlin Heidelberg, 2003. ISBN: 9783540438717. URL: http://books.google.de/books?id=OK%5C_d-w18EVgC.
- [12] J. Kienitz e D. Wetterau. *Financial Modelling: Theory, Implementation and Practice with MATLAB Source*. The Wiley Finance Series. Wiley, 2012. ISBN: 9781118413319. URL: <http://books.google.it/books?id=fZpu8-sNPo8C>.
- [13] Kwai Sun Leung, Yue Kuen Kwok e Seng Yuen Leung. «Finite-time dividend-ruin models». In: *Insurance: Mathematics and Economics* 42.1 (2008), pp. 154–162. URL: <http://EconPapers.repec.org/RePEc:eee:insuma:v:42:y:2008:i:1:p:154-162> (cit. a p. 87).
- [14] A.J. McNeil, R. Frey e P. Embrechts. *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton Series in Finance. Princeton University Press, 2010. ISBN: 9781400837571. URL: <http://books.google.it/books?id=vgy98mM9zQUC>.
- [15] P.J. Schönbucher. *Credit Derivatives Pricing Models: Models, Pricing and Implementation*. The Wiley Finance Series. Wiley, 2003. ISBN: 9780470868171. URL: http://books.google.it/books?id=YDeJ%5C_Kv8QZwC.
- [16] R. Seydel. *Tools for Computational Finance*. Universitext. Springer, 2009. ISBN: 9783540929291. URL: <http://books.google.it/books?id=1r8uixovrtQC>.
- [17] P. Wilmott, J.N. Dewynne e S. Howison. *Option Pricing: Mathematical Models and Computation*. Oxford Financial, 1998. URL: <http://books.google.it/books?id=cGuGcgAACAAJ>.
- [18] Pingping Zeng e Yue Kuen Kwok. «Pricing Barrier and Bermudan Style Options Under Time-Changed Lévy Processes: Fast Hilbert Transform Approach». In: *SIAM J. Scientific Computing* 36.3 (2014) (cit. alle pp. xvii, 19, 55, 69, 77, 83, 85).