

POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Matematica



Processi di Poisson con salti correlati tramite
self-decomposition: pricing di opzioni exchange

Relatore: Prof. Daniele Marazzina

Tesi di Laurea di:
Andrea Braccini Matr. 900333

Indice

Elenco delle figure	III
Elenco delle figure	III
Elenco delle tabelle	IV
1 Introduzione	1
2 Presentazione teorica	5
2.1 Distribuzione congiunta di Poisson	5
2.2 Ipergeometrica confluyente	9
2.3 Modelli con salti	11
2.3.1 GBM	11
2.3.2 GOU	14
2.3.3 Schwartz-Smith	17
3 Pricing	20
3.1 Opzione europea	20
3.2 Opzione exchange	23
4 Applicazione numerica	26
4.1 Applicazione al caso GBM	28
4.2 Applicazione al caso GOU	36
5 Conclusioni	42

Appendices	46
A Codici matlab	47
B Correlazione jump diffusion	55
C Probabilità usate per calibrazione caso GOU	56
Bibliografia	58

Elenco delle figure

4.1	Valori dell'opzione con $M_1 = M_2 = 1$ al variare di ρ_{N_1, N_2} per salti indipendenti, comuni e self-decomposable.	30
4.2	Valori di $Q_{m,n}$ in funzione di n con $m = 10, 20, 30, 40, 50, 60$. . .	32
4.3	Contour plot delle distribuzioni congiunte	35
5.1	simulazione MC con $a = 0.9$	43

Elenco delle tabelle

4.1	Parametri per GBM e per il processo compound Poisson	28
4.2	Valori dell'opzione spread senza salti e con processo indipendente di Poisson	29
4.3	Valori dell'opzione spread con common e sd compound Poisson . .	31
4.4	Prezzo opzione exchange con simulazioni di Montecarlo	33
4.5	Parametri calibrati per EEX e Powernext	39
4.6	Prezzo opzione exchange Powernext-EEX	40

Sommario

Lo scopo di questa tesi è di replicare e analizzare i risultati numerici e teorici proposti in [Cufaro Petroni and Sabino, 2018], dove viene descritto un nuovo metodo per prezzare opzioni exchange nel mercato energetico, utilizzando processi di Poisson con salti self-decomposable. Prenderemo in considerazione nel nostro studio due differenti sottostanti la cui correlazione fornisce non solo un fattore macro-economico globale, ma anche un fattore di propagazione ritardato degli shock tra i due asset. In particolare, il risultato più importante è mostrare come questo approccio riesca a mettere in evidenza con buona precisione l'istante di salto e non solo la frequenza di essi.

Capitolo 1

Introduzione

Diversi studi hanno evidenziato che la dinamica dei prezzi spot nel mercato energetico sono soggetti a mean reversion, stagionalità e salti (si vede ad esempio in [Cartea and Figueroa, 2005]). In questo lavoro, si prende in considerazione il problema della dipendenza in due dimensioni: considerando un processo di diffusione con l'aggiunta di un compound Poisson bidimensionale. Per modellare la dipendenza in due dimensioni del processo di Poisson si userà la self-decomposability della variabile aleatoria esponenziale (vedi [Petroni and Sabino, 2017]). Infatti, è possibile mostrare(vedi [Petroni and Sabino, 2017]) che date due variabili aleatorie indipendenti $Y, Z \sim \mathcal{E}(\lambda)$ e una variabile aleatoria di Bernoulli $B(1) \sim \mathcal{B}(1, 1 - a)$, dove $a = P(B(1) = 0)$, allora anche la variabile aleatoria X definita come $X = aY + B(1)Z$ ha legge esponenziale $\mathcal{E}(\lambda)$. Quindi X è il risultato di una somma pesata tra Y , con peso deterministico a , e Z , con peso aleatorio $B(1)$. Questo fatto è diretta conseguenza della self-decomposability della legge esponenziale. Come si può facilmente notare, X e Y non sono indipendenti e a rappresenta il coefficiente di correlazione tra loro. Tuttavia, è possibile definire una coppia di esponenziali con parametri diversi $X \sim \mathcal{E}(\lambda)$ $W \sim \mathcal{E}(\mu)$ e correlazione a , semplicemente riformulando l'espressione precedente in questo modo: $X = a\mu W/\lambda + B(1)Z$. A questo punto, considerando le due variabili aleatorie esponenziali definite precedentemente come due tempi aleatori con ritardo positivo $B(1)Z$, il modello può dare alcune risposte a delle comuni domande nel campo

della finanza.

- Se un'istituzione finanziaria dovesse fallire, dopo quanto tempo fallirebbero le altre istituzioni dipendenti?
- Se il mercato dovesse ricevere delle notizie interpretate come uno shock, dopo quanto tempo si avrebbe la reazione del mercato?
- Se diverse compagnie sono connesse tra loro, qual'è l'impatto sul rischio assicurativo?

Queste domande sono coperte nel nostro modello dal caso $a\mu/\lambda > 1$, cioè quando il secondo evento avviene successivamente al primo. Quello che faremo nel successivo capitolo, grazie alla self-decomposability della legge esponenziale, è costruire processi di Poisson bidimensionali con legge marginale dipendente e mostreremo che i due processi sono connessi da una forma di co-integrazione tra i loro salti. Il ritardo di propagazione delle informazioni descritto dal modello è molto simile a quello che succede nei network: in ogni nodo la reazione individuale causa un ritardo nella propagazione dell'informazione al nodo successivo e così via. Date le caratteristiche appena descritte, la chiameremo propagazione sinaptica. In articoli precedenti (vedi ad esempio [Cheang and Chiarella, 2011], [Caldana et al., 2015]) viene preso in considerazione il modello di Merton con due sottostanti, dove gli shock macro-economici sono rappresentati da salti comuni con ampiezza correlata, mentre gli shock indipendenti vengono aggiunti per modellare il rischio idiosincratico. In questo articolo, invece, rappresenteremo i macro-economici shock che si possono propagare con diverse tempistiche in ogni sottostante.

Il prossimo capitolo presenterà: i risultati teorici per costruire la distribuzione bidimensionale di Poisson, determinante per il pricing dell'opzione exchange con questo nuovo approccio, con qualche precisione sulla funzione ipergeometrica confluyente. L'insieme dei modelli utilizzati che sono: Geometric Brownian Motion, il

modello di Ornstein-Uhlenbeck e di Schwartz-Smith, tutti con l'aggiunta di salti sia per la parte macro-economica sia per la parte sinaptica. Il capitolo 3 contiene, invece, le formule chiuse per il pricing di una call europea e dell'opzione exchange. In seguito, avremo i risultati numerici per i casi GBM e GOU (Gaussian Ornstein-Uhlenbeck) dove non verrà presa in considerazione la parte macro-economica per meglio capire le caratteristiche sinaptiche che sono al centro del nostro studio.

Capitolo 2

Presentazione teorica

In questo capitolo, presenteremo i risultati teorici utilizzati per sviluppare i metodi di Pricing per le opzioni exchange.

2.1 Distribuzione congiunta di Poisson

Una legge con densità $f(x)$ e funzione caratteristica $\phi(u)$ è detta self-decomposable (vedi [Sato and Ken-Iti, 1999], [Petroni and Sabino, 2017]) se per ogni $0 < a < 1$ possiamo trovare un'altra densità $g_a(x)$ e una funzione caratteristica $\chi_a(u)$ tale che:

$$\phi(u) = \phi(au)\chi_a(u).$$

Potremmo anche dire che una variabile aleatoria è sd (self-decomposable) se la sua legge è sd: cioè significa che per $0 < a < 1$ possiamo sempre trovare due variabili aleatorie indipendenti Y (con la stessa legge di X) e Z_a , con densità $g_a(x)$ e funzione caratteristica $\chi_a(u)$, tali che in distribuzione valga:

$$X = aY + Z_a.$$

Quindi conoscendo X possiamo trovare due variabili aleatorie in grado di scomporla. Viceversa, se Y è sd e, per $0 < a < 1$, prendiamo una variabile aleatoria Z_a indipendente con legge conosciuta allora possiamo definire X quasi ovunque come:

$$X = aY + Z_a.$$

e saremo sicuri di avere la stessa legge di Y . Il coefficiente a gioca il ruolo di correlazione tra X e Y . Infatti, sfruttando l'indipendenza tra Y e Z_a e definendo $\mathbf{E}[X] = \mathbf{E}[Y] = \mu$, $Var[X] = Var[Y] = \sigma^2$ otteniamo:

$$\mathbf{E}[XY] = \mathbf{E}[(aY + Z_a)Y] = a\sigma^2 + \mu^2.$$

Quindi $r_{XY} = a$.

Come abbiamo già accennato, la legge esponenziale è un tipico esempio di legge sd. Ricordiamo che la legge esponenziale ha densità e funzione caratteristica così definite:

$$\begin{aligned} f_1(x) &= \lambda e^{-\lambda x} \text{ per } x > 0, \\ \phi_1(u) &= \frac{\lambda}{\lambda - iu}. \end{aligned}$$

Con questa legge possiamo trovare una forma esplicita per Z_a , infatti invertendo l'equazione $X = aY + Z_a$ in funzione di Z_a possiamo ricavare la funzione caratteristica $\chi_a(u)$:

$$\chi_a(u) = \frac{\phi_1(u)}{\phi_1(au)} = \frac{\lambda - iau}{\lambda - iu} = a + (1 - a)\frac{\lambda}{\lambda - iu} = a + (1 - a)\phi_1(u).$$

Dove notiamo subito che è una somma di una delta di Dirac in 0 e di una legge esponenziale:

$$Z_a \sim a\delta_0 + (1 - a)\mathcal{E}(\lambda).$$

Da qui, andiamo facilmente a definire densità e funzione caratteristica di Z_a :

$$\begin{aligned} g_a(z) &= a\delta(z) + (1 - a)e^{-\lambda z} \text{ per le } x > 0, \\ G_a(z) &= [a + (1 - a)(1 - e^{-\lambda z})] \text{ per le } x > 0, \end{aligned}$$

le quali, a loro volta, sono prodotto di due leggi indipendenti: un fattore è $Z \sim \mathcal{E}_1(\lambda)$ e l'altro è una Bernoulli $B(1) \sim \mathcal{B}(1, 1 - a)$ con $a = P(B(1) = 0)$. Quindi possiamo riscrivere Z_a come:

$$Z_a = B(1)Z.$$

Possiamo concludere che date due variabili aleatorie esponenziali $Y, Z \sim \mathcal{E}(\lambda)$ e una Bernoulli $B(1) \sim \mathcal{B}(1, 1 - a)$ (tutte tra loro indipendenti), la variabile aleatoria $X = aY + B(1)Z \sim \mathcal{E}(\lambda)$ è ancora una variabile aleatoria esponenziale, con $r_{XY} = a$.

Presi $\lambda_1, \lambda_2 > 0$ e $0 < a < 1$ costruiamo una sequenza di variabili aleatorie dalla definizione $X = aY + B(1)Z$:

$$X_k = aY_k + B_k(1)Z_k \text{ per } k = 1, 2, \dots$$

In modo tale che ogni $X_k, Y_k, Z_k, \sim \mathcal{E}(\lambda_2)$ e $B_k(1) \sim \mathcal{B}(1, 1 - a)$. Partendo dal fatto che $Z_k, Y_k, B_k(1)$ sono mutuamente indipendenti (con $X_0 = Y_0 = Z_0 = 0$), il passo successivo è definire altre due variabili aleatorie indipendenti:

$$\begin{aligned} S_n &= \frac{\lambda_2}{\lambda_1} \sum_{k=0}^n Y_k \sim \mathcal{E}_n(\lambda_1), \\ T_n &= \sum_{k=0}^n X_k \sim \mathcal{E}_n(\lambda_2), \end{aligned}$$

dove $\mathcal{E}_n(\lambda)$ rappresenta la legge Gamma con densità e funzione caratteristica:

$$\begin{aligned} f_n(x) &= \lambda \frac{(\lambda x)^{n-1}}{(n-1)!} e^{-\lambda x} \text{ per le } x \geq 0 \\ \phi_k(u) &= \left(\frac{\lambda}{\lambda - iu}\right)^n \text{ per le } n = 0, 1, 2, \dots \end{aligned}$$

Finalmente, possiamo ora definire i due processi di Poisson dipendenti $N_1(t) \sim \mathcal{P}(\lambda_1 t)$ e $N_2(t) \sim \mathcal{P}(\lambda_2 t)$, associati rispettivamente a S_n e T_n . Come descritto nell'introduzione, abbiamo bisogno della forma esplicita della distribuzione congiunta dei due processi Poisson cioè:

$$\begin{aligned} p_{m,n}(t) &= P\{N_1(t) = m, N_2(t) = n\} \\ &\text{con } n, m = 0, 1, 2, \dots \text{ e } t \geq 0. \end{aligned}$$

Per semplificarci la scrittura, introduciamo la seguente notazione che sostituirà la forma esplicita della densità di Poisson e della Binomiale:

$$\pi_k(\alpha) = e^{-\alpha} \frac{\alpha^k}{k!} \text{ dove } k = 0, 1..$$

$$\beta_l(n) = \binom{n}{l} a^{n-l} (1-a)^l \text{ dove } l \leq n = 0, 1..$$

La forma esplicita della distribuzione congiunta della legge bidimensionale di Poisson (dimostrato in [Petroni and Sabino, 2017]), si definisce in due casi separati: il primo è per $a\lambda_1 \geq \lambda_2$ e ha le seguente forma:

$$p_{m,n}(t) = \begin{cases} 0 & \text{se } n > m \geq 0 \\ Q_{n,n}(t) & \text{se } n = m > 0 \\ Q_{m,n}(t) - Q_{m,n+1}(t) & \text{se } m > n \geq 0 \end{cases}$$

Dove:

$$Q_{m,n}(t) = \sum_{k=n}^m (-1)^k \sum_{j=k}^m \binom{j}{k} \frac{\pi_{m-j}(\lambda_1 t)}{(-a)^j} \sum_{l=0}^n \beta_l(n) \pi_{j+l}(\lambda_2 t) \phi(j+1, j+l+1; \lambda_2(t)).$$

Invece, il secondo caso è per $a\lambda_1 \leq \lambda_2$:

$$p_{m,n}(t) = \begin{cases} A_{m,n}(t) - A_{m,n+1} + B_{m,n}(t) - B_{m,n-1}(t) & \text{se } n > m \geq 0 \\ A_{n,n}(t) - A_{n,n+1} + B_{n,n}(t) + C_{n,n}(t) & \text{se } n = m > 0 \\ A_{m,n}(t) - A_{m,n+1} + C_{m,n}(t) - C_{m,n+1}(t) & \text{se } m > n \geq 0 \end{cases}$$

Sempre per un discorso di semplificazione di scrittura, definiamo $w = \lambda_2 t - a\lambda_1 t$ e vediamo la forma esplicita di ogni componente di $p_{m,n}$:

$$A_{m,n}(t) = \pi_m(\lambda_1 t) \sum_{k=0}^n \beta_k(n) [1 + \pi_k(w) - \sum_{j=0}^k \pi_j(w)] \text{ per ogni } n, m \geq 0$$

$$B_{m,n}(t) = \pi_m(t)(\lambda_1 t) \sum_{k=0}^{m-n} \pi_k\left(\frac{w}{a}\right) \sum_{l=0}^{n+1} \beta_l(n+1) \frac{w^l k!}{(k+l)!} \phi(l, k+l+1, \frac{(1-a)w}{a})$$

per le $n \geq m \geq 0$. Infine:

$$C_{m,n} = \frac{e^{-(1-a)\lambda_1 t}}{a^m} \sum_{l=1}^n \beta_l(n) \sum_{k=n}^m \sum_{j=0}^{l-1} \binom{k+l-j-1}{k} (-1)^{l-1-j} \pi_j(\lambda_2 t) \\ \times \pi_{m+l-j}(a\lambda_1 t) \phi(k+l-j, m+l-j+1; a\lambda_1 t).$$

$\phi(a, b; z)$ per $0 \leq l \leq n \leq j \leq m$ è la funzione Ipergeometrica Confluente (vedi [Gradshteyn and Ryzhik, 2014], [Pearson et al., 2017]) che descriveremo meglio nella prossima sezione.

2.2 Ipergeometrica confluyente

In questa sezione ci occuperemo di approfondire la funzione Ipergeometrica confluyente, facendo riferimento ai risultati ottenuti in [Pearson et al., 2017]. Elencheremo qualche metodo per l'approssimazione numerica, in modo tale da poter calcolare la distribuzione congiunta dei Poisson presentata nella sezione precedente.

La funzione Ipergeometrica confluyente è soluzione dell'equazione differenziale lineare del secondo ordine:

$$z \frac{d^2 w}{dz^2} + (b - z) \frac{dw}{dz} - aw = 0,$$

dove $a, b, z \in \mathbb{C}$.

La soluzione viene chiamata anche funzione ipergeometrica di Kummer ed è definita dalla serie:

$$\phi(a, b, z) = \sum_{j=0}^{+\infty} \frac{(a)_j}{\Gamma(b+j)} \frac{z^j}{j!}, \quad (2.1)$$

dove il simbolo di Pochhammer $(a)_j$ è:

$$(a)_0 = 1, \quad (a)_j = a(a+1) \times \dots \times (a+j-1), \quad \text{per } j = 1, 2, \dots$$

Inoltre, la serie (2.1) è sempre convergente.

Quando b è non negativo, ci è utile definire la funzione ipergeometrica Gaussiana:

$${}_1F_1(a, b, z) = \Gamma(b) \phi(a, b, z) = \sum_{j=0}^{+\infty} \frac{(a)_j}{(b)_j} \frac{z^j}{j!}. \quad (2.2)$$

Spesso da alcuni autori viene denotata come $\phi(a, b, z)$ e chiamata lei stessa funzione confluyente ipergeometrica. La funzione gaussiana ipergeometrica è anche definita nel raggio unitario $|z| < 1$ come:

$$F(a, b, c, z) = \Gamma(b) \phi(a, b, z) = \sum_{j=0}^{+\infty} \frac{(a)_j (b)_j}{\Gamma(c+j)} \frac{z^j}{j!}, \quad (2.3)$$

e soddisfa l'equazione differenziale ordinaria di secondo ordine:

$$z(1-z)\frac{d^2w}{dz^2} + [c - (a+b+1)w]\frac{dw}{dz} - abw = 0. \quad (2.4)$$

Il modo più semplice per simulare la funzione confluyente ipergeometrica è quello di troncatura la serie di Taylor:

$${}_1F_1(a, b, z) \approx S_N = \sum_{j=0}^N \frac{(a)_j z^j}{(b)_j j!}, \quad (2.5)$$

dove $\frac{(a)_j z^j}{(b)_j j!}$ lo chiameremo A_j nei metodi di approssimazione. Un buon criterio per fermare la serie è $\frac{|A_{N+1}|}{|S_N|} < tol$ per qualche tol e qualche N , in alternativa si può controllare che la differenza tra due termini successivi non sia maggiore di tol . Di seguito mostreremo due metodi per l'approssimazione di ${}_1F_1(a, b, z)$:

- **Metodo a)**

Calcoliamo:

$$\begin{aligned} A_0 &= 1, & S_0 &= A_0, \\ A_{j+1} &= A_j \frac{a+j}{b+j} \frac{z}{j+1}, & S_{j+1} &= S_j + A_{j+1}, \quad j = 0, 1, 2.. \end{aligned}$$

- **Metodo b)**

$$\begin{aligned} S_{-1} &= S_0 = 1, & S_1 &= 1 + \frac{a}{b}z, \\ r_j &= \frac{a+j-1}{j(b+j-1)} \frac{z}{j+1}, & S_j &= S_{j-1} + (S_{j-1} - S_{j-2})r_j z, \quad j = 2, 3.. \end{aligned}$$

Un altro modo per approssimare $\phi(a, b, z)$ è quello di calcolarla tramite integrazione numerica, infatti quando $Re(b) > 0$ e $Re(a) > 0$ vale:

$$\phi(a, b, z) = \frac{1}{\Gamma(a)\Gamma(b-a)} \int_0^1 e^{zt}(1-t)^{b-a-1}t^{a-1}dt. \quad (2.6)$$

2.3 Modelli con salti

In questo capitolo andremo ad illustrare diversi modelli stocastici, molto utilizzati in letteratura per modellizzare il comportamento dei prodotti finanziari nel settore energetico.

2.3.1 GBM

Consideriamo una generalizzazione del modello di Merton introdotto in [Cheang and Chiarella, 2011] e [Caldana et al., 2015] con due sottostanti. Nella scelta originale dei due articoli sopra citati, si assumono due tipi di salti: i primi descrivono gli shock macro-economici e gli altri per la componente idiosincratca. In questo lavoro, come già accennato nell'introduzione, sarà proposta una terza componente cioè quella che chiameremo sinaptica.

Seguendo la notazione di [Mark S Joshi, 2003], consideriamo la seguente dinamica per i sottostanti:

$$\begin{aligned}
 S_i(T) = \exp & \left[\log_i S_i(0) + \left(\mu_i - \frac{1}{2} \sigma_i^2 \right) T + \sigma_i W_i(T) + \sum_{n_i=1}^{N_i(T)} \log J_{i,n_i} \right. \\
 & \left. + \sum_{n_i=1}^{P_i(T)} \log Q_{i,p_i} \right] \quad \text{per } i = 1, 2,
 \end{aligned} \tag{2.7}$$

con $dW_1(t)dW_2(t) = \rho^{(W)} dt$, dove $\rho^{(W)}$ rappresenta il coefficiente di correlazione tra i due moti Browniani. I processi di Poisson hanno log-normal jump size:

$$J_i = M_i \exp\left(-\frac{v_i^2}{2} + v_i Z_i\right), \quad i = 1, 2, \tag{2.8}$$

dove $Z_i \sim N(0, 1)$ sono leggi Gaussiane standard con correlazione $Corr(Z_1, Z_2) = \rho^{(D)}$. Analogamente:

$$Q_i = A_i \exp\left(-\frac{\alpha_i^2}{2} + \alpha_i Y_i\right), \quad i = 1, 2, \tag{2.9}$$

dove anche le variabili aleatorie $Y_i \sim N(0, 1)$ sono Gaussiane standard, indipendenti da Z_i . I parametri M_i, A_i, v_i, α_i sono costanti scalari, i processi di Poisson

$N_i(t)$ sono definiti nella sezione 2.1, mentre $P_i(t)$ sono due processi di Poisson indipendenti tra loro e da $N_i(t)$ con intensità ψ_i .

Applichiamo il logaritmo a (2.7), ottenendo:

$$\begin{aligned} \log S_i(T) = & \log_i S_i(0) + \left(\mu_i - \frac{1}{2}\sigma_i^2\right)T + \sigma_i W_i(T) + N_i(T) \log M_i + P_i(T) \log A_i - \frac{v_i^2}{2}N_i(T) + \\ & - \frac{\alpha_i^2}{2}P_i(T) + v_i \sum_{n_i=1}^{N_i(T)} Z_{i,n_i} + \alpha_i \sum_{n_i=1}^{P_i(T)} Y_{i,p_i} \quad i = 1, 2, \end{aligned} \quad (2.10)$$

dove abbiamo riposto particolare attenzione nell'applicare il logaritmo in $\exp(\sum_{n_i=1}^{N_i(T)} \log J_{i,n_i})$ e allo stesso modo in $\exp(\sum_{n_i=1}^{P_i(T)} \log Q_{i,p_i})$. Infatti:

$$\exp\left(\sum_{n_i=1}^{N_i(T)} \log J_{i,n_i}\right) = \exp\left(\sum_{n_i=1}^{N_i(T)} \log(M_i \exp(-\frac{v_i^2}{2} + v_i Z_i))\right) = \exp\left(\sum_{n_i=1}^{N_i(T)} (\log M_i + -\frac{v_i^2}{2} + v_i Z_i)\right).$$

Ora applicando il logaritmo e tenendo conto che $\log M_i$ e $\frac{v_i^2}{2}$ sono costanti, questo implica che la sommatoria è uguale a $N_i(T)$ -volte $\log M_i$ e $\frac{v_i^2}{2}$, torniamo a (2.10)

Possiamo riscrivere (2.10) in quest'altro modo:

$$\begin{aligned} \log S_i(T) = & \log S_i(0) + N_i(T) \log M_i + P_i(T) \log A_i \\ & + \left(\mu_i - \frac{1}{2}\sigma_i^2 - \frac{v_i^2}{2}N_i(T) - \frac{\alpha_i^2}{2}P_i(T)\right)T + \\ & \sqrt{\sigma_i^2 + N_i(T)v_i^2 P_i(T)\alpha_i^2} H_i, \quad i = 1, 2. \end{aligned} \quad (2.11)$$

Fissati $N_i(T) = l_i$ e $P_i(T) = m_i$, ricaviamo $(H_1, H_2) \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho_{l_1, l_2, m_1, m_2}^{(J)} \\ \rho_{l_1, l_2, m_1, m_2}^{(J)} & 1 \end{pmatrix}\right)$.

La forma esplicita di $\rho_{l_1, l_2, m_1, m_2}^{(J)}$ è riportata in Appendice 1.

Per abbreviare, chiameremo $v_{i, l_i, m_i}^{(J)}(T)$ o $(\sigma_{i, l_i, m_i}^{(J)}(T))^2$ terminal variance:

$$\begin{aligned} v_{i, l_i, m_i}^{(J)}(T) = & \left(\sigma_{i, l_i, m_i}^{(J)}(T)\right)^2 = \sigma_i^2 T + l_i v_i^2 + m_i \alpha_i^2 \\ = & v_i^{(C)}(T) + v_{i, l_i, m_i}^{(D)}(T), \end{aligned} \quad (2.12)$$

dove $v_i^{(C)}$ e $v_i^{(D)}$ sono le terminal variance della parte continua e della parte discontinua.

In [Cheang and Chiarella, 2011] è presentato un cambio di misura che permette di tener in conto di tutti i rischi dei salti. Lo scopo principale del nostro articolo è quello di mostrare l'impatto della scelta di tempi dei salti correlati con self-decomposition. Sebbene, la scelta della misura in [Cheang and Chiarella, 2011] sarebbe più curata per la nostra dinamica, abbiamo ugualmente voluto utilizzare la classica misura utilizzata in [Merton, 1976] dove tutti i rischi dei salti non sono considerati nel pricing. La condizione di non arbitraggio imposta in [Mark S Joshi, 2003] è:

$$\mu_i - r = -\lambda_i \mathbf{E}[J_i - 1] - \psi_i \mathbf{E}[Q_i - 1] \quad i = 1, 2. \quad (2.13)$$

Questa verrà impiegata nel capitolo 3 per trovare la formula chiusa dell'opzione exchange.

2.3.2 GOU

Sempre con l'intento di rappresentare mean reversion e salti, consideriamo il seguente processo stocastico che guida il mercato:

$$S_i(t) = F_i(0, t) \exp\{U_i(t) + h(t)\}, \quad (2.14)$$

dove $h(t)$ è una funzione deterministica e $F_i(0, t)$ è la curva dei forward al tempo $t=0$ e:

$$\begin{aligned} U_i(t) &= U_i(0)e^{-k_i t} + \sigma_i \int_0^t e^{-k_i(t-s)} dW_i(s) \\ &+ e^{-k_i t} \left(\sum_{n_i}^{N_i(t)} Z_{i,n_i} + \sum_{p_i=1}^{P_i(t)} Y_{i,p_i} \right) \\ &= U_i^C(t) + U_i^{D_1}(t) + U_i^{D_2}(t). \end{aligned} \quad (2.15)$$

Inoltre, $U_i(t)$ è guidato da un processo di OU con salti, in accordo con la seguente equazione differenziale stocastica:

$$dU_i(t) = -k_i U_i(t) dt + \sigma_i dW_i(t) + e^{-k_i t} (Z_i dN_i(t) + Y_i dP_i(t)); \quad (2.16)$$

k_i rappresenta il tasso di mean reversion e σ_i il coefficiente di diffusione. In contrasto con il caso GBM, abbiamo definito i jump size Z_{i,n_i} che sono riproduzioni indipendenti di $Z_i \sim \mathcal{N}(M_i, v_i^2)$ con $Corr(Z_1, Z_2) = \rho^{(D)}$. Analogamente, Y_{i,p_i} sono copie indipendenti di $Y_i \sim \mathcal{N}(A_i, \alpha_i^2)$ con $Corr(Y_1, Y_2) = 0$ e indipendenti da Z_{i,n_i} . Questa EDS è leggermente diversa da quella presentata in [Cartea and Figueroa, 2005], perché il coefficiente esponenziale di fronte alla componente dei salti, è scelto in modo tale che la soluzione non ha salti random con jump size dipendente dal tempo. Questo implica che il modello preso in considerazione in questo elaborato ha le ampiezze dei salti che decrescono nel tempo guidati dal tasso di mean reversion k_i .

Il mercato discusso dal modello è incompleto perché non è replicabile, quindi il prezzo spot scontato difficilmente si riesce a trovarlo nella misura neutrale al rischio. Da adesso in poi, assumeremo che il modello è nella misura neutrale al

rischio Q . Useremo lo stesso approccio nel lemma 3.1 in [Hambly et al., 2009] per ottenere la funzione $h(t)$ consistente con la curva dei forward.

Per ottenere la condizione di non arbitraggio (vedi [Mark S Joshi, 2003]), si impone $\mathbf{E}[S(T)|\mathcal{F}_t] = F(t, T)$ e per semplicità considereremo $\mathbf{E}[S(T)] = F(0, T)$. Allora si possono calcolare i tre termini $\mathbf{E}\left[e^{U_i^C(t)+U_i^{D1}(t)+U_i^{D2}(t)}\right] = \mathbf{E}\left[e^{U_i^C(t)}\right] + \mathbf{E}\left[e^{U_i^{D1}(t)}\right] + \mathbf{E}\left[e^{U_i^{D2}(t)}\right]$. Sappiamo che per la componente continua di un OU vale:

$$\mathbf{E}\left[e^{U_i^{(C)}}\right] = \exp\left(\mathbf{E}\left[U_i^C(t)\right] - \frac{1}{2}\mathbf{E}\left[U_i^C(t)\right]^2\right) = e^{a_i(t)}. \quad (2.17)$$

con:

$$\begin{aligned} \mathbf{E}\left[U_i^{(C)}\right] &= U(0)e^{-k_i t}, \\ \mathbf{V}\left[U_i^{(C)}\right] &= \frac{\sigma_i^2}{2k_i}(1 - e^{-2k_i t}). \end{aligned} \quad (2.18)$$

Senza perdita di generalità, si assume che $U_i(0) = 0$ per $i = 1, 2$. A questo punto, serve calcolare:

$$\mathbf{E}\left[e^{U_i^{D1}(t)}\right] = \mathbf{E}\left[\exp\left(e^{-k_i t} \sum_{n_i=1}^{N_i(t)} Z_{i,n_i}\right)\right] = e^{b_i(t)}, \quad (2.19)$$

$$\mathbf{E}\left[e^{U_i^{D2}(t)}\right] = \mathbf{E}\left[\exp\left(e^{-k_i t} \sum_{n_i=1}^{P_i(t)} Y_{i,n_i}\right)\right] = e^{c_i(t)}. \quad (2.20)$$

Sapendo che la funzione generatrice di momenti per un processo compound Poisson è:

$$\phi_1(u) = \mathbf{E}\left[\exp\left(u \sum_{n_i=1}^{N_i(t)} Z_{i,n_i}\right)\right] = \exp\{\lambda_i t (\phi_{Z_i(u)-1})\}, \quad (2.21)$$

dove:

$$\phi_{Z_i}(u) = \exp\left(M_i u + \frac{1}{2}v_i^2 u^2\right), \quad (2.22)$$

possiamo facilmente ricavare il valore atteso che cercavamo, usando le relazioni (2.21) e (2.22):

$$\begin{aligned} \mathbf{E} \left[e^{U_i^{D_1}(t)} \right] &= \phi_1(e^{k_i t}) \\ &= \exp\{\lambda_i t (e^{M_i e^{-k_i t} + \frac{1}{2} v_i^2 e^{-2k_i t}} - 1)\} \\ &= e^{b_i(t)}. \end{aligned} \quad (2.23)$$

Quindi:

$$b_i(t) = \lambda_i t \left(e^{e^{-k_i t} (M_i + \frac{1}{2} e^{-k_i t} v_i^2)} - 1 \right). \quad (2.24)$$

Conti analoghi vengono fatti in [Cufaro Petroni and Sabino, 2018] anche per $c_i(t)$:

$$c_i(t) = \psi_i(t) \left(e^{e^{-k_i t} (A_i + \frac{1}{2} e^{-k_i t} \alpha_i^2)} - 1 \right). \quad (2.25)$$

La condizione di non arbitraggio da imporre è $h(t) = -a_i(t) - b_i(t) - c_i(t)$ perché vogliamo:

$$\mathbf{E} [S(T)] = F(0, T). \quad (2.26)$$

Ricordando che la dinamica studiata è:

$$S(T) = F(0, T) \exp(U_i(T) + h(t))$$

e

$$U_i(T) = a_i(t) + b_i(t) + c_i(t),$$

allora è chiaro che per avere (2.26), bisogna imporre $h(t) = -a_i(t) - b_i(t) - c_i(t)$.

Mentre l'equazione per la dinamica spot può essere riscritta:

$$\begin{aligned} \log S_i(t) &= \log F_i(0) - a_i(t) - b_i(t) - c_i(t) + e^{-k_i t} (M_i N_i(t) + A_i P_i(t)) \\ &\quad + \frac{1}{2} e^{-2k_i t} (v_i^2 N_i(t) + \alpha_i^2 P_i(t)) \\ &\quad - \frac{1}{2} (\mathbf{V} [U_i^C(t)] + e^{-2k_i t} (v_i^2 N_i(t) + \alpha_i^2 P_i(t))) \\ &\quad \sqrt{\mathbf{V} [U_i^C(t)] + e^{-2k_i t} (v_i^2 N_i(t) + \alpha_i^2 P_i(t))} H_i. \end{aligned} \quad (2.27)$$

Come fatto nel precedente paragrafo, definiamo la terminal variance fissando $N_i(t) = l_i$ e $P_i(t) = m_i$:

$$\begin{aligned} v_{i,n_i,p_i}^{(J)}(T) &= \mathbf{V} [U_i^C(t)] + e^{-2kit} (v_i^2 l_i(t) + \alpha_i^2 m_i(t)) \\ &= v_i^{(C)}(T) + v_{i,l_i,m_i}^{(D)}. \end{aligned} \quad (2.28)$$

2.3.3 Schwartz-Smith

In questa sezione, prenderemo in analisi il modello di Schwartz–Smith a due fattori come descritto in [Schwartz and Smith, 2000] con l’aggiunta dei salti:

$$\begin{aligned} U_1(t) &= U_1(0)e^{-kt} + \sigma_1 \int_0^t e^{-k(t-s)} dW_1(s) + e^{-kt} \sum_{n_1=1}^{N_1(t)} Z_{1,n_1} \\ U_2(t) &= U_2(0) + \mu t + \sigma_2 W_2(t) + \sum_{n_2=1}^{N_2(t)} Z_{2,n_2} + \sum_{p=1}^{P(t)} Y_p \\ U(t) &= U_1(t) + U_2(t), \end{aligned} \quad (2.29)$$

dove $S(t) = F(0, t)e^{h(t)+U(t)}$, Z_{i,n_i} che sono copie indipendenti di $Z_i \sim \mathcal{N}(M_i, v_i^2)$ con $Corr(Z_1, Z_2) = \rho^{(D)}$. Analogamente, Y_{i,p_i} sono realizzazioni indipendenti di $Y_i \sim \mathcal{N}(A_i, \alpha_i^2)$ con $Corr(Y_1, Y_2) = 0$ e anche indipendente da Z_{i,n_i} .

Prendendo il differenziale di (2.29) e dopo qualche calcolo elementare otteniamo:

$$\begin{aligned} dU(t) &= k(\mu + U_2(t) - U(t)) dt + \sigma dW(t) + \\ &\quad + Z_1 (e^{-kt} dN_1(t) + Z_2 dN_2(t) + Y dP(t)) \end{aligned} \quad (2.30)$$

dove $\sigma^2 = \sigma_1^2 + \sigma_2^2 + 2\sigma_1\sigma_2\rho^{(W)}$.

Esattamente come nella procedura del precedente paragrafo, andiamo a cercare la condizione di non arbitraggio prendendo il valore atteso del processo spot:

$$\begin{aligned} \mathbf{E} [e^{U^C(t)}] &= \mathbf{E} [e^{U_1(0)e^{-kt} + \sigma_1 \int_0^t e^{-k(t-s)} dW_1(s) + e^{-kt} + U_2(0) + \mu t + \sigma_2 W_2(t)}] \\ &= \exp \left(\mathbf{E} [U^C(t)] - \frac{1}{2} \mathbf{V} [U^C(t)] \right) = e^{a(t)}. \end{aligned} \quad (2.31)$$

Assumiamo $U_1(0) = 0$ e $U_2(0) = 0$, abbiamo:

$$\begin{aligned} \mathbf{E} [U^C(t)] &= \mu t, \\ \mathbf{V} [U^C(t)] &= \frac{\sigma_1^2}{2k} (1 - e^{-2kt}) + \sigma_2 t + \frac{2\rho\sigma_1\sigma_2}{k} (1 - e^{-kt}). \end{aligned} \quad (2.32)$$

Dopo aver fatto la trasformazione $\epsilon_1 = Z_1$ $\epsilon_2 = \rho^{(D)} Z_1 + \sqrt{1 - (\rho^{(D)})^2}$ in distribuzione e dopo alcuni passaggi algebrici, possiamo scrivere per la componente discontinua:

$$\begin{aligned}
 e^{b(t)} &= \mathbf{E} \left[e^{U^{D_1(t)}} \right] = \mathbf{E} \left[\exp \left(e^{-kt} \sum_{n_1=1}^{N_1(t)} Z_{1,n_1} + \sum_{n_2=1}^{N_2(t)} Z_{2,n_2} \right) \right] \\
 &= \sum_{l_1, l_2=0}^{+\infty} p_{l_1, l_2} \exp \left(l_1 M_1 e^{-kt} + l_2 M_2 \right) \\
 &\quad \times \exp \left(\left(\sqrt{l_1} v_1 e^{-kt} + \rho^{(D)} \sqrt{l_2} v_2 \right)^2 \right) \\
 &\quad \times \exp \left(l_2 v_2^2 \left(1 - (\rho^{(D)})^2 \right) \right)
 \end{aligned} \tag{2.33}$$

e

$$e^{c(t)} = \mathbf{E} \left[e^{U^{D_2(t)}} \right] = \mathbf{E} \left[\sum_{p=1}^{P(t)} Y_p \right] = \exp \left(\psi t \left(\phi_Y(1) - 1 \right) \right), \tag{2.34}$$

dove $\phi(1) = \exp \left(A + \frac{1}{2} \alpha^2 \right)$ e $c(t) = \psi t \left(\phi_Y(1) - 1 \right)$.

Come abbiamo già visto per il caso GBM, la condizione di non arbitraggio è trovata con $h(t) = -a(t) - b(t) - c(t)$ dove $b(t)$ può essere calcolato numericamente.

Finalmente, tramite passaggi algebrici si può riscrivere la dinamica spot:

$$\begin{aligned}
 \log S_i(t) &= \log F(0, t) - b(t) - c(t) \\
 &\quad + e^{-kt} M_1 N_1(t) + M_2 N_2(t) + A P_t \\
 &\quad + \frac{1}{2} \left[\left(v_1 \sqrt{N_1(t)} e^{-kt} + \rho^{(D)} v_2 \sqrt{N_2(t)} \right)^2 \right. \\
 &\quad \left. + v_2^2 N_2(t) \left(1 - (\rho^{(D)})^2 + \alpha^2 P(t) \right) \right] \\
 &\quad - \frac{1}{2} v_{N_1(t), N_2(t), P(t)}^{(J)} + \sqrt{v_{N_1(t), N_2(t), P(t)}^{(J)}} H,
 \end{aligned} \tag{2.35}$$

dove

$$\begin{aligned}
 v_{N_1(t), N_2(t), P(t)}^{(J)} &= \mathbf{V} \left[U^{(C)}(t) \right] + \left(v_1 \sqrt{N_1(t)} e^{-kt} + \rho^{(D)} v_2 \sqrt{N_2(t)} \right)^2 \\
 &\quad + v_2^2 N_2(t) \left(1 - (\rho^{(D)})^2 \right) + \alpha^2 P(t).
 \end{aligned} \tag{2.36}$$

H è una gaussiana standard.

Capitolo 3

Pricing

In questo capitolo, andremo a mostrare le formule chiuse per prezzare un'opzione call europea con i sottostanti guidati dalle dinamiche mostrate nel capitolo 2. In seguito, grazie a questo primo risultato imposteremo la teoria per il pricing di un'opzione exchange.

3.1 Opzione europea

Per semplificare i conti, andremo a scrivere il prezzo dell'opzione call al tempo zero $c(0)$, in termini della formula di Black and Scholes:

$$c(0) = \mathbf{BS}(P_0, K, r, T, v, q), \quad (3.1)$$

dove P_0, K, r, T, v, q denotano il prezzo al tempo 0 del sottostante, lo strike, il tasso risk-free, la maturity, la terminal variance e i dividendi. Seguendo le procedure mostrate in [Mark S Joshi, 2003], il prezzo dell'opzione sarà ottenuto aggiungendo la terminal variance, i dividendi, il valore iniziale del sottostante e la componente che riguarda il processo di Poisson nella formula di BS.

- **GBM**

Primo passo da fare è riscrivere (3.3) utilizzando (2.13). Cioè vediamo il

tasso r nella condizione di non arbitraggio come $r \equiv \frac{v_i^2}{2}N_i(T) + \frac{\alpha_i^2}{2}P_i(T)$ e:

$$\begin{aligned} \mathbf{E}[J_i - 1] &= \mathbf{E}\left[M_i \exp\left(-\frac{v_i}{2} + v_i Z\right) - 1\right] \\ &= M_i \mathbf{E}\left[\exp\left(-\frac{v_i}{2} + v_i Z\right)\right] - 1 \\ &= M_i \exp\left(-\frac{v_i}{2}\right) \exp\left(\frac{v_i}{2}\right) - 1 \\ &= M_i - 1, \end{aligned} \tag{3.2}$$

dove abbiamo utilizzato che $Z \sim \mathcal{N}(0, 1)$. In maniera analoga si fanno i conti per $\mathbf{E}[Q_i - 1]$.

A questo punto, si può riscrivere la dinamica spot neutrale al rischio del sottostante:

$$\begin{aligned} \log S_i(T) &= \log S_i(0) + N_i(T) \log M_i + P_i(T) \log A_i \\ &\quad \lambda_i (1 - M_i) T + \psi_i (1 - A_i) T \\ &\quad \times \left(r - v_i^{(J, N_i(T))}\right) \sqrt{v_i^{(J, N_i(T))} H_i}. \end{aligned} \tag{3.3}$$

Fissando il valore di $N_i(T)$ notiamo che abbiamo la stessa dinamica di BS con valore iniziale del sottostante:

$$\begin{aligned} S_{1, n_1, p_1}(0) &= S_1(0) M_1^{n_1} A_1^{p_1} \\ &\quad \times \exp[\lambda_1 T (1 - M_1) + \psi_1 T (1 - A_1)], \end{aligned} \tag{3.4}$$

e $v_{1, n_1, p_1}^{(J)}$ è definita nell'equazione (2.12). Finalmente possiamo scrivere la formula chiusa per la call:

$$c(0) = \sum_{n_1, p_1=0}^{+\infty} \pi_{n_1}(\lambda_1 T) \pi_{p_1}(\psi_1 T) BS(S_{1, n_1, p_1}(0), K, r, T, v_{1, n_1, p_1}^{(J)}(T), 0). \tag{3.5}$$

- **GOU**

In maniera analoga al caso GBM, guardando (2.27) andiamo a ridefinire il valore iniziale del sottostante e consideriamo $r = 0$:

$$S_{1, n_1, p_1}(0) = F(0, T) e^{\beta_{1, n_1, p_1}(T)}, \tag{3.6}$$

dove $v_{1,n_1,p_1}^{(J)}$ è definito in (2.28) e:

$$\begin{aligned} \beta_{1,n_1,p_1}(t) &= -b_1(t) - c_1(t) + e^{-k_1 t} \\ &\times \left[n_1 \left(\frac{1}{2} e^{-k_1 t} v_1^2 + M_1 \right) \right. \\ &\left. + p_1 \left(\frac{1}{2} e^{-k_1 t} v_1^2 + M_1 \right) \right]. \end{aligned} \quad (3.7)$$

• **Schwartz – Smith**

Partendo anche qui dalla dinamica del sottostante per il caso Schwartz-Smith (2.35) e utilizzando lo stesso procedimento usato nei casi GBM e GOU, possiamo ottenere una formula semi-chiusa per la call europea:

$$c(0) = \sum_{n_1, n_2, p=0}^{+\infty} p_{n_1, n_2}(T) \pi_p(\psi_T) BS(S_{n_1, n_2, p}(0), K, 0, T, v_{n_1, n_2, p}^{(J)}(T), 0), \quad (3.8)$$

dove $p_{n_1, n_2}(T)$ è descritto nella sezione 2.1,

$$S_{n_1, n_2, p}(0) = F(0, t) e^{\beta_{n_1, n_2, p}(T)} \quad (3.9)$$

$$\begin{aligned} \beta_{n_1, n_2, p}(T) &= -b(t) - c(t) + e^{-kt} M_1 n_1 \\ &+ M_2 n_2 + Ap \\ &\frac{1}{2} \left[\left(v_1 \sqrt{n_1} e^{-kt} + \rho^{(D)} v_2 \sqrt{n_2} \right)^2 \right. \\ &\left. + v_2^2 n_2 \left(1 - (\rho^{(D)})^2 \right) + \alpha^2 p \right] \end{aligned} \quad (3.10)$$

e $v_{n_1, n_2, p}^{(J)}(T)$ è definita in (2.36).

3.2 Opzione exchange

L'applicazione alle opzioni exchange, si rivela come un ottimo campo per comparare il nuovo approccio con self-decomposition con gli altri metodi di salto più classici.

Considereremo per semplicità un'opzione exchange con strike uguale a 0, costruito sui risultati di [Margrabe, 1978]. Come nell'opzione europea fissiamo il valore di $N_1(T) = n_1$, $N_2(T) = n_2$ per riportarci alla forma riconosciuta di Black and Scholes. Facendo questo, abbiamo poi bisogno di aggiungere la formula esplicita della distribuzione di Poisson congiunta, correlate tramite self-decomposition. In modo tale da scorrere tramite la serie tutti i possibili valori possibili n_1 n_2 . Stesso discorso vale per la parte degli shock macro-economici, ma essendo indipendenti tra loro ci bastano le due distribuzioni marginali.

Dunque per i casi GBM e GOU abbiamo che il prezzo è:

$$s(0) = \sum_{n_1, n_2, p_1, p_2=0}^{+\infty} p_{n_1, n_2}(T) \pi_{p_1}(\psi_1 T) \pi_{p_2}(\psi_2 T) \mathbf{BS}(S_{1, n_1, p_1}(0), S_{2, n_2, p_2}(0), 0, T, v_{n_1, n_2, p_1, p_2}^{(M)}(T), 0), \quad (3.11)$$

dove $v_{n_1, n_2, p_1, p_2}^{(M)}(T) = v_{1, n_1, p_1}^{(J)}(T) + v_{2, n_2, p_2}^{(J)}(T) - 2\rho_{n_1, n_2, p_1, p_2}^{(J)}(T) \sqrt{v_{1, n_1, p_1}^{(J)}(T) v_{2, n_2, p_2}^{(J)}(T)}$ è la spread terminal variance. Nella letteratura, sono presenti differenti approssimazioni della formula quando lo strike è diverso da zero (ad esempio in [Kirk and Aron, 1995] e [Caldana et al., 2016]). Anche l'estensione con jump diffusion è riportato in altri articoli come [Caldana et al., 2016] e [Pellegrino, 2016]. Ad esempio il metodo discusso in [Caldana et al., 2016] o [Pellegrino, 2016] è applicabile se la funzione caratteristica del sottostante è conosciuta analiticamente. Questi approcci possono essere usati anche nel modello con self-decomposition ma, andrebbe calcolata una forma semplice per la funzione caratteristica del processo di Poisson bidimensionale self-decomposable.

Come possiamo notare nella formula chiusa per l'opzione exchange abbiamo una

sommatoria infinita, quindi il costo computazionale dipenderà fortemente da come si decide di troncare la serie. Un altro metodo che si può utilizzare per il pricing è Montecarlo e lo andremo a vedere nel prossimo capitolo.

Capitolo 4

Applicazione numerica

In questo capitolo mostreremo le applicazioni numeriche della teoria mostrata precedentemente, facendo comparazioni tra self-decomposition e la comune scelta per i salti (indipendenti o dipendenti) sia per il caso GBM che per il caso GOU.

Per meglio investigare le caratteristiche della self-decomposition, eviteremo di inserire nel nostro modello i salti dettati dalla componente macro-economica. Qui di seguito elenchiamo le formule chiuse per le differenti scelte di correlazione del processo di Poisson:

- **Salti indipendenti :**

$N_1(t)$ e $N_2(t)$ sono processi di Poisson indipendenti e la formula per l'opzione exchange è:

$$s(0) = \sum_{n_1, n_2=0}^{+\infty} \pi_{n_1}(\lambda_1 T) \pi_{n_2}(\lambda_2 T) \times \mathbf{BS}(S_{1, n_1}(0), S_{2, n_2}(0), 0, T, v_{n_1, n_2}^{(M)}(T), 0). \quad (4.1)$$

- **Salti dipendenti :**

$N_i(t) = N(t) + N_i^X$ con $i = 1, 2$, dove $N(t)$, con intensità λ , e N_i^X sono processi di Poisson mutuamente indipendenti.

La formula per l'opzione exchange è:

$$s(0) = \sum_{\substack{n=0 \\ n_1, n_2 \geq n}}^{+\infty} \pi_{n_1-n}(\lambda_1^X T) \pi_{n_2-n}(\lambda_2^X T) \pi_n(\lambda T) \times \mathbf{BS}(S_{1,n_1}(0), S_{2,n_2}(0), 0, T, v_{n_1, n_2}^{(M)}(T), 0). \quad (4.2)$$

• **Correlati tramite Self – decomposibility :**

Infine, prendiamo $N(t)$ con $i = 1, 2$ come descritto nella sezione 3.2, ma senza prendere in considerazione la componente macro-economica:

$$s(0) = \sum_{n_1, n_2=0}^{+\infty} p_{n_1, n_2}(T) \times \mathbf{BS}(S_{1,n_1}(0), S_{2,n_2}(0), 0, T, v_{n_1, n_2}^{(M)}(T), 0), \quad (4.3)$$

dove, le probabilità congiunte $p_{n_1, n_2}(T) = \mathbf{P}\{N_1(T) = n_1, N_2(T) = n_2\}$ sono prese dalla sezione 2.1.

Il payoff delle opzioni illustrate sopra, considerano i valori dei due sottostanti allo stesso tempo T : altri tipi di opzioni exchange possono avere tempi differenti per i due sottostanti, ad esempio $(S_1(T_1) - S_2(T_2))^+$, $T_2 < T_1$. In questo caso, andrebbero riadattate le probabilità congiunte considerando $p_{n_1, n_2}(T) = \mathbf{P}\{N_1(T_1) = n_1, N_2(T_2) = n_2\}$, la formula esplicita si può trovare in [Petroni and Sabino, 2017]. Il caso GBM presenta parametri realistici e verranno usati per mostrare i valori dell'opzione exchange con i tre diversi metodi illustrati in precedenza.

Nel caso GOU, invece, i parametri sono presi da due società presenti nel mercato energetico (vedi [Cufaro Petroni and Sabino, 2018]).

Inoltre, modellando la dipendenza dei processi di Poisson tramite self-decomposable, mostreremo come questo metodo dia più rilevanza al tempo dei salti rispetto alla loro frequenza. In particolare, vedremo casi in cui i prezzi delle opzioni exchange potrebbero sembrare contro intuitivi o addirittura sbagliati, se si ragiona sull'assunzione di pura correlazione lineare.

Per condurre queste analisi, si è deciso di troncare la serie infinita per i termini maggiori di $n_{i, max} = 3\lfloor \lambda_i T \rfloor$.

4.1 Applicazione al caso GBM

In questa sezione, compareremo il valore dell'opzione exchange usando le formule (4.1)-(4.3) al variare della correlazione tra i processi di Poisson. In particolare, assumendo $N_i = N(t) + N_i^X$ si ottiene facilmente $\mathbf{Cov}(N_1(t), N_2(t)) = \mathbf{V}[N(t)] = \lambda t$, mentre la correlazione istantanea è $\mathbf{Corr}(N_1, N_2) = \rho_{N_1(t), N_2(t)} = \frac{\lambda}{\sqrt{\lambda_1 \lambda_2}}$ e possiamo notare che è indipendente dal tempo. Nel caso di salti self-decomposable la correlazione si può ottenere numericamente. Consideriamo due casi:

- $\lambda_1 = \lambda_2 = 20$ quindi nel caso SD per ogni $a < 1$ avremo sempre $a\lambda_1 < \lambda_2$;
- $\lambda_1 = 40, \lambda_2 = 20$ dove per il caso SD possiamo trovare o $a\lambda_1 < \lambda_2$ per $a > 0.5$ o $a\lambda_1 > \lambda_2$ per $a > 0.5$

La scelta dei parametri è definita nella tabella 4.1.

Tabella 4.1: Parametri per GBM e per il processo compound Poisson

Parte continua			
Senza salti			
	Caso A	Caso B	Con salti
$S_1(0)$	100	100	100
$S_2(0)$	100	100	100
σ_1	0.49	0.37	0.2
σ_2	0.35	0.23	0.15
$\rho^{(W)}(\%)$	96	60	80
Parte discontinua			
	Caso A	Caso B	
$\rho^{(D)}(\%)$	99	50	
λ_1	20	40	
λ_2	20	20	
v_1	0.1	0.05	
v_2	0.07	0.04	
M_1	1.1	1.05	
M_2	1.1	1.05	

In entrambi i casi, consideriamo opzioni spread at-the-money con strike uguale a 0, $K = 0$ e $T = 1$ cosicché è possibile usare la formula di Margrabe. Si potrebbero utilizzare anche diverse tecniche di approssimazione per il valore delle opzioni spread con strike diverso da 0, senza invalidare il test.

Invece nella tabella 4.2 si trovano i valori dell'opzione exchange per un GBM con i parametri presenti nella tabella 4.1.

Tabella 4.2: Valori dell'opzione spread senza salti e con processo indipendente di Poisson

	Senza salti		Salti indipendenti	
	Caso A	Caso B	Caso A	Caso B
Prezzo	7.27	11.92	25.23	19.27

I calcoli numerici sono stati implementati su MATLAB R2018a su un computer 64-bit, processore con CPU 2,5GHz e 8GB RAM. La prima osservazione che possiamo fare è che le equazioni (4.1)-(4.3) dipendono separatamente sia dalle probabilità p_{n_1, n_2} sia dal valore della formula di BS. Chiaramente p_{n_1, n_2} non dipende dalla struttura scelta dei jump size, mentre il valore di BS è totalmente indipendente dalla struttura scelta per il processo di Poisson. La figura 5.1 mostra come la scelta dei jump size abbia un impatto fondamentale nel valore dell'opzione, infatti scegliendo il caso $M_1 = M_2 = 1$ il prezzo è indipendente dalla scelta della correlazione tra Poisson.

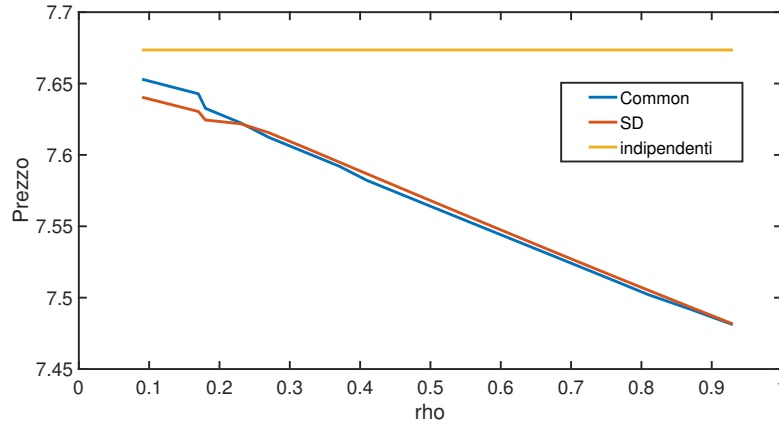


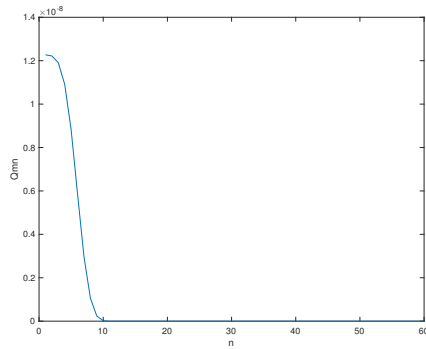
Figura 4.1: Valori dell'opzione con $M_1 = M_2 = 1$ al variare di ρ_{N_1, N_2} per salti indipendenti, comuni e self-decomposable.

In Tabella 4.3 andiamo a mostrare i risultati per i casi common Poisson e self-decomposability per il caso A, l'intensità λ di $N(t)$ è stata scelta in modo tale da far coincidere il valore della correlazione nei due modelli. L'effetto della correlazione nei processi di Poisson è visibile chiaramente, infatti con l'aumento di $\rho_{N_1(t), N_2(t)}$ si presenta una decrescita del valore dell'opzione. Questo comportamento è abbastanza intuitivo, perché la spread terminal variance decresce al crescere di $\rho_{N_1(t), N_2(t)}$ (vedi $v_{n_1, n_2, p_1, p_2}^{(M)}(T) = v_{1, n_1, p_1}^{(J)}(T) + v_{2, n_2, p_2}^{(J)}(T) - 2\rho_{n_1, n_2, p_1, p_2}^{(J)}(T)\sqrt{v_{1, n_1, p_1}^{(J)}(T)v_{2, n_2, p_2}^{(J)}(T)}$).

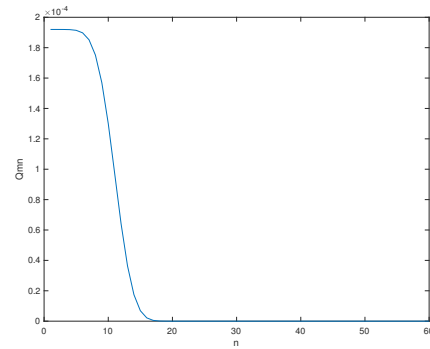
Tabella 4.3: Valori dell'opzione spread con common e sd compound Poisson

a	Caso A		Valore opzione	
	$\rho_{N_1, N_2}(\%)$	λ	Common	SD
0.1	9	1.80	23.9116	23.9054
0.15	14	2.71	23.5993	23.3197
0.2	18	3.63	23.1590	22.7166
0.25	23	4.55	22.5789	22.0945
0.3	27	5.47	22.0799	21.4518
0.35	32	6.40	21.4214	20.7862
0.4	37	7.34	20.7379	20.0956
0.45	41	8.29	20.1723	19.3769
0.5	46	9.24	19.4400	18.6266
0.55	51	10.20	18.6765	17.8404
0.6	56	11.17	17.8778	17.0128
0.65	61	12.16	17.0389	16.1369
0.7	66	13.15	16.1536	15.2031
0.75	71	14.17	15.2139	14.1986
0.8	76	15.20	14.2089	13.1046
0.85	81	16.26	13.1236	11.8922
0.9	87	17.36	11.6842	10.5141
0.95	93	18.53	10.0359	8.904

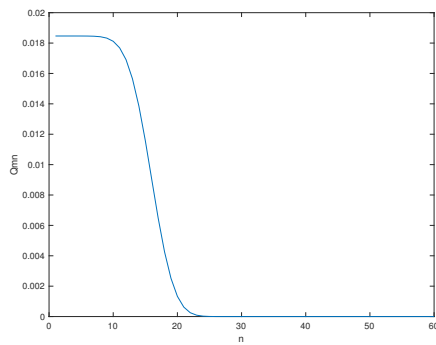
Per quanto riguarda i risultati del caso B, sono stati riscontrati dei grossi problemi numerici nei valori di $Q_{m,n}$, come si può vedere in figura 4.2. I valori di $Q_{m,n}$ sono fortemente dipendenti dal metodo di simulazione della funzione confluyente ipergeometrica; nella figura è stata utilizzata la stessa funzione di Matlab che implementa il metodo B dell'articolo [Pearson et al., 2017], mostrata in appendice A. Aumentando m , aumentano il numero di volte che viene chiamata la funzione confluyente ipergeometrica nella formula di $Q_{m,n}$, essendo approssimata con un metodo iterativo, mostra una forte instabilità al crescere del numero di iterazioni. Altre funzioni sono state testate, ma hanno sofferto della stessa problematica oppure sono risultate eccessivamente onerose dal punto di vista computazionale (con tempi di ore di calcolo per ottenere le funzioni $Q_{m,n}$).



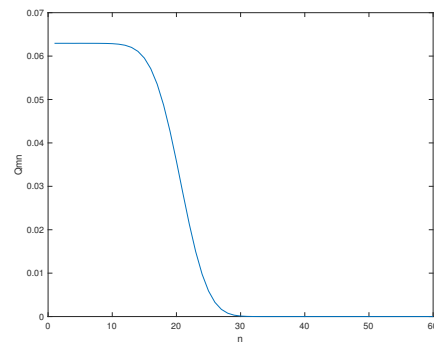
(a) $Q_{m,n}$ con $m = 10$



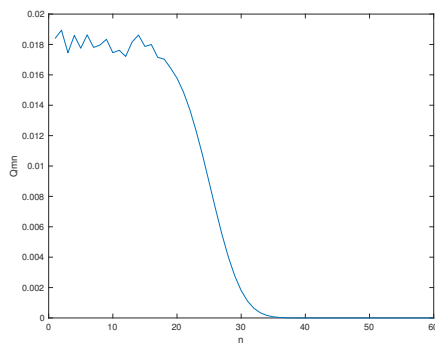
(b) $Q_{m,n}$ con $m = 20$



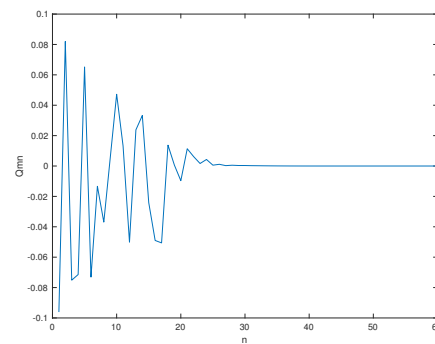
(c) $Q_{m,n}$ con $m = 30$



(d) $Q_{m,n}$ con $m = 40$



(e) $Q_{m,n}$ con $m = 50$



(f) $Q_{m,n}$ con $m = 60$

Figura 4.2: Valori di $Q_{m,n}$ in funzione di n con $m = 10, 20, 30, 40, 50, 60$

I risultati in tabella 4.3 sono stati messi a confronto con il metodo di Monte-carlo. La dinamica simulata è la (2.7) applicando la condizione di non arbitraggio

(2.26) e ottenendo:

$$\begin{aligned}
 S_i(T) = \exp & \left[\log_i S_i(0) + \left(\lambda_i(1 - M_i) - \frac{1}{2}\sigma_i^2 \right) T + \sigma_i W_i(T) \right. \\
 & \left. + \sum_{n_i=1}^{N_i(T)} \log J_{i,n_i} \right] \quad \text{per } i = 1, 2 .
 \end{aligned}
 \tag{4.4}$$

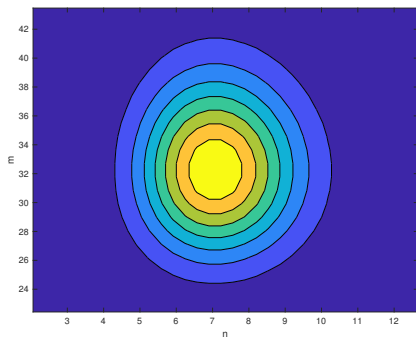
Abbiamo impostato 10^5 simulazioni e la griglia temporale è stata divisa in 300 step. I risultati in Tabella 4.4, sono coerenti con quelli ottenuti con la formula chiusa. I vantaggi di usare una simulazione Montecarlo sono principalmente sulla semplicità di implementazione per qualsiasi tipo di Payoff. In questo caso abbiamo anche il grosso vantaggio, rispetto alla formulazione chiusa, che il costo computazionale è indipendente dalla scelta delle intensità λ_1 e λ_2 .

Tabella 4.4: Prezzo opzione exchange con simulazioni di Montecarlo

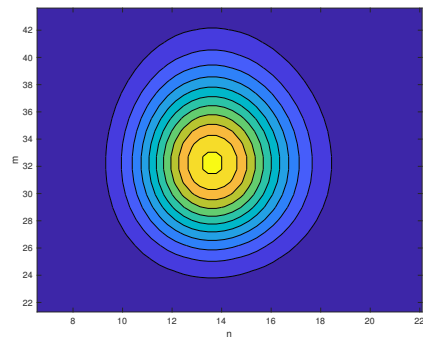
Valore in EUR		
a	prezzo	IC
0.1	24.3890	[24.0420,24.7359]
0.2	23.3921	[23.0616,23.7225]
0.3	22.0987	[21.7790,22.4184]
0.4	20.6680	[20.3740,20.9621]
0.5	19.5519	[19.2617,19.8422]
0.6	17.5917	[17.3355,17.8479]
0.7	16.2866	[16.0390,16.5341]
0.8	14.2609	[14.0341,14.4876]
0.9	11.8324	[11.6494,12.0154]

Un'ulteriore analisi è stata fatta sulle probabilità congiunte dei processi di Poisson: nella figura 4.3 possiamo vedere il contour plot per tutti e tre i casi di correlazione. Per i salti common e self-decomposable è stata scelta $a = 0.35$. Senza correlazione, la distribuzione di probabilità tende giustamente ad assumere una forma ad ellisse. Quel che possiamo notare invece tra i common e le self-decomposability, è che per le seconde c'è più concentrazione di probabilità attorno al valore atteso (ricordiamoci che nel caso a $\lambda_1 = \lambda_2 = 20$ e nel caso B $\lambda_1 = 40$

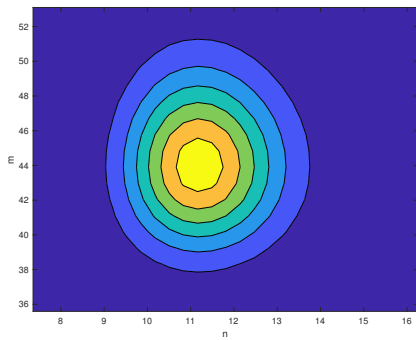
$\lambda_2 = 20$) e quando la differenza tra m e n è piccola. La scelta di salti common, inoltre, riduce la possibilità di salti, infatti abbiamo uno spettro più ristretto.



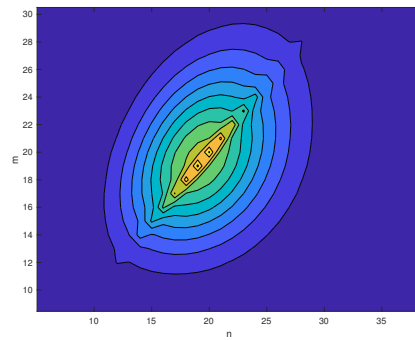
(a) *Salti indipendenti caso A*



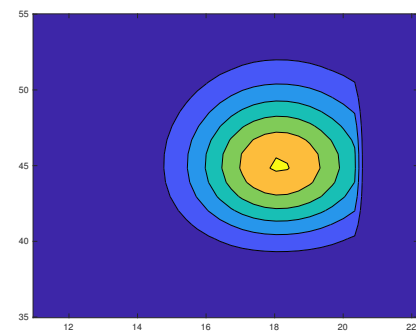
(b) *Salti indipendenti caso B*



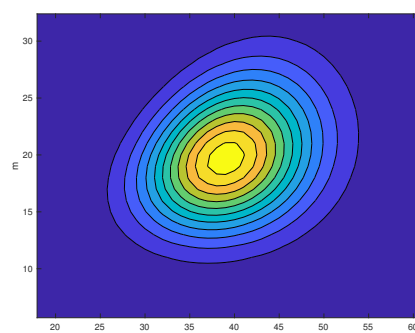
(c) *Common caso $a = 0.35$ caso A*



(d) *Sd caso $a = 0.8$ caso A*



(e) *Common con $a = 0.35$ caso B*



(f) *Sd caso $a = 0.35$ caso B*

Figura 4.3: Contour plot delle distribuzioni congiunte

4.2 Applicazione al caso GOU

In questa sezione, andremo ad applicare (4.1)-(4.3) per il pricing dell'opzione exchange tra EEX e Powernext (Powernext meno EEX). A questo fine, si è assunto che entrambe le società abbiano una dinamica Ornstein-Uhlenbeck più compound Poisson; esattamente come nel caso GBM, si è supposto che non ci siano costi di trasporto ($K = 0$). Il tempo $T = 0$ corrisponde alla fine di Dicembre 2015 e ci siamo concentrati sui primi due trimestri del 2016. La scelta temporale è dettata dal fatto che i modelli spot vengono utilizzati comunemente per periodi corti, a differenza dei modelli future.

In seguito verranno riportate le procedure utilizzate in [Cufaro Petroni and Sabino, 2018] per la calibrazione dei parametri del modello. Sebbene per la formula di pricing sia stato utilizzata una misura neutrale al rischio, c'è prima bisogno di una specificazione. Nella letteratura (come in [Lucia and Schwartz, 2002]) viene proposto di utilizzare l'assunzione di stimare i parametri direttamente nella misura di probabilità reale e poi di incorporare il market price of risk nel drift della dinamica. Non essendo la calibrazione dei parametri il principale obiettivo di questo elaborato, si è deciso di evitare di considerare il market price of risk. Come conseguenza la calibrazione si divide in due step: il primo, è la stima dei parametri con una sola dimensione $\theta_i = (k_i, \sigma_i, \lambda_i, M_i, v_i)$. Il secondo, riguarda i parametri bidimensionali.

Considerando una griglia temporale equi-distanziata $t_0, t_1, ..t_N$ con $t_{i+1} - t_i = \Delta t$ e il seguente schema di Eulero per la dinamica OU:

$$U_i(t + 1) = (1 - k_i \Delta t)U_i(t) + \sigma_i \sqrt{\Delta t} \epsilon_{i,t+1} + e^{-k_i t} \mathbf{1}_i(t + 1) Y_i, \quad (4.5)$$

dove:

$$\mathbf{1}_i(t + 1) = \begin{cases} 1 & \text{con probabilità } \lambda_i \Delta t \\ 0 & \text{con probabilità } 1 - \lambda_i \Delta t. \end{cases} \quad (4.6)$$

La densità di transizione può essere vista come combinazione di densità Gaussiane:

$$\begin{aligned}
 p_i(U_i(t+1), t+1|U_i(t), t) &= (1 - \lambda_i \Delta t) \mathcal{N}(\mu_i^C(t), \sigma_i^C(t)) \\
 &\quad + \lambda_i \Delta t \mathcal{N}(\mu_i^J(t), \sigma_i^J(t)),
 \end{aligned} \tag{4.7}$$

di cui:

$$\begin{aligned}
 \mu_i^C(t) &= (1 - k_i \Delta t) U_i(t), \\
 \mu_i^J(t) &= (1 - k_i \Delta t) U_i(t) + M_i e^{-k_i t}, \\
 \sigma_i^C(t) &= \sigma_i \sqrt{\Delta t}, \\
 (\sigma_i^J(t))^2 &= \sigma_i^2 \Delta t + e^{-2k_i t} v_i^2.
 \end{aligned}$$

I parametri $\theta_i = (k_i, \sigma_i, \lambda_i, M_i, v_i)$ possono essere calibrati massimizzando la funzione log-likelihood

$$\theta_i = \operatorname{argmax} \sum_{t=0}^{T-1} \log(p_i(U_i(t+1), t+1|U_i(t), t)). \tag{4.8}$$

La calibrazione dei parametri bidimensionali dipende da ciò che abbiamo mostrato nella sezione 2.1.

- **Salti indipendenti** : la distribuzione congiunta è semplicemente:

$$\begin{aligned}
 p_{0,0} &= (1 - \lambda_1 \Delta t)(1 - \lambda_2 \Delta t), \\
 p_{1,0} &= (\lambda_1 \Delta t)(1 - \lambda_2 \Delta t), \\
 p_{0,1} &= (1 - \lambda_1 \Delta t)(\lambda_2 \Delta t), \\
 p_{1,1} &= (\lambda_1 \Delta t)(\lambda_2 \Delta t).
 \end{aligned} \tag{4.9}$$

Gli ultimi due parametri rimanenti sono $\rho^{(J)}$ e $\rho^{(W)}$ che possono essere stimati massimizzando la log-likelihood del processo bidimensionale. La probabilità di transizione è:

$$\begin{aligned}
 p_i(U_i(t+1), t+1|U_i(t), t) &= \mathcal{N}(\mu^{CC}(t), \Sigma^{CC}(t)) p_{0,0} \\
 &\quad + \mathcal{N}(\mu^{CJ}(t), \Sigma^{CJ}(t)) p_{0,1} \\
 &\quad + \mathcal{N}(\mu^{JC}(t), \Sigma^{JC}(t)) p_{1,0} \\
 &\quad + \mathcal{N}(\mu^{JJ}(t), \Sigma^{JJ}(t)) p_{1,1},
 \end{aligned} \tag{4.10}$$

dove:

$$\begin{aligned}
 \mu^{CC}(t) &= (\mu_1^C, \mu_2^C), \\
 \Sigma^{CC}(t) &= \begin{pmatrix} (\sigma_1^C)^2 & \rho^{(W)} \sigma_1^C \sigma_2^C \\ \rho^{(W)} \sigma_1^C \sigma_2^C & (\sigma_2^C)^2 \end{pmatrix}, \\
 \mu^{CJ}(t) &= (\mu_1^C, \mu_2^J), \\
 \Sigma^{CJ}(t) &= \begin{pmatrix} (\sigma_1^C)^2 & \rho^{(W)} \sigma_1^C \sigma_2^J \\ \rho^{(W)} \sigma_1^C \sigma_2^J & (\sigma_2^J)^2 \end{pmatrix}, \\
 \mu^{JC}(t) &= (\mu_1^J, \mu_2^C), \\
 \Sigma^{JC}(t) &= \begin{pmatrix} (\sigma_1^J)^2 & \rho^{(W)} \sigma_1^J \sigma_2^C \\ \rho^{(W)} \sigma_1^J \sigma_2^C & (\sigma_2^C)^2 \end{pmatrix}, \\
 \mu^{JJ}(t) &= (\mu_1^J, \mu_2^J), \\
 \Sigma^{JJ}(t) &= \begin{pmatrix} (\sigma_1^J)^2 & \rho^{(W)} \sigma_1^J \sigma_2^J \\ \rho^{(W)} \sigma_1^J \sigma_2^J & (\sigma_2^J)^2 \end{pmatrix}.
 \end{aligned}$$

Non è stato tolto $o(\Delta t^2)$ perché c'è la necessità di stimare $\rho^{(D)}$.

- **Correlazione tramite salti comuni:** Non è possibile ricercare la presenza del processo di Poisson con correlazione comune, guardando singolarmente i logaritmi dei processi in maniera indipendente. Dopo qualche conto algebrico, si trova che $\mathbf{1}_i(t+1)$ è una variabile aleatoria di Bernoulli bidimensionale con:

$$\begin{aligned}
 p_{0,0} &= 1 - (\lambda_1^X + \lambda_2^X + \lambda) \Delta t, \\
 p_{1,0} &= \lambda_1^X \Delta t, \\
 p_{0,1} &= \lambda_2^X \Delta t, \\
 p_{1,1} &= \lambda \Delta t = 1 - p_{0,0} - p_{1,0} - p_{0,1}.
 \end{aligned} \tag{4.11}$$

In questo caso si è deciso di eliminare i termini $o(\Delta t^2)$, nel senso che si sta togliendo la possibilità che N_1^X e N_2^X possano saltare simultaneamente nell'unità temporale Δt . La forma della densità di transizione è simile a (4.10) ma con pesi differenti.

- **Correlazione tramite self – decomposition**

– Caso $a\lambda_1 > \lambda_2$: dalla sezione 2.1, fino ai termini $o(\Delta t^2)$ abbiamo

$$\begin{aligned} p_{0,0} &= 1 - \lambda_1 \Delta t, \\ p_{1,0} &= (\lambda_1 - \lambda_2) \Delta t - \lambda_1 \lambda_2 \frac{1-a}{a} \Delta t^2, \\ p_{0,1} &= 0, \\ p_{1,1} &= a \lambda_1 \Delta t. \end{aligned} \tag{4.12}$$

– Caso $a\lambda_1 \geq \lambda_2$: in maniera simile, eliminando i termini $o(\Delta t^2)$ (vedi appendice B) troviamo:

$$\begin{aligned} p_{0,0} &= 1 - ((1+a)\lambda_1 + \lambda_2) \Delta t, \\ p_{1,0} &= (\lambda_2 - a\lambda_1) \Delta t, \\ p_{0,1} &= (1-a)\lambda_1 \Delta t, \\ p_{1,1} &= a\lambda_1 \Delta t. \end{aligned} \tag{4.13}$$

I risultati della calibrazione sono visibili in tabella 4.5. Entrambi gli expected jump sizes sono negativi e la loro correlazione è molto piccola ($\rho^{(D)} \approx 0$): come conseguenza si è deciso di annullarli.

Tabella 4.5: Parametri calibrati per EEX e Powernext

Market	k	σ_i	μ_i	v_i	λ_i
EEX	42.50(3.66)	1.66(0.028)	-0.10(0.018)	0.16(0.0069)	95.32(16.74)
Powernext	41.64(2.97)	2.71(0.013)	-0.06(0.019)	0.38(0.033)	56.74(8.01)
Metodo	$\rho^{(W)}(\%)$	λ	a		
Indipendenti	43(5.5)	NA	NA		
Comuni	43(5.5)	34.12(6.12)	NA		
SD	43(5.5)	NA	0.44(0.09)		

In tabella 4.6 sono presenti i valori della power-spread nei primi due quarti Q1 e Q2 del 2016 e dettagliatamente nei tre mesi di Q1. Le formule utilizzate, come nel caso GBM, sono le equazioni (4.1)-(4.3).

Tabella 4.6: Prezzo opzione exchange Powernext-EEX

Valore in EUR					
Maturity	$F_1(0, T)$	$F_2(0, T)$	Indipendenti	Comuni	SD
Gennaio	30	6	23.9969	23.9344	19.0555
Febbraio	32	8	24.0000	23.9993	22.5013
Marzo	34	9	25.0000	25.0000	24.5819
Giugno	40	12	28.0000	28.0000	30.3199

Per salti indipendenti e comuni abbiamo praticamente gli stessi prezzi, mentre l'approccio self-decomposable presenta valori minori per le prime tre maturity e maggiore per l'ultima. Potremmo spiegare questa differenza considerando che per il caso common, abbiamo delle restrizioni nelle casistiche dei salti, infatti se l'informazione data dal mercato è associata a N_1^X o N_2^X causa un salto solamente per uno dei due sottostanti. Invece, con il nuovo approccio un'informazione ha una propagazione in entrambe le dinamiche.

Capitolo 5

Conclusioni

Utilizzando il concetto di self-decomposability introdotto in [Petroni and Sabino, 2017], abbiamo studiato l'uso di due processi di Poisson correlati per sviluppare modelli nel mercato energetico, in particolare per prezzare le opzioni exchange. Proprio grazie questo nuovo concetto possiamo pensare che i tempi dei salti dei due processi abbiano un ritardo dato dalla propagazione dell'informazione. Nella figura 5.1 abbiamo voluto dare una rappresentazione grafica a questa idea, infatti si può vedere, in una simulazione fatta con il metodo di Montecarlo, come le dinamiche dei due sottostanti siano correlate tramite i salti dei processi di Poisson.

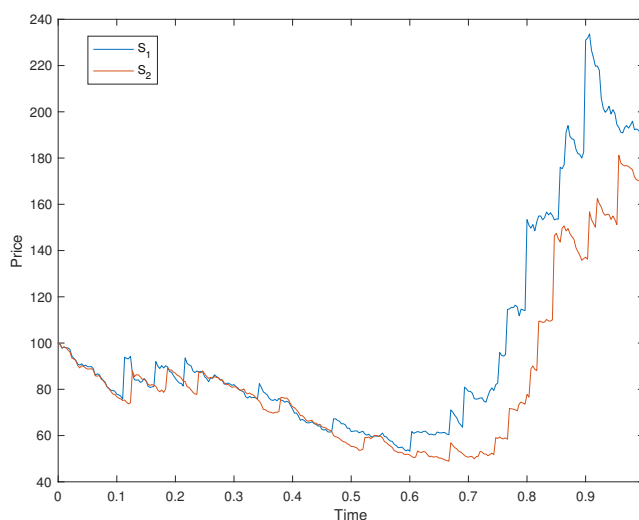


Figura 5.1: simulazione MC con $a = 0.9$

In contrasto con ciò che è descritto in [Cheang and Chiarella, 2011] dove gli shock macro-economici del sistema sono modellati tramite tempi con correlazione comune e gli shock indipendenti sono usati per la parte idiosincratca, in [Cufaro Petroni and Sabino, 2018] si è scelto di assumere che gli shock macro-economici possano avere un impatto in ogni sottostante e che la propagazione possa avere un ritardo correlato. I risultati numerici trovati tramite formula chiusa sono stati in seguito comparati con i risultati delle simulazioni di Montecarlo e abbiamo visto che sono coerenti tra loro. Il grosso vantaggio del metodo di Montecarlo è che è indipendente dalla scelta delle intensità dei due Poisson e che si può utilizzare per qualsiasi tipo di payoff, quindi anche con strike diverso da 0, mentre le formule chiuse andrebbero riviste. Il punto saliente per la computazione delle formule chiuse è senz'altro la simulazione della funzione confluyente ipergeometrica, una volta trovata una simulazione veloce di questa funzione si otterrebbero risultati con più precisione e velocità rispetto al metodo di Montecarlo. Il caso GOU è facilmente migliorabile, una prima ipotesi che si potrebbe rilassare è che l'ampiezza dei salti decresca esponenzialmente nel tempo.

Nella valutazione dei risultati numerici bisogna tenere in mente che usando la self-

decomposability non si ha una comune correlazione lineare. Come già accennato nell'introduzione altri possibili campi di applicazione di questo nuovo concetto, potrebbero essere il settore creditizio e assicurativo, dove il tempo degli eventi è fondamentale per avere dei modelli vicini alla realtà e questo approccio è più preciso di una comune correlazione per l'approssimazione dei tempi di salto. Un ulteriore passo si potrebbe compiere trovando la funzione caratteristica congiunta dei due Poisson per self-decomposability, cosicché da poter utilizzare metodi come FFT (Fast Fourier Transform) per il pricing delle opzioni.

Appendices

Appendice A

Codici matlab

Le prime tre funzioni di Matlab allegate saranno (4.1)-(4.3):

```
1 function [price]=IndjumpGBM(n1max,n2max,lambda1,lambda2,T...
   ,S10,S20,sigma1,sigma2,M1,M2,rhoP,rhoBM,v1,v2)
3 %funzione che calcola il prezzo di un'opzione exchange senza ...
   correlazione tra i due Poisson.
price=0;%inizializzo per la sommatoria
5 for i=0:n1max %sommatorie troncate da n1max=fix(3*lambda1*T)
   for j=0:n2max
7     var1=sigma1^2*T+v1^2*i;
   var2=sigma2^2*T+v2^2*j;
9     rho=(rhoBM*sigma1*sigma2*T+rhoP*sqrt(i*j)*v1*v2)...
       /(sqrt(var1*var2));
11    price=price+PiN(i,lambda1*T)*PiN(j,lambda2*T)...
       *BSmargrabe(0,0,S10,S20,var1,var2,T,rho,M1,M2,i,j...
13    ,lambda1,lambda2);
       %probabilità delle due marginali per la formula di ...
       B&S rivista
15    %PiN è la distribuzione di Poisson
end
17 end
end
```

```

function [price]=CommjumpGBM(nmax,nlmax,n2max,lambda1,lambda2...
2     ,lambda,T,S10,S20,sigma1,sigma2,M1,M2,rhoBM,rhoP,v1,v2)
%funzione che calcola il prezzo di un'opzione exchange con ...
    correlazione comune tra i due Poisson.
4 price=0; %inizializzo per la sommatoria
for i=0:nlmax %sommatorie troncate da nlmax=fix(3*lambda1*T)
6     for j=0:n2max %stesso discorso di nlmax
            for k=0:nmax
8                 if (i>=k && j>=k) %controllo che siano maggiori
                    var1=sigma1^2*T+v1^2*(i);
10                   var2=sigma2^2*T+v2^2*(j);
                    rho=(rhoBM*sigma1*sigma2*T+rhoP*sqrt((i)*(j))*v1*v2)...
12                       /sqrt(var1*var2);
                    price=price+PiN((i-k),(lambda1-lambda)*T)...
14                       *PiN((j-k),(lambda2-lambda)*T)*PiN(k,lambda*T)...
                    *BSmargrabe(0,0,S10,S20,var1,var2,T,rho...
16                       ,M1,M2,i,j,lambda1,lambda2);
            end
18         end
    end
20 end
end

```

```

function [price] =SD_GBM(n1max,n2max,lambda1,lambda2,T,S10,S20...
2     ,sigma1,sigma2,M1,M2,rhoBM,rhoP,v1,v2,w,a)
%funzione per calcolare il prezzo di un'opzione exchange, con ...
    correlazione
4 %tra i due Poisson processes self-decomposable.
%in input riceve i parametri del modello presenti in tabella 4.1
6 price=0; %inizializzo il prezzo per la sommatoria
for i=0:n1max %sommatorie troncate da n1max=fix(3*lambda1*T)
8     for j=0:min(i,n2max) %n2max come n1max, scelgo il max tra ...
        i e n2max perché avremmo probabilità nulle se supera il max
        var1=sigma1^2*T+v1^2*i;
10        var2=sigma2^2*T+v2^2*j;
        rho=(rhoBM*sigma1*sigma2*T+rhoP*sqrt(i*j)*v1*v2)...
12            / (sqrt(var1*var2)); %appendice B
        price=price+joint_poisson(a,lambda1,lambda2,i,j,w,T)*...
14            BSmargrabe2(0,0,S10,S20,var1,var2,...
            T,rho,M1,M2,i,j,lambda1,lambda2); %distribuzione ...
            congiunta per la formula rivista di B&S
16     end
    end
18 end

```

dove *joint poisson*, $Q_{m,n}$, $A_{m,n}$, $B_{m,n}$, $C_{m,n}$ e *ConfHyper* sono rispettivamente:

```

function Pmn= joint_poisson(a,lambda1,lambda2,m,n,w,T)
2 %funzione che calcola le probabilità congiunte dei due Poisson ...
    processes.
if (m>=0 && n>=0)
4     if (a*lambda1>lambda2)
        if (n>m)
6             Pmn=0;

8         elseif (m==n)
            Pmn=max(Qmn(n,n,a,lambda1,lambda2,T),0);
10
        elseif (m>n)
12            Pmn=max(Qmn(m,n,a,lambda1,lambda2,T)-...
                Qmn(m,n+1,a,lambda1,lambda2,T),0);
14        end
    else
16        if (n>m)
            Pmn=max(Amn(m,n,a,lambda1,T,w)-Amn(m,n+1,a,lambda1,T,w)...
18                +Bmn(m,n,a,lambda1,T,w)-Bmn(m,n-1,a,lambda1,T,w),0);

20        elseif (m==n)
            Pmn=max(Amn(m,n,a,lambda1,T,w)-Amn(m,n+1,a,lambda1,T,w)...
22                +Bmn(m,n,a,lambda1,T,w)+Cmn(n,n,lambda1,T,a,lambda2),0)

```



```

24     elseif (m>n)
        Pmn=max (Amn (m, n, a, lambda1, T, w) -Amn (m, n+1, a, lambda1, T, w) ...
26             +Cmn (m, n, lambda1, T, a, lambda2) -...
                Cmn (m, n+1, lambda1, T, a, lambda2) , 0);
28
        end
30 end
    end
32 end

```

```

function qmn = Qmn(m,n,T,lambda1,lambda2,a)
2 %funzione che serve per calcolare le probabilità congiunte dei ...
  due Poisson
  %processes.
4 qmn=0; %inizializzo per la sommatoria
  secondo=0;
6 terzo=0;
  if (n==0 && m==0)
8     qmn=PiN(0,lambda1*T);
  elseif (n==0)
10    qmn=PiN(m,lambda1*T);
  elseif (n>0 && m>0)
12    for k=n:m
        primo=(-1)^k;
14        for j=k:m
            for l=0:n
16                terzo=terzo+Beta(l,n,a)*PiN(j+1,lambda2*T) ...
                    *(ConfHyper(j+1,j+1+1,lambda2*T,10^-15));
18            end
            secondo=secondo+fattoriale(j)/fattoriale(k) ...
20                /fattoriale(j-k)*PiN(m-j,lambda1*T) ...
                    /(-a)^j*terzo;
22        terzo=0;
        end
24    qmn=qmn+primo*secondo;
        secondo=0;
26    end
  end
28 end

```

```

function amn = Amn(m,n,a,lambda1,T,w)
2 amn=0;
  Sum=0;
4   if (n>=0 && m>=0)
        for k=0:n
6           for l=0:k

```

```

Sum=Sum+PiN(l,w);
8         end
        amn=amn+Beta(k,n,a)*(1+PiN(k,w)-Sum);
10        Sum=0;
        end
12    end
    amn=amn*PiN(m,lambda1*T);
14 end

```

```

function bmn = Bmn(m,n,a,lambda1,T,w)
2 bmn=0;
  secondo=0;
4 if (n>=m && m>0 && n>0)
    for k=0:n-m
6        primo=PiN(k,w/a);
        for l=0:n+1
8            secondo=secondo+Beta(l,n+1,a)*w^(l)*fattoriale(k)...
                /fattoriale(k+1)*...
10           ConfHyper(l,k+1+1,(1-a)/a*w,10^-15);
        end
12        bmn=bmn+primo*secondo;
        secondo=0;
14    end
    bmn=PiN(m,lambda1*T)*bmn;
16 end
end

```

```

1 function cmn = Cmn(m,n,lambda1,T,a,lambda2)
  if (m*n==0)
3      cmn=0;
  elseif (m>=n && m>=1 && n>=1)
5      cmn=0;
      secondo=0;
7      for l=1:n+1
          primo=Beta(l,n,a); %distribuzione Binomiale
9          for k=n:m
              for j=0:l-1
11                 secondo=secondo+fattoriale(k+1-j-1)...
                    /fattoriale(k)/fattoriale(l-j-1)...
13                 *(-1)^(l-1-j)*PiN(j,lambda2*T)...
                    *PiN(m+1-j,a*lambda1*T)...
15                 *ConfHyper(k+1-j,m+1-j+1,a*lambda1*T,10^-15);
              end
17          end
          cmn=cmn+primo*secondo;
19          secondo=0;

```

```
    end
21 cmn=cmn*exp(-(1-a)*lambda1*T)/a^m;
    end
23 end
```

```
1 function y = ConfHyper(a,b,z,tol)
   %funzione che simula con metodo iterativo la funzione ...
   %confluente ipergeometrica.
3 %metodo B della sezione 2.2
   S=zeros(500,1);
5 r=zeros(500,1);
   S(1)=1;
7 S(2)=1;
   S(3)=1+a/b*z;
9 y=0;
   for i=4:503
11     r(i)=(a+i-3)/(i-2)/(b+i-3);
       S(i)=S(i-1)+(S(i-1)-S(i-2))*r(i)*z;
13     if (abs(S(i)-S(i-1))/abs(S(i-1))<tol && ...
          abs(S(i-1)-S(i-2))/abs(S(i-2))>tol )
       y=S(i-1);
15     break
       end
17 end
   if (y==0)
19     y=S(end);
   end
```

Ora mostreremo il codice di Matlab utilizzato per simulare due sottostanti con dinamica GBM più jump diffusion, con correlazione self-decomposable tra i due processi di Poisson.

```

function [S1,S2,JumpTimes1,JumpTimes2]=MC_SD(T,Nsim,Nsteps,...
2 lambda1,lambda2,sigma1,sigma2,S01,S02,M1,M2,v1,v2,rhoBM,rhoD,a)
%funziona che genera il metodo di Montecarlo per processi GBM ...
% con jump
4 %diffusion correlati tramite self-decomposability
%>> time grid
6 t=linspace(0,T,Nsteps+1); dt=T/Nsteps;
% risk neutral
8 mu1=log(M1)-v1^2/2; delta1=v1;
psi1=@(u) -sigma1^2*u.^2/2+lambda1*(exp(-delta1^2*u.^2/2+1i*mu1*u)-1);
10 gamma1=-psi1(-1i);
mu2=log(M2)-v2^2/2; delta2=v2;
12 psi2=@(u) -sigma2^2*u.^2/2+lambda2*(exp(-delta2^2*u.^2/2+1i*mu2*u)-1);
gamma2=-psi2(-1i);
14 %inizializzo
X1=zeros(Nsim, Nsteps+1);
16 X2=zeros(Nsim, Nsteps+1);
for i=1:Nsim %simulazioni degli scenari
18 JumpTimes1=0; count1=0; NT1=0;
JumpTimes2=0; count2=0; NT2=0;
20 while (JumpTimes1(end)<=T) %riempio il vettore dei tempi ...
% dei salti finchè uno dei due processi non raggiunge T
count1=count1+1;
22 count2=count2+1;
E2=icdf('exponential',rand,1/lambda2);
24 Be=binornd(1,1-a);
W=icdf('exponential',rand,1/lambda2);
26 E1=a*E2+Be*W;
JumpTimes1(count1+1)=JumpTimes1(count1)+E1;
28 JumpTimes2(count2+1)=JumpTimes2(count2)+E2;
end %cerco quale dei due processi non è pieno e in ...
% seguito lo riempio
30 JumpTimes1=JumpTimes1(2:end-1);
NT1=size(JumpTimes1,2);
32 if (JumpTimes2(end)>T)
x=find(JumpTimes2>1);
34 JumpTimes2=JumpTimes2(2:x(1)-1);
NT2=size(JumpTimes2,2);
36 else
while (JumpTimes2(end)<=T)
38 count2=count2+1;
E2=icdf('exponential',rand,1/lambda2);
40 Be=binornd(1,1-a);
W=icdf('exponential',rand,1/lambda2);

```

```

42         E1=a*E2+Be*W;
           JumpTimes2(count2+1)=JumpTimes2(count2)+E2;
44     end
           JumpTimes2=JumpTimes2(2:end-1);
46     NT2=size(JumpTimes2,2);
end
48     Disc_Gauss=mvnrnd([0 0],[v1^2 v1*v2*rhoD; v1*v2*rhoD ...
           v2^2],max(NT1,NT2));
           Jumpsize1=(-v1^2/2+Disc_Gauss(1:NT1,1));
50     Jumpsize2=(-v2^2/2+Disc_Gauss(1:NT2,2));
           JumpTimes1=JumpTimes1';
52     JumpTimes2=JumpTimes2';
           X1(i,1)=log(S01);
54     X2(i,1)=log(S02);
           for j=1:Nsteps
56         Z12=mvnrnd([0 0],dt*[1 rhoBM; rhoBM 1]);
           X1(i,j+1)=X1(i,j)+(gamma1)*dt+sigma1*Z12(1,1);
58         X2(i,j+1)=X2(i,j)+(gamma2)*dt+sigma2*Z12(1,2);
           for ii=1:NT1
60             if JumpTimes1(ii)>t(j) && JumpTimes1(ii)≤t(j+1) % ...
                   aggiungo i salti
                   X1(i,j+1)=X1(i,j+1)+Jumpsize1(ii)+log(M1);
62             end
           end
end
64     for jj=1:NT2
           if JumpTimes2(jj)>t(j) && JumpTimes2(jj)≤t(j+1)
66         X2(i,j+1)=X2(i,j+1)+Jumpsize2(jj)+log(M2);
           end
end
68     end
end
70 end
           S1=exp(X1);
72     S2=exp(X2);
end

```

Appendice B

Correlazione jump diffusion

- $\rho^{(J)}$ nel caso GBM con salti

$$\begin{aligned}
 \rho_{l_1, l_2, m_1, m_2}^{(J)} &= \text{Corr}[\sigma_1 W_1(T) + \sqrt{l_1} v_1 Z_1 \\
 &\quad + \sqrt{m_1} \alpha_1 Y_1, \sigma_2 W_2(T) + \sqrt{l_2} v_2 Z_2 + \sqrt{m_2} \alpha_2 Y_2] \\
 &= \frac{\rho^{(W)} \sigma_1 \sigma_2 T + \rho^{(D)} \sqrt{l_1 l_2} v_1 v_2}{\sqrt{v_{1, l_1, m_1}^{(J)}(T) v_{1, l_1, m_1}^{(J)}(T)}}
 \end{aligned} \tag{B.1}$$

- $\rho^{(J)}$ nel caso GOU con salti

$$\begin{aligned}
 \rho_{l_1, l_2, m_1, m_2}^{(J)} &= \text{Corr}[L_1, L_2] \\
 &= \frac{\frac{\rho^{(W)} \sigma_1 \sigma_2}{2\sqrt{k_1 k_2}} \sqrt{1 - e^{-2k_1 t}} \sqrt{1 - e^{-2k_2 t}} + \rho^{(D)} \sqrt{l_1 l_2} v_1 v_2 e^{-(k_1 + k_2)t}}{\sqrt{v_{1, l_1, m_1}^{(J)}(T) v_{1, l_1, m_1}^{(J)}(T)}}
 \end{aligned} \tag{B.2}$$

dove $L_i = \sigma_i \int_0^t e^{-k_i(t-s)} dW_i(s) + e^{-k_i t} (v_i \sqrt{l_i} Z_i + \sqrt{m_i} \alpha_i Y_i)$ per $i = 1, 2$.

Appendice C

Probabilità usate per calibrazione caso GOU

- $P_{0,0}$, $P_{0,1}$, $P_{1,0}$ e $P_{1,1}$

Dai risultati presi da [Petroni and Sabino, 2017], la distribuzione congiunta di $X_1 \sim \mathcal{E}(\lambda_1)$ e $X_2 \sim \mathcal{E}(\lambda_2)$ è

$$H(x_1, x_2) = \frac{\lambda_1}{\lambda_2} x_1 \wedge \frac{x_2}{a} \left[(1 - e^{(\lambda_1 x_1 \wedge \frac{\lambda_2 x_2}{a})}) - e^{-\lambda_2 x_2} (1 - e^{(1-a)(\frac{\lambda_2 x_2}{a})}) \right] \quad (\text{C.1})$$

Per Δt piccoli possiamo assumere che non avvenga più di un salto:

$$p_{1,1} = \mathbf{P}(X_1 \leq \Delta t, X_2 \leq \Delta t) = H(\Delta t, \Delta t),$$

$$p_{1,0} = \mathbf{P}(X_1 \leq \Delta t, X_2 \geq \Delta t) = F_1(\Delta t) - H(\Delta t, \Delta t),$$

$$p_{0,1} = \mathbf{P}(X_1 \geq \Delta t, X_2 \leq \Delta t) = F_2(\Delta t) - H(\Delta t, \Delta t),$$

$$p_{0,0} = \mathbf{P}(X_1 \geq \Delta t, X_2 \geq \Delta t) = 1 - p_{1,1} - p_{0,1} - p_{1,0}.$$

Bibliografia

- [Caldana et al., 2015] Caldana, R., Cheang, G. H., Chiarella, C., and Fusai, G. (2015). Correction: Exchange option under jump-diffusion dynamics. *Applied Mathematical Finance*, 22: pagine 99–103.
- [Caldana et al., 2016] Caldana, R., Fusai, G., Gnoatto, A., and Grasselli, M. (2016). General closed-form basket option pricing bounds. *Quantitative Finance*, 37: pagine 535–554.
- [Cartea and Figueroa, 2005] Cartea, A. and Figueroa, M. G. (2005). Pricing in electricity markets: a mean reverting jump diffusion model with seasonality. *Applied Mathematical Finance*, 12(4): pagine 313–335.
- [Cheang and Chiarella, 2011] Cheang, G. H. and Chiarella, C. (2011). Exchange options under jump-diffusion dynamics. *Applied Mathematical Finance*, 18(3): pagine 245–276.
- [Cufaro Petroni and Sabino, 2018] Cufaro Petroni, N. and Sabino, P. (2018). Pricing exchange options with correlated jump diffusion processes. *Quantitative Finance*, 1: pagine 1–13.
- [Gradshteyn and Ryzhik, 2014] Gradshteyn, I. S. and Ryzhik, I. M. (2014). *Table of integrals, series, and products*. Academic press.
- [Hambly et al., 2009] Hambly, B., Howison, S., and Kluge, T. (2009). Modelling spikes and pricing swing options in electricity markets. *Quantitative Finance*, 9(8): pagine 937–949.

- [Hanzon et al., 2012] Hanzon, B., Holland, F., and Sexton, C. (2012). Infinitely divisible 2-ept probability density functions. *Disponibile su www.2-ept.com*.
- [Iyer et al., 2002] Iyer, S. K., Manjunath, D., and Manivasakan, R. (2002). Bivariate exponential distributions using linear structures. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 156–166.
- [Kirk and Aron, 1995] Kirk, E. and Aron, J. (1995). Correlation in the energy markets. *Managing energy price risk*, 1: pagine 71–78.
- [Li et al.,] Li, M., Deng, S., and Zhou, J. Closed-form approximations for spread option prices and greeks. 15(3): pagine 58–80.
- [Li et al., 2010] Li, M., Zhou, J., and Deng, S.-J. (2010). Multi-asset spread option pricing and hedging. *Quantitative Finance*, 10(3): pagine 305–324.
- [Lucia and Schwartz, 2002] Lucia, J. J. and Schwartz, E. S. (2002). Electricity prices and power derivatives: Evidence from the nordic power exchange. *Review of derivatives research*, 5(1): pagine 5–50.
- [Margrabe, 1978] Margrabe, W. (1978). The value of an option to exchange one asset for another. *The journal of finance*, 33(1): pagine 177–186.
- [Mark S Joshi, 2003] Mark S Joshi, M. S. J. (2003). *The concepts and practice of mathematical finance*. Cambridge University Press.
- [Merton, 1976] Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of financial economics*, 3(1-2): pagine 125–144.
- [Muller, 2001] Muller, K. E. (2001). Computing the confluent hypergeometric function, $m(a, b, x)$. *Numerische Mathematik*, 90(1): pagine 179–196.

- [Pearson et al., 2017] Pearson, J. W., Olver, S., and Porter, M. A. (2017). Numerical methods for the computation of the confluent and gauss hypergeometric functions. *Numerical Algorithms*, 74(3): pagine 821–866.
- [Pellegrino, 2016] Pellegrino, T. (2016). A general closed form approximation pricing formula for basket and multi-asset spread options. *Journal of Mathematical Finance*, 6(5): pagina 944.
- [Pellegrino and Sabino,] Pellegrino, T. and Sabino, P. On the use of the moment-matching technique for pricing and hedging multi-asset spread options. *Energy Economics*, 45(1).
- [Pellegrino and Sabino, 2014] Pellegrino, T. and Sabino, P. (2014). Pricing and hedging multiasset spread options using a three-dimensional fourier cosine series expansion method. *Journal of Energy Markets*, 7(2).
- [Petroni and Sabino, 2017] Petroni, N. C. and Sabino, P. (2017). Coupling poisson processes by self-decomposability. *Mediterranean Journal of Mathematics*, volume 14: pagine 1–69.
- [Sato and Ken-Iti, 1999] Sato, K.-i. and Ken-Iti, S. (1999). *Lévy processes and infinitely divisible distributions*. Cambridge university press.
- [Schwartz and Smith, 2000] Schwartz, E. and Smith, J. E. (2000). Short-term variations and long-term dynamics in commodity prices. *Management Science*, 46(7): pagine 893–911.
- [Sexton and Hanzon, 2012] Sexton, C. and Hanzon, B. (2012). State space calculations for two-sided ept densities with financial modelling applications. *Disponibile su www.2-ept.com*.