

POLITECNICO DI MILANO  
FACOLTÀ DI INGEGNERIA DEI SISTEMI  
CORSO DI STUDI IN INGEGNERIA MATEMATICA



TESI DI LAUREA MAGISTRALE

**Un approccio algoritmico alla generazione  
di scenari per il pricing di ForEx options**

Relatore: Prof. Carlo Sgarra

Candidato:

Francesco Luigi ZACCARIA  
Matr. 770290

ANNO ACCADEMICO 2012-2013

# Abstract

In this thesis work we propose an operative solution to the generation of Foreign Exchange rates, Risk Management oriented.

In order to adopt fair risk measures, we strongly believe that the generated scenarios must be able to forecast future evolutions of ForEx rates rapidly and with the maximum detail.

Thus, we inspect the main theoretical and computational aspects regarding the formulation of G2++ market model, in order to realize a Java implementation.

The applied methodologic approach identifies and discuss four consecutive functional blocks in a detailed manner:

1. the implementation of the algorithm *Mersenne Twister for Graphic Processors* to generate uniformly distributed random variates in  $[0, 1)$ , validated by the most up-to-date randomness tests in literature,
2. the implementation of *Ziggurat* and *Box-Muller*, to generate Gaussian random variates, starting from the uniform distribution in  $[0, 1)$ , validated by the comparison with MATLAB performances,
3. the implementation of G2++ market model to describe future evolutions of the short rate  $r(t)$  of a single currency (G2++ Single Currency).

In this framework, we particularly analyze an up-to-date pricing formula for the European Swaption and a new algorithm to perform the model calibration, consisting in a composition of *Downhill Simplex* and *Simulated Annealing*.

4. the performing of an extension and generalization of G2++ Single Currency in the case of a market composed by  $N$  currencies (G2++ Multi Currency), to describe the future evolution of the  $N - 1$  Exchange Rates in terms of a fixed *domestic* currency.

# Sommario

Nel presente elaborato di tesi proponiamo una soluzione operativa per la generazione di scenari di tasso di interesse in ambito Foreign Exchange, rivolto ad applicazioni di Risk Management.

Al fine di adottare corrette misure di rischio, riteniamo di fondamentale importanza che gli scenari generati siano in grado di prevedere in tempi brevi e con il maggior dettaglio possibile le future evoluzioni dei tassi ForEx.

Il presente elaborato di tesi indaga i principali aspetti teorici e computazionali inclusi nella formulazione del modello di mercato G2++, al fine di realizzarne un'implementazione Java.

L'approccio metodologico applicato identifica e discute dettagliatamente quattro blocchi funzionali consecutivi:

1. l'implementazione dell'algoritmo *Mersenne Twister for Graphic Processors* per la generazione di variabili aleatorie uniformi nell'intervallo  $[0, 1)$ , validata mediante i più recenti test di randomicità presenti in letteratura,
2. l'implementazione di *Ziggurat* e *Box-Muller*, per la generazione di variabili aleatorie Gaussiane a partire dalla distribuzione uniforme in  $[0, 1)$ , validata dal confronto con le performance di MATLAB,
3. l'implementazione del modello G2++ per la descrizione dell'evoluzione temporale del tasso short  $r(t)$  di una singola valuta (G2++ Single Currency).  
In questo ambito, abbiamo in particolare analizzato una recente formula di pricing per la Swaption Europea ed un recente algoritmo di calibrazione del modello, rappresentato dalla composizione di *Downhill Simplex* e *Simulated Annealing*.
4. l'estensione e la generalizzazione di G2++ Single Currency al caso di un mercato composto da  $N$  valute (G2++ Multi Currency), per la descrizione dell'evoluzione temporale degli  $N - 1$  Exchange rates, rispetto alla valuta *domestica*.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
<b>2</b>	<b>Generazione di numeri casuali</b>	<b>9</b>
2.1	Framework: generazione di numeri casuali in $\mathbb{F}_2$	10
2.1.1	Principali definizioni e proprietà	10
2.1.2	La proprietà di equidistribuzione in k dimensioni	12
2.2	Mersenne Twister MT19937	13
2.2.1	Descrizione della ricorrenza	14
2.2.2	Tempering	15
2.2.3	Algoritmo per la generazione di un numero casuale	15
2.3	Mersenne Twister for Graphic Processors (MTGP)	16
2.3.1	Introduzione alla ricorsione	16
2.3.2	Computazione parallela di una singola ricorsione	17
2.3.3	Descrizione dell'algoritmo di ricorsione	18
2.4	Test statistici per il generatore MTGP	20
2.4.1	Strumenti statistici	21
2.4.2	Test effettuati	22
2.4.3	Commento sui test effettuati	26
2.5	Generazione di una distribuzione Gaussiana	27
2.5.1	Box-Muller	27
2.5.2	Ziggurat	30
2.5.3	Confronto tra i due metodi	32
<b>3</b>	<b>Il modello G2++ Single Currency</b>	<b>34</b>
3.1	Descrizione della dinamica	34
3.2	Pricing dei principali prodotti finanziari	37
3.2.1	Zero-Coupon Bond	37
3.2.2	Opzioni Europee su Zero-Coupon Bond	41
3.2.3	Caplet e Floorlet	44
3.2.4	Cap e Floor	45
3.3	Swaption Europea	45
3.3.1	Formula di Brigo e Mercurio	45
3.3.2	Formula di Choi e Shin	47

3.3.3	Una nota sulle due formule di pricing . . . . .	52
3.4	Calibrazione su Dati di Mercato . . . . .	53
3.4.1	Formulazione . . . . .	53
3.4.2	Formula di Black-Scholes per la valutazione della Swaption Europea . . . . .	55
3.4.3	Vincoli . . . . .	55
3.4.4	Caratterizzazione della Regione di Ottimizzazione . . . . .	55
3.4.5	Downhill Simplex . . . . .	56
3.4.6	Simulated Annealing . . . . .	59
<b>4</b>	<b>G2++ Multi Currency</b>	<b>62</b>
4.1	Descrizione delle dinamiche nelle rispettive misure risk-neutral	62
4.2	Dinamica del tasso di cambio rispetto ad un'unica Forward-measure . . . . .	65
4.3	Pricing dei principali prodotti finanziari . . . . .	66
4.3.1	Quanto Constant-Maturity Swap (QCMS) . . . . .	66
4.3.2	Quanto Caplet/Floorlet . . . . .	73
4.3.3	Quanto Cap/Floor . . . . .	75
4.3.4	Quanto Swaption . . . . .	75
<b>5</b>	<b>Conclusioni</b>	<b>80</b>
<b>A</b>	<b>Algoritmo di Berlekamp-Massey</b>	<b>82</b>
<b>B</b>	<b>Algoritmo [Marsaglia 1964]</b>	<b>84</b>
<b>C</b>	<b>Dimostrazione del Teorema 3.3.1</b>	<b>86</b>
<b>D</b>	<b>Codici Matlab</b>	<b>88</b>
D.1	Soluzione della dinamica . . . . .	88
D.2	Zero Coupon Bond . . . . .	90
D.3	Swap . . . . .	92
D.4	Gradiente dello Swap . . . . .	92
D.5	Swaption Europea . . . . .	93

# Capitolo 1

## Introduzione

Contestualmente al pricing di prodotti finanziari, al trading ed al risk management, è di fondamentale importanza poter disporre in tempi brevissimi di previsioni che siano in grado di aderire il più possibile alle future evoluzioni dei tassi, in modo da determinare prezzi corretti (nel pricing), prevedere i trend di mercato (nel trading) ed adottare opportune misure di rischio (nel risk management). Tali previsioni sono effettuate sulla base di modelli matematici, ognuno dei quali si fonda su distinte ipotesi di mercato e consiste tipicamente nella risoluzione di un sistema di equazioni differenziali stocastiche.

L'obiettivo del presente elaborato di tesi è quello di indagare tutti gli aspetti teorici e computazionali coinvolti nell'implementazione del modello di mercato G2++, secondo un approccio paradigmatico, applicabile a qualsiasi modello di mercato.

Al fine di realizzare un'implementazione Java del modello G2++, si identificano e discutono dettagliatamente quattro blocchi funzionali consecutivi:

1. la generazione di variabili aleatorie uniformi nell'intervallo  $[0, 1)$ ,
2. la generazione di variabili aleatorie Gaussiane, aventi media e varianza fissate, a partire dalla distribuzione uniforme in  $[0, 1)$ ,
3. l'implementazione del modello G2++ per la descrizione dell'evoluzione temporale del tasso short  $r(t)$  di una singola valuta (G2++ Single Currency),
4. l'estensione e la generalizzazione di G2++ Single Currency al caso di un mercato composto da  $N$  valute (G2++ Multi Currency), per la descrizione dell'evoluzione temporale degli  $N - 1$  tassi di cambio, rispetto alla valuta *domestica*.

Il secondo capitolo riguarda la generazione di numeri pseudo-casuali uniformemente distribuiti in  $[0, 1)$ . In seguito all'introduzione dei concetti fon-

damentali dalla teoria del campo di Galois  $\mathbb{F}_2 \{0, 1\}$ , esposta in [9], si considera l'algoritmo Mersenne Twister che, da quanto si apprende in letteratura, rappresenta lo stato dell'arte nella generazione di numeri pseudo-casuali. In due sezioni distinte del Capitolo si descrivono dettagliatamente gli algoritmi *Mersenne Twister MT19937* e *Mersenne Twister for Graphic Processors* (MTGP). Il primo, proposto nel 1997 da Matsumoto e Nishimura in [15] e rivolto ad applicazioni di Calcolo Seriale, rappresenta una forte evoluzione nei confronti dei precedenti Linear Congruential Generators in termini di equidistribuzione multidimensionale e periodo massimo della sequenza di numeri pseudo-casuali generati.

Il generatore MTGP, proposto da Matsumoto e Saito in [17], rappresenta la trasposizione dell'algoritmo seriale di Mersenne Twister MT19937 in un algoritmo ottimizzato per il Calcolo Parallelo. In particolare, Matsumoto e Saito propongono un paradigma di parallelizzazione che, nella relativa sezione, sarà illustrato ed a cui proponiamo un paradigma alternativo di parallelizzazione, motivandone la scelta.

Al fine di verificare la corretta traduzione in codice Java dell'algoritmo MTGP in Java e di testare l'efficacia nella generazione della distribuzione uniforme di numeri pseudo-casuali a 32-bit, sono stati effettuati alcuni test statistici molto gettonati in letteratura, in particolare il test proposto da Anderson e Darling in [1] e la suite "Some difficult-to-pass tests of randomness", redatta da Marsaglia e Tsang in [14]. Tale suite si compone dei test denominati *GCD* (Greatest Common Divisor), *Gorilla Test* e *Birthday Test*. Relativamente all'esperienza di Marsaglia, un efficiente generatore di numeri casuali ottiene esito positivo per ogni test della suite.

Il primo test, GCD, data una sequenza di numeri casuali (interi senza segno) a 32-bit, verifica se  $k$ , definito come il numero di passi necessari alla terminazione dell'algoritmo di Euclide per ogni coppia di numeri della sequenza, ed il massimo comun divisore risultante, ben approssimano una particolare distribuzione Normale, calcolata mediante simulazioni estensive, da Marsaglia.

Il Gorilla Test consiste nell'interpretare il generatore di numeri casuali come la proverbiale scimmia che, posta alla macchina da scrivere, digita una lunga sequenza di caratteri. Il numero delle parole mancanti, composte da  $k$  lettere, dovrebbe approssimare una distribuzione Normale con media e varianza fissate e dipendenti dalla lunghezza della parola, dal numero di lettere nell'alfabeto e dalla probabilità di estrarre una lettera dall'alfabeto.

Il Birthday Test utilizza ogni intero a 32-bit generato casualmente per determinare un compleanno in un anno composto da  $n$  giorni. Avendo scelto casualmente e disposto in ordine crescente  $m$  compleanni, il numero di compleanni coincidenti è una distribuzione di Poisson, avente parametro  $\lambda = m^2/(4n)$ .

Dei test effettuati si riportano gli esiti, tutti positivi, che permettono di affermare che MTGP è un generatore idoneo per simulare le realizzazioni di

una variabile aleatoria uniformemente distribuita in  $[0, 1)$ .

Nell'ultima sezione del secondo Capitolo si considerano Ziggurat [13] e Box-Muller [5], i due algoritmi più utilizzati per la conversione di una distribuzione uniforme di numeri casuali in una distribuzione Gaussiana, avente media e varianza fissate.

I due algoritmi sono descritti dettagliatamente e confrontati con l'algoritmo *randn* di Matlab, al fine di verificare la correttezza dell'implementazione Java effettuata e l'efficacia degli algoritmi di trasformazione considerati. Si è inoltre valutato e confrontato l'impatto computazionale tra i due algoritmi, concludendo che Ziggurat sia l'algoritmo preferibile per applicazioni di Calcolo Seriale, mentre Box-Muller sia opportuno per applicazioni di Calcolo Parallelo.

Il risultato globale, ottenuto alla fine del secondo capitolo, è la possibilità di replicare, con sufficiente accuratezza e limitando i costi computazionali, il moto Browniano, necessario per l'implementazione del modello G2++.

Nel terzo Capitolo si considera il modello G2++ per la descrizione dell'evoluzione temporale del tasso short  $r$ , la cui formulazione è descritta in [6]. Nella prima sezione si illustrano le equazioni differenziali stocastiche correlate che governano il modello, delle quali si calcolano le soluzioni. Al fine di ottimizzare la computazione di tali soluzioni, si illustra la decomposizione dei due moti Browniani correlati attuata, che permette di simularli indipendentemente mediante il generatore Mersenne Twister e, in cascata, Ziggurat/Box-Muller.

Nella seconda sezione si enunciano e dimostrano le formule di pricing dei principali prodotti finanziari quotati: Zero-Coupon Bond, opzioni su Zero-Coupon Bond, Caplet, Floorlet, Cap e Floor.

Nella terza sezione si discute un metodo numerico alternativo, proposto in [8], per il pricing di una Swaption Europea, computazionalmente meno oneroso rispetto alla formula di [6].

In seguito all'introduzione delle notazioni e del modello utilizzati da [8], si dimostra che tale modello differisce da G2++ per uno shift deterministico, di cui si calcola il valore.

I codici MATLAB delle funzioni utilizzate per il calcolo del prezzo della Swaption Europea sono riportati in appendice D.

Nella quarta sezione si introduce il problema della calibrazione del modello G2++ Single Currency, discutendo approfonditamente l'algoritmo proposto in [3], consistente nella composizione degli algoritmi Downhill Simplex e Simulated Annealing. Si illustra come l'algoritmo proposto sia talvolta in grado, rispetto ad altri algoritmi presenti in letteratura, di uscire dagli *avvalamenti* circostanti ad ottimi locali, aumentando la probabilità di convergere verso l'ottimo globale.



Nel quarto Capitolo si considera il modello G2++ in contesto multi-valuta, per descrivere l'evoluzione temporale dei tassi di cambio di  $N - 1$  valute *estere*, rispetto ad una valuta *domestica* fissata.

La formulazione del modello, contenuta in [6], considera un mercato composto da 2 valute e, senza perdita di generalità, può essere estesa ad un mercato composto da  $N$  valute. Nella prima sezione si conservano ed estendono le notazioni del precedente capitolo per descrivere le soluzioni delle dinamiche delle due valute considerate, ognuna rispetto alla propria misura risk-neutral.

Nella seconda sezione si descrive come modellare il tasso di cambio tra due valute, riconducendo le dinamiche relative alle due curve ad un'unica misura risk-neutral. Si pone, inoltre, attenzione alle necessarie approssimazioni numerico-modellistiche proposte in [6].

In particolare, si illustra come il modello G2++ Single Currency possa essere interpretato come unità fondamentale del modello G2++ Multi Currency. La metodologia, proposta nel presente lavoro di tesi, consiste nel calibrare indipendentemente ogni valuta rispetto alle proprie volatilità implicite di Cap/Swaption, per gestire successivamente le interazioni tra la valuta domestica ed ogni altra valuta estera, mantenendo l'impostazione *pairwise* di [6]. Gli input della calibrazione sono la curva degli Zero-Coupon Bond di mercato e la volatilità di Cap/Swaption nella rispettiva valuta che, trattandosi di dati di mercato, incorporano gli effetti delle interazioni tra la valuta considerata e tutte le altre presenti nel mercato.

Pertanto, non risulta necessario esplicitare o quantificare l'entità delle interazioni tra tutte le altre valute, escludendo la valuta domestica, ed è ammissibile considerare il tasso di cambio tra valuta domestica ed ogni altra delle  $N - 1$  valute.

Infine, nella terza sezione si riportano le principali formule di pricing, e le loro derivazioni, dei principali prodotti finanziari su tasso di cambio: Quanto Constant-Maturity Swap, Quanto Caplet e Quanto Floorlet, Quanto Cap e Quanto Floor, Quanto Swaption.

## Capitolo 2

# Generazione di numeri casuali

I *generatori di numeri casuali* sono richiesti sostanzialmente in qualsiasi tipo di applicazione computazionale, come la simulazione di sistemi stocastici, analisi numerica, algoritmi probabilistici ed, in particolare, in finanza. I numeri casuali possono essere generati da un dispositivo fisico, come, per esempio, nel caso del rumore termico prodotto da un componente elettronico, ma spesso sono l'output di un applicativo software che, dato un valore iniziale (detto *seed*), produce una sequenza deterministica di numeri, che si presume possa *imitare* le realizzazioni di variabili aleatorie indipendenti ed uniformemente distribuite. Producendo un output deterministico e periodico, è noto a priori che non è possibile replicare perfettamente il comportamento di variabili aleatorie indipendenti e che, pertanto, non è possibile superare tutti i possibili test statistici di uniformità ed indipendenza. Per porre in risalto questa caratteristica, si tende a qualificare questo tipo di generatori con il nome di *pseudo-casuali*.

Nelle prime tre sezioni del presente Capitolo, in seguito alla trattazione dei principali concetti ed algoritmi che costituiscono l'ambiente di lavoro per l'implementazione di generatori di numeri casuali, si considera in dettaglio lo Stato dell'Arte della generazione di numeri casuali, rappresentata dal generatore Mersenne Twister, di cui è stata implementata, durante lo stage curricolare in Unicredit, una versione ottimizzata per il Calcolo Parallelo: il Mersenne Twister for Graphic Processors (MTGP).

Nella quarta sezione sono esposti i principali test di casualità, redatti da (L'Ecuyer-Simard) e (Marsaglia), volti a verificare la qualità dei numeri casuali generati, sulla base di diverse cifre di merito. I risultati, prodotti dai test effettuati sul MTGP, sono riportati e confrontati con analoghi risultati in letteratura.

Nell'ultima sezione è presentato lo stato dell'Arte nella trasformazione di una distribuzione uniforme di probabilità in una distribuzione Gaussiana, necessaria per la modellazione del moto Browniano.

## 2.1 Framework: generazione di numeri casuali in $\mathbb{F}_2$

I calcolatori, come è ben noto, lavorano in aritmetica binaria. Risulta, pertanto, una scelta naturale implementare generatori di numeri casuali (RNGs) mediante ricorrenze aritmetiche in modulo 2. Così facendo, qualsiasi operazione aritmetica è ricondotta alla sua equivalente su stringhe di bit (shifts, rotazioni, ORs, XORs, maschere di bit) e risulta corretto descrivere lo stato di un generatore di numeri casuali mediante una ricorrenza lineare in modulo 2.

Si è dunque interessati a qualificare la classe di generatori lineari in  $\mathbb{F}_2$ , il campo finito di cardinalità 2 avente elementi  $\{0, 1\}$ . Si dimostra che tutti i più recenti e celebri generatori di numeri casuali possono essere ottenuti come casi particolari di una classe generale.

### 2.1.1 Principali definizioni e proprietà

**Definizione 2.1.1.** Un *generatore di numeri casuali* è definito da una ricorrenza lineare sul campo finito  $\mathbb{F}_2$  come segue:

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1}, \quad (2.1)$$

$$\mathbf{y}_n = \mathbf{B}\mathbf{x}_n \quad (2.2)$$

$$u_n = \sum_{l=1}^w y_{n,l-1} 2^{-l} \quad (2.3)$$

dove  $\mathbf{x}_n = (x_{n,0}, \dots, x_{n,k-1})^t \in \mathbb{F}_2^k$  è il *vettore (trasposto) di stato* a  $k$ -bit al passo  $n$ ,  $\mathbf{y}_n = (y_{n,0}, \dots, y_{n,w-1}) \in \mathbb{F}_2^w$  è il *vettore di output*, a  $w$ -bit al passo  $n$ ,  $k$  e  $w$  sono interi positivi,  $\mathbf{A}$  è la *matrice di transizione* ( $w \times w$ ) con elementi in  $\mathbb{F}_2$  e  $\mathbf{B}$  è la *matrice di trasformazione in output* con elementi in  $\mathbb{F}_2$ .

Il numero reale  $u_n \in [0, 1)$  è l'*output* al passo  $n$ . Tutte le operazioni nelle equazioni (2.1) e (2.2) sono effettuate in  $\mathbb{F}_2$ . Il periodo del generatore di numeri casuali non può essere superiore a  $2^k - 1$ , poichè  $\mathbf{x}_n$  ammette fino a  $2^k - 1$  valori non nulli. Quando questo massimo è raggiunto, si dice che il generatore di numeri casuali ha *periodo massimo*. Per discutere la periodicità e mostrare la costruzione di generatori di numeri casuali di massimo periodo, lineari in  $\mathbb{F}_2$ , si richiamano le principali definizioni e proprietà dell'algebra lineare, utilizzate in [9] e dalla teoria del campo di Galois  $\mathbb{F}_2$ . Sia  $\mathbb{F}_2[z]$  l'anello dei polinomi a coefficienti in  $\mathbb{F}_2$ .

**Definizione 2.1.2.** Si definisce *polinomio caratteristico* della matrice  $\mathbf{A}$

$$P(z) = \det(z\mathbf{I} - \mathbf{A}) = z^k - \alpha_1 z^{k-1} - \dots - \alpha_{k-1} z - \alpha_k, \quad (2.4)$$

ove  $\mathbf{I}$  è la matrice identità ed ogni  $\alpha_j$  è un elemento di  $\mathbb{F}_2$ .

**Teorema 2.1.1.**  $P(z)$  è anche il polinomio caratteristico della ricorrenza lineare (in  $\mathbb{F}_2$ )

$$x_n = \alpha_1 x_{n-1} + \dots + \alpha_k x_{n-k}. \quad (2.5)$$

**Definizione 2.1.3.** Si dice che la ricorrenza (2.5) ha *ordine*  $k$  ed è periodica di periodo  $\rho$ , i.e.  $(x_\rho, \dots, x_{\rho+k-1}) = (x_0, \dots, x_{k-1})$  per  $\rho > 0$ .

**Definizione 2.1.4.** Il *polinomio minimo* di  $\mathbf{A}$  è il polinomio  $Q(z) \in \mathbb{F}_2[z]$  di grado minimo tale che  $Q(\mathbf{A}) = 0$ . Ogni altro polinomio  $R(z) \in \mathbb{F}_2[z]$  per cui  $R(\mathbf{A}) = 0$  deve essere un multiplo del polinomio minimo. Questo implica, in particolare, che  $P(z)$  è multiplo di  $Q(z)$ . Nel contesto della costruzione di generatori di numeri casuali,  $Q(z)$  e  $P(z)$  sono quasi sempre identici, nelle buone costruzioni.

**Teorema 2.1.2.** Se la successione  $\{x_n, n \geq 0\}$  soddisfa (2.5), allora soddisfa anche l'equazione alle ricorrenze che corrisponde al polinomio minimo di  $\mathbf{A}$  (o a qualsiasi altro polinomio che sia multiplo di  $Q(z)$ ). Si ha, pertanto:

$$x_n = (\alpha_1 x_{n-1} + \dots + \alpha_k x_{n-k}) \quad (\text{in } \mathbb{F}_2). \quad (2.6)$$

Inoltre, le successioni  $\{x_{n,j}, n \geq 0\}$  e  $\{y_{n,j}, n \geq 0\}$  soddisfano (2.5) per  $0 \leq j < k$ .

**Dimostrazione 2.1.1.**

$$\begin{aligned} P(z) &= z^k - \alpha_1 z^{k-1} - \dots - \alpha_k, & P(\mathbf{A}) &= 0. \\ \mathbf{A}^k - \alpha_1 \mathbf{A}^{k-1} - \dots - \alpha_k &= 0 \\ \mathbf{A}^k &= \alpha_1 \mathbf{A}^{k-1} + \dots + \alpha_k \\ \mathbf{A}^k x_{n-k} &= \alpha_1 \mathbf{A}^{k-1} x_{n-k} + \dots + \alpha_k x_{n-k} \\ x_n &= \alpha_1 x_{n-1} + \dots + \alpha_k x_{n-k} \end{aligned}$$

**Definizione 2.1.5.** Un polinomio  $p \in \mathbb{F}_2[x]$  si dice *irriducibile* su  $\mathbb{F}_2$  (o *primo* in  $\mathbb{F}_2[x]$ ) se  $p$  ha grado positivo e  $p = bc$ , ove  $b, c \in \mathbb{F}_2[x]$ , implica che  $b$  oppure  $c$  sia un polinomio costante. In altre parole, un polinomio  $p$  è irriducibile se non ammette divisori, eccetto 1 e  $p$  stesso.

**Definizione 2.1.6.** Si dice che un polinomio  $P(z)$  è *primitivo* se e solo se è irriducibile e, per tutti i divisori primi  $p_i$  di  $r = 2^k - 1$ , si ha  $z^{r/p_i} \not\equiv 1 \pmod{P(z)}$ .

Le successioni  $\{x_{n,j}, n \geq 0\}$  e  $\{y_{n,j}, n \geq 0\}$  potrebbero essere soluzione anche di equazioni alle ricorrenze di ordine inferiore a  $k$ . Per ogni successione periodica in  $\mathbb{F}_2$ , esiste una ricorrenza lineare di *ordine minimo* risolta da questa successione, ed il polinomio caratteristico della ricorrenza è il polinomio minimo della successione, che può essere calcolato mediante l'algoritmo di

Berlekamp-Massey, riportato in Appendice A.

Le successioni  $\{x_{n,j}, n \geq 0\}$  possono avere polinomi minimi diversi, per differenti valori di  $j$ , e diversi anche rispetto alle successioni  $\{y_{n,j}, n \geq 0\}$ . Ma tutti questi polinomi minimi devono necessariamente essere divisori di  $P(z)$ . Se  $P(z)$  è irriducibile, allora  $P(z)$  deve essere il polinomio minimo di tutte queste successioni.

Si riporta, senza dimostrazione, il seguente importante

**Teorema 2.1.3.** *Le successioni (5) e (6) hanno periodo massimo se e solo se  $P(z)$  è un polinomio primitivo su  $\mathbb{F}_2$*

Un buon modo per stabilire se un polinomio è primitivo è di verificare in primo luogo l'irriducibilità, e poi verificare la seconda condizione. Si noti che, quando  $r$  è primo (in questo caso si dice *numero primo di Mersenne*), la seconda condizione è automaticamente verificata.

Nel caso dell'implementazione di generatori di numeri pseudo-casuali, si è interessati quasi esclusivamente ad equazioni alle ricorrenze di periodo massimo. Si può calcolare e verificare la sua primitività come segue. Si calcolano  $k$  numeri casuali mediante il generatore, partendo da uno stato iniziale arbitrario e non identicamente nullo  $\mathbf{x}_0$  e si calcola il polinomio minimo  $Q_0(z)$  di  $\{x_{n,0}, n \geq 0\}$  con l'algoritmo di Berlekamp-Massey. Se il grado di  $Q_0(z)$  è inferiore a  $k$ , allora  $P(z)$  è riducibile e si deve rifiutare il generatore, altrimenti  $P(z) = Q_0(z)$  e resta da verificare la primitività di  $P(z)$ .

## 2.1.2 La proprietà di equidistribuzione in $k$ dimensioni

La qualità dei numeri pseudo-casuali è un aspetto fondamentale nel contesto di simulazioni di tipo Monte Carlo. Oltre alla consueta cifra di merito rappresentata dal periodo, inteso come il numero di iterazioni necessarie affinché il generatore produca in output il seed, occorre introdurre la definizione di equidistribuzione multidimensionale.

**Definizione 2.1.7.** Si consideri una sequenza di numeri pseudo-casuali  $\mathbf{x}_i$  di interi con periodo  $P$ . Sia  $\text{trunc}_v(\mathbf{x})$  il numero intero formato dai primi  $v$  bits di  $\mathbf{x}$  e si consideri un intero periodo dei vettori a  $kv$ -bit

$$(\text{trunc}_v(\mathbf{x}_i), \text{trunc}_v(\mathbf{x}_{i+1}), \dots, \text{trunc}_v(\mathbf{x}_{i+k-1})) \quad (0 \leq i < P).$$

Si dice che  $\mathbf{x}_i$  è  *$k$ -distribuita con  $v$ -bit di accuratezza* se ognuna delle possibili  $2^{kv}$  possibili combinazioni di bit si verifica lo stesso numero di volte in un periodo, eccezion fatta per la combinazione identicamente nulla, che ricorre meno spesso.

Per ogni  $v = 1, 2, \dots, w$ , sia  $k(v)$  il massimo numero tale che la sequenza sia  $k(v)$ -distribuita con  $v$ -bit di accuratezza.

Il significato geometrico della precedente definizione è il seguente: sia  $x_i := \mathbf{x}_i/2^w$ , i.e. si normalizzino gli interi a  $w$ -bit in numeri reali nell'intervallo  $[0, 1)$ . Si posizionino  $P$  punti, aventi coordinate  $(x_i, x_{i+1}, \dots, x_{i+k-1})$  ( $i = 0, 1, \dots, P-1$ ) nel cubo unitario  $k$ -dimensionale. Si divida il cubo, lungo ogni dimensione, in  $2^v$  partizioni, ottenendo  $2^{kv}$  cubetti. Allora la sequenza è  $k$ -distribuita con accuratezza a  $v$ -bit se ogni cubetto contiene lo stesso numero di punti (eccetto per il cubo posto all'origine degli assi, che ne contiene uno in meno).

La  $k$ -distribuzione, come riportato in [15], è una proprietà molto importante in crittografia: si assuma che una sequenza di numeri pseudo-casuali sia  $k$ -distribuita con accuratezza a  $v$  bits, e che tutti i bits nel seed siano scelti in modo casuale. Allora, la conoscenza dei  $v$  bit più significativi dei primi  $l$  numeri generati, non permette di inferire alcunché circa i  $v$  bit più significativi del numero successivo, se  $l < k$ . Si ha, infatti, che ogni combinazione di bit ricorre con eguale probabilità nei  $v$  bit del numero seguente, per definizione di  $k$ -distribuzione.

La non predicibilità degli output di un generatore di numeri casuali, in più dimensioni, risulta una proprietà molto appetibile anche in finanza. Poiché il sistema simulato è sensibile solo alla storico dei  $k$  stati assunti, con accuratezza a  $v$ -bits, e non da stati futuri, è possibile effettuare simulazioni di tipo Monte Carlo, in contesto di non arbitraggio.

## 2.2 Mersenne Twister MT19937

Il *Mersenne<sup>1</sup> Twister* MT19937 è un generatore di numeri casuali sviluppato da Matsumoto e Nishimura nel 1997. Per un'accurata scelta di parametri, l'algoritmo è dotato di un periodo di  $2^{19937} - 1$  e della proprietà di equidistribuzione in spazi vettoriali fino a 623 dimensioni con accuratezza a 32 bit, considerando un vettore di stato di 624 elementi. Pertanto, MT19937 è attualmente uno dei generatori di numeri casuali più utilizzati per effettuare simulazioni Monte Carlo ed è stato incluso nell'ultima release del software statistico R come generatore predefinito.

Analogamente ai generatori lineari in  $\mathbb{F}_2$ , l'algoritmo del Mersenne Twister consiste unicamente in trasformazioni lineari fissate su uno spazio vettoriale fissato. Si illustrano l'algoritmo di generazione di numeri pseudo-casuali del Mersenne Twister, unitamente alla sequenza di operazioni binarie effettuate sull'output, detta *tempering*, al fine di migliorare la proprietà di equidistribuzione multidimensionale con un numero arbitrario di bits di accuratezza.

---

<sup>1</sup> $M_p$  si definisce *numero primo di Mersenne* se  $M_p = 2^p - 1$ , ove  $p$  è un numero primo

### 2.2.1 Descrizione della ricorrenza

Sia  $\mathbf{x}$  un *word vector*, ossia un vettore riga  $w$ -dimensionale sul campo di Galois  $\mathbb{F}_2 = \{0, 1\}$ , identificato con una stringa di bit di lunghezza  $w$ , con il bit meno significativo (LSB) a destra.

L'algoritmo del Mersenne Twister genera una sequenza di stringhe di bit, le quali rappresentano interi pseudo-casuali uniformemente distribuiti tra 0 e  $2^w - 1$ . Dividendo per  $2^w - 1$ , si può interpretare ogni stringa di bit come un numero reale in  $[0, 1)$ . La ricorrenza lineare in  $\mathbb{F}_2$  è definita come segue:

$$\mathbf{x}_{k+n} := \mathbf{x}_{k+m} \oplus (\mathbf{x}_k^u | \mathbf{x}_{k+1}^l) \mathbf{A}, \quad (k = 0, 1, \dots) \quad (2.7)$$

ove l'intero  $n$  rappresenta il *grado della ricorrenza*,  $r$  è un intero (nascosto nella definizione di  $\mathbf{x}_k^u$ ) tale che  $0 \leq r \leq w - 1$ ,  $m$  è un intero tale che  $1 \leq m \leq n$  ed  $\mathbf{A}$  è una matrice costante  $w \times w$  ad elementi in  $\mathbb{F}_2$ . Siano  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{n-1}$  i valori di inizializzazione (*seeds*) del vettore  $\mathbf{x}$ .

Allora, il generatore calcola l'elemento  $\mathbf{x}_n$  risolvendo la precedente equazione alle ricorrenze, per  $k = 0$ . Per  $k = 1, 2, \dots$ , il generatore determina  $\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \dots$ .

$\mathbf{x}_k^u$  indica i primi  $w - r$  bits di  $\mathbf{x}_k$ , e  $\mathbf{x}_{k+1}^l$  gli ultimi  $r$  bits di  $\mathbf{x}_{k+1}$ . Pertanto, se  $\mathbf{x} = (x_{w-1}, x_{w-2}, \dots, x_0)$ , allora per definizione  $\mathbf{x}^u$  è il vettore  $(x_{w-1}, \dots, x_r)$ , di lunghezza  $w - r$ , e  $\mathbf{x}^l$  è il vettore  $(x_{r-1}, \dots, x_0)$ , di lunghezza  $r$ .  $(\mathbf{x}_k^u | \mathbf{x}_{k+1}^l)$  è solo la concatenazione dei due precedenti vettori: si compone dei primi  $w - r$  bits di  $\mathbf{x}_k$  e degli ultimi  $r$  bits di  $\mathbf{x}_{k+1}$ .

Si è scelta una matrice  $\mathbf{A}$  in modo che la moltiplicazione per questa matrice sia molto veloce:

Allora la matrice  $\mathbf{A}$  è post-moltiplicata da questo vettore. Infine, a questo vettore si somma  $\mathbf{x}_{k+m}$  in modulo 2.

$$\mathbf{A} = \begin{pmatrix} & & & & 1 & & & \\ & & & & & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \\ a_{w-1} & a_{w-2} & \dots & \dots & a_0 & & & \end{pmatrix} \quad (2.8)$$

Allora il calcolo di  $\mathbf{x}\mathbf{A}$  può essere effettuato mediante operazioni bitwise:

$$\mathbf{x}\mathbf{A} = \begin{cases} \text{shiftright}(\mathbf{x}) & \text{se } x_0 = 0; \\ \text{shiftright}(\mathbf{x}) \oplus \mathbf{a} & \text{se } x_0 = 1. \end{cases} \quad (2.9)$$

ove  $\mathbf{a} = (a_{w-1}, a_{w-2}, \dots, a_0)$  e  $\mathbf{x} = (x_{w-1}, x_{w-2}, \dots, x_0)$ . Si verifica facilmente che tutte le operazioni contenute nell'equazione (2.7) possono essere effettuate mediante shift di bits, AND, OR e XOR.

## 2.2.2 Tempering

Per migliorare le proprietà di equidistribuzione con accuratezza a  $v$ -bits, si esegue una trasformazione lineare del vettore generato, post-moltiplicandolo per un'adeguata matrice  $\mathbf{T}$ ,  $w \times w$  ed invertibile, detta di *tempering*. Come matrice di tempering  $\mathbf{x} \mapsto \mathbf{z} = \mathbf{x}\mathbf{T}$ , si scelgono le seguenti trasformazioni

$$\mathbf{y} := \mathbf{x} \oplus (\mathbf{x} \gg u) \quad (2.10)$$

$$\mathbf{y} := \mathbf{y} \oplus ((\mathbf{x} \ll u) \text{ AND } \mathbf{b}) \quad (2.11)$$

$$\mathbf{y} := \mathbf{y} \oplus ((\mathbf{y} \ll u) \text{ AND } \mathbf{c}) \quad (2.12)$$

$$\mathbf{z} := \mathbf{y} \oplus (\mathbf{y} \gg l) \quad (2.13)$$

ove  $l, s, t$  e  $u$  sono interi,  $\mathbf{b}$  e  $\mathbf{c}$  sono maschere di bit adeguate alla lunghezza del vettore, e  $(\mathbf{x} \gg u)$  denota il rightshift di  $u$ -bit ( $(\mathbf{x} \ll u)$  è il leftshift di  $u$ -bit).

Per eseguire la ricorrenza (2.7) è sufficiente considerare un vettore di lunghezza  $n$  come area di lavoro, come segue. Siano  $x[0 : n - 1]$  un vettore di  $n$  numeri interi a  $w$ -bit,  $i$  una variabile intera ed  $\mathbf{u}$ ,  $\mathbf{l}$  ed  $\mathbf{a}$  costanti intere positive a  $w$ -bit.

## 2.2.3 Algoritmo per la generazione di un numero casuale

1.
  - $\mathbf{u} \leftarrow \underbrace{1 \cdots 1}_{w-r} \underbrace{0 \cdots 0}_r$ ; (maschera per i  $w - r$  bits più significativi)
  - $\mathbf{l} \leftarrow \underbrace{0 \cdots 0}_{w-r} \underbrace{1 \cdots 1}_r$ ; (maschera per gli  $r$  bits meno significativi)
  - $\mathbf{a} \leftarrow a_{w-1}a_{w-2} \cdots a_1a_0$ ; (l'ultima riga della matrice  $\mathbf{A}$ )
2.  $i \leftarrow 0$   
 $\mathbf{x}[0], \mathbf{x}[1], \dots, \mathbf{x}[n - 1] \leftarrow$  qualsiasi valore iniziale non nullo
3.  $\mathbf{y} \leftarrow (\mathbf{x}[i] \text{ AND } \mathbf{u}) \text{ OR } (\mathbf{x}[(i+1) \bmod n] \text{ AND } \mathbf{l})$ ; (calcolo di  $(\mathbf{x}_i^u | \mathbf{x}_{i+1}^l)$ )
4.  $\mathbf{x}[i] \leftarrow \mathbf{x}[(i + m) \bmod n] \text{ XOR } (\mathbf{y} \gg 1) \text{ XOR } \begin{cases} 0 & \text{se } LSB_{\mathbf{y}} = 0 \\ \mathbf{a} & \text{se } LSB_{\mathbf{y}} = 1 \end{cases}$
5. (calcolare  $\mathbf{x}[i]\mathbf{T}$ )
  - $\mathbf{y} \leftarrow \mathbf{x}[i]$
  - $\mathbf{y} \leftarrow \mathbf{y} \text{ XOR } (\mathbf{y} \gg u)$
  - $\mathbf{y} \leftarrow \mathbf{y} \text{ XOR } ((\mathbf{y} \ll s) \text{ AND } \mathbf{b})$



- $\mathbf{y} \leftarrow \mathbf{y} \text{ XOR } ((\mathbf{y} \ll t) \text{ AND } \mathbf{c})$
  - $\mathbf{y} \leftarrow \mathbf{y} \text{ XOR } (\mathbf{y} \gg l)$
  - output  $\mathbf{y}$
6.  $i \leftarrow (i + 1) \bmod n$
  7. Ritornare al punto 3)

## 2.3 Mersenne Twister for Graphic Processors (MTGP)

Nella presente sezione si considera una recente evoluzione del generatore Mersenne Twister, presentata in [17].

Si discute in dettaglio l’algoritmo per la generazione di numeri casuali, soffermandosi sulla proposta di parallelizzazione dell’algoritmo, introdotta da Saito e Matsumoto, su un’architettura CUDA, consistente nel paradigma “One block, one generator”. Nel caso in cui il generatore di numeri casuali sia applicato ad un motore Monte Carlo, si propone un pattern alternativo di parallelizzazione, più performante e computazionalmente meno oneroso, che può essere riassunto nella dicitura “One thread<sup>2</sup>, one generator”.

### 2.3.1 Introduzione alla ricorsione

Sia  $W$  l’insieme degli interi a  $w$ -bit ( $w$ : word size), e sia  $x_i \in W$  ( $i = 0, 1, 2, \dots$ ) una sequenza di interi a  $w$ -bit. Per generare una sequenza di numeri pseudo-casuali è prassi comune risolvere un’equazione alle ricorrenze: sia  $N$  un intero positivo (chiamato *grado*), sia  $f : W^N \rightarrow W$  una funzione, e siano  $x_0, x_1, \dots, x_{N-1}$  elementi di  $W$  (valori iniziali). Allora, la seguente ricorsione genera una sequenza di elementi in  $W$ :

$$x_{N+i} := f(x_{N-1+i}, x_{N-2+i}, \dots, x_i) \quad (i = 0, 1, 2, \dots) \quad (2.14)$$

Per una generazione più rapida, si preferisce adottare una  $f$  dipendente da meno variabili. Ma se le variabili non sono in numero sufficiente, la sequenza generata tende ad essere non randomica. Come trade-off, si è dunque considerato il seguente tipo di ricorsione:

$$x_{N+i} := f(x_{M+i}, x_{1+i}, x_i) \quad (i = 0, 1, 2, \dots). \quad (2.15)$$

L’intero positivo  $M$  è denominato *middle position*. Questo tipo di ricorsione può essere risolta (mediante un’accurata scelta di  $f$ ) utilizzando un vettore  $X[0 \dots L - 1]$  di parole di lunghezza  $L$  con  $L \geq N$  come segue.

---

<sup>2</sup>Si definisce *thread di esecuzione* la suddivisione di un processo, in due o più sottoprocessi, eseguiti concorrentemente da un sistema di elaborazione monoprocesore o multiprocesore

1. Custodire i valori iniziali  $x_0, \dots, x_{N-1}$  in  $X[0], X[1], \dots, X[N-1]$ .
2. Fissare la variabile intera  $i$  a 0.
3. Sia  $X[(N+i) \bmod L] \leftarrow f(X[(M+i) \bmod L], X[(1+i) \bmod L], X[i \bmod L])$ . (Questo calcola  $x_{N+i}$ )
4. Incrementare  $i \leftarrow i + 1$ . Se  $i \geq L$ , allora  $i \leftarrow i \bmod L$ .
5. Ritornare al passo 3.

### 2.3.2 Computazione parallela di una singola ricorsione

Sia  $L$  sufficientemente grande, tipicamente  $L \geq 2N - M$ . La proposta di parallelizzazione di Saito e Matsumoto, definita con il paradigma *one block, one generator*, prevede di calcolare

$$X[N+i] \leftarrow f(X[M+i], X[1+i], X[i]) \quad (2.16)$$

per  $0 \leq i \leq N - M - 1$  in parallelo. Quando  $i = N - M$  la computazione di  $X[N+i]$  richiede il valore di  $X[M+i] = X[N]$ , che può essere ottenuto solo dopo il passo 0 della ricorsione, producendo dunque l'upper bound  $N - M$  per il numero di thread paralleli eseguibili simultaneamente.

La strategia di Saito e Matsumoto consiste dunque nello sfruttare il parallelismo per  $i$  threads in un blocco, come rappresentato in figura.

MTGP: One Block  
one generator

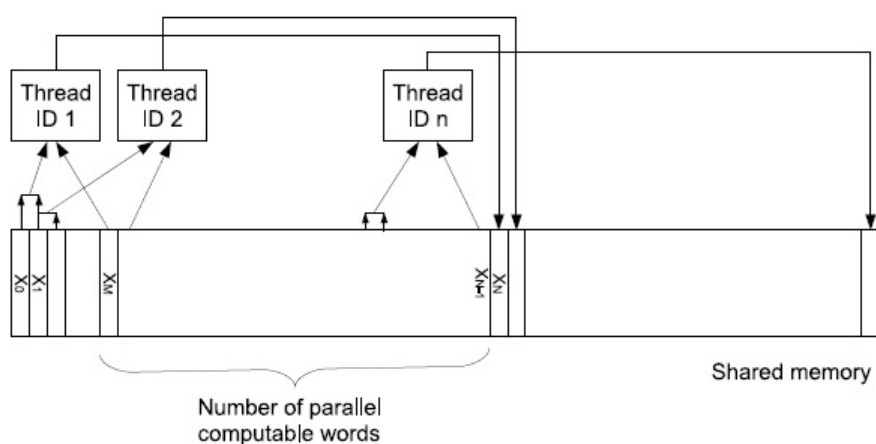


Figura 2.1: Strategia di Saito e Matsumoto: un blocco lavora su una singola ricorsione, utilizzando fino a  $N - M$  threads.

In termini procedurali, la computazione di una singola ricorsione è riassunta nel seguente algoritmo:

1. leggere  $X[i-1]$  ( $1 \leq i \leq n$ )
2. leggere  $X[i]$  ( $1 \leq i \leq n$ )
3. leggere  $X[M+i-1]$  ( $1 \leq i \leq n$ )
4. calcolare  $f(X[M+i-1], X[i], X[i-1])$  e scrivere il risultato in  $X[N+i]$  ( $1 \leq i \leq n$ )

Nel contesto di un motore Monte Carlo, tale paradigma di parallelizzazione presenta un upper bound molto stringente, rendendo, di fatto, poco efficiente la computazione.

Una possibile alternativa è costituita dal paradigma *one thread, one generator*. Fissato il numero di simulazioni, si istanzia un generatore Mersenne Twister, con parametri di periodo, tempering e seed distinti, per ogni traiettoria. In tal modo, ogni generatore costituisce una variabile aleatoria indipendente ed identicamente distribuita.

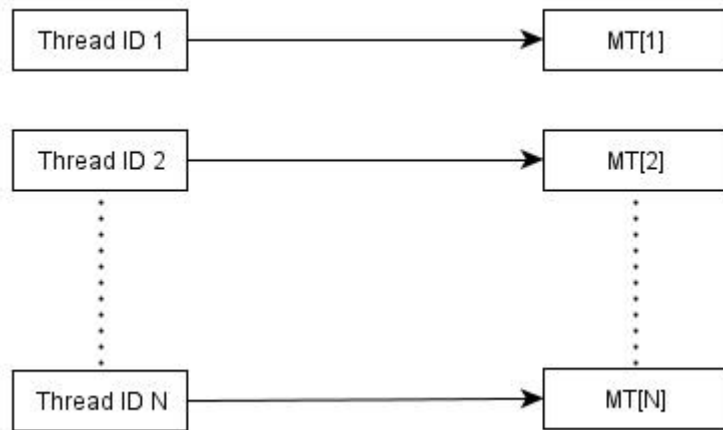


Figura 2.2: Strategia proposta: un generatore, inizializzato con parametrizzazioni distinte, per ogni thread

### 2.3.3 Descrizione dell'algoritmo di ricorsione

Sia  $p$  un esponente di Mersenne fissato. Siano inoltre  $N = \lceil p/w \rceil$ , lunghezza minima del vettore a  $w$ -bit per contenere  $p$  bits, e  $r = wN - p$ .

MTGP genera una sequenza di elementi in  $\mathbb{F}_2$  secondo la ricorsione

$$\mathbf{x}_{N+i} := g(\mathbf{x}_{M+i}, \mathbf{x}_{1+i}, \bar{\mathbf{x}}_i^{w-r}), \quad (2.17)$$

ove  $M$  è un intero che soddisfa l'equazione precedente.

La funzione alle ricorrenze  $g$ , lineare in  $\mathbb{F}_2$ , è univocamente determinata da sei parametri interi  $N, M, w, r, sh1, sh2$  ed una matrice  $\mathbf{R} \in \mathbb{R}^{4 \times w}$ .

**Definizione 2.3.1.** La ricorsione del MTGP, definita come segue, fa ricorso alle variabili temporanee  $t$  e  $u$ , interi a  $w$ -bit, identificati mediante vettori riga in  $\mathbb{F}_2^w$ :

$$\begin{aligned} \mathbf{t} &\leftarrow \mathbf{x}_{1+i} \oplus (\mathbf{x}_i^{w-r} \mid \mathbf{0}^r) \\ \mathbf{t} &\leftarrow \mathbf{t} \oplus (\mathbf{t} \ll sh1) \\ \mathbf{u} &\leftarrow \mathbf{t} \oplus (x_{M+i} \gg sh2) \\ \mathbf{x}_{N+i} &\leftarrow \mathbf{u} \oplus (\underline{\mathbf{u}}^4 \mathbf{R}) \end{aligned}$$

ove  $\underline{\mathbf{u}}^4$  indica il vettore riga quadridimensionale in  $\mathbb{F}_2$  contenente i quattro bit meno significativi (LSBs) di  $\mathbf{u}$ , e quindi la moltiplicazione  $\underline{\mathbf{u}}^4 \mathbf{R}$  con la matrice  $\mathbf{R}$  ( $4 \times w$ ) produce un vettore  $w$ -dimensionale. I parametri ( $N, M, w, r, sh1, sh2, \mathbf{R}$ ) sono scelti in modo tale da realizzare il periodo  $2^{wN-r} - 1$ .

Analogamente al suo predecessore, il MTGP, come ultimo step del suo algoritmo, sottopone la sequenza  $\mathbf{x}_i$  ad una trasformazione lineare fissata (tempering), per ottenere un migliore  $k(v)$  ( $v = 1, \dots, w$ ).

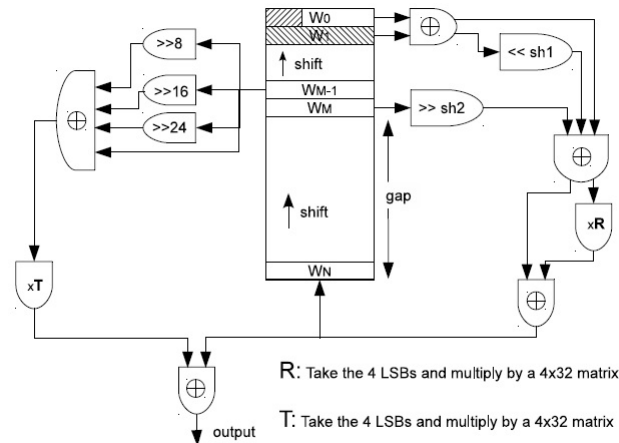


Figura 2.3: Descrizione circuitale del MTGP: il circuito a destra risolve la ricorsione, quello a sinistra effettua il tempering.

**Definizione 2.3.2.** Considerata una variabile intera a  $w$ -bit temporanea  $\mathbf{t}$ , si definisce *tempering* per il generatore MTGP la seguente trasformazione:

$$\mathbf{o}_{N+i} := \mathbf{x}_{N+i} \oplus h(\mathbf{x}_{M-1+i}) \quad (2.18)$$

ove  $\mathbf{o}_{N+i}$  corrisponde all' $(N + i)$ -esimo output ed  $h$  è un'opportuna trasformazione lineare.

- $\mathbf{t} \leftarrow \mathbf{x}_{M-1+i} \oplus (\mathbf{x}_{M-1+i} \gg 16)$
- $\mathbf{t} \leftarrow \mathbf{t} \oplus (\mathbf{t} \gg 8)$
- $\mathbf{o}_{N+i} \leftarrow \mathbf{x}_{N+i} \oplus \underline{\mathbf{t}}^4 \mathbf{T}$

ove  $\mathbf{T}$  è una matrice  $(4 \times w)$  in  $\mathbb{F}_2$  (matrice tempering). La trasformazione lineare contenuta nell'equazione (2.18) corrisponde a  $\underline{\mathbf{t}}^4 \mathbf{T}$ .

## 2.4 Test statistici per il generatore MTGP

Prima di procedere all'utilizzo di un generatore di numeri pseudo-casuali è necessario verificare la qualità della sequenza di numeri prodotta. Si richiede, pertanto, che il generatore in esame soddisfi le seguenti proprietà:

1. *Casualità.* È in grado di produrre una sequenza di variabili aleatorie uniformi, indipendentemente distribuite.
2. *Lungo Periodo.* È in grado di produrre una lunga sequenza di numeri casuali, prima di ripetersi.
3. *Efficienza.* L'esecuzione deve essere rapida e computazionalmente leggera.
4. *Ripetibilità.* Le condizioni iniziali (seed values) determinano completamente la risultante sequenza di variabili aleatorie.
5. *Portabilità.* Sequenze identiche di variabili aleatorie, per un assegnato stato iniziale, devono essere replicabili su diverse macchine e in diversi sistemi operativi.
6. *Omogeneità.* Tutti i sottoinsiemi di bits dei numeri casuali generati devono essere casuali, dal bit più significativo (MSB) al meno significativo (LSB).

Il MTGP, come riportato dagli autori stessi in [17], ha superato i test contenuti nella suite TestU01 [10]. Come ulteriore verifica sulla qualità dei numeri pseudo-casuali generati in più dimensioni, si è sottoposto il generatore MTGP ai test riportati in [14], che rappresentano un'antologia dei test più stringenti contenuti nella suite *Diehard*, di Marsaglia.

Nel presente paragrafo, in seguito ad una concisa introduzione degli strumenti statistici utilizzati per il testing del generatore MTGP, sono presentati i test sopra citati e si confrontano i risultati ottenuti con [14].

### 2.4.1 Strumenti statistici

**Definizione 2.4.1.** L'*ipotesi nulla*  $\mathcal{H}_0$  considerata richiede che i numeri casuali generati  $u_0, u_1, \dots, u_n$  siano variabili aleatorie indipendenti, estratte dalla distribuzione uniforme sull'intervallo  $(0, 1)$ . Soddisfare l'ipotesi  $\mathcal{H}_0$  equivale a richiedere che, per ogni intero  $t > 0$ , il vettore  $(u_0, \dots, u_{t-1})$  sia uniformemente distribuito nel cubo unitario t-dimensionale  $(0, 1)^t$ .

**Notazione:** si denota con  $\Psi_t$  l'insieme *finito* dei vettori t-dimensionali dei  $t$  numeri successivi prodotti dal generatore, generati a partire da qualsiasi stato iniziale ammissibile.

La cardinalità di  $\Psi_t$  non può essere superiore al numero di stati iniziali ammissibili per il generatore. Assumendo che lo stato iniziale sia scelto in modo casuale, i vettori generati in  $\Psi_t$  non possono replicare esattamente la distribuzione uniforme in  $(0, 1)^t$ , ma dovrebbero ben approssimarla.

**Definizione 2.4.2.** Un *test statistico* per generatori di numeri pseudo-casuali è uno *stimatore*  $Y$ , funzione di un numero finito di numeri reali  $u_n$ , output del generatore, la cui distribuzione, sotto ipotesi nulla, è nota o può essere ben approssimata.

Il numero di test che possono essere effettuati è potenzialmente infinito ed ogni test può evidenziare difetti diversi del generatore. Tuttavia, non esiste un test universale (o una batteria di test) che possa garantire che il generatore considerato sia pienamente affidabile per ogni tipo di applicazione.

Pur non essendo possibile sancire l'infallibilità di un generatore di numeri pseudo-casuali, è certamente possibile migliorarne la confidenza. In sintesi, la cifra di merito con cui qualificare i generatori di numeri casuali consiste nella capacità di superare test statistici di diverse difficoltà. Mentre i generatori di scarsa qualità falliranno persino i test più semplici, i buoni generatori falliranno solamente i test più complicati, che rappresentano scenari molto poco frequenti. Da studi condotti in [10] risulta che i generatori di numeri pseudo-casuali dotati di lungo periodo, buona struttura dell'insieme  $\Psi_t$  e basati su equazioni alle ricorrenze non banali, sono più inclini a superare la maggior parte dei test.

In statistica classica, nel contesto di un *test di ipotesi*, è necessario selezionare aprioristicamente un'*area di rifiuto*  $R$ , la cui probabilità sotto l'ipotesi

$\mathcal{H}_0$  corrisponde al livello di significatività (e.g., 0.05 or 0.01), e *rifiuta*  $\mathcal{H}_0$  se e solo se  $Y \in R$ . Questa procedura potrebbe non essere appropriata qualora la dimensione del campione fosse fissata, in particolare se il campione è piccolo. Gli autori di [10] suggeriscono, per testare un generatore di numeri pseudo-casuali, di considerare campioni molto vasti, con dimensione variabile. In tal modo, invece di dover selezionare un livello di significatività ed un area di rifiuto, si calcola semplicemente il p-value del test, definito come segue:

$$p = P[Y \geq y | \mathcal{H}_0] \quad (2.19)$$

dove  $y$  è il valore assunto dallo stimatore  $Y$ . Se  $Y$  ha distribuzione continua,  $p$  è una variabile aleatoria in  $U(0,1)$  sotto l'ipotesi  $\mathcal{H}_0$ . Per alcuni test il p-value può essere interpretato come una misura di uniformità: sarà prossimo ad 1 se il generatore produce risultati eccessivamente uniformi, mentre sarà prossimo a 0 in caso contrario.

Se il  $p$ -value è estremamente piccolo, e.g. meno di  $10^{-10}$ , allora è evidente che il generatore abbia fallito il test, ma se non è molto vicino a 0 o ad 1 allora il test non ha rivelato anomalie. Se il  $p$ -value è sospetto, ma senza indicare inequivocabilmente un rifiuto ( $p = 0.002$ , per esempio), allora il test può essere replicato indipendentemente, producendo altre sequenze di output mediante il medesimo generatore, finché il fallimento non diventa evidente o il  $p$ -value sospetto non viene confutato.

## 2.4.2 Test effettuati

### GCD test

Si considerino  $u$  e  $v$ , interi a 32-bit ed i tre risultati prodotti dall'applicazione dell'algoritmo di Euclide alla coppia  $(u, v)$ :

- $k$ : il numero di iterazioni necessarie per trovare il GCD (Greatest Common Divisor)
- una sequenza, di lunghezza variabile, di quozienti parziali
- il GCD, inteso come valore finale di  $u$

Reiterando l'algoritmo per diverse combinazioni della coppia  $(u, v)$  saranno prodotte tre liste di numeri interi positivi:

1. una lista di  $k$ , indipendenti ed identicamente distribuiti (i.i.d.)
2. una lista di sequenze, di lunghezza variabile, di quozienti parziali, i cui elementi non sono i.i.d.
3. una lista di GCD finali, i.i.d.

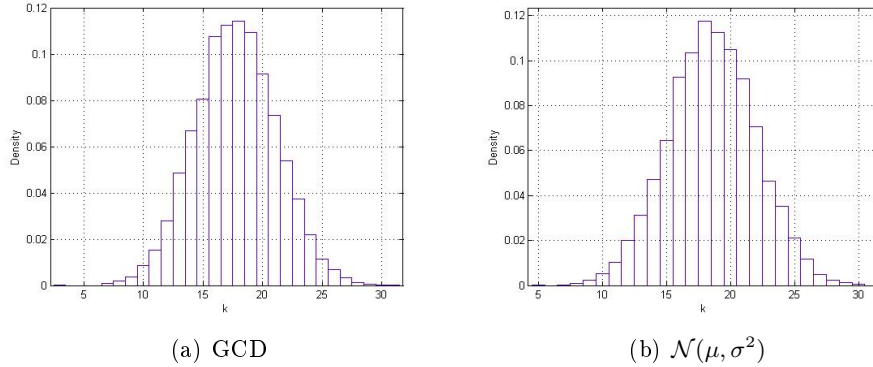


Figura 2.4: Confronto tra le due distribuzioni

Come riportato in [14], di  $k$  non si conosce la distribuzione analitica, ma studi empirici estensivi dimostrano che  $k$  è approssimativamente distribuita normalmente, con media  $\mu = 18.5785$  e deviazione standard  $\sigma = 3.405$ . Si può affermare che l'MTGP ha superato il GCD Test: si è confrontato, in MATLAB, la distribuzione empirica dei  $k$  calcolati a partire da 10000 numeri reali in  $[0, 1)$  con la distribuzione normale prodotta da MATLAB su un campione di medesima dimensione, con  $\mu = 18.5785$  e  $\sigma = 3.405$ . Il confronto grafico delle due distribuzioni è riportato in figura 2.4. Si ottiene che  $\mu_{GCD} = 17.7785$  e  $\sigma_{GCD} = 3.4148$ .

### Anderson-Darling $A^2$

Il problema di inferenza statistica considerato è quello di verificare l'ipotesi secondo cui un campione sia stato estratto da una popolazione con una specifica distribuzione cumulata  $F(x)$ . La procedura di test, proposta da Anderson e Darling in [1], è la seguente: siano  $x_1 \leq x_2 \leq \dots \leq x_n$  un campione di  $n$  osservazioni, disposte in ordine crescente, e sia  $u_i = F(x_i)$ . Si calcoli dunque

$$A_n^2 = -n - \frac{1}{n} \sum_{j=1}^n (2j-1) [\ln(u_j) + \ln(1 - A_{n-j+1})] \quad (2.20)$$

Se  $A_n^2$  è maggiore del valore critico, l'ipotesi deve essere rifiutata. Questa procedura può essere utilizzata per rifiutare l'ipotesi ogniqualvolta la distribuzione teorica differisca sensibilmente dalla distribuzione empirica, in particolar modo se differisce nelle code. Nella seguente tabella, redatta da Anderson e Darling in [1], sono riportati i valori critici asintotici per i più comuni livelli di significatività:



Livello di Significatività	Valore Critico
.10	1.933
.05	2.492
.01	3.857

Sono stati effettuati 1000 test di Anderson-Darling, in ognuno dei quali sono state prodotte 10000 variabili realizzazioni della variabile aleatoria uniforme.

Il  $p$ -value medio calcolato è di 0.9711. Si può dunque affermare che il generatore MTGP ha superato il test di Anderson-Darling.

### Birthday Spacings

Si divida l'intervallo  $[0, 1)$  in  $d$  segmenti uguali, determinando una partizione di  $[0, 1)^t$  in  $k = d^t$  ipercubi di uguale dimensione. Si numerino questi ipercubi da 0 a  $k - 1$  in ordine lessicografico: all'ipercubo avente l'angolo inferiore sinistro in  $(i_0/d, \dots, \beta_{t-1}/d)$  corrisponde l'indice  $i_{t-1} + di_{t-2} + \dots + d^{t-1}i_0$ . Si generino  $nt$  numeri casuali  $U_0, \dots, U_{nt-1}$ , si costruiscano i punti  $V_i = (U_{ti}, \dots, U_{ti+t-1})$ ,  $i = 0, \dots, n - 1$  e siano  $I_1 \leq I_2 \leq \dots \leq I_n$  gli indici degli ipercubi in cui sono contenuti questi punti, ordinati in modo crescente. Si calcolino gli *spacings*  $S_j = I_{j+1} - I_j$  per  $j = 1, \dots, n - 1$  e sia  $Y$  il numero di collisioni tra gli spacings, cioè il numero di valori di  $j$  in  $1, \dots, n - 2$  tali che  $S_{(j+1)} = S_{(j)}$ , ove  $S_{(1)}, \dots, S_{(n-1)}$  sono gli spacings ordinati.

Gli  $n$  punti possono essere interpretati come i compleanni di  $n$  persone in un anno composto da  $k$  giorni, da cui il nome di *birthday spacings*.

Knuth ha dimostrato che  $Y$ , se  $n$  è grande e  $\lambda = n^3/(4k)$  è piccolo, è approssimativamente distribuita come una variabile aleatoria di Poisson di parametro  $\lambda$ , sotto l'ipotesi  $H_0$ .

Nella seguente tabella è riportato l'esito medio su 100 Birthday test effettuati, relativamente ad un'istanza del generatore Mersenne Twister.

Nella prima riga si riportano gli spacings (10+ indica  $\geq 10$ ). Nella seconda riga si riportano le frequenze teoriche attese e nella terza le frequenze campionarie ottenute.

0	1	2	3	4	5	6	7	8	9	10+
92	366	733	977	976	782	521	298	149	66	27
91	377	779	952	931	787	501	321	156	69	36

Relativamente ai risultati ottenuti, si è calcolato il  $p$ -value complessivo: 0.7510.

Si può dunque affermare che il generatore Mersenne Twister ha superato il Birthday Spacings test.

## Gorilla Test

Il Gorilla Test è una versione potenziata del Monkey Test, contenuto nella suite di test statistici *Diehard*, realizzata da Marsaglia. Una nota di colore, dello stesso autore, riporta

*Few images invoke the mysteries and ultimate certainties of a sequence of random events as well as that of the proverbial monkey at a typewriter.*

L'idea è quella di utilizzare il generatore di numeri casuali per produrre una sequenza di lettere di un alfabeto, per poi studiare la frequenza delle parole a  $k$ -lettere contenute nella sequenza.

Probabilmente, il modo migliore per verificare la distribuzione delle parole a  $k$ -lettere prodotta dalla 'scimmia' è quello di creare una cella per ogni parola ammissibile e contare il numero di apparizioni di ogni parola all'interno della lunga sequenza di lettere generata.

Un'alternativa, più leggera computazionalmente, è quella di contare il numero delle parole a  $k$ -lettere mancanti nella stringa digitata dalla scimmia.

Per interi a 32-bit prodotti dal generatore, si specifichi una posizione di bit delle 32 ammissibili, assumendo in posizione 0 il bit più significativo (MSB) ed in posizione 31 il meno significativo (LSB). Si costruisca una successione di  $2^{26} + 25$  bits, composta dai valori assunti dal bit designato in ognuno dei  $2^{26} + 25$  numeri interi calcolati dal generatore. Se  $x$  è il numero delle parole a 26-bit che non compaiono nella sequenza, allora  $x$  dovrebbe essere approssimativamente distribuito come una Normale con media 24687971 e deviazione standard 4170.

Pertanto,  $\Phi((x - 24687971)/4170)$  è distribuita come una variabile aleatoria uniforme in  $[0, 1)$ , i.e., costituisce un  $p$ -value per il test, ove  $\Phi$  rappresenta la distribuzione Normale standard.

Sono riportati i  $p$ -values medi, calcolati su 100 simulazioni, relativi alla distribuzione di ogni bit.

$$p[< b32 > \dots < b29 >] = 0.5114, 0.4894, 0.4902, 0.4741,$$

$$p[< b28 > \dots < b25 >] = 0.4746, 0.4865, 0.5588, 0.4753$$

$$p[< b24 > \dots < b21 >] = 0.5184, 0.5056, 0.5209, 0.4821,$$

$$p[< b20 > \dots < b17 >] = 0.5148, 0.5015, 0.4799, 0.4509$$

$$p[< b16 > \dots < b13 >] = 0.5128, 0.5088, 0.5376, 0.5588,$$

$$p[< b12 > \dots < b9 >] = 0.5013, 0.4994, 0.4942, 0.5327$$

$$p[< b8 > \dots < b5 >] = 0.4941, 0.5087, 0.5142, 0.4426,$$

$$p[< b4 > \dots < b1 >] = 0.4929, 0.5354, 0.5004, 0.4838$$

Non essendovi evidenza di p-values uguali a 0, 1 oppure sospetti, in alcun bit, si può concludere che il generatore MTGP ha superato il Gorilla Test. Inoltre, in linea con le aspettative, si nota che il p-value, al crescere del numero di simulazioni, tende a 0.5, valore atteso della distribuzione uniforme in  $[0, 1)$ , confermando l'equidistribuzione di ogni bit nella sequenza generata.

### 2.4.3 Commento sui test effettuati

Secondo il criterio di valutazione, proposto in [14], MTGP può essere considerato un buon generatore di numeri pseudo-casuali, avendo ottenuto esito positivo in tutti i test della suite.

Ad ulteriore verifica della qualità dei numeri generati, è stato effettuato il test di Anderson-Darling [1], ottenendo esito positivo.

Nella seguente figura, a titolo esemplificativo, è riportata la distribuzione di 10000 punti nel cubo unitario.

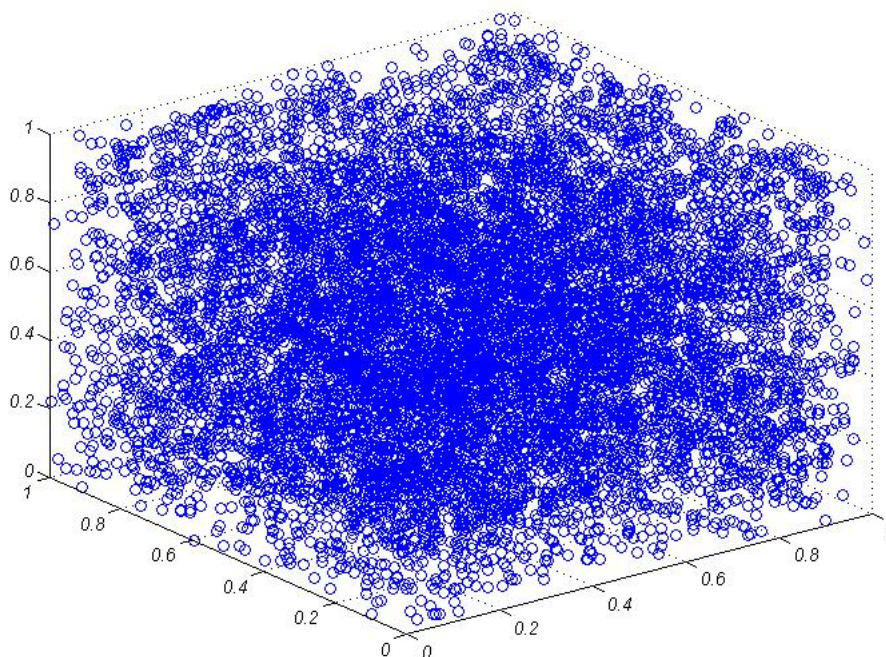


Figura 2.5: Simulazione di 10000 punti nel cubo unitario, generati mediante MTGP.

## 2.5 Generazione di una distribuzione Gaussiana

Nella grande maggioranza dei modelli per la generazione di scenari e per il pricing è necessario ricorrere a distribuzioni probabilistiche che non siano quella uniforme. A tal fine, nella presente sezione è presentato lo stato dell'arte circa gli algoritmi di mapping tra la distribuzione uniforme e la distribuzione Gaussiana, avente media e varianza fissate, necessaria per la modellazione del moto Browniano. In particolare, si descrivono gli algoritmi Box-Muller e Ziggurat, utilizzato di default da Matlab, confrontandone le performance in applicazioni di calcolo seriale e calcolo parallelo.

### 2.5.1 Box-Muller

Il seguente approccio può essere adottato per generare due variabili aleatorie aventi medesima distribuzione normale, a partire da due numeri casuali. Siano  $U_1$  e  $U_2$  variabili aleatorie indipendenti, aventi la medesima distribuzione nell'intervallo  $(0, 1)$ . Si considerino le variabili aleatorie:

$$X_1 = \sqrt{-2 \ln U_1} \cos 2\pi U_2 \quad X_2 = \sqrt{-2 \ln U_1} \sin 2\pi U_2 \quad (2.21)$$

Allora  $(X_1, X_2)$  sono due variabili aleatorie indipendenti aventi distribuzione normale  $\mathcal{N}(0, 1)$ . Si definisce la nuova funzione

$$U(R) = \frac{1}{2\pi} \int_{x^2+y^2 \leq R^2} e^{-\frac{x^2+y^2}{2}} dx dy. \quad (2.22)$$

Introducendo le sostituzioni

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \\ x^2 + y^2 &= r^2 \\ \arctan \frac{y}{x} &= \theta \\ \left| \frac{\partial(x, y)}{\partial(r, \theta)} \right| &= r, \end{aligned}$$

si ottiene che

$$U(R) = \frac{1}{2\pi} \int_0^{2\pi} d\theta \int_0^R r e^{-\frac{r^2}{2}} dr = \int_0^{\frac{R^2}{2}} e^{-u} du = 1 - e^{-\frac{R^2}{2}}$$

$U(R)$  è una funzione non decrescente che soddisfa

$$\begin{aligned} \lim_{R \rightarrow 0} U(R) &= 0 \\ \lim_{R \rightarrow \infty} U(R) &= 1 \end{aligned}$$

Si consideri  $p \in [0, 1)$ , allora

$$U(R) = p \Rightarrow R = \sqrt{-2 \ln(1-p)}$$

Fissando  $s = 1 - p \in [0, 1)$  e  $t \in [0, 1)$ , si ottiene

$$x = \begin{cases} \sqrt{-2 \ln s} \cos 2\pi t, \\ \sqrt{-2 \ln s} \sin 2\pi t, \end{cases} \quad (2.23)$$

Questo è lo schema per generare variabili aleatorie distribuite come  $\mathcal{N}(0, 1)$ . Più in generale, la variabile stocastica  $z \sim \mathcal{N}(\mu, \sigma^2)$  è data da

$$z = \mu + \sigma \sqrt{-2 \ln s} \cos 2\pi t$$

oppure

$$z = \mu + \sigma \sqrt{-2 \ln s} \sin 2\pi t$$

*Dimostrazione.* Si consideri la funzione caratteristica di una distribuzione gaussiana standard:

$$\phi(\theta) = \mathbb{E}[e^{i\theta x}] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{x^2}{2} + i\theta x} dx = e^{-\frac{\theta^2}{2}}.$$

La funzione caratteristica di una variabile aleatoria generata con il metodo Box-Muller,  $\phi_{BM}(\theta)$ , è rappresentata come segue:

$$\begin{aligned} \phi_{BM}(\theta) &= \mathbb{E}[e^{i\theta x}] = \int_0^1 ds \int_0^1 \exp[i\theta \sqrt{-2 \ln s} \cos(2\pi t)] dt = \\ &= \int_0^1 ds \int_0^1 \sum_{n=0}^{\infty} \frac{(i\theta \sqrt{-2 \ln s} \cos(2\pi t))^n}{n!} dt, \end{aligned}$$

dove  $s$  e  $t$  dipendono dalle variabili uniformi in  $[0, 1)$  indipendenti ed identicamente distribuite. L'integrale di  $s$  è calcolato come segue:

$$\begin{aligned} \int_0^1 (-2 \ln s)^{\frac{n}{2}} ds &= 2^{\frac{n}{2}} \int_0^1 (-\ln s)^{\frac{n}{2}} ds \\ &= 2^{\frac{n}{2}} \int_0^{\infty} (e^{-u}) u^{\frac{n}{2}} du = 2^{\frac{n}{2}} \Gamma\left(\frac{n}{2} + 1\right), \end{aligned}$$

dove  $u = -\ln s$  e  $\Gamma(z) = \int_0^{\infty} u^{z-1} e^{-u} du$ . In aggiunta, l'integrale in  $t$  è calcolato come:

$$\int_0^1 \cos^n(2\pi t) dt = \begin{cases} 0, & \text{se } n \text{ dispari} \\ 4 \int_0^{\frac{\pi}{2}} \frac{du}{2\pi} \cos^n u, & \text{se } n \text{ pari} \end{cases}$$

dove  $u = 2\pi t$ . Nel caso di  $n$  pari,

$$\frac{2}{\pi} \int_0^{\frac{\pi}{2}} \cos^n u \, du = \frac{1}{\pi} B\left(\frac{n+1}{2}, \frac{1}{2}\right) = \frac{1}{\pi} \frac{\Gamma(\frac{n+1}{2}) \Gamma(\frac{1}{2})}{\Gamma(\frac{n+1}{2} + \frac{1}{2})}$$

ove

$$B(p, q) = 2 \int_0^{\frac{\pi}{2}} \cos^{2p-1}(x) \sin^{2q-1}(x) \, dx = \frac{\Gamma(p) \Gamma(q)}{\Gamma(p+q)}.$$

Se  $n = 2m$ , la funzione caratteristica del metodo di Box-Muller  $\phi_{BM}(\theta)$  è calcolata come

$$\begin{aligned} \phi_{BM}(\theta) &= \sum_{m=0}^{\infty} \frac{(i\theta)^{2m}}{(2m)!} 2^m \Gamma(m+1) \frac{1}{\pi} \frac{\Gamma(m + \frac{1}{2}) \Gamma(\frac{1}{2})}{\Gamma(m+1)} \\ &= \sum_{m=0}^{\infty} \frac{(-\theta^2)^m}{\Gamma(2m+1)} \frac{2^m \Gamma(\frac{1}{2}) \Gamma(2m) 2\sqrt{\pi}}{\pi 2^{2m} \Gamma(m)} \\ &= \sum_{m=0}^{\infty} \left(\frac{\theta^2}{2}\right)^m \frac{2\Gamma(2m)}{\Gamma(2m+1)} \frac{\Gamma(\frac{1}{2})}{\sqrt{\pi} \Gamma(m)} \\ &= \sum_{m=0}^{\infty} \left(-\frac{\theta^2}{2}\right)^m \frac{2}{2m} \frac{1}{\Gamma(m)} \\ &= \sum_{m=0}^{\infty} \frac{1}{m!} \left(-\frac{\theta^2}{2}\right)^m \\ &= e^{-\frac{\theta^2}{2}}, \end{aligned}$$

ove  $z! = \Gamma(z+1)$ ,  $\Gamma(z+1) = z\Gamma(z)$ ,  $\Gamma(\frac{1}{2}) = \sqrt{\pi}$  e

$$\Gamma\left(m + \frac{1}{2}\right) = \frac{\Gamma(2m) 2\sqrt{\pi}}{2^{2m} \Gamma(m)}.$$

Poiché risulta  $\phi(\theta) = \phi_{BM}(\theta)$ , si ha che la variabile aleatoria definita in (2.23) è distribuita come una normale standard.  $\square$

Nella seguente figura si riporta il confronto sinottico tra le distribuzioni ottenute generando 100000 campioni, nel primo caso mediante l'algoritmo *randn* di MATLAB, nel secondo caso mediante Mersenne Twister + Box-Muller.

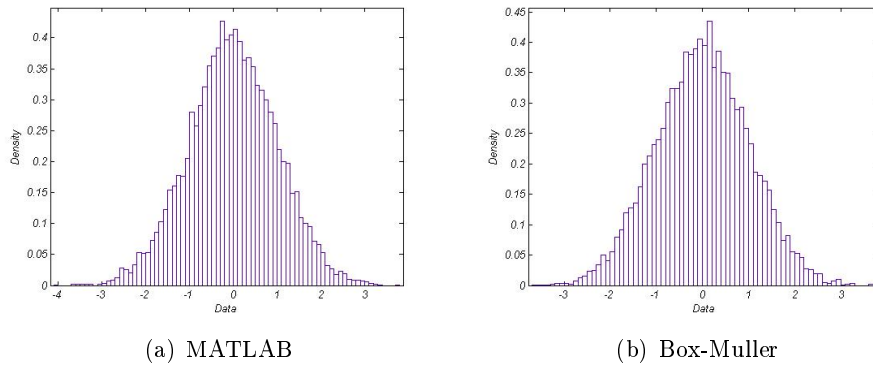


Figura 2.6: Confronto sinottico tra la funzione *randn* di MATLAB e l'algoritmo Box-Muller implementato

Dal calcolo dei principali momenti campionari della variabile aleatoria simulata non si notano differenze apprezzabili tra i due metodi. Si conclude, pertanto, che l'algoritmo implementato replica correttamente le performances di MATLAB.

### 2.5.2 Ziggurat

Sia  $\mathcal{C}$  l'insieme dei punti  $(x, y)$  sottostanti alla curva  $y = f(x)$ , avente area finita. Sia  $\mathcal{Z}$  l'insieme dei punti contenente  $\mathcal{C}$  :  $\mathcal{Z} \supset \mathcal{C}$ . L'idea del metodo di rifiuto è la seguente: scegliere uniformemente i punti  $(x, y)$  da  $\mathcal{Z}$  finché non si ottiene un punto contenuto in  $\mathcal{C}$  e restituire  $x$ , come la variabile aleatoria richiesta. La densità di tale  $x$  sarà  $cf(x)$ , ove  $c$  è la costante di normalizzazione che rende  $cf(x)$  la densità richiesta. I principali criteri di scelta per l'insieme di copertura  $\mathcal{Z}$  sono i seguenti:

1. Generare in modo semplice e veloce il punto casuale  $(x, y)$  da  $\mathcal{Z}$ ;
2. Disporre di un algoritmo efficiente per stabilire la regione di appartenenza del punto generato in 1.;
3. Massimizzare l'efficienza della procedura di rifiuto  $\frac{\text{area}\mathcal{C}}{\text{area}\mathcal{Z}} \rightarrow 1$ .

La Ziggurat partiziona la densità Normale standard in blocchi orizzontali di uguale area. Tutti i blocchi sono scatole rettangolari, eccezion fatta per la base, costituita dai blocchi contenenti la residua densità.

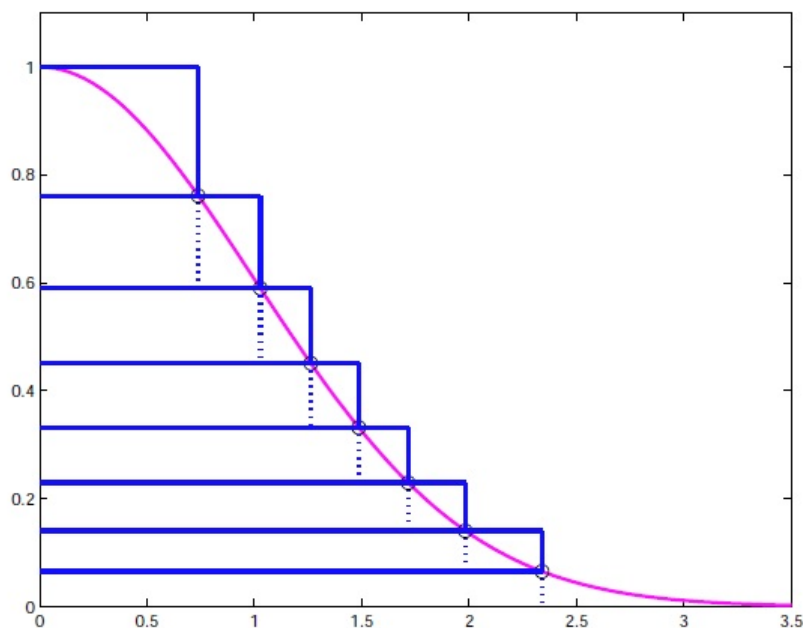


Figura 2.7: Ziggurat ad 8 blocchi sulla densità Normale standard

In figura 2.7 è riportato un esempio di Ziggurat con 3 partizioni, denominate  $B_0$  (la base),  $B_1$  e  $B_2$  (la cima). Imporre che tali partizioni abbiano medesima area implica

$$x_2[f(x_3) - f(x_2)] = x_1[f(x_2) - f(x_1)] = x_1 f(x_1) + \int_{x_1}^{+\infty} f(x) dx = V.$$

In generale, dividendo la densità in  $C$  blocchi, la partizione può essere calcolata risolvendo la seguente equazione non lineare:

$$x_{C-1}(r)[f(0) - f(x_{C-1}(r))] - V(r) = 0, \quad (2.24)$$

ove  $x_C = 0$ ,  $r$  denota la massima ascissa di  $x_1$ , e

$$V(r) = r f(r) + \int_r^{+\infty} f(x) dx.$$

Il valore di  $x_{C-1}(r)$  può essere calcolato, per sostituzione, da:

$$x_i = f^{-1}(f(x_{i-1}) + V(r)/x_{i-1}), \quad i = 2, \dots, C-1, \dots, x_1 = r.$$

Si denoti con  $R$  la soluzione di (2.24).

Risulta dunque immediato il metodo di rifiuto per la Ziggurat:

1. Si consideri una partizione  $B_i$ , scelta in modo casuale con probabilità  $1/C$



2. Si consideri un numero casuale contenuto nella partizione  $B_i$ , tale che  $z = U_0 x_i$  per  $i > 0$  e  $z = U_0 V / f(x_1)$  per  $i = 0$ .
3. Se  $z < x_{i+1}$  accettare  $z$ .
4. Se  $i = 0$ , accettare  $v$ , estratta dalla coda secondo l'algoritmo [Marsaglia 1964], riportato in appendice B.  
Altrimenti, se  $i > 0$  e  $U_1[f(x_i) - f(x_{i+1})] < f(z) - f(x_{i+1})$  accettare  $z$

Nella seguente figura si riporta il confronto sinottico tra le distribuzioni ottenute generando 100000 campioni, nel primo caso mediante l'algoritmo *randn* di MATLAB, nel secondo caso mediante Mersenne Twister + Ziggurat.

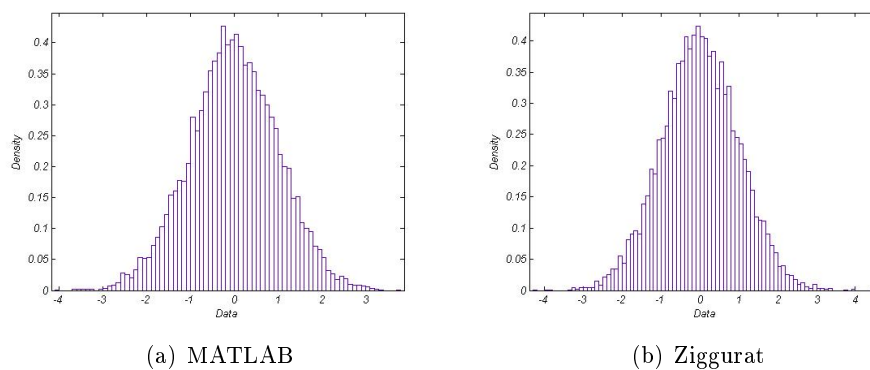


Figura 2.8: Confronto sinottico tra la funzione *randn* di MATLAB e l'algoritmo Ziggurat implementato

Dal calcolo dei principali momenti campionari della variabile aleatoria simulata non si notano differenze apprezzabili tra i due metodi. Si conclude, pertanto, che l'algoritmo implementato replica correttamente le performances di MATLAB.

### 2.5.3 Confronto tra i due metodi

L'algoritmo Box-Muller, non introducendo alcuna approssimazione, è in grado di replicare il corretto comportamento della distribuzione Gaussiana, anche nelle sue code. Tuttavia, la procedura di rifiuto risulta moderatamente costosa, in quanto, definita  $e$  l'efficienza di tale procedura, si ottiene:

$$e = \frac{\pi}{4} \approx 0.7854.$$

Di conseguenza, circa il 21% dei punti uniformemente distribuiti all'interno del quadrato unitario sono scartati.

L'algoritmo Ziggurat, invece, pur introducendo un'approssimazione non trascurabile per la gestione delle code della densità Gaussiana, implementa una procedura di rifiuto avente efficienza

$$e = \frac{\sqrt{2\pi}}{2nv}$$

, ove  $n$  è il numero di rettangoli che costituiscono la Ziggurat e  $v$  è la corrispondente area.

Nel caso di una Ziggurat composta da 256 rettangoli, ognuno di area  $v = 0.00492867323399$ , si ottiene un'efficienza del 99.33%.

Tuttavia, nel caso di implementazioni CUDA, il metodo Ziggurat non risulta applicabile, in quanto *table dependent*. Per circa il 2% dei numeri uniformi generati è necessario ricorrere all'algoritmo riportato nell'appendice B, che implica la generazione di almeno un'altra variabile uniforme. Questo tipo di eccezione, in un linguaggio di programmazione procedurale, è accettabile e risulta semplice da gestire, ma, nella GPU, la penalizzante performance della gestione dell'eccezione si applica a tutti i thread dello stesso warp<sup>3</sup>, anche se solo un thread la invoca. Se la dimensione del warp è 32 e la probabilità di incappare in una routine penalizzata è del 2%, allora la probabilità che un warp qualsiasi debba effettuare la routine rallentata risulta  $(1 - 0.02)^{32} \approx 52\%$ . Pertanto, per la generazione di distribuzioni Gaussiane su architetture CUDA, è fortemente consigliata l'adozione dell'algoritmo Box-Muller.

---

<sup>3</sup>In CUDA, si definisce *warp* un insieme di 32 threads, ovvero sia la dimensione minima dei dati processati nella modalità Single Instruction Multiple Data da un multiprocessore CUDA

## Capitolo 3

# Il modello $G2++$ Single Currency

Nel presente Capitolo si considera il modello  $G2++$  per la descrizione dell'evoluzione temporale del tasso short  $r$ , la cui formulazione è riportata in [6]. Nella prima sezione si riportano i risultati fondamentali contenuti in [6], utilizzati per il pricing dei principali prodotti finanziari (Zero-Coupon Bond, Cap e Swaptions), necessari per la calibrazione del modello su dati storici e per il pricing di prodotti più complessi.

Nella seconda sezione si considerano le formule di pricing per la Swaption Europea proposte da Brigo e Mercurio in [6] e da Choi e Shin in [8]. Per quest'ultima formula è necessario ri-definire alcune notazioni precedentemente utilizzate, in quanto il modello in [8] si dimostra differire dal modello in [6] per uno shift deterministico.

Il codice MATLAB relativo alla formula per la Swaption Europea secondo Choi e Shin è riportato in appendice D.

In coda alla sezione si effettua il confronto tra le due differenti formulazioni.

### 3.1 Descrizione della dinamica

Il modello  $G2++$  descrive l'evoluzione temporale del tasso d'interesse mediante un processo dato dalla somma di due fattori Gaussiani correlati e di una funzione deterministica, opportunamente scelta per raccordarsi perfettamente con la struttura a termine corrente dei fattori di sconto. Il modello risulta trattabile analiticamente, in quanto è possibile ricavare, mediante procedure numeriche efficienti e veloci, soluzioni in forma chiusa per il pricing di Zero-Coupon Bond, opzioni Vanilla scritte su Zero-Coupon Bond, Cap, floors e prodotti esotici.

Il modello  $G2++$  è largamente utilizzato, nonostante possa restituire tassi negativi in output. La rappresentazione della dinamica del tasso mediante

due equazioni stocastiche correlate, dette *fattori di rischio*, permette di descrivere in modo più accurato la variabilità dei tassi di mercato, rispetto ai modelli ad un fattore. Si assuma che la dinamica del tasso a breve istantaneo sia un processo stocastico, rispetto alla misura risk-neutral  $\mathbb{Q}$ , dato da:

$$r(t) := x(t) + y(t) + \varphi(t), \quad r(0) = r_0, \quad (3.1)$$

ove i processi  $\{x(t) : t \geq 0\}$  e  $\{y(t) : t \geq 0\}$  soddisfano

$$\begin{aligned} dx(t) &= -ax(t)dt + \sigma dW_1(t), \quad x(0) = 0, \\ dy(t) &= -by(t)dt + \eta dW_2(t), \quad y(0) = 0, \end{aligned} \quad (3.2)$$

e  $(W_1, W_2)$  è un moto Browniano bidimensionale, con correlazione istantanea  $\rho$  tale che

$$dW_1(t) dW_2(t) = \rho dt,$$

$r_0, a, b, \sigma, \eta$  sono costanti positive, e  $-1 \leq \rho \leq 1$ .

La funzione  $\varphi$  è deterministica e ben definita sull'intervallo temporale  $[0, T^*]$ , ove  $T^*$  è tipicamente fissato a 10, 30 o 50 anni. In particolare,  $\varphi(0) = r_0$ . Si denota con  $\mathcal{F}_t$  la sigma-algebra generata dalla coppia  $(x, y)$  fino al tempo  $t$ . Dalla semplice integrazione delle equazioni (3.2), per ogni  $s < t$ , si ottiene

$$\begin{aligned} r(t) &= x(s) e^{-a(t-s)} + y(s) e^{-b(t-s)} \\ &+ \sigma \int_s^t e^{-a(t-u)} dW_1(u) + \eta \int_s^t e^{-b(t-u)} dW_2(u) + \varphi(t), \end{aligned} \quad (3.3)$$

intendendo che  $r(t)$  condizionato a  $\mathcal{F}_s$  è normalmente distribuito con media e varianza date rispettivamente da:

$$\mathbb{E}\{r(t)|\mathcal{F}_s\} = x(s)e^{-a(t-s)} + y(s)e^{-b(t-s)} + \varphi(t), \quad (3.4)$$

$$\begin{aligned} Var\{r(t)|\mathcal{F}_s\} &= \frac{\sigma^2}{2a} \left[1 - e^{-2a(t-s)}\right] + \frac{\eta^2}{2b} \left[1 - e^{-2b(t-s)}\right] \\ &+ 2\rho \frac{\sigma\eta}{a+b} \left[1 - e^{-(a+b)(t-s)}\right]. \end{aligned} \quad (3.5)$$

In particolare,

$$r(t) = \sigma \int_0^t e^{-a(t-u)} dW_1(u) + \eta \int_0^t e^{-b(t-u)} dW_2(u) + \varphi(t) \quad (3.6)$$

La dinamica dei processi  $x$  ed  $y$  può essere espressa anche in termini di due moti Browniani indipendenti  $\widetilde{W}_1$  e  $\widetilde{W}_2$ , come segue:

$$\begin{aligned} dx(t) &= -ax(t)dt + \sigma d\widetilde{W}_1(t), \\ dy(t) &= -by(t)dt + \eta\rho\widetilde{W}_1(t) + \eta\sqrt{1-\rho^2}d\widetilde{W}_2(t), \end{aligned} \quad (3.7)$$

ove

$$\begin{aligned} dW_1(t) &= d\widetilde{W}_1(t), \\ dW_2(t) &= \rho d\widetilde{W}_1 + \sqrt{1 - \rho^2} d\widetilde{W}_2(t), \end{aligned} \quad (3.8)$$

in modo da poter scrivere:

$$\begin{aligned} r(t) &= x(s)e^{-a(t-s)} + y(s)e^{-b(t-s)} + \sigma \int_s^t e^{-a(t-u)} d\widetilde{W}_1(u) \\ &\quad + \eta\rho \int_s^t e^{-b(t-u)} d\widetilde{W}_1(u) + \eta\sqrt{1 - \rho^2} \int_s^t e^{-b(t-u)} d\widetilde{W}_2(u) + \varphi(t). \end{aligned}$$

La decorrelazione (3.8), essendo  $\rho$  un parametro fissato e noto, permette di simulare indipendentemente i moti browniani  $\widetilde{W}_1$  e  $\widetilde{W}_2$  mediante il generatore introdotto nel Capitolo 2.

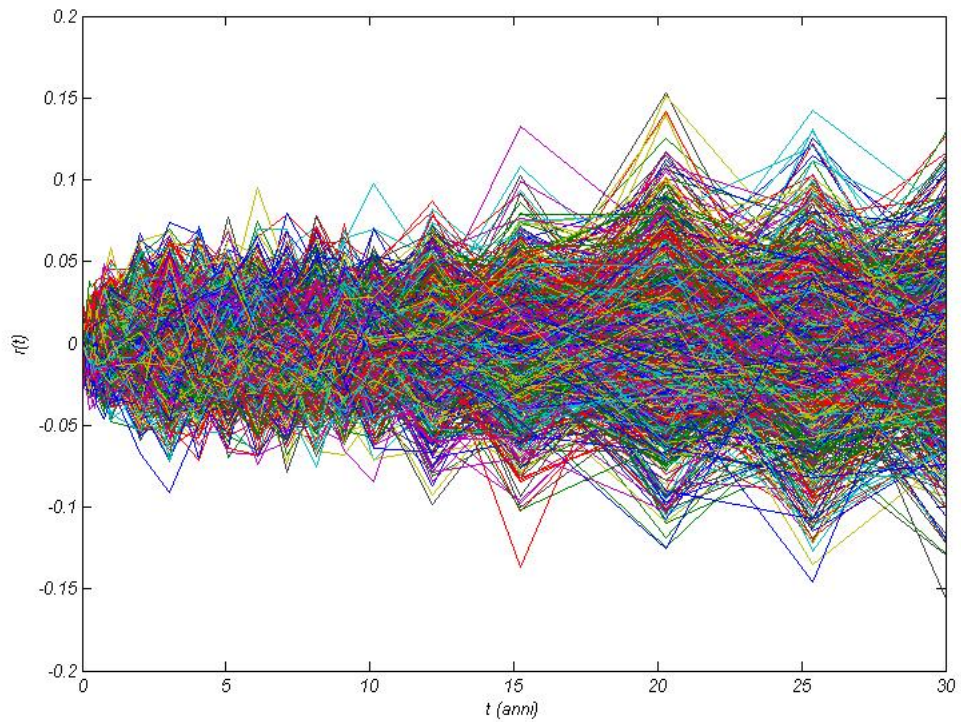


Figura 3.1: plot di 1000 traiettorie di  $r(t)$ -EUR/EONIA, ottenute simulando la soluzione della relativa EDS in tutti i punti della griglia utilizzata da MUREX, interpolati linearmente.

## 3.2 Pricing dei principali prodotti finanziari

Nella presente sezione si derivano e dimostrano le formule di pricing dei principali prodotti finanziari: Zero-Coupon Bond, Call e Put su Zero-Coupon Bond, Caplet e Floorlet, Cap e Floor.

### 3.2.1 Zero-Coupon Bond

Si denota con  $P(t, T)$  il prezzo al tempo  $t$  dello Zero-Coupon Bond, avente maturity in  $T$  e nominale unitario, tale che

$$P(t, T) = \mathbb{E}\{e^{-\int_t^T r_s ds} | \mathcal{F}_t\},$$

ove  $\mathbb{E}$  denota il valore atteso rispetto alla misura risk-neutral  $\mathbb{Q}$ . Per calcolare esplicitamente il precedente valore atteso, è necessario il seguente

**Lemma 3.2.1.** *Per ogni  $t, T$  la variabile aleatoria*

$$I(t, T) := \int_t^T [x(u) + y(u)] du$$

*condizionata alla sigma-algebra  $\mathcal{F}_t$  è distribuita normalmente con media  $M(t, T)$  e varianza  $V(t, T)$ , date rispettivamente da*

$$M(t, T) = \frac{1 - e^{-a(T-t)}}{a} x(t) + \frac{1 - e^{-b(T-t)}}{b} y(t) \quad (3.9)$$

e

$$\begin{aligned} V(t, T) = & \frac{\sigma^2}{a^2} \left[ T - t + \frac{2}{a} e^{-a(T-t)} - \frac{1}{2a} e^{-2a(T-t)} - \frac{3}{2a} \right] \\ & + \frac{\eta^2}{b^2} \left[ T - t + \frac{2}{b} e^{-b(T-t)} - \frac{1}{2b} e^{-2b(T-t)} - \frac{3}{2b} \right] \\ & + 2\rho \frac{\sigma\eta}{ab} \left[ T - t + \frac{e^{-a(T-t)} - 1}{a} + \frac{e^{-b(T-t)} - 1}{b} - \frac{e^{-(a+b)(T-t)} - 1}{a+b} \right] \end{aligned} \quad (3.10)$$

*Dimostrazione.* L'integrazione stocastica per parti implica

$$\int_t^T x(u) du = Tx(T) - tx(t) - \int_t^T u dx(u) = \int_t^T (T-u) dx(u) + (T-t)x(t) \quad (3.11)$$

Per definizione di  $x$ , l'ultimo integrale può essere riscritto sostituendo a  $dx(u)$  la corrispondente dinamica:

$$\int_t^T (T-u) dx(u) = -a \int_t^T (T-u)x(u) du + \sigma \int_t^T (T-u) dW_1(u)$$

e, sostituendo ad  $x(u)$  la corrispondente soluzione, si ottiene:

$$\begin{aligned} \int_t^T (T-u)x(u) du &= x(t) \int_t^T (T-u)e^{-a(u-t)} du \\ &\quad + \sigma \int_t^T (T-u) \int_t^u e^{-a(u-s)} dW_1(s) du. \end{aligned}$$

Calcolando separatamente gli ultimi due integrali, si ottiene

$$-ax(t) \int_t^T (T-u)e^{-a(u-t)} du = -x(t)(T-t) - \frac{e^{-a(T-t)}}{a}x(t)$$

e, integrando per parti,

$$\begin{aligned} &-a\sigma \int_t^T (T-u) \int_t^u e^{-a(u-s)} dW_1(s) du \\ &= -a\sigma \int_t^T \left( \int_t^u e^{as} dW_1(s) \right) du \left( \int_t^u (T-v)e^{-av} dv \right) \\ &= -a\sigma \left[ \left( \int_t^T e^{au} dW_1(t) \right) \left( \int_t^T (T-v)e^{-av} dv \right) \right. \\ &\quad \left. - \int_t^T \left( \int_t^u (T-v)e^{-av} dv \right) e^{au} dW_1(u) \right] \\ &= -a\sigma \int_t^T \left( \int_u^T (T-v)e^{-av} dv \right) e^{au} dW_1(u) \\ &= -\sigma \int_t^T \left[ (T-u) + \frac{e^{-a(T-u)} - 1}{a} \right] dW_1(u), \end{aligned}$$

utilizzando, nell'ultimo passaggio, la relazione

$$\int_u^T (T-v)e^{-av} dv = \frac{(T-u)e^{-au}}{a} + \frac{e^{-aT} - e^{-au}}{a^2}.$$

Sommando i termini precedenti, si ottiene

$$\int_t^T x(u) du = \frac{1 - e^{-a(T-t)}}{a}x(t) + \frac{\sigma}{a} \int_t^T \left[ 1 - e^{-a(T-u)} \right] dW_1(u).$$

L'espressione analoga per  $y$  è ottenuta sostituendo  $a$ ,  $\sigma$  e  $W_1$  rispettivamente con  $b$ ,  $\eta$  e  $W_2$ , i.e.,

$$\int_t^T y(u) du = \frac{1 - e^{-b(T-t)}}{b}y(t) + \frac{\eta}{b} \int_t^T \left[ 1 - e^{-b(T-u)} \right] dW_2(u),$$

in modo che (3.9) sia immediatamente soddisfatta. Dal calcolo della varianza condizionata si ottiene

$$\begin{aligned}
\text{Var} \{I(t, T) | \mathcal{F}_t\} &= \text{Var} \left\{ \frac{\sigma}{a} \int_t^T [1 - e^{-a(T-u)}] dW_1(u) \right. \\
&\quad \left. + \frac{\eta}{b} \int_t^T [1 - e^{-b(T-u)}] dW_2(u) | \mathcal{F}_t \right\} \\
&= \frac{\sigma^2}{a^2} \int_t^T [1 - e^{-a(T-u)}]^2 du + \frac{\eta^2}{b^2} \int_t^T [1 - e^{-b(T-u)}]^2 du \\
&\quad + 2\rho \frac{\sigma\eta}{ab} \int_t^T [1 - e^{-a(T-u)}] [1 - e^{-b(T-u)}] du.
\end{aligned}$$

Semplici integrazioni permettono di ottenere la (3.10).  $\square$

**Teorema 3.2.1.** *Il prezzo al tempo  $t$  di uno Zero-Coupon Bond con maturity in  $T$  e nominale unitario è*

$$\begin{aligned}
P(t, T) &= \exp \left\{ - \int_t^T \varphi(u) du - \frac{1 - e^{-a(T-t)}}{a} x(t) \right. \\
&\quad \left. - \frac{1 - e^{-b(T-t)}}{b} y(t) - \frac{1}{2} V(t, T) \right\}.
\end{aligned} \tag{3.12}$$

*Dimostrazione.* Essendo  $\varphi$  una funzione deterministica, il teorema è diretta conseguenza del Lemma 3.2.1 e del fatto che, se  $Z$  è una variabile aleatoria normale di media  $m_Z$  e varianza  $\sigma_Z^2$ , allora

$$\mathbb{E} \{ \exp(Z) \} = \exp(m_Z + \frac{1}{2} \sigma_Z^2).$$

$\square$

Si assuma che la struttura a termine dei fattori di sconto, attualmente osservata sul mercato, sia una funzione sufficientemente regolare  $T \leftarrow P^M(0, T)$ .

Se si denota con  $f^M(0, T)$  il tasso forward istantaneo al tempo 0 per una maturity  $T$  implicito nella struttura a termine  $T \leftarrow P^M(0, T)$ , i.e.

$$f^M(0, T) = - \frac{\partial \ln P^M(0, T)}{\partial T},$$

allora si ha il seguente

**Corollario 3.2.1.** *Il modello (3.1) si raccorda perfettamente con la struttura a termine dei fattori di sconto osservata sul mercato se e solo se, per ogni  $T$ ,*

$$\varphi(T) = f^M(0, T) + \frac{\sigma^2}{2a^2} (1 - e^{-aT})^2 + \frac{\eta^2}{2b^2} (1 - e^{-bT})^2 + \rho \frac{\sigma\eta}{ab} (1 - e^{-aT})(1 - e^{-bT}), \tag{3.13}$$



i.e., se e solo se

$$\exp\left\{-\int_t^T \varphi(u) du\right\} = \frac{P^M(0, T)}{P^M(0, t)} \exp\left\{-\frac{1}{2}[V(0, T) - V(0, t)]\right\}, \quad (3.14)$$

così che i corrispondenti prezzi degli Zero-Coupon Bond al tempo  $t$  siano dati da

$$P(t, T) = \frac{P^M(0, T)}{P^M(0, t)} \exp\{\mathcal{A}(t, T)\} \quad (3.15)$$

$$\begin{aligned} \mathcal{A}(t, T) := & \frac{1}{2}[V(t, T) - V(0, T) + V(0, t)] \\ & - \frac{1 - e^{-a(T-t)}}{a} x(t) - \frac{1 - e^{-b(T-t)}}{b} y(t). \end{aligned}$$

*Dimostrazione.* Il modello (3.1) si raccorda perfettamente con la struttura a termine dei fattori di sconto osservata sul mercato se e solo se, per ogni maturity  $T \geq T^*$ , il fattore di sconto  $P(0, T)$  prodotto dal modello (3.1) coincide con quello osservato sul mercato, i.e., se e solo se

$$P^M(0, T) = \exp\left\{-\int_0^T \varphi(u) du + \frac{1}{2}V(0, T)\right\}.$$

Si calcoli il logaritmo degli argomenti della precedente equazione e si derivi rispetto a  $T$ , così da ottenere (3.13), notando che

$$\begin{aligned} V(t, T) = & \frac{\sigma^2}{a^2} \int_t^T [1 - e^{-a(T-u)}]^2 du + \frac{\eta^2}{b^2} \int_t^T [1 - e^{-b(T-u)}]^2 du \\ & + 2\rho \frac{\sigma\eta}{ab} \int_t^T [1 - e^{-a(T-u)}] [1 - e^{-b(T-u)}] du. \end{aligned}$$

L'uguaglianza (3.14) segue dal fatto che

$$\begin{aligned} \exp\left\{-\int_t^T \varphi(u) du\right\} &= \exp\left\{-\int_0^T \varphi(u) du\right\} \exp\left\{\int_0^t \varphi(u) du\right\} \\ &= \frac{P^M(0, T) \exp\left\{-\frac{1}{2}V(0, T)\right\}}{P^M(0, t) \exp\left\{-\frac{1}{2}V(0, t)\right\}}. \end{aligned}$$

L'uguaglianza (3.15) segue immediatamente da (3.12) e (3.14).  $\square$

### 3.2.2 Opzioni Europee su Zero-Coupon Bond

Il prezzo al tempo  $t$  di un'opzione Europea con maturity  $T$  e strike  $K$ , scritta su uno Zero-Coupon Bond, con nominale unitario e maturity  $\tau$  è

$$\mathbf{ZBC}(t, T, S, K) = \mathbb{E} \left\{ e^{-\int_t^T r(s) ds} (P(T, S) - K)^+ | \mathcal{F}_t \right\}$$

Per calcolare esplicitamente il precedente valore atteso è necessario cambiare la misura di probabilità. In particolare, per ogni maturity fissata  $T$ , si denota con  $\mathbb{Q}^T$  la misura di probabilità definita dalla seguente derivata di Radon-Nikodym

$$\begin{aligned} \frac{d\mathbb{Q}^T}{d\mathbb{Q}} &= \frac{B(0)P(T, T)}{B(T)P(0, T)} = \frac{\exp \left\{ -\int_0^T r(u) du \right\}}{P(0, T)} \\ &= \frac{\exp \left\{ -\int_0^T \varphi(u) du - \int_0^T [x(u) - y(u)] du \right\}}{P(0, T)} \\ &= \exp \left\{ -\frac{1}{2}V(0, T) - \int_0^T [x(u) + y(u)] du \right\} \end{aligned} \quad (3.16)$$

ove  $B$  rappresenta il money market account. La misura  $\mathbb{Q}^T$  è la risk neutral T-forward measure. Il seguente lemma produce le dinamiche dei processi  $x$  ed  $y$  rispetto a  $\mathbb{Q}^T$ .

**Lemma 3.2.2.** *I processi  $x$  ed  $y$ , rispetto alla forward measure  $\mathbb{Q}^T$  sono rappresentati dalle seguenti dinamiche:*

$$\begin{aligned} dx(t) &= \left[ -ax(t) - \frac{\sigma^2}{a}(1 - e^{-a(T-t)}) - \rho \frac{\sigma\eta}{b}(1 - e^{-b(T-t)}) \right] dt + \sigma dW_1^T(t) \\ dy(t) &= \left[ -by(t) - \frac{\eta^2}{b}(1 - e^{-b(T-t)}) - \rho \frac{\sigma\eta}{a}(1 - e^{-a(T-t)}) \right] dt + \eta dW_2^T(t) \end{aligned} \quad (3.17)$$

ove  $W_1^T$  e  $W_2^T$  sono due moti browniani correlati, rispetto alla misura  $\mathbb{Q}^T$  tali che  $dW_1^T(t)dW_2^T(t) = \rho dt$ .

Inoltre, le soluzioni esplicite delle equazioni (3.17) sono, per  $s \leq t \leq T$ ,

$$\begin{aligned} x(t) &= x(s)e^{-a(t-s)} - M_x^T(s, t) + \sigma \int_s^t e^{-a(t-u)} dW_1^T(u) \\ y(t) &= y(s)e^{-b(t-s)} - M_y^T(s, t) + \eta \int_s^t e^{-b(t-u)} dW_2^T(u), \end{aligned} \quad (3.18)$$

ove

$$\begin{aligned} M_x^T(s, t) &= \left( \frac{\sigma^2}{a^2} + \rho \frac{\sigma\eta}{ab} \right) \left[ 1 - e^{-a(t-s)} \right] - \frac{\sigma^2}{2a^2} \left[ e^{-a(T-t)} - e^{-a(T+t-2s)} \right] \\ &\quad - \frac{\rho\sigma\eta}{b(a+b)} \left[ e^{-b(T-t)} - e^{-bT-at+(a+b)s} \right], \end{aligned}$$

$$M_y^T(s, t) = \left( \frac{\eta^2}{b^2} + \rho \frac{\sigma \eta}{ab} \right) \left[ 1 - e^{-b(t-s)} \right] - \frac{\eta^2}{2b^2} \left[ e^{-b(T-t)} - e^{-b(T+t-2s)} \right] \\ - \frac{\rho \sigma \eta}{a(a+b)} \left[ e^{-a(T-t)} - e^{-aT-bt+(a+b)s} \right]$$

così che, rispetto a  $\mathbb{Q}^T$ , la distribuzione di  $r(t)$ , condizionata a  $\mathcal{F}_s$  è normale, con media e varianza date rispettivamente da

$$\mathbb{E}^{\mathbb{Q}^T} \{r(t)|\mathcal{F}_s\} = x(s)e^{-a(t-s)} - M_x^T(s, t) + y(s)e^{-b(t-s)} - M_y^T(s, t) + \varphi(t)$$

$$\text{Var} \{r(t)|\mathcal{F}_s\} = \frac{\sigma^2}{2a} \left[ 1 - e^{-2a(t-s)} \right] + \frac{\eta^2}{2b} \left[ 1 - e^{-2b(t-s)} \right] \\ + 2\rho \frac{\sigma \eta}{a+b} \left[ 1 - e^{-(a+b)(t-s)} \right].$$

Per dimostrare il lemma è necessario il seguente

**Teorema 3.2.2** (Girsanov). *Si consideri la seguente equazione differenziale stocastica, avente coefficienti Lipschitziani,*

$$dX_t(\omega) = f(X_t(\omega))dt + \sigma(X_t(\omega))dW_t(\omega), \quad X_0 = x_0,$$

rispetto alla misura  $\mathbb{P}$ . Sia dato un nuovo coefficiente di drift  $f^*(x)$  e sia  $(f^*(x) - f(x))/\sigma(x)$  limitato. Sia  $\mathbb{P}^*$  una misura tale che

$$\frac{d\mathbb{P}^*}{d\mathbb{P}}(\omega) \Big|_{\mathcal{F}_t} = \exp \left\{ -\frac{1}{2} \int_0^t \left( \frac{f^*(X_s(\omega)) - f(X_s(\omega))}{\sigma(X_s(\omega))} \right)^2 ds \right. \\ \left. + \int_0^t \frac{f^*(X_s(\omega)) - f(X_s(\omega))}{\sigma(X_s(\omega))} dW_s(\omega) \right\}.$$

Allora  $\mathbb{P}^*$  è equivalente a  $\mathbb{P}$ . Inoltre, il processo  $W^*$ , definito da

$$dW_t^*(\omega) = - \left[ \frac{f^*(X_t(\omega)) - f(X_t(\omega))}{\sigma(X_t(\omega))} \right] dt + dW_t(\omega)$$

è un moto Browniano rispetto alla misura  $\mathbb{P}^*$ , e

$$dX_t(\omega) = f^*(X_t(\omega))dt + \sigma(X_t(\omega))dW_t^*(\omega), \quad X_0 = x_0.$$

*Dimostrazione.* La derivata di Radon-Nykodym, contenuta in (3.16), può essere riscritta in termini di due moti Browniani indipendenti  $\tilde{W}_1$  e  $\tilde{W}_2$ , come segue

$$\frac{d\mathbb{Q}^T}{d\mathbb{Q}} = \exp \left\{ -\frac{1}{2} V(0, T) - \int_0^T \left[ \frac{\sigma}{a} (1 - e^{-a(T-u)}) + \rho \frac{\eta}{b} (1 - e^{-b(T-u)}) \right] d\tilde{W}_1(u) \right. \\ \left. + \frac{\eta}{b} \sqrt{1 - \rho^2} \int_0^T (1 - e^{-b(T-u)}) d\tilde{W}_2(u) \right\}.$$

Poiché

$$\int_0^T \left[ \frac{\sigma}{a}(1 - e^{-a(T-u)}) + \rho \frac{\eta}{b}(1 - e^{-b(T-u)}) \right]^2 du \\ + \frac{\eta^2}{b^2}(1 - \rho^2) \int_0^T (1 - e^{-b(T-u)})^2 du = V(0, T),$$

per il teorema di Girsanov i due processi  $\tilde{W}_1^T$  e  $\tilde{W}_2^T$ , definiti da

$$\begin{aligned} d\tilde{W}_1^T(t) &= d\tilde{W}_1(t) + \left[ \frac{\sigma}{a}(1 - e^{-a(T-t)}) + \rho \frac{\eta}{b}(1 - e^{-b(T-t)}) \right] dt \\ d\tilde{W}_2^T(t) &= d\tilde{W}_2(t) + \frac{\eta}{b} \sqrt{1 - \rho^2} (1 - e^{-b(T-t)}) dt, \end{aligned} \quad (3.19)$$

sono due moti Browniani indipendenti rispetto alla misura  $\mathbb{Q}^T$ . Definendo  $W_1^T$  e  $W_2^T$  come

$$\begin{aligned} dW_1^T(t) &= d\tilde{W}_1^T(t) \\ dW_2^T(t) &= \rho d\tilde{W}_1^T(t) + \sqrt{1 - \rho^2} d\tilde{W}_2^T(t), \end{aligned}$$

e combinando (3.7) con (3.19) si ottiene (3.16). Le formule (3.18) seguono dall'integrazione di (3.16), utilizzando le seguenti equivalenze

$$\begin{aligned} \int_s^t \left[ \frac{\sigma^2}{a}(1 - e^{-a(T-u)}) + \rho \frac{\sigma\eta}{b}(1 - e^{-b(T-u)}) \right] e^{-a(t-u)} du &= M_x^T(s, t), \\ \int_s^t \left[ \frac{\eta^2}{b}(1 - e^{-b(T-u)}) + \rho \frac{\sigma\eta}{a}(1 - e^{-a(T-u)}) \right] e^{-b(t-u)} du &= M_y^T(s, t). \end{aligned}$$

□

**Teorema 3.2.3.** *Il prezzo al tempo t di un'opzione call Europea con maturity T e strike K, scritta su uno Zero-Coupon Bond con nominale unitario e maturity S è data da*

$$\begin{aligned} ZBC(t, T, S, K) &= P(t, S) \Phi \left( \frac{\ln \frac{P(t, S)}{KP(t, T)}}{\Sigma(t, T, S)} + \frac{1}{2} \Sigma(t, T, S) \right) \\ &\quad - P(t, T) K \Phi \left( \frac{\ln \frac{P(t, S)}{KP(t, T)}}{\Sigma(t, T, S)} - \frac{1}{2} \Sigma(t, T, S) \right), \end{aligned} \quad (3.20)$$

ove

$$\begin{aligned} \Sigma(t, T, S)^2 &= \frac{\sigma^2}{2a^3} \left[ 1 - e^{-a(S-T)} \right]^2 \left[ 1 - e^{-2a(T-t)} \right] \\ &\quad + \frac{\eta^2}{2b^3} \left[ 1 - e^{-b(S-T)} \right]^2 \left[ 1 - e^{-2b(T-t)} \right] \\ &\quad + 2\rho \frac{\sigma\eta}{ab(a+b)} \left[ 1 - e^{-a(S-T)} \right] \left[ 1 - e^{-b(S-T)} \right] \left[ 1 - e^{-(a+b)(T-t)} \right]. \end{aligned}$$

Analogamente, il prezzo al tempo  $t$  di un'opzione Put Europea, con maturity  $T$  e strike  $K$ , scritta su uno Zero-Coupon Bond con nozionale unitario e maturity  $S$  è data da

$$\begin{aligned} \mathbf{ZBP}(t, T, S, K) = & -P(t, S)\Phi\left(\frac{\ln\frac{KP(t, T)}{P(t, S)}}{\Sigma(t, T, S)} - \frac{1}{2}\Sigma(t, T, S)\right) \\ & + P(t, T)K\Phi\left(\frac{\ln\frac{KP(t, T)}{P(t, S)}}{\Sigma(t, T, S)} + \frac{1}{2}\Sigma(t, T, S)\right). \end{aligned} \quad (3.21)$$

### 3.2.3 Caplet e Floorlet

Dato l'istante iniziale  $t$  e le date future  $T_1$  e  $T_2$ , il payoff al tempo  $T_2$  di una Caplet in arrears è il seguente:

$$[L(T_1, T_2) - X]^+ \alpha(T_1, T_2)N,$$

ove  $N$  è il nominale,  $X$  è il tasso (strike) della Caplet,  $\alpha$  è la year fraction tra le date  $T_1$  e  $T_2$  e  $L(T_1, T_2)$  è il tasso LIBOR al tempo  $T_1$  per la maturity  $T_2$ , i.e.,

$$L(T_1, T_2) = \frac{1}{\alpha(T_1, T_2)} \left[ \frac{1}{P(T_1, T_2)} - 1 \right].$$

Il valore di non arbitraggio risulta dunque

$$\mathbf{Cpl}(t, T_1, T_2, N, X) = N' \mathbf{ZBP}(t, T_1, T_2, X') = \mathbf{ZBP}(t, T_1, T_2, N', N),$$

ove

$$\begin{aligned} X' &= \frac{1}{1 + X\alpha(T_1, T_2)} \\ N' &= N(1 + X\alpha(T_1, T_2)) \end{aligned}$$

Analogamente, il valore di non arbitraggio al tempo  $t$  della floorlet che paga

$$[X - L(T_1, T_2)]^+ \alpha(T_1, T_2)N$$

al tempo  $T_2$  è

$$\mathbf{Fl}(t, T_1, T_2, N, X) = N' \mathbf{ZBC}(t, T_1, T_2, X') = \mathbf{ZBC}(t, T_1, T_2, N', N).$$

### 3.2.4 Cap e Floor

Si denotano con  $\mathcal{T} = \{T_0, T_1, T_2, \dots, T_n\}$  l'insieme delle date di pagamento per un Cap/Floor. ( $T_0$  è la prima data di reset) e  $\tau = \{\tau_1, \dots, \tau_n\}$  l'insieme delle corrispondenti year fractions, intendendo con  $\tau_{i-1}$  la year fraction tra  $T_{i-1}$  e  $T_i$ .

Poiché un Cap (Floor) può essere interpretato come la somma dei sottostanti Caplet (Floorlet), il prezzo al tempo  $t$  di un Cap, con Cap rate (strike)  $X$ , nominale  $N$ , insieme di date di reset  $\mathcal{T}$  e year fractions  $\tau$  è dato da

$$\begin{aligned} \mathbf{Cap}(t, \mathcal{T}, \tau, N, X) &= \sum_{i=1}^n \left[ -N(1 + X\tau_i)P(t, T_i)\Phi \left( \frac{\ln \frac{P(t, T_{i-1})}{(1+X\tau_i)P(t, T_i)}}{\Sigma(t, T_{i-1}, T_i)} - \frac{1}{2}\Sigma(t, T_{i-1}, T_i) \right) \right. \\ &\quad \left. + P(t, T_{i-1})N\Phi \left( \frac{\ln \frac{P(t, T_{i-1})}{(1+X\tau_i)P(t, T_i)}}{\Sigma(t, T_{i-1}, T_i)} + \frac{1}{2}\Sigma(t, T_{i-1}, T_i) \right) \right], \end{aligned} \quad (3.22)$$

e il prezzo della corrispondente Floor risulta

$$\begin{aligned} \mathbf{Flr}(t, \mathcal{T}, \tau, N, X) &= \sum_{i=1}^n \left[ N(1 + X\tau_i)P(t, T_i)\Phi \left( \frac{\ln \frac{(1+X\tau_i)P(t, T_i)}{P(t, T_{i-1})}}{\Sigma(t, T_{i-1}, T_i)} + \frac{1}{2}\Sigma(t, T_{i-1}, T_i) \right) \right. \\ &\quad \left. - P(t, T_{i-1})N\Phi \left( \frac{\ln \frac{(1+X\tau_i)P(t, T_i)}{P(t, T_{i-1})}}{\Sigma(t, T_{i-1}, T_i)} - \frac{1}{2}\Sigma(t, T_{i-1}, T_i) \right) \right]. \end{aligned} \quad (3.23)$$

## 3.3 Swaption Europea

Nella presente sezione si considerano due distinte formule di pricing per la Swaption Europea, proposte in [6] ed in [8]. Nell'ultimo paragrafo della sezione si commentano le due formulazioni.

### 3.3.1 Formula di Brigo e Mercurio

Si consideri una swaption Europea con tasso strike  $X$ , maturity  $T$  e nominale  $N$ , che dà il diritto al possessore di entrare al tempo  $t_0 = T$  in uno swap di tasso di interesse con date di pagamento  $\mathcal{T} = \{t_1, \dots, t_n\}$ ,  $t_1 > T$ , in cui il possessore paga (riceve) il tasso fisso  $X$  e riceve (paga) il tasso LIBOR, fissato in arrears. Si denota con  $\tau_i$  la year fraction tra  $t_{i-1}$  e  $t_i$ , con  $i = 1, \dots, n$  e si considera  $c_i = X\tau_i$ , ove  $i = 1, \dots, n-1$  e  $c_n = 1 + X\tau_n$ . Si ha dunque il seguente

**Teorema 3.3.1.** *Il prezzo di non arbitraggio al tempo  $t = 0$  della swaption Europea appena descritta è dato dal calcolo del seguente integrale numerico:*

$$ES(0, T, \mathcal{T}, N, X, \omega) = N\omega P(0, T) \int_{-\infty}^{\infty} \frac{e^{-\frac{1}{2}\left(\frac{x-\mu_x}{\sigma_x}\right)^2}}{\sigma_x \sqrt{2\pi}} \left[ \Phi(-\omega h_1(x)) - \sum_{i=1}^n \lambda_i(x) e^{k_i(x)} \Phi(-\omega h_2(x)) \right] dx, \quad (3.24)$$

ove  $\omega = 1$  ( $\omega = -1$ ) per una payer (receiver) swaption,  $a$  e  $b$  sono i parametri di mean reversion del modello  $G2++$ ,

$$\begin{aligned} h_1(x) &:= \frac{\bar{y} - \mu_y}{\sigma_y \sqrt{1 - \rho_{xy}^2}} - \frac{\rho_{xy}(x - \mu_x)}{\sigma_x \sqrt{1 - \rho_{xy}^2}} \\ h_2(x) &:= h_1(x) + B(b, T, t_i) \sigma_y \sqrt{1 - \rho_{xy}^2} \\ \lambda_i(x) &= c_i A(T, t_i) e^{-B(a, T, t_i)x}, \\ k_i(x) &:= -B(b, T, t_i) \left[ \mu_y - \frac{1}{2}(1 - \rho_{xy}^2) \sigma_y^2 B(b, T, t_i) + \rho_{xy} \sigma_y \frac{x - \mu_x}{\sigma_x} \right], \\ B(z, t, T) &= \frac{1 - e^{-z(T-t)}}{z} \end{aligned}$$

$\bar{y} = \bar{y}(x)$  è l'unica soluzione dell'equazione

$$\sum_{i=1}^n c_i A(T, t_i) e^{-B(a, T, t_i)x - B(b, T, t_i)\bar{y}} = 1,$$

ove

$$A(t, T) = \frac{P^M(0, T)}{P^M(0, t)} \exp \left\{ \frac{1}{2} [V(t, T) - V(0, T) + V(0, t)] \right\}$$

e

$$\begin{aligned} \mu_x &:= -M_x^T(0, T), \\ \mu_y &:= -M_y^T(0, T), \\ \sigma_x &:= \sigma \sqrt{\frac{1 - e^{-2aT}}{2a}}, \\ \sigma_y &:= \eta \sqrt{\frac{1 - e^{-2bT}}{2b}}, \\ \rho_{xy} &:= \frac{\rho \sigma \eta}{(a + b) \sigma_x \sigma_y} \left[ 1 - e^{-(a+b)T} \right]. \end{aligned}$$

*Dimostrazione.* Si veda l'appendice B. □

Per il calcolo di  $\bar{y}$  si è fatto ricorso al metodo di bisezione.

### 3.3.2 Formula di Choi e Shin

Nel presente paragrafo si discute un metodo numerico alternativo, riportato in [8], per il pricing di una Swaption Europea, computazionalmente meno oneroso rispetto al metodo proposto in [6].

In seguito all'introduzione delle notazioni e del modello utilizzati da [8], si dimostra che tale modello differisce da G2++ per uno shift deterministico, di cui si calcola il valore.

I codici MATLAB delle funzioni utilizzate per il calcolo del prezzo della Swaption Europea sono riportati in appendice D.

**Definizione 3.3.1.** Mediante le seguenti scritte formali sono definiti gli operatori:

$$\mathbf{a} \circ \mathbf{b} = \mathbf{b} \circ \mathbf{a} = [a_j b_j]_j,$$

$$M \circ \mathbf{a} = \mathbf{a} \circ M = [M_{jk} a_j]_{j,k},$$

$$M \circ \mathbf{a}^T = \mathbf{a}^T \circ M = [M_{jk} a_k]_{j,k},$$

ove  $\mathbf{a} = [a_j]_j$  e  $\mathbf{b} = [b_j]_j$  sono vettori  $d$ -dimensionali e  $M$  è una matrice quadrata di ordine  $d$ .

Gli autori di [8], partendo dal framework Heath-Jarrow-Morton (HJM) esprimono la dinamica relativa allo Zero-Coupon Bond  $P(t, T)$  mediante la seguente equazione differenziale stocastica:

$$\frac{dP(t, T)}{P(t, T)} = r(t)dt - \boldsymbol{\sigma}_P(t, T)^T d\mathbf{W}^\beta(t), \quad (3.25)$$

ove  $r(t)$  è lo short rate,  $\boldsymbol{\sigma}_P(t, T)$  è il vettore di volatilità, e  $\mathbf{W}^\beta(t)$  è un moto Browniano  $d$ -dimensionale rispetto la misura neutrale al rischio  $\mathbb{Q}^\beta$ , le cui componenti sono correlate mediante la matrice  $R(t) = [\rho_{jk}(t)]_{j,k}$ .

Sia  $f(t, T)$  il tasso forward istantaneo, osservato in  $t$  e avente maturity in  $T$ . Si ha dunque

$$P(t, T) = \exp\left(-\int_t^T f(t, s) ds\right).$$

Sostituendo quest'ultima equazione in (3.25) si ottiene

$$df(t, T) = \boldsymbol{\sigma}_f(t, T)^T \boldsymbol{\sigma}_P(t, T)dt + \boldsymbol{\sigma}_f(t, T)^T d\mathbf{W}^\beta(t), \quad (3.26)$$

in cui la volatilità del tasso forward istantaneo  $\boldsymbol{\sigma}_f(t, T)$  risulta

$$\boldsymbol{\sigma}_f(t, T) = \frac{\partial}{\partial T} \boldsymbol{\sigma}_P(t, T).$$



Inoltre, il processo stocastico descrivente il tasso forward istantaneo è dato da

$$r(t) = f(t, t) = f(0, t) + \mathbf{1}^T \mathbf{x}(t), \quad (3.27)$$

per un vettore di stato  $d$ -dimensionale  $\mathbf{x}(t)$ , tale che  $\mathbf{x}(0) = 0$ ,

$$\mathbf{x}(t) = \int_0^t \boldsymbol{\sigma}_f(s, t) \circ \int_s^t \boldsymbol{\sigma}_f(s, u) du ds + \int_0^t \boldsymbol{\sigma}_f(s, t) \circ d\mathbf{W}^\beta(s). \quad (3.28)$$

Per ricondursi alla notazione adottata nel presente lavoro di tesi, si preferisce considerare quella particolare forma separabile di  $\boldsymbol{\sigma}_f$  tale che la dinamica dello short rate  $r(t)$  sia un processo mean-reverting di Ornstein-Uhlenbeck

$$d\mathbf{x}(t) = [\Pi(t)\mathbf{1} - \boldsymbol{\lambda} \circ \mathbf{x}(t)] dt + \boldsymbol{\sigma}_r(t)^T d\mathbf{W}^\beta(t) \quad (3.29)$$

ove  $\boldsymbol{\lambda}(t)$  è il coefficiente deterministico di mean reversion,  $\boldsymbol{\sigma}_r(t)$  è il vettore deterministico di volatilità dello short rate e  $\Pi(t)$  è una matrice di drift.

Questo è equivalente a fissare la separazione di  $\boldsymbol{\sigma}_f(t, T)$  come

$$\boldsymbol{\sigma}_f(t, T) = \boldsymbol{\sigma}_r(t, T) \circ \mathbf{m}(t, T), \quad (3.30)$$

ove  $\mathbf{m}(t, T)$  rappresenta il decadimento esponenziale tra  $t$  e  $T$

$$\mathbf{m}(t, T) = \left[ e^{-\int_t^T \lambda_k(s) ds} \right]_k \quad \text{per} \quad \boldsymbol{\lambda}(t) = [\lambda_k(t)]_k \quad (3.31)$$

Lo stato  $\mathbf{x}(t)$  è una distribuzione Gaussiana multivariata. La matrice di drift  $\Pi(t)$  risulta essere la matrice di covarianza di  $\mathbf{x}(t)$ . Si ottiene dunque

$$\begin{aligned} \Pi(t) &= \int_0^t (\boldsymbol{\sigma}_f(s, t) \circ d\mathbf{W}^\beta(s)) (\boldsymbol{\sigma}_f(s, t) \circ d\mathbf{W}^\beta(s))^T \\ &= \int_0^t \boldsymbol{\sigma}_r(s) \circ \mathbf{m}(s, t) \circ R(s) \circ \boldsymbol{\sigma}_r(s)^T \circ \mathbf{m}(s, t)^T ds \\ &= \left[ \int_0^t \rho_{jk}(t) \sigma_{rj}(s) \sigma_{rk}(s) m_j(s, t) m_k(s, t) ds \right]_{j,k} \end{aligned} \quad (3.32)$$

**Lemma 3.3.1.** *Il modello proposto nella forma di [8] differisce dal modello in [6] per uno shift deterministico  $\nu(t)$ .*

*Dimostrazione.* Si denotino con  $r(t)$  e  $\underline{x}(t)$  rispettivamente, il tasso short ed il vettore di stato nella notazione di [8] e siano  $\tilde{r}(t)$ ,  $\tilde{\underline{x}}$ , rispettivamente, il tasso short ed il vettore di stato nella notazione di [6]:

$$r(t) = \tilde{r}(t) + \nu(t), \quad (3.33)$$

$$dr(t) = d\tilde{r}(t) + d\nu(t)$$

$$\begin{aligned}
d\tilde{x}(t) + d\nu(t) &= [\Pi(t)\mathbf{1} - \lambda(t) \circ \tilde{x}(t) - \lambda(t) \circ \nu(t)] dt + d\tilde{x}(t) \\
&= \Pi(t)\mathbf{1}dt - \lambda(t) \circ \nu(t)dt - d\nu(t) - \lambda(t) \circ \tilde{x}(t) + \sigma_r(t)^T dW^\beta(t)
\end{aligned}$$

$$\begin{aligned}
\frac{d\nu(t)}{dt} &= -\lambda(t) \circ \nu(t) + \pi(t) \circ \mathbf{1} \\
\nu'(t) &= -\lambda(t) \circ \nu(t) + \pi(t) \circ \mathbf{1}
\end{aligned}$$

$$\nu(t) = e^{-\int_0^t \lambda(s) ds} \circ \int_0^t \Pi(s) \cdot \mathbf{1} \circ e^{\int_0^s \lambda(u) du} ds$$

□

Nel modello considerato, si ottiene dunque che

$$\sigma_f(s, t) = \begin{bmatrix} \sigma e^{-a(t-s)} \\ \eta e^{-b(t-s)} \end{bmatrix}$$

e

$$\begin{aligned}
\Pi(t) &= \begin{bmatrix} \frac{\sigma^2}{2a}(1 - e^{-2at}) & \frac{\sigma\eta\rho}{a+b}(1 - e^{-(a+b)t}) \\ \frac{\sigma\eta\rho}{a+b}(1 - e^{-(a+b)t}) & \frac{\eta^2}{2b}(1 - e^{-2bt}) \end{bmatrix} \\
\nu(t) &= \begin{bmatrix} \frac{\sigma^2}{2a^2}(1 - e^{-at})^2 + \frac{\sigma\eta\rho}{a+b} \left( \frac{1 - e^{-at}}{a} + \frac{e^{-(a+b)t} - e^{-at}}{b} \right) \\ \frac{\eta^2}{2b^2}(1 - e^{-bt})^2 + \frac{\sigma\eta\rho}{a+b} \left( \frac{1 - e^{-bt}}{b} + \frac{e^{-(a+b)t} - e^{-bt}}{a} \right) \end{bmatrix} \quad (3.34)
\end{aligned}$$

Risulta dunque possibile scrivere il prezzo dello Zero-Coupon Bond alla data futura  $t$  e avente maturity  $T$  utilizzando lo stato markoviano  $\mathbf{x}(t)$

$$P(t, T) = \frac{P^M(0, T)}{P^M(0, t)} \exp \left( -\mathbf{g}(t, T)^T \mathbf{x}(t) - \frac{1}{2} \mathbf{g}(t, T)^T \Pi(t) \mathbf{g}(t, T) \right) \quad (3.35)$$

ove  $\mathbf{g}(t, T) = \int_t^T \mathbf{m}(t, s) ds$ .

Si consideri lo swap sottostante ad una swaption, avente inizio alla data futura  $T_0$ , che paga il coupon fisso  $K$  (payer swap) in date prefissate  $(T_1, \dots, T_m)$ , e riceve il tasso variabile, tipicamente LIBOR. Il valore dello swap, al tempo  $t$ , è dato da

$$V(t) = P(t, T_0) - P(t, T_m) - \sum_{k=1}^m KP(t, T_k)\Delta_k, \quad (3.36)$$

ove  $\Delta_k$  rappresenta la year fraction tra  $T_{k-1}$  e  $T_k$ . In generale,

$$V(t) = \sum_{k=0}^m P(t, T_k)CF(T_k) \quad (3.37)$$

ove  $CF(T_k)$  rappresenta il flusso di cassa al tempo  $T_k$ .

Sia  $T_e$  la data di expiry della swaption per entrare nel sottostante swap, che paga il coupon fisso  $K$ . (Si assume, tipicamente, che  $T_e$  si collochi due giorni lavorativi prima dell'inizio dello swap  $T_0$ ).

Prima di calcolare il prezzo dello swap si preferisce decorrelare e normalizzare la variabile di stato  $\mathbf{x}(t)$  in  $\mathbf{z}(t)$  attraverso  $\mathbf{x}(t) = C(t)\mathbf{z}(t)$ , ove  $C(t)$  è la fattorizzazione di Cholesky della matrice di covarianza, i.e.  $\Pi(t) = C(t)C(t)^T$ . La formula dello Zero-Coupon Bond può dunque essere scritta nel seguente modo:

$$P(t, T) = \frac{P^M(0, T)}{P^M(0, t)} \exp \left( -\mathbf{a}(t, T)^T \mathbf{z}(t) - \frac{1}{2} |\mathbf{a}(t, T)|^2 \right) \quad (3.38)$$

ove  $\mathbf{a}(t, T) = C(t)^T \mathbf{g}(t, T)$ .

Il valore dello swap avente expiry  $T_e$  può dunque essere espresso come funzione dello stato  $\mathbf{z} = \mathbf{z}(T_e)$ .

$$V(\mathbf{z}) = \frac{1}{P(0, T_e)} \sum_{k=0}^m \text{DCF}_k \exp \left( -\mathbf{a}_k^T \mathbf{z} - \frac{1}{2} |\mathbf{a}_k|^2 \right) \quad (3.39)$$

ove  $\text{DCF}_k$  rappresenta il flusso di cassa scontato  $CF(T_k)P(0, T_k)$ .

Il prezzo della swaption payer risulta dunque il valore atteso di  $H(t)$ , rispetto alla  $T_e$ -forward measure

$$C = P(0, T_e) \mathbb{E}^{T_e} \{ \max(V(t), 0) \}. \quad (3.40)$$

Analogamente, il prezzo della swaption receiver è dato da

$$P = P(0, T_e) \mathbb{E}^{T_e} \{ \max(-V(t), 0) \}. \quad (3.41)$$

Il calcolo di (3.40) richiede un'integrazione  $d$ -dimensionale, ove  $d$  è il numero di fattori di rischio considerati. La maggiore difficoltà risiede nell'identificazione del dominio di integrazione  $\Omega$ , in cui lo swap sottostante ha valore positivo, ed il bordo  $\partial\Omega$ .

$$\Omega = \{t : V(t) \geq 0\}, \partial\Omega = \{t : V(t) = 0\}. \quad (3.42)$$

Si semplifica l'integrazione approssimando il bordo  $\partial\Omega$  con un iperpiano. Nella regione di esercizio, si identifica lo stato  $\mathbf{z}^*$ , avente massima densità di probabilità. Per approssimazione, si sceglie il piano tangente al bordo in  $\mathbf{z}^*$ . Poiché  $\mathbf{z}$  è distribuito come una normale multivariata non correlata, la densità di probabilità è decrescente ed è funzione di  $|\mathbf{z}|$ . Si ha, inoltre, che  $\mathbf{z}^*$  è il punto a distanza minima dall'origine  $\mathbf{0}$ , tra tutti i punti del bordo  $\partial\Omega$ , e che  $\mathbf{z}^*$  è multiplo scalare di  $\nabla V(\mathbf{z}^*)$ . Quest'ultima proprietà sarà utilizzata per trovare  $\mathbf{z}^*$ .

Il punto  $\mathbf{z}^*$  può essere determinato numericamente, mediante il seguente metodo iterativo, consistente in due passi:

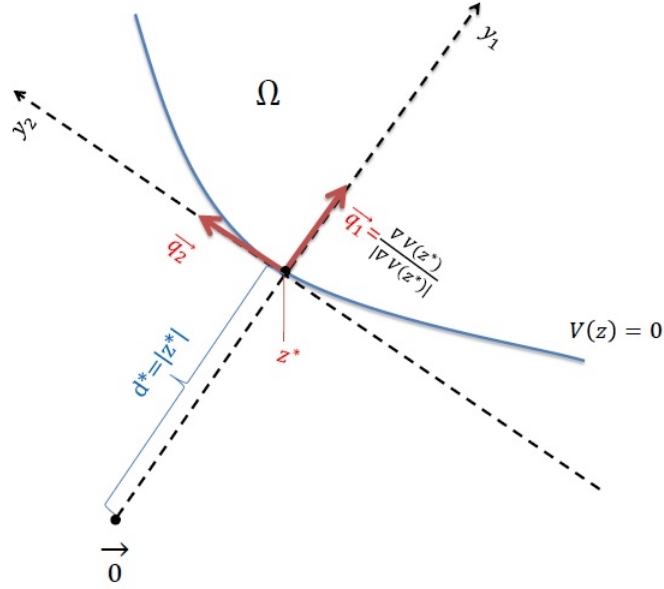


Figura 3.2: Iperpiano approssimante.

1. Partendo da  $\mathbf{z}^{(i)}$ , applicazione del metodo di massima discesa

$$\mathbf{z}^{(i+\frac{1}{2})} = \mathbf{z}^{(i)} - \frac{V(\mathbf{z}^{(i)})}{|\nabla V(\mathbf{z}^{(i)})|^2} \nabla V(\mathbf{z}^{(i)}) \quad (3.43)$$

2. Proiettare  $\mathbf{z}^{(i+\frac{1}{2})}$  nella direzione del gradiente  $\nabla \mathbf{z}^{(i+\frac{1}{2})}$

$$\mathbf{z}^{(i+1)} = \frac{\nabla V(\mathbf{z}^{(i+\frac{1}{2})})^T \mathbf{z}^{(i+\frac{1}{2})}}{|\nabla V(\mathbf{z}^{(i+\frac{1}{2})})|^2} \nabla V(\mathbf{z}^{(i+\frac{1}{2})}) \quad (3.44)$$

Choi e Shin ritengono che sia difficile dimostrare rigorosamente che lo schema iterativo converga ad una soluzione per ogni set di parametri fissato. Nel modello G2++ la variabile di stato è, al primo ordine, proporzionale al tasso di interesse. Pertanto, il valore dello swap risulta funzione lineare dello stato. Questo fatto assicura che il bordo  $\partial\Omega$  esiste ed è vicino ad un iperpiano.

La ricerca della soluzione è l'operazione computazionalmente più onerosa nel calcolo del prezzo della swaption, dato che il resto del calcolo si risolve analiticamente.

Per semplificare il calcolo di (3.40), si consideri  $\mathbf{q}_1 = \nabla V(\mathbf{z}^*)/|\nabla \mathbf{z}^*|$ , vettore unitario avente direzione  $\nabla \mathbf{z}^*$ . Si esprime dunque il vettore di stato

$$\mathbf{z} = u_1 \mathbf{q}_1 + u_2 \mathbf{q}_2 \quad (3.45)$$

in coordinate cartesiane  $\mathbf{u} = (u_1, u_2)$  rispetto alla base  $\{\mathbf{q}_1, \mathbf{q}_2\}$ .  
 I vettori di modulo unitario  $\mathbf{q}_2, \dots, \mathbf{q}_d$  possono essere scelti arbitrariamente, ma in modo che  $Q = [q_1, \dots, q_d]$  costituisca una matrice ortogonale.  
 Per rispettare la proprietà appena introdotta, si considerano uno scalare  $d^*$  ed i vettori  $\mathbf{b}_k$  tali che

$$\mathbf{x}^* = d^* \mathbf{q}_1, \quad \mathbf{a}_k = Q \mathbf{b}_k \quad (3.46)$$

Allora, il dominio approssimato  $\tilde{\Omega}$  è dato da

$$\tilde{\Omega} = \{\mathbf{z} : \mathbf{q}_1^T (\mathbf{z} - \mathbf{z}^*) \geq 0\} = \{\mathbf{y} : y_1 \geq d^*\} \quad (3.47)$$

e il valore dello swap diviene

$$H(\mathbf{z}) = \frac{1}{P(0, T_e)} \sum_{k=0}^m DCF_k \exp \left\{ -\frac{1}{2} |\mathbf{b}_k|^2 - \mathbf{b}_k^T \mathbf{y} \right\} \quad (3.48)$$

Effettuando l'integrazione rispetto alla coordinata  $\mathbf{y}$ ,  $y_1$  è l'unico asse rispetto al quale l'integrazione è non banale. Tutti gli altri integrali sommano ad 1. Si ottiene, pertanto, il prezzo della swaption in forma analitica:

$$\begin{aligned} C &= P(0, T_e) \int_{\Omega} V(\mathbf{z}) f(\mathbf{z}) d\mathbf{z} \approx P(0, T_e) \int_{\tilde{\Omega}} H(\mathbf{y}) f(\mathbf{y}) d\mathbf{y} \\ &= \sum_{k=0}^m DCF_k \int_{-\infty}^{+\infty} dy_2 \cdots dy_d \int_{d^*}^{+\infty} dy_1 e^{-\frac{1}{2} |\mathbf{b}_k|^2 - \sum_{j=1}^d b_{kj} y_j} n(y_1) \cdots n(y_d) \\ &= \sum_{k=0}^m DCF_k \int_{d^*}^{+\infty} dy_1 e^{\frac{1}{2} b_{k1}^2 - b_{k1} y_1} n(y_1) \\ &= \sum_{k=0}^m DCF_k \mathcal{N}(-b_{k1} - d^*) = \sum_{k=0}^m DCF_k \mathcal{N}(-\mathbf{a}_k + \mathbf{z}^*)^T \mathbf{q}_1 \end{aligned} \quad (3.49)$$

ove  $f(\cdot)$  è la densità di probabilità della distribuzione Gaussiana multivariata,  $n(\cdot)$  e  $\mathcal{N}(\cdot)$  sono, rispettivamente, la probabilità e la densità cumulata della distribuzione normale univariata.

### 3.3.3 Una nota sulle due formule di pricing

Nella derivazione della formula di pricing per la Swaption Europea in [8], si è notato un possibile errore nell'applicazione del metodo del gradiente. Infatti, nelle formule (3.43) e (3.44) si nota che, a denominatore, compare la norma al quadrato del gradiente. Tuttavia, tali termini non costituiscono una normalizzazione del gradiente e provocano la divergenza del metodo. Per quanto concerne l'implementazione della formula di pricing proposta in [6], le principali criticità riscontrate riguardano il calcolo dell'integrale e la

ricerca della soluzione dell'equazione (3.3.1).

L'integrale improprio è stato, ovviamente, troncato e calcolato in un intervallo finito. All'aumentare dell'estensione di tale intervallo, non si sono notate rivalutazioni apprezzabili nel prezzo della Swaption.

Relativamente alla soluzione dell'equazione (3.3.1), si è utilizzato il metodo di bisezione, il quale garantisce la convergenza verso la soluzione.

Nella presente sezione si è dimostrato formalmente come sia possibile ricavare il modello di [6] a partire da [8], e viceversa. Si nota, tuttavia, che nel modello di Choi e Shin, il significato finanziario dei parametri introdotti nel calcolo della matrice di covarianza  $\Pi(t)$  potrebbe essere diverso dai parametri adottati in [6]. In particolare, sostituendo la matrice di covarianza di [6] nel modello di [8] non si replica esattamente il prezzo dello Zero-Coupon Bond. Pur essendo stato possibile verificare la relazione di identità tra il termine  $M(t, T)$ , nella notazione di [6], ed il prodotto  $g(t, T)^T x(t)$ , tale relazione non sussiste per i termini  $g(t, T)^T \Pi(t) g(t, T)$  e  $V(0, T) - V(0, t) - V(t, T)$ .

Pertanto, non è stato possibile effettuare un confronto sull'accuratezza nel pricing della Swaption Europea tra i due modelli.

## 3.4 Calibrazione su Dati di Mercato

### 3.4.1 Formulazione

Al fine di modellizzare correttamente una struttura a termine, la possibilità di rappresentare lo stato corrente del mercato è una caratteristica fondamentale del modello in considerazione. Questa funzionalità può essere ottenuta mediante la creazione di modelli flessibili, che possono generare qualsiasi forma della struttura a termine. Per incrementare il dettaglio nella descrizione della dinamica si possono introdurre più fattori di rischio. Ogni fattore porta con sé una forma predefinita, e.g. uno shift o un twist, aumentando la complessità del modello e diminuendone la trattabilità numerica.

La qualità con cui si approssima un mercato può essere misurata confrontando quantità caratteristiche, per esempio, prezzi o volatilità, riguardanti un ampio spettro di strumenti finanziari quotati sul mercato con le analoghe quantità implicite nel modello.

Cambiando radicalmente punto di vista, si può cercare un modello di struttura a termine che minimizzi tali differenze per uno specifico strumento di mercato, i.e. si calibra il modello rispetto ai prezzi o alle volatilità di determinati prodotti finanziari.

La formulazione e la complessità del problema di calibrazione dipendono dai gradi di libertà presenti nella struttura, ovverosia dal numero di parametri liberi contenuti nel modello.

**Problema 3.4.1.1.** Sia  $\Phi \subset \mathbb{R}^d$  l'insieme dei parametri liberi del modello in esame, e sia inoltre  $PV : \Phi \subset \mathbb{R}^d \rightarrow \mathbb{R}$  una formula di pricing (e.g. per le Swaption Europea), funzione dei parametri liberi. Inoltre, sia  $\tilde{\sigma}_i$  la volatilità implicita di Black-Scholes osservata sul mercato e sia  $\sigma_i(PV_i(\phi))$  la volatilità implicita nel prezzo  $PV_i(\phi)$  del modello associato all'insieme dei parametri  $\phi \in \Phi$ . Nel processo di calibrazione si cerca il minimo globale  $M$  della funzione  $E : \mathbb{R}^d \rightarrow \mathbb{R}$ ,

$$M = \inf_{\phi \in \Phi} E(\phi), \quad E(\phi) = \sum_{i=0}^I \omega_i |\sigma(PV_i(\phi)) - \tilde{\sigma}_i|^2, \quad (3.50)$$

ove l'intero  $I$  denota il numero di tutti gli strumenti finanziari considerati per effettuare la calibrazione e  $\omega_i \in \mathbb{R}$  denota il peso attribuito ad ogni strumento. A strumenti finanziari distinti si possono far corrispondere pesi distinti e, a loro volta, contestualmente al medesimo strumento finanziario, si possono pesare in modo distinto le maturities.

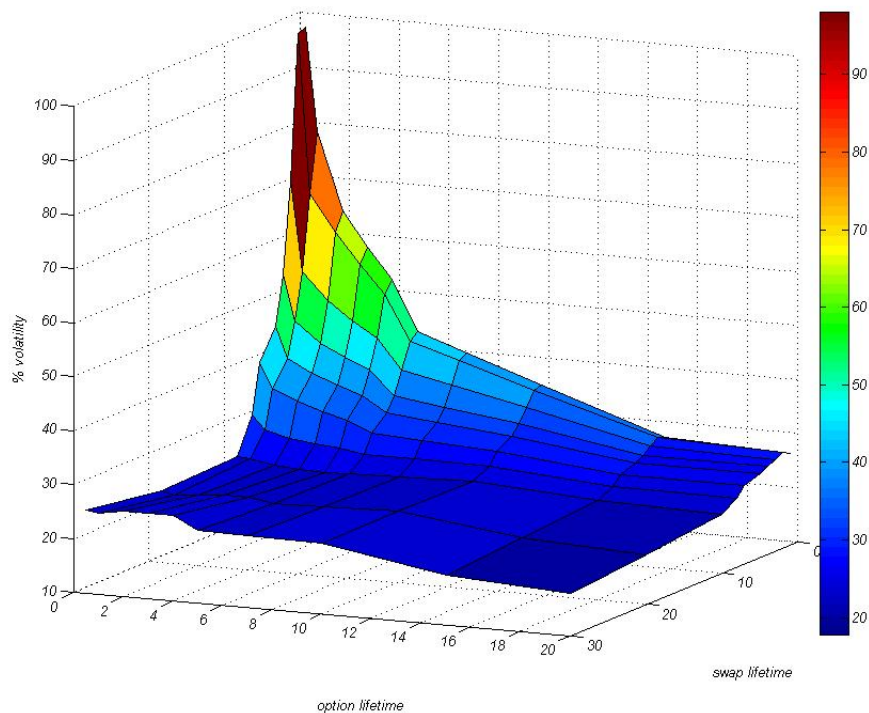


Figura 3.3: Superficie di volatilità della Swaption Europea in EUR, riportata su MUREX in data 04.06.2013

### 3.4.2 Formula di Black-Scholes per la valutazione della Swaption Europea

Senza effettuarne la derivazione, si riporta la formula di pricing per la Swaption Europea (payer), secondo Black e Scholes:

$$ES = m^{-1} \left[ BF\Phi(d) - BK\Phi(d - \sqrt{nhv}) \right] \quad (3.51)$$

ove

$$d = \frac{\ln(F/K) + nh\frac{v^2}{2}}{\sqrt{nhv}},$$

$F$  = forward swap rate iniziale,

$K$  = tasso strike,

$m$  = numero di pagamenti annui,

$h = 1/m =$  tenor,

$n$  = maturity della swaption, in numero di periodi di lunghezza  $h$ ,

$nh$  = maturity della swaption in anni,

$$B = b_t^{n+1} + b_t^{n+2} + \dots + b_t^{t+\tau},$$

$b_t^n$  = fattore di sconto tra  $n$  e  $t$ ,

$\Phi$  = CDF della distribuzione Normale standard,

$v$  = volatilità annualizzata.

### 3.4.3 Vincoli

I parametri contenuti nella formulazione del modello G2++ devono rispettare i seguenti vincoli:

$$\sigma \geq 0, \eta \geq 0$$

$$a > 0, b > 0$$

$$-1 \leq \rho \leq 1$$

### 3.4.4 Caratterizzazione della Regione di Ottimizzazione

La regione di ottimizzazione è determinata dalle realizzazioni della funzione errore (3.50). L'esito della ricerca del minimo globale dipende fortemente dalle caratteristiche della funzione  $E$ , definita precedentemente, riconducibili al modello utilizzato ed alle quote di mercato considerate. Molto spesso  $E$  è funzione non lineare ed il corrispondente problema di minimizzazione si presenta come un problema multidimensionale, in quanto il modello è governato da un set di parametri liberi. In aggiunta alla non linearità, la funzione



difficilmente esibisce convessità (o concavità), complicando il problema di ottimizzazione e rendendo inutilizzabili alcuni metodi di risoluzione. Solo in alcuni casi molto semplici, come nel caso di volatilità costante, la funzione  $E$  risulta monotona ma, in generale, questa situazione non si verifica.

### 3.4.5 Downhill Simplex

Downhill Simplex, proposto da J.Nelder e W.Nead nel 1965, è un metodo di ottimizzazione che richiede la sola valutazione puntuale di funzioni, senza richiederne le derivate. Non è molto efficiente in termini di numero di valutazioni della funzione, ma si tratta di un metodo robusto, i cui risultati sono affidabili.

**Definizione 3.4.1.** Un *simplexso*  $N$ -dimensionale è una figura geometrica consistente di  $N + 1$  punti (vertici) tutti connessi tra loro mediante segmenti, facce poligonali, etc.

Nel prosieguo, si considerano solo simplexsi *non degeneri*, i.e., contenenti un volume  $N$ -dimensionale finito.

Assumendo come origine un qualsiasi punto del simplexso, tutti gli altri  $N$  punti definiscono le *direzioni* che generano l'intero spazio vettoriale  $N$ -dimensionale.

Il metodo Downhill Simplex deve essere inizializzato con  $N + 1$  punti, che definiscono il simplexso di partenza. È possibile considerare un punto  $\mathbf{P}_0$  (non importa quale) come punto iniziale, ed interpretare gli altri  $N$  come

$$\mathbf{P}_i = \mathbf{P}_0 + \Delta \mathbf{e}_i \quad (3.52)$$

ove  $\mathbf{e}_i$  rappresentano gli  $N$  vettori unitari, e  $\delta$  la presunta scala caratteristica del problema. (È possibile considerare diversi  $\Delta_i$  per ogni direzione del vettore). Il metodo Downhill Simplex impiega una serie di passi, la maggior parte dei quali consistono nella sostituzione del punto in cui la funzione assume il valore più lontano dall'ottimo con un punto opposto rispetto al centroide dei rimanenti  $N$  punti. Tali iterazioni, dette *riflessioni*, sono definite in modo tale da mantenere costante il volume del simplexso. Se il punto trovato esibisce un valore più vicino all'ottimo, rispetto al punto di partenza, allora il simplexso si deforma lungo questa direzione. Altrimenti, si dice che il metodo si sta addentrando in un *avvallamento*. Pertanto, seguono la contrazione del simplexso rispetto a quella direzione e la ricerca di una nuova direzione verso l'ottimo.

Nel caso in cui il metodo non riesca ad identificare una direzione preferenziale per la ricerca dell'ottimo, il simplexso si contrae rispetto alle direzioni ammissibili, restituendo in output il punto di ottimo globale.

Le condizioni d'arresto più utilizzate sono le seguenti:

1. la norma della distanza tra due step consecutivi sia inferiore ad  $\epsilon$ , arbitrariamente piccolo
2. il valore assoluto della differenza delle valutazioni della funzione in uno step e nel successivo sia inferiore ad  $\epsilon$ ,
3. la norma del gradiente sia inferiore ad  $\epsilon$ .

Per evitare che il criterio adottato possa essere eluso, per motivi numerici, è buona norma effettuare un *reset* della routine di minimizzazione multidimensionale nel candidato punto di minimo. Per tale operazione, è necessario provvedere alla reinizializzazione di tutte le quantità ausiliare introdotte nel metodo, ovverosia tutti gli altri  $N$  vertici del semplice, secondo l'equazione (3.52), essendo  $\mathbf{P}_0$  uno dei vertici dell'ultimo minimo identificato.

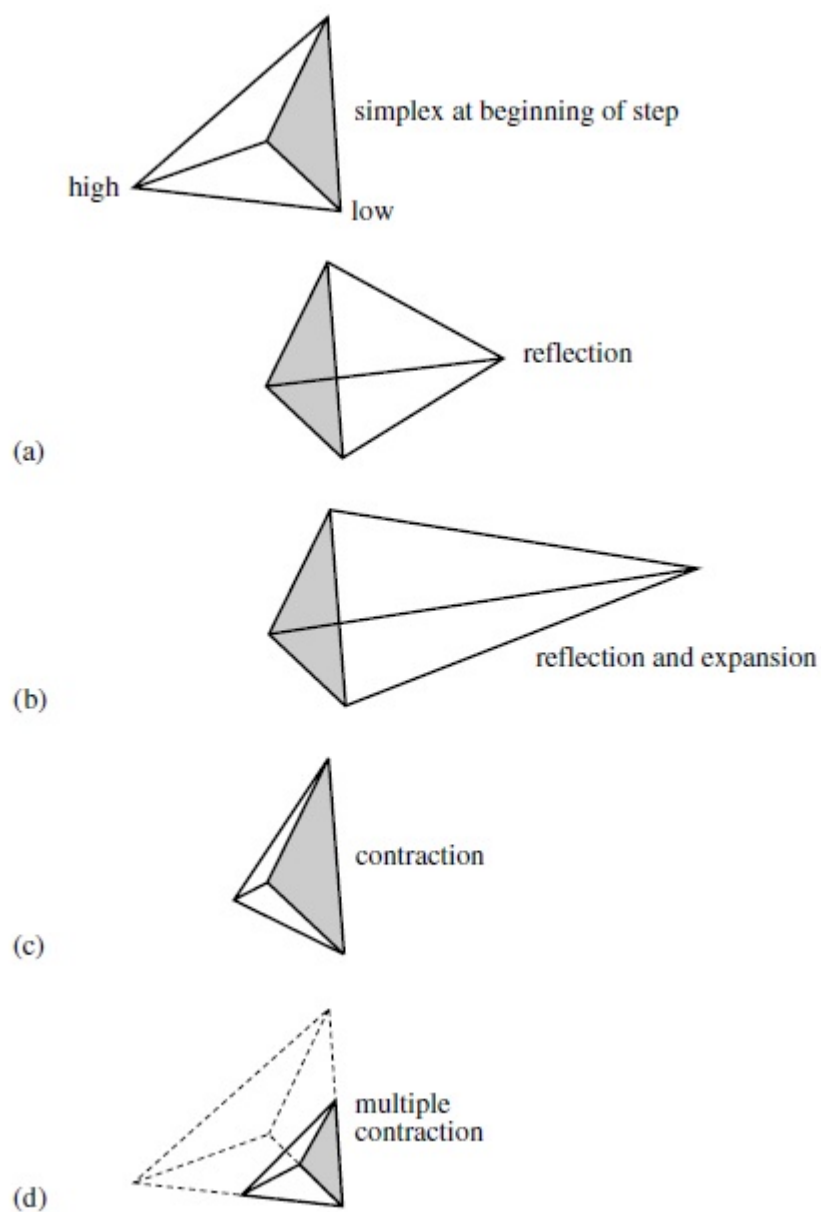


Figura 3.4: Possibili esiti di un'iterazione del DownHill Simplex, applicato ad un tetraedro. Un'iterazione del metodo può produrre (a) una riflessione rispetto al punto di massimo, (b) una riflessione ed un'espansione rispetto al punto di massimo, (c) una contrazione lungo una dimensione rispetto al punto di massimo, oppure (d) una contrazione rispetto a tutte le dimensioni verso il punto di minimo. Un'adequata sequenza di tali steps convergerà sempre al minimo della funzione.

Downhill simplex considera tre parametri per il controllo sui movimenti del semplice. La scelta di tali valori influenza la velocità e l'accuratezza del metodo. In particolare,  $\alpha \in \mathbb{R}$  controlla la *riflessione*,  $\beta \in \mathbb{R}$  la *contrazione* e  $\gamma \in \mathbb{R}$  l'*espansione*.

### 3.4.6 Simulated Annealing

Simulated Annealing è un metodo di ottimizzazione globale, Capace di produrre risultati rimarchevoli anche nel caso in cui l'ottimo globale sia circondato da ottimi locali.

Il metodo si ispira alla termodinamica, in particolare al modo in cui si congelano e si cristallizzano i liquidi. In natura, la mobilità termica degli atomi decresce per i liquidi in raffreddamento e si annulla nel caso dei cristalli, costituendo lo stato a minima energia del sistema. Il metodo Simulated Annealing conserva questa idea e consente di effettuare movimenti nella direzione in cui la soluzione sembra allontanarsi dall'ottimo, in dipendenza della "temperatura". Conseguentemente, risulta possibile che l'algoritmo abbandoni un ottimo locale, aumentando la probabilità di trovare l'ottimo globale. La distribuzione di probabilità di Boltzmann regola la verosomiglianza nell'esplorazione dello spazio dei parametri, permettendo temporanei allontanamenti dall'ottimo.

Alla procedura di annealing, appena descritta, segue una procedura iterativa per il calcolo della soluzione ottima. Affinché l'algoritmo di ricerca dell'ottimo globale sia robusto, ovvero si comporti in modo soddisfacente anche in presenza di avvallamenti sottili, [3] propone una modifica dell'algoritmo Downhill Simplex, consistente nel porre in cascata Simulated Annealing a DownHill Simplex.

In tal modo, una volta selezionata una candidata direzione per la ricerca dell'ottimo, interviene l'ulteriore filtro di Simulated Annealing: mediante la distribuzione di Boltzmann si calcola la probabilità di effettuare un'iterazione nella direzione identificata da Downhill Simplex.

Gli elementi principali del metodo Simulated Annealing (SA) sono i seguenti:

1. Un insieme finito  $S$ .
2. Una funzione costo  $J$  definita su  $S$ . Sia  $S^* \subset S$  l'insieme dei minimi globali della funzione  $J$ , ipotizzando che sia un sottoinsieme proprio di  $S$ .
3. Per ogni  $i \in S$ , un insieme  $S(i) \subset S - \{i\}$ , intorno di  $i$ .
4. Per ogni  $i$ , una famiglia positiva di coefficienti  $q_{ij}$ ,  $j \in S(i)$ , tale che  $\sum_{j \in S(i)} q_{ij} = 1$ . Si assume che  $j \in S(i)$  se e solo se  $i \in S(j)$ .

5. Una funzione non crescente  $T : \mathbb{N} \rightarrow (0, +\infty)$ , definita *cooling schedule*.  $T(t)$  denota la *temperatura* al tempo  $t$ .
6. Uno stato iniziale  $x(0) \in S$ .

Dati i precedenti elementi, l'algoritmo SA consiste in una catena di Markov inhomogenea, a tempo discreto, avente la seguente legge evolutiva: se lo stato attuale è uguale ad  $i$ , si sceglie in modo casuale un punto del suo intorno; la probabilità che il punto  $j \in S(i)$  sia scelto è uguale a  $q_{ij}$ . Una volta scelto  $j$ , lo stato successivo è determinato come segue:

- Se  $J(j) \leq J(i)$ , allora  $x(t+1) = j$ .
- Se  $J(j) > J(i)$ , allora  $x(t+1) = j$ , con probabilità  $\exp\{-J(j) - J(i)/T(t)\}$  altrimenti  $x(t+1) = i$

Formalmente,

$$\begin{aligned} \mathbb{P}[x(t+1) = j | x(t) = i] \\ = q_{ij} \exp\left(-\frac{1}{T(t)} \max\{0, J(j) - J(i)\}\right) \quad \text{se } j \neq i, j \in S(i). \end{aligned} \quad (3.53)$$

Se  $j \neq i$  e  $j \notin S(i)$ , allora  $\mathbb{P}[x(t+1) = j | x(t) = i] = 0$ .

L'idea dell'algoritmo SA è maggiormente comprensibile se si considera una catena di Markov omogenea  $x_T(t)$ , in cui la temperatura  $T(t)$  è una costante fissata  $T$ . Si assuma, inoltre, che  $x_T(t)$  sia irriducibile ed aperiodica, e che  $q_{ij} = q_{ji}$  per ogni  $i, j$ . Allora  $x_T(t)$  è una Catena di Markov reversibile<sup>1</sup>, la cui distribuzione invariante di probabilità è data da

$$\pi_T(i) = \frac{1}{Z_T} \exp\left[-\frac{J(i)}{T}\right], \quad i \in S, \quad (3.54)$$

ove  $Z_T$  è una costante di normalizzazione. È evidente che, per  $T \rightarrow 0$ , la distribuzione di probabilità  $\pi_T$  si restringe all'insieme  $S^*$  dei minimi globali di  $J$ . (Questa proprietà è valida anche rilassando  $q_{ij} = q_{ji}$ ).

La distribuzione di probabilità (3.54), nota come distribuzione di Boltzmann, gioca un importante ruolo nella meccanica statistica.

<sup>1</sup>(Criterio di Kolmogorov) Una Catena di Markov si dice *reversibile* se e solo se le probabilità di transizione soddisfano

$$p(j_1, j_2)p(j_2, j_3) \cdots p(j_{n-1}, j_n)p(j_n, j_1) = p(j_1, j_n)p(j_n, j_{n-1}) \cdots p(j_2, j_1)$$

per ogni sequenza di stati  $j_1, j_2, \dots, j_n \in E$ .

Per generare un campione dello spazio  $S$ , avente distribuzione  $\pi_T$ , è necessario simulare la Catena di Markov  $x_T(t)$  finché raggiunge l'equilibrio, secondo il metodo di rifiuto noto come *Algoritmo Metropolis*, descritto in [7]. Nel contesto di ottimizzazione, è possibile generare un elemento ottimo di  $S$  con alta probabilità se si calcola un campione casuale in accordo alla distribuzione  $\pi_T$ , ove  $T \rightarrow 0$ .

Una difficoltà intrinseca in questo approccio è rappresentata dal fatto che, per  $T \rightarrow 0$ , la catena di Markov  $x_T(t)$  può impiegare molto tempo per convergere al valore d'equilibrio. L'algoritmo SA accelera la convergenza introducendo una *cooling procedure*  $T(t)$  lentamente decrescente.

Pertanto, l'algoritmo SA può essere interpretato anche come un algoritmo di ricerca locale, in cui vi è probabilità non nulla di effettuare steps in direzioni di apparente allontanamento dall'ottimo, rendendo possibile l'uscita da un avvallamento o da un minimo locale.

## Capitolo 4

# G2++ Multi Currency

Nel presente Capitolo si considera il modello G2++ Multi Currency, formulato in [6], per la descrizione dell'evoluzione temporale del valore di una valuta *domestica* ed una valuta *estera*.

Il modello in esame si presenta come naturale estensione di G2++ Single Currency, le cui necessarie semplificazioni, volte a rendere risolvibile un sistema altrimenti sovravincolato, sono discusse nel prosieguo.

Nella prima sezione si richiamano alcune notazioni del precedente Capitolo per descrivere l'evoluzione temporale di due valute fissate, nelle rispettive misure risk-neutral.

Nella seconda sezione del presente Capitolo si descrive come modellizzare il tasso di cambio tra le due valute, riconducendo le dinamiche relative alle due curve ad un'unica misura risk-neutral. Si pone, inoltre, attenzione alle necessarie approssimazioni numerico-modellistiche proposte in [6].

Nella terza sezione si riportano le principali formule di pricing, e le loro derivazioni, dei principali prodotti finanziari su tasso di cambio: Quanto Constant-Maturity Swap, Quanto Caplet e Quanto Floorlet, Quanto Cap e Quanto Floor, Quanto Swaption.

Senza perdita di generalità, il modello può essere esteso ad un insieme di valute di cardinalità  $N$ , ove  $N > 2$ .

### 4.1 Descrizione delle dinamiche nelle rispettive misure risk-neutral

Si consideri la dinamica per un tasso a breve istantaneo, nella valuta 1, rispetto alla misura risk-neutral  $\mathbb{Q}_1$ , data da

$$r_1(t) = x_1(t) + y_1(t) + \varphi_1(t), \quad r_1(0) = r_0^1, \quad (4.1)$$

ove i processi  $\{x_1(t) : t \geq 0\}$  e  $\{y_1(t) : t \geq 0\}$  soddisfano

$$\begin{aligned} dx_1(t) &= -a_1x_1(t)dt + \sigma_1dW_1^x(t), & x_1(0) &= 0, \\ dy_1(t) &= -b_1y_1(t)dt + \eta_1dW_1^y(t), & y_1(0) &= 0, \end{aligned} \quad (4.2)$$

e  $W_1^x$  e  $W_1^y$  sono due moti Browniani correlati, tali che  $dW_1^x(t)dW_1^y(t) = \rho_1dt$ , e  $r_0^1$ ,  $a_1$ ,  $b_1$ ,  $\sigma_1$ ,  $\eta_1$ ,  $\rho_1$  sono costanti opportune. La funzione  $\varphi_1$  è deterministica e ben definita sull'intervallo  $[0, T]$ , ove  $T$  è un orizzonte temporale fissato. In particolare, si ha  $\varphi_1(0) = r_0^1$ . Si denoti con  $\mathcal{F}_t^1$  la sigma-algebra generata dalla coppia  $(x_1, y_1)$  fino al tempo  $t$ .

Integrando semplicemente le equazioni (4.2) si ottiene, per ogni  $s \leq t$ ,

$$\begin{aligned} r_1(t) &= x_1(s)e^{-a_1(t-s)} + y_1(s)e^{-b_1(t-s)} + \sigma_1 \int_s^t e^{-a_1(t-u)} dW_1^x(u) \\ &\quad + \eta_1 \int_s^t e^{-b_1(t-u)} dW_1^y(u) + \varphi_1(t). \end{aligned}$$

Il processo  $r_1(t)$ , condizionato alla filtrazione  $\mathcal{F}_s$ , risulta normalmente distribuito, con media e varianza date rispettivamente da:

$$\begin{aligned} \mathbb{E}[r_1(t)|\mathcal{F}_s^1] &= x_1(s)e^{-a_1(t-s)} + y_1(s)e^{-b_1(t-s)} + \varphi_1(t), \\ \text{Var}(r_1(t)|\mathcal{F}_s^1) &= \frac{\sigma_1^2}{2a_1}[1 - e^{-2a_1(t-s)}] + \frac{\eta_1^2}{2b_1}[1 - e^{-2b_1(t-s)}] \\ &\quad + 2\rho_1 \frac{\sigma_1\eta_1}{a_1 + b_1}[1 - e^{-(a_1+b_1)(t-s)}]. \end{aligned}$$

La seconda curva è modellata in modo analogo. Si assume, infatti, che la dinamica per il tasso a breve istantaneo, nella valuta 2, rispetto alla misura risk-neutral  $\mathbb{Q}_2$ , sia data da

$$r_2(t) = x_2(t) + y_2(t) + \varphi_2(t), \quad r_2(0) = r_0^2, \quad (4.3)$$

ove i processi  $\{x_2(t) : t \geq 0\}$  e  $\{y_2(t) : t \geq 0\}$  soddisfano

$$\begin{aligned} dx_2(t) &= -a_2x_2(t)dt + \sigma_2dW_2^x(t), & x_2(0) &= 0, \\ dy_2(t) &= -b_2y_2(t)dt + \eta_2dW_2^y(t), & y_2(0) &= 0, \end{aligned}$$

e  $W_2^x$  e  $W_2^y$  sono due moti Browniani correlati, tali che  $dW_2^x(t)dW_2^y(t) = \rho_2dt$ , e  $r_0^2$ ,  $a_2$ ,  $b_2$ ,  $\sigma_2$ ,  $\eta_2$ ,  $\rho_2$  sono costanti opportune. La funzione  $\varphi_2$  è deterministica e ben definita sull'intervallo  $[0, T]$ , ove  $T$  è un orizzonte temporale fissato. In particolare, si ha  $\varphi_2(0) = r_0^2$ . Si denoti con  $\mathcal{F}_t^2$  la sigma-algebra generata dalla coppia  $(x_2, y_2)$  fino al tempo  $t$ .

Avendo descritto l'evoluzione temporale delle due valute considerate, ognuna rispetto alla propria misura risk-neutral, risulta necessario identificare e quantificare le correlazioni tra i fattori di rischio delle due curve:



$$d \begin{bmatrix} W_1^x \\ W_1^y \\ W_2^x \\ W_2^y \end{bmatrix} d[W_1^x W_1^y W_2^x W_2^y] = \begin{bmatrix} 1 & \rho_1 & \gamma_{x_1,x_2} & \gamma_{x_1,y_2} \\ \cdot & 1 & \gamma_{y_1,x_2} & \gamma_{y_1,y_2} \\ \cdot & \cdot & 1 & \rho_2 \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} dt,$$

ove gli elementi non specificati sono determinati per simmetria. La matrice di correlazione risulta, ovviamente semidefinita positiva.

- La correlazione istantanea degli shocks tra il primo ed il secondo fattore della prima curva corrisponde a  $\rho_1$ , precedentemente definita.
- La correlazione istantanea degli shocks tra il primo fattore della prima curva ed il primo fattore della seconda curva è definita con il nuovo parametro  $\gamma_{x_1,x_2}$
- La correlazione istantanea degli shocks tra il primo fattore della prima curva ed il secondo fattore della seconda curva è definita con il nuovo parametro  $\gamma_{x_1,y_2}$
- La correlazione istantanea degli shocks tra il secondo fattore della prima curva ed il secondo fattore della seconda curva è definita con il nuovo parametro  $\gamma_{y_1,y_2}$
- La correlazione istantanea degli shocks tra il secondo fattore della prima curva ed il primo fattore della seconda curva è definita con il nuovo parametro  $\gamma_{y_1,x_2}$
- La correlazione istantanea degli shocks tra il primo fattore della seconda curva ed il secondo fattore della seconda curva corrisponde a  $\rho_2$ , precedentemente definita.

Per un'immediata interpretazione delle correlazioni, tipicamente si preferisce focalizzare l'attenzione sulle correlazioni tra i processi  $r_1$  ed  $r_2$ , piuttosto che sulle correlazioni tra i singoli fattori delle due curve. Si osserva, infatti, che

$$\text{Corr}\{dr_1, dr_2\} = \frac{\sigma_1 + \sigma_2\gamma_{x_1,x_2} + \eta_1\sigma_2\gamma_{y_1,x_2} + \sigma_1\eta_2\gamma_{x_1,y_2} + \eta_1\eta_2\gamma_{y_1,y_2}}{\sqrt{\sigma_1^2 + \eta_1^2 + 2\rho_1\sigma_1\eta_1}\sqrt{\sigma_2^2 + \eta_2^2 + 2\rho_2\sigma_2\eta_2}} \quad (4.4)$$

L'approccio suggerito in [6] consiste nel seguente algoritmo:

1. Effettuare una calibrazione per i parametri  $a_i, b_i, \sigma_i, \eta_i, \rho_i$  sul mercato dei Cap o delle swaptions relativo alla curva "i, ove  $i = 1, 2$ .
2. Simulare indipendentemente l'evoluzione delle due curve di tasso, come se le valute considerate esistessero in due mercati tra loro non comunicanti.

3. Poiché l'equazione (4.4) contiene 4 incognite, effettuare la seguente semplificazione

$$\gamma_{x_1, x_2} = \gamma_{y_1, x_2} = \gamma_{x_1, y_2} = \gamma_{y_1, y_2} := \gamma,$$

ottenendo

$$\gamma = \text{Corr}\{dr_1, dr_2\} \frac{\sqrt{\sigma_1^2 + \eta_1^2 + 2\rho_1\sigma_1\eta_1} \sqrt{\sigma_2^2 + \eta_2^2 + 2\rho_2\sigma_2\eta_2}}{(\sigma_1 + \eta_1)(\sigma_2 + \eta_2)}. \quad (4.5)$$

## 4.2 Dinamica del tasso di cambio rispetto ad un'unica Forward-measure

Nella presente sezione si illustra come esprimere le dinamiche dei fattori  $x_2$  ed  $y_2$ , relativi alla curva "2", rispetto alla misura risk-neutral relativa alla curva "1". In altri termini, si descrive l'evoluzione temporale del tasso di cambio tra le valute "1" e "2". Sia  $X(t)$  l'ammontare di valuta "2" necessario per comprare un'unità della valuta "1".

Si assuma la seguente dinamica per il processo  $X$ , rispetto alla misura  $\mathbb{Q}_2$ :

$$dX(t) = [(r_2(t) - r_1(t)) X(t)dt + \nu X(t) dW^X(t)], \quad (4.6)$$

ove  $\nu$  rappresenta la volatilità annua percentuale del tasso di cambio tra le valute "1" e "2".

Si desidera, inoltre, modellare le correlazioni istantanee tra i tassi di cambio  $X$  ed i due fattori della curva "2",  $x_2$  e  $y_2$ . Si consideri la seguente matrice di correlazione

$$d \begin{bmatrix} W^X \\ W_2^x \\ W_2^y \end{bmatrix} d[W^X \ W_2^x \ W_2^y] = \begin{bmatrix} 1 & c_{x_2, X} & c_{y_2, X} \\ \cdot & 1 & \rho_2 \\ \cdot & \cdot & 1 \end{bmatrix} dt$$

ove le componenti non specificate sono determinate per simmetria. Si definiscono:

- $c_{x_2, X}$ , correlazione istantanea degli shocks tra il primo fattore della seconda curva ed il tasso di cambio tra le valute "1" e "2",
- $c_{y_2, X}$ , correlazione istantanea degli shocks tra il secondo fattore della seconda curva ed il tasso di cambio tra le valute "1" e "2",

La correlazione istantanea degli shocks tra il primo fattore della seconda curva ed il secondo fattore della seconda curva,  $\rho_2$ , è stata precedentemente definita.

Analogamente alla sezione precedente, invece di considerare la correlazione tra il tasso di cambio  $X$  ed i fattori del tasso estero, si preferisce quantificare la correlazione tra  $X$  ed il tasso “2”, come evidenziato dalla seguente equazione:

$$\text{Corr}\{dX, dr_2\} = \frac{\sigma_2 c_{x_2, X} + \eta_2 c_{y_2, X}}{\sqrt{\sigma_2^2 + \eta_2^2 + 2\rho_2 \sigma_2 \eta_2}}. \quad (4.7)$$

Una semplificazione, suggerita in [6], è la seguente:

$$c_{x_2, X} = c_{y_2, X} = c_X.$$

Trattandosi di una relazione invertibile, è possibile esplicitare il valore di  $c_X$ :

$$c_X = \text{Corr}\{dX, dr_2\} \frac{\sqrt{\sigma_2^2 + \eta_2^2 + 2\rho_2 \sigma_2 \eta_2}}{\sigma_2 + \eta_2}. \quad (4.8)$$

É dunque possibile esprimere le equazioni relative ai fattori  $x_2$  e  $y_2$ , descriventi la curva “2” rispetto alla misura risk-neutral  $\mathbb{Q}_1$ , associata alla valuta “1”. Si utilizza la tecnica del cambio di numerario per trasformare il numerario della valuta “2” nel numerario

$$X(t) \times (\text{bank account nella valuta “1”}),$$

ottenendo dunque

$$\begin{aligned} dx_2(t) &= [-a_2 x_2(t) + \sigma_2 \nu c_{x_2, X}] dt + \sigma_2 dU_2^x(t) \\ dy_2(t) &= [-b_2 y_2(t) + \eta_2 \nu c_{y_2, X}] dt + \eta_2 dU_2^y(t), \end{aligned} \quad (4.9)$$

ove  $U_2^x$  e  $U_2^y$  sono due moti Browniani rispetto alla misura  $\mathbb{Q}_1$ , la cui struttura di correlazione è la medesima di  $W_2^x$  e  $W_2^y$ .

## 4.3 Pricing dei principali prodotti finanziari

### 4.3.1 Quanto Constant-Maturity Swap (QCMS)

#### Descrizione del contratto

Si consideri il seguente contratto. La controparte A paga alla controparte B una somma nella valuta “1”, espressa dal tasso swap a  $c$ -anni associato alla valuta “2”. I pagamenti avvengono ogni  $\delta$  anni (tipicamente  $\delta = 0.5$  anni). Sulle stesse date di pagamento, la controparte B paga alla controparte A il tasso LIBOR, associato alla valuta “1”, in Capitalizzazione semplice all’anno  $\delta$ . Formalmente, il payoff può essere espresso come segue.

Si consideri un nominale unitario. Si consideri l’istante attuale come  $t = 0$  e si denoti con  $\mathcal{T} = \{T_1, \dots, T_n\}$  l’insieme delle date di pagamento in cui i

flussi delle due gambe sono scambiati. Si denoti con  $\tau_i$  la year fraction tra  $T_{i-1}$  e  $T_i$  che, in generale, sarà molto vicina a  $\delta$ . Si consideri  $T_0 = 0$ .

- Al tempo  $T$ ,  $i \geq 1$ , la controparte A paga a B, nella valuta “1”, l’ammontare corrispondente al tasso swap a  $c$  anni  $S_{i-1, i-1+c/\delta}(T_{i-1})$  rispetto alla curva “2”, come fissato all’istante  $T_{i-1}$ :

$$S_{i-1, i-1+c/\delta}(T_{i-1})\tau_i \text{ unità di valuta 1.}$$

In generale, il tasso swap al tempo  $t \leq T_{j-1}$  per uno swap i cui pagamenti si verificano agli istanti  $T_j, T_{j+1}, \dots, T_{m-1}$  e con prima data di reset in  $T_{j-1}$  è definito come

$$S_{j-1, m-1}(t) = \frac{P_2(t, T_{j-1}) - P_2(t, T_{m-1})}{\sum_{k=j}^{m-1} \tau_k P_2(t, T_k)} \quad (4.10)$$

(i cui indici sono shiftati di uno per convenienza di notazione). In generale,  $P_2(t, T)$  denota il fattore di sconto per la curva “2” al tempo  $t$  per una maturity  $T$ . Analogamente per “1”.

- Al tempo  $T_i$ ,  $i \geq 1$ , l’Istituzione B paga ad A il tasso LIBOR relativo alla curva “1”:

$$\tau_i L_1(T_{i-1}, T_i) \text{ unità di valuta “1”.}$$

La quantità

$$L_1(T_{i-1}, T_i) = \frac{1}{\tau_i} \left( \frac{1}{P_1(T_{i-1}, T_i)} - 1 \right)$$

denota il tasso LIBOR al tempo  $T_{i-1}$  per la maturity  $T_i$  per la curva “1”.

Il valore netto del contratto, visto da B, al tempo 0 è

$$\begin{aligned} & \mathbb{E}_1 \left\{ \sum_{i=1}^n \exp \left( - \int_0^{T_i} r_1(s) ds \right) [S_{i-1, i-1+c/\delta}(T_{i-1}) - L_1(T_{i-1}, T_i)] \tau_i \right\} \\ & = P_1(0, T_n) \sum_{i=1}^n \tau_i \mathbb{E}_1^n \left\{ \frac{S_{i-1, i-1+c/\delta}(T_{i-1})}{P_1(T_i, T_n)} \right\} - (1 - P_1(0, T_n)), \end{aligned}$$

ove  $\mathbb{E}_1$  denota il valore atteso rispetto alla misura risk-neutral per la curva “1”, ed  $\mathbb{E}_1^n$  è il valore atteso rispetto alla  $T_i$ -forward measure  $\mathbb{Q}_1^{T_i}$  per la curva “1”. Si deve calcolare

$$\mathbb{E}_1^n [S_{i-1, i-1+c/\delta}(T_{i-1}) / P_1(T_i, T_n)] \quad (4.11)$$

per ogni  $i$ . Si può utilizzare un modello di tasso a breve con formule analitiche per calcolare i prezzi degli Zero-Coupon Bond, nei termini dei fattori

che compongono il tasso a breve. In tal caso, (4.10) rappresenta una formula analitica per i tassi swap, in termini di tali fattori, ed è sufficiente simulare i paths dei fattori dei tassi a breve fino al tempo  $T_n$ , rispetto alla misura  $\mathbb{Q}_1^n$ . Se tale dinamica è lineare, le densità di transizione sono Gaussiane. La transizione dei fattori del tasso a breve tra due istanti  $T_{i-1}$ ,  $T_i$  può essere dunque simulata one-shot da tali densità Gaussiane, senza dover effettuare una discretizzazione temporale sulla dinamica tra  $T_{i-1}$  e  $T_i$ .

Allo stesso tempo, il modello scelto deve permettere una struttura di volatilità realistica per ogni curva, ed inoltre la correlazione tra diversi tassi deve essere modellata nella curva “2”, dal momento che sono coinvolti dei tassi swap. Una buona scelta può essere, dunque, quella di un modello Gaussiano a due fattori.

### QCMS nel modello G2++

Partendo dal modello appena descritto, si deducono le formule analitiche richieste per il payoff di un contratto Quanto-CMS.

Il modello ammette la seguente formula per lo Zero-Coupon Bond della curva “1”:

$$P_1(t, T; x_1(t), y_1(t)) = \frac{P_1^M(0, T)}{P_1^M(0, t)} \exp \left\{ \frac{1}{2} [V_1(t, T) - V_1(0, T) + V_1(0, t)] - \frac{1 - e^{-a_1(T-t)}}{a_1} x_1(t) - \frac{1 - e^{-b_1(T-t)}}{b_1} y_1(t) \right\}$$

ove  $V_1(t, T)$  è definito in modo analogo a (3.10). La corrispondente formula per lo Zero-Coupon Bond della curva “2” è analoga. Basta sostituire “2” in luogo di “1”.

I tassi swap possono essere calcolati analiticamente in termini di  $x$  e  $y$  dalle precedenti formule per i fattori di sconto. Per il tasso swap (forward) della curva “2” si può riscrivere (4.10) in termini delle quantità del modello:

$$S_{j-1, m-1}(t; x_2(t), y_2(t)) = \frac{P_2(t, T_{j-1}; x_2(t), y_2(t)) - P_2(t, T_{m-1}; x_2(t), y_2(t))}{\sum_{k=j}^{m-1} \tau_k P_2(t, T_k; x_2(t), y_2(t))} \quad (4.12)$$

Per calcolare (4.11), è necessaria la dinamica di  $x_1$ ,  $y_1$ ,  $x_2$  e  $y_2$  rispetto alla  $T_n$ -forward measure  $\mathbb{Q}_1^n$  della curva “1”.

A tal proposito, in (4.9) si sono formulate le equazioni per i fattori  $x_2$  e  $y_2$  rispetto alla misura risk-neutral  $\mathbb{Q}_1$ , associata al mercato “1”. Dopo aver modellato il tasso di cambio tra le valute “1” e “2”, si è descritta la correlazione istantanea tra il tasso di cambio “x” ed i due fattori,  $x_2$  e  $y_2$ , della curva “2”, ottenendo la dinamica (4.9).

Si consideri ora la dinamica per  $x_1$ ,  $y_1$ ,  $x_2$  e  $y_2$ , come da (4.2) e (4.9). Un secondo cambio di numerario per il mercato “1” può essere ora effettuato,

passando dal bank account nella valuta “1” al prezzo dello Zero-Coupon Bond nella valuta “1” con maturity  $T_n$ . Si fissi  $T_n = T$ . Applicando nuovamente il cambio di numerario, si ottengono le seguenti dinamiche:

$$dx_1(t) = \left[ -a_1 x_1(t) - \frac{\sigma_1^2}{a_1} (1 - e^{-a_1(T-t)}) - \rho_1 \frac{\sigma_1 \eta_1}{b_1} (1 - e^{-b_1(T-t)}) \right] dt + \sigma_1 dZ_1^x(t),$$

$$dy_1(t) = \left[ -b_1 y_1(t) - \frac{\eta_1^2}{b_1} (1 - e^{-b_1(T-t)}) - \rho_1 \frac{\sigma_1 \eta_1}{a_1} (1 - e^{-a_1(T-t)}) \right] dt + \eta_1 dZ_1^y(t),$$

$$dx_2(t) = \left[ -a_2 x_2(t) - \sigma_2 \left( \gamma_{x_1, x_2} \frac{\sigma_1}{a_1} (1 - e^{-a_1(T-t)}) + \gamma_{x_2, y_1} \frac{\eta_1}{b_1} (1 - e^{-b_1(T-t)}) - \nu c_{x_2, X} \right) \right] dt + \sigma_2 dZ_2^x(t),$$

$$dy_2(t) = \left[ b_2 y_2(t) - \eta_2 \left( \gamma_{y_1, y_2} \frac{\eta_1}{b_1} (1 - e^{-b_1(T-t)}) + \gamma_{x_1, y_2} \frac{\sigma_1}{a_1} (1 - e^{-a_1(T-t)}) - \nu c_{y_2, X} \right) \right] dt + \eta_2 dZ_2^y(t)$$

ove  $Z$  sono moti Browniani rispetto alla misura  $\mathbb{Q}_1^n$  con la stessa struttura di correlazione dei moti  $W$ .

Integrando il precedente processo Gaussiano quadri-dimensionale ed utilizzando una versione quadri-dimensionale dell'Isometria di Ito, si ottiene la seguente densità di transizione esatta:

$$\begin{aligned} x_1(t) &= e^{-a_1(t-s)} x_1(s) - M_{x_1}^T(s, t) + N_1(t-s), \\ y_1(t) &= e^{-b_1(t-s)} y_1(s) - M_{y_1}^T(s, t) + N_2(t-s), \\ x_2(t) &= e^{-a_2(t-s)} x_2(s) - M_{x_2}^T(s, t) + N_3(t-s), \\ y_2(t) &= e^{-b_2(t-s)} y_2(s) - M_{y_2}^T(s, t) + N_4(t-s), \end{aligned} \tag{4.13}$$

per  $s \leq t \leq T$ , ove

$$\begin{aligned} M_{x_1}^T(s, t) &= \left( \frac{\sigma_1^2}{a_1^2} + \rho_1 \frac{\sigma_1 \eta_1}{a_1 b_1} \right) [1 - e^{-a_1(t-s)}] \\ &\quad - \frac{\sigma_1^2}{2a_1^2} [e^{-a_1(T-t)} - e^{-a_1(T+t-2s)}] \\ &\quad - \frac{\rho_1 \sigma_1 \eta_1}{b_1(a_1 + b_1)} [e^{-b_1(T-t)} - e^{-b_1 T - a_1 T - a_1 t + (a_1 + b_1)s}] \end{aligned}$$

$$\begin{aligned}
M_{y_1}^T(s, t) &= \left( \frac{\eta_1^2}{b_1^2} + \rho_1 \frac{\sigma_1 \eta_1}{a_1 b_1} \right) \left[ 1 - e^{b_1(t-s)} \right] \\
&\quad - \frac{\eta_1^2}{2b_1^2} \left[ e^{-b_1(T-t)} - e^{-b_1(T+t-2s)} \right] \\
&\quad - \frac{\rho_1 \sigma_1 \eta_1}{a_1(a_1 + b_1)} \left[ e^{-a_1(T-t)} - e^{-a_1 T - b_1 t + (a_1 + b_1)s} \right]
\end{aligned}$$

$$\begin{aligned}
M_{x_2}^T(s, t) &= \frac{\sigma_1 \sigma_2 \gamma_{x_1, x_2}}{a_1} \left( \frac{1 - e^{-a_2(t-s)}}{a_2} - \frac{e^{-a_1(T-t)} - e^{-a_1(T-s) - a_2(t-s)}}{a_1 + a_2} \right) \\
&\quad + \frac{\eta_1 \sigma_2 \gamma_{x_2, y_1}}{b_1} \left( \frac{1 - e^{a_2(t-s)}}{a_2} - \frac{e^{-b_1(T-t)} - e^{-b_1(T-s) - a_2(t-s)}}{b_1 + a_2} \right) \\
&\quad - \sigma_2 \nu c_{X, x_2} \frac{1 - e^{-a_2(t-s)}}{a_2}
\end{aligned}$$

$$\begin{aligned}
M_{y_2}^T(s, t) &= \frac{\eta_1 \eta_2 \gamma_{y_1, y_2}}{b_1} \left( \frac{1 - e^{-b_2(t-s)}}{b_2} - \frac{e^{-b_1(T-t)} - e^{-b_1(T-s) - b_2(t-s)}}{b_1 + b_2} \right) \\
&\quad + \frac{\sigma_1 \eta_2 \gamma_{y_2, x_1}}{a_1} \left( \frac{1 - e^{-b_2(t-s)}}{b_2} - \frac{e^{-a_1(T-t)} - e^{-b_1(T-s) - b_2(t-s)}}{a_1 + b_2} \right) \\
&\quad - \eta_2 \nu c_{X, y_2} \frac{1 - e^{b_2(t-s)}}{b_2}
\end{aligned}$$

ove  $N(t-s)$  è un vettore Gaussiano quadri-dimensionale di media 0 e matrice di covarianza  $C(t-s)$  data da

$$C(t-s) = \begin{bmatrix} \sigma_1^2 \frac{1 - e^{-2a_1(t-s)}}{2a_1} & \cdot & \cdot & \cdot \\ \sigma_1 \eta_1 \rho_1 \frac{1 - e^{-(a_1 + b_1)(t-s)}}{a_1 + b_1} & \eta_1^2 \frac{1 - e^{-2b_1(t-s)}}{2b_1} & \cdot & \cdot \\ \sigma_1 \sigma_2 \gamma_{x_1, x_2} \frac{1 - e^{-(a_1 + a_2)(t-s)}}{a_1 + a_2} & \eta_1 \sigma_2 \gamma_{y_1, x_2} \frac{1 - e^{-(b_1 + a_2)(t-s)}}{b_1 + a_2} & (***) & \cdot \\ \sigma_1 \eta_2 \gamma_{x_1, y_2} \frac{1 - e^{-(a_1 + b_2)(t-s)}}{a_1 + b_2} & \eta_1 \eta_2 \gamma_{y_1, y_2} \frac{1 - e^{-(b_1 + b_2)(t-s)}}{b_1 + b_2} & (***) & (*) \end{bmatrix}$$

$$(***) = \sigma_2^2 \frac{1 - e^{-2a_2(t-s)}}{2a_2}$$

$$(**) = \eta_2 \sigma_2 \rho_2 \frac{1 - e^{-(b_2 + a_2)(t-s)}}{b_2 + a_2}, \quad (*) = \eta_2^2 \frac{1 - e^{-2b_2(t-s)}}{b_2}$$

Sia  $C$  una matrice di covarianza. Al fine di generare un vettore di variabili aleatorie, avente matrice di varianza-covarianza assegnata, è possibile seguire il seguente algoritmo:

1. effettuare la decomposizione di Cholesky di  $C$ :

$$C = LL^T$$

ove  $L$  è una matrice triangolare inferiore,

2. generare un vettore  $X$  di variabili aleatorie incorrelate,
3. applicare la proiezione  $L$  ad  $X$

$$Y = LX,$$

ottenendo  $Y$ , vettore di variabili aleatorie correlate.

### Pricing Monte Carlo del contratto QCMS

Il prezzo del contratto, visto dalla controparte B, al tempo 0 è

$$\begin{aligned} \mathbb{E}^1 \left\{ \sum_{i=1}^n \exp \left( - \int_0^{T_i} r_1(s) ds \right) [S_{i-1, i-1+c/\delta}(T_{i-1}) - L_1(T_{i-1}, T_i)] \tau_i \right\} \\ = P_1^M(0, T_n) \sum_{i=1}^n \tau_i \mathbb{E}_1^n \left[ \frac{S_{i-1, i-1+c/\delta}(T_{i-1}; x_2(T_{i-1}), y_2(T_{i-1}))}{P_1(T_i), T_n, x_1(T_i), y_1(T_i)} \right] \\ - (1 - P_1^M(0, T_n)). \end{aligned} \quad (4.14)$$

È dunque necessario calcolare

$$\mathbb{E}_1^n \left[ \frac{S_{i-1, i-1+c/\delta}(T_{i-1}; x_2(T_{i-1}), y_2(T_{i-1}))}{P_1(T_i), T_n, x_1(T_i), y_1(T_i)} \right] \quad (4.15)$$

per ogni  $i$  e sostituire nella precedente formula. Si procede nel seguente modo:

**Inputs.** Si descrivono gli inputs necessari per la valutazione del contratto.

1. Il numero  $np$  di scenari della valutazione Monte Carlo
2. Le curve iniziali “1” e “2” al tempo 0, intese come fattori di sconto, possibilmente interpolati):  $T \rightarrow P_1^M(0, T)$  e  $T \rightarrow P_2^M(0, T)$ .
3. I parametri  $a_1, b_1, \sigma_1, \eta_1$  e  $\rho_1$ , ottenuti calibrando il modello (4.1) al mercato dei Cap o delle swaption nella valuta “1”.
4. I parametri  $a_2, b_2, \sigma_2, \eta_2$  e  $\rho_2$ , ottenuti calibrando il modello (4.3) al mercato dei Cap o delle swaption nella valuta “2”.
5. Le correlazioni istantanee tra le due curve,  $\text{Corr} \{dr_1, dr_2\}$ , da cui può essere calcolato il corrispondente  $\gamma$  mediante la (4.5).



6. La correlazione istantanea tra la curva “2” ed il tasso di cambio “1” - “2”,  $\text{Corr}\{dX, dr_2\}$ , da cui può essere calcolato il corrispondente parametro  $c_X$ , attraverso la (4.8)
7. La volatilità annua percentuale del tasso di cambio “1” - “2”.

**Schema.** Si descrive l’algoritmo per calcolare il prezzo di un contratto CMS, mediante simulazioni Monte Carlo.

1. Si fissino l’istante iniziale  $t$  a  $t = T_0 = 0$ ,  $x_1(T_0) = y_1(T_0) = x_2(T_0) = y_2(T_0) = 0$ . Sia ora  $i = 1$ .
2. Se  $i \geq 2$ , si calcolino i seguenti fattori di sconto al tempo  $t = T_{i-1}$  per la curva “2”, in ogni scenario “p”, utilizzando la formula (4.3.1):

$$\begin{aligned} & P_2(T_{i-1}, T_i; x_2^p(T_{i-1}), y_2^p(T_{i-1})), \\ & P_2(T_{i-1}, T_{i+1}; x_2^p(T_{i-1}), y_2^p(T_{i-1})), \\ & \dots, \\ & P_2(T_{i-1}, T_{i-1+c/\delta}; x_2^p(T_{i-1}), y_2^p(T_{i-1})). \end{aligned}$$

Altrimenti, se  $i = 1$ , i fattori di sconto al tempo  $T_{i-1}$  sono noti in input.

3. Se  $i \geq 2$ , si calcoli il tasso swap al tempo  $t = T_{i-1}$  in ogni scenario “p”,  $S_{i-1, i-1+c/\delta}^p = S_{i-1, i-1+c/\delta}(T_{i-1}; x_2^p(T_{i-1}), y_2^p(T_{i-1}))$ , mediante la formula (4.12), ove  $j = i$ ,  $m = i + c/\delta$  e  $t = T_{i-1}$ .  
Se  $i = 1$ , i tassi swap al tempo  $T_{i-1}$  sono noti in input.

4. Utilizzare la formula (4.13) per generare  $np$  simulazioni

$$x_1^p(T_i), y_1^p(T_i), x_2^p(T_i), y_2^p(T_i), p = 1, 2, \dots, np$$

di  $x_1(T_i)$ ,  $y_1(T_i)$ ,  $x_2(T_i)$ ,  $y_2(T_i)$  partendo dai rispettivi valori, precedentemente generati  $x_1(T_{i-1})$ ,  $y_1(T_{i-1})$ ,  $x_2(T_{i-1})$ ,  $y_2(T_{i-1})$ . La formula (4.13) è applicata con  $s = T_{i-1}$  e  $t = T_i$  e con le  $np$  realizzazioni di  $N_1, \dots, N_4$ , generati da una variabile aleatoria Gaussiana quadrimensionale, avente media zero e matrice di covarianza  $C(T_i - T_{i-1})$ .

5. Calcolare, per la curva “1”, il fattore di sconto da  $T_n$  a  $t = T_i$  in ogni scenario, utilizzando la formula (4.3.1):

$$P_1(T_i, T_n; x_1^p(T_i), y_1^p(T_i)).$$

6. Calcolare in ogni scenario “p” il rapporto

$$\frac{S_{i-1, i-1+c/\delta}^p(T_{i-1})}{P_1(T_i, T_n, x_1(T_i), y_1(T_i))}.$$

7. Calcolare la media delle precedenti quantità, rispetto al numero di scenari. Si ottiene la valutazione Monte Carlo di (4.15). Memorizzare questo valore.
8. Incrementare  $i$  di un'unità.
9. Se  $i \leq n$  ritornare allo step 2, altrimenti
10. Calcolare il prezzo finale, aggiungendo tutti i termini calcolati e memorizzati allo step 7, in accordo alla formula (4.14).

### 4.3.2 Quanto Caplet/Floorlet

Dato un mercato domestico ed un mercato estero, si consideri la struttura a termine dei fattori di sconto osservati nei mercati domestico ed estero al tempo  $t$ , dati rispettivamente da  $T \rightarrow P(t, T)$  e  $T \rightarrow P^f(t, T)$  per  $T \geq t$ . Si denoti con  $\chi(t)$  il tasso di cambio al tempo  $t$  tra le valute dei due mercati, tale che 1 unità di valuta estera corrisponde a  $\chi(t)$  unità di valuta domestica. Si noti che  $\chi(t)$  rappresenta il reciproco del tasso di cambio  $X(t)$  precedentemente definito.

Fissate le date future  $T_1$  e  $T_2$ , una quanto Caplet che restituisce al tempo  $T_2$

$$\left[ F^f(T_1; T_1, T_2) - K \right]^+ \tau_{1,2} N \quad \text{in valuta domestica,} \quad (4.16)$$

ove  $N$  è il valore nominale,  $K$  è il tasso strike della Caplet,  $\tau_{1,2}$  è la year fraction tra le scadenze  $T_1$  e  $T_2$  e  $F^f(t; T_1, T_2)$  è il tasso forward in valuta estera al tempo  $t$  nell'intervallo  $[T_1, T_2]$ , i.e.,

$$F^f(t; T_1, T_2) = \frac{P^f(t, T_1) - P^f(t, T_2)}{\tau_{1,2} P^f(t, T_2)},$$

assumendo la medesima year fraction in entrambi i mercati. Il valore di non arbitraggio, al tempo  $t$ , del payoff (4.16) è dato da

$$\mathbf{QCpl}(t, T_1, T_2, N, K) = \tau_{1,2} N P(t, T_2) \mathbb{E}^2 \left\{ \left[ F^f(T_1; T_1, T_2) - K \right]^+ \mid \mathcal{F}_t \right\},$$

ove  $\mathbb{E}^2$  denota il valore atteso rispetto alla forward measure domestica  $\mathbb{Q}^2$ , indotta dal numerario  $P(t, T_2)$ .

Per calcolare tale valore atteso la distribuzione di  $F^f(T_1; T_1, T_2)$ , rispetto alla misura  $\mathbb{Q}^2$  deve essere nota. Rispetto alla forward measure estera, associata al numerario  $P^f(t, T_2)$ ,  $F^f(t; T_1, T_2)$  è una martingala e, pertanto, è possibile scrivere la seguente dinamica:

$$dF^f(t; T_1, T_2) = \sigma F^f(t; T_1, T_2) dW^f(t).$$

**Definizione 4.3.1.** Si definisce *tasso di cambio forward* al tempo  $t$ , avente maturity  $T_2$

$$F_\chi(t, T_2) = \chi(t) \frac{P^f(t, T_2)}{P(t, T_2)},$$

martingala rispetto alla misura  $\mathbb{Q}^2$ .

Si assuma che

$$dF_\chi(t, T_2) = \sigma_{F_\chi} F_\chi(t, T_2) dW_\chi(t),$$

ove  $W_\chi$  è un moto Browniano standard rispetto a  $\mathbb{Q}^2$ , tale che  $dW_\chi(t)dW^f(t) = \rho dt$ .

Allora, applicando il cambio di numerario, rispetto alla forward measure domestica  $\mathbb{Q}^2$  si ottiene

$$dF^f(t; T_1, T_2) = F^f(t; T_1, T_2) \left[ -\rho \sigma_{F_\chi} \sigma dt + \sigma dW_2^f(t) \right],$$

cosicché, rispetto a  $\mathbb{Q}^2$ ,  $F^f(t; T_1, T_2)$  è distribuita come una Lognormale avente rispettivamente media e varianza

$$\begin{aligned} \mathbb{E}^2 \left\{ \ln \frac{F^f(T_1; T_1, T_2)}{F^f(t; T_1, T_2)} \middle| \mathcal{F}_t \right\} &= \left( \mu - \frac{1}{2} \sigma^2 \right) (T_1 - t) \\ \text{Var}^2 \left\{ \ln \frac{F^f(T_1; T_1, T_2)}{F^f(t; T_1, T_2)} \middle| \mathcal{F}_t \right\} &= \sigma^2 (T_1 - t) \end{aligned}$$

ove  $\mu := -\rho \sigma_{F_\chi} \sigma$ .

Questo implica immediatamente che

$$\mathbf{QCpl}(t, T_1, T_2, N, K) = \tau_{1,2} NP(t, T_2) \left[ F^f(t; T_1, T_2) e^{\mu(T_1-t)} \phi(d_1) - K \phi(d_2) \right]$$

$$\begin{aligned} d_1 &= \frac{\ln \frac{F^f(t; T_1, T_2)}{K} + (\mu + \frac{1}{2} \sigma^2)(T_1 - t)}{\sigma \sqrt{T_1 - t}} \\ d_2 &= d_1 - \sigma \sqrt{T_1 - t} \end{aligned}$$

Analogamente, il prezzo di non arbitraggio di una quanto Floorlet, il cui payoff a scadenza, in valuta domestica, è

$$\left[ K - F^f(T_1; T_1, T_2) \right]^+ \tau_{1,2} N,$$

$$\mathbf{QFll}(t, T_1, T_2, N, K) = \tau_{1,2} NP(t, T_2) \left[ K \phi(-d_2) - F^f(t; T_1, T_2) e^{\mu(T_1-t)} \phi(-d_1) \right]$$

### 4.3.3 Quanto Cap/Floor

Si denoti con  $\mathcal{T} = \{T_0, T_1, \dots, T_n\}$  gli istanti corrispondenti alle date di pagamento del contratto Cap/Floor e sia  $\tau_i$  la year fraction corrispondente all'intervallo  $(T_{i-1}, T_i)$ ,  $i = 1, \dots, n$ . Sia  $\sigma_i$  la volatilità di  $F^f(t; T_{i-1}, T_i)$ ,  $\rho_i$  la correlazione tra  $F_\chi(t, T_i)$  e  $F^f(t; T_{i-1}, T_i)$  e sia  $\sigma_{F_\chi}^i$  la volatilità del tasso forward  $F_\chi(t, T_i)$ . Si fissi  $\tau := \{\tau_1, \dots, \tau_n\}$  e

$$\mu_i = -\rho_i \sigma_{F_\chi}^i \sigma_i.$$

Poiché il prezzo di un Cap (Floor) è dato dalla somma dei sottostanti Caplet (Floorlet), il prezzo al tempo  $t$  di un Cap avente tasso strike  $K$ , nominale  $N$  e date di pagamento  $\mathcal{T}$  è dato da

$$\begin{aligned} & \mathbf{QCap}(t, \mathcal{T}, \tau, N, K) \\ &= \sum_{i=1}^n \tau_i NP(t, T_i) \left[ F^f(t; T_{i-1}, T_i) e^{\mu_i(T_{i-1}-t)} \right. \\ & \quad \left. \Phi \left( \frac{\ln \frac{F^f(t; T_{i-1}, T_i)}{K} + (\mu_i + \frac{1}{2}\sigma_i^2)(T_{i-1}-t)}{\sigma_i \sqrt{T_{i-1}-T_i}} \right) \right. \\ & \quad \left. - K \Phi \left( \frac{\ln \frac{F^f(t; T_{i-1}, T_i)}{K} + (\mu_i + \frac{1}{2}\sigma_i^2)(T_{i-1}-t)}{\sigma_i \sqrt{T_{i-1}-t}} \right) \right], \end{aligned} \quad (4.17)$$

e, analogamente, il prezzo del Floor risulta

$$\begin{aligned} & \mathbf{QFlr}(t, \mathcal{T}, \tau, N, K) \\ &= \sum_{i=1}^n \tau_i NP(t, T_i) \left[ -F^f(t; T_{i-1}, T_i) e^{\mu_i(T_{i-1}-t)} \right. \\ & \quad \left. \Phi \left( \frac{\ln \frac{K}{F^f(t; T_{i-1}, T_i)} - (\mu_i + \frac{1}{2}\sigma_i^2)(T_{i-1}-t)}{\sigma_i \sqrt{T_{i-1}-T_i}} \right) \right. \\ & \quad \left. + K \Phi \left( \frac{\ln \frac{K}{F^f(t; T_{i-1}, T_i)} - (\mu_i + \frac{1}{2}\sigma_i^2)(T_{i-1}-t)}{\sigma_i \sqrt{T_{i-1}-t}} \right) \right]. \end{aligned} \quad (4.18)$$

### 4.3.4 Quanto Swaption

Sia  $\mathcal{T} = \{T = T_0, T_1, T_2, \dots, T_n\}$  un insieme di  $n+1$  date, tali che  $0 < T < T_1 < T_2 < \dots < T_n$ . Si assumano una valuta domestica ed una valuta estera, ove sono definiti i fattori di sconto ed i forward swap rates, corrispondenti alle date  $\mathcal{T}$ . Si denotino, rispettivamente con  $P(t, T_i)$  e  $P^f(t, T_i)$ , gli Zero-Coupon Bond domestico ed estero al tempo  $t$  per la maturity  $T_i$ ,  $i = 0, 1, \dots, n$ .

Si ipotizzi che il tasso di cambio evolva come il precedente processo  $\chi$ . Il

forward swap rate domestico al tempo  $0 < t < T$ , corrispondente allo swap avente inizio in  $T$  e date di pagamento  $T_i$ ,  $i = 0, 1, \dots, n$  è dato da

$$S(t) = \frac{P(t, T) - P(t, T_n)}{C(t)}, \quad (4.19)$$

ove

$$C(t) = \sum_{i=1}^n \tau_i P(t, T_i) \quad (4.20)$$

e  $\tau_i$  è la year fraction relativa all'intervallo  $[T_{i-1}, T_i]$ . Analogamente, il forward swap rate estero al tempo  $0 < t < T$  corrispondente allo swap avente inizio al tempo  $T$  e date di pagamento  $T_i$ ,  $i = 1, \dots, n$  è dato da

$$S^f(t) = \frac{P^f(t, T) - P^f(t, T_n)}{C^f(t)}, \quad (4.21)$$

ove

$$C^f(t) = \sum_{i=1}^n \tau_i P^f(t, T_i). \quad (4.22)$$

Si assumono, per semplicità, medesime year fractions per entrambe le valute. Dati i tassi swap (4.19) e (4.21), lo strike  $K > 0$  e la maturity  $T$ , il payoff al tempo  $T$

$$C(T) \left[ wS^f(T) - wS(T) - wK \right]^+ \quad (4.23)$$

ove  $w = 1$  nel caso call e  $w = -1$  nel caso put. Considerando il numerario  $C(t)$ , il prezzo di non arbitraggio al tempo  $t$  di (4.23) risulta

$$\mathbf{QES}(t, \mathcal{T}, \tau, K) = C(t) \mathbb{E}^C \left\{ \left[ wS^f(T) - wS(T) - wK \right]^+ \mid \mathcal{F}_t \right\}, \quad (4.24)$$

ove  $\mathbb{E}^C$  denota il valore atteso rispetto alla forward-swap measure domestica  $\mathbb{Q}^C$  indotta da  $C(t)$ , e  $\mathcal{F}_t$  denota la sigma-algebra generata da  $(S^f, S)$  al tempo  $t$ .

Per definizione di forward-swap measure domestica, il processo  $S$  è una martingala rispetto a tale misura. Assumendo dinamica lognormale, il forward swap domestico, rispetto alla misura  $\mathbb{Q}^C$  è dunque descritto da

$$dS(t) = \sigma S(t) dW(t) \quad (4.25)$$

ove  $\sigma$  è una costante reale positiva e  $W$  è un moto Browniano rispetto alla misura  $\mathbb{Q}^C$ . Il forward swap rate  $S^f$  è una martingala rispetto alla forward-swap measure estera. Si assume che, rispetto a  $\mathbb{Q}^C$  le seguenti martingale

$$M_1(t) = \frac{\chi(t)(P^f(t, T) - P^f(t, T_n))}{C(t)}$$

$$M_2(t) = \frac{\chi(t)C^f(t)}{C(t)}$$

siano distribuite in modo lognormale, ma anche  $S^f(t) = \frac{M_1(t)}{M_2(t)}$  sia distribuita in modo lognormale, con dinamica

$$dS^f(t) = S^f(t) \left[ \mu^f dt + \sigma^f dW^f(t) \right]$$

ove  $\mu^f$  e  $\sigma^f$  sono costanti reali e  $W^f$  è un moto Browniano rispetto a  $\mathbb{Q}^C$ . Inoltre, si nota che il drift  $\mu^f dt$  corrisponde alla covarianza istantanea, cambiata di segno, tra i logaritmi di  $M_2(t)$  e  $S^f(t)$ .

Per applicazioni pratiche  $M_2(t)$  può essere sostituito dal tasso di cambio forward al tempo  $T$ .

$$F_\chi(t, T) = \frac{\chi(t)P(t, T)}{P(t, T)}.$$

In tal modo, si ottiene l'uguaglianza approssimata

$$\mu^f \approx -\rho_{F,f} \sigma^f \sigma_F$$

ove  $\sigma_F$  è la volatilità, assunta costante, del processo  $F_\chi$  e  $\rho_{F,f}$  è la correlazione istantanea tra  $F_\chi$  e  $S^f$ , i.e., tra il tasso di cambio forward e lo swap rate estero.

Essendo note le distribuzioni di  $S$  e  $S^f$  rispetto alla misura  $\mathbb{Q}^C$ , è possibile calcolare il valore atteso (4.23).

Si noti, infatti, che rispetto alla misura  $\mathbb{Q}^C$ , la distribuzione congiunta di

$$\left( \ln \frac{S^f(T)}{S^f(t)}, \ln \frac{S(T)}{S(t)} \right) \text{ condizionata a } \mathcal{F}_t$$

è una normale bivariata, avente media

$$\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} = \begin{bmatrix} (\mu^f - \frac{1}{2}(\sigma^f)^2) \\ -\frac{1}{2}\sigma^2 \end{bmatrix} \tau$$

e matrice di covarianza

$$\begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma^f\sigma & \sigma^2 \end{bmatrix} \tau$$

ove  $\tau = T - t$  e  $\rho$  è la correlazione istantanea tra i forward swap rates  $S^f$  ed  $S$ , i.e.,  $dW(t)dW^f(t) = \rho dt$ . Pertanto, è possibile applicare la seguente decomposizione: si definiscano

$$X := \ln \frac{S^f(T)}{S^f(t)},$$

$$Y := \ln \frac{S(T)}{S(t)}.$$

Allora

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp \left[ -\frac{\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\rho\frac{x-\mu_x}{\sigma_x}\frac{y-\mu_y}{\sigma_y} + \left(\frac{y-\mu_y}{\sigma_y}\right)^2}{2(1-\rho^2)} \right].$$

Per il teorema della probabilità condizionata,

$$f_{X|Y} = \frac{1}{\sigma_y\sqrt{2\pi}\sqrt{1-\rho_i^2}} \exp \left[ -\frac{\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\rho_i\frac{x-\mu_x}{\sigma_x}\frac{y-\mu_y}{\sigma_y} + \left(\frac{y-\mu_y}{\sigma_y}\right)^2}{2(1-\rho_i^2)} \right]$$

$$f_X(x) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp \left[ -\frac{1}{2} \left( \frac{x-\mu_x}{\sigma_x} \right)^2 \right]$$

$$f_Y(y) = \frac{1}{\sigma_y\sqrt{2\pi}} \exp \left[ -\frac{1}{2} \left( \frac{y-\mu_y}{\sigma_y} \right)^2 \right].$$

**QES**( $t$ ) =

$$\begin{aligned} &= C(t) \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (wS^f(t)e^x - wS(t)e^y - wK)^+ f_{X,Y}(x,y) dx dy \\ &= C(t) \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (wS^f(t)e^x - wS(t)e^y - wK)^+ f_{X|Y}(x,y) f_Y(y) dx dy \\ &= C(t) \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} (wS^f(t)e^x - wS(t)e^y - wK)^+ f_{X|Y}(x,y) dx \right] f_Y(y) dy \end{aligned} \tag{4.26}$$

L'integrale interno può essere calcolato esplicitamente, dato che  $wS(t)e^y$  può essere interpretato come una costante nell'integrazione rispetto ad  $x$ , e risulta un integrale "alla Black-Scholes", dato che  $f_{X|Y}(x,y)$ , fissata  $y$ , è la densità di una variabile aleatoria Gaussiana. Più precisamente, si applica la seguente formula, ove  $w \in \{-1, 1\}$  e  $A > 0$ ,  $K > 0$  e  $V > 0$  sono costanti reali:

$$\begin{aligned} &\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}V} (wAe^z - wK)^+ e^{-\frac{1}{2}\frac{(z-M)^2}{V^2}} dz \\ &= wAE^{M+\frac{1}{2}V^2} \Phi \left( w \frac{M - \ln \frac{K}{A} + V^2}{V} \right) - wK \Phi \left( w \frac{M - \ln \frac{K}{A}}{V} \right), \end{aligned}$$

da cui si ottiene

$$\begin{aligned}
\mathbf{QES}(t) &= C(t) \int_{-\infty}^{+\infty} (wS^f(t)e^x - wS(t)e^y - wK)^+ f_{X|Y}(x, y) dx \\
&= C(t) \int_{-\infty}^{+\infty} \left\{ wS^f(t) \exp \left[ \mu_x + \rho\sigma_x \frac{y - \mu_y}{\sigma_y} + \frac{1}{2}\sigma_x^2(1 - \rho^2) \right] \right. \\
&\quad \cdot \Phi \left( \frac{\mu_x + \rho\sigma_x \frac{y - \mu_y}{\sigma_y} - \ln \frac{K+S(t)e^y}{S^f(t)} + \sigma_x^2(1 - \rho^2)}{\sigma_x \sqrt{1 - \rho^2}} \right) \\
&\quad \left. - w(K + S(t)e^y) \Phi \left( \frac{\mu_x + \rho\sigma_x \frac{y - \mu_y}{\sigma_y} - \ln \frac{K+S(t)e^y}{S^f(t)}}{\sigma_x \sqrt{1 - \rho^2}} \right) \right\} f_Y(y) dy.
\end{aligned}$$

É dunque dimostrato il seguente

**Teorema 4.3.1.** *Il prezzo di non arbitraggio della quanto swaption avente payoff (4.23) è*

$$\mathbf{QES} = C(t) \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}v^2} f(v) dv, \quad (4.27)$$

ove

$$\begin{aligned}
f(v) &= wS^f(t) \exp \left\{ \left( \mu^f - \frac{1}{2}\rho^2(\sigma^f)^2 \right) \tau + \rho\sigma^f \sqrt{\tau} v \right\} \\
&\quad \cdot \Phi \left( \frac{\ln \frac{S^f(t)}{h(v)} + [\mu^f + (\frac{1}{2} - \rho^2)(\sigma^f)^2] \tau + \rho\sigma^f \sqrt{\tau} v}{\sigma^f \sqrt{\tau} \sqrt{1 - \rho^2}} \right) \\
&\quad - wh(v) \Phi \left( \frac{\ln \frac{S^f(t)}{h(v)} + [\mu^f - \frac{1}{2}(\sigma^f)^2] \tau + \rho\sigma^f \sqrt{\tau} v}{\sigma^f \sqrt{\tau} \sqrt{1 - \rho^2}} \right)
\end{aligned} \quad (4.28)$$

e

$$h(v) = K + S(t) e^{-\frac{1}{2}\sigma^2\tau + \sigma\sqrt{\tau}v}.$$

La formula (4.28) può essere calcolata mediante una semplice integrazione numerica. Gli unici input di tale formula sono  $S^f(t)$  e  $S(t)$ . Infatti,  $\sigma^f$ ,  $\sigma$  e  $\sigma_F$  possono essere facilmente ottenuti dalle volatilità implicite di mercato e  $\rho$  e  $\rho_{F,f}$  possono essere stimati su base storica.



## Capitolo 5

# Conclusioni

Nel presente lavoro di tesi abbiamo proposto un paradigma per la generazione di scenari, applicato al modello G2++, per la previsione dell'evoluzione temporale dei tassi di cambio tra una valuta domestica ed un insieme di valute estere.

Si è posta l'attenzione su tutti gli aspetti modellistici coinvolti: la generazione di variabili casuali uniformemente distribuite nell'intervallo  $[0, 1)$ , gli algoritmi di trasformazione della distribuzione uniforme in distribuzione Gaussiana al fine di modellare il moto Browniano, e l'implementazione del modello di mercato G2++ Single Currency, utilizzato come unità fondamentale per l'implementazione del modello G2++ Multi Currency.

Per la generazione di variabili aleatorie uniformi è stata effettuata un'implementazione seriale in Java dell'algoritmo "Mersenne Twister for Graphic Processors", validata mediante i test statistici più utilizzati in letteratura. Due possibili sviluppi futuri consistono nella traduzione di tale implementazione in CUDA e nell'introduzione di "DCMT (Dinamic Creation of Mersenne Twisters)", algoritmo per la creazione on-the-fly dei parametri di periodo e tempering per un'istanza di "Mersenne Twister".

Per quanto concerne la trasformazione della distribuzione uniforme in distribuzione Gaussiana si sono analizzati gli algoritmi "Ziggurat" e "Box-Muller", dei quali è stato effettuato un confronto, avente come cifra di merito il costo computazionale, concludendo che "Ziggurat" sia l'algoritmo preferibile per applicazioni di Calcolo Seriale e "Box-Muller" sia più indicato per applicazioni di Calcolo Parallelo.

Avendo replicato con sufficiente accuratezza il comportamento del moto Browniano, è stato possibile implementare in Java il modello G2++ Single Currency, formulato in [6]. Di tale modello sono state implementate le formule di pricing dei principali prodotti finanziari quotati. In particolare, si è discussa ed implementata una formula di pricing alternativa per la Swap-

tion Europea, proposta in [8]. Si è dimostrato formalmente che il modello adottato da Choi e Shin differisce da G2++ per uno shift deterministico. Una questione aperta consiste nella definizione della matrice di covarianza  $\Pi(t)$ , nella notazione di [8], che non coincide con quella utilizzata da Brigo e Mercurio.

Proponiamo, inoltre, due possibili sviluppi futuri, riguardante il modello G2++.

Il primo consiste nel verificare se l'introduzione di due fattori di rischio nel modello migliora il dettaglio previsionale e in quale entità, confrontando i tassi ed i prezzi simulati con quelli di mercato.

Al fine di velocizzare ed ottimizzare la computazione, il secondo sviluppo futuro che proponiamo consiste nel tradurre in CUDA il codice Java implementato per la generazione delle traiettorie dello short-rate  $r(t)$ , imponendo che ogni thread di computazione si occupi di un'intera traiettoria.

Infine, abbiamo evidenziato come il modello G2++ Single Currency possa essere interpretato come unità fondamentale del G2++ Multi Currency, formulato nel caso di 2 valute in [6], per la descrizione dell'evoluzione dei tassi di cambio di  $N - 1$  valute *estere*, rispetto ad una valuta *domestica* fissata.

La metodologia proposta consiste nel calibrare indipendentemente ogni valuta rispetto alle proprie volatilità implicite di Cap/Swaption, per gestire successivamente le interazioni tra la valuta domestica ed ogni altra valuta estera, mantenendo l'impostazione *pairwise* di [6]. Gli input della calibrazione sono la curva degli Zero-Coupon Bond di mercato e la volatilità di Cap/Swaption nella rispettiva valuta che, trattandosi di dati di mercato, incorporano gli effetti delle interazioni tra la valuta considerata e tutte le altre presenti nel mercato.

Pertanto, non risulta necessario esplicitare o quantificare l'entità delle interazioni tra tutte le altre valute, escludendo la valuta domestica, ed è ammissibile considerare il tasso di cambio tra valuta domestica ed ogni altra delle  $N - 1$  valute.

Relativamente all'intero lavoro di tesi proposto, ulteriori sviluppi potrebbero essere rappresentati dall'implementazione di recenti algoritmi per la generazione di numeri quasi-casuali ed altri modelli di tasso, al fine di garantire una maggiore versatilità del generatore di scenari.

## Appendice A

# Algoritmo di Berlekamp-Massey

Dati  $n$  elementi  $s_1, s_2, \dots, s_n$  di un campo finito (o infinito) si cercano  $l$  costanti  $c_1, c_2, \dots, c_l$ , in esso contenute, tali che

$$s_{j+1} = - \sum_{\nu=1}^l c_\nu s_{j+1-\nu}, \quad j = l, \dots, n-1. \quad (\text{A.1})$$

Sia  $L_n(s)$  il più piccolo valore di  $l$ , per il quale esiste un registro a scorrimento, che genera  $s_1, \dots, s_n$ ; si desidera trovare  $c_1, \dots, c_l$ , ove  $l = L_n(s)$ . In altre parole, si può affermare

$$L_{n+1}(s) = \max [L_n(s), n+1 - L_n(s)]. \quad (\text{A.2})$$

Si considerino  $s_1, \dots, s_n$ , generati mediante l'equazione (A.1), con  $L_n(s)$  costanti. Se (A.1) è soddisfatta anche per  $j = n$ , allora  $L_{n+1}(s) = L_n(s)$  e si possono utilizzare le medesime costanti. Se, invece, (A.1) non è soddisfatta per  $j = n$ , allora

$$L_{n+1}(s) = \begin{cases} L_n(s) & \text{se } 2L_n(s) > n \\ n+1 - L_n(s) & \text{se } 2L_n(s) \leq n \end{cases}$$

e le nuove costanti  $L_{n+1}(s)$  devono essere trovate.

Dato il polinomio  $C(x) = 1 + c_1x + \dots + c_lx^l$ , si definisce il seguente algoritmo (Berlekamp-Massey)

1. Inizializzazione:  
 $C(x) \leftarrow B(x) \leftarrow 1,$   
 $j \leftarrow b \leftarrow 1 + l \leftarrow k \leftarrow 0.$
2. Se  $k = n$ , STOP, altrimenti calcolare  
 $d = s_{k+1} + \sum_{i=1}^l c_i s_{k+1-i}.$

3. Se  $d = 0$ , STEP 6; altrimenti STEP 4.
4. Se  $2l > k$ , allora
 
$$C(x) \leftarrow C(x) - bd^{-1}x^j B(x),$$
 STEP 6;  
 altrimenti STEP 5.
5.  $T(x) \leftarrow C(x),$   
 $C(x) \leftarrow C(x) - db^{-1}x^j B(x),$   
 $B(x) \leftarrow T(x),$   
 $l \leftarrow k + 1 - l,$   
 $b \leftarrow d,$   
 $j \leftarrow 0.$
6.  $j \leftarrow j + 1,$   
 $k \leftarrow k + 1,$  STEP 2.

## Appendice B

### Algoritmo [Marsaglia 1964]

Contestualmente all'algoritmo Ziggurat, nella presente appendice si illustra un possibile algoritmo per la generazione di variabili aleatorie normalmente distribuite dalla coda della distribuzione Gaussiana, che rappresenta la principale criticità del metodo Ziggurat.

Il metodo proposto in [12], facendo ricorso unicamente ad operazioni elementari, risulta semplice ed immediato da implementare.

L'idea è di trasformare la coda della distribuzione normale in un intervallo di lunghezza unitaria ed utilizzare il metodo di rifiuto.

Per generare una variabile Normale standard  $X$ , condizionata a  $X > a$ , i.e., avente densità  $Ce^{-\frac{x^2}{2}}$ ,  $x > a$ , si generano due variabili aleatorie uniformemente distribuite  $U_1, U_2$  sull'intervallo  $(0, 1)$ , finché

$$U_2 < a \sqrt{(a^2 - 2\ln U_1)}, \quad (\text{B.1})$$

per poi imporre

$$X = \sqrt{(a^2 - 2\ln U_1)}.$$

La densità di  $U_1$ , data la condizione (B.1), è un multiplo di  $\sqrt{(a^2 - 2\ln u_1)}$ , ove  $0 < u_1 < 1$ . Pertanto, la densità di  $X = \sqrt{(a^2 - 2\ln U_1)}$  è multiplo di  $e^{-\frac{x^2}{2}}$ , con  $a < x < \infty$ .

Nella seguente figura sono riportati i grafici di  $u_2(a) = a \sqrt{(a^2 - 2\ln(u_1))}$  per  $a = 1, 2, 3$ . Per  $a = 3$ , la probabilità dell'evento (B.1) risulta 0.88.

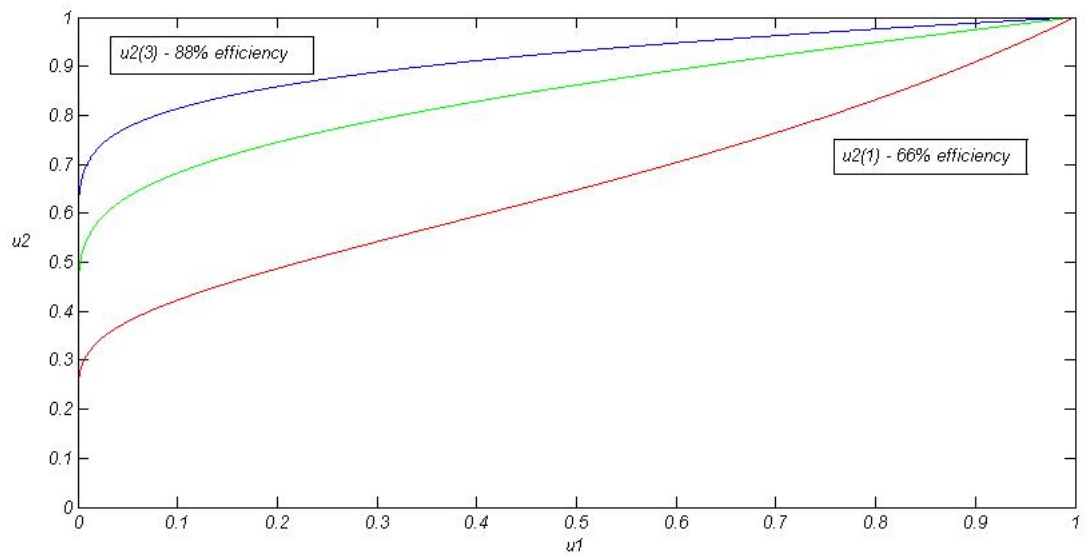


Figura B.1: Se il punto  $(U_1, U_2)$  è scelto in modo uniforme all'interno dell'area sottostante alla curva  $u_2 = a \sqrt{a^2 - 2 \ln(u_1)}$ , allora  $X = \sqrt{a^2 - 2 \ln U_1}$  è una variabile aleatoria Normalmente distribuita e tale che  $X > a$ .

## Appendice C

# Dimostrazione del Teorema 3.3.1

Dalla formula generale di pricing

$$\pi_t = P(t, T) \mathbb{E}^T \{ H_T | \mathcal{F}_t \},$$

ove  $\pi_t$  e  $H_T$  sono, rispettivamente, il prezzo al tempo  $t$  ed il payoff del contratto considerato e  $\mathbb{E}^T$  denota il valore atteso nella T-forward measure. Il prezzo di non arbitraggio della swaption Europea risulta

$$\begin{aligned} & \mathbf{ES}(0, T, \mathcal{T}, N, X, \omega) \\ &= NP(0, T) \mathbb{E}^T \left\{ \left[ \omega \left( 1 - \sum_{i=1}^n c_i P(T, t_i) \right) \right]^+ \right\} \\ &= NP(0, T) \int_{\mathbb{R}^2} \left[ \omega \left( 1 - \sum_{i=1}^n c_i A(T, t_i) e^{-B(a, T, t_i)x - B(b, T, t_i)y} \right) \right]^+ f(x, y) dx dy, \end{aligned}$$

ove  $f$  è la densità del vettore aleatorio  $(x(T), y(T))$ , i.e.,

$$f(x, y) = \frac{\exp \left\{ -\frac{1}{2(1-\rho_{xy}^2)} \left[ \left( \frac{x-\mu_x}{\sigma_x} \right)^2 - 2\rho_{xy} \frac{(x-\mu_x)(y-\mu_y)}{\sigma_x \sigma_y} + \left( \frac{y-\mu_y}{\sigma_y} \right)^2 \right] \right\}}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{xy}^2}}.$$

Fissando  $x$  nell'integrale ed integrando rispetto ad  $y$ , tra  $-\infty$  e  $+\infty$ , si ottiene

$$\int_{\bar{y}(x)}^{+\infty \cdot \omega} \left( 1 - \sum_{i=1}^n \lambda_i e^{-B(b, T, t_i)} \right) \gamma e^{E+F(y-\mu_y)-G(y-\mu_y)^2} dy$$

ove

$$\gamma := \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{xy}^2}}$$

$$\begin{aligned}
E &:= -\frac{1}{2(1-\rho_{xy}^2)} \left( \frac{x-\mu_x}{\sigma_x} \right)^2 \\
F &:= \frac{\rho_{xy}}{1-\rho_{xy}^2} \frac{x-\mu_x}{\sigma_x \sigma_y} \\
G &:= \frac{1}{2(1-\rho_{xy}^2) \sigma_y^2}
\end{aligned}$$

(la dipendenza rispetto ad  $x$  è omessa per semplicità). Utilizzando la formula generale

$$\int_a^b e^{-Ax^2+Bx} dx = \sqrt{\frac{\pi}{A}} e^{\frac{B^2}{4A}} \left[ \Phi\left(b\sqrt{2A} - \frac{B}{\sqrt{2A}}\right) - \Phi\left(a\sqrt{2A} - \frac{B}{\sqrt{2A}}\right) \right]$$

con  $A > 0$ ,  $a$ ,  $b$  e  $B$  costanti reali, l'integrale diviene

$$\begin{aligned}
&\gamma \sqrt{\frac{\pi}{G}} e^{E+\frac{F^2}{4G}} \left[ \Phi\left(+\infty \cdot \omega\sqrt{2G} - \frac{F}{\sqrt{2G}}\right) - \Phi\left((\bar{y} - \mu_y)\sqrt{2G} - \frac{F}{\sqrt{2G}}\right) \right] \\
&- \gamma \sqrt{\frac{\pi}{G}} e^E \sum_{i=1}^n \lambda_i e^{-B(b,T,t_i)\mu_y + \frac{[F-B(b,T,t_i)]^2}{4G}} \cdot \\
&\left[ \Phi\left(+\infty \cdot \omega\sqrt{2G} - \frac{F-B(b,T,t_i)}{\sqrt{2G}}\right) - \Phi\left((\bar{y} - \mu_y)\sqrt{2G} - \frac{F-B(b,T,t_i)}{\sqrt{2G}}\right) \right].
\end{aligned}$$

Effettuando alcune semplificazioni algebriche, si ottiene

$$\begin{aligned}
&\gamma \sqrt{\frac{\pi}{G}} e^{E+\frac{F^2}{4G}} \left\{ \frac{\omega+1}{2} - \Phi\left((\bar{y} - \mu_y)\sqrt{2G} - \frac{F}{\sqrt{2G}}\right) \right. \\
&\quad \left. - \sum_{i=1}^n \lambda_i e^{-B(b,T,t_i)\mu_y + \frac{B(b,T,t_i)[B(b,T,t_i)-2F]}{4G}} \right. \\
&\quad \left. \cdot \left[ \frac{\omega+1}{2} - \Phi\left((\bar{y} - \mu_y)\sqrt{2G} - \frac{F-B(b,T,t_i)}{\sqrt{2G}}\right) \right] \right\}.
\end{aligned}$$

Poiché  $(\omega+1)/2 - \Phi(z) = \omega\Phi(-\omega z)$ , per ogni costante reale  $z$ , si ottiene (3.24) notando che

$$\begin{aligned}
\gamma \sqrt{\frac{\pi}{G}} &= \frac{1}{\sigma_x \sqrt{2\pi}}, \\
E + \frac{F^2}{4G} &= -\frac{1}{2} \left( \frac{x-\mu_x}{\sigma_x} \right)^2, \\
\frac{F}{\sqrt{2G}} &= \frac{\rho_{xy}}{\sqrt{1-\rho_{xy}^2}} \frac{x-\mu_x}{\sigma_x} \\
\sqrt{2G} &= \frac{1}{\sigma_y \sqrt{1-\rho_{xy}^2}}
\end{aligned}$$



## Appendice D

# Codici Matlab

Nella presente appendice sono riportati i codici MATLAB utilizzati per l'implementazione della formula di pricing della Swaption Europea, proposta in [8]. Il frequente ricorso a variabili di tipo *global* è volto ad alleggerire la lettura del codice.

### D.1 Soluzione della dinamica

```
function [x,y] = ou_euler_maruyama()

global a;
global sigma;
global b;
global eta;
global rho;

%% OU_EULER_MARUYAMA applica il metodo di Eulero-Maruyama
% all'EDS di Ornstein-Uhlenbeck.
%
%      dx = theta * ( mu - x(t) ) dt + sigma * dW,
%      x(0) = x0,
%
%      La traiettoria discretizzata del moto Browniano considera
%      passi temporali di lunghezza costante.
%
%      Si utilizza la scala temporale DT_LARGE per la parte deterministica
%      dell'equazione e la scala DT_SMALL per la discretizzazione del
%      moto Browniano, in modo tale che
%
%      dt_large = r * dt_small = tmax / n
%
%      Per un'EDS nella forma:
```

```

%
%      dx = f(x(t)) dt + g(x(t)) dW(t)
%
%      il metodo di Eulero-Maruyama consiste in:
%
%      x(j) = x(j-1) + f(X(j-1)) * dt_large + g(X(j-1)) * dW(j-1)
%
%      ove dW(j-1) rappresenta la somma di R variabili aleatorie
%      Normali standard, moltiplicate per la radice quadrata di DT_SMALL.
%
%      Si noti che, se SIGMA = 0, il problema diviene deterministico,
%      ed ha soluzione
%
%      x(t) = mu + ( x0 - mu ) * exp ( - theta * t )
%
%      Input:
%
%      THETA, MU, SIGMA, reali, parametri del modello.
%
%      X0, reale, condizione iniziale dell'EDS.
%
%      TMAX, reale, istante finale.
%
%      N, intero, numero di passi nella scala DT_LARGE.
%
%      R, intero, numero di passi nella scala DT_SMALL per ogni passo
%      nella scala DT_LARGE
%
%
%
t = 1/360 * [ 0 31 62 95 125 153 184 275 368 738 1109 1479 1849 ...
            2219 2590 2963 3332 3701 4442 5553 7405 9258 11110 ];

dt_small = 0.01;
x = zeros ( 1, size(t,2));

x(1) = 0;
for j = 1 : (size(t,2)-1)
    pil2 = sigma*eta*rho/(a+b) * (1-exp(-(a+b)*(t(j+1)-t(j)))));
    pil1 = sigma^2/(2*a) * (1-exp(-2*a*(t(j+1)-t(j)))));
    mux = (pil1+pil2)/a;

    r = round((t(j+1)-t(j))/dt_small);
    dw = sqrt ( dt_small ) * randn ( 1, r );
    x(j+1) = x(j) + (t(j+1)-t(j)) * a * ( mux - x(j) ) ...
              + sigma * sum ( dw(1:r) );
end

y = zeros ( 1, size(t,2));
y(1) = 0;

```

```

for j = 1 : (size(t,2)-1)
    pi21 = sigma*eta*rho/(a+b) * (1-exp(-(a+b)*(t(j+1)-t(j)))));
    pi22 = eta^2/(2*b) * (1-exp(-2*b*(t(j+1)-t(j)))));
    muy = (pi21+pi22)/b;
    r = round((t(j+1)-t(j))/dt_small);
    dw = sqrt(dt_small) * randn(1, r);
    y(j+1) = y(j) + (t(j+1)-t(j)) * a * (muy - y(j)) ...
        + eta * sum(dw(1:r));
end

z = x+y;

% Grafico della soluzione approssimata.

plot(t, z, 'r-', 'LineWidth', 3)
xlabel('t')
ylabel('X(t)')
title('Euler-Maruyama solution of Ornstein-Uhlenbeck SDE')
grid on

return

end

```

## D.2 Zero Coupon Bond

$$P(t, T) = \frac{P^M(0, T)}{P^M(0, t)} \exp \left\{ -\mathbf{a}(t, T)^T \mathbf{z}(t) - \frac{1}{2} |\mathbf{a}(t, T)|^2 \right\}$$

```

function zcb = P(t,T)

global a;
global sigma;
global b;
global eta;
global rho;

global x;
global y;

global time_grid;

PM = [ 0.999997750 0.999923183 0.999866751 ...
        0.999807547 0.999758346 0.999710843 ...
        0.999666732 0.999487688 0.999190802 ...
        0.996530670 0.990721556 0.980835736 ...
        0.966660552 0.949329199 0.929322614 ...

```

```

0.907336161 0.883558599 0.858792393 ...
0.808161681 0.735699687 0.637426126 ...
0.560557865 0.496748363 ];

if (t==0 & T==0)
    zcb=1.0;

    return;
end

if (t==0)
    zcb= interp1(time_grid,PM,T);

    return;
end

PI = zeros(2,2);
PI(1,1) = sigma^2/(2*a)*(1-exp(-2*a*t));
PI(1,2) = sigma*eta*rho/(a+b)*(1-exp(-(a+b)*t));
PI(2,1) = PI(1,2);
PI(2,2) = eta^2/(2*b)*(1-exp(-2*b*t));

C = chol(PI);

g = [(1-exp(-a*(T-t)))/a ; (1-exp(-b*(T-t)))/b];

aa = C' * g;

stateX = interp1(time_grid,x,t);
stateY = interp1(time_grid,y,t);
state = [stateX ; stateY];

PMT= interp1(time_grid,PM,T);
PMt= interp1(time_grid,PM,t);

z = inv(C)* state;

zcb = PMT/PMt * exp ( -aa' * z - 0.5*(norm(aa))^2);

end

```

### D.3 Swap

$$V(\mathbf{z}) = \frac{1}{P(0, T_e)} \sum_{k=0}^m \text{DCF}_k \exp \left\{ -\mathbf{a}_k^T \mathbf{z} - \frac{1}{2} |\mathbf{a}_k|^2 \right\}$$

```

function V = swap(Te, Tk, coupons, z)

global C;
global sigma;
global eta;
global rho;

global a;
global b;

PI = [ sigma^2/(2*a)*(1-exp(-2*a*Te)) ...
       sigma*eta*rho/(a+b)*(1-exp(-(a+b)*Te)) ; ...
       sigma*eta*rho/(a+b)*(1-exp(-(a+b)*Te)) ...
       eta^2/(2*b)*(1-exp(-2*b*Te)) ];

C = chol(PI);

cF = 0;

for k=1:size(Tk)

    g = [(1-exp(-a*(Tk(k)-Te)))/a ; (1-exp(-b*(Tk(k)-Te)))/b];
    ak = C' * g;

    cF = cF + coupons(k)*P(0, Tk(k)) * exp(-ak'*z - 0.5*(norm(ak))^2);
end

V = 1/P(0, Te) * cashFlows;

end

```

### D.4 Gradiente dello Swap

$$\Delta V(\mathbf{z}) = \frac{1}{P(0, T_e)} \sum_{k=0}^m \text{DCF}_k (-\mathbf{a}_k) \exp \left\{ -\mathbf{a}_k^T \mathbf{z} - \frac{1}{2} |\mathbf{a}_k|^2 \right\}$$

```

function gradV = swapGradient(Te, Tk, coupons, z)

global C;

```

```

global a;
global b;

temp = [0 0]';

for k=1:size(Tk)

    g = [(1-exp(-a*(Tk(k)-Te)))/a ; (1-exp(-b*(Tk(k)-Te)))/b];

    ak = C' * g;

    temp = temp + coupons(k)*P(0,Tk(k)) * ...
            (-ak) * exp(-ak'*z - 0.5*(norm(ak))^2);
end

gradV = 1/P(0,Te) * temp;

end

```

## D.5 Swaption Europea

$$ES = \sum_{k=0}^m DCF_k \mathcal{N}(-(\mathbf{a}_k + \mathbf{z}^*)^T \mathbf{q}_1)$$

```

function C = swaption(Te, Tk, coupons)

global a;
global b;

z = [1;1];

epsilon = 10E-12;

zstar = [0;0];

while( abs(z-zstar) >= epsilon )

    z = zstar;

    zfirst = z - swap(Te, Tk, coupons, z) / ...
                norm(swapGradient(Te, Tk, coupons, z)) * ...
                swapGradient(Te, Tk, coupons, z);

    zstar = swapGradient(Te, Tk, coupons, zfirst)' * zfirst / ...
            norm(swapGradient(Te, Tk, coupons, zfirst)) * ...

```

```

swapGradient(Te, Tk, coupons, zfirst);

end

q1 = swapGradient(Te, Tk, coupons, zstar) / ...
      norm( swapGradient(Te, Tk, coupons, zstar) )^2;

C=0;

for k=1:size(Tk)
    g = [(1-exp(-a*(Tk(k)-Te)))/a ; (1-exp(-b*(Tk(k)-Te)))/b];
    ak = C' * g;
    partial = -(ak + zstar)' * q1;
    C = C + coupons(k)*P(0,Tk(k)) * normcdf(partial,0,1);
end

end

```

# Bibliografia

- [1] T.W.Anderson, D.A.Darling, *A Test of Goodness of Fit*, Journal of the American Statistical Association, Vol.49, No.268, 1954, <http://www.jstor.org/stable/2281537>
- [2] D.Bertsimas, J.Tsitsiklis, *Simulated Annealing*, Statistical Science, (1993), Vol.8, No.1, 10-15, <http://projecteuclid.org/euclid.ss/1177011077>
- [3] I.Beyna, *Interest Rate Derivatives: Valuation, Calibration and Sensitivity Analysis*, Springer 2013
- [4] T.Bjork, *Arbitrage Theory in Continuous Time*, Oxford University Press, New York 2004
- [5] G.E.P. Box, M.E. Muller, *A note on the generation of random normal deviates*, Annals of Mathematical Statistics, Vol.29, No.2, (1958), <http://projecteuclid.org/euclid.aoms/1177706645>
- [6] D.Brigo, F.Mercurio, *Interest Rates Models: Theory and Practice*, Springer 2006
- [7] W.K.Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, Vol.57, No.1, (April 1970), pp.97-109, <http://www.isds.duke.edu/~scs/Courses/Stat376/Papers/Basic/Hastings1970.pdf>
- [8] Choi, Jaehyuk and Shin, SungChan, *Fast Swaption Pricing in Gaussian Term Structure Model* (February 13, 2013), <http://ssrn.com/abstract=2228525>
- [9] P.L'Ecuyer, F.Panneton  $\mathbb{F}_2$ -Linear Random Number Generators, contenuto in *Advancing the Frontiers of Simulation*, Springer, (2009), <http://www.iro.umontreal.ca/~lecuyer/myftp/papers/f2lin.pdf>
- [10] P.L'Ecuyer, R.Simard, *TestU01: A C Library for Empirical Testing of Random Number Generators*, ACM Transactions on Mathematical Software, Vol.33, No.4, Article 22, (August 2007), <http://www.iro.umontreal.ca/~lecuyer/myftp/papers/testu01.pdf>



- [11] P.L'Ecuyer, R.Simard *On the Performance of Birthday Spacings Tests with Certain Families of Random Number Generators*, Gordon and Breach Science Publishers, New York, (1989), [http://www.researchgate.net/publication/222537306\\_On\\_the\\_performance\\_of\\_birthday\\_spacings\\_tests\\_with\\_certain\\_families\\_of\\_random\\_number\\_generators](http://www.researchgate.net/publication/222537306_On_the_performance_of_birthday_spacings_tests_with_certain_families_of_random_number_generators)
- [12] G.Marsaglia, *Generating a Variable from the Tail of a Normal Distribution*, Technometrics, Vol.6, Iss.1, (1964), <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=AD0423993>
- [13] G.Marsaglia, W.W.Tsang, *The Ziggurat Method for Generating Random Variables*, Journal of Statistical Software, Vol. 5, Issue 8, (October 2000), <http://www.jstatsoft.org/v05/i08/paper>
- [14] G.Marsaglia, W.W.Tsang, *Some Difficult-to-pass Tests of Randomness*, Journal of Statistical Software, Vol.7, Issue 3, (January 2002), <http://www.jstatsoft.org/v07/i03/paper>
- [15] M.Matsumoto, T.Nishimura, *Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator.*, ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1. (January 1998), pp. 3-30, <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/ARTICLES/mt.pdf>
- [16] W.H.Press, S.A.Teukolsky, W.T.Vetterling, B.P.Flannery, *Numerical Recipes* - Cambridge University Press (2007)
- [17] M.Saito, M.Matsumoto, *Variants of Mersenne Twister Suitable for Graphic Processors*, ACM Transactions on Mathematical Software 39, 2, Article 12 (February 2013), <http://doi.acm.org/10.1145/2427023.2427029>