

POLITECNICO DI MILANO

Facoltà di Ingegneria dei Sistemi

Corso di Studi in Ingegneria Matematica



Elaborato di Laurea di Secondo Livello

**METODI QUASI MONTE CARLO
PER APPLICAZIONI FINANZIARIE**

Relatore:

Daniele MARAZZINA

Tesi di laurea di
Emma CEREDA
Matr. n. 734451

Anno Accademico 2009-2010

Indice

Introduzione	3
1 Metodi Monte Carlo	6
1.1 La generazione di numeri casuali	6
1.2 Valutazione delle opzioni mediante simulazione	7
1.3 Numero di simulazione e varianza della stima	10
1.3.1 Tecniche per la riduzione della varianza	12
1.4 Simulazione di ϵ	15
2 Metodo quasi-Monte Carlo	17
2.1 Metodi quasi-Monte Carlo	17
2.2 Sequenze low-discrepancy	18
2.2.1 Stima dell'errore	21
2.3 Inversione di Moro	24
2.4 Tipi di sequenze	26
2.4.1 Sequenza di Van der Corput	26
2.4.2 Sequenza di Halton	28
2.4.3 Sequenza di Faure	32
2.4.4 Sequenza di Sobol	34
2.4.5 Confronti	38
3 Pricing di opzioni Europee	41
3.1 Opzioni Europee	41
3.1.1 Skip iniziale	45
3.1.2 Date di monitoraggio maggiori di 1	47
4 Pricing di opzioni Esotiche	52
4.1 Opzioni Asiatiche	52
4.2 Opzioni Barriera	55
4.3 Opzioni Lookback	57

5 Pricing di opzioni Americane	60
5.1 Scelta dei polinomi	61
5.2 Opzioni americane	62
Conclusioni	65
A Codici	67
A.1 Sequenze a bassa discepanza	67
A.1.1 Sequenza di Halton	68
A.1.2 Sequenza di Faure	69
A.1.3 Sequenza di Sobol	70
A.2 Algoritmo di Moro	72
A.3 Pricing di opzioni	73
Bibliografia	76

Introduzione

Molti modelli studiati ed applicati in ambito finanziario non permettono di ricavare soluzioni in forma chiusa e quindi richiedono l'utilizzo di tecniche di tipo numerico per ottenere una soluzione approssimata accettabile. In particolare negli ultimi anni, data anche la crescente complessità computazionale richiesta, si è registrato un aumento dello studio dei metodi numerici. Tra questi il ruolo di grande importanza è ricoperto dai metodi di simulazione Monte Carlo. Il maggior limite che si riscontra nell'approssimazione Monte Carlo è l'alto costo computazionale legato alla lenta velocità di convergenza. Esistono delle strategie che permettono di aumentare la velocità di convergenza di questa tecnica. Queste metodologie modificano il metodo Monte Carlo impiegando come set di ingressi serie deterministiche al posto delle sequenze pseudo-casuali. Questo metodo prende il nome di metodo Quasi Monte Carlo.

I punti della successione vengono scelti in modo deterministico, mirando ad avere una distribuzione il più uniforme possibile sullo spazio indipendentemente dalla sua dimensione e generando ciascun punto il più 'lontano' possibile da tutti i punti che lo precedono. Le sequenze generate in questo modo vengono chiamate a bassa discrepanza.

Lo scopo di questa tesi è capire quale tipo di sequenza, se pseudo-casuale o a bassa discrepanza, conviene utilizzare nel caso in cui si debba prezzare un certo tipo di opzione. L'obiettivo è comprendere per quali casi e per quali dimensioni del problema in esame i metodi Quasi Monte Carlo risultino essere migliori del metodo Monte Carlo.

La tesi è organizzata nel seguente modo:

Capitolo 1 Una breve descrizione del metodo Monte Carlo. In questa prima parte abbiamo analizzato l'impiego di una sequenza pseudo-random nel campo finanziario utilizzata per prezzare alcuni tipi di opzioni. Abbiamo considerato l'analisi della varianza della stima e le relative tecniche di riduzione della varianza.

Capitolo 2 Il metodo Quasi Monte Carlo. Dall'approccio pseudo-casuale del metodo Monte Carlo passiamo all'approccio deterministico del metodo Quasi Monte Carlo tramite le sequenze a bassa discrepanza. Dopo aver studiato le proprietà di queste sequenze in generale, abbiamo approfondito alcune delle sequenze più note sottolineandone le proprietà e i difetti: Halton, Faure e Sobol. Le implementazioni delle sequenze sono in Appendice.

Capitolo 3 In questa parte riportiamo lo studio effettuato sull'utilizzo delle sequenze a bassa discrepanza per il prezzaggio di opzioni semplici, le opzioni Europee. Abbiamo confrontato il metodo Monte Carlo e il metodo Quasi Monte Carlo nel caso in cui invece di simulare solo l'asset a maturity simuliamo anche l'intero cammino del sottostante utilizzando un'approssimazione discreta. In questo modo, variando il numero delle date di monitoraggio abbiamo potuto esaminare l'impatto dell'aumento della dimensione nelle sequenze a bassa discrepanza.

Per valutare la qualità delle simulazioni, si è utilizzato l'errore relativo percentuale:

$$\text{errore} = \left| \frac{\hat{C} - C_{BS}}{C_{BS}} \right| 100$$

dove C_{BS} è il prezzo fornito dalla formula di Black & Scholes e \hat{C} il prezzo trovato tramite simulazione.

Capitolo 4 Il capitolo è incentrato sulle opzioni di tipo Esotico. Abbiamo analizzato la convergenza dei metodi Quasi Monte Carlo nel caso di opzioni Asiatiche, Barriera e Lookback, riscontrando il problema della degradazione delle sequenze nel caso in cui la dimensione sia elevata.

Capitolo 5 I problemi connessi con la valutazione delle opzioni Americane hanno destato e continuano a destare grande attenzione. Le difficoltà si possono sintetizzare nel fatto che l'epoca ottima di esercizio non è determinata. Per il loro prezzaggio abbiamo utilizzato l'algoritmo di Longstaff e Schwartz in cui per simulare il cammino del sottostante si fa uso della simulazione Quasi Monte Carlo. I risultati ottenuti li abbiamo confrontati con il metodo Monte Carlo.

Infine vorrei ringraziare tutte le persone che mi sono state d'aiuto durante l'elaborazione della tesi. La maggior parte dei ringraziamenti vanno al mio relatore Daniele Marazzina che mi ha offerto il suo appoggio e il suo aiuto

costantemente. Un ringraziamento speciale va alla mia sorellina Lidia e a Paolo per la loro pazienza e per il loro aiuto nella stesura della tesi. Ai miei genitori che mi hanno sostenuto durante i miei studi e a mia nonna per le sue continue preghiere. E infine non posso non ringraziare la mia amica Alessandra per essere sempre stata presente.

Milano, Luglio 2010.

Capitolo 1

Metodi Monte Carlo

La simulazione è una procedura che consente non solo di seguire l'evolversi di un processo ma anche di prevedere possibili scenari futuri. Un processo di simulazione consiste nel costruire un modello che imita da vicino il sistema in esame, oppure nel generare dal modello numerosi campioni di casi possibili, studiarne il comportamento al trascorrere del tempo e infine analizzare i risultati evidenziando le decisioni alternative.

Il termine simulazione Monte Carlo fu coniato all'inizio della seconda guerra mondiale da J. von Neumann e da S. Ulam mentre lavoravano al progetto Manhattan presso il centro di ricerche nucleari di Los Alamos. Questi due ricercatori utilizzarono l'espedito di sostituire i parametri delle equazioni che descrivono la dinamica delle esplosioni nucleari con un insieme di numeri casuali. Essi si ispirarono, nella scelta del nome Monte Carlo, all'aleatorietà dei guadagni che caratterizza il suo celebre casinò.

1.1 La generazione di numeri casuali

La distribuzione uniforme in $[0, 1]$ riveste un ruolo fondamentale nella teoria della simulazione Monte Carlo. I metodi di simulazione che verranno trattati in seguito, traggono l'elemento di casualità dalla disponibilità di una successione (finita o infinita) di variabili casuali distribuite uniformemente tra 0 e 1 e tra loro indipendenti.

Una variabile aleatoria U continua è distribuita in modo uniforme in $[0, 1]$ se ha come funzione di densità

$$f_U(u) = \begin{cases} 1 & \text{se } u \in [0, 1] \\ 0 & \text{altrimenti} \end{cases}$$

Per generare un'osservazione casuale c da una distribuzione non uniforme si possono utilizzare varie tecniche, tutte rivolte a trasformare i numeri casuali distribuiti uniformemente tra 0 e 1 in osservazioni estratte dalla distribuzione in esame. Data una v.a. X la sua funzione di distribuzione $F_X(x)$ è, come noto, una funzione crescente da 0 a 1. Se $F_X(x)$ è continua e strettamente crescente ($F_X(x_1) < F_X(x_2)$ se $x_1 < x_2$), allora assume tutti i valori fra 0 e 1 e si può definire una funzione inversa $F_X^{-1}(y)$ tale che

$$y = F_X(x) \iff x = F_X^{-1}(y).$$

Pertanto per ottenere un numero casuale c da una v.a. X basta generare un numero r dalla distribuzione uniforme in $[0, 1]$ e calcolare

$$c = F_X^{-1}(r).$$

In realtà nessuna sequenza generata attraverso un algoritmo e quindi attraverso un computer può veramente definirsi casuale. Pertanto potremmo limitarci a dire che possiamo generare solo sequenze di numeri *pseudo-casuali*, ovvero numeri che sembrino apparentemente casuali. Una sequenza pseudo-casuale è costituita da una successione di valori x_j , nel quale i primi k valori x_0, x_1, \dots, x_{k-1} sono scelti in modo da essere difficilmente prevedibili, e ciascuno dei successivi x_j , $j \geq k$, è ottenuto applicando una funzione F sui k precedenti valori $x_{j-1}, x_{j-2}, \dots, x_{j-k}$. La sequenza x_0, x_1, \dots, x_{k-1} dei primi k valori prende il nome di *seme* della sequenza. Naturalmente l'intera sequenza dei numeri generati dipende dal valore del seme.

In genere per la generazione di sequenze di numeri casuali ci si affida ad un generatore congruenziale lineare della forma:

$$x_{j+1} = (a \cdot x_j + b) \bmod m$$

dove a è detto moltiplicatore, b traslazione e m modulo.

1.2 Valutazione delle opzioni mediante simulazione

L'applicazione delle tecniche di simulazione Monte Carlo alla valutazione delle opzioni finanziarie è stata proposta per la prima volta in [Boyle (1977)]. Di fondamentale importanza è la procedura utilizzata per generare le traiettorie del prezzo S del bene sottostante, $\{S_t\}_{t \in [0, T]}$. Le traiettorie vengono costruite in base all'ipotesi di poter operare in un'economia neutrale al rischio. La validità di tale ipotesi è garantita, ad esempio nel caso di un'opzione *call*

europea, dalla possibilità di costruire un portafoglio di copertura che replica perfettamente il *payoff* dell'opzione. Se è lecito supporre che gli investitori siano neutrali al rischio, i calcoli connessi con la valutazione dei titoli derivati si semplificano enormemente in quanto il tasso di rendimento atteso di ogni titolo coincide con il tasso di interesse per le attività non rischiose (r tasso risk-free). Inoltre, in un'economia neutrale al rischio il valore atteso all'epoca $t < T$ di un flusso di cassa aleatorio disponibile all'epoca T (time to maturity) si può ottenere semplicemente attualizzando il valore medio del flusso di cassa al tasso di interesse privo di rischio

$$price = e^{-r(T-t)} E[f(\{S_t\})]$$

dove il valore atteso $E[\cdot]$ è sotto la misura neutrale al rischio e $f(\cdot)$ è il cash flow del derivato in T che può dipendere dall'intera storia del sottostante. Nel caso discreto, la scadenza T viene divisa in s intervalli di tempo di lunghezza dt ($n = 1, 2, \dots, s = T/dt$) e la traiettoria discretizzata del prezzo del sottostante tra 0 e T viene indicata come S_0, \dots, S_T .

Nel caso particolare di un'opzione call europea abbiamo che il valore C_t in corrispondenza dell'istante temporale t ($0 \leq t \leq T$), in ipotesi di neutralità al rischio, è dato da

$$C_t = e^{-r(T-t)} E[C_T] \quad (1.1)$$

dove

$$C_T = \max[0, S_T - X] \quad (1.2)$$

è il payoff dell'opzione con X il prezzo d'esercizio.

Per simulare l'andamento del prezzo del sottostante assumiamo che l'asset non paghi i dividendi e segua un processo Browniano geometrico. L'ennesimo prezzo $S(t_n)$ segue ricorsivamente la formula:

$$S(t_n) = S(t_{n-1}) e^{(r - \frac{\sigma^2}{2})dt + \sigma \epsilon_n \sqrt{dt}} \quad (1.3)$$

dove r è il tasso risk-free; σ è la volatilità annua del prezzo del sottostante; per ogni intervallo di tempo ϵ_n è un numero casuale estratto da una distribuzione normale standardizzata.

La Figura 1.1 illustra tre simulazioni di andamento del sottostante, ognuna con dieci intervalli di tempo o date di monitoraggio, per un asset con $r = 10\%$, $\sigma = 30\%$ e $T = 1$.

La relazione (1.3) traduce una proprietà del moto Browniano geometrico che afferma che la v.a. $\log S(t_n)$ si distribuisce in modo normale con media $\log S(t_{n-1}) + (r - \frac{\sigma^2}{2})dt$ e deviazione standard $\sigma \sqrt{dt}$.

Dopo aver simulato la traiettoria del prezzo del sottostante si calcola il payoff

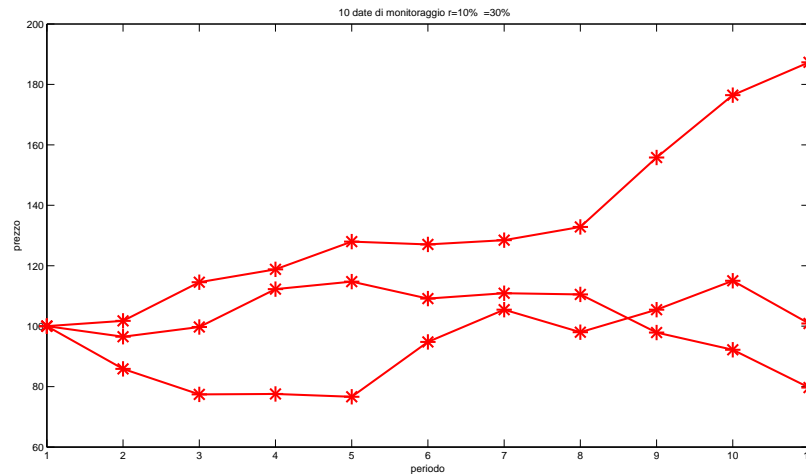


Figura 1.1: Sottostante con 10 date di monitoraggio $r = 10\%$ $\sigma = 30\%$

dell'opzione mediante la formula (1.2).

Si può generare un nuovo percorso k per il prezzo del sottostante $S_{k,t}$, calcolare il corrispondente payoff per l'opzione e procedere con tale tecnica in modo da disporre di un numero N elevato di payoff, la cui media aritmetica può essere utilizzata per stimare il valore medio $E(C_T)$ dell'opzione call all'epoca della scadenza.

$$E(C_T) = \frac{1}{N} \sum_{k=1}^N \max[0, S_{k,T} - X].$$

il valore stimato dell'opzione \widehat{C}_t al tempo $t = 0$ è calcolato attualizzando $E(C_T)$ al tasso privo di rischio

$$\widehat{C}_0 = e^{-rT} E(C_T).$$

La simulazione Monte Carlo è particolarmente utile per prezzare le opzioni esotiche; per alcune opzioni di questo tipo infatti non esiste una formula analitica di valutazione. Si tratta semplicemente di adeguare la funzione di payoff al particolare tipo di opzione in esame. Ad esempio nel caso di un'opzione call asiatica di tipo average price con parametro di riferimento la media aritmetica discreta calcolata in base agli n valori $S_k(t_1), S_k(t_2), \dots, S_k(t_n)$, il payoff della k -esima simulazione è

$$\max\left[0, \frac{1}{n} \sum_{i=1}^n S_k(t_i) - X\right].$$

Si osservi che se si deve prezzare un'opzione asiatica o più in generale un'opzione path dependent bisogna simulare tutti i prezzi del sottostante dall'istante iniziale fino all'istante finale; se invece il payoff dell'opzione non dipende dalla traiettoria percorsa dal prezzo del sottostante la procedura di simulazione diviene molto più semplice e rapida in quanto le N relazioni del tipo (1.3) si possono sostituire con un'unica relazione.

In generale possiamo riassumere la valutazione delle opzioni mediante simulazione tramite i seguenti passi:

- generare un campione di cammini per il sottostante
- calcolare il payoff per ogni percorso
- calcolare la media dei payoff campionati per ottenere una stima del valore atteso
- scontare il payoff atteso al tasso privo di rischio per ottenere una stima del valore del titolo.

1.3 Numero di simulazione e varianza della stima

La tecnica di simulazione Monte Carlo applicata alla valutazione delle opzioni finanziarie prevede di generare un insieme di numeri casuali che consentono di costruire N traiettorie per il prezzo del sottostante. Sulla base di tali traiettorie si calcolano i payoff attualizzati z_i ; la media aritmetica campionaria di tali payoff attualizzati

$$\bar{z} = \frac{1}{N} \sum_{i=1}^N z_i$$

rappresenta la stima del valore dell'opzione ottenuta mediante simulazione. Indicato con C_0 il valore reale dell'opzione in esame è logico aspettarsi che all'aumentare del numero N delle traiettorie la media aritmetica campionaria dia una stima via via migliore di C_0 .

Per analizzare il grado di affidabilità di tale stima, si prefissa uno scostamento

assoluto massimo e_{max} fra il valore stimato \bar{z} e il valore vero dell'opzione C_0 e un livello di confidenza α tale che

$$Prob(|\bar{z} - C_0| \leq e_{max}) \geq 1 - \alpha. \quad (1.4)$$

In base al teorema centrale del limite possiamo affermare che la media aritmetica campionaria \bar{z} , al divergere del numero delle traiettorie N , tende a una distribuzione normale con media C_0 e varianza $\sigma_{\bar{z}}^2 = \sigma^2/N$, dove con σ^2 si è indicata la varianza della v.a. payoff attualizzato.

Si osservi che la varianza della media campionaria tende ad annullarsi al divergere del numero delle traiettorie N ; in altri termini le stime ottenute calcolando la media campionaria \bar{z} convergono al valore cercato C_0 . Si tratta, pertanto, di trovare il numero minimo di traiettorie N che garantisce il verificarsi della relazione (1.4). A tale scopo si osservi che la v.a.

$$Z_N = \frac{(\bar{z} - C_0)\sqrt{N}}{\sigma} = \frac{(\bar{z} - C_0)}{\sigma_{\bar{z}}} \quad (1.5)$$

si distribuisce asintoticamente secondo una legge di probabilità normale standardizzata e quindi per ogni x vale:

$$\lim_{N \rightarrow \infty} Prob(Z_N \leq x) = \Phi(x) \quad (1.6)$$

dove $\Phi(x)$ è la distribuzione normale cumulata. Lo stesso limite vale se σ è sostituito dalla deviazione standard campionaria s_C ; questo perché σ in pratica molto spesso non è conosciuto, mentre s_C si calcola facilmente dalle simulazioni. Per il teorema centrale del limite l'errore nelle simulazioni segue la distribuzione:

$$\bar{z} - C_0 \approx N(0, \sigma_{\bar{z}}^2)$$

e per il vero valore C_0 abbiamo il seguente intervallo di confidenza:

$$\bar{z} \pm z_{\alpha/2} \frac{s_C}{\sqrt{N}}$$

dove al tendere di N all'infinito, la probabilità che questo intervallo contenga il valore esatto C_0 si avvicina a $1 - \alpha$. L'errore assoluto nella stima di C_0 è proporzionale quindi a σ/\sqrt{N} . Per esempio, per ridurre l'errore di un fattore 10, il numero di simulazioni che bisogna effettuare deve crescere di un fattore 100. Per questo motivo la velocità di convergenza dei metodi Monte Carlo viene considerata lenta.

1.3.1 Tecniche per la riduzione della varianza

La valutazione delle opzioni finanziarie mediante tecniche di simulazione se da un lato presenta il vantaggio di fornire delle stime caratterizzate da un errore indipendente dalla dimensione del problema (dal numero di tappe che si devono considerare per la costruzione delle singole traiettorie) e ben delineato da un punto di vista statistico, dall'altro lato richiede spesso di considerare un numero molto elevato di traiettorie e quindi consente di ottenere delle stime sufficientemente accurate solo a fronte di un considerevole dispendio in termini di mole e tempi di calcolo. Dato che nelle tecniche di simulazione l'errore di stima risulta essere proporzionale alla varianza delle stime stesse, per ridurre l'errore bisogna ridurre la varianza. Le tecniche di riduzione della varianza che consideriamo sono la tecnica della variabile antitetica, il metodo della variabile di controllo e l'importance sampling. Per i dettagli si veda [Glasserman (2004)].

Il primo consiste nel considerare delle osservazioni casuali complementari. Si calcola una prima stima \hat{z}_1 con la procedura di simulazione tradizionale e una seconda stima \hat{z}_2 ottenuta cambiando il segno a tutti i numeri casuali generati (cioè se ϵ_k è un numero casuale generato dalla distribuzione normale standardizzata, allora $-\epsilon_k$ è il corrispondente numero casuale utilizzato per calcolare \hat{z}_2). La media aritmetica fra le due stime \hat{z}_1 e \hat{z}_2 viene utilizzata come stima finale del valore dell'opzione.

Se gli stimatori Z_1 e Z_2 di C_0 sono corretti anche lo stimatore

$$Z_{av} = \frac{Z_1 + Z_2}{2}$$

è corretto. Infatti, considerando che

$$E[Z_1] = E[Z_2] = C_0 \quad \text{e} \quad Var(Z_1) = Var(Z_2)$$

$$E[Z_{av}] = \frac{E[Z_1] + E[Z_2]}{2} = C_0$$

e la sua varianza è

$$Var(Z_{av}) = \frac{1}{4}Var(Z_1) + \frac{1}{4}Var(Z_2) + \frac{1}{2}Cov(Z_1, Z_2) = \frac{1}{2}Var(Z_1) + \frac{1}{2}Cov(Z_1, Z_2)$$

La tecnica delle osservazioni complementari risulta efficiente se

$$Var(Z_{av}) \leq \frac{1}{2}Var(Z_1)$$

che equivale a richiedere che

$$Cov(Z_1, Z_2) \leq 0. \quad (1.7)$$

Definiamo una funzione monotona ϕ tale che $Z_i = \phi(\epsilon_i)$ con $\epsilon_1, \epsilon_1, \dots, \epsilon_N$ variabili i.i.d $N(0, 1)$. La condizione (1.7) è soddisfatta se prendiamo $Z_1 = \phi(\epsilon)$ e $Z_2 = \phi(-\epsilon)$, con $-\epsilon_1, -\epsilon_1, \dots, -\epsilon_N$ ancora variabili i.i.d $N(0, 1)$. Infatti:

$$E[\phi(\epsilon)\phi(-\epsilon)] \leq E[\phi(\epsilon)]E[\phi(-\epsilon)]$$

quindi

$$Cov(Z_1, Z_2) = E[\phi(\epsilon)\phi(-\epsilon)] - E[\phi(\epsilon)]E[\phi(-\epsilon)] \leq 0.$$

Nel caso non si debba lavorare con delle variabili normali ma con delle variabili uniformi U_i , la variabile $1 - U_i$ ha la stessa distribuzione di U_i e la coppia $(U_i, 1 - U_i)$ costituisce una coppia di variabili antitetiche in quanto le due sono correlate negativamente.

La seconda tecnica consiste nel sostituire il calcolo di un valore medio incognito con il calcolo della differenza tra il valore medio incognito e un altro valore medio il cui valore è noto. Sia P_A il prezzo di un'opzione A di tipo asiatico con parametro di riferimento la media aritmetica. Per tale opzione non è disponibile una formula analitica di valutazione. Sia, inoltre, G un'opzione ancora di tipo asiatico che differisce dall'opzione A per il solo fatto che il suo payoff dipende dalla media geometrica invece che dalla media aritmetica dei prezzi del bene sottostante. Per l'opzione G è disponibile una formula analitica che fornisce il prezzo P_G . Utilizzando la procedura della variabile di controllo si riesce a sfruttare la conoscenza del prezzo noto P_G per calcolare il prezzo incognito P_A . Infatti, indicati con \hat{P}_A e \hat{P}_G i payoff attualizzati delle due opzioni, si ha

$$P_A = E[\hat{P}_A] \quad P_G = E[\hat{P}_G]$$

e quindi il prezzo P_A si può esprimere come la somma tra il prezzo noto P_G e il valor medio della differenza fra \hat{P}_A e \hat{P}_G , cioè

$$P_A = P_G + E[\hat{P}_A - \hat{P}_G]$$

Uno stimatore non distorto del prezzo P_A è pertanto

$$\hat{P}_A^{cv} = \hat{P}_A + \alpha(P_G - \hat{P}_G)$$

Questa relazione evidenzia che lo stimatore \hat{P}_A^{cv} corregge lo stimatore \hat{P}_A in base alla differenza fra il valore noto P_G e il prezzo stimato \hat{P}_G . La differenza $(P_G - \hat{P}_G)$ funge da controllo per la stima di P_A . Dato che

$$Var(\hat{P}_A^{cv}) = Var(\hat{P}_A) + \alpha^2 Var(\hat{P}_G) - 2\alpha Cov(\hat{P}_A, \hat{P}_G)$$

il metodo della variabile di controllo risulta efficiente se la covarianza fra \hat{P}_A e \hat{P}_G è elevata. Il valore di α che minimizza la varianza è

$$\alpha = -\frac{Cov(\hat{P}_A, \hat{P}_G)}{Var(\hat{P}_G)}.$$

Un'altra tecnica per la riduzione della varianza della stima è la procedura nota come *importance sampling*.

L'idea alla base è che certi valori delle variabili casuali in una simulazione hanno un impatto maggiore sul parametro che si stima rispetto ad altri. Se questi 'importanti' valori vengono enfatizzati da una campionatura più frequente, scegliendo una diversa distribuzione, la varianza dello stimatore può essere ridotta. Questo uso di distribuzioni si tradurrà in uno stimatore distorto se applicato direttamente alla simulazione. Per rendere lo stimatore non distorto, gli output della simulazione vengono ponderati con un peso dato dal rapporto di verosimiglianza. In particolare nel metodo Monte Carlo si vuole esprimere il valore atteso sotto una certa misura di probabilità come valore atteso rispetto ad un'altra misura di probabilità.

Consideriamo per esempio la valutazione di un'opzione call tramite (1.1). Un approccio diretto genera campioni del valore finale del sottostante S_T consistenti con un moto browniano geometrico con drift r e volatilità σ .

Se volessimo generare campioni di S_T come moti browniani geometrici con drift μ , dobbiamo pesare il relativo valore atteso con il rapporto di verosimiglianza L

$$L = \left(\frac{S_T}{S_0}\right)^{\frac{r-\mu}{\sigma^2}} \exp\left(\frac{(r^2 - \mu^2)T}{2\sigma^2}\right)$$

ottenendo quindi

$$E_r[\max(S_T - K, 0)] = E_\mu[\max(S_T - K, 0)L].$$

Idealmente, si vorrebbe scegliere la distribuzione in modo da ridurre la varianza. Nel precedente esempio, si ottiene uno stimatore di varianza nulla campionando S_T dalla densità

$$f(x) = c^{-1} \max(x - K, 0) e^{-rT} g(x)$$

dove g è la (log-normale) densità di S_T e c è una costante di normalizzazione.

1.4 Simulazione di ϵ

La chiave delle simulazioni Monte Carlo è la generazione di ϵ , variabile aleatoria distribuita come una $N(0, 1)$. In pratica, gli ϵ sono calcolati tramite un algoritmo che trasforma numeri uniformi su $(0, 1)$ in variabili normali. Per ogni intervallo di tempo la distribuzione degli ϵ simulati lungo i cammini del sottostante deve approssimare una normale $N(0, 1)$; parallelamente i numeri random nell'intervallo $(0, 1)$ devono essere distribuiti uniformemente su $(0, 1)$.

In Figura 1.2 sono riportati 1000 numeri random uniformi. L'asse orizz-

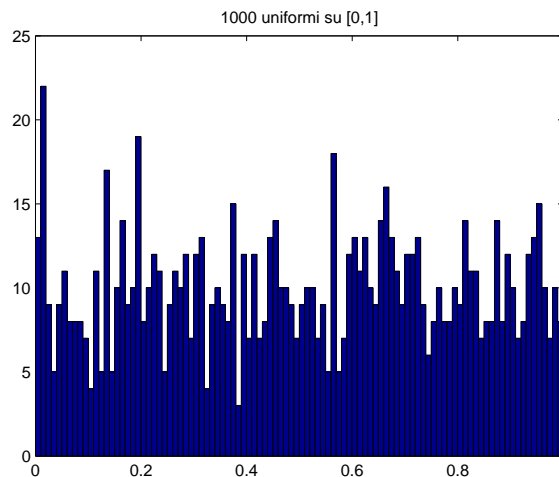


Figura 1.2

zontale è diviso in 100 intervalli di lunghezza 0.01. Per avere una copertura uniforme ci aspetteremmo di vedere 10 punti per ogni intervallo. Questo non è il caso; il secondo intervallo ad esempio contiene più di 20 punti e ce ne sono altri che ne contengono meno di 5. Il fatto che un intervallo contenga tanti punti indica che ci sono dei punti che sono clusterizzati. All'aumentare del numero dei punti, tuttavia, la distribuzione diventa più uniforme.

Per avere una simulazione accurata di $N(0, 1)$ viene richiesto un grande numero di cammini. Se il numero non è abbastanza grande, la distribuzione degli ϵ può non essere approssimabile a una $N(0, 1)$; cioè gli ϵ possono formare dei cluster e non coprire uniformemente il dominio.

Non solo vogliamo una copertura uniforme su tutto $(0, 1)$ ma anche su tutti gli intervalli nel quale il dominio viene suddiviso.

La Figura 1.3 mostra 1000 punti bidimensionali uniformemente distribuiti su $(0, 1)$. La formazione di cluster crea aree prive di punti.

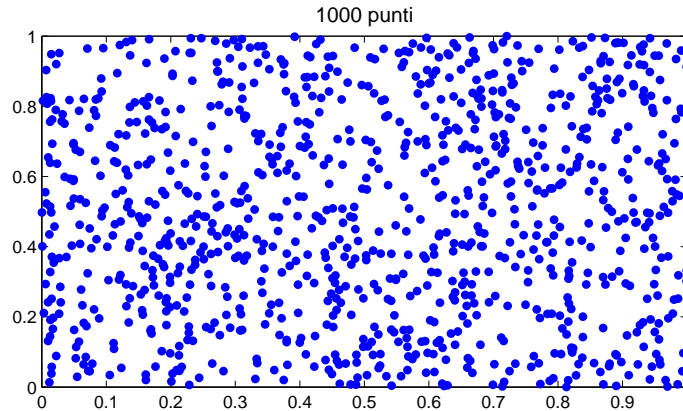


Figura 1.3

In generale più i punti sono distribuiti uniformemente sul dominio, più accurata sarà la simulazione. Questo è l'approccio delle sequenze chiamate *low-discrepancy*. I punti sono scelti in modo deterministico al fine di minimizzare la presenza di cluster e aumentare l'accuratezza. Nonostante questo, il metodo Monte Carlo è flessibile e semplice da implementare e modificare. Inoltre, la creazione di computer sempre più potenti ha favorito la crescita e lo sviluppo di questo metodo. Uno svantaggio di questo metodo che è per problemi molto complessi è richiesto un grande numero di simulazioni per ottenere dei risultati precisi. Le tecniche di controllo della varianza e il metodo della variabile antitetica sono state usate per aumentare la precisione. Nel capitolo seguente mostreremo l'altra tecnica che si sta diffondendo molto rapidamente: utilizzare delle sequenze deterministiche al posto di sequenze pseudo-casuali.

Capitolo 2

Metodo quasi-Monte Carlo

Come già detto nel precedente capitolo, la simulazione Monte Carlo, che utilizza sequenze pseudo-casuali, genera degli errori dell'ordine di $1/\sqrt{M}$, dove M è il numero delle simulazioni. Questi errori sono indipendenti dalla dimensione del problema. Le sequenze low-discrepancy, d'altra parte, generano un upper bound dell'errore dell'ordine di $(\log M)^s/M$, dove s è il numero delle dimensioni.

Tre sequenze a bassa discrepanza per generare numeri uniformemente distribuiti sull'intervallo $(0, 1)$ sono le sequenze di Halton, Faure e Sobol.

2.1 Metodi quasi-Monte Carlo

Il termine procedura quasi-Monte Carlo compare per la prima volta in un rapporto tecnico del centro di ricerche nucleari di Los Alamos scritto da R. D. Richtmeyer nel 1951. Inizialmente solo pochi specialisti hanno utilizzato i metodi quasi-Monte Carlo ma negli ultimi anni l'interesse per tali procedure si è sensibilmente ampliato: in molte applicazioni numeriche, infatti, esse si sono dimostrate più accurate dei metodi Monte Carlo standard.

Un metodo quasi-Monte Carlo si può definire in modo sintetico come la versione deterministica del metodo di simulazione Monte Carlo tradizionale. Il determinismo riguarda la costruzione delle traiettorie, che avviene sulla base di osservazioni generate con tecniche deterministiche, e non aleatorie.

Nella valutazione delle opzioni finanziarie, la dimensione del problema coincide con il numero delle tappe lungo ciascuna traiettoria del sottostante.

L'idea fondamentale dei metodi quasi-Monte Carlo è molto semplice: i punti della successione vengono scelti in modo deterministico, mirando ad avere una distribuzione il più uniforme possibile sullo spazio indipendentemente

dalla sua dimensione e generando ciascun punto il più ‘lontano’ possibile da tutti i punti che lo precedono.

In questo caso l’aggiunta di nuovi punti non comporta il cambiamento della posizione dei punti scelti in precedenza come avviene invece per il metodo a griglia con punti equispaziati (Figura 2.1). Quando viene aggiunto il quarto punto, i punti precedenti vengono spostati.

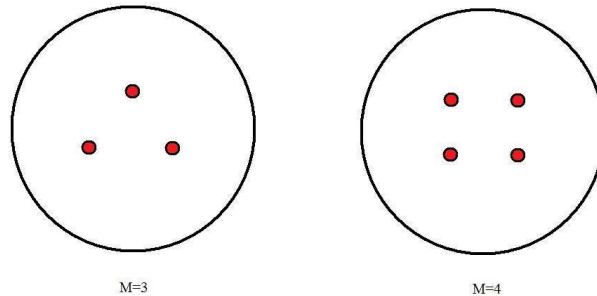


Figura 2.1: *metodo a griglia con punti equispaziati*

2.2 Sequenze low-discrepancy

Le successioni di tipo quasi-Monte Carlo sono dette anche successioni a bassa discrepanza. L’accuratezza di questo approccio dipende dal modo in cui i punti deterministici sono uniformemente dispersi nel dominio. La *discrepanza* è collegata a una misura della distanza media tra i punti: più i punti sono dispersi uniformemente, più bassa sarà la discrepanza.

L’idea basilare di questo approccio è facilmente spiegabile nel caso monodimensionale. Supponiamo di voler integrare la funzione $f(x)$ sull’intervallo $[0, 1]$ utilizzando una sequenza di M punti. Al posto di utilizzare una sequenza random, utilizziamo una sequenza deterministica di punti che sono uniformemente distribuiti. Con questa scelta, l’accuratezza della stima sarà maggiore di quella ottenuta utilizzando l’approccio Monte Carlo. Se usiamo una griglia equispaziata otteniamo il metodo dei trapezi che ha un errore di $O(M^2)$. Se vogliamo calcolare integrali multidimensionali, senza perdita di generalità, possiamo assumere che il dominio di integrazione sia contenuto in un ipercubo unitario s -dimensionale. I vantaggi derivanti dall’utilizzo di una griglia equispaziata in una dimensione non vengono mantenuti nel caso di

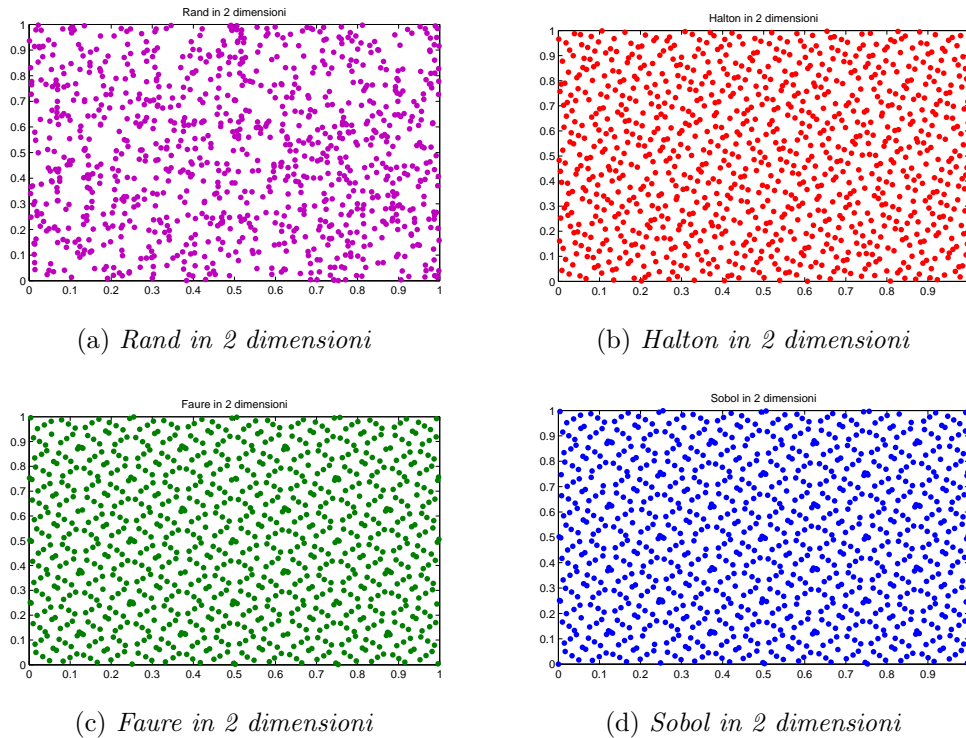


Figura 2.2

più dimensioni. La ragione principale è che il bound dell'errore per il metodo trapezoidale s -dimensionale è $O(M^{2/s})$. Inoltre, se usiamo una griglia cartesiana, dobbiamo decidere prima il numero di punti da utilizzare per coprire uniformemente tutto il dominio. Ciò è restrittivo in quanto, nelle applicazioni numeriche, si vuole essere in grado di aggiungere punti sequenzialmente fino a quando non si incontri un determinato criterio di arresto.

Le sequenze low-discrepancy hanno la proprietà che, una volta aggiunti i punti successivi, l'intera sequenza rimane uniformemente distribuita su tutta la regione.

Dato l'insieme di M punti x_1, x_2, \dots, x_M contenuto nell'ipercubo unitario $I^s = [0, 1]^s$ e un insieme $E \subset [0, 1]^s$ si consideri la quantità

$$D(E, M) = \frac{A(E, M)}{M} - V(E) = \left| \frac{\#\text{punti} \in E}{\#\text{punti}} - \text{volume}E \right|$$

dove $A(E, M)$ è il numero di punti k ($1 \leq k \leq M$) con $x_k \in E$ e $V(E)$ è il volume di E .

Se calcoliamo $D(E, M)$, con E fissato, nelle figure (2.3a) e (2.3b) otteniamo per la prima $\frac{5}{13} - 0.6 \times 0.5 = 0.0846$ e per la seconda $\frac{10}{13} - 0.8 \times 0.9 = 0.0492$.

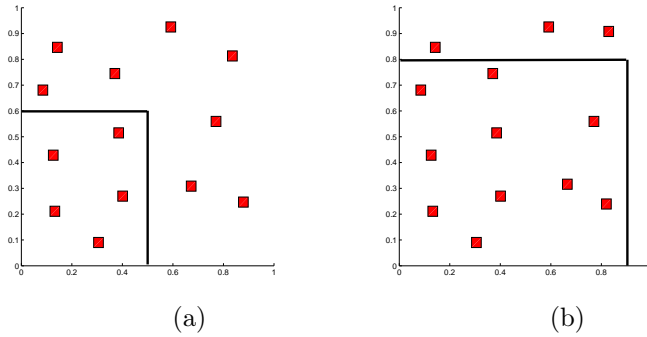


Figura 2.3: 13 punti su $[0,1]$

La discrepanza dell'insieme di punti x_1, x_2, \dots, x_M è definita come il *sup* di $|D(E, M)|$ su tutti gli insiemi E .

Un'utile variante della nozione di discrepanza di un insieme di M punti è la discrepanza $D^*(E, M)$ ottenuta calcolando il *sup* su tutti gli insiemi E del tipo

$$\prod_{i=1}^s [0, u_i] \quad \text{con} \quad u_i \in [0, 1].$$

Intuitivamente la D_M^* indica quanto uniformemente i punti x_1, \dots, x_M sono distribuiti in I^s .

Per calcolare l'errore di stima che si commette quando si utilizzano le sequenze a bassa discrepanza, bisogna utilizzare il seguente teorema:

Teorema Koksma-Hlawka Sia $I^s = [0, 1]^s$ e f una funzione a variazione limitata $V(f)$ su I^s nel senso di Hardy e Krause. Allora per ogni $x_1, \dots, x_M \in I^s$ si ha

$$\left| \frac{1}{M} \sum_{n=1}^M f(x_n) - \int_{I^s} f(u) du \right| \leq V(f) D_M^*(x_1, \dots, x_M)$$

Il bound dell'errore fornito da questo teorema però è poco d'aiuto nella maggior parte delle situazioni pratiche. Nelle applicazioni finanziarie infatti i payoff sono normalmente delle funzioni continue (tranne alcune importanti eccezioni come le opzioni barriera) ma non sono sufficientemente smooth da

avere una variazione finita.

Nel caso monodimensionale esiste una forma esplicita molto semplice per rappresentare la discrepanza di un insieme di punti x_1, x_2, \dots, x_M : se tali punti sono ordinati in modo non decrescente, cioè se $x_1 \leq x_2 \leq \dots \leq x_M$, allora si dimostra che vale la relazione

$$D_M^* = \frac{1}{2M} + \max_{1 \leq m \leq M} \left| x_m - \frac{2m-1}{2M} \right|.$$

Da tale equazione si ha $D^* \geq \frac{1}{2M}$ e il minimo viene raggiunto se i punti vengono scelti come segue:

$$x_m = \frac{2m-1}{2M} \quad m = 1, 2, \dots, M.$$

In dimensioni elevate non c'è una forma semplice per la discrepanza delle sequenze.

Ci sono molti esempi di sequenze low-discrepancy; le più note sono quelle di Halton (1960), Sobol (1967) e Faure (1982). Per queste sequenze la formula asintotica della discrepanza è

$$D_M^* = O\left(\frac{(\log M)^s}{M}\right)$$

Questo limite per la discrepanza implica una costante che dipende generalmente dalla dimensione s della sequenza. Queste costanti sono molto difficili da stimare accuratamente per problemi di grandi dimensioni.

2.2.1 Stima dell'errore

Supponiamo di voler integrare una funzione $f \in L^2(\Omega_s)$ e calcolare:

$$I_{s,\rho}f = \int_{\Omega_s} f(x)\rho(x)dx$$

dove $\Omega_s \subseteq \mathbb{R}^s$ e ρ è una densità di probabilità (nel nostro caso considereremo $\Omega_s = [0, 1]^s$ e $\rho(x) = 1$):

$$\rho(x) \geq 0, \text{ and } \int_{\Omega_s} \rho(x)dx = 1$$

Se utilizziamo il metodo Monte Carlo, l'integrale viene approssimato:

$$I_{s,\rho}f \approx M_{M,s}f = \frac{1}{M} \sum_{k=1}^M f(t_k)$$

dove t_1, \dots, t_M sono dei campioni indipendenti di Ω_s con densità ρ .

Sia $\sigma(f)^2 = I_{s,\rho}((f - I_{s,\rho}f)^2) = I_{s,\rho}(f^2) - (I_{s,\rho}f)^2$

l'errore quadratico medio, r.m.s, è definito come

$$r.m.s = \frac{\sigma(f)}{\sqrt{M}} \quad (2.1)$$

approssimabile nel caso MC con

$$\sqrt{\frac{1}{M(M-1)} \sum_{k=1}^M (f(t_k) - M_{M,s}f)^2} \quad (2.2)$$

Il metodo MC fornisce uno stimatore non distorto dell'errore, infatti:

$$\begin{aligned} E[M_{M,s}f] &= \int_{\Omega_s} \dots \int_{\Omega_s} \left(\frac{1}{M} \sum_{k=1}^M f(t_k) \right) \rho(t_1) \dots \rho(t_M) dt_1 \dots dt_M \\ &= \frac{1}{M} \sum_{k=1}^M \int_{\Omega_s} f(t_k) \rho(t_k) dt_k \\ &= I_{s,\rho}f \end{aligned} \quad (2.3)$$

Il quadrato dell'errore *r.m.s* sarà dato dalla varianza di $M_{M,s}f$, ovvero

$$\begin{aligned} Var(M_{M,s}f) &= E[(M_{M,s}f - I_{s,\rho}f)^2] = Var\left(\frac{1}{M} \sum_{k=1}^M f(t_k)\right) \\ &= \frac{1}{M^2} \sum_{k=1}^M Var(f(t_k)) = \frac{\sigma(f)^2}{M} \end{aligned}$$

Se consideriamo ora il metodo Quasi Monte Carlo, l'integrale viene approssimato attraverso i punti deterministici t_1, \dots, t_M

$$I_s f \approx Q_{M,s}^{QMC} f = \sum_{k=1}^M f(t_k)$$

Se l'errore quadratico medio per il metodo Monte Carlo è facilmente calcolabile, stimando la varianza della funzione, tramite la formula (2.2), per il metodo QMC non esiste una formula analoga per l'errore.

A questo scopo introduciamo le 'randomly shifted QMC rules'. L'idea è di 'shiftare' i punti t_k di una quantità Δ scelta da una distribuzione uniforme su $[0, 1)^s$. Se i punti ottenuti dopo lo shift escono dall'intervallo si sottrae 1

al punto in modo da farlo rientrare nell'intervallo dalla parte opposta. Otteniamo quindi, come per QMC, la corrispondente formula:

$$Q_{M,s,\Delta}^{QMC} f = \sum_{k=1}^M f(t_k + \Delta)$$

Se prendiamo q diversi Δ , presi da una distribuzione uniforme su $[0, 1)^s$, possiamo utilizzare come stima la media

$$\bar{Q}_{M,s} f = \frac{1}{q} \sum_{i=1}^q Q_{M,s,\Delta_i}^{QMC} f$$

Ora l'approssimazione ha l'aspetto dell'approssimazione MC e possiamo stimare l'errore come

$$\sqrt{\frac{1}{q(q-1)} \sum_{i=1}^q (Q_{M,s,\Delta_i}^{QMC} f - \bar{Q}_{M,s} f)^2}$$

Alcuni esempi

Applichiamo le randomly shifted QMC rules al prezzaggio di un'opzione europea che vedremo in seguito nel Capitolo 3.

Utilizziamo per esempio la sequenza di Faure: gli errori, riferiti a 100000 simulazioni, senza lo shift li prendiamo dalle Tabelle (3.3), (3.4) e (3.5)

dimensione	10	25	50
senza shift	0.1670	0.0159	0.0827
con shift	0.0029	0.0053	0.0061

Tabella 2.1: *Stima dell'errore per QMC*

2.3 Inversione di Moro

Tra i passi che dobbiamo compiere per ottenere una simulazione di un cammino è la trasformazione dei numeri uniformi, provenienti dalle sequenze a bassa discrepanza, in numeri normali. La via principale e diretta per fare questa trasformazione è utilizzare l'inversa della funzione di distribuzione normale cumulata:

$$Y(x) = \frac{1}{\sqrt{2\pi}} \int_{-\text{inf}}^x e^{-\frac{t^2}{2}} dt$$

In un problema di simulazione, abbiamo la $Y(x)$ che proviene da una distribuzione uniforme $Y(x) \approx U[0, 1]$, e vogliamo il relativo campione x , distribuito normalmente. In altre parole, vogliamo $x(Y)$, cioè l'inverso della distribuzione cumulativa. Quindi, dato un campione u preso da una distribuzione uniforme, si vuole ottenere il valore $x(u)$, cioè, il campione corrispondente per la distribuzione di probabilità desiderabile.

Il modo più conosciuto di generare punti di distribuzione normale è quello di utilizzare l'algoritmo di Box-Muller. Lo svantaggio è che l'algoritmo è molto lento sia per le sequenze pseudo-random (MC) sia per quelle low-discrepancy (QMC). Inoltre, per le sequenze a bassa discrepanza è stato riportato che questo non è un buon modo di procedere, in quanto altera le proprietà della sequenza a bassa discrepanza (altera l'ordine della sequenza e rimescola l'uniformità della sequenza). Il metodo Box-Muller e le sue varianti infatti modificano la spaziatura dei punti delle sequenze low-discrepancy. Un altro algoritmo di inversione tradizionale, più veloce di quello di Box-Muller, è quello di Beasley e Springer; questo algoritmo però diventa inaccurato per le code della distribuzione normale.

Quello che propone [Moro (1995)] è un algoritmo ibrido che utilizza l'algoritmo di Beasley e Springer per la parte centrale della distribuzione normale e un altro algoritmo basato sulle serie troncate di Chebychev. Per la simmetria della distribuzione normale, possiamo lavorare solo sull'intervallo $[0.5, 1]$. Per bassi valori della deviazione standard, $Y < 0.92$, si usa la funzione di Beasley e Springer:

$$x(Y) = y \frac{\sum_{n=0}^3 a_n y^{2n}}{1 + \sum_{n=1}^4 b_n y^{2n}}$$

dove $y = Y - 0.5$.

Per $Y > 0.92$ si approssima $x(Y)$ con il polinomio:

$$x(Y) = \sum_{n=0}^8 c_n z^n$$

dove $z = k_1 \ln(-\ln(1 - Y)) + k_2$; a_n, b_n, c_n sono coefficienti scelti opportunamente e k_1 e k_2 sono costanti tali che $z = -1$ quando $Y = 0.92$ e $z = 1$ quando $Y = 1 - 10^{-12}$.

Per le nostre simulazioni abbiamo utilizzato la funzione Matlab *icdf*. Se confrontiamo infatti i valori in Tabella 2.2 ottenuti applicando *icdf* e l'algoritmo di Moro (per l'implementazione si veda l'Appendice) alla sequenza di Faure, i valori sono i medesimi.

Sequenza	moro	icdf
Faure		
0.9375	1.5341	1.5341
0.0313	-1.8627	-1.8627
0.5313	0.0784	0.0784
0.2813	-0.5791	-0.5791
0.7813	0.7764	0.7764
0.1563	-1.0100	-1.0100
0.6563	0.4023	0.4023
0.4063	-0.2372	-0.2372
0.9063	1.3180	1.3180
0.0938	-1.3180	-1.3180
0.5938	0.2372	0.2372
0.3438	-0.4023	-0.4023
0.8438	1.0100	1.0100
0.2188	-0.7764	-0.7764
0.7188	0.5791	0.5791
0.4688	-0.0784	-0.0784
0.9688	1.8627	1.8627
0.0156	-2.1539	-2.1539
0.5156	0.0392	0.0392

Tabella 2.2: *Confronto Moro e Icdf*

2.4 Tipi di sequenze

Dopo aver analizzato le proprietà generali delle sequenze a bassa discrepanza passiamo ora ad illustrare le più note.

2.4.1 Sequenza di Van der Corput

La sequenza Van der Corput (1935) è la componente di diverse importanti sequenze a bassa discrepanza. Le sequenze che descriveremo in seguito infatti partono dalla sequenza Van der Corput e su di essa vengono applicate delle modifiche.

Questa sequenza inizia da zero, ed è confinata nell'intervallo $[0, 1)$. L'intervallo $[0, 1)$ non contiene 1 perché la sequenza non riesce a raggiungere il numero 1 (per un numero di punti molto grande possiamo ottenere punti molto prossimi a 1). Tuttavia, in molte applicazioni è più comodo partire da 1 invece di 0, in 0 infatti ci sono dei problemi nell'inversione per ottenere la distribuzione normale.

Generazione dell'elemento n della sequenza

Per ottenere l'elemento n -esimo x_n della sequenza Van der Corput, avente come base b , bisogna procedere in tre fasi:

1. Il numero in base decimale n viene scritto in base b .

$$n = \sum_{i=0}^I a_i(n)b^i$$

I è il più piccolo intero tale che $a_i(n) = 0$ per ogni $i > I$.

$$I = \text{parte intera} \left(\frac{\ln(n)}{\ln(b)} \right).$$

Ad esempio, $n = 4$ in base 2 è 100: $4 = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$ dove $b = 2$, $a_0 = 0$, $a_1 = 0$, $a_2 = 1$

2. le cifre del numero in base b si riflettono rispetto al punto decimale. Nell'esempio: $100 \rightarrow 0,001$.

3. Si prende il numero decimale corrispondente al numero in base b .

$$X_n = \phi_b(n) = \sum_{i=0}^I \frac{a_i(n)}{b^{i+1}}$$

Nell'esempio, il numero in base 2 0,001 corrisponde al numero decimale $\frac{1}{8}(= 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3})$, dove, come prima $b = 2$, $a_0 = 0$, $a_1 = 0$, $a_2 = 1$.

Esempio:

La sequenza in base 2 inizia:

$$\begin{aligned} n = 1 & \quad 1 = 1 \times 2^0 & \rightarrow 1, & \quad x_1 = \phi_2(1) = \frac{1}{2}; \\ n = 2 & \quad 2 = 1 \times 2^1 + 0 \times 2^0 & \rightarrow 10, & \quad x_2 = \phi_2(2) = \frac{0}{2} + \frac{1}{4} = \frac{1}{4}; \\ n = 3 & \quad 3 = 1 \times 2^1 + 1 \times 2^0 & \rightarrow 11, & \quad x_3 = \phi_2(3) = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}; \\ n = 4 & \quad 2 = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 & \rightarrow 100, & \quad x_4 = \phi_2(4) = \frac{0}{2} + \frac{0}{4} + \frac{1}{8} = \frac{1}{8}; \end{aligned}$$

Si possono costruire delle sequenze con basi differenti. Le basi sono sempre numeri primi. In tabella riportiamo i primi 15 numeri in base 2 e in base 3.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\phi_2(n)$	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{5}{8}$	$\frac{3}{8}$	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{9}{16}$	$\frac{5}{16}$	$\frac{13}{16}$	$\frac{3}{16}$	$\frac{11}{16}$	$\frac{7}{16}$	$\frac{15}{16}$
$\phi_3(n)$	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{9}$	$\frac{4}{9}$	$\frac{7}{9}$	$\frac{2}{9}$	$\frac{5}{9}$	$\frac{8}{9}$	$\frac{1}{27}$	$\frac{10}{27}$	$\frac{19}{27}$	$\frac{4}{27}$	$\frac{13}{27}$	$\frac{22}{27}$	$\frac{7}{27}$

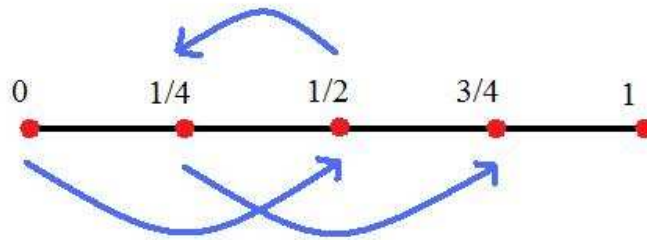
In ogni sequenza si possono individuare dei ‘cicli’: dopo ogni $2^n - 1$ punti, la sequenza è ‘maximally spread out’, cioè il più grande intervallo $(a, b) \subset (0, 1)$ che non contiene nessun punto della sequenza è il più piccolo possibile.

Differenti basi hanno diverse lunghezze del ciclo, cioè la quantità di numeri presenti nel ciclo. All’interno del ciclo i numeri risultano crescenti e il fattore di crescita è proporzionale a potenze del reciproco della base; l’ampiezza della griglia tende a diminuire passando da un ciclo al successivo e ciascun numero presenta la proprietà di riempire i buchi lasciati dai numeri che lo hanno preceduto.

Nella sequenza in base 2, ad esempio, i numeri sono raggruppati a due a due:

$$0 \quad \frac{1}{2} \quad \left| \quad \frac{1}{4} \quad \frac{3}{4} \quad \right| \quad \frac{1}{8} \quad \frac{5}{8} \quad \left| \quad \frac{3}{8} \quad \frac{7}{8} \quad \right| \quad \dots$$

In base 2, le coppie $(0, \frac{1}{2})$ e $(\frac{1}{4}, \frac{3}{4})$ sono i primi due cicli con lunghezza pari a 2 (il ciclo è composto da 2 numeri). Per le altre basi il ciclo è più largo. Per esempio, in base 3 la sequenza ha a denominatore le potenze del 3, e la

Figura 2.4: *Primi 2 cicli*

lunghezza del ciclo è 3. Per esempio il primo ciclo è $(0, \frac{1}{3}, \frac{2}{3})$.

$$0 \quad \frac{1}{3} \quad \frac{2}{3} \quad \left| \quad \frac{1}{9} \quad \frac{4}{9} \quad \frac{7}{9} \quad \left| \quad \frac{2}{9} \quad \frac{5}{9} \quad \frac{8}{9} \quad \left| \quad \dots$$

La sequenza di Van der Corput è la sequenza a bassa discrepanza base in una dimensione. Tuttavia, come generare sequenze multi-dimensionali a bassa discrepanza? Halton (1960), Faure (1982) e Sobol (1967), sono le sequenze più note a bassa discrepanza. Il processo di costruzione di queste nuove sequenze low-discrepancy inizia sub-dividendo l'ipercubo unitario in sotto-unità di volume costante, che hanno facce parallele alle facce dell'ipercubo. L'idea è di inserire un numero in ciascuno di questi sub-volumi prima di passare ad una griglia più fine. Le sequenze di Faure e Sobol sono come la sequenza di Halton, ma con una sola base e con un riordino degli elementi. La differenza principale è legata alla costruzione multi-dimensionale delle sequenze. Le sequenze di Faure e le sequenze di Sobol cambiano dimensione con una permutazione dei vettori di numeri quasi-random. Si effettua cioè un riordinamento del vettore all'interno di ogni dimensione. La sfida principale per le sequenze a bassa discrepanza è quello di evitare i clustering multidimensionali causati dalle correlazioni tra le dimensioni. Vogliamo generare sequenze low-discrepancy con nessuna correlazione per ogni coppia di dimensioni.

2.4.2 Sequenza di Halton

La prima dimensione della sequenza di Halton non è altro che la sequenza Van der Corput in base 2 prima descritta. Per le altre dimensioni vengono usate come basi dei numeri primi: per la seconda dimensione si utilizza la

sequenza Van der Corput in base 3, per la terza dimensione la sequenza Van der Corput in base 5 La dimensione s della sequenza usa l' s -esimo numero primo come base. Più elevata è la dimensione più aumenta il ciclo di base e il tempo di calcolo. La sequenza corrispondente al numero primo p ha cicli di lunghezza p con numeri crescenti monotonicamente. Questa caratteristica rende i primi termini di due sequenze altamente correlati, perlomeno il primo ciclo di ognuna delle sequenze. In una dimensione, utilizzando come base il numero p , l' n -esimo elemento della sequenza H_n è ottenuto con gli stessi passaggi usati per la costruzione della sequenza Van der Corput:

1. il numero n viene scritto in base p .

Per esempio, supponiamo $p=3$ e $n=22$, possiamo scrivere 22 in base 3 come

$$22 = 2 \times 3^2 + 1 \times 3^1 + 1 \times 3^0 \longrightarrow 211$$

2. le cifre del numero si riflettono rispetto al punto decimale. Nell'esempio $211 \rightarrow 0,112$

3. H_n è il numero decimale corrispondente al numero in base p .

In generale se vogliamo creare una sequenza d -dimensionale, si prendono d basi b_1, b_2, \dots, b_d . Si parte dal numero intero m e si sceglie un t_{mj} abbastanza grande. Si rappresenta m in ogni base:

$$m = \sum_{k=0}^{t_{mj}} a_{jk}(m) b_j^k \quad j = 1 \dots d$$

si calcola il punto della sequenza $x_{i,j}$

$$x_{ij} = \sum_{k=0}^{t_{mj}} a_{jk}(m) b_j^{k-t_{mj}-1} \quad j = 1 \dots d$$

si pone $m = m + 1$ e si ripete. In un problema d -dimensionale, ogni componente della sequenza è costruita con differenti basi prime p (vengono usati i primi d numeri primi). Ogni volta che il numero di cifre in n aumenta, la frazione riservata alle cifre di n diventa un fattore di p meglio raffinato, ovvero all'aumentare di n i punti della sequenza di Halton si adattano meglio

alle griglie cartesiane.

	Dimensione 1 Base 2	Dimensione 2 Base 3	Dimensione 3 Base 5
n=1	1/2	1/3	1/5
n=2	1/4	2/3	2/5
n=3	3/4	1/9	3/5
n=4	1/8	4/9	4/5
n=5	5/8	7/9	1/25
n=6	3/8	2/9	6/25
n=7	7/8	5/9	11/25
n=8	1/16	8/9	16/25

La sequenza conserva le sue caratteristiche di base a partire da un numero iniziale diverso, in modo che non sia necessario partire da 0 o 1. In aggiunta, ci sono alcuni vantaggi a togliere i primi numeri di una sequenza in modo da migliorare l'uniformità in grandi dimensioni.

Il problema principale per le sequenze pure quasi-random è la loro de-

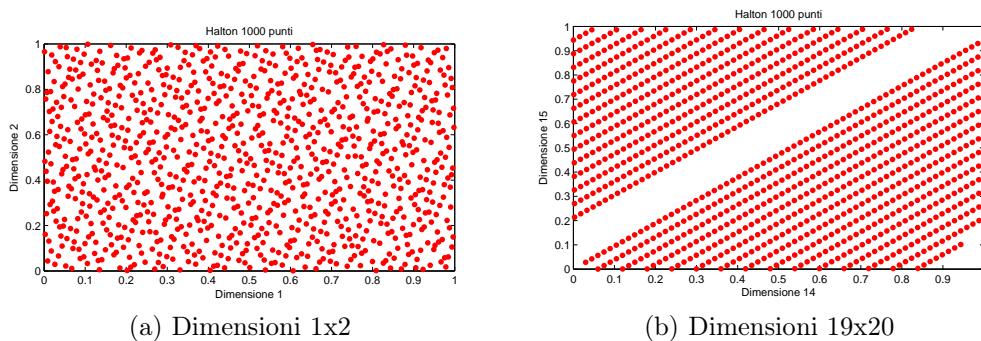


Figura 2.5

gradazione, quando la dimensione è grande. Il processo di generazione di punti uniformemente distribuiti in $[0, 1)^d$ diventa sempre più difficile, perché quando d aumenta, i punti di una dimensione generati presentano forte correlazione con le altre dimensioni (si veda la Figura 2.5b).

La Figura 2.6 riporta i primi 10 termini delle prime 8 dimensioni della sequenza di Halton ed evidenzia come al crescere della base diminuiscono le proprietà di uniformità della serie.

Figura 2.6: Primi 10 termini della sequenza di Halton in 8 dimensioni

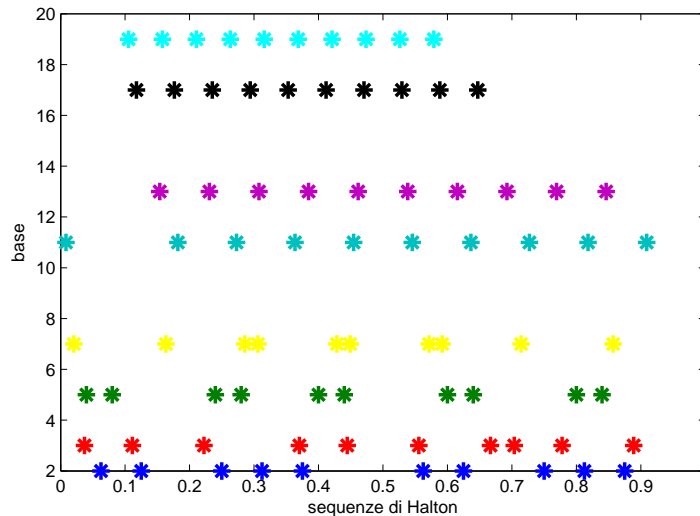
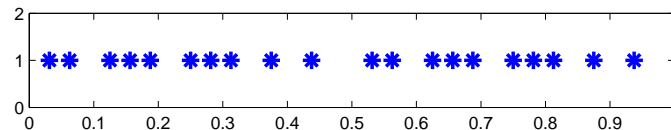


Figura 2.7: Primi 20 termini della sequenza di Halton in base 19



Questo problema non si presenta se il numero di punti della sequenza è molto maggiore rispetto alla base utilizzata. Già con 20 punti e come base 19 la sequenza si distribuisce uniformemente (Figura 2.7).

La sequenza di Halton con d grande presenta un ciclo di lunghezza elevata, a causa del grande numero primo usato come base. Ad esempio, nella dimensione 50, viene utilizzata la base 229 che è il 50-esimo numero primo. L'elevata lunghezza del ciclo significa che la sequenza in grandi dimensioni ha bisogno di diversi numeri per coprire tutto l'intervallo $[0, 1)$. Come sottolineato da

[Galanti & Jung (1997)]: ‘Ciò implica che la velocità con cui i punti della griglia più fine sono generati diminuisce all’aumentare della dimensione’. La Figura 2.5b indica che la sequenza di Halton, nonostante appaia uniforme in 2 dimensioni (Figura 2.5a), mostra che, in elevate dimensioni c’è correlazione tra i punti. In generale, la sequenza di Halton diventa insufficiente dopo la dimensione 14. I punti nelle successive dimensioni sono fortemente correlati e i punti iniziali sono concentrati vicino allo 0.

2.4.3 Sequenza di Faure

La sequenza di Faure differisce dalla sequenza di Halton per due aspetti. In primo luogo, utilizza una sola base per tutte le dimensioni. In secondo luogo, utilizza una permutazione degli elementi del vettore per ogni dimensione. La base della sequenza di Faure è il più piccolo numero primo che è maggiore o uguale al numero di dimensioni del problema, o uguale a 2 per un problema a una dimensione. Così facendo, la sequenza di Faure lavora con sequenze di Van der Corput di ciclo lungo per problemi di grandi dimensioni. Cicli lunghi hanno il problema di maggiore tempo di calcolo (rispetto a sequenze di ciclo più brevi).

Come si è verificato con le sequenze di Halton in grandi dimensioni, c’è il

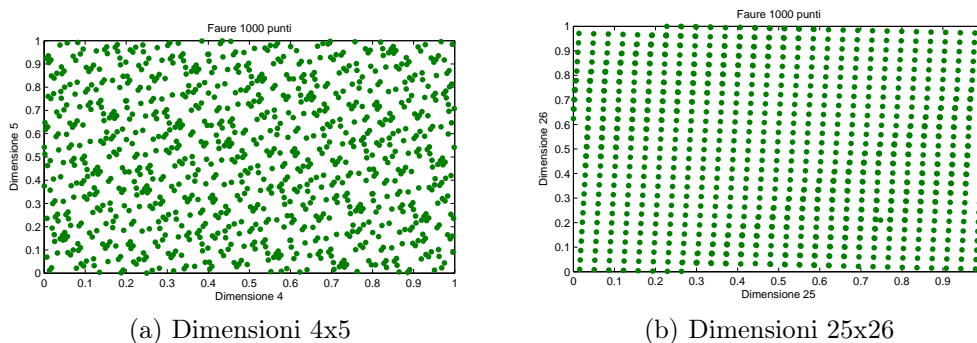


Figura 2.8

problema della bassa velocità con cui la sequenza di Faure genera i punti di una griglia sempre più fine per coprire l’ipercubo di lato unitario. Tuttavia, questo problema non è grave come nella sequenza di Halton. Per esempio, se la dimensione del problema è 50, l’ultima sequenza di Halton (in dimensione 50) utilizza il cinquantesimo numero primo che è 229, mentre la sequenza di Faure utilizza il primo numero primo dopo 50, che è 53, molto più piccolo di 229.

Riordinando la sequenza all’interno di ogni dimensione, la sequenza di Faure

previene per dimensioni non troppo elevate, alcuni problemi di correlazione che si sono verificati con la sequenza di Halton. La sequenza di Faure mostra un degrado per le alte dimensioni, dalla dimensione 25 circa (Figure 2.8b). L'algoritmo per la sequenza d -dimensionale di Faure utilizza le due equazioni usate per la costruzione della sequenza Van der Corput .

1. Si inizia rappresentando il numero intero n in base p come

$$n = \sum_{i=0}^I a_i^1(n) p^i.$$

2. La prima dimensione del punto di Faure è data riflettendo le cifre rispetto al punto decimale.
3. Poi come nel caso Van der Corput:

$$x_n^1 = \phi_p^1(n) = \sum_{i=0}^I \frac{a_i^1(n)}{p^{i+1}}.$$

Per trovare tutti gli altri punti delle altre dimensioni, si usa una formula ricorsiva (dalla dimensione $k-1$ alla dimensione k).

Innanzitutto bisogna assumere di conoscere tutti gli $a_i^{k-1}(n)$. Il livello successivo dei coefficienti è ottenuto moltiplicando $a_i^{k-1}(n)$ per una matrice triangolare superiore:

$$\mathbf{X} = \begin{pmatrix} \binom{0}{0} & \binom{0}{1} & \binom{0}{2} & \binom{0}{3} & \dots \\ 0 & \binom{1}{1} & \binom{1}{2} & \binom{1}{3} & \dots \\ 0 & 0 & \binom{2}{2} & \binom{2}{3} & \dots \\ 0 & 0 & 0 & \binom{3}{3} & \dots \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots \\ 0 & 1 & 2 & 3 & \dots \\ 0 & 0 & 1 & 3 & \dots \\ 0 & 0 & 0 & 1 & \dots \end{pmatrix}$$

In questo modo gli $a_i^k(n)$ sono ottenuti dalla formula

$$a_i^k(n) = \sum_{j \geq i}^m \frac{j!}{i!(j-i)!} a_j^{k-1}(n) \text{ mod } b$$

dove $a \text{ mod } b$ è il resto della divisione tra a e b .

I punti successivi della sequenza sono ottenuti da:

$$x_n^k = \phi_p^k(n) = \sum_{i=0}^I \frac{a_i^k(n)}{p^{i+1}} \quad 2 \leq k \leq d.$$

Questa procedura ricorsiva permette di generare i d punti corrispondenti al numero intero n con base p ($\geq s$).

Per esempio se prendiamo $d=3$ e $p=3$, i primi elementi della sequenza sono:

	$a_0(n)$	$a_1(n)$	$a_2(n)$	Dimensione 1 X_n^1	Dimensione 2 X_n^2	Dimensione 3 X_n^3
n=1	1	0	0	1/3	1/3	1/3
n=2	2	0	0	2/3	2/3	2/3
n=3	0	1	0	1/9	4/9	7/9
n=4	1	1	0	4/9	7/9	1/9
n=5	2	1	0	7/9	1/9	4/9
n=6	0	2	0	2/9	8/9	5/9
n=7	1	2	0	5/9	2/9	8/9
n=8	2	0	1	8/9	5/9	2/9

Come viene riportato da [Galanti & Jung (1997)], la sequenza di Faure soffre del problema della scelta del punto iniziale n_0 e, in particolare per le alte dimensioni e bassi valori di n_0 , i numeri di Faure esibiscono dei cluster vicini allo zero. Al fine di ridurre questo problema, Faure suggerisce di eliminare i primi $n = (p^4 - 1)$ punti, dove p è la base. Il problema di start-up è segnalato anche per altre sequenze con lo stesso suggerimento pratico di disfarsi di alcuni numeri della sequenza iniziale.

2.4.4 Sequenza di Sobol

La sequenza di Sobol (si veda [Bratley & Fox (1988)]), come la sequenza di Faure, ha la stessa base per tutte le dimensioni e procede a un riordino degli elementi vettoriali all'interno di ogni dimensione. La sequenza di Sobol è più semplice (e più veloce) rispetto alla sequenza di Faure: Sobol usa la base 2 per tutte le dimensioni. Vi è quindi un certo vantaggio nel tempo di calcolo a causa della minor lunghezza del ciclo. Per quanto riguarda il riordino dei vettori, la sequenza di Sobol, è più complessa rispetto a quella di Faure. Il riordino di Sobol è basato su un insieme di 'numeri direzionali', $v_{i,j}$. I numeri $v_{i,j}$ sono dati dall'equazione

$$v_{i,j} = \frac{m_{i,j}}{2^i},$$

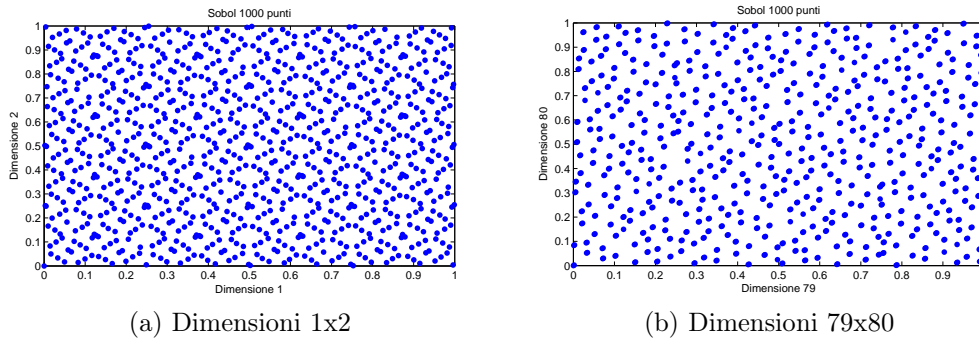


Figura 2.9

dove gli $m_{i,j}$ sono i numeri dispari interi positivi minori di 2^i oppure in forma binaria

$$v_{i,j} = 0, v_1 v_2 v_3 \dots$$

Costruiamo innanzitutto la sequenza costituita da $x^{1,j}, x^{2,j}, \dots$, con $0 \leq x^{i,j} \leq 1$ in una generica dimensione j . Per ottenere i $v_{i,j}$ dobbiamo scegliere un polinomio, primitivo, di grado d con i coefficienti che valgono 0 o 1, ad esempio

$$P = x^d + a_1 x^{d-1} + \dots + a_{d-1} x + 1$$

Una volta scelto il polinomio, usiamo i coefficienti $a_{i,j}$ per calcolare ricorsivamente i $v_{i,j}$

$$v_{i,j} = a_1 v_{i-1,j} \oplus a_2 v_{i-2,j} \oplus \dots \oplus a_{d-1} v_{i-d+1,j} \oplus v_{i-d,j} \oplus [v_{i-d,j}/2^d] \quad i > d$$

dove l'operatore \oplus (*bitxor*) indica l'operazione 'bit-by-bit' di esclusione

$$\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 \end{array}$$

e l'ultimo termine è $v_{i-d,j}$ shiftato a destra di d posti. Equivalentemente possiamo riscrivere la formula di ricorrenza in funzione delle $m_{i,j}$

$$m_{i,j} = 2a_{1,j}m_{i-1,j} \oplus 2^2 a_{2,j}m_{i-2,j} \oplus \dots \oplus 2^{d-1} a_{d-1,j}m_{i-d+1,j} \oplus 2^d m_{i-d,j} \oplus m_{i-d,j}. \quad (2.4)$$

Utilizzando un polinomio primitivo di grado d , i valori di $m_{1,j}, m_{2,j}, \dots, m_{d,j}$ possono essere scelti arbitrariamente in modo che $0 < m_{i,j} < 2^i$; i valori

$m_{d+1,j}, m_{d+2,j}, \dots$ sono determinati univocamente dalla formula (2.4).

Esempio:

Scegliamo $P = x^3 + x + 1$, di grado $d=3$ con i valori iniziali $m_{1,j} = 1$, $m_{2,j} = 3$, $m_{3,j} = 7$.

La formula ricorsiva diviene:

$$m_{i,j} = 4m_{i-2,j} \oplus 8m_{i-3,j} \oplus m_{i-3,j}$$

Allora

$$\begin{aligned} m_{4,j} &= 12 \oplus 8 \oplus 1 \\ &= 1100 \oplus 1000 \oplus 0001 \quad \text{in binario} \\ &= 0101 \quad \text{in binario} \\ &= 5 \\ m_{5,j} &= 28 \oplus 24 \oplus 3 \\ &= 11100 \oplus 11000 \oplus 00011 \\ &= 00111 \\ &= 7 \\ m_{6,j} &= 20 \oplus 56 \oplus 7 \\ &= 010100 \oplus 111000 \oplus 000111 \\ &= 101011 \\ &= 43 \\ &\dots \end{aligned}$$

(2.5)

Scrivendo in modo binaria le $v_{i,j}$ corrispondenti otteniamo

$$v_{1,j} = 0, 1 \quad v_{2,j} = 0, 11 \quad v_{3,j} = 0, 111 \quad v_{4,j} = 0, 0101 \quad v_{5,j} = 0, 00111 \dots$$

A questo punto per generare i punti della sequenza $x^{1,j}, x^{2,j}, \dots$ si usa

$$x^{n,j} = b_1 v_{1,j} \oplus b_2 v_{2,j} \oplus \dots$$

dove $\dots b_3 b_2 b_1$ è la rappresentazione binaria di n . Quindi

$$\begin{aligned}
 0 &\rightarrow 0 & x^{0,j} &= 0 \rightarrow 0 \\
 1 &\rightarrow 1 & x^{1,j} &= 0, 1 \rightarrow 0, 5 \\
 2 &\rightarrow 10 & x^{2,j} &= 0, 11 \rightarrow 0, 75 \\
 3 &\rightarrow 11 & x^{3,j} &= 0, 1 \oplus 0, 11 \rightarrow 0, 25 \\
 4 &\rightarrow 100 & x^{4,j} &= 0, 111 \rightarrow 0, 875 \\
 5 &\rightarrow 101 & x^{5,j} &= 0, 1 \oplus 0, 111 \rightarrow 0, 375
 \end{aligned}$$

(2.6)

Questo è l'algoritmo originale di Sobol. La versione di Antonov e Saleev ([Joe & Kuo (2003)]) che utilizza la rappresentazione di Gray è più efficiente nella pratica.

Quello che cambia è la definizione degli $x^{n,j}$

$$x^{n,j} = g_1 v_{1,j} \oplus g_2 v_{2,j} \oplus \dots$$

dove $\dots g_3 g_2 g_1$ è la rappresentazione di Gray del numero i definita come segue

$$\text{gray}(i) = i \oplus [i/2] = \dots i_3 i_2 i_1 \oplus \dots i_4 i_3 i_2$$

i	$\text{gray}(i)$
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

con la proprietà che la rappresentazione di $\text{gray}(i)$ e $\text{gray}(i - 1)$ differiscono solo per una cifra e quindi si possono generare i punti modo ricorsivo

utilizzando

$$x^{0,j} = 0 \quad \text{e} \quad x^{i,j} = x^{i-1,j} \oplus v_{c_{i-1},j}$$

dove c_i è l'indice del primo 0 digitale a partire da destra nella rappresentazione binaria.

Abbiamo $c_0 = 1, c_1 = 2, c_2 = 1, c_3 = 3, c_4 = 1, c_5 = 2 \dots$ Con l'implementazione di Gray otteniamo i punti in ordine differente rispetto all'implementazione originale di Sobol ma la proprietà di uniformità è comunque mantenuta. Questo perché ogni blocco di 2^m punti è lo stesso dell'implementazione originale. Il vantaggio di questa implementazione è che si può iniziare a generare la sequenza da qualsiasi punto ed è più efficiente computazionalmente.

2.4.5 Confronti

Tempi computazionali

Per le simulazioni si è usato un processore AMD Turion(tm) X2 Ultra Dual Core Mobile ZM-80 con RAM 4,00 GB e sistema operativo a 32 bit Windows Vista.

Gli algoritmi per la generazione delle sequenze a bassa discrepanza sono riportati in Appendice.

In Tabella 2.3 sono riportati i tempi impiegati dagli algoritmi per la creazione delle sequenze a bassa discrepanza in 10 dimensioni mentre nella Tabella 2.4 sono riportati i tempi per 1000 simulazioni variando la dimensione.

N° simulazioni	1000	10000	100000
Halton	0.2697	2.6820	26.6414
Faure	0.0635	0.6155	6.1166
Sobol	0.0033	0.0014	0.0011

Tabella 2.3: *tempi computazionali in secondi delle sequenze a bassa discrepanza in 10 dimensioni*

Mentre per la generazione di una matrice di dimensioni (100000 x 25) il comando *rand*, utilizzato nel metodo Monte Carlo impiega un tempo praticamente nullo, la generazione delle sequenze a bassa discrepanza richiede tempi più lunghi. Dalla Tabella 2.3 si può vedere che l'algoritmo più lento risulta essere quello di Halton. Sia per Halton che per Faure al crescere del numero di simulazioni i tempi aumentano considerevolmente. Dalla Tabella 2.4 invece notiamo che il crescere delle dimensioni non influisce molto sui

Dimensione simulazioni	1	5	25
Halton	0.0538	0.1399	0.6727
Faure	0.0948	0.0743	0.0726
Sobol	0.0257	0.0052	0.0019

Tabella 2.4: *tempi computazionali in secondi delle sequenze a bassa discrepanza con 1000 simulazioni*

tempi di Faure e Sobol, mentre per Halton si.

Confronti in letteratura

Esistono vari confronti empirici di sequenze quasi-random in letteratura. Sarkar e Prasad (1987) non hanno trovato alcuna differenza tra la sequenza di Faure e la sequenza di Halton nel trovare la soluzione di equazioni integrali con metodi Monte Carlo. [Fox (1986)], d'altro canto, trova la performance della sequenza di Faure migliore rispetto a quella di Halton. Ciò è supportato anche dallo studio di [Bratley e Fox (1988)], che hanno trovato che le prestazioni della sequenza di Sobol è approssimativamente la stessa di quella della sequenza di Faure. In un applicazione in strumenti finanziari derivati, [Papageorgiou e Traub (1996)], hanno concluso che la sequenza di Faure era superiore rispetto alla sequenza di Sobol. Alcune delle performance inferiori delle sequenze di Sobol sono state attribuite alla cattiva scelta dei valori di partenza (Jackel (2002)).

In generale possiamo affermare che il problema principale per le sequenze semplici quasi-random è la loro degradazione, quando la dimensione è grande. Il processo di generazione di punti uniformemente distribuiti diventa sempre più difficile con l'aumentare della dimensione, perché lo spazio da riempire diventa troppo grande. Nonostante la sequenza di Sobol riesca a riempire i vuoti ad un più alto tasso (grazie alla lunghezza dei suoi cicli più breve), c'è il problema che in alte dimensioni i punti di Sobol tendono a ripetersi nelle varie dimensioni (si veda la Figura 2.5b). La soluzione tradizionale è quella di scartare i primi n punti. Come è stato osservato dall'esperienza di Faure è meglio se i primi n punti scartati siano uguali al numero primo utilizzato per generare la sequenza. [Boyle, Broadie e Glasserman (1997)] eliminano i primi 64 punti per la sequenza di Sobol, ma questo numero può essere scelto arbitrariamente.

L'altro problema delle sequenze quasi-random è la correlazione tra le dimensioni, soprattutto se stiamo lavorando con grandi dimensioni. Come mostra la Figura 2.3b, la sequenza di Halton diventa insufficiente dopo la dimensione

14. In pratica, a causa della correlazione, molte persone preferiscono evitare l'uso di sequenza di Halton per più di 6 o 8 dimensioni. Per le sequenze di Faure, la degradazione inizia con la dimensione 25 (si veda la Figura 2.4*b*). Concludendo possiamo affermare che la sequenza di Halton è dominata dalle sequenze di Faure e di Sobol. Anche se l'applicazione del metodo di Halton è molto più facile rispetto a quello di Sobol e di Faure, non è saggio utilizzare questa sequenza in dimensioni elevate. Per le dimensioni basse, invece, tutte le sequenze low-discrepancy possono essere utilizzate con successo.

Capitolo 3

Pricing di opzioni Europee

In questa sezione illustreremo l'implementazione del metodo Quasi Monte Carlo nel caso di opzioni Europee plain vanilla. Il prezzo esatto, sotto l'ipotesi che il sottostante segua un andamento lognormale, lo conosciamo e pertanto possiamo usare questo tipo di opzioni per esaminare la convergenza del metodo.

3.1 Opzioni Europee

Un'opzione europea call (put) conferisce al possessore dell'opzione il diritto di acquistare (vendere) il sottostante S entro una certa data T (maturity o data di scadenza) per un certo prezzo K (prezzo di esercizio). Supponiamo che S_T sia il prezzo finale del sottostante a scadenza. Il payoff di una posizione lunga su un'opzione call europea è $\max(S_T - K, 0)$, mentre per una posizione corta è $\min(K - S_T, 0)$. Il payoff di una posizione lunga in un'opzione put europea è $\max(K - S_T, 0)$, per la posizione corta, $\min(S_T - K, 0)$. Il prezzo esatto di questo tipo di opzione è possibile calcolarlo analiticamente, sempre sotto l'ipotesi che il sottostante segua un andamento lognormale, mediante le formule di Black & Scholes :

$$C(S, t) = S_t N(d_1) - K e^{-r(T-t)} N(d_2)$$
$$P(S, t) = K e^{-r(T-t)} N(-d_2) - S_t N(-d_1)$$

dove

- S_t è il prezzo del titolo sottostante;
- K è il prezzo d'esercizio dell'opzione;

- r è il tasso d'interesse privo di rischio, espresso in base annua;
- $N(\cdot)$ denota la funzione di ripartizione di una variabile casuale normale;
- σ^2 è la varianza percentuale istantanea del logaritmo del prezzo del titolo sottostante, espressa anch'essa su base annua;
-

$$d_1 = \frac{\ln \frac{S_t}{K} + \left(r + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{T-t}} \quad d_2 = d_1 - \sigma\sqrt{T-t}.$$

Siccome l'opzione non è path-dependent (non dipende dalla traiettoria dell'asset), non è necessario simulare tutto il cammino del sottostante, ma solo il valore finale. Creiamo N simulazioni indipendenti del valore a scadenza

$$S_T^n = S_0 e^{(r - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}\epsilon_n} \quad n = 1, 2, \dots, N$$

con $\epsilon_n \sim N(0, 1)$.

Per poter applicare i metodi MC e QMC, dobbiamo ricordare che il prezzo di una call, sotto misura di probabilità neutrale al rischio, è dato dalla formula di Feynman-Kac

$$C(S_t, t) = e^{-r(T-t)} E[\text{payoff}(S_T)]$$

ovvero, sostituendo le N simulazioni effettuate

$$C(S_0, 0) = \frac{1}{N} e^{-rT} \sum_{n=1}^N \max(S_T^n - K, 0)$$

Cioè che contraddistingue i due metodi è ϵ_n : se viene simulato con MC, ϵ_n apparterrà a una sequenza pseudo-casuale, altrimenti, con QMC, a una sequenza deterministica.

Per valutare la qualità delle simulazioni, calcoliamo l'errore relativo percentuale:

$$\text{errore} = \left| \frac{\hat{C} - C_{BS}}{C_{BS}} \right| 100 \quad (3.1)$$

dove C_{BS} è il prezzo fornito dalla formula di Black & Scholes e \hat{C} il prezzo trovato tramite simulazione.

Consideriamo un'opzione call con scadenza 1 anno con i seguenti parametri:

- $S_0 = 100$

- $K = 100$
- $r = 10\%$
- $\sigma = 30\%$

Il valore di Black & Scholes è: 16.7341.

La Figura 3.1 riporta i valori della call ottenuti tramite il metodo MC e QMC su un range di simulazioni da 10000 a 100000 mentre la Figura 3.2 e la Tabella 3.1 riportano gli errori calcolati tramite la (3.1). Oltre al metodo MC semplice abbiamo anche confrontato i valori ottenuti tramite le tecniche di riduzione della varianza:

- *MC1* utilizza la variabile antitetica,
- *MC2* la variabile di controllo. Il prezzo della call è

$$\text{Prezzo} = e^{-rT} \max(S_T - K, 0) + \alpha(S_T - Ef)$$

$$\text{dove } Ef = S_0 e^{rT} \text{ e } \alpha = \frac{\text{Cov}(S_T, e^{-rT} \max(S_T - K, 0))}{S_0^2 e^{2rT} (e^{T\sigma^2} - 1)}$$

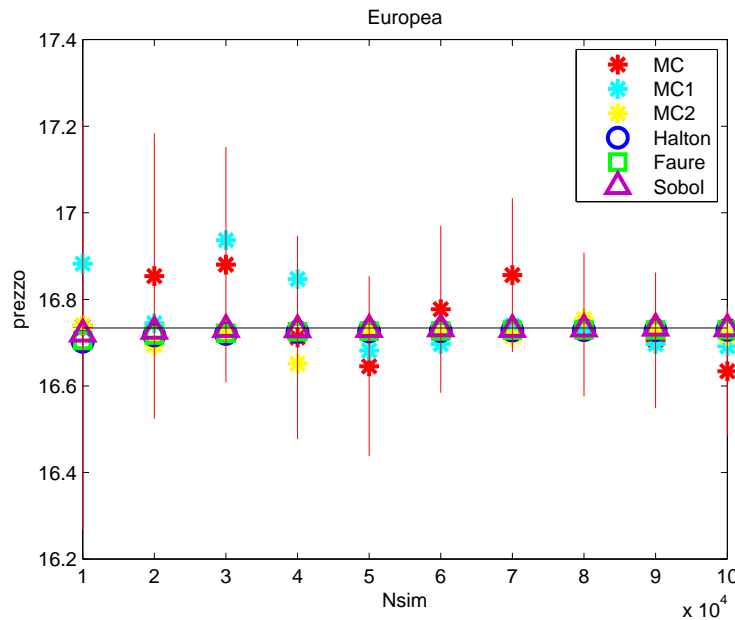


Figura 3.1: Prezzo $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 30\%$ per il metodo MC è riportato anche l'intervallo di confidenza

Il metodo Monte Carlo non essendo un metodo deterministico fornisce ogni volta che si fa partire l'algoritmo dei valori differenti del prezzo in quanto la sequenza pseudo-casuale viene generata con un seme differente. Ha più senso quindi parlare non di valore ottenuto ma di intervallo di confidenza per il prezzo dell'opzione. Per questo motivo oltre al prezzo medio inseriamo nei grafici dei prezzi l'intervallo di confidenza (solo per MC per maggiore chiarezza nelle figure). I metodi quasi Monte Carlo invece sono deterministici quindi forniscono sempre lo stesso valore ogni volta che si utilizza l'algoritmo di prezzaggio.

Per il metodo QMC abbiamo utilizzato le sequenze di Halton, Faure e Sobol. Si noti che in questo caso, essendo la data di monitoraggio unica, quindi la dimensione pari ad 1, le sequenze di Halton e Faure coincidono: Halton infatti per la prima dimensione utilizza la base 2 e Faure il primo numero primo maggiore o uguale alla dimensione, quindi 2.

Nelle figure dei prezzi la linea nera indica il prezzo fornito da Black & Scholes.

Dalla Tabella 3.1 si può notare come gli errori del metodo MC e le sue varianti MC1 e MC2 siano superiori agli errori dei metodi QMC.

Tra i metodi Quasi Monte Carlo la sequenza di Sobol è quella con l'errore minore; a partire da 50000 simulazioni l'errore è inferiore al 0.02%.

In generale per tutti i metodi QMC si può notare come questi errori decrescono all'aumentare del numero di simulazioni.

N° simulazioni	MC	MC1	MC2	Halton	Faure	Sobol
10000	0.0295	0.8873	0.0356	0.1902	0.1731	0.0839
20000	0.7141	0.0645	0.2390	0.1033	0.0992	0.0453
30000	0.8726	1.2123	0.0936	0.0830	0.0745	0.0203
40000	0.1318	0.6734	0.4924	0.0557	0.0547	0.0244
50000	0.5293	0.3132	0.0077	0.0493	0.0451	0.0187
60000	0.2568	0.2221	0.0658	0.0449	0.0429	0.0106
70000	0.7292	0.0311	0.1203	0.0355	0.0313	0.0192
80000	0.0466	0.0074	0.1256	0.0299	0.0296	0.0131
90000	0.1717	0.2220	0.0295	0.0307	0.0285	0.0018
100000	0.5991	0.2551	0.1141	0.0267	0.0257	0.0100

Tabella 3.1: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 30\%$

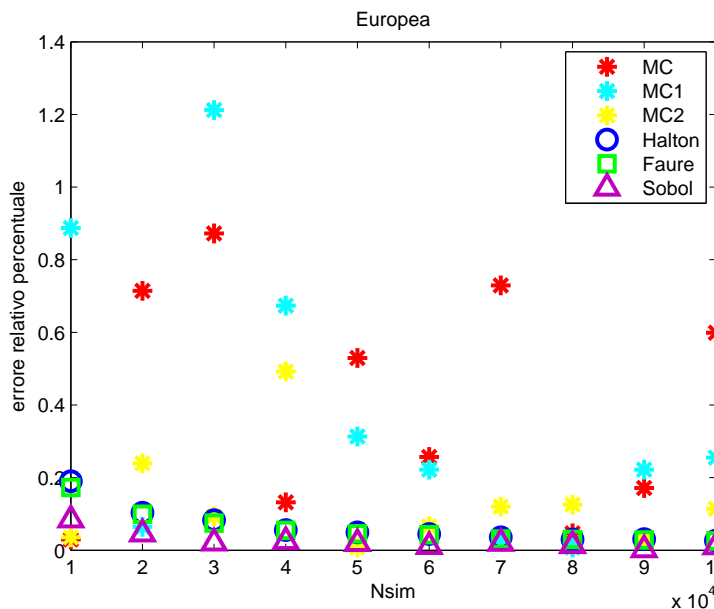


Figura 3.2: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 30\%$

3.1.1 Skip iniziale

Per cercare di evitare, o meglio limitare, la degradazione delle sequenze QMC nel corso degli anni sono stati proposti molti accorgimenti nella costruzione delle sequenze stesse. [Bratley & Fox (1988)] nel loro articolo confermano l'esistenza di valori favorevoli per il numero di simulazioni nella sequenza di Sobol che accrescono la proprietà di uniformità della sequenza. Il numero di simulazioni in s dimensioni, deve essere pari a 2^k dove $k \geq \max(2s, \tau_s + 1)$. I valori di τ_s per valori bassi di s sono

s	1	2	3	4	5	6	7	8	9	10	11	12	13
τ_s	0	0	1	3	5	8	11	15	19	23	27	31	35

Un'altro aspetto che è stato considerato è l'eliminazione dei primi punti delle sequenze. Secondo alcuni la sequenza di Sobol migliora la sua performance se vengono eliminati un numero di punti pari alla più grande potenza del 2 più piccola del numero di punti usati. [Joe & Kuo (2003)] sminuiscono l'impatto di questo accorgimento nel loro articolo.

Nell'articolo di [Fox (1986)] viene proposto invece per la sequenza di Faure di eliminare una porzione iniziale di punti della sequenza pari a $base^4 - 1$. Abbiamo confrontato le sequenze di Faure e Sobol costruite sia con lo skip

di punti iniziali sia senza. Nel pricing di un'opzione con una sola data di monitoraggio la presenza dello skip non è di grande rilevanza. La situazione cambia invece all'aumentare del numero di date di monitoraggio, ovvero all'aumentare della dimensione (Figura 3.3 e Tabella 3.2): la sequenza di Faure con lo skip subisce un notevole miglioramento. Rimane invece invariata la sequenza di Sobol.

Per le nostre simulazioni quindi utilizzeremo la sequenza di Faure con lo skip iniziale, mentre la sequenza di Sobol senza l'eliminazione di punti.

N° simulazioni	Faure senza skip	Faure con skip	Sobol senza skip	Sobol con skip
10000	3.7045	0.9149	1.1248	1.2644
20000	3.0670	0.0803	1.0173	1.0454
30000	2.9768	0.1210	0.4390	0.4300
40000	2.7728	0.0788	0.4216	0.4291
50000	2.6764	0.2589	0.3227	0.3312
60000	2.5066	0.3178	0.1380	0.1457
70000	2.3803	0.0948	0.1939	0.1856
80000	2.2312	0.1255	0.1045	0.1080
90000	2.1712	0.0222	0.0567	0.0521
100000	2.0446	0.0827	0.0635	0.0562

Tabella 3.2: Errori con 50 date di monitoraggio

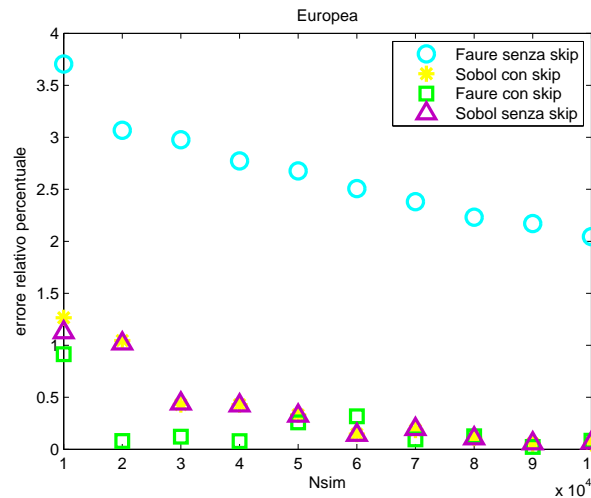


Figura 3.3: Errori con 50 date di monitoraggio

3.1.2 Date di monitoraggio maggiori di 1

Invece di simulare solo l'asset a maturity simuliamo anche l'intero cammino utilizzando un'approssimazione discreta.

Il prezzo non cambia perchè nella formula del prezzo rimane il valore a scadenza S_T , $C(S_0, 0) = e^{-rT} \max(S_T - K, 0)$.

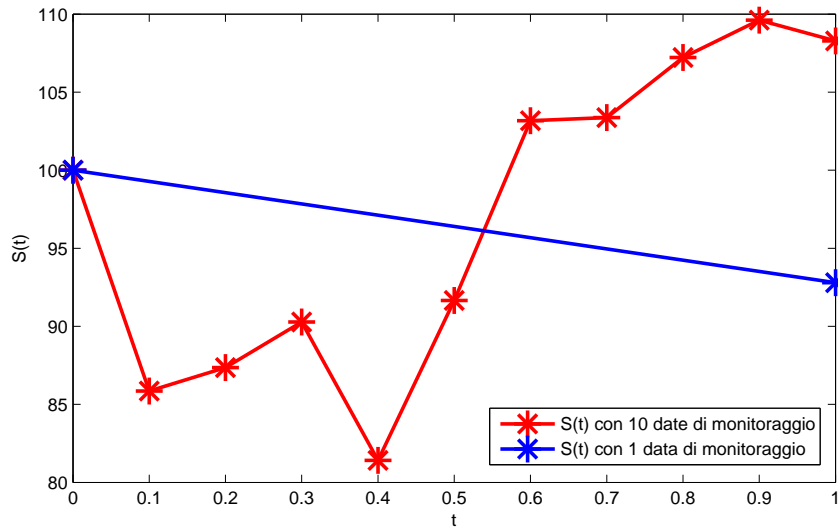


Figura 3.4: simulazione del prezzo del sottostante

In Figura 3.4, al posto di considerare solo il valore blu finale del sottostante S_T con $T = 1$ generato con una sola data di monitoraggio, consideriamo il valore rosso finale S_T con $T = 1$, valore finale di un cammino monitorato 10 volte. In questo modo, variando il numero delle date di monitoraggio possiamo esaminare l'impatto dell'aumento della dimensione nelle sequenze a bassa discrepanza.

Utilizziamo la stessa opzione call della sezione precedente:

- $S_0 = 100$
- $K = 100$
- $r = 10\%$
- $\sigma = 30\%$

Se utilizziamo 10 date di monitoraggio otteniamo gli errori in Tabella 3.3.

N° simulazioni	MC	MC1	MC2	Halton	Faure	Sobol
10000	2.2508	0.8020	0.0956	1.0991	0.1192	0.2677
20000	0.3214	0.4160	0.2928	0.5505	0.2245	0.0037
30000	0.0845	0.4958	0.1136	0.4051	0.2285	0.0946
40000	0.1638	0.0572	0.1258	0.3601	0.2001	0.0500
50000	1.2879	0.0801	0.1471	0.2305	0.1888	0.0305
60000	0.4217	0.3627	0.2488	0.1847	0.2402	0.0316
70000	0.3275	0.1860	0.0931	0.1727	0.1966	0.0309
80000	0.2654	0.0946	0.1591	0.1714	0.2045	0.0078
90000	0.4628	0.2102	0.1496	0.1893	0.1794	0.0006
100000	0.8661	0.0721	0.2715	0.1769	0.1670	0.0018

Tabella 3.3: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 30\%$ con 10 date di monitoraggio

Se confrontiamo la Tabella 3.3 con la Tabella 3.1 notiamo che gli errori sono maggiori rispetto al caso di una sola data di monitoraggio. Anche in questo caso le sequenze di Faure e Sobol sono migliori rispetto a tutte le altre. Halton nonostante sia peggio rispetto a Faure e Sobol, è comunque migliore del metodo MC. Tra i metodi Monte Carlo il migliore risulta essere MC2, variabile di controllo.

Per 25, 50, 100, 200 date di monitoraggio otteniamo i risultati nelle tabelle 3.4, 3.5, 3.6 e 3.7.

N° simulazioni	MC	MC1	MC2	Halton	Faure	Sobol
10000	2.4418	0.8365	0.0218	4.6260	0.3798	0.5972
20000	0.9067	0.8520	0.3530	2.5726	0.0347	0.3159
30000	0.7414	0.1835	0.2029	1.8639	0.0946	0.1307
40000	0.1077	0.1089	0.0827	1.4662	0.1375	0.0026
50000	0.6869	0.8503	0.5207	1.1108	0.0171	0.0168
60000	0.4843	0.2329	0.0366	0.9059	0.0034	0.0879
70000	0.4032	0.2147	0.0922	0.8416	0.0127	0.0899
80000	1.0260	0.1272	0.3333	0.7232	0.0144	0.0891
90000	0.3343	0.5548	0.0314	0.6772	0.0053	0.0908
100000	0.4258	0.2109	0.0644	0.6144	0.0159	0.1096

Tabella 3.4: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 30\%$ con 25 date di monitoraggio

N° simulazioni	MC	MC1	MC2	Halton	Faure	Sobol
10000	0.7291	0.3999	0.0218	15.4187	0.9149	1.1248
20000	0.6240	0.5541	0.3530	8.3804	0.0803	1.0173
30000	0.1630	0.1675	0.2029	5.5281	0.1210	0.4390
40000	0.0095	0.0677	0.0827	3.8081	0.0788	0.4216
50000	0.7373	0.1683	0.5207	3.3764	0.2589	0.3227
60000	0.0017	0.0759	0.0366	2.7152	0.3178	0.1380
70000	0.4852	0.2110	0.0922	2.6108	0.0948	0.1939
80000	0.2634	0.0560	0.3333	2.0812	0.1255	0.1045
90000	0.0466	0.2356	0.0314	2.1365	0.0222	0.0567
100000	0.2876	0.0866	0.0644	1.8856	0.0827	0.0635

Tabella 3.5: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 30\%$ con 50 date di monitoraggio

N° simulazioni	MC	MC1	MC2	Halton	Faure	Sobol
10000	0.2504	0.3499	0.6728	28.7560	1.0932	0.5152
20000	0.4909	0.3613	0.1376	23.0043	0.1849	1.1045
30000	0.0558	0.1777	0.0312	17.9786	0.1630	0.7107
40000	1.3380	0.3560	0.1542	14.8790	0.0813	0.6129
50000	0.7984	0.3701	0.0932	12.5398	0.1352	0.7123
60000	0.5150	0.1854	0.2464	10.9154	0.1156	0.5661
70000	0.7032	0.2376	0.2116	10.0475	0.0282	0.4154
80000	0.4020	0.5841	0.0110	8.8066	0.0060	0.3097
90000	0.2320	0.3242	0.2020	8.4524	0.0807	0.4125
100000	1.0261	0.2292	0.0031	7.4456	0.0400	0.3519

Tabella 3.6: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 30\%$ con 100 date di monitoraggio

Dopo la decima dimensione, gli errori di Halton sono troppo elevati da poterlo considerare affidabile. Con l'aumentare del numero delle dimensioni infatti inizia a delinarsi il problema della degradazione per la sequenza di Halton. Fino a 100 dimensioni Faure fornisce errori minori rispetto a Sobol. Per dimensioni ancora più elevate Sobol e Faure sono comparabili. Confrontando gli errori di queste due sequenze con gli errori dei metodi MC, verificiamo che per dimensioni basse, minori di 25, le sequenze a bassa discrepanza sono migliori delle sequenze pseudo-casuali. Fino a 100 dimensioni Faure e Sobol sono migliori di MC ma non di MC1 e MC2. Per 200 dimensioni il metodo migliore risulta essere MC2.

N° simulazioni	MC	MC1	MC2	Halton	Faure	Sobol
10000	2.0800	0.4056	0.2019	27.0861	0.2707	0.4660
20000	0.6223	0.2793	0.3420	30.3873	0.4213	0.0235
30000	0.0081	0.1571	0.0346	27.9296	0.0957	0.2685
40000	0.7652	0.0367	0.0167	26.4956	0.3834	0.1728
50000	0.2939	0.6639	0.0520	23.6364	0.7205	0.5261
60000	0.9070	0.3580	0.0780	22.0193	0.6265	0.5934
70000	0.3709	0.2944	0.0304	21.3812	0.2003	0.5984
80000	0.7756	0.4093	0.0463	20.5146	0.6091	0.3931
90000	0.0153	0.0296	0.0474	19.8821	0.1114	0.4258
100000	0.2995	0.1861	0.0038	19.1421	0.0116	0.3555

Tabella 3.7: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 30\%$ con 200 date di monitoraggio

In Figura 3.5 riportiamo gli errori al variare del numero delle dimensioni fissando il numero di simulazioni a 20000. Gli errori di Halton sono elevati a causa della degradazione della sequenza. Per questo motivo in Figura 3.6 riportiamo solo gli errori riferiti a MC, Faure e Sobol. Per dimensioni inferiori a 20, la sequenza di Sobol risulta essere migliore della sequenza di Faure; da 20 a 170 dimensioni gli errori di Faure sono minori di quelli di Sobol; per dimensioni elevate, superiori a 170, Sobol appare essere il migliore.

Figura 3.5: errore relativo con 20000 simulazioni

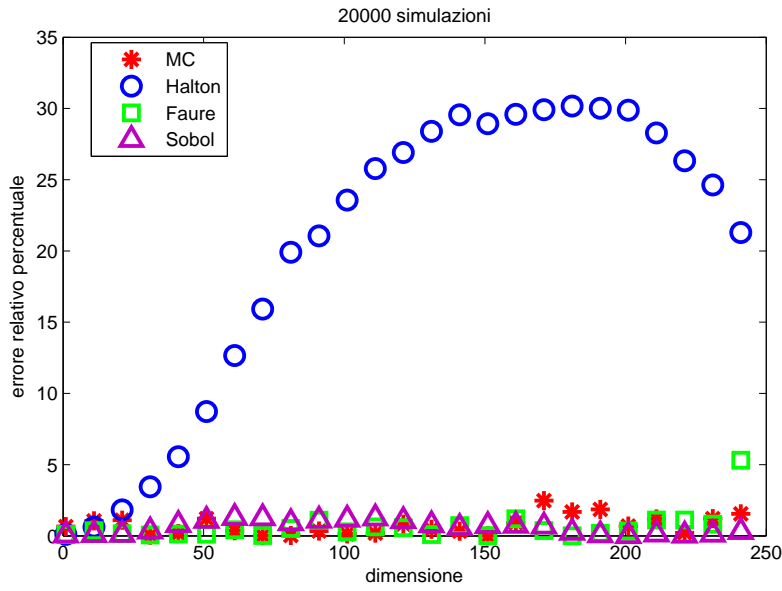
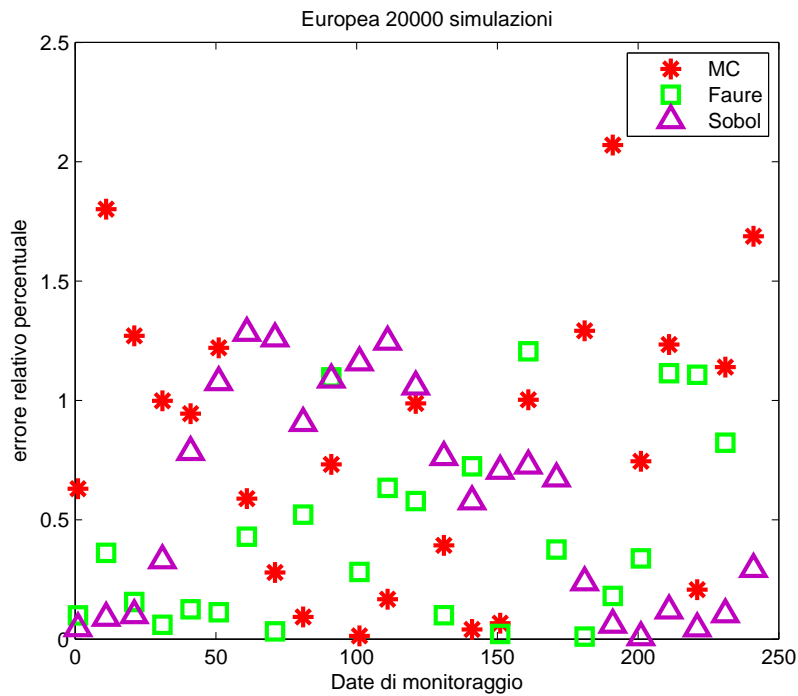


Figura 3.6: errore relativo con 20000 simulazioni



Capitolo 4

Pricing di opzioni Esotiche

Le opzioni Esotiche o path dependent sono opzioni il cui payoff alla scadenza dipende non solo dal valore alla scadenza ma anche dalla storia passata del sottostante.

Esempi di opzioni path dependent sono dati dalle opzioni americane, che consentono l'esercizio dell'opzione in qualunque momento della loro vita, le opzioni asiatiche, il cui valore a scadenza dipende da una media dei valori assunti dal sottostante nel corso della vita dell'opzione, e le opzioni barriera, che possono essere esercitate a seconda che il prezzo del sottostante attraversi o meno un dato valore, detto appunto barriera.

4.1 Opzioni Asiatiche

Le opzioni asiatiche sono probabilmente lo strumento derivato che meglio illustra il concetto di opzione path dependent; il loro valore a scadenza dipende da una media dei valori assunti dal sottostante lungo la vita dell'opzione stessa.

Un caso elementare è quello dell'opzione call asiatica average strike in tempo discreto, con media aritmetica, il cui valore, al momento della scadenza T , è dato da:

$$C(S_T, T) = \max \left(S_T - \frac{1}{T} \sum_{t=0}^T S_t, 0 \right) \quad (4.1)$$

dove S_t denota il prezzo del sottostante.

Un caso analogo è quello dell'opzione call asiatica average price in tempo discreto con valore:

$$C(S_T, T) = \max \left(\frac{1}{T} \sum_{t=0}^T S_t - K, 0 \right) \quad (4.2)$$

dove K è il prezzo d'esercizio. Le possibili variazioni sul tema sono numerose; si può passare dal tempo discreto al tempo continuo (così che i segni di sommatoria sopra sono sostituiti da integrali), nonché dalla media aritmetica alla media geometrica.

Quello su cui noi ci focalizzeremo sono le opzioni a media aritmetica del tipo (4.2).

Per confrontare i risultati useremo come prezzi di riferimento quelli calcolati da [Turnbull (1992)], indicati nei grafici da una linea nera.

Consideriamo un'opzione con i seguenti parametri (Figura 4.1):

- $S_0 = 100$
- $K = 100$
- $T = 1$
- $r = 9\%$
- $\sigma = 50\%$
- 53 date di monitoraggio

Il prezzo fornito da Turnbull è 12.98.

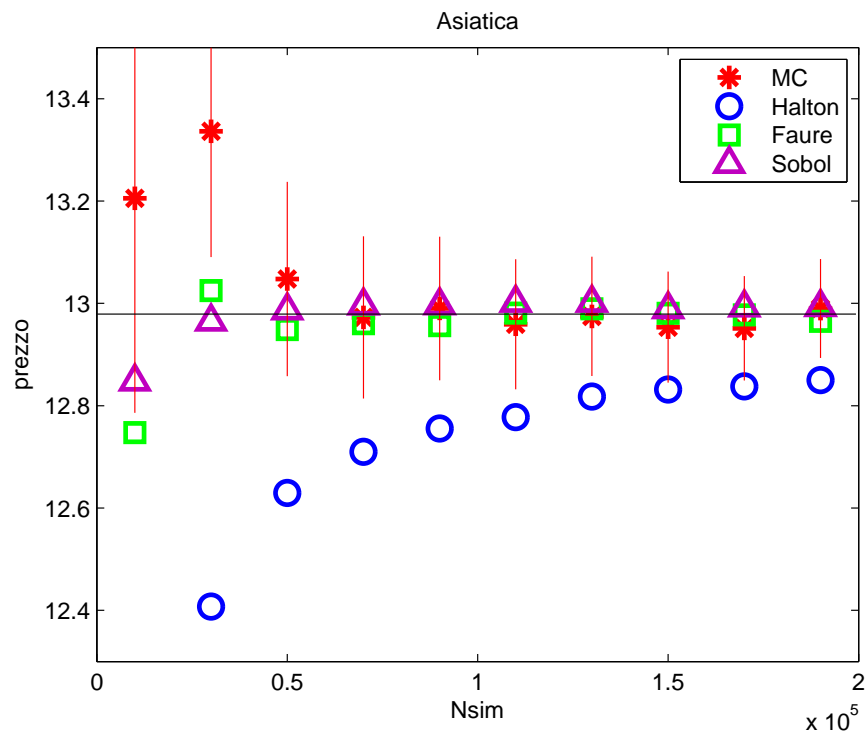
N° simulazioni	MC	Halton	Faure	Sobol
10000	13.2052 (1.7351)	11.4022 (12.1554)	12.7475 (1.7909)	12.8475 (1.0206)
30000	13.3365 (2.7462)	12.4074 (4.4113)	13.0249 (0.3460)	12.9647 (0.1182)
50000	13.0478 (0.5220)	12.6293 (2.7015)	12.9489 (0.2393)	12.9862 (0.0474)
70000	12.9725 (0.0575)	12.7100 (2.0804)	12.9601 (0.1532)	12.9960 (0.1232)
90000	12.9901 (0.0777)	12.7553 (1.7312)	12.9564 (0.1818)	12.9961 (0.1244)
110000	12.9591 (0.1609)	12.7776 (1.5595)	12.9810 (0.0079)	13.0013 (0.1643)
130000	12.9746 (0.0418)	12.8180 (1.2478)	12.9893 (0.0718)	13.0013 (0.1641)
150000	12.9536 (0.2035)	12.8318 (1.1417)	12.9806 (0.0047)	12.9888 (0.0675)
170000	12.9512 (0.2221)	12.8378 (1.0955)	12.9777 (0.0175)	12.9922 (0.0937)
190000	12.9899 (0.0766)	12.8501 (1.0011)	12.9649 (0.1163)	12.9929 (0.0992)

Tabella 4.1: Prezzi, (\cdot) errore rispetto a Turnbull 12.98

Possiamo notare come le sequenze di Faure e di Sobol hanno errori molto piccoli dopo 100000 simulazioni, minori degli errori di MC. La sequenza di Halton invece si è rivelata ancora una volta inaffidabile con errori che superano sempre l'1%. Questa bassa convergenza è dovuta all'alta dimensionalità del problema.

Se infatti proviamo a diminuire la dimensione (Tabella 4.2), quindi il numero di date di monitoraggio, possiamo verificare quello che accadeva per le opzioni Europee: si vede una convergenza maggiore (soprattutto per la sequenza

Figura 4.1: Prezzo Asiatica 53 date di monitoraggio



di Halton) rispetto a prima dei metodi QMC.

- $S_0 = 100$
- $K = 100$
- $T = 5/52$
- $r = 9\%$
- $\sigma = 50\%$
- 6 date di monitoraggio

N° simulazioni	MC	Halton	Faure	Sobol
10000	3.5308	3.6245	3.6335	3.6320
30000	3.6667	3.6347	3.6359	3.6373
50000	3.6444	3.6366	3.6370	3.6379
70000	3.6465	3.6372	3.6381	3.6378
90000	3.6270	3.6377	3.6370	3.6390
110000	3.6751	3.6377	3.6383	3.6391
130000	3.6310	3.6379	3.6387	3.6387
150000	3.6263	3.6382	3.6387	3.6380
170000	3.6469	3.6384	3.6390	3.6384
190000	3.6523	3.6384	3.6388	3.6384

Tabella 4.2: Prezzo con 6 date di monitoraggio

4.2 Opzioni Barriera

Un'altra classe di opzioni Esotiche sono le opzioni Barriera, dove il diritto all'esercizio viene meno se il valore del sottostante attraversa un certo valore L (barriera out), oppure lo strumento inizia ad esistere solo se si attraversa una certa barriera (barriera in).

Per esempio, sia N il numero di cammini, per un'opzione call down & out il payoff scontato dell' i -esimo cammino è dato da:

$$C(S_i) = \begin{cases} 0 & \text{se } \exists j : S_i(t_j) \leq L \\ e^{-rT} \max(S_T - K) & \text{se } S_i(t_j) > L \quad \forall j \end{cases}$$

Il prezzo finale sarà la media sulle N simulazioni del sottostante. Applicheremo il metodo MC e QMC a opzioni Barriera down & out.

- $S_0 = 100$
- $K = 100$
- $T = 1$
- $r = 10\%$
- $\sigma = 20\%$
- barriera inferiore $L = 90$

Se lavoriamo in basse dimensioni (Tabella 4.3), 5 date di monitoraggio, notiamo che i metodi QMC convergono e i prezzi di Halton sono molto simili

N° simulazioni	MC	Halton	Faure	Sobol
10000	12.4217	12.5794	12.5793	12.6487
30000	12.6547	12.6179	12.6334	12.6397
50000	12.6452	12.6251	12.6386	12.6308
70000	12.5232	12.6283	12.6317	12.6379
90000	12.6291	12.6316	12.6431	12.6396
110000	12.6420	12.6331	12.6382	12.6404
130000	12.5730	12.6348	12.6426	12.6403
150000	12.6202	12.6346	12.6422	12.6376
170000	12.6544	12.6364	12.6397	12.6383
190000	12.5996	12.6375	12.6401	12.6392

Tabella 4.3: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 20\%$ $L = 90$ con 5 date di monitoraggio

a quelli di Sobol. Se però aumentiamo la dimensione questa convergenza sparisce (Tabella 4.4): al contrario delle opzioni Asiatiche, dove per 50 date di monitoraggio, le sequenze di Faure e Sobol si comportavano molto bene, per le Barriera le sequenze a bassa discrepanza sembrano convergere molto lentamente e in modo meno regolare. Questa irregolarità è data dal fatto che il payoff della simulazione è pari a 0 nel momento in cui la barriera viene toccata. Se infatti proviamo ad allontanare la barriera da S_0 ($L = 80$) otteniamo dei risultati migliori, Tabella 4.5, in quanto la barriera verrà toccata con minore probabilità.

N° simulazioni	MC	Halton	Faure	Sobol
10000	11.8144	10.2353	11.7555	11.5131
30000	11.7070	11.2855	11.8384	11.7384
50000	11.8604	11.5269	11.8282	11.7936
70000	11.8337	11.6052	11.8210	11.8096
90000	11.8861	11.6719	11.8224	11.8338
110000	11.7972	11.6973	11.8394	11.8402
130000	11.8326	11.7215	11.8498	11.8456
150000	11.8841	11.7412	11.8409	11.8454
170000	11.8307	11.7509	11.8418	11.8525
190000	11.7994	11.7638	11.8300	11.8546

Tabella 4.4: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 20\%$ $L = 90$ con 50 date di monitoraggio

N° simulazioni	MC	Halton	Faure	Sobol
10000	13.2519	11.5783	13.1255	13.0605
30000	13.2153	12.6463	13.2193	13.1416
50000	13.1751	12.8668	13.1823	13.1567
70000	13.3156	12.9488	13.1943	13.1697
90000	13.1869	13.0073	13.1998	13.1882
110000	13.1438	13.0321	13.2058	13.1933
130000	13.2045	13.0668	13.2121	13.1955
150000	13.1717	13.0857	13.2003	13.1889
170000	13.1455	13.0896	13.2009	13.1933
190000	13.2667	13.1071	13.1910	13.1967

Tabella 4.5: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 20\%$ $L = 80$ con 50 date di monitoraggio

4.3 Opzioni Lookback

Nelle opzioni Lookback il valore dello strike price è pari al massimo, S_{max} , o al minimo, S_{min} , (naturalmente dipende dal tipo di opzione) dei prezzi assunti dal sottostante nel periodo di validità dell'opzione stessa.

Se consideriamo T la scadenza, i payoff per una call e per una put sono:

$$C(S_T, T) = \max(S_T - S_{min}, 0)$$

$$P(S_T, T) = \max(S_{max} - S_T, 0)$$

Consideriamo

- $S_0 = 100$
- $K = 100$
- $T = 1$
- $r = 10\%$
- $\sigma = 30\%$

Come per le altre opzioni, i metodi QMC convergono per basse dimensioni (Tabella 4.6). Il valore a cui tendono è 24.39. Quando la dimensione aumenta, si veda la Tabella 4.7 e la Figura 4.2, Faure è in accordo con MC, mentre Sobol mostra una convergenza più lenta. Halton invece si dimostra ancora una volta inefficiente.

N° simulazioni	MC	Halton	Faure	Sobol
10000	24.3888	24.3888	24.4017	24.3888
30000	24.3787	24.3787	24.3851	24.3787
50000	24.3860	24.3860	24.3816	24.3860
70000	24.3895	24.3895	24.3808	24.3895
90000	24.3922	24.3922	24.3816	24.3922
110000	24.3916	24.3916	24.3845	24.3916
130000	24.3935	24.3935	24.3884	24.3935
150000	24.3875	24.3875	24.3915	24.3875
170000	24.3890	24.3890	24.3893	24.3890
190000	24.3890	24.3890	24.3901	24.3890

Tabella 4.6: Errori con $S_0 = 100$ $K = 100$ $T = 1$ $r = 10\%$ $\sigma = 20\%$ $L = 90$ con 10 date di monitoraggio

N° simulazioni	MC	Halton	Faure	Sobol
10000	28.6463	22.1350	28.0904	28.1104
30000	27.7717	24.8568	28.2389	28.0708
50000	28.2188	25.9250	28.1812	28.0549
70000	28.3151	26.4604	28.2391	28.0968
90000	28.1501	26.7051	28.1795	28.1002
110000	28.2673	27.0437	28.1786	28.1107
130000	28.1894	27.2990	28.1798	28.1178
150000	28.0858	27.3798	28.1831	28.1172
170000	28.1637	27.4040	28.1860	28.1331
190000	28.1785	27.5168	28.1725	28.1325

Tabella 4.7: Prezzo con 100 date di monitoraggio

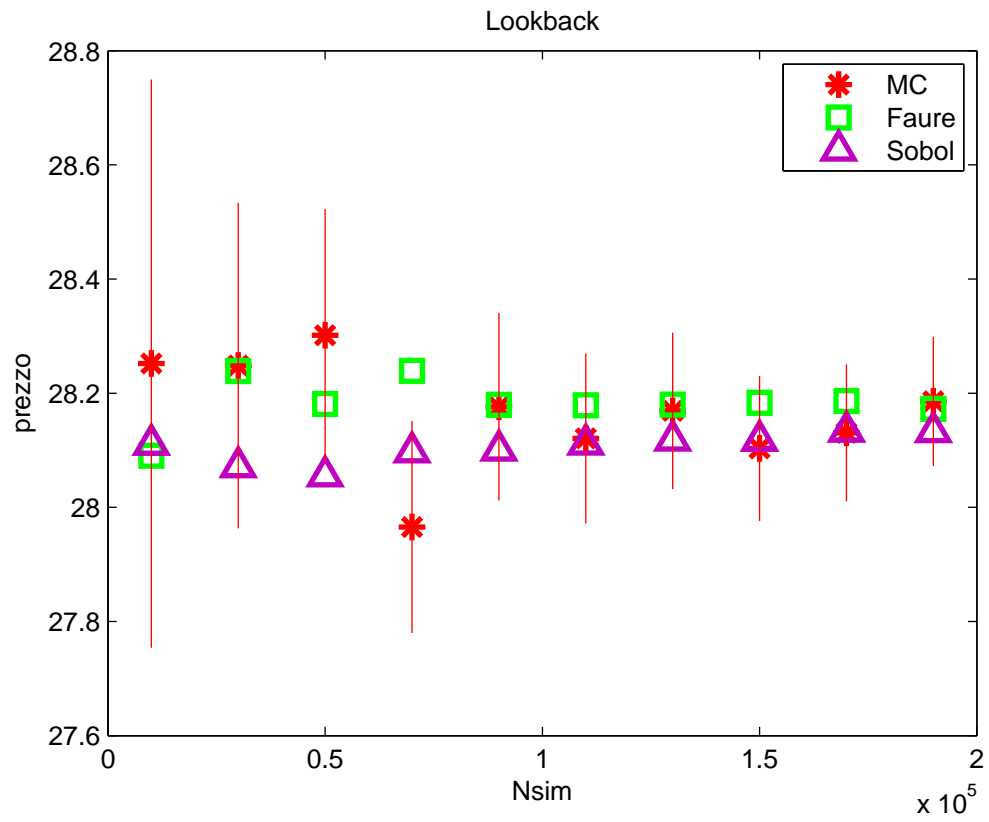


Figura 4.2: Prezzo Lookback con 100 date di monitoraggio

Capitolo 5

Pricing di opzioni Americane

Un'opzione Americana con scadenza T , può essere approssimata a un'opzione Bermuda in modo che l'esercizio anticipato sia possibile solo in determinate date $0 < \tau_1 < \tau_2 < \dots < \tau_M = T$ ($\Delta\tau = \frac{T}{M}$, $\tau_i = i\Delta\tau$). Il valore di un'opzione Americana, sotto l'ipotesi di neutralità al rischio è data da:

$$\text{prezzo} = \max_{\tau \leq T} E[e^{-r\tau} \Phi(K, S_\tau, \tau, \Psi_\tau)]$$

dove τ è lo *stopping time* e Φ è la funzione payoff definita come

$$\Phi_i(S) = \max(\phi(K, S), \Psi_i(S))$$

dove Ψ_i è il *valore di continuazione* al tempo τ_i , mentre $\phi(K, S)$ è il valore d'esercizio immediato, ovvero $\max(K - S, 0)$.

Ad ogni stopping time τ_i il possessore dell'opzione deve decidere se esercitare o no l'opzione. Essa viene esercitata se il valore di continuazione è minore rispetto al valore d'esercizio immediato.

Il valore di continuazione Ψ_i è definito come

$$\Psi_i(S) = e^{-r\Delta t} E[\Phi_{i+1}(S_{i+1}) | S_i = S]$$

[Longstaff & Schwartz (2001)] stimano il valore di continuazione con una regressione ai minimi quadrati scrivendo il valore di continuazione come una combinazione lineare di basi ortonormali, ad esempio utilizzando come $p_l(S)$ i polinomi di potenze (combinazioni di monomi), Laguerre, Hermite, Legendre o Chebyshev:

$$\Psi_i(S) = \sum_{l=0}^L \alpha_l p_l(S)$$

Per simulare il cammino di S viene utilizzata o la simulazione Monte Carlo o Quasi Monte Carlo.

I valori trovati con la regressione vengono poi comparati con i valori d'esercizio immediato e in questo modo si decide se esercitare l'opzione. Questa procedura viene ripetuta ricorsivamente andando indietro nel tempo. Dopo aver identificato i cash flows lungo la traiettoria del sottostante, si scontano fino a 0 e si trova così il prezzo dell'opzione Americana.

5.1 Scelta dei polinomi

[Moreno & Navas (2003)] analizzano l'impatto nell'utilizzare diversi tipi di polinomi e diversi gradi per i polinomi stessi. Una regressione lineare può essere interpretata come la proiezione della variabile dipendente sullo span generato dalle variabili indipendenti. Ogni polinomio sopra citato può essere espresso come combinazione lineare degli altri polinomi. Lo span generato quindi sarà il medesimo per ogni famiglia di polinomi e il prezzo dell'opzione sarà il medesimo.

Nella tabelle seguenti riportiamo i valori ottenuti prezzando un'opzione Americana con 50000 simulazioni, 10 date di monitoraggio e

- $S_0 = 40$
- $K = 40$
- $r = 6\%$
- $\sigma = 20\%$
- $T = 1$

gradi	2	3	4	5	6
Potenze	2.25093	2.278377	2.282211	2.281423	2.27976
Laguerre	2.25093	2.278377	2.282211	2.281423	2.27976
HermiteA	2.25093	2.278377	2.282211	2.281423	2.27976
HermiteB	2.25093	2.278377	2.282211	2.281423	2.27976
Legendre	2.25093	2.278377	2.282211	2.281423	2.27976
Chebyshev1	2.25093	2.278377	2.282211	2.281423	2.27976
Chebyshev2	2.25093	2.278377	2.282211	2.281423	2.27976

Tabella 5.1: *Monte Carlo*

In Tabella 5.1 i valori sono ottenuti con il metodo MC mentre in Tabella 5.2 utilizziamo la sequenza di Sobol. I medesimi risultati sono ottenuti con

gradi	2	3	4	5	6
Potenze	2.301978	2.329901	2.339714	2.339751	2.342086
Laguerre	2.301978	2.329901	2.339714	2.339751	2.342085
HermiteA	2.301978	2.329901	2.339714	2.339751	2.342086
HermiteB	2.301978	2.329901	2.339714	2.339751	2.342086
Legendre	2.301978	2.329901	2.339714	2.339751	2.342086
Chebyshev1	2.301978	2.329901	2.339714	2.339751	2.342086
Chebyshev2	2.301978	2.329901	2.339714	2.339751	2.342086

Tabella 5.2: *Sobol*

le sequenze di Faure e Halton.

Come ci aspettavamo il prezzo è il medesimo per tutti i polinomi, mentre se fissiamo il polinomio notiamo che i prezzi aumentano all'aumentare del numero di termini del polinomio. Nelle nostre simulazioni utilizzeremo il polinomio di Laguerre con tre termini, consigliato da [Longstaff & Schwartz (2001)].

5.2 Opzioni americane

Confrontiamo ora il metodo Monte Carlo con il metodo Quasi Monte Carlo. Vogliamo prezzare la seguente opzione:

- $S_0 = 40$
- $K = 40$
- $r = 6\%$
- $\sigma = 20\%$
- $T = 1$

I risultati in Tabella 5.3 e Figura 5.1 mostrano che tutte le sequenze a bassa discrepanza convergono al valore 2.267 con 5 date di monitoraggio. Passando a 10 dimensioni (Tabella 5.4) Faure e Sobol convergono al valore 2.291, mentre Halton converge più lentamente e sembra avere quindi un'accuratezza minore.

Aumentando la dimensione (Tabella 5.5) i metodi QMC (escluso Halton) hanno una convergenza meno accurata e meno regolare, probabilmente dovuta al deterioramento della qualità della sequenza. Tanto più che, è facile

intuire quale sarà il prezzo esatto con una accuratezza di una sola cifra decimale, o al massimo due. Il prezzo sarà infatti compreso fra 2.30 e 2.32 .

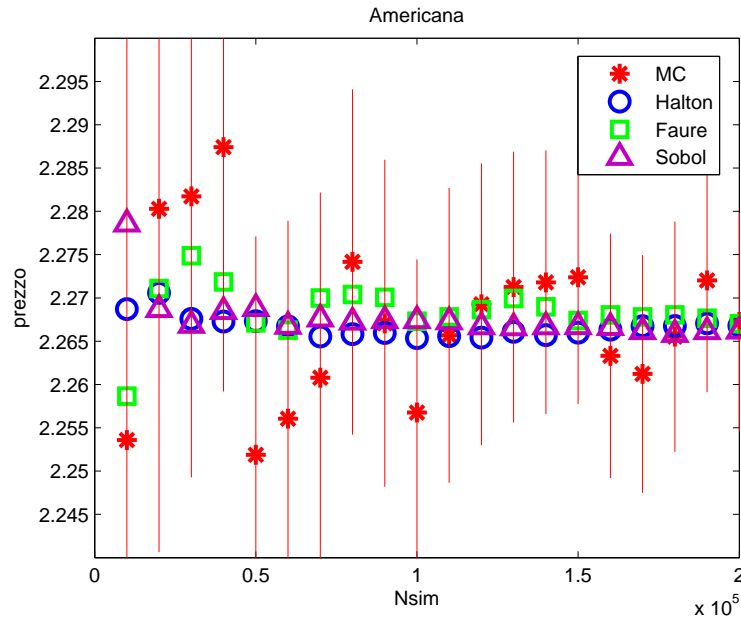
N° simulazioni	MC	Halton	Faure	Sobol
10000	2.2536	2.2687	2.2587	2.2786
30000	2.2817	2.2676	2.2749	2.2669
50000	2.2519	2.2673	2.2671	2.2688
70000	2.2608	2.2655	2.2700	2.2676
90000	2.2671	2.2660	2.2701	2.2674
110000	2.2657	2.2656	2.2678	2.2673
130000	2.2713	2.2661	2.2699	2.2666
150000	2.2724	2.2660	2.2674	2.2667
170000	2.2612	2.2667	2.2678	2.2661
190000	2.2720	2.2671	2.2676	2.2661

Tabella 5.3: Prezzo con 5 date di monitoraggio

N° simulazioni	MC	Halton	Faure	Sobol
10000	2.3062	2.3247	2.2840	2.2898
20000	2.3064	2.3078	2.2910	2.2906
30000	2.2755	2.2978	2.2892	2.2887
40000	2.3097	2.2939	2.2935	2.2867
50000	2.2854	2.2905	2.2923	2.2902
60000	2.2911	2.2873	2.2897	2.2904
70000	2.2965	2.2871	2.2894	2.2901
80000	2.2704	2.2867	2.2908	2.2909
90000	2.2858	2.2855	2.2906	2.2910
100000	2.2999	2.2862	2.2899	2.2909

Tabella 5.4: Prezzo con 10 date di monitoraggio

Figura 5.1: Prezzo Americana con 5 date di monitoraggio



N° simulazioni	MC	Halton	Faure	Sobol
10000	2.3234	2.3999	2.2895	2.3653
20000	2.2766	2.3508	2.2960	2.3379
30000	2.3107	2.3331	2.2998	2.3433
40000	2.3087	2.3339	2.2954	2.3331
50000	2.2836	2.3255	2.2994	2.3299
60000	2.3101	2.3260	2.3029	2.3265
70000	2.2943	2.3199	2.3078	2.3263
80000	2.3001	2.3180	2.3070	2.3221
90000	2.2951	2.3134	2.3068	2.3192
100000	2.3151	2.3121	2.3065	2.3191

Tabella 5.5: Prezzo con 50 date di monitoraggio

Conclusioni

I metodi Quasi Monte Carlo forniscono un modo per migliorare l'accuratezza e l'affidabilità delle simulazioni Monte Carlo.

L'idea base è di utilizzare delle sequenze deterministiche al posto delle sequenze pseudo-casuali.

In questa tesi abbiamo illustrato le proprietà dell'approccio Quasi Monte Carlo e abbiamo confrontato i due metodi valutando diversi tipi di opzioni. Le sequenze a bassa discrepanza si sono comportate in modo differente a seconda del tipo di opzione esaminata e la corrispondente complessità del problema che le corrisponde.

La sequenza di Halton in generale è dominata dalle sequenze di Faure e Sobol. Nonostante l'implementazione del metodo di Halton sia più semplice rispetto alle altre due sequenze, non è consigliato usarlo in problemi di grandi dimensioni. La sequenza di Sobol invece appare più resistente alla degradazione delle alte dimensioni.

In primo luogo sembra si possa affermare che per problemi caratterizzati da una dimensione non particolarmente elevata, cioè con un numero di date di monitoraggio non elevato, le successioni a bassa discrepanza consentono di ottenere degli errori migliori rispetto al metodo Monte Carlo ordinario. Con l'aumentare della dimensione del problema questa miglior efficienza diminuisce tanto che da un certo punto in poi non si può affermare con certezza quale sia il metodo da preferire.

Nel caso di opzioni europee, con un numero di date di monitoraggio non elevato, fino a 10, gli errori forniti dal metodo Monte Carlo sono maggiori rispetto alle sequenze a bassa discrepanza. Tra i metodi Quasi Monte Carlo la sequenza di Sobol è quella con l'errore minore. Con l'aumentare della dimensione la sequenza di Halton diventa inaffidabile. Fino a 100 dimensioni Faure fornisce errori minori rispetto a Sobol; per dimensioni elevate invece, superiori a 170, Sobol appare essere il migliore.

La stessa situazione si ripete nel caso di opzioni asiatiche: per basse dimensio-

ni si nota una convergenza maggiore di tutte le sequenze a bassa discrepanza; per dimensioni maggiori, 50 date di monitoraggio, solo Faure e Sobol mantengono errori inferiori al metodo Monte Carlo.

Diversi sono i casi delle opzioni barriera e delle opzioni lookback: la convergenza non è più regolare e veloce come negli altri casi. Per queste due opzioni i metodi Quasi Monte Carlo sono molto affidabili solo per basse dimensioni. Probabilmente ciò è dovuto al fatto che in entrambi i casi il prezzaggio si basa sul valore minimo (o massimo) della sequenza (nelle lookback per il payoff, nelle barriera per vedere se la barriera è toccata). Questa ricerca di valore massimo o minimo sembra pertanto avere bisogno di un campionamento preciso, che i metodi Quasi Monte Carlo ad alte dimensioni sembrano non riuscire a dare.

Se applichiamo le sequenze a bassa discrepanza all'algoritmo di Longstaff e Schwartz per il prezzaggio di opzioni americane, troviamo le medesime situazioni riscontrate negli altri tipi di opzioni citati precedentemente. Per dimensioni basse tutte le sequenze a bassa discrepanza sono da preferirsi rispetto al metodo Monte Carlo. A partire da 10 date di monitoraggio Halton diventa inaffidabile.

Concludendo possiamo affermare che le sequenze a bassa discrepanza sono migliori rispetto al metodo Monte Carlo nel caso di monitoraggio discreto se il numero delle date di monitoraggio, fissate dal contratto che si deve prezzare, non è eccessivo. Nel caso di monitoraggio continuo invece, dove si deve far tendere le date di monitoraggio (e quindi la dimensione) a $+\infty$ i metodi QMC non sembrano indicati.

Appendice A

Codici

A.1 Sequenze a bassa discepanza

Dati in ingresso un numero k e una base b , la funzione B_ARY restituisce un vettore costituito dai coefficienti dell'espansione in base b del numero k . Ad esempio $B_ARY(5, 2) = [1, 0, 1]$.

```
function a=B_ARY(k,b)
if (k>0)
    jmax=floor(log(k)/log(b));
    a=zeros(1,jmax+1);
    q=b^jmax;
    for j=1:jmax+1
        a(j)=floor(k/q);
        k=k-q*a(j);
        q=q/b;
    end
end
```

Dopo aver scritto il numero k in base b , per calcolare i coefficienti dell'espansione sempre in base b del numero $k+1$, si utilizza la funzione $NEXTB_ARY$. La funzione riceve in ingresso il vettore ain dei coefficienti del numero k e restituisce il vettore $aout$. Ad esempio $NEXTB_ARY([1, 0, 1], 2) = [1, 1, 0]$.

```
function aout=NEXTB_ARY(ain,b)
m=length(ain);
carry=1;
aout=zeros(m,1);
```

```

for i=1:m
    if (ain(i)~=b-1)
        aout(i)=ain(i)+1;
        carry=0;
        aout(i+1:end)=ain(i+1:end);
        break;
    end
end
if (carry==1)
    aout=cat(1,aout,1);
end

```

A.1.1 Sequenza di Halton

La funzione *Halton* genera N punti della sequenza di Halton in d dimensioni. In ingresso viene passato anche il seme della sequenza n_0 . Come base per la generica dimensione n viene utilizzato l' n -esimo numero primo. Se vogliamo una sequenza in 5 dimensioni per esempio, le basi che verranno utilizzate per ogni singola dimensione sono 2, 3, 5, 7 e 11. Per ogni dimensione viene costruita la sequenza di Van der Corput, tramite l'omonima funzione, passando come base il numero primo corrispondente a quella dimensione.

```

function P=halton(n0,N,d)
P=zeros(N,d);
base=zeros(1,d);
j=2;
for ii=1:d
    b=primo(j);
    j=b+1;
    base(ii)=b;
    P(:,ii)=Van_der_Corput(n0,N,base(ii));
end

function P=Van_der_Corput(n0,N,b)
nmax=n0+N-1;
rmax=1+floor(log(nmax)/log(b));
P=zeros(N,1);
r=1+floor(log(max([1 n0-1]))/log(b));
qnext=b^r;

```

```

a=fliplr(B_ARY(n0-1,b));
bpwrs=1./b.^(1:rmax);
bpwrs_r=bpwrs(1:r);
for k=n0:nmax
    a=NEXTB_ARY(a,b);
    if (k==qnext)
        r=r+1;
        bpwrs_r=bpwrs(1:r);
        qnext=b*qnext;
    end
    P(k-n0+1)=bpwrs_r*a;
end

```

A.1.2 Sequenza di Faure

Per generare N punti d dimensionali della sequenza di Faure in base b , utilizziamo la funzione *Faure_pts*. Il punto iniziale della sequenza è n_0 . Come base b si sceglie il più piccolo numero primo maggiore o uguale a N . La creazione delle matrici di permutazione (per riordinare la sequenza all'interno di ogni dimensione) viene effettuata tramite *FaureMat*.

```

function P=Faure_pts(n0,N,d,b)
nmax=n0+N-1;
rmax=1+floor(log(nmax)/log(b));
P=zeros(N,d);
r=1+floor(log(max([1 n0-1]))/log(b));
qnext=b^r;
a=fliplr(B_ARY(n0-1,b));
bpwrs=1./b.^(1:rmax);
C=diag(ones(rmax,1));
MC=zeros(rmax*(d-1),rmax);
for i=1:d-1
    MC((i-1)*rmax+1:i*rmax,:)=FaureMat(rmax,i,C);
end
bpwrs_r=bpwrs(1:r);
for k=n0:nmax
    a=NEXTB_ARY(a,b);
    if (k==qnext)
        r=r+1;
        bpwrs_r=bpwrs(1:r);
    end
end

```

```

        qnext=b*qnext;
    end
    P(k-n0+1,1)=sum(bpwr_r*a);
    nn=rmax-r;
    My=MC*[a;zeros(nn,1)];
    My=mod(My,b);
    P(k-n0+1,2:d)=[bpwr_r,zeros(1,nn)]*reshape(My,rmax,d-1);
end

```

```

function C=FaureMat(r,i,C)
for m=2:r
    C(1,m)=i*C(1,m-1);
end
for n=3:r
    for m=2:n-1
        C(m,n)=C(m-1,n-1)+i*C(m,n-1);
    end
end
end

```

A.1.3 Sequenza di Sobol

Per generare N punti d dimensionali della sequenza di Sobol, utilizziamo la funzione `wsobol_V`.

Per i polinomi primitivi e i primi numeri direzionali (`SobolData`) abbiamo usato quelli forniti dal sito

<http://web.maths.unsw.edu.au/fkuo/sobol/index.html>.

```

function SobD=wsobol_V(N,d)
N=N-1;
N_of_directions=ceil(log2(N+1));
load soboldata
SobD=zeros([N+1,d]);
m=ones([1,N_of_directions]);
for i=2:d
    poly=wdec2bas(SobolData(i-1,4),2);
    end_dir=find(SobolData(i-1,6:end)==0);
    if isempty(end_dir)
        temp=dirnums(poly,N_of_directions,SobolData(i-1,5:end));
        m(i,1:length(temp))=temp;
    else

```

```

        temp=dirnums(poly , N_of_directions , ...
        SobolData ( i -1,5:(5+end_dir (1) ) ) );
        m( i , 1:length ( temp) )=temp;
    end
end
SobD=sobolgenerator_V(N,m);

```

La funzione *wdec2bas* converte il numero decimale n in un vettore dei coefficienti in base r . Per la sequenza di Sobol verrà usata solamente la base 2.

```

function y=wdec2bas(n,r)
i=1;
y(1)=0;
while n>0
    y(i)=rem(n,r);
    n=floor(n/r);
    i=i+1;
end;

```

Conoscendo le direzioni m , i successivi k numeri direzionali vengono calcolati tramite *dirnums*.

```

function m=dirnums(poly,k,m)
d=length(poly)-1;
poly=poly(2:end-1);
for i=d+1:k
    m(i)=bitxor(2^d*m(i-d),m(i-d));
    for j=1:(d-1)
        m(i)=bitxor(2^j*poly(j)*m(i-j),m(i));
    end;
end;

```

La funzione *sobolgenerator_V*(n,m) genera i punti della sequenza $x^{1,j}, x^{2,j}, \dots$ tramite

$$x^{n,j} = b_1 v_{1,j} \oplus b_2 v_{2,j} \oplus \dots$$

dove $\dots b_3 b_2 b_1$ è la rappresentazione binaria di n e $v_{i,j} = \frac{m_{i,j}}{2^i}$.

```

function x=sobolgenerator_V(n,m)
N=30;

```



```

x=zeros(n+1,size(m,1));
for i=1:n
    c=1;
    dived=i-1;
    remain=mod(dived,2);
    while remain==1
        dived=floor(dived/2);
        remain=mod(dived,2);
        c=c+1;
    end;
    x(i+1,:)=bitxor(floor(2^N*x(i,:)),floor(2^(N-c)*...
        (transpose(m(1:size(x,2),c)))))/2^N;
end;

```

A.2 Algoritmo di Moro

Algoritmo di Moro per l'approssimazione dell'inversa della normale, si veda [Moro (1995)].

Le costanti che vengono utilizzate nell'algoritmo sono:

			c=	0.3374754822726147
				0.9761690190917186
				0.1607979714918209
a=	2.50662823884	b=	-8.47351093090	0.0276438810333863
	-18.61500062529		23.08336743743	0.0038405729373609
	41.39119773534		-21.06224101826	0.0003951896511919
	-25.44106049637		3.13082909833	0.0000321767881768
				0.0000002888167364
				0.0000003960315187

```

function r=moro(u)
x=u-0.5;
index=find(abs(x)<0.42);
r=zeros(size(x,1),size(x,2));
r(index)=x(index).^2;
r(index)=x(index).*(((a(4).*r(index)+a(3)).*r(index)+a(2)).*...
    r(index)+a(1))./((((b(4).*r(index)+b(3)).*...
    r(index)+b(2)).*r(index)+b(1)).*r(index)+1));
index2=find(abs(x)>=0.42);
r(index2)=u(index2);
index2a=find(abs(x)>=0.42 & x>0);

```

```

r(index2a)=1-u(index2a);
r(index2)=log(-log(r(index2)));
r(index2)=c(1)+r(index2).*(c(2)+r(index2).*(c(3)+r(index2).*...
(c(4)+r(index2).*(c(5)+r(index2).*(c(6)+r(index2).*...
(c(7)+r(index2).*(c(8)+r(index2).*c(9))))));
index2b=find(abs(x)>=0.42 & x<0);
r(index2b)=-r(index2b);

```

A.3 Pricing di opzioni

Per simulare il sottostante con

$$S(t_n) = S(t_{n-1})e^{(r-\frac{\sigma^2}{2})dt+\sigma\epsilon_n\sqrt{dt}}$$

utilizziamo la funzione *Asset*, dove *Path(:,i)* è $S(t_{n-1})$ e *Path(:,i+1)* è $S(t_n)$. Il parametro *MC_Halton_Faure_Sobol* indica quale sequenza vogliamo usare. Se si pone pari ad 0, viene utilizzata la sequenza pseudo-casuale altrimenti le sequenze di Halton (1), Faure (2) o Sobol (3). *Nstep* è il numero delle date di monitoraggio mentre *Nsim* è il numero di simulazioni. Dopo aver generato le sequenze si calcola l'inversa della normale e si ottiene ϵ_n .

```

function Path=Asset(S0,r,sigma,T,Nstep,Nsim,MC_Halton_Faure_Sobol)
Path=zeros(Nsim,Nstep+1);
Path(:,1)=S0;
dt=T/Nstep;
mu=r-sigma^2/2;
if (MC_Halton_Faure_Sobol==0)
    epsilon=randn(Nsim,Nstep);
elseif (MC_Halton_Faure_Sobol==1)
    n0=2;
    A=halton(n0,Nsim,Nstep);
    epsilon=icdf('norm',A,0,1);
elseif (MC_Halton_Faure_Sobol==2)
    base=primo(Nstep);
    n0=base^4-1;
    A=Faure_pts(n0,Nsim,Nstep,base);
    epsilon=icdf('norm',A,0,1);
else
    B=wsobol_V(Nsim+1,Nstep);
    A=B(2:end,:);

```

```

    epsilon=icdf('norm',A,0,1);
end
for i=1:Nstep
    Path(:,i+1)=Path(:,i).*exp(mu*dt+sigma*sqrt(dt)*epsilon(:,i));
end

```

A questo punto possiamo passare al prezzaggio delle opzioni. Una volta generato il sottostante tramite la funzione *Asset*

$$S = \text{Asset}(S_0, r, \text{sigma}, T, N\text{step}, N\text{sim}, MC_{\text{HaltonFaureSobol}});$$

si calcola il payoff attualizzato dell'opzione per ogni simulazione, con K prezzo d'esercizio; il prezzo è la media dei payoff ottenuti.

Il parametro *flag* posto pari ad 1 corrisponde ad un'opzione call, -1 ad una put.

Nel caso di opzioni **Europee**:

```

    payoff=max(flag.*(S(:,end)-K),0);
    Prezzo=mean(exp(-r*T).*payoff);

```

Per le opzioni **Asiatiche**:

```

for i=1:Nsim
    payoff(i)=exp(-r*T)*max(flag*(mean(S(i,1:end))-K),0);
end
Prezzo=mean(payoff);

```

Per le opzioni **Barriera**, dove L è la barriera inferiore e U quella superiore:

```

for i=1:Nsim
    touch=0;
    if L>0
        touch=touch+sum(S(i,:)<=L);
    end
    if U>0
        touch=touch+sum(S(i,*)>=U);
    end
end

```

```

    if touch==0
        payoff(i)=max((S(i,end)-K)*flag,0);
    else
        payoff(i)=0;
    end
end
Prezzo=mean(exp(-r*T).*payoff);

```

Per le opzioni **Lookback**:

```

if flag==1
    payoff=max(max(S')-K,0);
else
    payoff=max(K-min(S'),0);
end
Prezzo=mean(exp(-r*T).*payoff);

```

Per apprezzare le opzioni **Americane** abbiamo utilizzato l'algoritmo di Longstaff e Schwartz. C è il valore di continuazione definito nel Capitolo 5 mentre CF è il cash flow.

```

dt = T/Nstep;
ExTime = (Nsim+1)*ones(Nstep,1);
CF = zeros(size(S));
CF(end,:) = max((S(end,:) - K). * flag, 0);
for ii = size(S)-1:-1:2
    if flag==-1
        Idx=find(S(ii,:) < K);
    else
        Idx=find(S(ii,:) > K);
    end
    X=S(ii,Idx)';
    X1=X/S0;
    Y=CF(ii+1,Idx)'.*exp(-r*dt);
    R=[ones(size(X1)) (1-X1) 1/2*(2-4*X1-X1.^2)];
    a=R\Y;
    C=R*a;
    Jdx=max((X-K). * flag, 0) > C;
    nIdx=setdiff((1:Nsim), Idx(Jdx));

```

```
CF(ii,Idx(Jdx))=max(K-X(Jdx),0);
ExTime(Idx(Jdx))=ii;
CF(ii,nIdx)=CF(ii+1,nIdx);
CF(ii,nIdx)=exp(-r*dt)*CF(ii+1,nIdx);
end
Prezzo=mean((CF(2,:))*exp(-r*dt));
```

Bibliografia

- [1] Areal N., Rodrigues A., Armada M. (2008). On improving the least squares Monte Carlo option valuation method. *Review of Derivatives Research*, 11 pp. 119-151.
- [2] Boyle P. (1977). Options: A Monte Carlo Approach. *Journal of Financial Economics*, 4 pp. 323-338.
- [3] Boyle P., Broadie M., Glasserman P. (1997). Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8) pp. 1267-1321.
- [4] Bratley P., Fox B. (1988). ALGORITHM 659: Implementing Sobol's Quasi Random Sequence Generator. *ACM Transaction on Mathematical Software*, 14(1) pp. 88-100.
- [5] Bratley P., Fox B., Niederreiter H. (1992). Implementation and tests of low-discrepancy sequences. *ACM Transactions on Modeling and Computer Simulation*, 2(3) pp. 195-213.
- [6] Broadie M., Detemple J. (1996). American option valuation: New bounds, approximation, and a comparison of existing methods. *Review of Financial Studies*, 9(4) pp. 1211-1250.
- [7] Faure H. (1982). Discr pance de suites associ es   un syst me de num ration (en dimension s). *Acta Arithmetica*, 41(4) pp. 337-351.
- [8] Fox P. (1986). ALGORITHM 647: Implementation and Relative Efficiency of Quasi Random Sequence Generators. *ACM Transaction on Mathematical Software*, 12(4) pp. 362-376.
- [9] Friedel I., Keller A. (2002). Fast generation of randomized low-discrepancy point sets. In: K.-T. Fang, F.J. Hickernell and H. Niederreiter, Editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, Springer, Berlin, pp. 257-273.

- [10] Galanti S., Jung A. (1997). Low-discrepancy sequences: Monte Carlo simulation of option prices. *The Journal of Derivates*, pp. 63-83.
- [11] Gamba A., (2003). Real options valuation: A Monte Carlo approach. *University of Verona*.
- [12] Glasserman P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer, New York.
- [13] Giles M., Kuo F., Sloan I., WaterHouse B. (2008). Quasi Monte Carlo for finance applications. *The ANZIAM Journal*, 50 pp. C308-C323.
- [14] Halton J. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimentional integrals. *Numerische Mathematik*, 2 pp. 84-90.
- [15] Joe S., Kuo F. (2003). Remark on Algorithm 647: Implementing Sobol's quasi-random sequence generator. *ACM Transaction on Mathematical Software*, 29 pp. 49-57.
- [16] Joy C., Boyle P., Tan K. (1996). Quasi-Monte Carlo methods in Numerical Finance. *Management Science*, 42(6) pp. 926-938.
- [17] L'Ecuyer P. (2003). Quasi-Monte Carlo methods for simulation. *Proceedings of the 2003 Winter Simulation Conference*, pp. 81-89.
- [18] L'Ecuyer P. (2004). Quasi-Monte Carlo in finance. *Proceedings of the 2004 Winter Simulation Conference*, pp. 1645-1655 .
- [19] Longstaff F., Schwartz E. (2001). Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14(1) pp. 113-147.
- [20] Moreno M., Navas J. (2003). On the robustness of least-squares Monte Carlo (LSM) for pricing American derivatives. *Review of Derivatitves Research*, 6(2) pp. 107-128.
- [21] Moro B. (1995). The full Monte. *Risk*, 8(2) pp. 57-58.
- [22] Niederreiter H. (1987). Point Sets and Sequences with Small Discrepancy. *Monatshefte für Mathematik*, 104 pp. 273-337.
- [23] Niederreiter H. (1988). Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30(1) pp. 51-70.

- [24] Papageoriou A. (2002). The brownian bridge does not offer a consistent advantage in quasi Monte Carlo integration. *Journal of Complexity*, 18(1) pp. 171-186.
- [25] Papageorgiou A., Traub J. (1996), Beating Monte Carlo, *Risk*, 9(6) pp. 63-65.
- [26] Press W., Teukolsky S., Vetterling W., Flannery B. (1992). *Numerical recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge.
- [27] Sloan I. (2010). Integration and Approximation in High Dimensions - a Tutorial. *Uncertainty Quantification*, Edinburgh.
- [28] Sobol I. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4) pp. 86-112.
- [29] Sobol I. (1979). On the Systematic Search in a Hypercube. *SIAM Journal on Numerical Analysis*, 16(5) pp. 790-793.
- [30] Sobol I. (1998). On quasi-Monte Carlo integrations. *Mathematics and Computers in Simulation*, 47(2) pp. 103-112.
- [31] Tian T., Burrage K. (2003). Accuracy issues of Monte Carlo methods for valuing American options. *The ANZIAM Journal*, 44(E) pp. C739-C758.
- [32] Turnbull S. (1992). Average Intelligence. *Risk*, 5(2) pp. 53-59.