

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

SCUOLA DI SCIENZE  
Corso di Laurea in Matematica

**MOTI BROWNIANI:  
SIMULAZIONI E  
APPLICAZIONI ALLA FINANZA**

Tesi di Laurea in Probabilità e Statistica Matematica

Relatore:  
Chiar.mo Prof.  
ANDREA PASCUCCI

Presentata da:  
MATTEO MURATORI

Anno Accademico 2022-2023



# Introduzione

Il moto Browniano è una categoria di processi stocastici che trova fondamentali applicazioni in matematica finanziaria. Il termine "Browniano" deriva dal nome del botanico britannico Robert Brown, il quale, nell'ottocento, fu il primo ad osservare un moto di questo tipo. Infatti egli notò come alcune particelle di polline in sospensione nell'acqua si muovessero continuamente, percorrendo traiettorie irregolari. Questi movimenti vennero studiati anche da Albert Einstein, che nel 1905 riuscì a formalizzare il modello da un punto di vista teorico. Il primo a dare una definizione rigorosa di moto Browniano, invece, fu Norbert Wiener nel 1923.

Nel primo capitolo di questo elaborato vedremo le principali caratteristiche dei processi stocastici che vengono classificati come moti Browniani, distinguendo tra le varie categorie. Di pari passo analizzeremo alcuni metodi e algoritmi per simulare tali processi, sia nello spazio monodimensionale che in spazi di dimensione superiore. Il secondo capitolo sarà invece dedicato ai derivati finanziari, e in particolare alla valutazione del payoff di alcune tipologie di derivati. Per adempiere a questo compito applicheremo quanto appreso nel capitolo precedente riguardo alla simulazione dei moti Browniani e ci avvarremo del metodo Monte Carlo, con l'obiettivo di calcolare il valore atteso dei payoff a cui siamo interessati.

L'ambiente di sviluppo di cui faremo uso per l'implementazione di questi modelli sarà Wolfram Mathematica, che ci consentirà di eseguire velocemente i calcoli necessari alla realizzazione dei nostri scopi e di rappresentare alcuni risultati ottenuti in dei grafici, rendendo il tutto più visivamente intuitivo.



# Indice

<b>Introduzione</b>	<b>i</b>
<b>1 Moto Browniano</b>	<b>1</b>
1.1 Moto Browniano standard . . . . .	1
1.2 Moto Browniano generalizzato . . . . .	5
1.3 Moto Browniano multidimensionale . . . . .	9
1.4 Moto Browniano geometrico . . . . .	14
<b>2 Derivati finanziari</b>	<b>17</b>
2.1 Principali tipologie di derivati finanziari . . . . .	17
2.2 Metodo Monte Carlo . . . . .	21
2.3 Valutazione del payoff di derivati . . . . .	25
2.4 Esempi numerici . . . . .	27
<b>Bibliografia</b>	<b>33</b>



# Elenco delle figure

1.1	Grafico moto Browniano standard . . . . .	2
1.2	Grafico moto Browniano generalizzato . . . . .	6
1.3	Grafico moto Browniano multidimensionale . . . . .	10
1.4	Grafico moto Browniano geometrico . . . . .	15
2.1	Integrale col metodo Monte Carlo . . . . .	25
2.2	Payoff opzione con barriera . . . . .	30



# Capitolo 1

## Moto Browniano

Di seguito tratteremo alcune delle più importanti tipologie di moti Browniani, e in particolare inizieremo dal moto Browniano standard e dal moto Browniano generalizzato, vedendo entrambi prima nel caso monodimensionale e poi in quello multidimensionale. Concluderemo con il moto Browniano geometrico, che si rivelerà la tipologia più importante ai fini delle applicazioni alla finanza. Per ognuna delle categorie in questione alterneremo una breve descrizione teorica allo studio degli algoritmi che ci permettono di simulare le traiettorie di eventuali corpi che seguono moti di questo tipo. Ovviamente, trattandosi di processi stocastici a tempo continuo, non è possibile nella pratica simulare l'intera traiettoria. Per questo ci limiteremo a discretizzare l'intervallo di tempo in questione ad un sottinsieme finito di istanti, e a lavorare soltanto su tali punti.

### 1.1 Moto Browniano standard

**Definizione 1.1.** *Diciamo che un processo stocastico  $\{W(t), 0 \leq t \leq T\}$  è un **moto Browniano standard** se valgono le seguenti condizioni:*

1.  $W(0) = 0$ ;
2. *la funzione  $t \rightarrow W(t)$  è continua su  $[0, T]$  quasi certamente;*

3. per ogni  $k \in \mathbb{N}$  e per ogni scelta di  $\{t_1, t_2, \dots, t_k\} \subset ]0, T[$  le variabili aleatorie  $\{W(t_1) - W(0), W(t_2) - W(t_1), \dots, W(t_k) - W(t_{k-1}), W(T) - W(t_k)\}$  sono indipendenti;
4.  $W(t) - W(s) \sim N(0, t - s)$  per ogni  $0 \leq s < t \leq T$ .

Consideriamo il processo  $\{W(t), 0 \leq t \leq T\}$  moto Browniano standard. Dalla definizione 1.1 è immediato dedurre la legge della variabile aleatoria  $W(t)$  con  $t \in [0, T]$ , infatti, siccome  $W(t) = W(t) - W(0)$ , segue che:

$$W(t) \sim N(0, t). \quad (1.1)$$

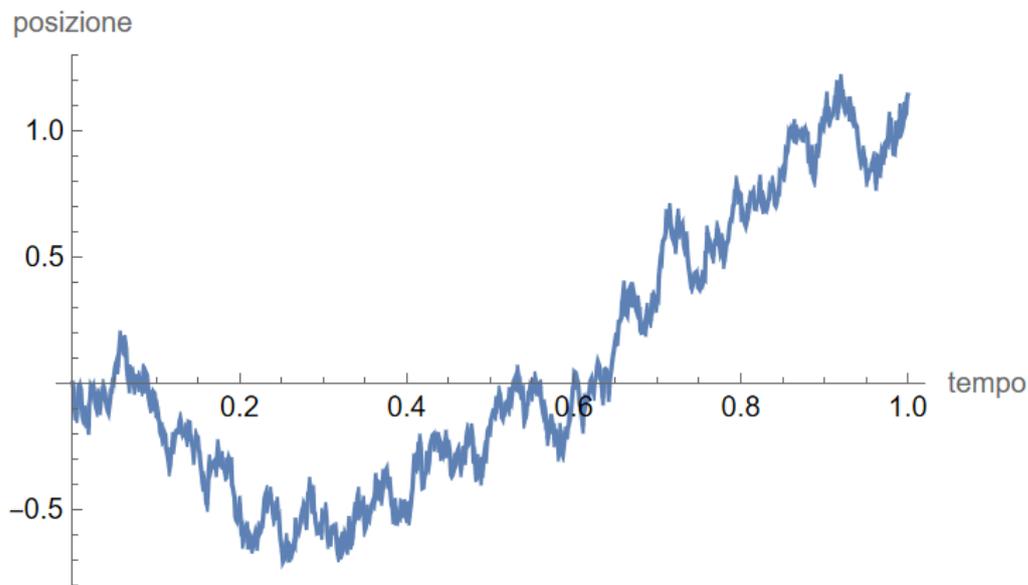


Figura 1.1: Grafico di un moto Browniano standard

Ci sono due principali strategie che ci permettono di simulare un moto Browniano standard: la *Random Walk Construction* e la *Brownian Bridge Construction*.

La **Random Walk Construction** si implementa attraverso una ricorrenza che si può ricavare dalla definizione 1.1 in maniera diretta. Infatti, si

procede nel seguente modo: sia  $W$  un moto Browniano standard, supponiamo di essere interessati ai valori  $W(t_1), W(t_2), \dots, W(t_k)$ , simulazioni di  $W$ , rispettivamente, agli istanti  $t_1, t_2, \dots, t_k$  fissati. Definiamo, inoltre,  $t_0 := 0$  e  $Z_1, Z_2, \dots, Z_k$  simulazioni indipendenti di una variabile aleatoria normale standard. Ovviamente, non è necessario cercare il valore di  $W(t_0)$  in quanto, essendo  $W(t)$  un moto Browniano standard, esso avrà valore iniziale nullo certamente. Affinchè rispettino quanto richiesto dalla definizione 1.1, è sufficiente imporre che le  $W(t_1) - W(t_0), W(t_2) - W(t_1), \dots, W(t_k) - W(t_{k-1})$  siano indipendenti, e che  $W(t_{i+1}) - W(t_i) \sim N(0, t_{i+1} - t_i)$  per ogni  $i \in \{0, 1, \dots, k-1\}$ . Grazie a ciò che sappiamo sulla trasformazione lineare di variabili aleatorie con legge normale possiamo affermare che i numeri  $\sqrt{t_{i+1} - t_i} Z_{i+1}$  con  $i = \{0, 1, \dots, k-1\}$  sono proprio simulazioni di variabili aleatorie indipendenti, aventi legge, rispettivamente,  $N(0, t_1), N(0, t_2 - t_1), \dots, N(0, t_k - t_{k-1})$ . Quindi, ponendo  $W(t_{i+1}) - W(t_i) = \sqrt{t_{i+1} - t_i} Z_{i+1}$  per ogni  $i \in \{0, 1, \dots, k-1\}$ , siamo riusciti a descrivere un metodo per ottenere le  $W(t_1) - W(0), W(t_2) - W(t_1), \dots, W(t_k) - W(t_{k-1})$ , ovvero l'incremento del valore del processo ad ogni step, con le caratteristiche richieste. Da questo deduciamo che, grazie alla ricorsione

$$W(t_{i+1}) = W(t_i) + \sqrt{t_{i+1} - t_i} Z_{i+1}, \quad i = \{0, 1, \dots, k-1\}, \quad (1.2)$$

si calcolano tutti i valori posti come obiettivo, ottenendo la simulazione del processo stocastico a cui si aspirava.

Come si può evincere osservando la ricorsione (1.2), la Random Walk Construction calcola i valori cercati in ordine cronologico: dal primo all'ultimo istante. Questo non è sempre un vantaggio, in quanto una delle più importanti caratteristiche di un moto Browniano è proprio la posizione finale, e talvolta, in caso di simulazioni particolarmente lunghe, la Random Walk Construction non ci consente di ottenere questo dato in tempi sufficientemente brevi. Per questo motivo un altro metodo altrettanto valido per arrivare allo stesso risultato è la **Brownian Bridge Construction**. Tale strategia consiste nel trovare innanzitutto il valore finale del processo e, successivamente, rendere più fine la simulazione, occupandosi dei punti

intermedi del moto. In particolare, denotando con  $t_1, t_2, \dots, t_k$  gli istanti di tempo ai quali si è interessati, si comincia calcolando  $W(t_k)$ , per poi proseguire con  $W(t_{\lfloor \frac{k}{2} \rfloor})$ , dopodichè con  $W(t_{\lfloor \frac{k}{4} \rfloor})$  e  $W(t_{\lfloor \frac{3k}{4} \rfloor})$  e così via, prendendo sempre il punto medio dell'intervallo di valori non ancora trovati, fino a completamento della simulazione. Quando si procede, come in questo caso, non in ordine cronologico, occorre ricordarsi di condizionare rispetto ai valori già ottenuti. In particolare se si è a conoscenza del fatto che  $W(s_1) = x_1, W(s_2) = x_2, \dots, W(s_k) = x_k$ , e si è interessati a  $W(s)$  con  $s_1 < \dots < s_i < s < s_{i+1} < \dots < s_n$ , bisogna utilizzare la legge di  $W(s)$  condizionata agli eventi  $W(s_1) = x_1, W(s_2) = x_2 \dots W(s_k) = x_k$ , denotata con  $\mu_{(W(s)|W(s_1)=x_1, W(s_2)=x_2, \dots, W(s_k)=x_k)}$ . Fortunatamente, le cose si semplificano abbondantemente grazie alla *proprietà di Markov*, di cui i moti Browniani godono. Infatti, questa ci consente di affermare che la legge di  $W(s)$  condizionata a  $W(s_1) = x_1, W(s_2) = x_2, \dots, W(s_k) = x_k$  coincide con la legge di  $W(s)$  condizionata *soltanto* a  $W(s_i) = x_i, W(s_{i+1}) = x_{i+1}$ , ovvero i valori di cui si è già conoscenza più vicini a  $s$ . Segue che  $\mu_{(W(s)|W(s_1)=x_1, W(s_2)=x_2, \dots, W(s_k)=x_k)} = N\left(\frac{(s_{i+1} - s)x_i + (s - s_i)x_{i+1}}{s_{i+1} - s_i}, \frac{(s_{i+1} - s)(s - s_i)}{(s_{i+1} - s_i)}\right)$ , per cui  $W(s)$  può essere ottenuto con la formula:

$$W(s) = \frac{(s_{i+1} - s)x_i + (s - s_i)x_{i+1}}{s_{i+1} - s_i} + \sqrt{\frac{(s_{i+1} - s)(s - s_i)}{(s_{i+1} - s_i)}} Z, \quad (1.3)$$

dove  $Z$  è la simulazione di una variabile aleatoria normale standard indipendente dai  $W(s_1), W(s_2), \dots, W(s_k)$ . Quindi, utilizzando la Random Walk Construction per simulare  $W(t_k) - W(0) = W(t_k)$  e, successivamente, applicando la (1.3) ad ognuno degli istanti  $t_1, t_2, \dots, t_{k-1}$ , nell'ordine che abbiamo descritto precedentemente, si trovano i valori  $W(t_1), W(t_2), \dots, W(t_k)$ , e di conseguenza la simulazione del processo  $W$  negli istanti prestabiliti.

Entrambi i metodi visti sono classificati tra le tecniche *esatte* per arrivare ad avere una istanza di un moto Browniano, poichè la distribuzione congiunta del vettore ottenuto  $(W(t_1), W(t_2), \dots, W(t_k))$  coincide con quella dei valori agli istanti  $t_1, t_2, \dots, t_k$  del processo  $W$ . Spesso quando si implementa l'algoritmo per la simulazione di un moto Browniano standard risulta comodo

utilizzare la Brownian Bridge Construction sui primi  $2^m$  istanti di tempo, dove  $m = \max\{m \in \mathbb{N} \mid 2^m \leq k\}$ , e la Random Walk Construction sui restanti. Questo segue dal fatto che, siccome la Brownian Bridge Construction consiste nel dimezzare sempre l'indice dell'ultimo istante di tempo considerato, si presta nel lavorare con una quantità di istanti che sia una potenza di 2.

## 1.2 Moto Browniano generalizzato

**Definizione 1.2.** Diciamo che un processo stocastico  $\{X(t), 0 \leq t \leq T\}$  è un **moto Browniano generalizzato**, con drift  $\mu \in \mathbb{R}$  e coefficiente di diffusione  $\sigma \in \mathbb{R}^+$ , se il processo  $\frac{X(t) - \mu t}{\sigma}$  è un moto Browniano standard.

*Osservazione 1.3.* Un moto Browniano standard è anche un moto Browniano generalizzato.

Consideriamo  $X(t)$  moto Browniano generalizzato con drift e coefficiente di diffusione, rispettivamente,  $\mu \in \mathbb{R}$  e  $\sigma \in \mathbb{R}^+$ . Dalla definizione è noto che  $W(t) := \frac{X(t) - \mu t}{\sigma}$  è un moto Browniano standard, e vale che per ogni  $t \in [0, T]$

$$X(t) = \sigma W(t) + \mu t. \quad (1.4)$$

Da questo, e grazie a quanto sappiamo sulla trasformazione di variabili aleatorie con legge normale, non è difficile individuare la legge di  $X(t)$  con  $t \in [0, T]$ :

$$X(t) \sim N(\mu t, \sigma^2 t) \quad (1.5)$$

Per quanto abbiamo asserito nella definizione 1.2, un moto Browniano generalizzato  $X(t)$ , può essere considerato tale soltanto se  $X(0) = 0$ . Questa è soltanto una normalizzazione: quando parleremo di moti Browniani ci riferiremo a tutti i processi  $X(t) + c$ , dove  $X(t)$  moto Browniano generalizzato e  $c \in \mathbb{R}$ .

Anche per la simulazione di moti Browniani generalizzati si possono utilizzare la *Random Walk Construction* e la *Brownian Bridge Construction*, che saranno la generalizzazione delle omonime tecniche viste per il caso standard.

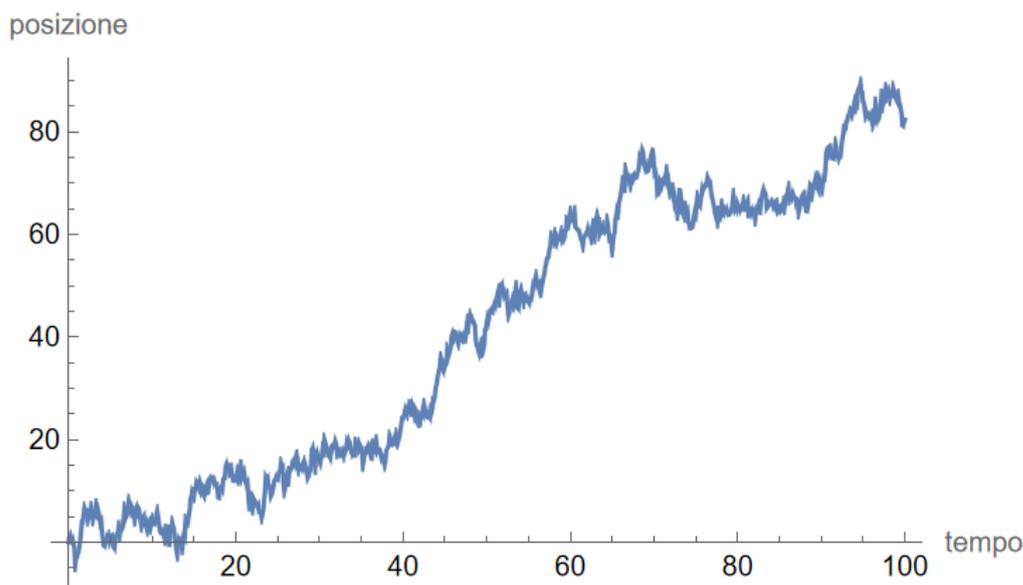


Figura 1.2: Grafico di un moto Browniano generalizzato, avente drift 1 e coefficiente di diffusione 10

Per ottenere il valore di un moto Browniano  $X(t)$  negli istanti  $t_1, t_2, \dots, t_k$  tramite la **Random Walk Construction** si parte osservando la (1.4), e notando che per conoscere il valore di  $X(t_i)$  per ogni  $i \in \{1, 2, \dots, k\}$  è sufficiente conoscere i valori negli stessi istanti del processo  $W$ . Trattandosi  $W$  di un moto Browniano standard, possiamo utilizzare la (1.2) per simulare tutti i  $W(t_i)$ . Da questo segue che, considerando  $Z_1, Z_2, \dots, Z_k$  istanze di variabili aleatorie normali standard indipendenti, e implementando la ricorsione

$$X(t_{i+1}) = X(t_i) + \mu(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1}, \quad i = \{0, 1, \dots, k-1\}, \quad (1.6)$$

si trovano i valori a cui si era interessati. Quanto appena detto può essere implementato con il codice Mathematica:

```
#Inizializzazione delle variabili mu, coeff e t#
lun = Length[t];
z = RandomVariate[MultinormalDistribution
[ConstantArray[0, lun], IdentityMatrix[{lun, lun}]]];
```

```

(*simula tante variabili aleatorie con distribuzione
normale standard quanti gli istanti di tempo*)
x = ConstantArray[0, lun];
x[[1]] = Sqrt[coeff*t[[1]]]*z[[1]] + mu*t[[1]];
(*Iterazione Random Walk*)
For[n = 2, n < lun + 1, n++,
  x[[n]] = x[[n - 1]] + Sqrt[coeff*(t[[n]] - t[[n - 1]])]
  *z[[n]] + mu*(t[[n]] - t[[n - 1]]);
  (*calcola il valore del processo x in ogni punto*)
];

```

Si noti che questo codice può essere utilizzato anche per la Random Walk Construction di un moto Browniano standard, semplicemente settando le variabili `mu` e `coeff` rispettivamente a 0 e 1.

Per quanto riguarda la **Brownian Bridge Construction**, dobbiamo distinguere due casi. Se il moto Browniano di cui ci stiamo occupando  $X(t)$  ha coefficiente di diffusione  $\sigma = 1$  questa procedura rimane quasi invariata rispetto al caso standard. Infatti, quanto detto a tal proposito sulla legge di  $W(s)$  con  $W$  moto Browniano standard, condizionata ai valori già simulati, rimane vero anche per  $X(s)$ . Il valore del processo in tutti gli istanti, eccetto il primo e l'ultimo, può quindi essere ottenuto utilizzando la (1.3), seguendo lo stesso ordine descritto per la Brownian Bridge standard. Di tutta la procedura, l'unico step che subisce variazioni è il secondo in ordine cronologico, ovvero la simulazione del valore all'istante di tempo maggiore. Infatti, denotando tale istante con  $t_k$ , sappiamo dalla (1.5) che  $X(t_k) \sim N(\mu t_k, t_k)$ , per cui, conoscendo  $Z$  valore di una variabile aleatoria normale standard, possiamo porre  $X(t_k) = \mu t_k + \sqrt{t_k} Z$ . Analizziamo ora il secondo caso: se  $X(t)$  moto Browniano generalizzato con  $\sigma \neq 1$  possiamo considerare il processo  $Y(t) := \frac{X(t)}{\sigma}$ , e notare che si tratta di un moto Browniano con drift  $\frac{\mu}{\sigma}$  e coefficiente di diffusione 1. Quindi, è possibile utilizzare la Brownian Bridge Construction nella versione appena vista, per simulare il valore di tale processo negli istanti di tempo a cui si è interessati

$t_1, t_2, \dots, t_k$ . Dopodichè i rispettivi valori di  $X$  si ottengono semplicemente ponendo  $(X(t_1), X(t_2), \dots, X(t_k)) = \sigma(Y(t_1), Y(t_2), \dots, Y(t_k))$ . Riportiamo il corrispondente codice Mathematica (anche in questo caso se  $\mu$  e  $\text{coeff}$  sono rispettivamente 0 e 1 diventa l'algoritmo Brownian Bridge per il caso standard):

```
#Inizializzazione delle variabili mu, coeff e t#
lun = Length[t];
m = IntegerPart[Log2[lun]];
h = 2^m;
z = RandomVariate[MultinormalDistribution
[ConstantArray[0, lun], IdentityMatrix[{lun, lun}]]];
(*simula tante variabili aleatorie con distribuzione
normale standard quanti gli istanti di tempo*)
jmax = 1;
x = ConstantArray[0, lun];
x[[h]] = Sqrt[coeff*t[[h]]]*z[[h]] + mu*t[[h]];
(*Iterazione Brownian Bridge*)
For[k = 1, k < m + 1, k++,
  imin = h/2; i = imin;
  l = 0; r = h;
  For[j = 1, j < jmax + 1, j++,
    a = If[l == 0, (t[[i]]*x[[r]])/t[[r]],
    ((t[[r]]-t[[i]])*x[[l]] + (t[[i]]-t[[l]])*
    x[[r]])/(t[[r]]-t[[l]])];
    b = Sqrt[coeff*If[l == 0, t[[i]]*(t[[r]]-t[[i]])
    /t[[r]], (t[[i]]-t[[l]])*(t[[r]]-t[[i]])/
    (t[[r]]-t[[l]])]];
    x[[i]] = a+b*z[[i]];
    i = i + h; l = l + h; r = r + h;
  ];
  jmax = 2*jmax;
```

```

|         h = imin;
|     ];

```

Questo codice è strutturato per calcolare soltanto le prime  $2^m$  posizioni, dove  $m = \lfloor \log_2 k \rfloor$  (in pratica  $2^m$  è la più grande potenza di due minore o uguale di  $k$ ). Le restanti si possono ottenere con la Random Walk Costruction.

### 1.3 Moto Browniano multidimensionale

**Definizione 1.4.** Diciamo che un processo stocastico a valori in  $\mathbb{R}^d$   $\{W(t) = (W_1(t), W_2(t), \dots, W_d(t))^T, 0 \leq t \leq T\}$  è un **moto Browniano standard multidimensionale** se valgono le seguenti condizioni:

1.  $W(0) = 0$ ;
2. la funzione  $t \rightarrow W(t)$  è continua su  $[0, T]$  quasi certamente;
3. per ogni  $k \in \mathbb{N}$  e per ogni scelta di  $\{t_1, t_2, \dots, t_k\} \subset ]0, T[$  le variabili aleatorie  $\{W(t_1) - W(0), W(t_2) - W(t_1), \dots, W(t_k) - W(t_{k-1}), W(T) - W(t_k)\}$  sono indipendenti;
4.  $W(t) - W(s) \sim N(0, (t - s)I)$  per ogni  $0 \leq s < t \leq T$ , dove  $I$  è la matrice identità  $d \times d$ .

*Osservazione 1.5.* Se  $(W_1(t), W_2(t), \dots, W_d(t))$  è un moto Browniano multidimensionale, allora  $W_i(t)$ , per ogni  $i \in \{1, 2, \dots, d\}$ , è un moto Browniano standard in una dimensione.

**Definizione 1.6.** Diciamo che un processo stocastico a valori in  $\mathbb{R}^d$   $\{X(t) = (X_1(t), X_2(t), \dots, X_d(t))^T, 0 \leq t \leq T\}$  è un **moto Browniano multidimensionale**, con drift  $\mu \in \mathbb{R}^d$  e matrice di covarianza  $\Sigma \in \mathbb{R}^{d \times d}$  simmetrica semidefinita positiva, se valgono le seguenti condizioni:

1. la funzione  $t \rightarrow X(t)$  è continua su  $[0, T]$  quasi certamente;

2. per ogni  $k \in \mathbb{N}$  e per ogni scelta di  $\{t_1, t_2, \dots, t_k\} \subset ]0, T[$  le variabili aleatorie  $\{X(t_1) - X(0), X(t_2) - X(t_1), \dots, X(t_k) - X(t_{k-1}), X(T) - X(t_k)\}$  sono indipendenti;
3.  $X(t) - X(s) \sim N((t-s)\mu, (t-s)\Sigma)$  per ogni  $0 \leq s < t \leq T$ .

Il valore iniziale di un moto Browniano multidimensionale è, quindi, un vettore arbitrario, e non necessariamente 0. Analogamente a quanto osservato per il caso monodimensionale, un moto Browniano standard multidimensionale è un caso particolare di moto Browniano multidimensionale. Per le proprietà della trasformazione lineare di variabili aleatorie con distribuzione normale osserviamo che, preso  $W(t)$  moto Browniano standard su  $\mathbb{R}^j$ , presi  $\mu \in \mathbb{R}^d$ ,  $\Sigma \in \mathbb{R}^{d \times d}$  e  $B \in \mathbb{R}^{d \times j}$  tali che  $BB^T = \Sigma$ , il processo  $X(t) := \mu t + BW(t)$  è un moto Browniano su  $\mathbb{R}^d$  con drift  $\mu$ , matrice di covarianza  $\Sigma$  e valore iniziale 0.

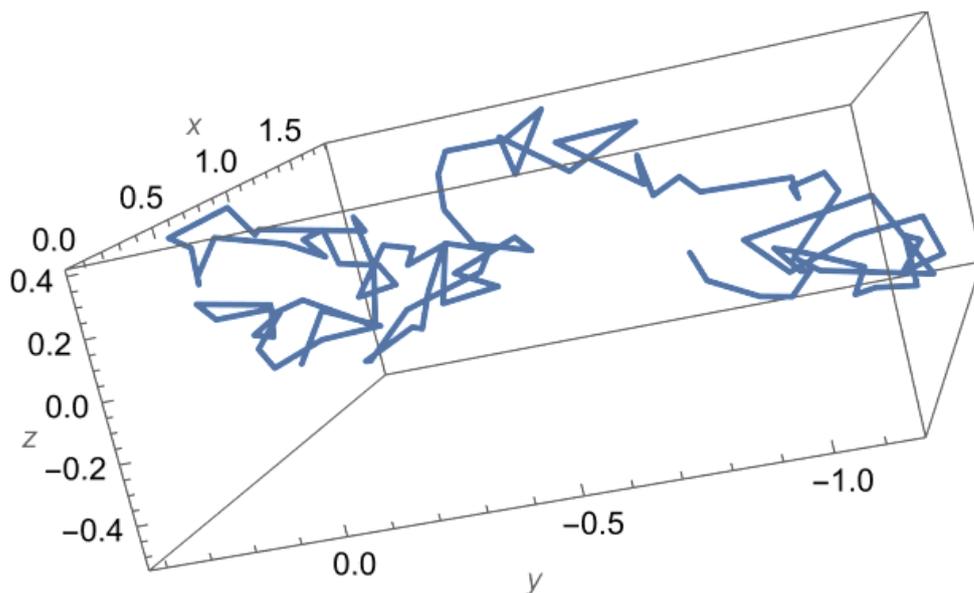


Figura 1.3: Grafico della traiettoria di un moto Browniano tridimensionale standard

Anche i moti Browniani multidimensionali possono essere costruiti utilizzando delle versioni ancora più generali della *Random Walk Construction* e della *Brownian Bridge Construction*.

La **Random Walk Construction** per il caso multidimensionale standard si ricava facendo lo stesso ragionamento sulla legge degli incrementi visto per il caso monodimensionale, ottenendo una ricorsione del tutto analoga. In particolare, denotando con  $W(t)$  un moto Browniano standard a valori in  $\mathbb{R}^d$ , con  $t_1, t_2, \dots, t_k$  la discretizzazione dell'intervallo di tempo su cui si vuole avere il valore del processo, e con  $Z_1, Z_2, \dots, Z_k$  le simulazioni di vettori aleatori indipendenti con legge  $N(0, I)$ , si implementa la ricorsione:

$$W(t_{i+1}) = W(t_i) + \sqrt{t_{i+1} - t_i} Z_{i+1}, \quad i = \{0, 1, \dots, k-1\}, \quad (1.7)$$

ottenendo quanto richiesto. Supponiamo, invece, che il processo con cui si sta lavorando sia un moto Browniano non standard  $X(t)$  con drift  $\mu \in \mathbb{R}^d$ , matrice di covarianza  $\Sigma \in \mathbb{R}^{d \times d}$  e posizione iniziale  $c \in \mathbb{R}^d$ . Supponiamo senza perdita di generalità che  $c = 0$  (altrimenti si simula il processo  $X(t) - c$  e poi si calcola  $X(t)$  in maniera banale). Analogamente a quanto fatto per i moti unidimensionali, siamo interessati a scrivere  $X(t)$  come trasformazione lineare di un moto Browniano standard. Per fare ciò è sufficiente trovare una matrice  $B \in \mathbb{R}^{d \times d}$  tale che  $BB^T = \Sigma$ , in modo che il processo  $\mu t + BW(t)$  abbia le stesse caratteristiche di  $X(t)$ . A questo punto, seguendo una procedura del tutto analoga a quella con cui si ricava la ricorsione 1.6, si ha che è possibile trovare il valore di  $X$  sugli istanti di tempo  $t_1, t_2, \dots, t_k$  attraverso i seguenti passaggi:

$$X(t_{i+1}) = X(t_i) + \mu(t_{i+1} - t_i) + B\sqrt{t_{i+1} - t_i} Z_{i+1}, \quad i = \{0, 1, \dots, k-1\}, \quad (1.8)$$

dove  $Z_1, Z_2, \dots, Z_n$  sono i soliti vettori aleatori indipendenti normali standard. Per quanto riguarda il calcolo della fattorizzazione di  $\Sigma$ , abbiamo due casi. Se  $\Sigma$  definita positiva si può utilizzare la *fattorizzazione di Cholesky*, ovvero trovare  $B \in \mathbb{R}^{d \times d}$  triangolare inferiore tale che  $BB^T = \Sigma$ . Questa tecnica è computazionalmente vantaggiosa rispetto alle altre esistenti, poichè si

ottiene una matrice che ha tutti zeri sopra la diagonale, e questo semplifica le operazioni di calcolo dei valori del processo  $X$ , tramite la (1.8). Se invece  $\Sigma$  è semidefinita positiva ma non definita positiva, si può utilizzare la *decomposizione spettrale* ( $\Sigma$  simmetrica), ovvero considerare le matrici  $D \in \mathbb{R}^{d \times d}$  diagonale, e  $V \in \mathbb{R}^{d \times d}$  ortogonale, tali che  $\Sigma = VDVT^T$ . Siccome gli elementi sulla diagonale di  $D$  sono gli autovalori di  $\Sigma$  e quindi tutti positivi o nulli ( $\Sigma$  semidefinita positiva), allora esiste  $D^{\frac{1}{2}}$ , e vale che  $D = D^{\frac{1}{2}}D^{\frac{1}{2}}$ . Si ottiene che  $\Sigma = (VD^{\frac{1}{2}})(D^{\frac{1}{2}}V^T)$ , per cui definendo  $B := VD^{\frac{1}{2}}$  si ha la fattorizzazione cercata.

Studiamo, ora, la costruzione di un moto Browniano su  $\mathbb{R}^d$ , tramite la **Brownian Bridge Construction**. Se il processo di cui si vuole ottenere il valore è un moto Browniano standard  $W(t)$ , la simulazione è banale. Infatti, considerando  $\tilde{W}_1(t), \tilde{W}_2(t), \dots, \tilde{W}_d(t)$  moti Browniani standard monodimensionali indipendenti, il processo  $(\tilde{W}_1(t), \tilde{W}_2(t), \dots, \tilde{W}_d(t))$  rispetta tutte le proprietà per essere classificato come moto Browniano standard a valori in  $\mathbb{R}^d$ . Per questo, per avere una simulazione di  $W(t)$  è sufficiente applicare separatamente e in maniera indipendente la Brownian Bridge Construction studiata per il caso monodimensionale standard ad ognuna delle componenti, sull'insieme di istanti a cui si è interessati. Invece, nel caso in cui si abbia bisogno di ottenere il valore di un moto Browniano  $X(t)$  con drift  $\mu \in \mathbb{R}^d$  e matrice di covarianza  $\Sigma \in \mathbb{R}^{d \times d}$ , su  $t_1, t_2, \dots, t_k$ , si simula prima il valore di un moto Browniano standard  $W(t)$  sullo stesso insieme di istanti, dopodichè si applica la trasformazione lineare già studiata per la Random Walk Construction: fattorizzando  $\Sigma$  in modo che  $\Sigma = BB^T$ , si può porre  $X(t_i) = \mu t_i + BW(t_i)$  per ogni  $i \in \{1, 2, \dots, k\}$ , ottenendo tutti i valori cercati.

Di seguito è riportato il codice Mathematica che costruisce un moto Browniano multidimensionale nel caso in cui  $\Sigma$  sia definita positiva, utilizzando la Brownian Bridge Construction per le prime  $2^m$  posizioni, dove  $m = \lfloor \log_2 k \rfloor$ , e la Random Walk Construction per le restanti.

```

#Inizializzazione delle variabili mu, sigma, t, w0 e d#
lun = Length[t];
m = IntegerPart[Log2[lun]];
nd = lun*d;
Z = RandomVariate[MultinormalDistribution[
ConstantArray[0, nd], IdentityMatrix[{nd, nd}]]];
(*simula tante v.a. con distribuzione normale
standard, quanti gli istanti di tempo
moltiplicati per la dimensione*)
Z = ArrayReshape[Z, {d, lun}];
w = ConstantArray[0, {d, lun}];
(*calcolo delle prime 2^m posizioni
(Brownian Bridge Construction)*)
For[p = 1, p < d + 1, p++,
  h = 2^m; jmax = 1;
  w[[p, h]] = Sqrt[t[[h]]]*Z[[p, h]];
  For[k = 1, k < m + 1, k++,
    imin = h/2; i = imin;
    l = 0; r = h;
    For[j = 1, j < jmax + 1, j++,
      a = If[l == 0, t[[i]]*w[[p, r]]/t[[r]],
        ((t[[r]]-t[[i]])*w[[p,l]]+(t[[i]]-t[[l]])
        *w[[p, r]])/(t[[r]] - t[[l]]));
      b = Sqrt[If[l == 0, t[[i]]*(t[[r]] - t[[i]])/
        t[[r]], (t[[i]] - t[[l]])*(t[[r]]-t[[i]])/
        (t[[r]] - t[[l]])]];
      w[[p, i]] = a + b*Z[[p, i]];
      i = i + h; l = l + h; r = r + h;
    ];
  jmax = 2*jmax; h = imin;
];
];

```

```

];
h = 2^m;
(*calcolo delle rimanenti posizioni
(Random Walk Construction)*)
For[p = 1, p < d + 1, p++,
  For[n = h + 1, n < lun + 1, n++,
    w[[p, n]] = w[[p, n - 1]] +
    Sqrt[t[[n]] - t[[n - 1]]]*Z[[p, n]];
  ];
];
B = Transpose[CholeskyDecomposition[sigma]];
(*calcola la fattorizzazione di Cholesky di sigma*)
X = ConstantArray[0, {d, lun}];
For[j = 1, j < lun + 1, j++,
  X[[All, j]] = mu*t[[j]] + B.w[[All, j]] + w0;
];

```

## 1.4 Moto Browniano geometrico

**Definizione 1.7.** Diciamo che un processo stocastico  $\{S(t), 0 \leq t \leq T\}$  è un **moto Browniano geometrico** con drift  $\mu \in \mathbb{R}$ , volatilità  $\sigma \in \mathbb{R}^+$  e posizione iniziale  $S(0) \in \mathbb{R}^+$  se esiste un moto Browniano generalizzato  $X(t)$ , con drift  $\mu$  e coefficiente di diffusione  $\sigma$ , tale che  $S(t) = S(0) \exp(X(t))$ .

Sia  $S(t)$  un moto Browniano geometrico con drift  $\mu \in \mathbb{R}$ , volatilità  $\sigma \in \mathbb{R}^+$  e posizione iniziale  $S(0) \in \mathbb{R}^+$ , dalla definizione possiamo scrivere che  $S(t) = S(0) \exp(X(t)) = f(X(t))$  dove  $f(x) := S(0)e^x$  e  $X(t)$  è un moto Browniano generalizzato con drift  $\mu \in \mathbb{R}$ , coefficiente di diffusione  $\sigma \in \mathbb{R}^+$ . Ricordando che esiste  $W(t)$  moto Browniano standard tale che  $X(t) = \sigma W(t) + \mu t$ , da una applicazione della *formula di Itô* si ottiene la seguente:

$$S(t) = S(0) \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W(t)\right), \quad (1.9)$$

per ogni  $t \in [0, T]$ . Osservando la definizione si nota facilmente che  $S(t)$  è un processo a valori positivi. Di conseguenza, applicando la (1.9) otteniamo che, fissati  $t, u \in [0, T]$ , vale  $\frac{S(t)}{S(u)} = \frac{S(0) \exp((\mu - \frac{1}{2}\sigma^2)t + \sigma W(t))}{S(0) \exp((\mu - \frac{1}{2}\sigma^2)u + \sigma W(u))}$ . Da questo, facendo i conti:

$$S(t) = S(u) \exp((\mu - \frac{1}{2}\sigma^2)(t - u) + \sigma(W(t) - W(u))). \quad (1.10)$$

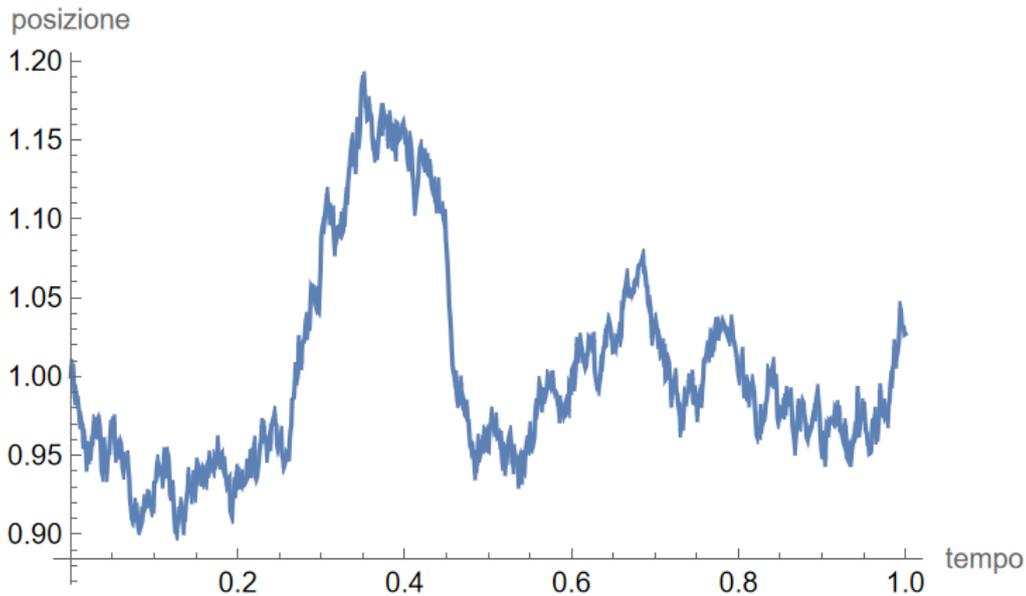


Figura 1.4: Grafico di un moto Browniano geometrico, avente drift 0.1, volatilità 0.2, e posizione iniziale 1

La simulazione di un moto Browniano geometrico  $S(t)$  sugli istanti di tempo  $t_1, t_2, \dots, t_k$  è banale grazie a quanto appena detto. Infatti, dalla (1.10) segue che se si simulano  $W(t_1), W(t_2), \dots, W(t_k)$ , ad esempio tramite la Random Walk Construction, i valori di  $S$  negli stessi punti si ricavano in maniera deterministica. Quindi, ricordando la (1.2), e considerando  $Z_1, Z_2, \dots, Z_k$

simulazioni di normali standard indipendenti, abbiamo che ponendo

$$S(t_{i+1}) = S(t_i) \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1}\right), \quad (1.11)$$

$$i = \{0, 1, \dots, k - 1\},$$

si ottiene quanto desiderato. Implementando questa iterazione in Mathematica, si ottiene il seguente codice:

```

lun = Length[t];
z = RandomVariate[MultinormalDistribution[
ConstantArray[0, lun], IdentityMatrix[{lun, lun}]]];
(*genera tante variabili aleatorie con distribuzione
normale standard quanti gli istanti di tempo*)
s = ConstantArray[0, lun];
(*calcolo delle posizioni*)
s[[1]] = s0*Exp[(mu - 0.5*(sigma)^2)*
t[[1]]+sigma*Sqrt[t[[1]]]*z[[1]]];
For[i = 2, i < lun + 1, i++,
  s[[i]]=s[[i-1]]*Exp[(mu-0.5*(sigma)^2)
*(t[[i]]-t[[i-1]]) + sigma*
Sqrt[t[[i]]-t[[i-1]]]*z[[i]]];
];

```

# Capitolo 2

## Derivati finanziari

In questo capitolo tratteremo una categoria particolare di titoli finanziari, ovvero i derivati finanziari, dandone la definizione e distinguendo tra le principali tipologie esistenti. Dopodichè parleremo del metodo Monte Carlo: una tecnica che consente di calcolare il valore di alcuni integrali, tramite l'iterazione della simulazione di apposite variabili aleatorie. Il funzionamento del metodo Monte Carlo è garantito dalla legge dei grandi numeri. Infine applicheremo il metodo Monte Carlo e quanto detto nello scorso capitolo riguardo alla simulazione dei moti Browniani (e in particolare alla simulazione dei moti Browniani geometrici), alla valutazione del payoff di derivati finanziari. Infatti il moto Browniano geometrico e il prezzo dei titoli finanziari sono processi stocastici che hanno in comune la caratteristica di essere a valori positivi, per questo, quando si cerca di modellizzare l'andamento del prezzo di un titolo nel tempo, spesso si utilizza proprio il moto Browniano geometrico, con una accorta scelta del drift e della volatilità.

### 2.1 Principali tipologie di derivati finanziari

**Definizione 2.1.** *Un titolo finanziario si dice essere un **derivato finanziario** se il suo valore dipende da quello di un altro titolo, detto sottostante.*

Il titolo sottostante può essere un titolo finanziario, come un'azione o un indice di un mercato, oppure può essere un bene reale, come una materia prima o una certa quantità di una valuta. Il prezzo di un derivato finanziario al momento del riscatto è detto *payoff*. Nel seguito denoteremo con  $0$  l'istante di tempo in cui viene sottoscritto il contratto che rappresenta il derivato, e con  $S(t)$  il prezzo del titolo sottostante al tempo  $t$ , con  $t \in [0, T]$ . Esistono due principali categorie di derivati finanziari: i *future contracts* e le *opzioni*.

**Definizione 2.2.** *Un **future contract** è un contratto, e in particolare un derivato finanziario, in cui l'acquirente si impegna a comprare o vendere un altro titolo ad un prezzo predefinito  $K$ , detto *strike*, ad un certo istante futuro  $T$ , detto *scadenza*.*

**Definizione 2.3.** *Un'opzione è un contratto, e in particolare un derivato finanziario, che dà il diritto ma non l'obbligo di comprare o vendere un altro titolo ad un prezzo predefinito  $K$ , detto *strike*, entro un certo istante futuro  $T$ , detto *scadenza*. Inoltre, un'opzione si dice essere una **call** se permette l'acquisto del titolo sottostante, mentre si dice essere una **put** se ne permette la vendita.*

**Definizione 2.4.** *Un'opzione è detta **path-dependent** se il suo payoff dipende anche da qualche  $S(t)$  con  $t \in [0, T[$ , valore del titolo sottostante al tempo  $t$ , e non soltanto dal valore alla scadenza  $S(T)$ .*

Le opzioni si distinguono a loro volta in diverse tipologie, a seconda di come ne viene calcolato il payoff. Di seguito sono elencate le più importanti.

- **OPZIONI EUROPEE** Si tratta della tipologia più semplice di opzione. Il diritto di compra o di vendita del titolo sottostante può essere esercitato soltanto alla scadenza dell'opzione. Il payoff nelle call e nelle put si ottiene, rispettivamente, calcolando

$$(S(T) - K)^+ \quad \text{e} \quad (K - S(T))^+$$

- **OPZIONI AMERICANE** A differenza delle opzioni europee, in questa tipologia l'opzione può essere riscattata in qualsiasi istante di tempo

$t \in [0, T]$ . Analogamente a quanto accade nel caso europeo, i payoff di call e put saranno, rispettivamente,

$$(S(\tilde{t}) - K)^+ \quad \text{e} \quad (K - S(\tilde{t}))^+,$$

dove  $\tilde{t} \in [0, T]$  denota l'istante in cui si esercita l'opzione.

- **OPZIONI ASIATICHE DISCRETE** Possono essere riscattate, allo stesso modo di quanto accade nel caso europeo, soltanto alla scadenza  $T$ . La caratteristica che contraddistingue le opzioni asiatiche è che l'opzione non viene valutata soltanto sul valore attuale del titolo sottostante e su uno strike prefissato, ma anche sulla media di alcuni valori del titolo sottostante monitorati tra la sottoscrizione del contratto e la scadenza. In particolare, nel caso di una opzione asiatica discreta, fissati  $t_1, t_2, \dots, t_n \in [0, T]$ , e definito

$$\tilde{S} := \frac{1}{n} \sum_{i=1}^n S(t_i),$$

il payoff di una call di questo tipo può essere fissato ad uno dei due valori tra

$$(\tilde{S} - K)^+ \quad \text{e} \quad (S(T) - \tilde{S})^+,$$

e analogamente quello di una put ad uno tra

$$(K - \tilde{S})^+ \quad \text{e} \quad (\tilde{S} - S(T))^+,$$

a seconda che il payoff sia prefissato ad un certo valore  $K$ , oppure che sia proprio la media dei valori del titolo sottostante sugli istanti in cui viene monitorato. Le opzioni asiatiche sono esempio di opzione path-dependent.

- **OPZIONI ASIATICHE CONTINUE** Sono analoghe alle opzioni asiatiche discrete, con l'unica differenza che, invece di essere monitorate su un insieme discreto di istanti, vengono monitorate in un intervallo di tempo  $[t, u] \subseteq [0, T]$ . Il payoff si calcola utilizzando le stesse formule

del caso discreto, dopo aver definito  $\tilde{S}$  come la media continua di  $S(t)$  su  $[t, u]$ , ovvero

$$\tilde{S} := \frac{1}{u-t} \int_t^u S(\tau) d\tau.$$

- **OPZIONI GEOMETRICHE** Sono il corrispondente delle opzioni asiatiche discrete per la media geometrica, ovvero si fissano  $t_1, t_2, \dots, t_n$  e si definisce  $\tilde{S}$  come la media geometrica dei valori del processo  $S(t)$  valutato su tali istanti, ossia

$$\tilde{S} := \left( \prod_{i=1}^n S(t_i) \right)^{\frac{1}{n}},$$

Dopodichè si calcola il payoff con le stesse formule descritte per le asiatiche discrete. Anche queste opzioni ammettono una versione a monitoraggio continuo.

- **OPZIONI CON BARRIERA** Questa tipologia di opzione funziona come le europee, con la differenza che si aggiunge una condizione necessaria per la validità dell'opzione, che riguarda come si comporta il titolo sottostante tra la sottoscrizione e la scadenza. Un esempio di opzione con barriera è la *down-and-out* call, opzione che resta valida soltanto se il valore del titolo sottostante non scende mai sotto un certo limite (detto appunto *barriera*) negli istanti di monitoraggio  $\{t_0, t_1, \dots, t_n = T\}$  (o in tutti gli istanti fino alla scadenza, nel caso in cui il monitoraggio sia continuo). In questo caso il payoff sarà quindi

$$\mathbf{1}_{\{\tau(b) > T\}} (S(T) - K)^+,$$

dove  $b$  è la barriera, e

$$\tau(b) := \inf_{i=1,2,\dots,n} \{t_i \mid S(t_i) < b\}.$$

- **OPZIONI LOOKBACK** Sono opzioni esercitabili solo alla scadenza, il cui payoff viene valutato sui valori estremi tra quelli che il titolo sottostante assume tra la sottoscrizione e la scadenza. Infatti, supponiamo che il valore del titolo sottostante sia monitorato sull'insieme

discreto di istanti  $\{t_1, t_2, \dots, t_n = T\}$ , allora il payoff di una call e di una put lookback sarà, rispettivamente,

$$\left(\max_{i=1,2,\dots,n} S(t_i) - S(T)\right)^+ \quad \text{e} \quad \left(S(T) - \min_{i=1,2,\dots,n} S(t_i)\right)^+.$$

Anche queste opzioni hanno una corrispondente versione a monitoraggio continuo.

## 2.2 Metodo Monte Carlo

Il metodo **Monte Carlo** è un metodo utilizzato per l'approssimazione di integrali, sia in una che in più dimensioni. Questo metodo consiste nel considerare una certa variabile aleatoria, il cui valore atteso coincida con il valore dell'integrale a cui si è interessati, e procedere iterando la simulazione di tale variabile aleatoria, al fine di calcolare proprio  $\mathbb{E}[X]$ , e di conseguenza l'integrale. Per capire il funzionamento del metodo Monte Carlo è necessario conoscere un insieme di due teoremi: la legge dei grandi numeri.

Consideriamo una successione di variabili aleatorie indipendenti identicamente distribuite  $(X_n)_{n \in \mathbb{N}}$ , definite sullo stesso spazio di probabilità  $(\Omega, \mathcal{F}, \mathbb{P})$ . Definiamo le variabili aleatorie

$$S_n := X_1 + X_2 + \dots + X_n \quad \text{e} \quad M_n := \frac{S_n}{n}$$

e le costanti

$$\mu := \mathbb{E}[X_1] \quad \text{e} \quad \sigma^2 := \text{var}(X_1)$$

Sotto queste ipotesi abbiamo i seguenti due teoremi, detti rispettivamente legge dei grandi numeri debole e legge dei grandi numeri forte.

**Teorema 2.5 (Legge dei grandi numeri debole).** *Sia  $(X_n)_{n \in \mathbb{N}}$  una successione di variabili aleatorie in una dimensione, indipendenti e identicamente distribuite, tale che  $X_n \in L^2(\Omega, \mathbb{P})$  per ogni  $n \in \mathbb{N}$ . Allora si ha che:*

$$M_n \xrightarrow{L^2} \mu \tag{2.1}$$

*Dimostrazione.* Per linearità del valore atteso, e considerando che le  $X_i$  sono equidistribuite:

$$\mathbb{E}[M_n] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[X_i] = \mu \quad \forall n \in \mathbb{N}.$$

Utilizzando una proprietà nota della varianza di una somma di variabili aleatorie indipendenti, si ottiene che

$$\mathbb{E}[(M_n - \mu)^2] = \text{var}(M_n) = \frac{\sum_{i=1}^n \text{var}(X_i)}{n^2} = \frac{\sigma^2}{n} \quad \forall n \in \mathbb{N}.$$

Di conseguenza si arriva alla tesi semplicemente osservando che

$$\lim_{n \rightarrow +\infty} \mathbb{E}[(M_n - \mu)^2] = \lim_{n \rightarrow +\infty} \frac{\sigma^2}{n} = 0.$$

□

**Teorema 2.6 (Legge dei grandi numeri forte).** *Nelle stesse ipotesi del teorema precedente si ha che:*

$$M_n \xrightarrow{q.c.} \mu \quad (2.2)$$

*Dimostrazione.* Si può supporre SPDG che  $\mu = 0$ , poichè in caso contrario si ripete la dimostrazione con le variabili  $X_i - \mu$ . Consideriamo la sottosuccessione  $(M_{n^2})_{n \in \mathbb{N}}$  della successione  $(M_n)_{n \in \mathbb{N}}$ . Allora

$$\mathbb{E}\left[\sum_{n=1}^N M_{n^2}^2\right] = \sum_{n=1}^N \mathbb{E}[M_{n^2}^2] = \sum_{n=1}^N \frac{\sigma^2}{n^2} \quad \forall N \in \mathbb{N}.$$

Per il teorema di Beppo-Levi

$$\mathbb{E}\left[\sum_{n=1}^{+\infty} M_{n^2}^2\right] = \sum_{n=1}^{+\infty} \frac{\sigma^2}{n^2} \quad \forall N \in \mathbb{N}.$$

Quest'ultima serie è convergente, per cui  $\sum_{n=1}^{+\infty} M_{n^2}^2$  converge quasi certamente e di conseguenza, per una condizione necessaria sulla convergenza delle serie

$$M_{n^2} \xrightarrow{q.c.} 0.$$

Definiamo  $p_n := \lfloor \sqrt{n} \rfloor$ , e notiamo che vale che

$$\begin{aligned} p_n \leq \sqrt{n} < p_n + 1 &\implies p_n^2 \leq n < (p_n + 1)^2 \implies \\ n - (p_n + 1)^2 < 0 &\implies n - p_n^2 < 2p_n + 1 \end{aligned}$$

Osserviamo che per definizione di  $M_n$

$$M_n - \frac{p_n^2}{n} M_{p_n^2} = \frac{1}{n} \sum_{i=p_n^2+1}^n X_i.$$

Da questo, e da quanto detto sopra

$$\begin{aligned} \mathbb{E}[(M_n - \frac{p_n^2}{n} M_{p_n^2})^2] &= \frac{1}{n^2} \text{var}(\sum_{i=p_n^2+1}^n X_i) = \frac{n - p_n^2}{n^2} \sigma^2 \\ &\leq \frac{2p_n + 1}{n^2} \sigma^2 \leq \frac{2\sqrt{n} + 1}{n^2} \sigma^2 \leq \frac{3\sigma}{n^{3/2}} \quad \forall n \in \mathbb{N}. \end{aligned}$$

Sfruttando ancora una volta il teorema di Beppo-Levi

$$\mathbb{E}[\sum_{n=1}^{+\infty} (M_n - \frac{p_n^2}{n} M_{p_n^2})^2] \leq \sum_{n=1}^{+\infty} \frac{3\sigma^2}{n^{3/2}},$$

che è una serie convergente. Dalla condizione necessaria per la convergenza delle serie, come fatto sopra, si conclude che

$$M_n - \frac{p_n^2}{n} M_{p_n^2} \xrightarrow{q.c.} 0,$$

e notando che  $\lim_{n \rightarrow +\infty} \frac{p_n^2}{n} = 1$  si ha la tesi.  $\square$

Le leggi dei grandi numeri garantiscono la convergenza della variabile aleatoria media alla variabile aleatoria costante valore atteso. In parole povere, si ha che, scelto  $n$  abbastanza grande, risulterà che  $M_n \simeq \mu$  quasi certamente, con una approssimazione che dipenderà proprio dalla grandezza di  $n$ . Da questo fatto segue il funzionamento del metodo Monte Carlo. Infatti, se si effettuano  $n$  simulazioni *indipendenti* di una variabile  $X$ , queste possono essere considerate proprio come i valori di  $X_1, X_2, \dots, X_n$  variabili aleatorie indipendenti, tutte con la stessa legge di  $X$  (e quindi equidistribuite). Allora,

calcolando la media aritmetica di tali valori, si ottiene una determinazione della variabile aleatoria  $M_n$ , definita come nelle notazioni utilizzate finora, che approssimerà il valore di  $\mu$ , ovvero  $\mathbb{E}[X]$ .

*Esempio 2.7* (Calcolo dell'integrale  $\int_0^1 e^x dx$  utilizzando il metodo Monte Carlo). Consideriamo la variabile aleatoria  $X \sim \text{Unif}([0, 1])$ . Allora per quanto sappiamo sul calcolo del valore atteso delle variabili aleatorie, siccome  $\exp \in L^2(\mathbb{R}, \text{Unif}[0, 1])$

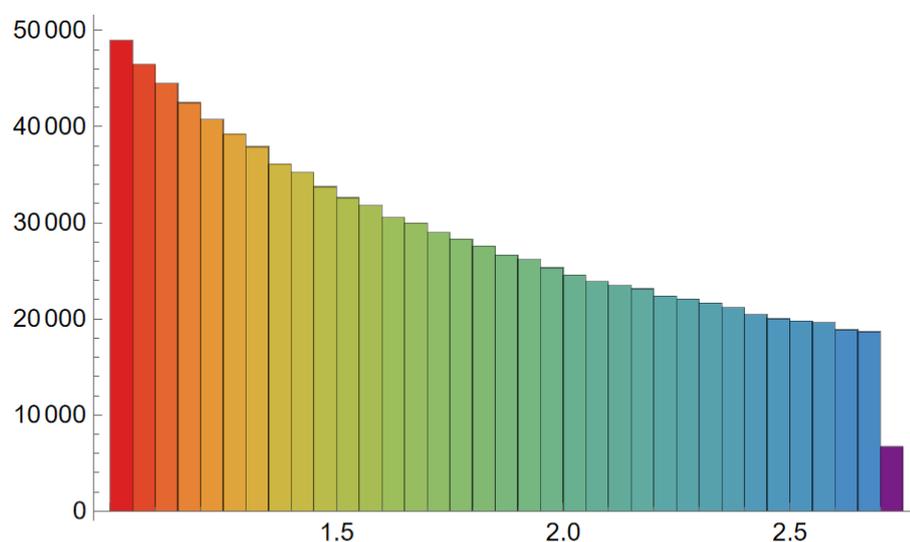
$$\int_0^1 e^x dx = \mathbb{E}[\exp(X)].$$

Quindi, se si itera un numero  $n$  di volte la simulazione della variabile aleatoria  $X$ , valutando la funzione esponenziale in tali valori, si possono calcolare  $n$  realizzazioni della variabile aleatoria  $\exp(X)$ . Sfruttando la legge dei grandi numeri, facendo la media aritmetica di tali simulazioni, si approssima il valore atteso della variabile aleatoria  $\exp(X)$  (se  $n$  sufficientemente grande), e di conseguenza l'integrale a cui si era interessati. La procedura descritta sinora si può implementare tramite il software di calcolo Mathematica, con il seguente codice:

```
f[x_] := Exp[x] (*Definisce la funzione e^x*)
n = 1000000; (*Scelta del numero n di simulazioni*)
vett = RandomVariate[UniformDistribution[{0, 1}], n];
(*Si crea il vettore delle simulazioni di X*)
m = Mean[f[vett]]; (*Media delle simulazioni di e^X*)
Histogram[f[vett], ColorFunction -> Function[{height},
ColorData["Rainbow"][height]]]
(*Istogramma con i valori delle simulazioni*)
Print[m, " = approssimazione dell'integrale con il
metodo Monte Carlo] (*Stampa il risultato ottenuto*)
```

Questo codice definisce la funzione esponenziale, dopodichè sorteggia 1000000 di numeri reali compresi tra 0 e 1 con distribuzione uniforme. Di ognuno di questi calcola l'esponenziale, per poi far la media dei risultati ottenuti, e rappresentare in un istogramma dove questi si posizionano sulla

retta reale. Infine viene stampata a video la media appena calcolata, ovvero l'approssimazione ottenuta dell'integrale. Ciò che si visualizza eseguendo il codice è rappresentato nell'immagine 2.1. Si può notare come il risultato



1.71817 = approssimazione con il metodo Monte Carlo di  $\int_0^1 e^x dx$

Figura 2.1: Output del codice Mathematica per il calcolo dell'integrale tra 0 e 1 della funzione esponenziale, utilizzando il metodo Monte Carlo

dell'integrale calcolato utilizzando il metodo Monte Carlo ben approssimi il risultato effettivo, ottenuto applicando il metodo classico. Infatti

$$\int_0^1 e^x dx = [e^x]_0^1 = e - 1 \simeq 1,71828.$$

## 2.3 Valutazione del payoff di derivati

Consideriamo un titolo finanziario  $S$ , il cui valore, che varia nel tempo in maniera aleatoria, nel seguito verrà rappresentato dal processo stocastico  $\{S(t), t \in \mathbb{R}^+\}$ . Consideriamo, ora, un derivato finanziario sul titolo  $S$ , e denotiamo con  $X$  il payoff di tale derivato. Ovviamente, il valore di  $X$  varia

a seconda di come il valore del titolo sottostante si comporta nel tempo, per cui può essere considerato come una variabile aleatoria. Quando ci si occupa della compravendita di derivati finanziari, il valore atteso del payoff  $\mathbb{E}[X]$ , è uno dei principali fattori su cui si basa il prezzaggio del derivato. Infatti conoscendo questo valore atteso è possibile capire se con l'acquisto di tale derivato ad un certo prezzo fissato ci si *attende*, appunto, di andare in perdita o in guadagno, e di conseguenza se sia o meno conveniente. Per questo il calcolo di tale valore, risulta di fondamentale importanza, sia da parte del venditore che da parte di un ipotetico acquirente.

Per quanto appena detto, è naturale chiedersi come effettuare il calcolo di  $\mathbb{E}[X]$ . Sappiamo che il modello *Black&Scholes* modella il titolo sottostante con una soluzione dell'equazione differenziale stocastica

$$dS(t) = rS(t)dt + \sigma S(t)dW(t), \quad (2.3)$$

dove  $r \in \mathbb{R}$  è il *tasso privo di rischio atteso* del mercato (che supponiamo costante),  $\sigma \in \mathbb{R}^+$  è la *volatilità* del titolo e  $W(t)$  è un moto Browniano. Risolvendo l'equazione si trova che

$$S(t) = S(0) \exp\left(\left(r - \frac{1}{2}\sigma^2\right)t + \sigma W(t)\right), \quad (2.4)$$

per cui, ricordandosi la definizione 1.7, si nota che se  $S$  è il titolo sottostante allora è un moto Browniano geometrico, con drift  $r$ , volatilità  $\sigma$  e valore iniziale  $S(0)$ .

A questo punto, conoscendo il valore dei parametri  $r \in \mathbb{R}$  e  $\sigma \in \mathbb{R}^+$ , è possibile sfruttare quanto appreso nel primo capitolo, riguardo alla simulazione della traiettoria di moti Browniani (e in particolare di moti Browniani geometrici) per simulare il processo  $S$ , ovvero il comportamento del titolo sottostante tra la sottoscrizione del contratto e la scadenza. Se si ha una traiettoria del processo  $S$ , si può calcolare il corrispondente valore del derivato al momento del riscatto, e di conseguenza il valore della variabile aleatoria payoff  $X$ . Sarà possibile, quindi, approssimare il valore atteso di  $X$  nel seguente modo: si simula un numero molto elevato di volte la traiettoria di  $S$ ,

e per ognuna di queste si calcola il payoff risultante (utilizzando la formula prevista per la tipologia di opzione in questione). Ognuno di questi payoff ottenuti può essere visto come una simulazione della variabile aleatoria  $X$  e di conseguenza, utilizzando il metodo Monte Carlo, si calcola la media aritmetica di questi valori e si approssima  $\mathbb{E}[X]$ .

Finora abbiamo implicitamente supposto che il titolo sottostante non paghi alcun dividendo. Ora, invece, consideriamo il caso in cui  $S$  paghi un *dividendo costante continuo*  $\delta$ , ovvero che al tempo  $t \in \mathbb{R}^+$  venga pagato un dividendo di  $\delta S(t)$ . Allora, in questo caso, si denota con  $D(t)$  la quantità degli interessi pagati dal titolo nell'intervallo di tempo  $t \in [0, T]$  e degli interessi pagati su tali dividendi. Di conseguenza il processo  $D(t)$  è soluzione dell'equazione differenziale stocastica

$$dD(t) = \delta S(t)dt + rD(t)dt. \quad (2.5)$$

Se  $S(t)$  è un moto Browniano geometrico con drift  $\mu \in \mathbb{R}$  e volatilità  $\sigma \in \mathbb{R}^+$  allora per la proprietà di *martingalità* di cui il processo  $S(t) + D(t)$  gode, si ottiene che  $\mu + \delta = r$ , ovvero che

$$\mu = r - \delta, \quad (2.6)$$

e di conseguenza il titolo  $S$  è un moto Browniano geometrico con drift  $r - \delta$  e volatilità  $\sigma$ . In pratica il drift diminuisce di  $\delta$  rispetto al primo caso trattato, in cui i dividendi erano nulli.

## 2.4 Esempi numerici

Vediamo ora qualche esempio pratico di ciò che abbiamo descritto nella sezione precedente.

*Esempio 2.8* (Calcolo del payoff atteso di un'opzione con barriera). Consideriamo un'opzione call con barriera di tipo *down-and-out*, su un mercato con tasso di interesse costante 0,1. Fissiamo a 20 la barriera dell'opzione, a 10 lo

strike e a 5 l'istante di scadenza dell'opzione, decidendo che tra la sottoscrizione e la scadenza il valore del titolo sottostante  $S(t)$  verrà monitorato in 101 istanti equidistanti tra loro. Supponiamo che  $S$  paghi un dividendo costante continuo di 0,05, abbia 0,2 di volatilità, e che al momento della sottoscrizione dell'opzione valga 30. Allora, utilizzando la (2.6),  $S(t)$  è un moto Browniano geometrico, con drift  $0,1-0,05=0,05$ , volatilità 0,2 e valore iniziale 30. Tale moto può essere simulato con quello che sappiamo dal capitolo 1, e iterando la simulazione un numero alto di volte si sfrutta il metodo Monte Carlo per calcolare il valore atteso del payoff. Questa procedura si implementa con il seguente codice Mathematica.

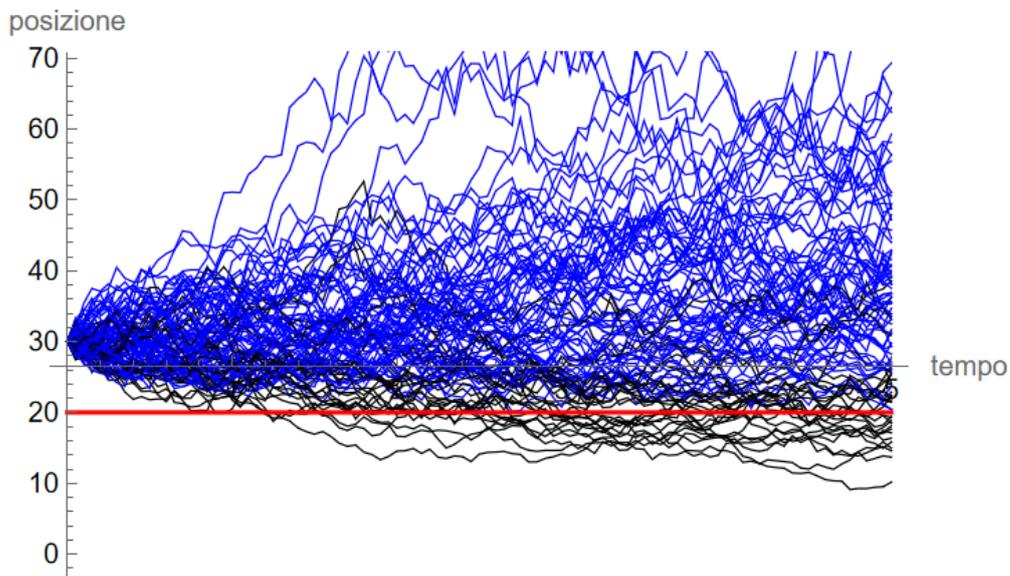
```
(*inizializza i dati*)
r = 0.1; delta = 0.05;
vol = 0.2; att = 30;
b = 20; T = 5;
K = 10; t = Range[0, 5, 0.05];
sim = {}; grafici = {};
(*metodo Monte Carlo*)
For[j = 1, j <= 100, j++,
  S = GeomBrownianPath[r - delta, vol, t, att];
  (*simula il valore del titolo sottostante*)
  superato = If[Min[S] < b, 0, 1];
  (*controlla se la barriera viene superata*)
  grafici = If[Min[S] < b,
    Append[grafici, ListLinePlot[Thread[{t, S}],
      PlotStyle -> Directive[Black, Thickness[0.002]]]],
    Append[grafici, ListLinePlot[Thread[{t, S}],
      PlotStyle -> Directive[Blue, Thickness[0.002]]]]];
  If[val - K > 0, superato*(val - K), 0];
  (*calcola il payoff*)
  sim = Append[sim, p];
];
```

```
grafici = Append[grafici,
ListLinePlot[{{0, b}, {T, b}}, PlotStyle -> Red]];
payoff = Mean[sim];
Show[grafici, PlotRange -> {{0, T}, Automatic},
AxesLabel -> {tempo, posizione}]
Print["Il payoff dell'opzione e': ", payoff];
```

Eseguendo il codice si ottiene la stampa a video rappresentata nell'immagine 2.2, contenente il grafico delle traiettorie simulate del processo  $S$  (in nero quelle che scendono al disotto della barriera, in blu le altre e in rosso la barriera), e il valore atteso del payoff calcolato, che risulterà essere 24,0923. Si noti che, essendo il drift positivo e il valore iniziale di  $S$  più alto della barriera, la maggior parte delle traiettorie non scendono mai sotto la barriera, e sono quindi valide. Per questo motivo il valore atteso del payoff resta abbastanza alto.

*Esempio 2.9* (Calcolo del payoff atteso di una *currency option*). Una *currency option* è un'opzione il cui bene sottostante è una certa quantità di una certa valuta straniera. In questo caso il prezzo della valuta straniera non è altro che il tasso di cambio (quantità di monete locali per moneta straniera). Siccome la moneta straniera guadagna interessi ad un certo tasso costante  $r_f$ , tali interessi possono essere considerati come dei dividendi continui pagati dalla valuta. Detto questo, se  $\sigma$  è la volatilità del tasso di cambio, si può sfruttare la (2.6) e modellare il prezzo della moneta straniera come un moto Browniano geometrico con drift  $r - r_f$ , e volatilità  $\sigma$ , dove  $r$  è il solito tasso privo di rischio atteso del mercato.

Consideriamo una *currency option* di tipo put, su un mercato con tasso privo di rischio atteso 0,5, con la moneta straniera sottostante tale che il suo tasso di interesse è 0,1, la volatilità del tasso di cambio è 0,3 e il valore del tasso di cambio alla sottoscrizione dell'opzione è 2. Fissiamo a 10 sia la scadenza che lo strike dell'opzione, e supponiamo che l'opzione permetta di vendere 8 di queste monete. Allora, per quanto detto, il tasso di cambio può essere modellato come un moto Browniano geometrico, con drift 0,4, volatilità



Il payoff dell'opzione è: 24.0923

Figura 2.2: Output del codice Mathematica per il calcolo del payoff dell'opzione con barriera descritta nell'esempio 2.8

0,3 e valore iniziale 2. Implementando ed eseguendo il codice Mathematica

```
(*Inizializzazione variabili*)
r = 0.5; rf = 0.1;
vol = 0.3; att = 2;
T = 10; K = 10;
q = 8; sim = {};
(*metodo montecarlo*)
For[j = 1, j <= 100000, j++,
  S = GeomBrownianPath[r - rf, vol, {T}, att]
  (*simula il percorso browniano*)
  p = If[K - q*S[[1]] > 0, K - q*S[[1]], 0];
  (*calcola il payoff*)
  sim = Append[sim, p];
```

```
];  
payoff = Mean[sim];  
Print["Il payoff dell'opzione e': ", payoff];
```

si ottiene la stampa a video

```
Il payoff dell'opzione e': 0.0000488327
```

Si noti che il valore atteso del payoff risulta essere molto basso. Questo fatto si spiega perchè nella maggior parte delle simulazioni del valore del tasso di cambio a scadenza, questo supera lo strike, per cui non è conveniente riscattare l'opzione.

*Esempio 2.10* (Calcolo del payoff atteso di una *commodities option*). Una *commodities option* è un'opzione il cui bene sottostante è una materia prima. Le materie prime possono essere viste come beni che pagano dividendi negativi per via dei costi di immagazzinamento, ma allo stesso tempo possedere tali beni porta anche vantaggi, come ad esempio la possibilità di consumarli nel caso vi sia carenza. Per questo possiamo denotare con  $\delta$  la differenza tra un valore che approssima i vantaggi ottenuti dalla materia prima sottostante e i costi di immagazzinamento, e considerare  $\delta$  come i dividendi pagati dal bene in questione. Quindi, denotando con  $r$  il tasso privo di rischio atteso del mercato, e  $\sigma$  la volatilità del valore del bene sottostante, sarà possibile modellare tale valore come un moto Browniano geometrico con drift  $r - \delta$  e volatilità  $\sigma$ .

Consideriamo una *commodities option call* su un mercato con tasso di interesse atteso 0,3. Supponiamo che la materia prima sottostante abbia un costo continuo di immagazzinamento di 0,1, che i vantaggi dati dal possesso del bene siano approssimati con il valore 0,2, che abbia una volatilità di 0,1 e che il suo valore al momento della sottoscrizione dell'opzione sia di 10. Fissiamo a 10 anche l'istante di scadenza dell'opzione, e a 2 il valore dello strike. Per quanto detto, in questo caso il valore della materia prima sarà un moto Browniano geometrico con drift 0,2, volatilità 0,1 e valore iniziale 10. Il payoff atteso di questa opzione si può calcolare con il codice Mathematica

```
(*Inizializzazione variabili*)
r = 0.3; delta = 0.2;
c = 0.1; vol = 0.1;
T = 10; K = 2;
att = 10; sim = {};
(*metodo montecarlo*)
For[j = 1, j <= 100000, j++,
  S = GeomBrownianPath[r-(delta-c), vol, {T}, att];
  (*simula il percorso browniano*)
  p = If[S[[1]] - K > 0, S[[1]] - K, 0];
  (*calcola il payoff*)
  sim = Append[sim, p];
];
payoff = Mean[sim];
Print["Il payoff dell'opzione e': ", payoff];
```

che stamperà a video il seguente risultato:

Il payoff dell'opzione e': 71.9063

Al contrario dell'esempio precedente, in questo caso il payoff risulta essere molto alto, infatti in quasi tutte le simulazioni il valore della materia prima supera lo strike, per cui, trattandosi di una call, riscattare l'opzione è quasi sempre conveniente.

# Bibliografia

- [1] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2003.
- [2] Andrea Pascucci. *Calcolo stocastico per la finanza*. Springer, 2008.
- [3] Andrea Pascucci. *Teoria della Probabilità*. Springer, 2020.



# Ringraziamenti

Inizio ringraziando il mio relatore, prof. Andrea Pascucci, per avermi dato l'opportunità di redigere questa tesi, un compito che si è rivelato per me molto stimolante.

Ringrazio i miei amici e mio fratello, che in questi tre anni (ma in realtà da sempre) sono riusciti a rendere le mie giornate decisamente più allegre e spensierate di quello che altrimenti sarebbero state; ringrazio i miei genitori, senza i quali arrivare a questo traguardo sarebbe stato impossibile; ringrazio le mie nonne, che sono da sempre le mie prime sostenitrici, e ringrazio tutto il resto della mia famiglia.

Vorrei concludere menzionando la mia insegnante di matematica dei primi due anni delle superiori, che purtroppo da qualche mese non c'è più. Essendo stata lei per prima a trasmettermi la passione per la matematica, una parte importante del merito di questo risultato è da attribuire a lei.