

Metodi Monte Carlo in Finanza

LUCIA CARAMELLINO*

Indice

1	Metodi Monte Carlo: generalità	2
2	Simulazione di un moto Browniano e di un moto Browniano geometrico	3
3	Metodi numerici Monte Carlo per la finanza	5
3.1	Prezzo di call e put standard	6
3.2	Opzioni asiatiche	6
3.3	Calcolo numerico della delta: differenze finite e formule di rappresentazione in termini di aspettazione	7
3.3.1	Differenze finite	7
3.3.2	Formule di rappresentazione per le derivate	8
3.4	Copertura dinamica	11
4	Alcuni codici in C	13
4.1	Generatore aleatorio KNUTH	13
4.2	Generazione di una v.a. uniforme su $(0, 1)$	15
4.3	Generazione di una v.a. gaussiana $N(m, \sigma^2)$	15
4.4	Funzione di ripartizione di una gaussiana standard	16

*Dipartimento di Matematica, Università di Roma *Tor Vergata*, via della Ricerca Scientifica, I-00133 Roma, Italy. www.mat.uniroma2.it/~caramell1.

1 Metodi Monte Carlo: generalità

Il metodo Monte Carlo è un metodo **stocastico** per il calcolo numerico di quantità **deterministiche**. L'idea è la seguente.

Supponiamo di voler calcolare una quantità α e per di più supponiamo che α si possa rappresentare come la media di una v.a. X :

$$\alpha = \mathbb{E}(X).$$

Ovviamente, stiamo supponendo anche di lavorare su un opportuno spazio di probabilità $(\Omega, \mathcal{F}, \mathbb{P})$, dove $X : (\Omega, \mathcal{F}) \rightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ è definita e il simbolo \mathbb{E} denota l'aspettazione sotto \mathbb{P} . Indichiamo con Λ la legge di¹ X . Va da sé che X deve essere integrabile. Immaginiamo che X sia anche di quadrato integrabile e sia $\{X^q\}_q$ una successione di v.a. indipendenti, tutte con legge Λ . Allora, per la Legge dei Grandi Numeri, si ha:

$$\hat{\alpha}_Q := \frac{1}{Q} \sum_{q=1}^Q X^q \xrightarrow{Q \rightarrow \infty} \mathbb{E}(X) = \alpha.$$

Dunque, un modo per approssimare α è quello di calcolare la media empirica di un buon numero di v.a. indipendenti con legge uguale alla legge di X . Questo è il metodo Monte Carlo. Osserviamo che:

- nella pratica, questo metodo si implementa in poche righe di programma al calcolatore se è possibile simulare v.a. con legge Λ ;
- la stima $\hat{\alpha}_Q$ di α è soggetta a due errori: il primo conseguente al fatto di fissare un valore (seppur grande) di Q , il secondo dovuto al fatto che $\hat{\alpha}_Q$ è una **variabile aleatoria** (il che, rozzamente, significa che facendo girare il programma più volte, con lo stesso valore per Q , si possono ottenere stime diverse $\hat{\alpha}_Q$ per α).

La “bontà” della stima (ovvero, la velocità di convergenza) si può studiare usando il Teorema del Limite Centrale, che riportiamo nella seguente, qui particolarmente utile, versione:

$$\sqrt{Q} (\hat{\alpha}_Q - \alpha) \xrightarrow{\mathcal{L}} N(0, C)$$

dove il simbolo $\xrightarrow{\mathcal{L}}$ denota la convergenza in legge, $N(0, C)$ è la legge gaussiana su \mathbb{R}^d di media 0 e matrice di covarianza C , e C qui sta per la matrice di covarianza associata al vettore aleatorio $X = (X_1, \dots, X_d)$: $C_{ij} = \text{Cov}(X_i, X_j)$. In particolare, se Q è grande allora vale la seguente approssimazione:

$$\hat{\alpha}_Q \cong \alpha + \frac{1}{\sqrt{Q}} C^{1/2} Z, \tag{1}$$

¹Cioè, $\Lambda(A) = \mathbb{P}(X \in A)$, per $A \in \mathcal{B}(\mathbb{R}^d)$.

con Z v.a. gaussiana standard su \mathbb{R}^d . Ciò significa che l'errore **aleatorio** che si compie approssimando α con $\hat{\alpha}_Q$ è dell'ordine $\frac{1}{\sqrt{Q}} C^{1/2} Z$, con Z gaussiana standard. In altre parole, l'ordine di convergenza di un metodo Monte Carlo è del tipo $1/\sqrt{Q}$, e cioè: è lento! Ciò nonostante, occorre sottolineare che il metodo è estremamente flessibile ed inoltre, come abbiamo appena visto, la velocità di convergenza è indipendente dalla dimensione d (e quest'ultima è, forse, la peculiarità del metodo).

Infine, grazie alla (1), è possibile individuare un intervallo di confidenza (IC) approssimato per α , a livello δ desiderato. Infatti, supponiamo per semplicità che $d = 1$, e quindi

$$\hat{\alpha}_Q \cong \alpha + \frac{\sigma}{\sqrt{Q}} Z, \quad (2)$$

con $Z \sim N(0, 1)$. Fissato un livello δ , un IC bilatero per α , a livello δ , ha quindi estremi

$$\hat{\alpha}_Q \pm \frac{\sigma}{\sqrt{Q}} \phi_{\frac{1+\delta}{2}}, \quad (3)$$

dove $\phi_{\frac{1+\delta}{2}}$ denota il quantile di ordine $(1 + \delta)/2$ della legge gaussiana standard. Ad esempio, per un IC al 95% (cioè $\delta = 0.95$), $\phi_{\frac{1+\delta}{2}} = \phi_{0.975} = 1.96$; se invece $\delta = 0.9$ allora $\phi_{\frac{1+\delta}{2}} = \phi_{0.95} = 1.645$.

Volendo usare un metodo Monte Carlo, nella pratica è cruciale conoscere sia la stima $\hat{\alpha}_Q$ di α sia consegnare all'utente un IC, tipicamente al 95% (cioè, $\delta = 0.95$). Osserviamo però che nella (3) compare σ^2 , cioè la varianza di X , che spesso non si conosce. In tal caso, si sostituisce nella (3) la tipica stima $\hat{\sigma}_Q^2$ (non distorta) per σ^2 :

$$\hat{\sigma}_Q^2 = \frac{1}{Q-1} \sum_{q=1}^Q (X^q - \hat{\alpha}_Q)^2 = \frac{1}{Q-1} \sum_{q=1}^Q (X^q)^2 - \hat{\alpha}_Q^2,$$

cosicché l'IC approssimato al 95% ha estremi

$$\hat{\alpha}_Q \pm 1.96 \sqrt{\frac{\hat{\sigma}_Q^2}{Q}}.$$

2 Simulazione di un moto Browniano e di un moto Browniano geometrico

La simulazione di un moto Browniano standard B su \mathbb{R}^d si basa sulle seguenti proprietà che lo caratterizzano:

- $B_0 = 0$;
- B è a incrementi indipendenti;

- gli incrementi $B_t - B_s$, $t > s \geq 0$, sono gaussiani, di media 0 e matrice di covarianza $(t - s) I_{d \times d}$, essendo $I_{d \times d}$ la matrice identica $d \times d$.

Dunque, per generare una traiettoria Browniana sull'intervallo temporale $[0, T]$, dapprima occorre suddividere $[0, T]$ in sottointervalli di ampiezza piccola. Ad esempio, fissato $h = T/N$, indichiamo con $t_k = kh$, $k = 1, \dots, N$, gli estremi di questi intervalli. Per simulare B ai tempi t_k , basterà prendere N v.a. gaussiane standard e indipendenti U_1, \dots, U_N e porre:

$$B_0 = 0 \quad \text{e per } k = 1, 2, \dots, N, \quad B_{t_k} = B_{t_{k-1}} + \sqrt{h} U_k.$$

In altre parole,

$$B_{t_k} = \sqrt{h} \sum_{j=1}^k U_j, \quad k = 0, 1, \dots, N$$

(dove, al solito, definiamo $\sum_{j=1}^0 (\cdot) = 0$). Quest'ultima rappresentazione è nota come l'approssimazione del moto Browniano con una passeggiata aleatoria gaussiana. Per simulare un moto Browniano basta quindi essere in grado di generare v.a. gaussiane. L'algoritmo più noto in letteratura è quello di Box-Muller, il quale si basa sul seguente (semplice) risultato:

[Box-Muller] *date due variabili aleatorie Z_1, Z_2 distribuite uniformemente in $(0, 1)$, allora*

$$\begin{aligned} V_1 &= \sqrt{-2 \log Z_1} \cos(2\pi Z_2) \\ V_2 &= \sqrt{-2 \log Z_1} \sin(2\pi Z_2) \end{aligned}$$

sono variabili aleatorie normali standard indipendenti.

Nel paragrafo 4.3 è proposto un codice in C per la simulazione di una v.a. gaussiana con l'algoritmo di Box-Muller. Ricordiamo infine che un vettore gaussiano standard su \mathbb{R}^d è formato semplicemente da d v.a. gaussiane reali standard e indipendenti.

La generazione del moto Browniano geometrico è a questo punto immediata. Ricordiamo che un processo S_t su \mathbb{R}^d segue un moto Browniano geometrico, o equivalentemente si evolve secondo il modello di Black e Scholes, se

$$\begin{aligned} dS_t^i &= \mu^i S_t^i dt + \sum_{j=1}^d \sigma_j^i S_t^i dB^j(t) \\ S_0^i &= x^i, \end{aligned} \quad i = 1, \dots, d, \quad (4)$$

dove σ denota la volatilità, e rappresenta un matrice $d \times d$ invertibile. È facile vedere che, per $i = 1, \dots, d$,

$$S_t^i = x^i \exp \left(\left(\mu^i - \frac{1}{2} \sum_{j=1}^d \sigma_j^{i^2} \right) t + \sum_{j=1}^d \sigma_j^i B_t^j \right),$$

cosicché una simulazione di S_t segue immediatamente da una generazione del moto Browniano: fissato $h = T/N$ e detti $t_k = kh$, $k = 1, \dots, N$, si ha

$S_0^i = x$ e per $k = 1, 2, \dots, N$,

$$S_{t_k} = S_{t_{k-1}} \exp\left(\left(\mu^i - \frac{1}{2} \sum_{j=1}^d \sigma_j^2\right)h + \sqrt{h} \sum_{j=1}^d \sigma_j^i U_k^j\right),$$

per $i = 1, \dots, d$, dove U_1, \dots, U_N sono N v.a. i.i.d., gaussiane standard su \mathbb{R}^d ($U_k = (U_k^1, \dots, U_k^d)$).

Va da sé che se si vuole simulare S_t sotto la misura di rischio neutro (di martingala equivalente), occorre scegliere

$$\mu^i = r, \quad \text{per ogni } i,$$

dove r denota il tasso di interesse istantaneo. In caso contrario, μ sarà un vettore formato da quantità determinate dal mercato finanziario.

3 Metodi numerici Monte Carlo per la finanza

I metodi Monte Carlo trovano nella finanza una naturale applicazione per la risoluzione numerica dei problemi di prezzaggio e copertura delle opzioni. Infatti, supponiamo di essere in un mercato completo, in cui S_t^0 e S_t^1, \dots, S_t^d denotano i processi di prezzo del titolo non rischioso e dei d titoli rischiosi. Supponiamo che $(S_t^1, \dots, S_t^d)_t$ si evolvano seguendo il modello di Black e Scholes su \mathbb{R}^d , cioè secondo la (4), e che

$$dS_t^0 = rS_t^0 dt, \quad S_0^0 = 1,$$

dove al solito r denota il tasso di interesse istantaneo, supposto costante. Indichiamo anche con \mathcal{F}_t la filtrazione, completata con gli insiemi di probabilità nulla, sotto la quale il moto Browniano B che compare in (4) risulta un moto Browniano standard.

La formula del prezzo di un'opzione europea di payoff h e maturità T è data da

$$\text{prezzo al tempo } t = \mathbb{E}^*(h e^{-r(T-t)} \mid \mathcal{F}_t),$$

dove \mathbb{E}^* denota l'aspettazione valutata sotto la misura di rischio neutro (in sostanza, basta porre $\mu^i = r$, per ogni i , nella (4)). Se supponiamo che il payoff h sia tale che

$$P(t, S_t) = \mathbb{E}^*(h e^{-r(T-t)} \mid \mathcal{F}_t),$$

per qualche funzione $P = P(t, x)$ abbastanza regolare, allora sappiamo anche che il portafoglio di copertura è dato da

$$V_t = H_t^0 S_t^0 + \sum_{i=1}^d H_t^i S_t^i, \quad \text{con}$$

$$H_t^i = \partial_{x_i} P(t, S_t), \quad i = 1, \dots, d, \quad \text{e } H_t^0 = \left(P(t, S_t) - \sum_{i=1}^d H_t^i S_t^i\right) e^{-rt}.$$

La rappresentazione del prezzo in termini di una media suggerisce immediatamente l'uso di metodi Monte Carlo.

3.1 Prezzo di call e put standard

Supponiamo per semplicità che $d = 1$ e di avere a che fare con un'opzione call oppure put, cioè

$$h = h_{\text{call}} = (S_T - K)_+ \quad \text{oppure} \quad h = h_{\text{put}} = (K - S_T)_+$$

In tal caso, sappiamo che la funzione P che dà il prezzo è data da

$$\begin{aligned} P_{\text{call}}(t, x) &= x \Phi(d_1(T-t, x)) - K e^{-r(T-t)} \Phi(d_2(T-t, x)) \\ P_{\text{put}}(t, x) &= K e^{-r(T-t)} \Phi(-d_2(T-t, x)) - x \Phi(-d_1(T-t, x)) \end{aligned} \quad (5)$$

dove

$$d_1(s, x) = \frac{\ln(x/K) + (r + \sigma^2/2)s}{\sigma \sqrt{s}} \quad \text{e} \quad d_2(s, x) = d_1(s, x) - \sigma \sqrt{s}$$

e con Φ la funzione di ripartizione di una gaussiana standard (una implementazione per il calcolo di Φ è proposta nel paragrafo 4.4). Anche la copertura ha una formula esplicita, data da

$$\begin{aligned} \Delta_{\text{call}}(t, x) &= \partial_x P_{\text{call}}(t, x) = \Phi(d_1(T-t, x)) \\ \Delta_{\text{put}}(t, x) &= \partial_x P_{\text{put}}(t, x) = \Phi(d_1(T-t, x)) - 1 \end{aligned} \quad (6)$$

In tal caso quindi non occorre appellarsi a metodi numerici: le formule esistono, basta usarle! Ciò nonostante, proponiamo il seguente esercizio.

Esercizio 3.1. Calcolare numericamente, con il metodo Monte Carlo, il prezzo di una call e di una put standard in 0, con IC al 95%, e studiare empiricamente la convergenza ai risultati forniti dalle formule esatte quando il numero di simulazioni Q tende a $+\infty$.

A titolo di esempio, si considerino i seguenti parametri: $S_0 = K = 100$, $T = 1$ anno, $r = 0.05$, $\sigma = 0.2$.

3.2 Opzioni asiatiche

Diverso è il caso delle opzioni asiatiche, quando cioè non esistono formule esatte per calcolare prezzo e/o copertura. Ricordiamo che un'opzione asiatica ha payoff h che dipende dalla media temporale $\int_0^T S_t dt/T$ del sottostante su tutto l'intervallo $[0, T]$. Ad esempio, una call asiatica a maturità T e con prezzo di esercizio K ha payoff

$$h = \left(\frac{1}{T} \int_0^T S_t dt - K \right)_+$$

Si può far vedere² che in questo caso il prezzo è dato da una funzione F di t , S_t ed anche $\int_0^t S_u du$, anche se la funzione F non si riesce a scrivere in “formula chiusa”, ma si può rappresentare a sua volta come l'aspettazione di una opportuna media condizionata.

²Si veda, ad esempio, il Problema 7, a pag. 91, del testo di Lamberton e Lapeyre

Esercizio 3.2. Usando semplicemente il payoff e la rappresentazione in forma di aspettazione del prezzo, calcolare numericamente, con il metodo Monte Carlo, il prezzo in 0 di una call asiatica, con IC al 95%. Si considerino gli stessi parametri suggeriti nell'Esercizio 3.1.

3.3 Calcolo numerico della delta: differenze finite e formule di rappresentazione in termini di aspettazione

Consideriamo il modello (4), di Black e Scholes, e supponiamo di lavorare con un'opzione a maturità T di payoff

$$h = f(S_T) \equiv f(S_T^1, \dots, S_T^d)$$

dove f è una funzione a crescita polinomiale. In tal caso, usando la markovianità delle diffusions, sappiamo che il prezzo è determinato da $P(t, S_t)$, dove

$$P(t, x) = e^{-r(T-t)} \mathbb{E}^*(f(x^1 Z_{T-t}^1, \dots, x^d Z_{T-t}^d))$$

dove

$$Z_{T-t}^i = \exp\left(\left(r - \frac{1}{2} \sum_{j=1}^d \sigma_j^2\right)(T-t) + \sum_{j=1}^d \sigma_j^i (B_T^j - B_t^j)\right), \quad i = 1, \dots, d.$$

La funzione (d -dimensionale) delta, che dà la copertura dell'opzione, è quindi

$$\Delta_i(t, x) = \partial_{x^i} P(t, x) = e^{-r(T-t)} \partial_{x^i} \mathbb{E}^*(f(x^1 Z_{T-t}^1, \dots, x^d Z_{T-t}^d)).$$

3.3.1 Differenze finite

Un primo modo per calcolare numericamente la delta è quello di usare le differenze finite, approssimare, cioè, la derivata parziale con il rapporto incrementale sull' i -esima direzione:

$$\begin{aligned} \Delta_i(t, x) &\simeq \Delta_i^\delta(t, x) \\ &= \frac{e^{-r(T-t)}}{2\delta} \left(\mathbb{E}^*(f(x^1 Z_{T-t}^1, \dots, (x^i + \delta) Z_{T-t}^i, \dots, x^d Z_{T-t}^d)) \right. \\ &\quad \left. - \mathbb{E}^*(f(x^1 Z_{T-t}^1, \dots, (x^i - \delta) Z_{T-t}^i, \dots, x^d Z_{T-t}^d)) \right) \end{aligned}$$

Ovviamente, nella pratica δ è scelto "piccolo" (cioè piccolo percentualmente a x). Va da sé che, se si conoscono proprietà di ulteriore regolarità della funzione Δ_i , ad esempio derivabile con una qualche stima dell'andamento della derivata, usando la formula di Taylor δ si può scegliere in modo da limitare l'errore che si commette nell'approssimazione della derivata con le differenze finite.

A questa prima approssimazione, di tipo analitico, si somma un'altra approssimazione, di tipo Monte Carlo: le medie sopra coinvolte si approssimano a loro volta con la media empirica valutata su Q simulazioni, cioè $\Delta_i(t, x) \simeq \Delta_i^\delta(t, x) \simeq \Delta_i^{\delta, Q}(t, x)$, dove

$$\begin{aligned} \Delta_i^{\delta, Q}(t, x) e^{r(T-t)} &= \\ &= \frac{1}{Q} \sum_{q=1}^Q \frac{1}{2\delta} \left(f(x^1 Z_{T-t}^{1,q}, \dots, (x^i + \delta) Z_{T-t}^{i,q}, \dots, x^d Z_{T-t}^{d,q}) \right. \\ &\quad \left. - f(x^1 Z_{T-t}^{1,q}, \dots, (x^i - \delta) Z_{T-t}^{i,q}, \dots, x^d Z_{T-t}^{d,q}) \right) \end{aligned} \quad (7)$$

dove $Z_{T-t}^{j,q}$ denota la q -esima simulazione della v.a. Z_{T-t}^j , per $j = 1 \dots, d$. Osserviamo che, a priori, si potrebbe anche usare la stima

$$\begin{aligned} \Delta_i^{\delta, Q}(t, x) e^{r(T-t)} &= \\ &= \frac{1}{2\delta} \left[\frac{1}{Q_1} \sum_{q_1=1}^{Q_1} f(x^1 Z_{T-t}^{1,q_1}, \dots, (x^i + \delta) Z_{T-t}^{i,q_1}, \dots, x^d Z_{T-t}^{d,q_1}) \right. \\ &\quad \left. - \frac{1}{Q_2} \sum_{q_2=1}^{Q_2} f(x^1 Z_{T-t}^{1,q_2}, \dots, (x^i - \delta) Z_{T-t}^{i,q_2}, \dots, x^d Z_{T-t}^{d,q_2}) \right] \end{aligned}$$

ovvero fare uso di Q_1 simulazioni per approssimare la prima media e di Q_2 simulazioni, eventualmente diverse dalle prime, per il calcolo della seconda media. Si può far vedere³ che la stima in (7) è più stabile, cosa che in questo caso è piuttosto importante perché già l'approssimazione della derivata con le differenze finite dà una certa instabilità, cui occorrerebbe non assommarne altra.

3.3.2 Formule di rappresentazione per le derivate

Vediamo ora un altro modo per calcolare numericamente la delta. L'idea è quella di rappresentare le derivate come aspettative. In tal modo, si bypassa l'approssimazione analitica, cioè tramite le differenze finite, e si considera direttamente l'approssimazione Monte Carlo. Questo tipo di rappresentazione, oggi "alla moda" nella ricerca matematica in Finanza, si può determinare facendo uso del calcolo di Malliavin, una teoria di calcolo differenziale stocastico piuttosto complicata. Ma se è il modello di Black e Scholes quello di riferimento, si possono ottenere formule utili anche senza l'ausilio del calcolo di Malliavin. Vediamo come.

Indichiamo con $p(t, T, x, y)$ la densità di transizione (che esiste, lo sappiamo e comunque la calcoleremo tra breve) di $S_T^{t,x}$, cioè della soluzione al tempo

³Si veda, ad esempio, il bel libro di Glasserman.

T dell'eds che compare in (4) che al tempo $t < T$ parte dal punto x , cioè con dato iniziale $S_t^{t,x} = x$. Consideriamo la seguente scrittura

$$P(t, x) = e^{-r(T-t)} \mathbb{E}^*(f(S_T^{t,x})).$$

Allora, possiamo scrivere

$$\begin{aligned} \Delta_i(t, x) &= e^{-r(T-t)} \partial_{x^i} \mathbb{E}^*(f(S_T^{t,x})) = e^{-r(T-t)} \partial_{x^i} \int_{\mathbb{R}_+^d} f(y) p(t, T, x, y) dy \\ &= e^{-r(T-t)} \int_{\mathbb{R}_+^d} f(y) \partial_{x^i} p(t, T, x, y) dy \\ &= e^{-r(T-t)} \int_{\mathbb{R}_+^d} f(y) \frac{\partial_{x^i} p(t, T, x, y)}{p(t, T, x, y)} p(t, T, x, y) dy \\ &= e^{-r(T-t)} \int_{\mathbb{R}_+^d} f(y) \partial_{x^i} \ln p(t, T, x, y) p(t, T, x, y) dy \end{aligned}$$

da cui si ottiene

$$\Delta_i(t, x) = e^{-r(T-t)} \mathbb{E}^*(f(S_T^{t,x}) \Theta_i), \quad \text{dove } \Theta_i = \partial_{x^i} \ln p(t, T, x, y) \Big|_{y=S_T^{t,x}}.$$

Prima di proseguire, osserviamo che

- quanto scritto sopra è corretto se è lecito il passaggio della derivata sotto segno di integrale. Una condizione sufficiente è che la v.a. $f(S_T^{t,x}) \Theta_i$ sia integrabile, e ciò è vero, come vedremo in seguito quando scriveremo esplicitamente il peso Θ_i .
- A priori, non occorre che il modello di riferimento sia Black e Scholes: semplicemente, basta l'esistenza della densità di transizione $p(t, T, x, y)$. Il problema è però che allo scopo di scrivere il peso Θ_i occorre poter scrivere $\ln p(t, T, x, y)$, cioè conoscere esplicitamente la densità di transizione, il che è possibile solo in pochissimi casi, tra cui appunto il moto Browniano geometrico.
- Vale la pena di sottolineare l'importanza della formula appena trovata: scrivere la derivata direttamente come un'aspettazione consente l'uso immediato del Monte Carlo, cioè

$$\Delta_i(t, x) \cong \frac{1}{Q} \sum_{q=1}^Q f(S_T^{t,x,q}) \Theta_i^q$$

dove $S_T^{t,x,q}$ e Θ_i^q denotano rispettivamente la q -esima generazione di $S_T^{t,x}$ e di Θ_i . Dunque, si elide completamente l'approssimazione (analitica) della derivata con le differenze finite.

- Infine, il peso Θ_i è indipendente da f : vuol dire che, una volta determinato, va bene per qualsiasi tipo di opzione di payoff della forma $h = f(S_T)$, con f a crescita polinomiale.

Per concludere questa sezione, diamo un'espressione esplicita al peso Θ_i . Possiamo scrivere

$$S_T^{t,x,i} = x^i e^{m_i + \sqrt{T-t}(\sigma Z)^i},$$

con $m^i = (r - \sum_{j=1}^d \sigma_j^2/2)(T-t)$ e $Z = (B_T - B_t)/\sqrt{T-t} \sim N(0, I_{d \times d})$. Dunque, $y \mapsto p(t, T, x, y)$ è la densità del vettore aleatorio $\psi(Z)$, dove

$$\psi^i(z) = x^i e^{m^i + \sqrt{T-t}(\sigma z)^i}, \quad i = 1, \dots, d.$$

Posto $\phi = \psi^{-1}$, usando il Teorema del cambio di variabile si ha

$$p(t, T, x, y) = \left[\prod_{i=1}^d \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}|\phi^i(y)|^2} \right) \right] \cdot |\det D\phi(y)|,$$

dove $D\phi(y)$ denota la matrice Jacobiana di ϕ e $\phi(y)$ ha componenti date da

$$\phi^i(y) = \frac{1}{\sqrt{T-t}} \sum_{j=1}^d A_{ij} \left(\ln \frac{y^j}{x^j} - m^j \right), \quad i = 1, \dots, d,$$

essendo A la matrice inversa di σ . Poiché

$$\partial_{y^k} \phi^i(y) = \frac{1}{\sqrt{T-t}} A_{ik} y^k,$$

al variare di x otteniamo

$$x \mapsto \ln p(t, T, x, y) = -\frac{1}{2(T-t)} \sum_{k=1}^d \left(\sum_{j=1}^d A_{kj} \left(\ln \frac{y^j}{x^j} - m^j \right) \right)^2 + const,$$

dove $const$ denota "qualcosa" che non dipende da x . Allora,

$$\partial_{x^i} \ln p(t, T, x, y) = \frac{1}{x^i(T-t)} \sum_{j=1}^d \left(\sum_{k=1}^d A_{kj} A_{ki} \right) \left(\ln \frac{y^j}{x^j} - m^j \right).$$

Sostituendo y^j con $S_T^{t,x,j}$ si ha

$$\ln \frac{S_T^{t,x,j}}{x^j} - m^j = \sum_{\ell=1}^d \sigma_\ell^j (B_T^\ell - B_t^\ell)$$

e ricordando che $A = \sigma^{-1}$, si ottiene immediatamente

$$\Theta_i = \partial_{x^i} \ln p(t, T, x, y) \Big|_{y=S_T^{t,x}} = \frac{((B_T - B_t)^t \cdot A)^i}{x^i(T-t)}, \quad i = 1, \dots, d. \quad (8)$$

dove $((B_T - B_t)^t \cdot A)^i$ denota l' i -esima entrata del prodotto tra il vettore (riga) $(B_T - B_t)^t$ e la matrice A . Se σ è diagonale, cioè i d processi di prezzo sono indipendenti, la (8) diviene

$$\Theta_i = \frac{B_T^i - B_t^i}{x^i \sigma_i^i(T-t)}, \quad i = 1, \dots, d,$$

e in dimensione $d = 1$,

$$\Theta = \frac{B_T - B_t}{x\sigma(T-t)}, \quad i = 1, \dots, d.$$

Rimane solo da verificare che effettivamente $f(S_T^{t,x})\Theta_i$ è integrabile. Supponendo f a crescita polinomiale, si ha

$$|f(S_T^{t,x})\theta_i| \leq \text{const}(1 + (S_T^{t,x})^p)|B_T^i - B_t^i| \leq \text{const}(1 + e^{c|B_T - B_t|})|B_T^i - B_t^i|$$

dove const e c denotano costanti positive opportune. Poiché $B_T - B_t$ è una v.a. gaussiana, la v.a. $(1 + e^{c|B_T - B_t|})|B_T^i - B_t^i|$ è integrabile e la tesi è immediata.

Esercizio 3.3. Calcolare numericamente, con il metodo Monte Carlo, la copertura di una call e di una put standard in 0, con IC al 95%, sia con il metodo alle differenze finite sia con la formula di rappresentazione della derivata. Studiare empiricamente la convergenza ai risultati forniti dalle formule esatte quando il numero di simulazioni Q tende a $+\infty$.

Si considerino gli stessi parametri dell'esercizio 3.1 e, per quanto riguarda il metodo alle differenze finite, si prenda $\delta = S_0 10^{-3}$ (si modifichi eventualmente δ nello studio empirico della convergenza).

3.4 Copertura dinamica

In questo paragrafo vedremo come vengono usate nella pratica le formule del prezzo e della copertura studiate nei modelli continui. Per semplicità, supponiamo $d = 1$, che il processo di prezzo del titolo rischioso si evolva seguendo il modello di Black e Scholes (4) e supponiamo di lavorare con un'opzione call standard, di maturità T e prezzo di esercizio K . Seguiremo il punto di vista del venditore dell'opzione, che quindi, una volta avuto il premio iniziale, deve costruirsi man mano il portafoglio replicante.

Il problema fondamentale con le formule trovate, basate su un modello a tempo continuo, è che "il tempo continuo" nella pratica non esiste, cioè non è possibile fare diverse operazioni nello stesso istante. Dunque, il venditore dell'opzione si costruisce un portafoglio che sarà un'approssimazione del portafoglio replicante determinato nella teoria. Quindi, innanzitutto deve fissare delle date in cui gestirà le sue operazioni di riaggiustamento delle quote del titolo rischioso e non. Ciò significa suddividere l'intervallo temporale $[0, T]$ in sotto intervalli di estremi $0 = t_0 < t_1 < \dots < t_N = T$. Ad

esempio, posto $h = T/N$, $t_k = kh$, $k = 0, 1, \dots, N$. Il portafoglio che si troverà a gestire è quindi un'approssimazione V^h di quello teorico V , dato da

$$V_t = P(t, S_t) = \Delta(t, S_t) S_t + \Delta_0(t, S_t) S_t^0.$$

Passo 0 All'istante iniziale $t_0 = 0$, si riceve il compenso della vendita dell'opzione, quindi

$$V_0^h = V_0 = P(0, S_0).$$

Occorre però anche calcolare le quote di titolo rischioso e non rischioso che dovrebbero essere investite, date rispettivamente da

$$\Delta_0 = \Delta(0, S_0) \quad \text{e} \quad \Delta_0^0 = V_0 - \Delta_0 S_0.$$

Il condizionale scelto sopra è d'obbligo: in teoria, l'investimento è istantaneo, e ciò non è ovviamente possibile nella realtà. Quindi, queste quote saranno utilizzate nell'attimo successivo, cioè al tempo t_1 .

Passo 1 All'istante t_1 , dapprima si aggiorna il portafoglio, usando le quote determinate al tempo 0:

$$V_{t_1}^h = \Delta_0 S_{t_1} + \Delta_0^0 e^{rt_1} = \Delta_0 S_{t_1} + (V_0 - \Delta_0 S_0) e^{rh}.$$

A questo punto, si aggiornano le quote:

$$\Delta_{t_1} = \Delta(t_1, S_{t_1}) \quad \text{e} \quad \Delta_{t_1}^0 = (V_{t_1}^h - \Delta_{t_1} S_{t_1}) e^{-rt_1}.$$

⋮

Passo k All'istante t_k , si conoscono le quote

$$\Delta_{t_{k-1}} \quad \text{e} \quad \Delta_{t_{k-1}}^0 = (V_{t_{k-1}}^h - \Delta_{t_{k-1}} S_{t_{k-1}}) e^{-rt_{k-1}}$$

da investire rispettivamente nel titolo rischioso e nel titolo non rischioso. Il portafoglio quindi è

$$V_{t_k}^h = \Delta_{t_{k-1}} S_{t_k} + (V_{t_{k-1}}^h - \Delta_{t_{k-1}} S_{t_{k-1}}) e^{rh}$$

e le quote con cui lavorare al passo $k + 1 \leq N$ sono aggiornate nel modo solito:

$$\Delta_{t_k} = \Delta(t_k, S_{t_k}) \quad \text{e} \quad \Delta_{t_k}^0 = (V_{t_k}^h - \Delta_{t_k} S_{t_k}) e^{-rt_k}.$$

⋮

Passo N All'istante finale $t_N = T$, il portafoglio è quindi

$$V_T^h = \Delta_{t_{N-1}} S_{t_N} + (V_{t_{N-1}}^h - \Delta_{t_{N-1}} S_T) e^{rh}$$

e, con buona approssimazione, per h piccolo dev'essere

$$V_T^h \cong V_T = \text{payoff} = (S_T - K)_+$$

Osserviamo che, di fatto, abbiamo approssimato la strategia replicante, per di più in modo predicibile, com'è naturale che sia in finanza. Ovviamente, nella pratica spesso non si conosce esplicitamente la funzione Δ , e quindi si procede con un'ulteriore approssimazione numerica (ad esempio, come spiegato nella sezione precedente).

Esercizio 3.4. Implementare al calcolatore l'algoritmo della copertura dinamica appena descritto, nel caso di una call. Ovviamente, occorre simulare una possibile traiettoria del processo di prezzo e, per verificare che la (quasi) uguaglianza finale vale per davvero, si sconsiglia (eufemismo!!!) di simulare il prezzo sotto la misura di rischio neutro.

Si considerino gli stessi parametri suggeriti nell'esercizio 3.1 e le formule del prezzo e della copertura della call come in (5) e (6).

4 Alcuni codici in C

Presentiamo alcuni codici di procedure utili per la risoluzione degli esercizi. È possibile scaricare file di testo contenenti queste procedure all'indirizzo <http://www.mat.uniroma2.it/~caramell/did.0506/mmmf.htm>

4.1 Generatore aleatorio KNUTH

Knuth è un generatore di numeri aleatori su $(0,1)$, molto usato e non standard in C.

```
void KNUTH(double *sample) {
    static long M= 1000000000;
    static long SEED= 161803398;

    /* Initialize the sequence with a positive seed */
    static alea= 1;
    static int inc1, inc2;
    static long t_alea[56];
    long X_n, y_k;
    int i, ii, l;
```

```

/* First call to the sequence */
if(counter == 1)
{
    X_n= SEED- alea;
    X_n%= M;
    t_alea[55]= X_n;
    y_k= 1;
    /* Initialization of the table */
    for(i= 1; i<= 54; i++)
    {
        ii= (21*i)%55; /* 21 was chosen to alleviate initial
                        nonrandomness problems */
        t_alea[ii]= y_k;
        y_k= X_n - y_k;
        if(y_k < 0)
            y_k+= M;
        X_n= t_alea[ii];
    }

    /* Randomization of the elements of the table */
    for(l= 1; l<= 4; l++)
    {
        for(i= 1; i<= 55; i++)
        {
            t_alea[i]-= t_alea[1+(i+30)%55];
            if(t_alea[i] < 0)
                t_alea[i]+= M;
        }
    }
    inc1= 0;
    inc2= 31;
        /* 31 is a special value of Knuth: 31= 55-24 */
    alea= 1;
}

counter+= 1;

/* For each call to the sequence,
    computation of a new point */
if(++inc1 == 56)
    inc1= 1;
if(++inc2 == 56)
    inc2= 1;
/* Subtractive method*/

```

```

X_n= t_alea[inc1] - t_alea[inc2];

if(X_n < 0)
    X_n+= M;
t_alea[inc1]= X_n;
/* Normalized value */
*sample=X_n*INV_M;

return;
}

```

4.2 Generazione di una v.a. uniforme su $(0, 1)$

- Usando il generatore standard di C:

```

double nrandom(void) {
    return ((double)rand()/(double)RAND_MAX);
}

```

- Usando Knuth:

```

double nrandom(void){
    double random_number;
    return KNUTH(&random_number);
}

```

4.3 Generazione di una v.a. gaussiana $N(m, \sigma^2)$

```

double normale(double m, double sigma2) {
    double Z1, Z2; double X_st, X;
    double pigreco = 3.14159265358979323846;
    Z1 = nrandom();
    Z2 = nrandom();
    X_st = sqrt(-2*log(Z1)) * cos(2*pigreco*Z2);
    X = m + sqrt(sigma2) * X_st;
    return (X);
}

```

La procedura `nrandom` è nel paragrafo 4.2. Osserviamo che, in linea di principio, si generano due v.a. normali indipendenti.

4.4 Funzione di ripartizione di una gaussiana standard

```
/*One-Dimensional Normal Law.
Cumulative distribution function.
Abramowitz, Milton et Stegun,
Handbook of MathematicalFunctions, 1968,
Dover Publication, New York, page 932 (26.2.18).
Precision 10-7*/

double N(double x) {
const double p= 0.2316419;
const double b1= 0.319381530;
const double b2= -0.356563782;
const double b3= 1.781477937;
const double b4= -1.821255978;
const double b5= 1.330274429;
const double one_over_twopi= 0.39894228;

double t;

if(x >= 0.0)
{
t = 1.0 / ( 1.0 + p * x );
return (1.0 - one_over_twopi * exp( -x * x / 2.0 ) *
t * ( t *( t * ( t * ( t * b5 + b4 ) + b3 ) + b2 )
+ b1 ));
}
else
{/* x < 0 */
t = 1.0 / ( 1.0 - p * x );
return ( one_over_twopi * exp( -x * x / 2.0 ) *
t * ( t *( t * ( t * ( t * b5 + b4 ) + b3 ) + b2 )
+ b1 ));
}
}
```