



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI ECONOMIA “GIORGIO FUÀ”

Corso di Laurea Magistrale in Scienze Economiche e Finanziarie

*Pricing delle opzioni finanziarie:
modelli teorici e implementazione numerica*

*The Pricing of financial options:
Theory and Numerical Methods*

Relatore:
Chiar.mo Prof.
Riccardo De Blasis

Tesi di Laurea di:
Passerini Gaia

Anno Accademico 2025 - 2026

Indice

1	Modello dell'albero binomiale	8
1.1	Definizione formale del modello	9
1.2	Tecniche risolutive	12
2	Processi di Itô	16
2.1	Dalla discretizzazione al tempo continuo	16
2.2	Definizione formale del modello	19
3	Modello di Black-Scholes-Merton	21
3.1	Definizione formale del modello	23
3.2	Tecniche risolutive	25
3.3	Generalizzazione del modello: parametri variabili	27
4	Metodo delle differenze finite	32
4.1	Definizione formale del modello	33
4.2	Tecniche risolutive - Metodo Implicito delle differenze finite . . .	34
4.3	Tecniche risolutive - Metodo Esplicito delle differenze finite . . .	36
4.4	Approccio Informatico al Metodo delle differenze finite	37
5	Metodo Monte Carlo	39
5.1	Definizione formale del modello	40
5.2	Tecniche risolutive	41
6	Applicazione e analisi comparativa dei modelli	42
6.1	Logica algoritmica e implementazione del software	42

6.2	Analisi critica dei risultati ottenuti	44
6.3	Logica algoritmica: il caso della Put Americana	48
6.4	Analisi critica dei risultati ottenuti	49
6.5	Analisi comparativa tra put europea e put americana	55
7	Analisi empirica sul titolo Apple	58
7.1	Calibrazione dei parametri: il caso della call europea	58
7.2	Analisi critica dei risultati ottenuti	61
7.3	Analisi della Vega nei diversi modelli di pricing	66
7.4	Calibrazione dei parametri: il caso della put americana	67
7.5	Analisi critica dei risultati ottenuti	68
7.6	Analisi di Theta nei diversi modelli di pricing	71

Introduzione

Questo elaborato è dedicato allo studio del pricing delle opzioni, uno dei temi centrali della finanza quantitativa moderna.

Le opzioni appartengono alla categoria degli strumenti finanziari derivati, ossia contratti il cui valore dipende dall'andamento di un'attività sottostante. Quest'ultima può essere rappresentata da diverse tipologie di asset, tra cui azioni, ETF, materie prime o altri strumenti finanziari. Il valore del derivato risulta quindi funzione del valore assunto dall'attività sottostante.

Dopo aver definito la natura e le caratteristiche di tali strumenti, è opportuno richiamare brevemente i principali contesti nei quali essi vengono negoziati. In generale, i derivati possono essere scambiati sia nei mercati regolamentati, come il Chicago Board Options Exchange (CBOE), uno dei più importanti mercati mondiali per la negoziazione di opzioni, sia nei mercati over-the-counter (OTC) ossia mercati non regolamentati in cui le transazioni avvengono direttamente tra le controparti. Le diverse sedi di negoziazione comportano variazioni nelle procedure di scambio, nei meccanismi di compensazione e nei requisiti contrattuali, senza tuttavia modificare la struttura fondamentale dei contratti.

Per una panoramica esaustiva su tali tematiche, dai requisiti contrattuali alle convenzioni di mercato comunemente adottate nella teoria del pricing, si rimanda ai lavori fondamentali di Hull (Hull 2018, Hull & White 1988).

Tra le varie tipologie di strumenti citati, particolare rilevanza assume la categoria delle opzioni, oggetto principale del presente elaborato. Queste ultime sono quindi un derivato,

in cui si ha il diritto di acquistare o vendere un'attività sottostante ad un prezzo specifico (prezzo d'esercizio) in una data futura prestabilita. In particolare tali derivati si dividono in **opzioni americane**, in cui il pagamento può avvenire entro una certa data e quelle **europee**, dove il pagamento può avvenire unicamente nella data specificata (expiration date).

Tale distinzione assume un ruolo centrale nella teoria della valutazione, poiché determina differenti caratteristiche di esercizio e, conseguentemente, differenti metodologie di pricing.

Alla luce delle caratteristiche appena descritte, è importante sottolineare l'unicità di questo strumento derivato che si differenzia dagli altri, come futures o forwards, in quanto offre al portatore il diritto di usufruire o meno del contratto. Negli altri derivati vige l'obbligo di compravendita, non si ha un diritto ad esercitare o meno.

Nelle opzioni l'esercizio avviene solo quando conveniente, eliminando la rischiosità associata all'oscillazione dei prezzi dell'attività sottostante. Per ottenere tale diritto, l'acquirente dell'opzione paga un premio, che consente il trasferimento del rischio.

Troviamo altre due classi d'opzioni: **call e put**. Le prime forniscono il diritto d'acquistare un'attività sottostante ad un prezzo specifico, le seconde forniscono il diritto alla vendita. Si noti che, quando sconveniente, l'opzione non viene esercitata portando alla perdita relativa al solo costo della stessa. Nel caso delle put risulta conveniente esercitare quando il prezzo d'esercizio K è superiore a quello del sottostante S_T ; in tal caso infatti si blocca un prezzo di vendita superiore a quello del mercato.

Viceversa per $S_T \geq K$ non conviene esercitare poiché si andrebbe a vendere ad un prezzo inferiore rispetto a quello che si sarebbe trovato nel mercato.

Per quanto riguarda le call il discorso è inverso in quanto si tratta di operazioni d'acquisto. Per $S_T \geq K$ è conveniente esercitare, per $S_T < K$ l'opzione non viene esercitata. Infatti in tal caso si bloccherebbe un prezzo d'acquisto più alto di quello registrato nel mercato.

Supponendo di fissare un valore per il prezzo d'esercizio, in seguito è fornito un grafico dei payoff dei derivati citati. Si costruisce un piano cartesiano che pone sulle ascisse il prezzo del sottostante e sulle ordinate il profitto o la perdita rilevata. All'interno di tale figura si osserva la cosiddetta posizione lunga, che rappresenta la posizione assunta dall'investitore che acquista il contratto derivato. In questo caso, il possessore del contratto ha il diritto di esercitarlo alle condizioni stabilite.

In contrapposizione, la posizione corta è assunta dal soggetto che vende o sottoscrive il contratto derivato e che, pertanto, si impegna a rispettarne le condizioni qualora la controparte decida di esercitare il diritto previsto dal contratto. Il profitto o la perdita della posizione corta risultano quindi speculari rispetto a quelli della posizione lunga.

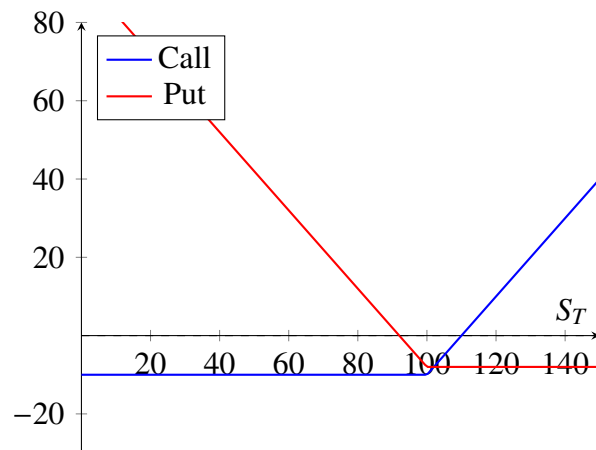


Figura 1: Profili payoff Call e Put europee

Le principali attività sottostanti le opzioni sono quelle tipiche degli altri strumenti derivati come azioni, valute estere, ETF, futures o indici azionari.

Ce ne sono varie tipologie: esotiche, flessibili, settimanali.

Le opzioni sono trattate principalmente nel mercato regolamentato, da cui il ruolo cruciale del broker nello stabilire i prezzi.

Dopo aver definito la struttura e le caratteristiche delle opzioni, emerge naturalmente l'importanza della loro finalità e della loro corretta valutazione. Il pricing delle opzioni rappresenta infatti uno dei temi centrali della moderna teoria finanziaria, poiché richiede l'integrazione di strumenti matematici, probabilistici ed economici per determinarne il valore equo.

In generale, i principali obiettivi associati all'utilizzo degli strumenti derivati riguardano l'adozione di strategie di copertura (hedging), l'individuazione e lo sfruttamento di opportunità di arbitraggio, nonché la possibilità di modificare la natura di una passività o di un investimento senza incorrere nei costi connessi alle operazioni dirette di compravendita. Proprio in relazione a tali finalità risulta fondamentale disporre di metodi affidabili per la valutazione di questi strumenti.

In tale contesto, dal punto di vista teorico lo studio del prezzo delle opzioni ha portato alla formulazione e all'analisi di diversi modelli matematici, sviluppati con l'obiettivo di descrivere in modo rigoroso il funzionamento dei mercati finanziari e l'evoluzione dei prezzi delle attività sottostanti. Attraverso tali modelli è stato possibile approfondire concetti fondamentali della teoria finanziaria, come l'assenza di arbitraggio e l'utilizzo delle probabilità neutrali al rischio.

Accanto alla sua rilevanza teorica, il problema del pricing delle opzioni riveste anche un ruolo di primaria importanza dal punto di vista applicativo. Una valutazione accurata di tali strumenti consente infatti agli operatori di mercato, tra cui investitori, istituzioni finanziarie e gestori del rischio, di assumere decisioni consapevoli nella definizione di strategie di copertura, di investimento e di gestione del rischio. In questo quadro, l'utilizzo di modelli di pricing affidabili risulta quindi fondamentale non solo per la corretta determinazione del valore degli strumenti derivati, ma anche per il miglioramento dell'efficienza complessiva dei mercati finanziari.

Alla luce delle considerazioni precedenti, la presente tesi si propone di analizzare alcuni tra i principali modelli matematici sviluppati per la valutazione delle opzioni,

evidenziandone le ipotesi di base, le proprietà teoriche e le principali applicazioni.

I diversi modelli analizzati nell'elaborato, pur nelle loro differenze strutturali, sono accomunati dall'obiettivo di fornire una rappresentazione coerente dell'evoluzione del sottostante e delle sue implicazioni sul valore del contratto. In questo senso, il confronto tra approcci discreti e continui, così come l'adozione di metodi numerici e simulativi, risponde all'esigenza di affrontare il problema del pricing da prospettive complementari. Ogni modello evidenzia infatti specifici aspetti del fenomeno.

Nel Capitolo 1 viene introdotto il modello dell'albero binomiale (Cox, Ross & Rubinstein, 1979), che rappresenta uno dei più semplici e intuitivi modelli discreti per la valutazione delle opzioni. In tale contesto l'evoluzione del prezzo del sottostante viene descritta attraverso un processo a tempo discreto, che consente di comprendere in modo diretto il pricing in assenza di arbitraggio.

Nel Capitolo 2 vengono introdotti gli strumenti fondamentali del calcolo stocastico, con particolare attenzione al processo di Itô. In questo capitolo si analizza il passaggio dal caso discreto al caso continuo, mostrando che i modelli discreti possono essere interpretati come approssimazioni di processi stocastici continui e fornendo le basi teoriche necessarie per la modellizzazione di processi sempre più sofisticati.

Nel Capitolo 3 viene presentato il modello a tempo continuo di Black-Scholes-Merton per la valutazione delle opzioni, derivato utilizzando gli strumenti del calcolo stocastico introdotti nei capitoli precedenti.

Nel Capitolo 4 viene illustrato il metodo numerico delle differenze finite (Courant, Friedrichs & Lewy, 1928 - Brennan & Schwartz, 1977), utilizzato per risolvere numericamente l'equazione differenziale alle derivate parziali associata al modello di Black-Scholes e per ottenere approssimazioni del prezzo delle opzioni in contesti più generali.

Nel Capitolo 5 viene presentato il metodo di simulazione Monte Carlo, un approccio numerico ampiamente utilizzato in finanza per stimare il valore atteso dei payoff di strumenti finanziari, soprattutto in situazioni in cui non risulta semplice o possibile esplicitare una soluzione analitica.

Nel dettaglio verranno introdotte anche applicazioni a livello informatico sottolineando ulteriormente la trasversalità di tali studi.

L'obiettivo complessivo della tesi è quindi quello di presentare e confrontare diversi approcci al problema della valutazione delle opzioni, sia in tempo discreto sia in tempo continuo, evidenziando il ruolo centrale degli strumenti matematici e numerici nello sviluppo dei moderni modelli di pricing.

1. Modello dell'albero binomiale

Il **modello dell'albero binomiale** schematizza l'evoluzione del sottostante attraverso una struttura discreta di elegante rigore logico (Cox et al. 1979).

In questo scenario, il tempo non è un flusso continuo, ma procede per intervalli definiti, in corrispondenza dei quali il prezzo del titolo può evolvere verso l'alto o verso il basso seguendo coefficienti prefissati.

Questa architettura binaria (Rendleman & Bartter 1979), rappresentata mediante un albero, tesse una rete di percorsi possibili che terminano a scadenza del contratto. La determinazione del prezzo dell'opzione avviene proiettando il valore del sottostante fino ai nodi terminali; ripercorre poi l'intera struttura a ritroso: attraverso l'attualizzazione dei valori attesi del derivato lungo ogni ramificazione, il modello converge verso il valore equo dell'opzione al tempo presente.

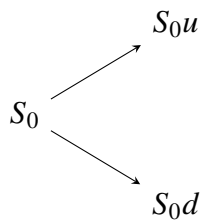
L'interesse per il modello binomiale non risiede tanto nella sua semplicità formale, quanto nella capacità di rendere operativo il concetto di pricing in assenza di arbitraggio all'interno di un contesto a tempo discreto pienamente controllabile.

In primo luogo, la scelta di una struttura discreta consente di isolare in modo trasparente i meccanismi fondamentali del pricing, quali la replicabilità del payoff e la valutazione in ambito neutrale al rischio, evitando inizialmente le complessità tecniche del tempo continuo. Questa caratteristica risulta particolarmente rilevante anche in ambito applicativo, ad esempio nella valutazione di opzioni americane o di strumenti con possibilità di esercizio anticipato, dove la flessibilità della rappresentazione ad albero permette di modellare decisioni dinamiche in modo naturale.

In secondo luogo, la discrezionalità temporale si presta in modo diretto all'implementazione numerica, poiché molte procedure computazionali si basano su schemi di discretizzazione. Il modello binomiale si configura quindi come una base naturale per il calcolo, assumendo anche un ruolo importante come strumento di controllo e validazione: esso consente di confrontare risultati, analizzare la sensibilità ai parametri e valutare l'impatto delle ipotesi modellistiche. Tale trasparenza lo rende particolarmente adatto a contesti operativi in cui è richiesta affidabilità e tracciabilità, come nel risk management e nelle applicazioni regolamentari.

1.1 Definizione formale del modello

Si consideri un titolo di prezzo attuale S_0 e un'opzione scritta sullo stesso di valore f e scadenza T . Si supponga che, da oggi a T , il prezzo possa salire a S_0u (per $u > 1$) o scendere fino a S_0d (con d fattore di sconto < 1).



In T il valore del portafoglio lungo su Δ azioni e corto sul derivato vale:

$$\begin{array}{c} S_0u\Delta - f_u \\ \nearrow \\ S_0 \\ \searrow \\ S_0d\Delta - f_d \end{array}$$

Affinchè il portafoglio possa essere considerato privo di rischio, è necessario che il suo valore risulti identico nei due possibili scenari di evoluzione del prezzo, da cui

$$\Delta = \frac{f_u - f_d}{S_0u - S_0d}.$$

Le variabili f_u e f_d rappresentano i valori del derivato nei nodi finali, rispettivamente nel nodo alto o up e nel nodo relativo alla discesa (down).

Possiamo quindi esprimere Δ come rapporto tra la variazione del prezzo dell'opzione e quella del prezzo del sottostante.

Partendo da tale formula è possibile calcolare il valore del portafoglio al tempo corrente. Basti attualizzare moltiplicando per e^{-rT} ed ottenendo $(S_0u\Delta - f_u)e^{-rT}$.

Per quanto supposto inizialmente, si può anche definire il valore attuale del portafoglio lungo su Δ azioni e corto sul derivato come $S_0\Delta - f$.

Per ricavare f si eguagliano le due espressioni appena citate, si esplicita Δ e si introduce la variabile

$$p = \frac{e^{rT} - d}{u - d}.$$

Si ottiene f pari ad un valore atteso attualizzato:

$$f = e^{-rT} [pf_u + (1 - p)f_d] \quad (1.1)$$

Dimostrazione. (formula 1.1)

Innanzitutto si eguagliano le due formule relative al valore attuale del portafoglio e si

esplicita Δ :

$$S_0\Delta - f = (S_0u\Delta - f_u)e^{-rT}$$

$$f = S_0\Delta - (S_0u\Delta - f_u)e^{-rT}$$

$$\begin{aligned} f &= S_0 \frac{f_u - f_d}{S_0(u - d)} - (S_0u \frac{f_u - f_d}{S_0(u - d)} - f_u)e^{-rT} = \\ &= \frac{f_u - f_d}{u - d} - (u \frac{f_u - f_d}{u - d} - f_u)e^{-rT} = \\ &= \frac{f_u - f_d}{u - d} (1 - ue^{-rT}) + f_ue^{-rT} = \end{aligned}$$

Calcolando il mcm si ottiene

$$\begin{aligned} f &= \frac{f_u - f_d(1 - ue^{-rT}) + (u - d)f_ue^{-rT}}{u - d} = \\ &= \frac{f_u(1 - de^{-rT})}{u - d} + \frac{f_d(ue^{-rT} - 1)}{u - d}. \end{aligned}$$

Raccogliendo e^{-rT} e introducendo la nuova variabile $p = \frac{e^{rt} - d}{u - d}$ si ottiene (1.1).

Il valore attuale dell'opzione si calcola come valore atteso dei nodi finali attualizzato per il passo temporale.

Va ora dimostrato che $0 \leq p \leq 1$ essendo essa una probabilità. Più nel dettaglio p è la probabilità che, in un mondo neutrale al rischio, il prezzo S salga; $(1 - p) = \frac{u - e^{rT}}{u - d}$ è la probabilità che, in un mondo neutrale al rischio, il prezzo S scenda.

$$0 \leq p \leq 1 \iff d \leq e^{rT} \leq u$$

Applicando il logaritmo naturale ad entrambi i membri nella prima disuguaglianza si ha

$$\ln d \leq rT.$$

Questo è sempre vero, infatti il logaritmo di una quantità compresa tra 0 e 1 (quale il

fattore di sconto d) è negativo; d'altra parte sia r che T sono sempre positivi, da cui la tesi.

Per quanto riguarda la disuguaglianza di destra, si applica anche qui il logaritmo naturale ambo le parti ottenendo

$$rT \leq \ln(u).$$

Si moltiplica e divide u per S , ricavando un rendimento logaritmico. Tale disuguaglianza sottolinea quindi che

$$\text{rend titolo privo di rischio} \leq \text{rend up di un titolo rischioso}.$$

Necessariamente anche questo è sempre vero, altrimenti si genererebbe la possibilità di arbitraggi con investimenti nel solo titolo non rischioso. $\square$

Nel calcolo di u e d , solitamente si suppone che l'albero binomiale oscilli seguendo le oscillazioni del sottostante. Questo è possibile grazie al **Teorema di Girsanov** che enuncia quanto segue: nel passaggio dal mercato reale al modello neutrale al rischio (tipico dell'albero), la componente di volatilità rimane invariata, mentre a variare è esclusivamente il tasso di crescita.

Se si suppone di calcolare u e d considerando costante la volatilità del modello si ottengono:

$$u = e^{\sigma\sqrt{\Delta t}}$$

$$d = e^{-\sigma\sqrt{\Delta t}}$$

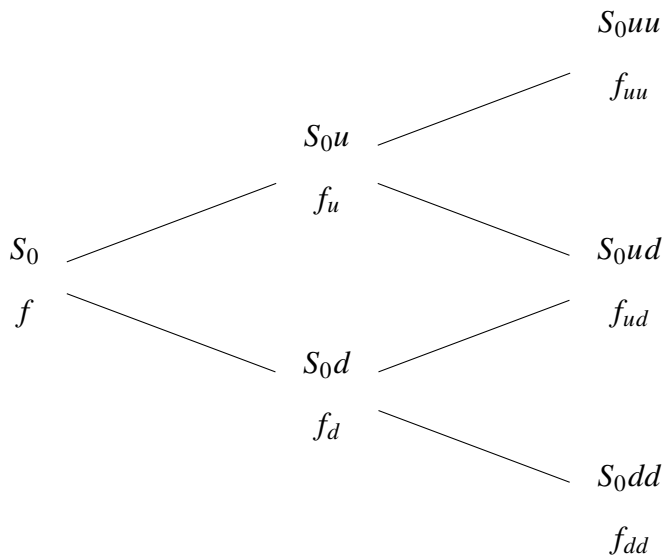
1.2 Tecniche risolutive

Per quanto riguarda un albero con un singolo passo temporale, come quello appena studiato, la tecnica risolutiva è la seguente. Innanzitutto attraverso il calcolo di u e d si ricavano

i prezzi in salita e discesa del sottostante e la probabilità di ottenerli. Successivamente si calcola il payoff del derivato in tali nodi finali.

Supponiamo di analizzare una call a partire dal nodo up (ripetendo analogamente per la discesa), se $S_{0u} < K$ l'opzione non è esercitata e quindi $f_u = 0$; se $S_{0u} \geq K$ il payoff ottenuto dalla call è $S_{0u} - K$. A questo punto si calcola il prezzo dell'opzione come valore atteso attraverso (1.1).

Nel caso della put il discorso si inverte, considerando un profitto nullo per $S_{0u} > K$ e un payoff positivo nel caso opposto pari a $K - S_{0u}$. Per un albero a due stadi invece questo percorso va reiterato.



In questo caso si calcolano i payoff nei nodi finali, a seconda di avere una call o una put. Si calcolano poi

$$f_u = (f_{uu}p + (1-p)f_{ud})e^{-r\Delta t}$$

e

$$f_d = (f_{du}p + (1-p)f_{dd})e^{-r\Delta t},$$

dove Δt è un singolo passo temporale. Poi per valutare f , si sfrutta (1.1) che ricordiamo

essere:

$$f = e^{-r\Delta t} [p f_u + (1 - p) f_d]$$

Se non si vogliono considerare i nodi intermedi, si può determinare f per un albero a due stadi come segue:

$$e^{-2r\Delta t} (p^2 f_{uu} + 2p(1 - p) f_{ud} + (1 - p)^2 f_{dd}).$$

Si osservi, ad esempio, il caso di f_{dd} : per raggiungere tale stato è necessario effettuare due discese consecutive, ciascuna con probabilità pari a $(1 - p)$. Tale procedura risulta applicabile solo nel caso delle opzioni europee, esercitabili unicamente a scadenza.

Si può generalizzare per n nodi ottenendo:

$$c = e^{-rT} \sum_{j=0}^n \binom{n}{j} p^j (1 - p)^{n-j} \max(S_0 u^j d^{n-j} - K, 0) \quad (1.2)$$

$$p = e^{-rT} \sum_{j=0}^n \binom{n}{j} p^j (1 - p)^{n-j} \max(K - S_0 u^j d^{n-j}, 0) \quad (1.3)$$

Per quanto riguarda le opzioni americane, è necessario analizzare i nodi intermedi a causa del possibile esercizio anticipato. In particolare se questo risulta superiore al valore di f_i (calcolato come valore atteso attualizzato), gli va sostituito.

Va sottolineato inoltre che dalla definizione introdotta per Δ , cioè

$$\Delta = \frac{f_u - f_d}{S_0 u - S_0 d},$$

si può considerare tale grandezza come un rapporto di copertura. Ci consente di effettuare due tipologie di analisi: se ci si basa su un portafoglio privo di rischio, si calcola Δ , il valore dell'opzione nei nodi finali e si attualizza. D'altro canto se si considera il mondo neutrale al rischio, tipico del modello dell'albero binomiale, si sfruttano le probabilità

sopra citate.

Questo modello rappresenta un modello a tempo discreto. È possibile però dimostrare la convergenza ad uno a tempo continuo.

Si dimostra infatti come per $\Delta t \rightarrow 0$ o analogamente per $n \rightarrow \infty$ il modello dell'albero binomiale converga alla formula analitica di Black-Scholes-Merton.

Dimostrazione. Si effettua l'analisi a partire dalla definizione (1.2) di una call nel modello numerico. Si ricorda che

$$c = e^{-rT} \sum_{j=0}^n \binom{n}{j} p^j (1-p)^{n-j} \max(S_0 u^j d^{n-j} - K, 0)$$

Il primo passo è l'eliminazione della funzione di massimo; per farlo è necessario trovare l'indice j a partire dal quale si hanno payoff positivi. Ciò avviene quando

$$S_0 u^j d^{n-j} > K.$$

Si applica il logaritmo ambo i membri, poiché j si trova all'esponente ottenendo

$$j > \alpha, \text{ per } \alpha = \frac{n - \ln \frac{S_0}{K}}{2 \sigma \sqrt{\frac{T}{n}}};$$

$$c = e^{-rT} \sum_{j>\alpha} \binom{n}{j} p^j (1-p)^{n-j} (S_0 u^j d^{n-j} - K).$$

Dopo aver moltiplicato sia il termine in S_0 che K per la probabilità di raggiungere un determinato nodo, si introducono le due seguenti grandezze

$$U_1 = \sum_{j>\alpha} \binom{n}{j} p^j (1-p)^{n-j} u^j d^{n-j},$$

$$U_2 = \sum_{j>\alpha} \binom{n}{j} p^j (1-p)^{n-j}.$$

U_2 rappresenta la probabilità di ottenere payoff positivi e segue una distribuzione binomiale. Per far sì che anche U_1 segua la medesima distribuzione è necessario eliminare i termini $u^j d^{n-j}$. A tale scopo si definisce

$$p^* = \frac{pu}{pu + (1-p)d},$$

da cui

$$U_1 = e^{rT} \sum_{j > \alpha} \binom{n}{j} (p^*)^j (1-p^*)^{n-j}.$$

La funzione esponenziale compare a causa della modifica algebrica.

Per $n \rightarrow \infty$ una distribuzione binomiale converge ad una normale consentendo l'utilizzo della formula BSM (introdotta in seguito):

$$\text{per } n \rightarrow \infty \quad c = e^{-rT} [S_0 U_1 - K U_2]$$

Si dimostra in modo analogo per le put. $\square$

2. Processi di Itô

2.1 Dalla discretizzazione al tempo continuo

Nonostante l'efficacia del modello binomiale, la realtà dei mercati finanziari è caratterizzata da scambi quasi istantanei, rendendo la discretizzazione temporale un limite alla precisione del pricing.

Per superare tali limiti ed operare in un contesto di tempo continuo, l'obiettivo principale diventa la formalizzazione dei processi di Itô. Questi rappresentano lo strumento

risolutivo, poiché permettono di descrivere analiticamente la dinamica dei prezzi in ogni istante infinitesimale.

Tuttavia, per comprendere appieno la natura di tali processi, è necessario inquadrarli all'interno della teoria più ampia da cui originano: quella dei processi stocastici.

Un **processo stocastico** è un modello matematico che descrive l'evoluzione nel tempo di una variabile il cui valore oscilla in modo incerto.

All'interno di questa categoria, come sottolineato, si collocano i **processi di Itô** (Itô, 1944), che rappresentano una particolare classe di processi stocastici con proprietà markoviane.

Lo sviluppo teorico dei processi stocastici si inserisce nel contesto della teoria moderna della probabilità (Kolmogorov, 1933) e della formalizzazione matematica del moto browniano (Wiener, 1923), a sua volta basata su precedenti studi del moto browniano (Einstein, 1905).

Si parte analizzando i cosiddetti **processi Markoviani**.

Questi ultimi rappresentano un tipo particolare di processo stocastico in cui l'evoluzione futura della variabile dipende solo dal suo valore attuale e non dal sentiero temporale o i modi per arrivare a quel determinato valore. Si assume che i prezzi delle azioni seguano un processo Markoviano, cioè la loro distribuzione probabilistica, in un qualsiasi futuro istante di tempo, non dipende dal sentiero temporale seguito in passato dal prezzo delle azioni. L'ipotesi di processo Markoviano risulta coerente con la forma debole d'efficienza dei mercati. Questa teoria sostiene che il prezzo dell'azione inglobi già in sé le informazioni circa la serie storica dei prezzi.

Di conseguenza, l'analisi tecnica, che si basa proprio sullo studio dei dati passati di mercato, risulta inutile ai fini della previsione delle future variazioni dei prezzi.

Formalmente una variabile si dice seguire un processo Markoviano se la distribuzione probabilistica della sua variazione per un periodo T segue $\varphi(0, T)$; se si considera un più breve intervallo Δt , la distribuzione della variazione segue $\varphi(0, \Delta t)$, con $\varphi(m, v)$ la

funzione di densità di una normale con media m e varianza v .

Si definisce **processo di Wiener** un particolare processo Markoviano con variazione media nulla e varianza unitaria.

Questo modello viene usato anche per descrivere il moto delle particelle durante shock molecolari, per questo viene anche detto moto browniano.

Formalmente, una variabile z segue un processo di Wiener se

$$-\Delta z = \epsilon \sqrt{\Delta t},$$

-i valori di Δz relativi a due diversi periodi di tempo sono indipendenti.

Si suppone che ϵ sia un'estrazione da $\varphi(0, 1)$, con φ una normale. Quindi Δz amplifica l'estrazione da una normale standardizzata di radice di Δt .

Sfruttando la seconda proprietà del processo, si consideri la variazione di z su un intervallo di periodo T :

$$z(T) - z(0) = \sum_{i=1}^N \epsilon_i \sqrt{\Delta t};$$

praticamente si divide T in N sottointervalli di ampiezza Δt e in ognuno di questi si valuta Δz . Applicando la proprietà si ottiene che la media di $z(T) - z(0)$ è la somma delle medie sui singoli intervalli, per cui è nulla.

Per la varianza si possono sommare le singole varianze solo grazie all'indipendenza delle variabili ottenendo $N\Delta t$, cioè T .

Il limite di questo modello è l'utilizzo di drift rate (media delle variazioni su unità di tempo) nullo e variance rate (varianza delle variazioni su unità di tempo) unitario. Per questo si introducono i **processi di Wiener generalizzati**, che hanno drift rate e variance rate pari a costanti arbitrarie non nulle.

Formalmente si dice che una variabile x segue un processo di Wiener generalizzato con drift rate a e variance rate b^2 se esprimibile nelle seguenti forme a tempo continuo o discreto rispettivamente

$$dx = a dt + b dz;$$

$$\Delta x = a\Delta t + b\Delta z,$$

con z che segue un processo di Wiener. La prima parte dell'espressione è detta deterministica mentre la seconda stocastica. Si noti che tali modelli non sono particolarmente idonei per valutare le oscillazioni dei prezzi azionari; infatti questi ultimi si basano su variazioni percentuali mentre i processi appena introdotti su variazioni assolute. Tra i processi Markoviani quindi, i processi di Wiener non sono i più adatti a valutare i prezzi azionari.

Per fare questo si usa il cosiddetto **processo di Itô**.

2.2 Definizione formale del modello

Una variabile x si dice seguire un processo di Itô se ha drift rate e variance rate che dipendono dal tempo:

$$dx = a(x, t)dt + b(x, t)dz;$$

$$\Delta x = a(x, t)\Delta t + b(x, t)\Delta z = a(x, t)\Delta t + b(x, t)\epsilon\sqrt{\Delta t},$$

con z che segue Wiener.

Nel caso dei prezzi azionari S , il processo viene definito nel modo seguente

$$dS = \mu S dt + \sigma S dz, \tag{2.1}$$

$$\Delta S = \mu S \Delta t + \sigma S \Delta z, \tag{2.2}$$

anche qui con z che segue un moto browniano. A partire da questo modello per la variazione dei prezzi azionari, va costruito un modello che descriva il processo seguito dal derivato. Per fare questo si utilizza il **lemma di Itô**.

A partire dal processo del sottostante x , questo modello ci fornisce il percorso del

derivato $G(x,t)$ scritto sullo stesso. Si ottiene

$$dG(x,t) = \left(\frac{\partial G}{\partial x} a + \frac{\partial G}{\partial t} + \frac{1}{2} \frac{\partial^2 G}{\partial x^2} b^2 \right) dt + \frac{\partial G}{\partial x} b dz \quad (2.3)$$

Dimostrazione. (2.3 - Lemma di Itô)

Innanzitutto si sviluppa G secondo la regola di Taylor, arrestandosi al secondo ordine

$$\begin{aligned} \Delta G \approx & G_x \Delta x + G_t \Delta t + \frac{1}{2} G_{xx} (\Delta x)^2 \\ & + G_{xt} (\Delta x)(\Delta t) + \frac{1}{2} G_{tt} (\Delta t)^2 + o\left((\Delta x)^2 + (\Delta t)^2\right). \end{aligned}$$

Si ipotizzano trascurabili tutti i termini di ordine superiore a Δt , quindi si elimina la derivata seconda temporale e anche quella mista. Infatti in essa compaiono sia il termine Δt che Δx , avente una componente dell'ordine di $\sqrt{\Delta t}$.

ΔG è quindi approssimabile come $\Delta G \approx G_x \Delta x + G_t \Delta t + \frac{1}{2} G_{xx} (\Delta x)^2$.

Si sostituisce a Δx il percorso $a(x,t)\Delta t + b(x,t)\epsilon\sqrt{\Delta t}$, sempre trascurando i termini di ordine superiore a Δt si ottiene

$$\Delta G = G_x \Delta x + G_t \Delta t + \frac{1}{2} G_{xx} b^2 \epsilon^2 \Delta t.$$

Dalle formule sulla varianza è noto che essa risulta esprimibile come differenza tra il momento secondo e il momento primo al quadrato. Quest'ultimo è nullo poiché ϵ rappresenta una estrazione casuale da una normale standardizzata.

Ci si riconduce quindi ad eguagliare momento secondo e varianza di ϵ

$$E(\epsilon^2) = 1,$$

da cui

$$E(\epsilon^2 \Delta t) = \Delta t.$$

Si ha quindi che la varianza di $\epsilon^2 \Delta t$ è proporzionale a Δt^2 e quindi seguendo le Hp va ignorata.

Prendendo i limiti e sostituendo a dx la sua espressione sfruttando Itô si ottiene la tesi del lemma (2.3):

$$dG = (G_x a + G_t + \frac{1}{2} G_{xx} b^2) dt + G_x b dz \square$$

Supponendo quindi di partire da un'azione S modellizzata col processo di Itô (2.1) è possibile ottenere il percorso del derivato attraverso il lemma di Itô. Sia f l'opzione scritta su S si ricava

$$df = (\frac{\partial f}{\partial S} \mu S + \frac{\partial f}{\partial t} + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2) dt + \sigma S dz$$

3. Modello di Black-Scholes-Merton

Il **modello di Black-Scholes-Merton** rappresenta il pilastro della finanza moderna, nato per risolvere definitivamente il problema del pricing delle opzioni europee in un contesto di mercato continuo.

Esso affronta il limite principale dei modelli numerici e discreti: la mancanza di una formula chiusa che permetta un calcolo istantaneo e preciso del valore equo. Attraverso la formulazione di un'equazione differenziale alle derivate parziali (PDE), il modello trasforma il pricing da un'approssimazione iterativa ad una soluzione matematica certa, basata sul principio del delta hedging (ovvero la costituzione di un portafoglio di copertura che rende la posizione priva di rischio) e sull'assenza di arbitraggio.

Questo pilastro della finanza moderna è stato definito nel 1973 attraverso due contributi fondamentali: lo studio sulla valutazione delle opzioni e delle passività aziendali (Black & Scholes 1973) e l'analisi del pricing razionale dei titoli derivati (Merton 1973).

In questo modello si parte considerando che i prezzi azionari seguano una random walk. Sia ΔS la variazione del prezzo su un intervallo Δt , allora si può definire il rendimento del titolo come $\frac{\Delta S}{S}$ supponendo che si distribuisca secondo una normale con media $\mu\Delta t$ e deviazione standard $\sigma\sqrt{\Delta t}$.

Queste ipotesi implicano che la differenza $\ln(S_T) - \ln(S_0)$ si distribuisca secondo una normale con media $(\mu - \frac{\sigma^2}{2})T$ e varianza $\sigma^2 T$. Ciò deriva dall'applicazione del lemma di Itô a $G = \ln(S)$. In tal caso infatti si ottiene che il derivato G ha un processo del tipo

$$\Delta G = (\mu - \frac{\sigma^2}{2})\Delta t + \sigma\Delta z,$$

La media e la varianza del derivato risultano pari a $(\mu - \frac{\sigma^2}{2})T$ e $\sigma^2 T$ rispettivamente.

ΔG è anche esprimibile come differenza tra G_{t+1} e G_t , cioè $\ln(S_{t+1}) - \ln(S_t)$, la cui media e varianza è quindi riconducibile a quanto enunciato.

Si noti che il logaritmo del valore atteso è diverso dal valore atteso del logaritmo, infatti

$$E(S_T) = S_0 e^{\mu T}$$

$$E(\frac{S_T}{S_0}) = e^{\mu T}$$

$$\ln[E(\frac{S_T}{S_0})] = \mu T,$$

questo valore risulta maggiore del valore atteso del logaritmo che, per quanto appena dimostrato, è $(\mu - \frac{\sigma^2}{2})T$.

3.1 Definizione formale del modello

Per trovare le equazioni che definiscano il pricing delle opzioni si costruisce un modello semplificato basato sulle seguenti ipotesi:

1. Le azioni seguono un processo di Itô con μ e σ costanti.
2. Non ci sono costi di transazione.
3. Non esistono opportunità di arbitraggio.
4. Non sono permesse vendite allo scoperto.
5. Non vengono distribuiti dividendi.
6. Le azioni sono perfettamente divisibili.
7. Il mercato è supposto continuo.
8. Il tasso risk-free è lo stesso per tutte le scadenze.

L'adozione del modello Black-Scholes-Merton risponde all'esigenza di disporre di un benchmark teorico forte rispetto al quale valutare ogni altra metodologia di pricing. La sua importanza applicativa non deriva tanto dal realismo delle ipotesi appena introdotte, quanto dal fatto che fornisce una struttura coerente e chiusa per interpretare i prezzi di mercato.

Quindi in ambito operativo, il modello è spesso usato non per prezzare direttamente, ma per calibrare e interpretare i dati di mercato. Le sue limitazioni, come l'ipotesi di volatilità costante o l'adozione di distribuzioni lognormali, non lo rendono obsoleto, ma piuttosto un punto di riferimento rispetto al quale misurare deviazioni e costruire modelli ancora più sofisticati.

In questo contesto, risulta naturale proseguire con l'introduzione delle formule esplicite di pricing che discendono da tale impostazione teorica.

Queste ultime sono riconducibili a quelle ottenute attraverso la convergenza del modello binomiale. Per una call e una put si ha rispettivamente

$$c = S_0 N(d_1) - K e^{-rT} N(d_2) \quad (3.1)$$

$$p = K e^{-rT} N(-d_2) - S_0 N(-d_1) \quad (3.2)$$

$N(d_i)$ rappresentano le funzioni di ripartizione di una normale standardizzata, mentre

$$d_1 = \frac{\ln(\frac{S_0}{K}) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}},$$

$$d_2 = d_1 - \sigma\sqrt{T}.$$

Si noti che quando S_0 aumenta molto

$$c \rightarrow S_0 - K e^{-rT}$$

$$p \rightarrow 0$$

Infatti per $S_0 \rightarrow \infty$ d_1 , ma anche d_2 , compaiono con tale termine a numeratore. Il logaritmo di una quantità che tende ad infinito è esso stesso infinito.

Per $d_i \rightarrow \infty$ le funzioni di ripartizione, che calcolano l'area al di sotto della curva normale, tendono alla massima superficie considerabile cioè 1. La call tende allora al suo payoff e la put a 0. Quest'ultima infatti manifesta segni negativi avanti alle d_i .

Per $S_0 \rightarrow 0$ si ha invece che

$$c \rightarrow 0$$

e

$$p \rightarrow K e^{-rT} - S_0.$$

Infatti in questo caso il logaritmo che compare al numeratore delle d_i tende a meno infi-

nito. Questo implica che la funzione di ripartizione tende alla minima area considerabile, cioè quella nulla. Per la put anche qui il discorso è inverso, quindi essa tende al suo payoff.

3.2 Tecniche risolutive

Andremo ora a dimostrare l'ottenimento delle formule (3.1), (3.2).

Dimostrazione.

Dalla prima ipotesi esprimiamo il processo seguito dall'azione S tramite (2.2):

$$\Delta S = \mu S \Delta t + \sigma S \Delta z = \mu S \Delta t + \sigma S \sqrt{\Delta t} \epsilon$$

da cui il derivato f segue

$$\Delta f = \left(\frac{\partial f}{\partial S} \mu S + \frac{\partial f}{\partial t} + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \right) \Delta t + \frac{\partial f}{\partial S} \sigma S \Delta z,$$

dove S e f hanno stessa fonte d'incertezza associata a Δz .

Si costruisce il portafoglio corto sul derivato e lungo su $\frac{\partial f}{\partial S}$ azioni. Tale valore è analogo al Δ ottenuto nel modello binomiale. Quest'ultimo rappresenta il rapporto tra la variazione del prezzo dell'opzione e quella del sottostante. Passando al caso continuo si ottiene proprio la derivata parziale di f rispetto a S .

Il valore del portafoglio è quindi

$$\Pi = -f + \frac{\partial f}{\partial S} S,$$

la variazione del portafoglio è

$$\Delta \Pi = -\Delta f + \frac{\partial f}{\partial S} \Delta S.$$

Sostituendo a ΔS e Δf quanto ottenuto da Itô si ha

$$\Delta \Pi = \left(-\frac{\partial f}{\partial t} - \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \right) \Delta t.$$

Quindi il rendimento (associato ai termini in μ) e l'incertezza dell'azione (associata a $\sigma \Delta z$) non influenzano la determinazione del valore del portafoglio e di conseguenza nemmeno la determinazione del valore del derivato.

In assenza di opportunità d'arbitraggio il rendimento del portafoglio dovrà coincidere con quello di un titolo risk-free. Da cui:

$$\frac{\Delta \Pi}{\Pi} = r \Delta t.$$

Sostituendo a Π e $\Delta \Pi$ quanto sopra dimostrato, si ottiene l'equazione differenziale di Black-Scholes-Merton:

$$\frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \quad (3.3)$$

Il derivato da valutare è identificato dalle condizioni al contorno. Se come derivato assumo un'opzione si considerano le seguenti condizioni in T :

$$c = \max(S_T - K, 0),$$

$$p = \max(K - S_T, 0).$$

Risolvendo l'equazione differenziale con queste condizioni al contorno (vedi Appendice) si arriva alle medesime soluzioni ottenute per convergenza del modello binomiale. $\square$

I prezzi delle azioni possono essere calcolati idealmente con tale equazione differenziale, ma sono realmente generati dalle pressioni di mercato che si trovano all'interno della Borsa; posso allora ricavare la volatilità implicitamente dai prezzi delle opzioni sul mercato. La volatilità implicita di un'opzione europea è la volatilità che, sostituita nella

formula di Black-Scholes-Merton, fornisce il valore di mercato dell'opzione. Spesso i brokers quotano le volatilità piuttosto dei prezzi delle opzioni.

È inoltre interessante sottolineare come, per quanto riguarda i dividendi, le opzioni europee possano essere comunque valutate con tale modello, ma occorre detrarre dal prezzo dell'azione il valore attuale dei dividendi.

3.3 Generalizzazione del modello: parametri variabili

Il modello di Black-Scholes-Merton (BSM) è universalmente considerato uno dei pilastri fondamentali della finanza quantitativa moderna. La sua introduzione, che si ricorda essere nel 1973, ha fornito un modello analitico rigoroso per il pricing delle opzioni europee, basato su otto ipotesi semplificatrici che frequentemente non trovano riscontro nelle dinamiche reali dei mercati.

In particolare, il modello tradizionale di BSM poggia su una base concettuale che definisce il mercato come uno spazio finanziario euclideo e statico. In ambito topologico, uno spazio euclideo è caratterizzato da una curvatura nulla; trasposto in finanza, questo implica che la relazione tra il prezzo del sottostante e il rischio sia lineare, uniforme e priva di distorsioni locali.

Per quanto riguarda la definizione "statica" dello spazio, questa non si riferisce all'assenza di movimento dei prezzi, bensì alle ipotesi stringenti sopracitate: nel modello del '73, infatti, la volatilità (σ) e il tasso d'interesse (r) sono assunti come costanti per tutta la durata del contratto. L'osservazione empirica rivela invece che tale spazio è deformato da asimmetrie informative e shock di liquidità, rendendo necessaria una generalizzazione a parametri variabili.

Per rappresentare al meglio la realtà finanziaria, la ricerca ha colto delle analogie con gli studi di meccanica quantistica, individuando nell'equazione di Schrödinger (El-

Nabulsi & Anukool 2026) lo strumento idoneo per una modellizzazione non più statica. Si va a creare un profondo parallelismo tra tale equazione e il pricing delle opzioni.

Il passaggio dall'equazione BSM alla sua forma generalizzata di Schrödinger avviene quindi partendo dalla fisica quantistica e reinterpretando i risultati in ambito finanziario.

Matematicamente, si definisce l'equazione di Black-Scholes-Merton come quell'equazione che descrive l'evoluzione del prezzo di un derivato attraverso la PDE (3.3) che ricordiamo:

$$\frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} - rf = 0,$$

dove $f(S, t)$ è il prezzo del derivato, S il prezzo del sottostante, σ la volatilità e r il tasso d'interesse privo di rischio.

Come dimostrato in Appendice, tale PDE può essere ricondotta all'equazione del calore, classico esempio di processo diffusivo a coefficienti costanti (Smith 1985, Zachmanoglou & Thoe 1986). Tale trasformazione è ampiamente utilizzata anche in ambito matematico-finanziario e quantistico (Wilmott 1998, Tychonov & Samarski 1964, Clewlow & Strickland 1998, Haberman 2013).

L'equazione fondamentale in tale ambito è la sopracitata equazione di Schrödinger, che in una dimensione si scrive:

$$i \frac{\partial \psi}{\partial t} = \hat{H} \psi, \quad \hat{H} = -\frac{\partial^2}{\partial x^2} \frac{\hbar^2}{2m} + U(x) \quad (3.4)$$

dove $\psi(x, t)$ è la funzione d'onda (la probabilità di trovare una particella in una determinata posizione ad un certo istante) e H è l'operatore hamiltoniano, somma di un termine cinetico e di un potenziale $U(x)$. Tale equazione descrive un'evoluzione di tipo oscillatorio e conservativo, cioè indica che la probabilità totale si redistribuisce nello spazio senza dissiparsi nel tempo.

Sottolineando il parallelismo con l'ambito finanziario si può considerare l'hamiltoniano come una grandezza che incapsula l'interazione tra la forza propagativa della volatilità

(assimilabile all'energia cinetica) e l'effetto di pass-through dei tassi d'interesse (assimilabile all'energia potenziale). Quest'ultima grandezza sottolinea come le variazioni dei tassi ufficiali, effettuate dalle BC, si ripercuotano nelle altre banche in termini di prestiti, mutui ecc.

Il legame tra la formula quantistica e l'equazione del calore emerge applicando una trasformazione del tempo detta rotazione di Wick, che consiste nel sostituire il tempo reale con un tempo immaginario secondo $t \rightarrow -i\tau$. Questa trasformazione converte l'evoluzione oscillatoria dell'equazione di Schrödinger in una diffusiva più vicina all'equazione del calore.

L'equazione di Schrödinger assume allora la forma

$$\frac{\partial \psi}{\partial \tau} = -\hat{H}\psi \quad (3.5)$$

che, in presenza di un potenziale assegnato, coincide con una equazione di diffusione con termine di potenziale (incorporato nell'hamiltoniano).

Il parallelismo tra il mondo fisico e finanziario diventa ancora più marcato se si introduce una trasformazione di gauge, ossia un riscaldamento esponenziale della funzione del tipo

$$\psi(x, \tau) = e^{r\tau} u(x, \tau),$$

che consente di esplicitare la dipendenza dal tasso d'interesse in questa nuova versione generalizzata, non più a parametri costanti.

A seguito di tale trasformazione, l'equazione assume la forma

$$\frac{\partial \psi}{\partial \tau} = \frac{1}{2}\sigma^2 \frac{\partial^2 \psi}{\partial x^2} - U(x)\psi \quad (3.6)$$

dove $U(x)$ rappresenta un potenziale efficace associato al tasso di interesse. In questa formulazione finale, l'equazione di Black-Scholes-Merton appare formalmente equivalente

ad una equazione di Schrödinger in tempo immaginario con potenziale, rendendo esplicita la profonda analogia tra dinamica dei prezzi finanziari assumendo neutralità al rischio ed evoluzione di sistemi quantistici descritti dalla fisica matematica.

Per comprendere al meglio tale paragone vanno analizzate più nel dettaglio le grandezze introdotte e il loro duplice significato, in termini finanziari e fisici.

Partendo dal tasso d'interesse r si può sottolineare come, attraverso la trasformazione di gauge, questo da semplice coefficiente è ricondotto ad un potenziale $U(\tau)$. In fisica un potenziale è una grandezza associata ad un campo di forze, il quale è in grado di modificare ed influenzare i corpi situati nello spazio circostante.

In relazione all'analogia finanziaria introdotta, si può considerare il prezzo come un corpo, il mercato come lo spazio in cui tale corpo è collocato. Infine il tasso d'interesse rappresenta una delle forze che influenza il decorso probabilistico del prezzo.

Se si considerasse r costante come nell'approccio tradizionale, il potenziale (che dipende da tale grandezza) rimane costante e si definisce il campo come uniforme. In questa nuova constatazione si assume invece che r e σ variano al trascorrere del tempo. Ciò significa che il potenziale U stesso dipende dal tempo.

Nel dettaglio dalla trasformazione di gauge si può dimostrare che il potenziale risulta definito come somma di due componenti:

$$U(x, \tau) = r(\tau) + f(\sigma(x, \tau), \partial_x \sigma, \partial_{xx} \sigma). \quad (3.7)$$

Il primo contributo è di tipo globale, mentre il secondo deriva dalla struttura della volatilità e dalle sue variazioni spazio-temporali.

$r(\tau)$ non risulta dipendente dalla variabile x rappresentante il log-prezzo. Ad una variazione del tasso conseguirà quindi una variazione costante del potenziale:

$$U(x, t) \rightarrow U(x, t) + \Delta r(\tau) \forall x \quad (3.8)$$

Ciò che risulta modificato è quindi il livello complessivo del potenziale, generando una traslazione verticale uniforme. Non si perde quindi l'omogeneità spaziale, in quanto questa è legata al termine in x . Per quanto riguarda l'ambito finanziario, l'omogeneità spaziale va interpretata come la non creazione di zone costituite da prezzi privilegiati o favorevoli.

Il secondo contributo dipende da σ , che a sua volta dipende da x . Ciò significa che la volatilità può variare non solo al trascorrere del tempo, ma anche in funzione del livello del sottostante x .

Di conseguenza a variazioni di σ e delle sue derivate, si genera una modifica non uniforme del potenziale. Si creano cioè aree a potenziale più elevato ed aree in cui risulta più basso.

Riprendendo l'equazione di Schrödinger (3.6) si possono analizzare tali casistiche.

Si parte considerando un punto x_0 a potenziale elevato: se $U(x_0, t)$ è un valore grande, il termine $-U(x_0, t)\psi(x_0, t)$ diventa fortemente negativo.

Quindi il primo membro di (3.6), nonché derivata della densità rispetto al tempo, risulta particolarmente negativo rispetto ad altre zone. Ne segue che la densità $\psi(x, t)$ nella regione considerata decresce velocemente, cioè si genera una minore concentrazione di probabilità.

Nel caso contrario, se si assumesse $U(x_0, t)$ un valore basso, il sottraendo dell'equazione risulta ridotto. Rispetto ad altre regioni quindi, la funzione di densità è relativamente più alta: si genera un maggiore accumulo relativo di probabilità.

In conclusione si può dire che quando σ dipende dalla variabile spaziale x , le sue derivate rispetto a quest'ultima generano eterogeneità nel potenziale $U(x, t)$; ne consegue la creazione di regioni lungo lo spazio dei prezzi in cui l'andamento probabilistico risulta favorevole o più svantaggioso.

4. Metodo delle differenze finite

Il **metodo delle differenze finite** rappresenta una sorta di congiunzione tra l'eleganza teorica delle equazioni differenziali e la necessità pratica di ottenere soluzioni numeriche in assenza di formule chiuse (Hull & White 1990, Figlewski & Gao 1999).

Sebbene il modello di Black-Scholes-Merton fornisca un benchmark teorico fondamentale, la sua applicabilità è limitata da ipotesi spesso troppo restrittive o da payoff complessi che non ammettono soluzioni analitiche, come analizzato nel precedente capitolo.

Il metodo delle differenze finite nasce quindi per colmare questo divario in scenari dove i parametri non sono costanti o dove sono presenti vincoli di esercizio anticipato, superando l'impossibilità di ottenere formule chiuse per strumenti finanziari più sofisticati.

Sebbene le sue radici teoriche risalgano alla matematica classica, la sua formalizzazione e lo studio della sua stabilità computazionale si devono a contributi fondamentali nel campo dell'analisi numerica moderna (Courant et al. 1928), trovando poi una specifica applicazione finanziaria nella risoluzione dell'equazione di Black-Scholes (Brennan & Schwartz 1977).

Invece di cercare una soluzione analitica globale, questa tecnica scompone il dominio continuo del prezzo e del tempo in una griglia rigida di punti discreti. L'equazione differenziale che governa il valore del derivato viene così convertita in un sistema di equazioni alle differenze, risolvibili iterativamente attraverso il calcolo computazionale.

Dal punto di vista operativo, le differenze finite trovano largo impiego in contesti in cui è richiesta elevata affidabilità della soluzione numerica, come nei sistemi di pricing

interni alle istituzioni finanziarie.

In primo luogo, la soluzione ottenuta risulta pienamente deterministica: a parità di parametri e discretizzazione, essa viene riprodotta in modo identico ad ogni esecuzione. Questa proprietà rende la soluzione particolarmente adatta a processi di validazione e confronto nel tempo, in quanto elimina la componente di variabilità tipica dei metodi simulativi. Di conseguenza, la soluzione numerica può essere analizzata, verificata e documentata in modo rigoroso, soddisfacendo i requisiti di trasparenza e tracciabilità richiesti in ambito regolamentare.

In secondo luogo, il metodo consente di intervenire direttamente sulla struttura della soluzione numerica. La definizione della griglia e la scelta dello schema alle differenze determinano infatti la forma, la regolarità e la precisione della soluzione stessa. Ciò permette di modellare in modo mirato specifiche caratteristiche del problema, ad esempio introducendo vincoli locali o raffinando la discretizzazione in regioni di maggiore interesse. In presenza di payoff complessi o di condizioni come l'esercizio anticipato, la soluzione può essere costruita e adattata punto per punto, mantenendo il controllo sul suo comportamento.

In questo senso, la soluzione numerica non è soltanto un output del modello, ma un oggetto su cui è possibile intervenire e che può essere progettato in funzione delle esigenze applicative.

4.1 Definizione formale del modello

Il metodo delle differenze finite può essere svolto implicitamente o esplicitamente.

Il primo permette di costruire una relazione che lega tre valori dell'opzione al tempo $i\Delta t$ col valore al tempo $(i+1)\Delta t$. Il metodo esplicito invece, porta alla costruzione di una relazione che lega un valore del derivato in $i\Delta t$ con tre diversi valori al tempo $(i+1)\Delta t$.

Si parte analizzando il caso di una put americana scritta su un titolo con dividend yield q noto. In tal caso si può sfruttare (3.3) con l'aggiunta del fattore q :

$$\frac{\partial f}{\partial t} + (r - q)S \frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf$$

Sia T la vita residua dell'opzione. Questa viene divisa in N sottointervalli di ampiezza Δt .

Si definisce poi S_{MAX} come il livello sufficientemente elevato del sottostante rispetto al quale il valore della put è nullo. Si discretizza il dominio del sottostante in M intervalli di ampiezza $\Delta S = S_{MAX}/M$. Risultano allora $M + 1$ prezzi azionari equispaziati tra loro del tipo $j\Delta S$, con $j = 0, \dots, M$.

Si può ora costruire una griglia, ponendo sulle ascisse gli $N + 1$ intervallini $i\Delta t$ ($i = 0, \dots, N$) e sulle ordinate $j\Delta S$. Ciò genera una griglia di $(N + 1) \times (M + 1)$ elementi dove la variabile $f_{i,j}$ indicherà il valore dell'opzione in corrispondenza del punto (i, j) .

4.2 Tecniche risolutive - Metodo Implicito delle differenze finite

Come sottolineato lo scopo del modello è ricondurre l'equazione differenziale di riferimento a delle equazioni alle differenze. Per farlo si parte esprimendo $\frac{\partial f_{i,j}}{\partial S}$ mediante approssimazione sia in avanti che indietro.

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j}}{\Delta S} \quad (4.1)$$

$$\frac{\partial f}{\partial S} = \frac{f_{i,j} - f_{i,j-1}}{\Delta S} \quad (4.2)$$

Si prende la media delle due approssimazioni ottenendo

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S} \quad (4.3)$$

Per quanto riguarda la derivata temporale del valore del derivato verrà usata un'approssimazione in avanti del tipo

$$\frac{\partial f}{\partial t} = \frac{f_{i+1,j} - f_{i,j}}{\Delta t}. \quad (4.4)$$

Manca da esplicitare la derivata seconda rispetto a S che risulta data da

$$\begin{aligned} \frac{\partial^2 f}{\partial S^2} &= \frac{(\frac{f_{i,j+1} - f_{i,j}}{\Delta S} - \frac{f_{i,j} - f_{i,j-1}}{\Delta S})}{\Delta S} \\ \frac{\partial^2 f}{\partial S^2} &= \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta S^2} \end{aligned} \quad (4.5)$$

Sostituendo (4.3), (4.4) e (4.5) nell'equazione differenziale di riferimento si ottiene

$$a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1} = f_{i+1,j} \quad (4.6)$$

$$\begin{aligned} a_j &= \frac{1}{2}(r - q)j\Delta t - \frac{1}{2}\sigma^2 j^2 \Delta t, \\ b_j &= 1 + \sigma^2 j^2 \Delta t + r\Delta t, \\ c_j &= -\frac{1}{2}(r - q)j\Delta t - \frac{1}{2}\sigma^2 j^2 \Delta t, \\ S &= j\Delta S. \end{aligned}$$

In tale equazione si noti che j varia tra 1 e $M - 1$ ed i varia tra 0 e $N - 1$, poiché nel caso temporale non compare il termine $i - 1$ ma solo $i + 1$.

Per applicare l'iteratività vanno esplicitati i valori della put nei bordi della griglia, in particolare per $S = 0$, $S = S_{MAX}$ e per $t = T$. In quest'ultimo caso è noto il payoff della put : $f_{N,j} = \max(K - S, 0) = \max(K - j\Delta S, 0)$, per $j = 0, \dots, M$.

Per $S = 0$ il valore è dato dal prezzo d'esercizio: $f_{i,0} = K$, per $i = 0, \dots, N$.

Infine, come già sottolineato, il valore della put in corrispondenza di S_{MAX} è nullo:

$f_{i,M} = 0$, per $i = 0, \dots, N$.

È possibile notare che per $i = N - 1$, cioè per i punti in corrispondenza di $T - \Delta t$, l'equazione (4.6) è un sistema di $M - 1$ equazioni simultanee. Infatti i risulta fissato e pari a $N - 1$, mentre j varia tra 1 e $M - 1$. Poiché i valori della funzione f sui bordi sono assegnati, l'equazione risulta definita in $M - 1$ incognite. Ne segue quindi un sistema di $M - 1$ equazioni in $M - 1$ incognite.

Risulta un vettore soluzione

$$(f_{N-1,0} = K, f_{N-1,1}, \dots, f_{N-1,M-1}, f_{N-1,M} = 0).$$

Ogni componente va confrontata col valore derivante dall'esercizio anticipato $K - j\Delta S$. Qualora $f_{N-1,j} < K - j\Delta S$, l'esercizio anticipato risulta conveniente per $i = N - 1$. In tal caso si sostituisce a $f_{N-1,j}$ il valore dell'esercizio anticipato. Viceversa per $f_{N-1,j} \geq K - j\Delta S$.

Questo procedimento va proseguito a ritroso, considerando poi l'intervallo temporale $T - 2\Delta t$ e così via. Al termine del processo si ricava il vettore relativo al tempo attuale: $(f_{0,1}, f_{0,2}, \dots, f_{0,M-1})$.

4.3 Tecniche risolutive - Metodo Esplicito delle differenze finite

Questo metodo è simile al precedente, si procede sempre sostituendo all'equazione differenziale di riferimento i valori delle derivate parziali calcolati mediante approssimazione. Nel metodo esplicito delle differenze finite si suppone che le derivate parziali di f rispetto al sottostante siano le stesse sia per i punti (i, j) che per i punti $(i + 1, j)$, risultando

$$\frac{\partial f}{\partial S} = \frac{f_{i+1,j+1} - f_{i+1,j-1}}{2\Delta S} \quad (4.7)$$

$$\frac{\partial^2 f}{\partial S^2} = \frac{f_{i+1,j+1} - 2f_{i+1,j} + f_{i+1,j-1}}{\Delta S^2} \quad (4.8)$$

Sfruttando (4.7) e (4.8) è possibile ottenere la seguente equazione differenziale:

$$f_{i,j} = a^*_j f_{i+1,j-1} + b^*_j f_{i+1,j} + c^*_j f_{i+1,j+1} \quad (4.9)$$

$$\begin{aligned} a^*_j &= \frac{1}{1+r\Delta t} \left[-\frac{1}{2}(r-q)j\Delta t + \frac{1}{2}\sigma^2 j^2 \Delta t \right], \\ b^*_j &= \frac{1}{1+r\Delta t} \left(1 - \sigma^2 j^2 \Delta t \right), \\ c^*_j &= \frac{1}{1+r\Delta t} \left[\frac{1}{2}(r-q)j\Delta t + \frac{1}{2}\sigma^2 j^2 \Delta t \right]. \end{aligned}$$

Si noti che, a differenza del metodo precedente, qui si può calcolare direttamente il valore i -esimo come combinazione lineare dei valori al passo temporale successivo. Non c'è quindi bisogno di risolvere alcun sistema lineare. Nel caso implicito invece i nodi di un dato passo temporale dipendono strettamente dai nodi adiacenti, il che rende impossibile l'attuazione di un approccio risolutivo diretto.

Affinchè il metodo esplicito risulti stabile, è necessario imporre una condizione sui coefficienti dello schema, in particolare su b^*_j . In genere si richiede che i coefficienti siano non negativi e che la loro somma sia pari a uno, il che porta a una restrizione sul passo temporale Δt . In particolare, una condizione sufficiente di stabilità è

$$\sigma^2 j^2 \Delta t \leq 1. \quad (4.10)$$

4.4 Approccio Informatico al Metodo delle differenze finite

Se implementato computazionalmente, il metodo delle differenze finite risulta più efficiente quando applicato a $\ln S$.

In tal caso l'equazione differenziale che descrive $Z = \ln S$ è

$$\frac{\partial f}{\partial t} + (r - q - \frac{\sigma^2}{2}) \frac{\partial f}{\partial Z} + \frac{1}{2} \sigma^2 \frac{\partial^2 f}{\partial Z^2} = r f.$$

La si può ottenere applicando la regola della catena

$$\frac{\partial f}{\partial S} = \frac{\partial f}{\partial Z} \frac{\partial Z}{\partial S} = \frac{\partial f}{\partial Z} \frac{1}{S}.$$

Derivando nuovamente rispetto a S , si ottiene

$$\frac{\partial^2 f}{\partial S^2} = \frac{1}{S^2} \left(\frac{\partial^2 f}{\partial Z^2} - \frac{\partial f}{\partial Z} \right).$$

Sostituendo tali espressioni in (3.3) considerando anche dividend yield q noto si ha

$$\frac{\partial f}{\partial t} + (r - q) S \frac{\partial f}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = r f,$$

da cui

$$\frac{\partial f}{\partial t} + (r - q) S \frac{\partial f}{\partial Z} \frac{1}{S} + \frac{1}{2} \sigma^2 S^2 \left(\frac{1}{S^2} \left(\frac{\partial^2 f}{\partial Z^2} - \frac{\partial f}{\partial Z} \right) \right) = r f,$$

$$\frac{\partial f}{\partial t} + (r - q) \frac{\partial f}{\partial Z} + \frac{1}{2} \sigma^2 \left(\frac{\partial^2 f}{\partial Z^2} - \frac{\partial f}{\partial Z} \right) = r f,$$

$$\frac{\partial f}{\partial t} + \left(r - q - \frac{\sigma^2}{2} \right) \frac{\partial f}{\partial Z} + \frac{1}{2} \sigma^2 \frac{\partial^2 f}{\partial Z^2} = r f.$$

Applicando il metodo implicito delle differenze finite si ha

$$\alpha_j f_{i,j-1} + \beta_j f_{i,j} + \gamma_j f_{i,j+1} = f_{i+1,j} \quad (4.11)$$

$$\begin{aligned}
\alpha_j &= \frac{\Delta t}{2\Delta Z} \left(r - q - \frac{\sigma^2}{2} \right) - \frac{\Delta t}{2\Delta Z^2} \sigma^2, \\
\beta_j &= 1 + \frac{\Delta t}{\Delta Z^2} \sigma^2 + r\Delta t, \\
\gamma_j &= -\frac{\Delta t}{2\Delta Z} \left(r - q - \frac{\sigma^2}{2} \right) - \frac{\Delta t}{2\Delta Z^2} \sigma^2.
\end{aligned}$$

Nel caso dell'applicazione del metodo esplicito si ha

$$f_{i,j} = \alpha_j^* f_{i+1,j-1} + \beta_j^* f_{i+1,j} + \gamma_j^* f_{i+1,j+1} \quad (4.12)$$

$$\begin{aligned}
\alpha_j^* &= \frac{1}{1+r\Delta t} \left[-\frac{\Delta t}{2\Delta Z} \left(r - q - \frac{\sigma^2}{2} \right) + \frac{\Delta t}{2\Delta Z^2} \sigma^2 \right], \\
\beta_j^* &= \frac{1}{1+r\Delta t} \left(1 - \frac{\Delta t}{\Delta Z^2} \sigma^2 \right), \\
\gamma_j^* &= \frac{1}{1+r\Delta t} \left[\frac{\Delta t}{2\Delta Z} \left(r - q - \frac{\sigma^2}{2} \right) + \frac{\Delta t}{2\Delta Z^2} \sigma^2 \right].
\end{aligned}$$

5. Metodo Monte Carlo

Nonostante l'efficacia dei metodi basati sulle differenze finite, la loro applicabilità trova un limite strutturale legato all'alta dimensionalità: all'aumentare delle variabili o della complessità del payoff, il numero di nodi della griglia cresce esponenzialmente, rendendo il calcolo computazionalmente proibitivo. Inoltre, i modelli precedenti faticano a gestire opzioni path-dependent, il cui valore dipende dall'intero percorso seguito dal prezzo e non solo dal valore finale.

Il **Metodo Monte Carlo** nasce per risolvere esattamente questo problema, introducendo un nuovo approccio. A differenza dei metodi precedenti, non si cerca più di ri-

solvere numericamente un'equazione differenziale nè di costruire una struttura rigida di nodi predefiniti per approssimare il valore del derivato. Al contrario, si sfrutta la simulazione diretta di un elevato numero di possibili scenari di mercato, permettendo di gestire con estrema flessibilità payoff complessi e variabili multiple, laddove gli approcci deterministici e le griglie di calcolo risulterebbero inadeguati.

5.1 Definizione formale del modello

Il metodo Monte Carlo affonda le sue radici storiche nelle ricerche condotte presso i laboratori di Los Alamos durante gli anni '40, nate dall'esigenza di risolvere complessi problemi di diffusione dei neutroni attraverso l'uso della statistica computazionale (Metropolis & Ulam 1949).

Un esempio classico per illustrare l'efficacia di tale approccio è attraverso il calcolo numerico di π . Per ottenere quest'ultimo si può calcolare la probabilità che, lanciando un dado, questo cada all'interno di un cerchio iscritto in un quadrato ad area unitaria.

Effettuando un numero molto elevato di lanci, si riesce a stimare l'area del cerchio; moltiplicandola per quattro si ottiene π (area cerchio = $r^2\pi = (0,5)^2\pi$).

In ambito finanziario l'introduzione sistematica di questo approccio numerico si deve principalmente agli studi dell'economista Boyle, che hanno dimostrato come la valutazione dei derivati potesse essere affrontata simulando le traiettorie stocastiche del prezzo del sottostante (Boyle 1977, Boyle et al. 1997, Broadie et al. 1997).

In questa cornice, il valore dell'opzione emerge come la media attualizzata dei payoff calcolati lungo ogni possibile sentiero simulato.

Questo modello manifesta vantaggi come facilità d'implementazione, infatti la logica di base risulta relativamente semplice. Inoltre ha un'alta dimensionalità. Fornisce una stima accurata anche per opzioni scritte su più sottostanti, dove altri metodi diventano complessi o non sono applicabili. Questo accade frequentemente nella pratica finanziaria,

ad esempio per strumenti esotici, prodotti strutturati o modelli multidimensionali. La rilevanza del metodo non sta quindi nella precisione teorica, ma nella sua scalabilità e adattabilità. Monte Carlo consente di incorporare facilmente caratteristiche realistiche, come correlazioni tra asset o dinamiche non lineari, che sarebbero difficili da trattare con altri metodi.

Comporta ovviamente anche degli svantaggi; ad esempio il modello è caratterizzato da convergenza lenta.

Si definisce infatti l'errore standard come il rapporto tra la deviazione standard dei payoff attualizzati (ottenuti attraverso la simulazione) e la radice quadrata delle osservazioni effettuate. L'errore allora diminuisce proporzionalmente al numero di osservazioni. Per avere un'elevata precisione servono quindi milioni d'analisi che generano un costo computazionale elevato. Si ha inoltre difficoltà nel caso delle opzioni americane, poiché il modello base non gestisce bene l'ottimizzazione delle stesse. Infine esso ha un'elevata sensibilità ai parametri.

La qualità dei risultati dipende dalla corretta modellizzazione del processo stocastico; errori nei parametri portano a stime poco precise.

5.2 Tecniche risolutive

Questo modello viene spesso implementato a partire da migliaia di estrazioni casuali ϵ da una normale standardizzata. Generalmente viene sviluppato tramite foglio elettronico.

Una volta ottenute queste estrazioni si va a calcolare S_T come segue:

$$S_T = S_0 e^{\frac{\sigma^2}{2}T + \sigma\sqrt{T}\epsilon},$$

con T , σ , K e S_0 noti.

Si valuta ora il payoff delle call e delle put come $\max(S_T - K, 0)$ e $\max(K - S_T, 0)$ rispettivamente.

Si esplicita per ogni nuova estrazione la media cumulata; al termine delle simulazioni si avrà la media dei payoff per tutte le osservazioni effettuate.

Quest'ultima va poi scontata ottenendo il valore attuale dell'opzione.

6. Applicazione e analisi comparativa dei modelli

Al fine di validare la robustezza dei modelli teorici precedentemente esposti, si è proceduto all'implementazione computazionale in ambiente Python (codice in Appendice) effettuando un'analisi circa il prezzo corretto di varie tipologie d'opzioni.

L'obiettivo di tale analisi è confrontare l'accuratezza e l'efficienza dei diversi approcci algoritmici.

6.1 Logica algoritmica e implementazione del software

La struttura computazionale sviluppata è stata declinata in modo da recepire le differenti logiche matematiche sottostanti a ciascun modello di valutazione.

La funzione dedicata al **modello di Black-Scholes-Merton** opera in tempo costante, implementando la formula analitica mediante l'utilizzo della funzione di ripartizione della normale standard $N(d_i)$.

Al contrario, il **modello binomiale** è strutturato su una logica ricorsiva: il codice inizializza un vettore di prezzi a scadenza e, attraverso un ciclo iterativo `for`, procede a ritroso attualizzando i payoff attesi. Attraverso questa scomposizione discreta si stima il processo stocastico a tempo continuo.

La riduzione dell'errore di approssimazione risulta strettamente legata all'incremento degli intervalli temporali (supposti pari a 1000), riflettendo la nota proprietà di convergenza del modello binomiale.

La **simulazione Monte Carlo** adotta una filosofia differente, grazie alle funzioni avanzate della libreria `numpy`, che permettono di elaborare grandi blocchi di dati simultaneamente. Invece di eseguire cicli singoli per ogni traiettoria, il codice genera simultaneamente un vettore di centomila variabili casuali normali standard ϵ_i , proiettando in un unico passaggio tutti i possibili percorsi di prezzo che il titolo potrebbe seguire fino alla scadenza.

Per ogni sentiero simulato si procede al calcolo del relativo payoff; il valore attuale dell'opzione emerge infine calcolando la media aritmetica di tali payoff e applicando il fattore di attualizzazione e^{-rT} . Va sottolineato che, basandosi su un vettore di variabili casuali normali, ristampando più volte il risultato, questo sarà diverso dai precedenti.

Infine, l'implementazione delle **differenze finite** affronta la risoluzione della PDE di Black-Scholes (3.3) previa trasformazione logaritmica della variabile prezzo ($Z = \ln S$). Tale trasformazione è essenziale nel codice per rendere uniforme la struttura dei dati; trasformando i prezzi in logaritmi, il modello può operare su una griglia di calcolo regolare ed equispaziata, dove le variazioni percentuali del titolo vengono elaborate con una coerenza matematica superiore rispetto all'utilizzo dei prezzi assoluti. Mentre il metodo esplicito aggiorna la soluzione tramite una combinazione lineare dei valori futuri già calcolati, lo schema implicito risolve un sistema di equazioni lineari ad ogni intervallo temporale tramite il modulo `linalg.solve`. Questo metodo garantisce una robustezza numerica superiore, permettendo di ottenere stime affidabili anche con suddivisioni della griglia più rade.

6.2 Analisi critica dei risultati ottenuti

La prima analisi riguarda il prezzaggio di un'opzione call europea con i seguenti parametri: prezzo corrente del sottostante $S_0 = 42\$$, prezzo d'esercizio $K = 40\$$, scadenza $T = 0,5$ (sei mesi), tasso d'interesse privo di rischio e volatilità annua pari a $r = 10\%$ e $\sigma = 20\%$ rispettivamente. I risultati ottenuti, sintetizzati nella Tabella 1, mostrano una convergenza estremamente precisa verso il valore teorico ottenuto tramite BSM di 4,7594\$.

Tale valore viene assunto come benchmark in quanto deriva da una soluzione analitica esatta del problema, sotto le ipotesi del modello, e rappresenta quindi il riferimento più affidabile per valutare la precisione e la stabilità delle metodologie numeriche impiegate.

Tabella 1: Confronto dei risultati numerici per la call in esame.

Metodo	Prezzo (\$)	Punti di forza	Punti di debolezza
Black-Scholes	4.7594	Precisione assoluta	Rigidità delle ipotesi
Binomiale (1000 step)	4.7598	Versatilità (Americane)	Oscillazioni numeriche
Monte Carlo (100k iter.)	4.7579	Facilità implementativa	Errore statistico
Diff. Finite (Implicito)	4.7571	Stabilità numerica	Complessità algebrica

L'analisi visiva prodotta dagli algoritmi implementati non si limita ad una validazione numerica, ma conferma le proprietà asintotiche e le caratteristiche di convergenza intrinseche ai diversi modelli di calcolo.

Attraverso il codice si può verificare computazionalmente come, all'aumentare dei passi temporali presi in considerazione, ci si avvicini al benchmark dato da BSM.

Nel modello binomiale ad esempio si osserva come, nei primi passi temporali, il prezzo dell'opzione oscilli marcatamente attorno al valore analitico di Black-Scholes. Tali fluttuazioni, tipiche della struttura discreta dell'albero di Cox-Ross-Rubinstein, tendono a smorzarsi progressivamente all'aumentare della granularità temporale, cioè all'aumentare dei passi temporali che costituiscono l'albero.

L'evidenza empirica (Figura 2) suggerisce che un numero di step superiore a 500 rappresenta quel compromesso in cui l'errore di discretizzazione diventa trascurabile rispetto allo sforzo computazionale richiesto.

Il modello binomiale rappresenta quindi il ponte ideale tra teoria e pratica. Il suo punto di forza risiede nella trasparenza del sentiero seguito dei prezzi; ogni nodo dell'albero è visibile ed interpretabile.

A differenza del modello BSM è adottabile nel caso delle opzioni americane, dove la possibilità di esercizio anticipato richiede un semplice controllo puntuale in ogni passo. Di contro, soffre di un'efficienza computazionale limitata quando il numero di step diventa estremamente elevato, specialmente in presenza di opzioni con più sottostanti.

Proseguendo nell'analisi dei singoli modelli, è possibile delineare un quadro approfondito dei relativi punti di forza e delle criticità operative.

Il modello di Black-Scholes-Merton rimane il benchmark di riferimento per la sua eleganza formale. Il suo principale punto di forza risiede nella velocità d'esecuzione, essendo basato su una funzione esplicita che non richiede iterazioni.

Tuttavia, la sua debolezza è legata alla sua estrema rigidità; le ipotesi di tale modello sono infatti molto stringenti: mondo a volatilità e tassi costanti, fallendo nel catturare fenomeni di mercato reali più evidenti ad esempio nelle opzioni americane. In termini di implementazione, è il più semplice da calcolare computazionalmente, poiché la sua natura chiusa si presta perfettamente alla scrittura di formule semplici e compatte.

La Simulazione Monte Carlo è uno strumento potente poiché in grado di valutare opzioni path-dependent, ovvero il cui payoff dipende dall'intero percorso del prezzo e non solo dal valore finale. Inoltre si può notare dalle poche righe di codice allegato in Appendice, come tale metodo risulti implementativamente semplice.

La debolezza strutturale è legata all'errore standard come sottolineato nel capitolo dedicato. Nell'esempio considerato possiamo infatti notare che per ottenere una buona

approssimazione del benchmark si è necessitato di centomila iterazioni. C'è appunto un trade-off tra precisione e costo computazionale.

Al riguardo risulta interessante approfondire la convergenza del metodo Monte Carlo e del modello binomiale all'aumentare delle simulazioni e dei passi temporali rispettivamente.

Per evidenziare tale processo di convergenza verso il valore teorico di riferimento, si riportano di seguito i grafici che mostrano l'andamento del prezzo dell'opzione al variare dei parametri computazionali, basati sui risultati ottenuti sempre tramite implementazione in ambiente Python.

N	Prezzo	Errore
10	4.800306	0.040883
25	4.772077	0.012654
50	4.761510	0.002088
100	4.761818	0.002396
250	4.760343	0.000921
500	4.759342	0.000080
1000	4.759817	0.000395

(a) Binomiale

M	Prezzo	Errore
1000	4.684157	0.075265
5000	4.694332	0.065091
10000	4.786238	0.026816
50000	4.738426	0.020996
100000	4.748967	0.010455

(b) Monte Carlo

Tabella 2: Confronto della convergenza dei metodi numerici

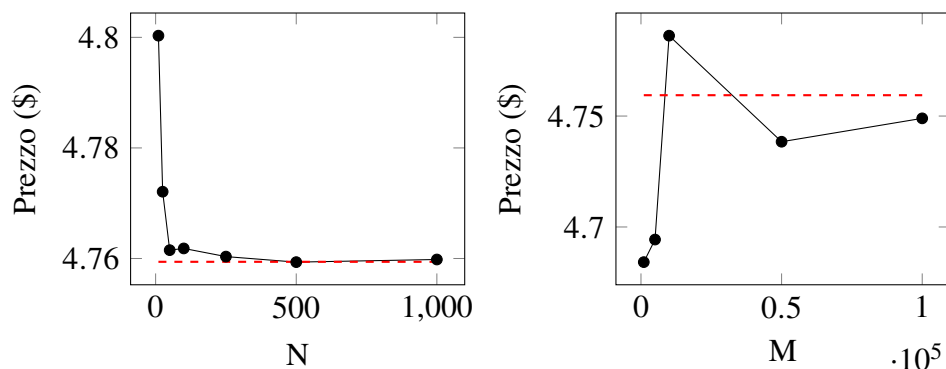


Figura 2: Confronto della convergenza all'aumentare delle iterazioni

Per il modello binomiale, i risultati evidenziano una convergenza regolare e monotona verso il valore teorico all'aumentare del numero di passi N . Questo comportamento riflette la natura del metodo, che approssima il processo continuo del sottostante mediante una discretizzazione temporale: all'incremento di N , la struttura ad albero diventa più fine e l'approssimazione migliora progressivamente. L'errore osservato tende a ridursi rapidamente, confermando l'elevata efficienza del metodo in contesti a bassa dimensionalità.

Nel caso del metodo Monte Carlo, la convergenza risulta più lenta e caratterizzata da oscillazioni dovute alla componente stocastica. I risultati ottenuti mostrano chiaramente come l'approssimazione migliori all'aumentare del numero di simulazioni M , in accordo con la teoria della legge dei grandi numeri. Tuttavia, la velocità di convergenza è limitata, poiché come precedentemente enunciato l'errore decresce con ordine $O(1/\sqrt{M})$.

Ciò rende necessario un significativo incremento del numero di simulazioni per ottenere miglioramenti apprezzabili.

Il confronto tra i due metodi mette in evidenza un aspetto cruciale: mentre il modello binomiale garantisce una convergenza più rapida in questo specifico contesto, il metodo Monte Carlo offre maggiore flessibilità ed è particolarmente adatto a problemi più complessi, come quelli ad alta dimensionalità o con payoff non lineari.

In conclusione, entrambi i metodi convergono correttamente al valore teorico di riferimento, ma con velocità e caratteristiche differenti, pienamente coerenti con le aspettative teoriche.

Per il metodo alle differenze finite è necessario effettuare una distinzione tra implicito ed esplicito. Il punto di forza del metodo implicito è la sua stabilità, che permette di gestire scenari dove altri metodi numerici fallirebbero per divergenza. La debolezza risiede nell'elevata barriera matematica iniziale: la necessità di risolvere sistemi lineari complessi rende l'implementazione più soggetta ad errori di programmazione.

6.3 Logica algoritmica: il caso della Put Americana

Per validare l'efficacia dei modelli numerici nella valutazione di derivati complessi, si è proceduto anche all'implementazione in ambiente Python di algoritmi dedicati al pricing di una put americana.

A differenza delle opzioni europee, la natura americana del contratto introduce la possibilità di esercizio anticipato, rendendo necessaria l'adozione di schemi ricorsivi o iterativi che non ammettono una soluzione analitica chiusa come il classico modello di Black-Scholes-Merton. Va sottolineato come, nel caso delle opzioni americane, non si possa adottare il modello appena citato e nemmeno il metodo Monte Carlo, in quanto non ottimali.

L'obiettivo di tale analisi è confrontare la convergenza dell'approccio binomiale rispetto ai metodi alle differenze finite, utilizzando il primo come benchmark di riferimento data la sua nota precisione asintotica per questa tipologia di contratti.

La sfida computazionale nel prezzaggio di un'opzione americana risiede nella risoluzione di un problema di frontiera libera. In ogni istante temporale precedente la scadenza, il detentore deve scegliere se mantenere l'opzione o esercitarla immediatamente.

Nel **modello binomiale**, tale logica è integrata direttamente nel ciclo iterativo a ritroso. Per ogni nodo dell'albero, il software confronta il valore attualizzato atteso dai nodi futuri con il payoff immediato $K - S_t$. Matematicamente, il codice applica l'operatore:

$$V_i = \max(V_{cont}, K - S_i) \quad (6.1)$$

L'implementazione delle **differenze finite** affronta la PDE di Black-Scholes operando sulla variabile trasformata $Z = \ln S$, come già sottolineato.

In termini di programmazione, il codice sfrutta la potenza di numpy per la gestione vettoriale dei coefficienti α , β e γ . Un aspetto critico emerso in fase di sviluppo del

metodo esplicito riguarda la sensibilità dello schema alla condizione di stabilità. Affinchè l'errore di discretizzazione non si amplifichi enormemente, è necessario rispettare il vincolo numerico legato al rapporto tra lo step temporale e lo spazio: $\Delta t \leq \frac{\Delta z^2}{\sigma^2}$.

Nel codice implementato, questa condizione emerge nella definizione del coefficiente centrale β_s : qualora Δt risultasse troppo elevato rispetto a Δz^2 , β_s assumerebbe un valore negativo, portando il sistema alla divergenza numerica (prezzi infiniti o non fisici).

In questo scenario, il modello produce un paradosso fisico e finanziario. Un peso negativo implicherebbe infatti che un aumento del valore futuro dell'opzione in un determinato nodo provochi una diminuzione del suo valore attuale. Tale "sottrazione di valore" dal futuro è impossibile in un mercato privo di arbitraggio e viola le leggi della diffusione.

Al contrario, il **metodo implicito** risolve tale criticità tramite la costruzione di una matrice tridiagonale. Utilizzando la funzione `numpy.linalg.solve`, il sistema viene risolto in modo molto più stabile, permettendo di ottenere stime affidabili anche con griglie temporali meno dense.

Per integrare la natura americana, il valore ai nodi viene aggiornato ad ogni step tramite la funzione `numpy.maximum`, che confronta il vettore dei prezzi calcolato con il payoff intrinseco ($K - \exp(z_grid)$), forzando il rispetto del vincolo di esercizio anticipato su tutta la griglia logaritmica.

Questo garantisce che, indipendentemente dal metodo adottato, il valore dell'opzione incorpori correttamente il premio per l'esercizio anticipato.

6.4 Analisi critica dei risultati ottenuti

Per testare i modelli è stata simulata una put americana con i seguenti parametri: $S_0 = 45\$$, $K = 50\$$, $T = 0.75$, $r = 3\%$ e $\sigma = 25\%$. La scelta di uno strike superiore al prezzo corrente è volta a testare specificamente la capacità dei modelli di catturare il valore dell'esercizio anticipato.

Tabella 3: Confronto dei risultati numerici per la put americana.

Metodo	Prezzo (\$)	Osservazioni Tecniche
Binomiale (2000 step)	6.4990	Elevata precisione, utilizzato come benchmark.
Diff. Finite (Esplicito)	6.4963	Rapido, ma sensibile alla condizione di stabilità.
Diff. Finite (Implicito)	6.4956	Massima stabilità numerica, converge anche con griglie rade.

Come emerso, l'albero binomiale riesce a catturare in maniera naturale la possibilità di esercizio anticipato, incorporando localmente la decisione ottimale ad ogni nodo. Questo approccio discreto, pur essendo concettualmente semplice, si dimostra estremamente efficace dal punto di vista numerico, specialmente in contesti unidimensionali come quello analizzato.

La scelta di adottare $N = 2000$ step non è arbitraria, ma si basa sull'evidenza empirica osservata nel caso della call europea, dove già per valori superiori a 500 step si otteneva una convergenza estremamente accurata.

Nel caso dell'opzione americana, tuttavia, la presenza della possibilità di esercizio anticipato introduce una maggiore complessità nella struttura del problema.

Per tale motivo, si è deciso di incrementare ulteriormente il numero di step, al fine di garantire una discretizzazione sufficientemente fine da catturare con precisione la frontiera di esercizio ottimale.

Passando al metodo delle differenze finite esplicite, si osserva come il prezzo ottenuto sia molto vicino al benchmark, ma ciò avviene sotto il vincolo stringente della condizione di stabilità. Questo aspetto rappresenta il principale limite operativo dello schema esplicito come analizzato.

Infatti, la necessità di rispettare il vincolo $\Delta t \leq \frac{\Delta z^2}{\sigma^2}$ (derivante da (4.10) e modificata per l'adozione del log-prezzo) implica, nella pratica, l'utilizzo di griglie temporali molto dense, con conseguente aumento del costo computazionale.

Si evidenzia quindi un trade-off tra velocità computazionale per singolo step e numero totale di iterazioni richieste.

Inoltre, la sensibilità dello schema a perturbazioni dei parametri discreti suggerisce una minore robustezza in contesti più complessi o in presenza di condizioni di mercato più estreme.

Il metodo implicito delle differenze finite, al contrario, mostra una stabilità significativamente superiore. La costruzione del sistema lineare tridiagonale e la sua risoluzione garantiscono che lo schema sia incondizionatamente stabile rispetto alla scelta dei passi di discretizzazione.

Questo consente di utilizzare griglie temporali meno dense senza compromettere la qualità del risultato, rendendo il metodo particolarmente adatto ad applicazioni in cui il tempo rappresenta un vincolo rilevante.

Un ulteriore aspetto da sottolineare riguarda l'integrazione del vincolo di esercizio anticipato nei metodi alle differenze finite.

Infatti l'aggiornamento del valore dell'opzione tramite l'operatore di massimo (`numpy.maximum`), può essere interpretato in termini più intuitivi come un controllo sistematico della convenienza dell'esercizio anticipato. In particolare, a ogni passo temporale e per ogni punto della griglia, si confrontano due quantità: da un lato il valore dell'opzione calcolato numericamente tramite modello e dall'altra il guadagno immediato derivante dall'esercizio ($K - S_t$).

In questo modo, il prezzo dell'opzione non è determinato solo dalla soluzione della PDE di Black-Scholes, ma viene aggiustato imponendo che non sia mai inferiore al valore intrinseco, ossia al guadagno che si otterrebbe esercitando immediatamente l'opzione, pari a $(K - S_t)$ nel caso della put.

Questo riflette un principio economico rilevante, già richiamato nell'introduzione, secondo cui il detentore dell'opzione non manterrebbe il contratto qualora l'esercizio immediato risultasse più conveniente.

Il fatto che i risultati ottenuti siano in linea con quelli del modello binomiale suggerisce che questo meccanismo è stato implementato correttamente, e che il modello riesce

a incorporare in modo coerente il valore aggiuntivo legato alla possibilità di esercizio anticipato.

Come analizzato per la call, risulta interessante approfondire la convergenza dei due metodi.

Per quanto riguarda il metodo alle differenze finite, si è scelto di effettuare il confronto esclusivamente con lo schema implicito.

Si riportano di seguito i grafici relativi alla loro convergenza.

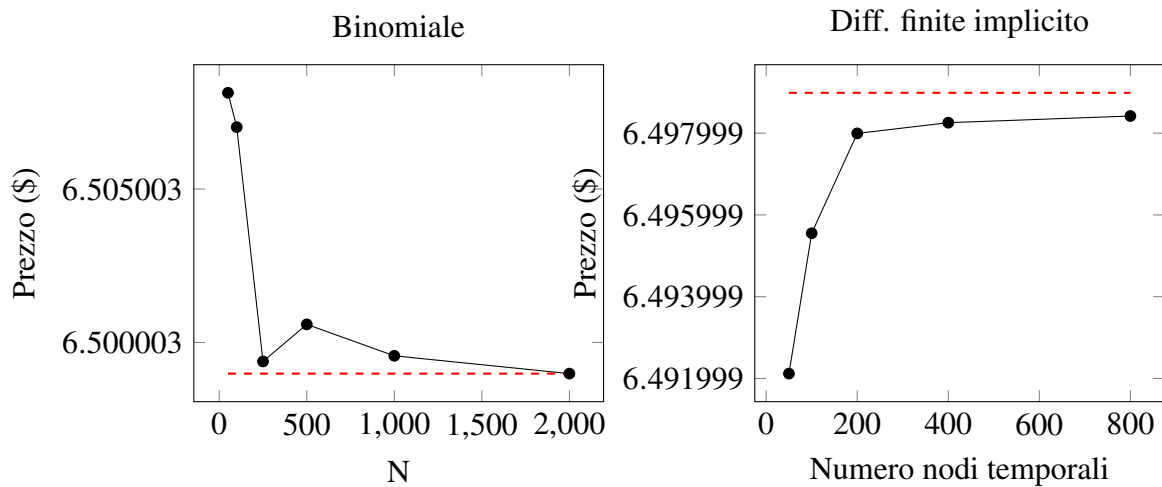


Figura 3: Confronto della convergenza

Dal confronto grafico emergono alcune considerazioni rilevanti.

Il modello binomiale mostra una convergenza regolare e monotona verso il valore di riferimento, come osservato anche nell'esempio precedente. Questo comportamento è coerente con la natura del metodo, che approssima il processo continuo tramite una discretizzazione sempre più fine.

L'aumento del numero di passi consente inoltre di approssimare con maggiore precisione la frontiera di esercizio anticipato, ossia la soglia temporale e di prezzo che separa la regione in cui è ottimale esercitare immediatamente l'opzione da quella in cui conviene mantenerla.

Il metodo alle differenze finite implicito evidenzia anch'esso una convergenza stabile, ma con una dinamica leggermente diversa. La presenza di una griglia spazio-temporale consente infatti di ottenere buoni livelli di precisione anche con un numero relativamente contenuto di nodi temporali. Questo riflette uno dei principali vantaggi dello schema implicito, ovvero la capacità di mantenere stabilità e accuratezza senza richiedere discretizzazioni estremamente fini.

Come emerge dalla Figura 3 infatti, utilizzando circa 600 nodi temporali si arriva ad un valore praticamente analogo a quello ottenuto dal benchmark.

Questo risultato rafforza la validità dell'implementazione numerica e conferma che il vincolo di esercizio anticipato è stato correttamente incorporato in entrambi i modelli.

In sintesi, mentre il modello binomiale garantisce maggiore trasparenza interpretativa e rappresenta un benchmark naturale, il metodo implicito delle differenze finite si distingue per efficienza numerica e robustezza, rendendolo particolarmente adatto a contesti computazionalmente più complessi.

A completamento dell'analisi della convergenza dei metodi numerici, risulta particolarmente significativo estendere il confronto non solo al valore dell'opzione, ma anche alle strutture implicite che ne derivano, come la frontiera di esercizio anticipato, (precedentemente citata) che rappresenta un elemento chiave nella caratterizzazione delle opzioni americane.

Il confronto tra la frontiera di esercizio anticipato ottenuta tramite il metodo alle differenze finite implicito e quella derivata dal modello binomiale, evidenzia una notevole coerenza tra approcci numericamente differenti.

Formalmente la curva $S^*(t)$, detta **early exercise boundary**, rappresenta il livello critico del prezzo del sottostante al di sotto del quale risulta ottimale esercitare anticipatamente un'opzione put americana. Tale curva separa quindi la regione di continuazione (dove è conveniente mantenere il derivato) dalla regione di esercizio (dove è conveniente l'esercizio anticipato).

Inoltre la curva è una funzione crescente rispetto al tempo a scadenza, riflette cioè la progressiva perdita di valore temporale dell'opzione.

In altre parole il grafico mostra una funzione crescente nel tempo poiché la riduzione della vita residua dell'opzione erode il beneficio dell'attesa, innalzando progressivamente il prezzo critico al quale conviene incassare il valore intrinseco anzichè mantenere la posizione.

Una trattazione più formale e dettagliata della sua derivazione teorica e delle sue proprietà viene riportata in Appendice.

In particolare, la soluzione basata sul metodo delle differenze finite produce una frontiera più regolare e continua, mentre il modello binomiale restituisce una versione intrinsecamente discreta della stessa quantità.

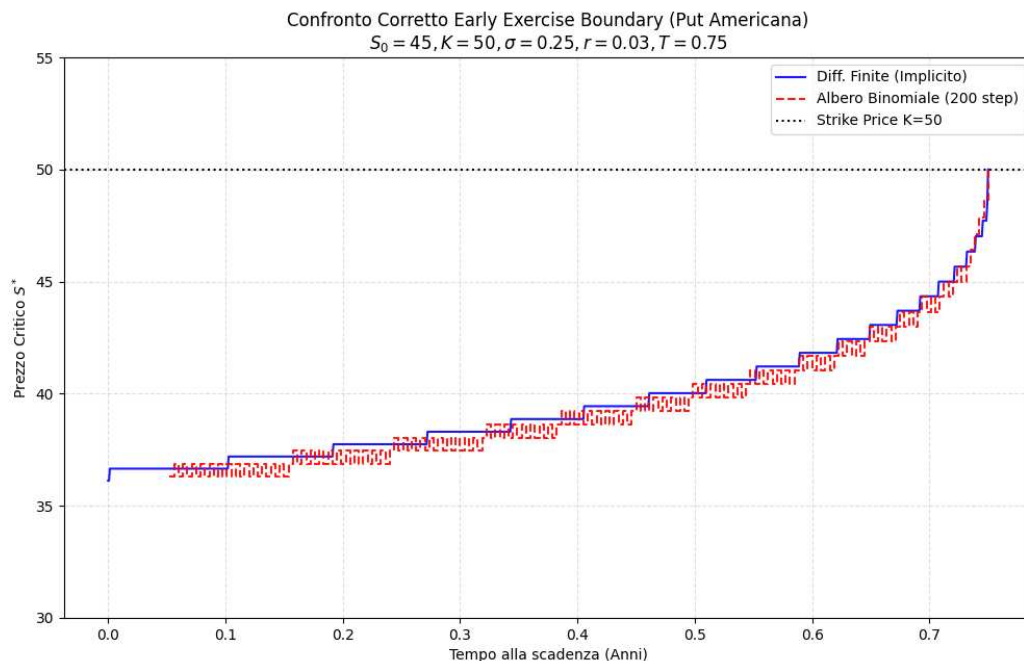


Figura 4

Come si può osservare nella Figura 4 emerge come la sovrapposizione delle due curve confermi la correttezza dell'implementazione numerica e la robustezza del risultato

rispetto alla scelta del metodo di discretizzazione.

Si sottolinea inoltre che l'estrazione numerica della frontiera è stata ottenuta attraverso l'implementazione computazionale in ambiente Python, facendo uso di algoritmi volti all'identificazione della regione di esercizio e alla ricostruzione temporale della curva $S^*(t)$.

In particolare, la regione di esercizio viene individuata mediante il confronto puntuale tra il valore di continuazione dell'opzione e il payoff intrinseco.

Per quanto riguarda l'implementazione del metodo delle differenze finite, il confine della regione di esercizio anticipato è identificato isolando i nodi della griglia in cui il valore dell'opzione coincide con il payoff immediato. Computazionalmente, ciò è realizzato individuando i punti in cui la differenza tra il valore calcolato e il payoff risulta trascurabile, implementando la condizione logica `np.abs(payload - v) < 1e-12`.

Questo approccio viene applicato per gestire le approssimazioni numeriche intrinseche al metodo delle differenze finite.

Infine, la ricostruzione della curva $S^*(t)$ viene ottenuta selezionando, per ciascun istante temporale, il valore massimo del sottostante appartenente alla regione di esercizio (il cosiddetto prezzo critico).

Tale operazione è effettuata mediante l'istruzione `np.where(np.abs(payload - v) < 1e-12)` sulla griglia spaziale ordinata o tramite `np.maximum(continuation, payload)` nel modello binomiale. Questo processo permette di tracciare con precisione l'evoluzione temporale della frontiera di esercizio anticipato, evidenziando la separazione tra la regione di holding (continuazione) e quella di esercizio immediato.

6.5 Analisi comparativa tra put europea e put americana

Per valutare l'impatto della facoltà di esercizio anticipato sul valore del derivato, risulta particolarmente interessante confrontare i risultati ottenuti nel caso della put euro-

pea con quelli precedentemente analizzati per la put americana. Il codice per il pricing dell'opzione europea viene fornito in Appendice.

Sono stati mantenuti invariati tutti i parametri di mercato:

$$S_0 = 45\$, \quad K = 50\$, \quad T = 0.75, \quad r = 3\%, \quad \sigma = 25\%$$

in modo tale che l'unica differenza tra i due contratti risieda nella struttura dell'esercizio.

Tabella 4: Confronto tra put europea e put americana

Metodo	Put Europea (\$)	Put Americana (\$)
Black-Scholes	6.2820	—
Binomiale	6.2819	6.4990
Diff. Finite Esplicito	6.2811	6.4963
Diff. Finite Implicito	6.2808	6.4956

Come emerge chiaramente dalla Tabella 4, il valore della put americana risulta sistematicamente superiore rispetto a quello della corrispondente put europea.

Questo risultato è coerente con la teoria finanziaria. La possibilità di esercitare anticipatamente l'opzione rappresenta infatti una flessibilità aggiuntiva concessa al detentore del contratto, che non può ridurre il valore ma soltanto incrementarlo.

Nel caso europeo, l'investitore è obbligato ad attendere la scadenza per realizzare il payoff dell'opzione. Di conseguenza il prezzo del derivato riflette esclusivamente il valore atteso del payoff finale sotto le ipotesi di neutralità al rischio.

Nel caso americano, invece, il possessore può decidere in ogni istante se mantenere la posizione oppure esercitare immediatamente il contratto. Tale facoltà introduce un valore opzionale aggiuntivo che viene incorporato nel prezzo attraverso il cosiddetto **premio per l'esercizio anticipato**.

Dal confronto numerico si osserva infatti che:

$$6.4990 - 6.2819 = 0.2171.$$

La differenza di 22 centesimi per contratto rappresenta il valore economico attribuibile alla possibilità di esercizio anticipato.

Dal punto di vista numerico, la differenza tra i due problemi risulta ancora più significativa.

La put europea è descritta da una PDE lineare standard e non richiede alcun controllo addizionale durante la procedura iterativa. Di conseguenza sia il metodo binomiale sia i metodi alle differenze finite convergono direttamente verso la soluzione teorica di Black-Scholes. D'altra parte la put americana introduce un problema di frontiera libera. In ogni nodo della discretizzazione è necessario verificare se il valore di continuazione dell'opzione sia inferiore al payoff immediato.

Dal punto di vista computazionale, ciò comporta una maggiore complessità e richiede una discretizzazione più accurata per approssimare correttamente la frontiera di esercizio ottimale.

Il confronto tra i risultati ottenuti conferma quindi che la principale difficoltà numerica associata alle opzioni americane non deriva dalla non applicabilità della PDE di Black-Scholes, ma dalla necessità di incorporare correttamente il vincolo di esercizio anticipato.

In assenza di tale vincolo, come nel caso europeo, tutti i metodi convergono rapidamente verso la soluzione analitica. Quando invece viene introdotta la possibilità di esercizio anticipato, il problema assume una struttura significativamente più complessa, richiedendo tecniche numeriche in grado di identificare correttamente la frontiera libera e il corrispondente valore ottimale dell'opzione. Tale complessità è tuttavia compensata dalla maggiore flessibilità offerta all'investitore, che può scegliere di esercitare l'opzione nel momento più conveniente, beneficiando di una protezione più efficace contro

movimenti sfavorevoli del sottostante.

7. Analisi empirica sul titolo Apple

Nei capitoli precedenti sono stati analizzati i principali modelli di pricing delle opzioni, con particolare riferimento alla valutazione di una call europea e di una put americana. In tale contesto, le applicazioni numeriche sono state condotte utilizzando dati simulati, con l'obiettivo di evidenziare il funzionamento dei diversi modelli teorici e le relative differenze in termini di prezzo e sensitività.

Il presente capitolo introduce invece un'analisi empirica basata su dati reali di mercato relativi al titolo Apple, con l'obiettivo di stimare in modo coerente i parametri necessari al pricing, in particolare la volatilità. In questo modo si passa da un contesto puramente teorico e simulativo ad un'impostazione più aderente alla realtà dei mercati finanziari.

7.1 Calibrazione dei parametri: il caso della call europea

I dati utilizzati consistono in una serie storica di prezzi giornalieri del titolo, scaricati tramite la libreria `yfinance`, che consente l'accesso ai dati forniti da Yahoo Finance.

È stata considerata una finestra temporale di cinque anni (08/05/2021 – 08/05/2026), scelta al fine di ottenere una stima sufficientemente stabile della volatilità storica.

A partire dai prezzi di chiusura, si costruisce la serie dei rendimenti logaritmici, definiti come:

$$r_t = \ln \left(\frac{S_t}{S_{t-1}} \right), \quad (7.1)$$

dove S_t rappresenta il prezzo del sottostante al tempo t .

L'utilizzo dei rendimenti logaritmici è coerente con l'ipotesi di modellizzazione introdotte nei capitoli precedenti.

La volatilità viene quindi stimata come deviazione standard dei rendimenti logaritmici giornalieri e successivamente annualizzata secondo la relazione:

$$\sigma = \sigma_{\text{daily}} \sqrt{252}, \quad (7.2)$$

dove il fattore 252 rappresenta il numero convenzionale di giorni di trading in un anno.

Il valore ottenuto costituisce un input fondamentale nei modelli di valutazione delle opzioni precedentemente analizzati. In particolare, per la call europea la volatilità entra direttamente nella formula chiusa di Black-Scholes, mentre per la put americana rappresenta un parametro essenziale nei metodi numerici basati su discretizzazione.

Rispetto all'approccio teorico fondato su dati simulati, l'utilizzo di una volatilità empirica consente una calibrazione più realistica del modello, riducendo la distanza tra pricing teorico e osservazioni di mercato.

Il codice utilizzato per il download dei dati, il calcolo dei rendimenti e la stima della volatilità è riportato integralmente in Appendice.

Per quanto riguarda il tasso d'interesse privo di rischio, necessario ai fini del pricing, si assume come riferimento il rendimento dei titoli governativi statunitensi con scadenza coerente con l'orizzonte temporale dell'opzione considerata.

Si supponga che l'analisi iniziale riguarda una call europea con scadenza pari ad un anno; risulta quindi idoneo adottare un tasso risk-free annuale ottenuto dai Treasury statunitensi a dodici mesi, pari al 4,15%.

Come valore iniziale del sottostante viene invece considerato l'ultimo prezzo disponibile all'interno della serie storica scaricata tramite Python (293.32\$). In questo modo il pricing viene effettuato utilizzando informazioni effettivamente osservate sul mercato al momento dell'analisi.

Per completare la definizione dei parametri, occorre inoltre specificare il valore dello strike price dell'opzione considerata. Nel presente lavoro si assume uno strike K inferiore al prezzo corrente del sottostante, in modo da analizzare una call europea detta in-the-money. Tale scelta consente di considerare un'opzione caratterizzata da valore intrinseco positivo già al momento iniziale dell'analisi. Si assume quindi uno strike price pari a 290\$.

La scelta di una call in-the-money assume inoltre particolare rilevanza nell'interpretazione economica dei risultati. In tali condizioni infatti, il prezzo dell'opzione incorpora non soltanto valore temporale, ma anche una componente significativa di valore intrinseco. Ciò consente di evidenziare con maggiore chiarezza le differenze tra i vari approcci teorici e numerici analizzati nei capitoli precedenti, specialmente in termini di sensibilità rispetto alla volatilità e al tempo residuo alla scadenza.

Nella Tabella 5 sono riportate le ultime osservazioni della serie storica utilizzata, comprensive dei rendimenti logaritmici ottenuti mediante calcolatore.

Data	Prezzo di chiusura (\$)	Log-return	Volume
04/05/2026	276.829987	-0.011886	46668400
05/05/2026	284.179993	0.026204	49311700
06/05/2026	287.510010	0.011650	58336100
07/05/2026	287.440002	-0.000244	45224300
08/05/2026	293.320007	0.020250	52631200

Tabella 5: Ultime osservazioni della serie storica Apple elaborate tramite Python.

Una volta determinati i parametri di mercato, si procede all'applicazione dei diversi modelli di pricing già introdotti nei capitoli precedenti, utilizzando il medesimo framework implementativo sviluppato per il caso simulato.

In questo modo sarà possibile confrontare i risultati ottenuti in un contesto empirico reale, evidenziando analogie e differenze rispetto alle applicazioni teoriche precedenti.

Tra i modelli considerati, il modello di Black-Scholes-Merton verrà assunto come benchmark principale dell'analisi, per gli stessi motivi enunciati in precedenza circa la

chiusura della formula. I risultati ottenuti mediante gli altri approcci numerici verranno pertanto confrontati con il prezzo fornito dal modello BSM, al fine di valutarne coerenza, convergenza e capacità di approssimazione.

Si applica quindi il codice allegato in Appendice relativo al pricing della call europea, in questo caso con i parametri ottenuti tramite analisi di mercato. Il sottostante risulta pari a 293,320007\$, la volatilità calcolata con (7.2) è 0,273915. Come enunciato, lo strike price si ipotizza pari a 290\$, l'orizzonte temporale di un anno e il tasso risk-free risulta uguale a 4,15%.

7.2 Analisi critica dei risultati ottenuti

Tabella 6: Confronto dei risultati numerici per la call europea scritta sul titolo Apple.

Metodo	Prezzo (\$)
Black-Scholes-Merton	39.2590
Binomiale (1000 step)	39.2649
Monte Carlo (100000 simulazioni)	39.1545
Differenze finite implicite	39.2156

Come già discusso nella parte teorica della trattazione, il modello di Black-Scholes-Merton viene assunto come benchmark principale dell'analisi. I risultati ottenuti mostrano una forte coerenza tra i diversi approcci numerici, con scostamenti contenuti rispetto al valore teorico di riferimento.

Il modello binomiale converge rapidamente verso il prezzo BSM all'aumentare del numero di passi temporali, confermando empiricamente le proprietà asintotiche del modello di Cox-Ross-Rubinstein.

Analogamente, la simulazione Monte Carlo produce valori compatibili con il benchmark, pur mantenendo le tipiche oscillazioni dovute alla componente stocastica della simulazione.

Anche il metodo alle differenze finite implicite fornisce un'approssimazione molto vicina al valore teorico, evidenziando la stabilità numerica già osservata nella precedente applicazione con dati simulati.

Nel complesso, i risultati ottenuti confermano come i diversi approcci numerici convergano correttamente verso il valore teorico fornito dal modello Black-Scholes-Merton, pur presentando differenti caratteristiche in termini di velocità di convergenza, stabilità e costo computazionale.

N	Prezzo	Errore
10	38.844295	0.414711
25	39.532766	0.273760
50	39.253590	0.005416
100	39.280368	0.021362
250	39.281303	0.022297
500	39.273634	0.014628
1000	39.264920	0.005914

(a) Convergenza binomiale

M	Prezzo	Errore
1000	39.633449	0.374443
5000	39.892051	0.633045
10000	38.715348	0.543658
50000	39.394206	0.135200
100000	39.154449	0.104556

(b) Convergenza Monte Carlo

Tabella 7: Confronto della convergenza dei metodi numerici

Come osservato nella precedente applicazione teorica, il modello binomiale evidenzia una convergenza progressiva verso il benchmark all'aumentare del numero di step temporali, mentre la simulazione Monte Carlo presenta una convergenza più lenta e caratterizzata da oscillazioni statistiche.

Tali evidenze risultano pienamente coerenti con le proprietà teoriche dei modelli analizzati e confermano la robustezza del framework implementativo utilizzato anche in presenza di dati empirici reali.

Per completezza, si riportano inoltre i grafici relativi alla convergenza dei principali metodi numerici (Figura 5).

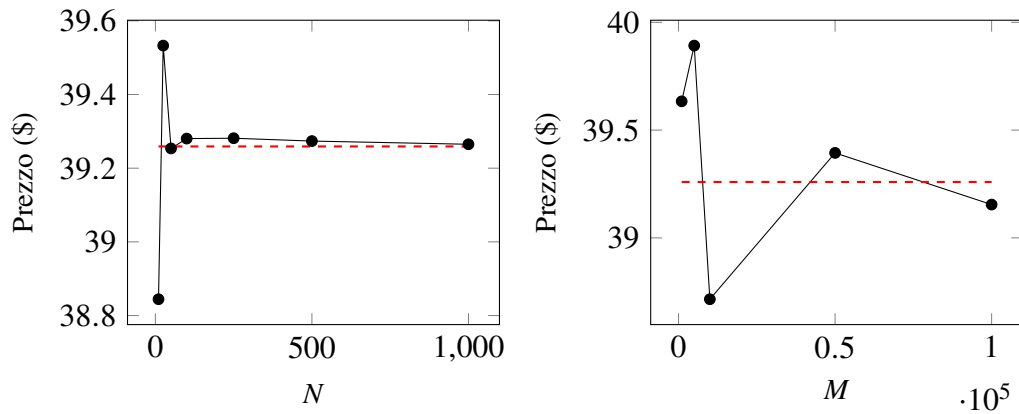


Figura 5: Confronto della convergenza all'aumentare delle iterazioni

I grafici confermano visivamente quanto già osservato a livello numerico. Il modello binomiale presenta una convergenza relativamente stabile verso il benchmark teorico, mentre il metodo Monte Carlo manifesta oscillazioni più marcate dovute alla natura probabilistica della simulazione.

Nonostante ciò, entrambi i metodi risultano coerenti con il valore teorico fornito dal benchmark, confermando l'affidabilità delle implementazioni numeriche adottate.

Un'ulteriore analisi riguarda il confronto tra il prezzo della call europea su Apple ottenuto mediante le implementazioni appena citate e la corrispondente quotazione osservata sul mercato. Tale confronto consente di valutare l'impatto delle diverse metodologie di stima dei parametri e, in particolare, del diverso ruolo assunto dalla volatilità nella determinazione del valore dell'opzione.

Nella sezione precedente il prezzo dell'opzione è stato determinato utilizzando dati di mercato relativi al titolo Apple e stimando i parametri necessari al modello, tra cui il tasso privo di rischio e la volatilità storica del sottostante. Il valore ottenuto riflette pertanto una valutazione fondata sulle informazioni desumibili dall'andamento passato del titolo.

Per effettuare il confronto, sono state invece raccolte le quotazioni effettive delle opzioni Apple tramite la libreria `yfinance`, che consente l'accesso alle option chain pubblicate da Yahoo Finance. In particolare, è stata selezionata una call con strike price pari

a 290\$ e scadenza prossima all'orizzonte temporale considerato nell'analisi precedente. Diversamente dalla valutazione ottenuta mediante la volatilità storica, il prezzo di mercato incorpora le aspettative degli operatori sulla dinamica futura del sottostante, sintetizzate nella volatilità implicita. Il confronto tra i due valori consente quindi di evidenziare eventuali differenze tra le informazioni contenute nei dati storici e quelle espresse dal mercato al momento della negoziazione.

Il confronto tra i diversi approcci è riportato nella Tabella 8.

Tabella 8: Confronto tra pricing teorico e prezzo di mercato della call Apple

Metodo	Prezzo (\$)
Black-Scholes-Merton	39.2590
Binomiale	39.2649
Monte Carlo	39.1545
Differenze finite implicite	39.2156
Prezzo di mercato	49.0000

Il prezzo osservato sul mercato risulta differente rispetto ai valori ottenuti mediante i modelli teorici implementati. Tale scostamento è principalmente riconducibile alla volatilità implicita incorporata nelle quotazioni delle opzioni.

Il modello Black-Scholes-Merton, come anche gli altri implementati, utilizza una volatilità storica stimata sulla base dei rendimenti passati del sottostante; d'altra parte il mercato incorpora aspettative prospettiche sulla volatilità futura del titolo.

Più precisamente, la volatilità implicita può essere definita come quel valore della volatilità che, inserito all'interno della formula di Black-Scholes-Merton, consente di ottenere esattamente il prezzo osservato sul mercato.

Dal punto di vista economico, la volatilità implicita costituisce quindi una misura forward-looking, ossia una stima prospettica della variabilità futura del sottostante formulata dagli operatori di mercato. Essa incorpora non soltanto l'andamento storico del titolo, ma anche aspettative future, condizioni macroeconomiche, eventi societari attesi,

livelli di domanda e offerta e, più in generale, la percezione del rischio da parte degli investitori.

Nel caso analizzato, la volatilità implicita osservata nelle quotazioni delle opzioni Apple risulta superiore rispetto alla volatilità storica. Tale differenza contribuisce a spiegare lo scostamento tra il prezzo teorico ottenuto mediante i modelli implementati ed il prezzo effettivamente osservato sul mercato.

Nell'esempio considerato infatti, la volatilità implicita risulta pari a 0.3321, valore differente rispetto alla volatilità storica stimata pari a 0.273915.

Ciò evidenzia come gli operatori di mercato attribuiscono una dinamica futura del sottostante diversa da quella desumibile esclusivamente dall'analisi storica.

Questo fenomeno rappresenta uno degli aspetti centrali della finanza quantitativa applicata e costituisce uno dei principali limiti strutturali dell'ipotesi di volatilità costante del modello Black-Scholes-Merton.

È inoltre opportuno sottolineare che una parte dello scostamento tra i prezzi teorici e le quotazioni reali può essere attribuito anche alla diversa maturity dell'opzione considerata.

Nel framework sviluppato nel presente elaborato, il pricing è stato effettuato assumendo un orizzonte temporale esattamente pari ad un anno. Tuttavia, le option chain disponibili sul mercato reale non presentano necessariamente scadenze coincidenti con la maturity teorica adottata nel modello.

Infatti, la prima scadenza disponibile sufficientemente vicina all'orizzonte annuale risulta leggermente superiore ad un anno.

Tale differenza temporale, sebbene contenuta, produce un incremento del valore temporale dell'opzione e contribuisce quindi ad aumentare il prezzo osservato sul mercato rispetto al valore teorico ottenuto mediante il modello Black-Scholes-Merton.

7.3 Analisi della Vega nei diversi modelli di pricing

Dopo aver confrontato i prezzi ottenuti mediante i principali modelli di pricing, l'analisi viene estesa allo studio della Vega, ossia della sensibilità del prezzo dell'opzione rispetto a variazioni della volatilità del sottostante.

Nel caso del modello BSM, Vega viene calcolata direttamente tramite formula analitica.

Per i modelli numerici, invece, essa viene approssimata mediante differenze finite centrate:

$$\text{Vega} \approx \frac{V(\sigma + h) - V(\sigma - h)}{2h}, \quad (7.3)$$

con h parametro incrementale sufficientemente piccolo.

Nel dettaglio nel modello Black-Scholes-Merton, la Vega di una call europea è espressa dalla relazione:

$$\text{Vega}_{BSM} = S_0 \sqrt{T} \phi(d_1) \quad (7.4)$$

dove:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma \sqrt{T}} \quad (7.5)$$

e $\phi(d_1)$ rappresenta la densità della distribuzione normale standard.

Nel presente lavoro, Vega verrà calcolata non soltanto mediante la formula analitica del modello BSM (benchmark), ma anche attraverso i principali metodi numerici precedentemente implementati. Ciò consentirà di confrontare la sensibilità del prezzo dell'opzione rispetto alla volatilità nei diversi approcci computazionali e di valutarne la coerenza rispetto al benchmark teorico.

Per effettuare tale analisi ci si avvale nuovamente del linguaggio Python, allegato in Appendice. Si calcola la sensibilità del benchmark tramite formula chiusa (7.4), mentre negli altri approcci si sfrutta (7.3) e infine si stampano i risultati per i vari modelli.

Tabella 9: Confronto della Vega nei diversi modelli di pricing

Metodo	Vega
Black-Scholes-Merton	110.815692
Binomiale	110.880811
Monte Carlo	117.698322
Differenze finite implicite	108.496897

I risultati ottenuti evidenziano una forte coerenza tra i diversi approcci numerici. In particolare, il modello binomiale e il metodo alle differenze finite producono valori della Vega molto vicini a quelli ottenuti tramite la formula analitica di Black-Scholes-Merton.

La simulazione Monte Carlo presenta invece oscillazioni leggermente maggiori, dovute alla natura stocastica del metodo. Tuttavia, all'aumentare del numero di simulazioni, anche la stima della Vega converge progressivamente verso il benchmark teorico.

Dal punto di vista economico, i valori ottenuti confermano l'elevata sensibilità del prezzo della call rispetto alla volatilità del sottostante.

In particolare, è interessante sottolineare la forte positività della Vega; mentre in molti strumenti finanziari un aumento della volatilità può tradursi anche in perdite elevate, in questo caso il diritto (e non l'obbligo) di esercitare limita tale rischio e consente invece di beneficiare dell'incremento dei potenziali guadagni.

L'analisi della Vega permette quindi di completare lo studio delle caratteristiche quantitative dell'opzione considerata, mostrando non soltanto il valore del contratto nei diversi modelli, ma anche la sua reattività rispetto alle variazioni delle condizioni di mercato.

7.4 Calibrazione dei parametri: il caso della put americana

Dopo aver validato l'efficacia dei modelli analitici e numerici sulla call europea applicata al mercato reale, l'analisi si sposta sulla valutazione di una put americana avente come sottostante il medesimo titolo Apple. Tale passaggio permette di estendere le metodologie computazionali precedentemente verificate ad una classe di derivati più complessa,

introducendo la sfida della frontiera libera e dell'esercizio anticipato, finora discussi solo in termini teorici.

Per garantire coerenza statistica con l'analisi della call, sono stati mantenuti i parametri derivanti dallo studio della serie storica: la volatilità (σ) risulta pari a 0,273915, mentre il tasso risk-free è 0,0415.

L'obiettivo è ora testare la capacità degli algoritmi binomiali e alle differenze finite di catturare il premio legato alla natura americana dell'opzione, utilizzando i dati reali di Apple come banco di prova definitivo per la robustezza numerica del software implementato.

Applicando il medesimo codice dell'approccio simulato, si vuole verificare la convergenza e la solidità dei modelli a fronte di parametri empirici, mantenendo il confronto tra l'approccio binomiale, utilizzato come benchmark, e i metodi alle differenze finite.

7.5 Analisi critica dei risultati ottenuti

Per testare tale robustezza, è stata simulata una put con strike $K = 294\$$, fissato in prossimità del prezzo spot corrente $S_0 = 293,32\$$. Tale configurazione è introdotta per mantenere una certa coerenza con il caso della call, in cui si presuppone l'opzione in-the-money.

Tabella 10: Confronto dei risultati numerici per la put americana scritta sul titolo Apple

Metodo	Prezzo (\$)
Binomiale (2000 step)	27.258773
Diff. Finite (Esplicito)	27.239230
Diff. Finite (Implicito)	27.232150

Come emerge dai dati riportati nella Tabella 10, il metodo delle differenze finite implicite mostra una convergenza estremamente stabile. La capacità di mantenere l'accuratezza senza richiedere passi temporali infinitesimali lo rende particolarmente efficiente per analisi di lungo periodo (come l'orizzonte annuale qui considerato). Il fatto che i risultati

siano in linea con il benchmark binomiale conferma che il vincolo di esercizio anticipato è stato correttamente incorporato, riflettendo il principio per cui il detentore non manterrebbe il contratto qualora l'esercizio immediato risultasse più vantaggioso del valore di continuazione.

Un elemento chiave nella caratterizzazione della put su Apple è la frontiera di esercizio anticipato che, come sottolineato in precedenza, rappresenta la curva che separa la regione di continuazione da quella dove è preferibile l'esercizio anticipato.

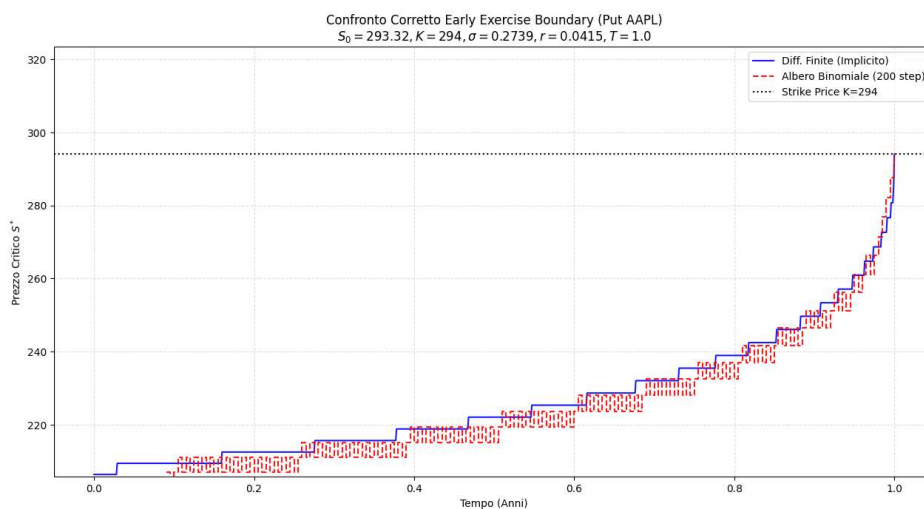


Figura 6

L'estrazione numerica di tale soglia, effettuata impiegando il medesimo apparato algoritmico sviluppato per il caso teorico e adattandone esclusivamente i parametri di input, conferma la coerenza strutturale del modello implementato.

Come evidenziato nella Figura 6, la curva si manifesta come una funzione monotona crescente rispetto al tempo a scadenza. Tale andamento riflette la progressiva erosione del valore temporale dell'opzione: con la contrazione dell'orizzonte temporale residuo, il beneficio legato all'attesa diminuisce drasticamente, innalzando progressivamente il prezzo critico al quale risulta ottimale incassare il valore intrinseco anzichè mantenere la posizione aperta.

La sostanziale sovrapposizione tra la frontiera ottenuta tramite lo schema alle differenze finite e quella derivata dal modello binomiale - pur nella natura discreta di quest'ultimo - valida ulteriormente la robustezza della procedura di calcolo. Questo processo permette quindi di mappare con precisione le decisioni ottimali dell'investitore.

Anche nel caso della put americana, i risultati teorici ottenuti mediante i modelli numerici vengono confrontati con le effettive quotazioni osservate sul mercato, scaricate tramite Python sempre dalla piattaforma Yahoo Finance.

È stata selezionata una put con strike price pari a 290\$, corrispondente al valore di mercato più prossimo a quello impiegato nell'analisi storica (294\$). Al fine di preservare la coerenza del confronto, si è mantenuta invariata una maturity annuale.

Tabella 11: Confronto tra pricing teorico e prezzo di mercato della put americana Apple

Metodo	Prezzo (\$)
Binomiale	27.258773
Differenze finite implicite	27.232150
Prezzo di mercato	19,7000

Anche in questo caso emergono differenze tra il pricing teorico e la quotazione effettivamente osservata sul mercato.

Nel caso delle opzioni americane, tali differenze risultano influenzate non soltanto dalla volatilità implicita, ma anche dalla possibilità di esercizio anticipato e dalla diversa percezione del rischio incorporata dagli operatori di mercato.

La volatilità implicita osservata nella put risulta pari a 0.2353, confermando come il mercato incorpori aspettative dinamiche differenti rispetto alla volatilità storica utilizzata nei modelli implementati.

Nel caso della put americana si osserva però una maggiore aderenza tra i valori teorici stimati ed i prezzi effettivamente quotati sul mercato, così come tra volatilità implicita e volatilità storica.

Tale evidenza può essere ricondotta alla rilevanza del diritto di esercizio anticipato, che viene considerato in maniera simile dagli operatori, riducendo le discrepanze tra modello e mercato.

Inoltre, le put sono spesso più richieste per hedging e questo può rendere il loro pricing maggiormente stabile e coerente. Vengono infatti utilizzate per proteggersi dal rischio di ribasso. Dal punto di vista comportamentale la maggiore sensibilità degli investitori alle perdite rispetto ai guadagni può contribuire ad una domanda più strutturata di protezione contro i ribassi rispetto alla speculazione al rialzo. Questa caratteristica contribuisce a rendere la volatilità implicita più ancorata a dinamiche osservabili e meno distante dalle stime storiche.

Nel complesso, il confronto tra i modelli teorici implementati e le effettive quotazioni di mercato mostra come gli approcci numerici analizzati risultino coerenti dal punto di vista computazionale, ma evidenzia al contempo i limiti strutturali delle ipotesi teoriche sottostanti.

In particolare, il mercato incorpora informazioni prospettiche e componenti di rischio non direttamente catturate dalla volatilità storica utilizzata nei modelli classici.

L'introduzione della volatilità implicita consente quindi di colmare parzialmente il divario tra pricing teorico e osservazioni empiriche, rappresentando uno degli strumenti fondamentali della moderna finanza quantitativa applicata.

7.6 Analisi di Theta nei diversi modelli di pricing

Mentre nell'analisi della call europea l'attenzione è stata posta sulla sensibilità a variazioni della volatilità (Vega), per la put americana risulta maggiormente significativo indagare il legame tra il prezzo del derivato e il trascorrere del tempo, in quanto strettamente connesso alla dinamica della frontiera di esercizio anticipato.

L'analisi della sensibilità del prezzo rispetto al tempo, espressa dal Theta (Θ), assume particolare rilevanza nel caso considerato della put americana. A differenza delle opzioni

europee, il decadimento temporale di un'opzione americana deve tener conto del valore derivante dal diritto di esercizio anticipato, che varia man mano che ci si avvicina alla scadenza.

In questa sezione, il Theta viene calcolato esclusivamente mediante i modelli numerici precedentemente implementati (binomiale e metodo delle differenze finite), in quanto capaci di modellizzare correttamente la condizione di esercizio anticipato.

Per entrambi i modelli, il Theta viene approssimato come segue:

$$\Theta \approx \frac{V(T - \Delta t) - V(T)}{\Delta t} \quad (7.6)$$

dove $\Delta t = 1/365$, e V rappresenta il valore di continuazione.

Tabella 12: Confronto del Theta nei diversi modelli di pricing

Metodo	Theta
Binomiale	-11.042834
Differenze finite (implicite)	-11.042643

I risultati ottenuti (mediante codice Python allegato in Appendice) mostrano un'elevata convergenza tra il modello binomiale e il metodo delle differenze finite. Il valore negativo del Theta conferma che, a parità di altre condizioni, il valore della put americana diminuisce con il passare del tempo. Con il trascorrere del tempo, il diritto "ad aspettare" vale sempre meno.

Dal punto di vista economico, la precisione di questa misura è fondamentale per la gestione del rischio di portafoglio, in quanto permette di quantificare l'erosione del valore temporale in contratti dove l'opzione di esercizio anticipato è in-the-money o prossima ad esserlo. La coerenza tra i due approcci numerici valida la robustezza degli algoritmi utilizzati per la stima delle sensibilità considerate.

Conclusioni

Il percorso d'indagine sviluppato in questo elaborato ha permesso di delineare un quadro organico e dettagliato del pricing delle opzioni, muovendo dalle solide basi teoriche del calcolo stocastico fino alla verifica empirica su dati di mercato reali. Sebbene l'eleganza analitica del modello di Black-Scholes-Merton rappresenti un punto di partenza fondamentale, l'analisi condotta ha dimostrato come la reale comprensione della dinamica dei prezzi derivi necessariamente da una sintesi tra rigore matematico e implementazione numerica.

Il fulcro del lavoro, esposto nella parte applicativa del Capitolo 7, ha evidenziato l'efficacia dei diversi approcci algoritmici attraverso lo studio del titolo Apple. L'implementazione in ambiente Python ha permesso di validare la robustezza dei modelli, mostrando come, per le call europee, il metodo binomiale di Cox-Ross-Rubinstein e il metodo delle differenze finite convergano con precisione verso il benchmark analitico all'aumentare della discretizzazione temporale. In particolare, si è osservato che il modello binomiale, pur nella sua semplicità discreta, offre una convergenza estremamente stabile, mentre la simulazione Monte Carlo, nonostante la versatilità nel gestire payoff complessi, richiede un numero di simulazioni elevato per mitigare le oscillazioni stocastiche intrinseche al metodo.

Un passaggio critico dell'analisi è stato rappresentato dalla calibrazione dei parametri. Il confronto tra i prezzi teorici e le quotazioni effettive di mercato ha messo in luce uno scostamento significativo: per una call Apple con strike a 290\$, il modello teorico basa-

to sulla volatilità storica ha prodotto un valore di circa 39.26\$, a fronte di un prezzo di mercato di 49.00\$. Questa discrepanza sottolinea come la volatilità storica, pur essendo un indicatore fondamentale, non riesca a catturare interamente le aspettative future degli operatori. La volatilità implicita emerge dunque non solo come un parametro di calibrazione, ma come il vero veicolo informativo che riflette il sentiment del mercato e i premi per il rischio non modellati dalle assunzioni standard di Black-Scholes.

L'estensione dell'analisi alle opzioni put americane ha permesso di affrontare il problema della frontiera libera e dell'esercizio anticipato. L'utilizzo del metodo delle differenze finite implicite si è rivelato superiore in termini di stabilità computazionale, consentendo di mappare con precisione il prezzo critico del sottostante. Si è dimostrato come tale frontiera sia una funzione crescente rispetto al tempo a scadenza: l'erosione del valore temporale (misurata dalla greca Theta) innalza progressivamente la soglia oltre la quale diventa ottimale incassare il valore intrinseco dell'opzione anziché mantenere la posizione.

In sintesi, la tesi ha confermato che non esiste un unico modello superiore in assoluto, ma che la scelta dello strumento dipende dal compromesso tra precisione desiderata, complessità del payoff e risorse computazionali disponibili. Se il modello binomiale e le differenze finite si distinguono per la precisione deterministica e la gestione dell'esercizio anticipato, il metodo Monte Carlo rimane insostituibile per prodotti strutturati multidimensionali. L'integrazione di queste metodologie, supportata da una solida infrastruttura informatica, costituisce oggi il requisito imprescindibile per una valutazione consapevole e prudentiale dei derivati finanziari in scenari di mercato sempre più complessi e volatili.

Appendice

1. Cenni sulle equazioni alle derivate parziali

Un'equazione differenziale alle derivate parziali (PDE) è un'equazione che coinvolge una funzione incognita di due o più variabili e alcune delle sue derivate parziali.

Data $k \geq 1$ una costante arbitraria e V un sottoinsieme aperto di $\mathbb{R}^n$, si definisce un'equazione differenziale alle derivate parziali di ordine k l'espressione

$$F(D^k v(x), D^{k-1} v(x), \dots, Dv(x), v(x), x) = 0 \quad (x \in V)$$

con

$$F : \mathbb{R}^{n^k} \times \dots \times \mathbb{R}^n \times \mathbb{R} \times V \rightarrow \mathbb{R}.$$

Applicata al contesto dei derivati, l'equazione di BSM è riconducibile ad una particolare PDE lineare del secondo ordine nota come equazione del calore (Evans 2010, Tychonov & Samarski 1967) .

Formalmente un'equazione differenziale alle derivate parziali del secondo ordine in due variabili reali x e t può essere scritta nella forma generale

$$Av_{xx} + Bv_{xt} + Cv_{tt} + \text{derivate d'ordine inferiore} = 0,$$

dove A, B, C sono coefficienti reali. La classificazione dell'equazione dipende dal segno

del discriminante

$$\Delta = B^2 - 4AC.$$

In particolare, l'equazione si dice:

- ellittica se $\Delta < 0$,
- parabolica se $\Delta = 0$,
- iperbolica se $\Delta > 0$.

Le equazioni paraboliche descrivono fenomeni di diffusione e la più nota è l'equazione del calore:

$$\frac{\partial v}{\partial t} = \alpha \frac{\partial^2 v}{\partial x^2},$$

dove $\alpha > 0$ rappresenta il coefficiente di diffusione.

Per ricavare le formule di call e put si va a risolvere l'equazione differenziale di Black-Scholes-Merton applicando le condizioni iniziali sopra citate; in maniera più rigorosa si va a risolvere l'equazione di BSM trasformandola in **un problema di Cauchy per l'equazione del calore**. Un problema di questo tipo consiste nel determinare una funzione $v(x, t)$ tale che

$$\begin{cases} \frac{\partial v}{\partial t} = \alpha \frac{\partial^2 v}{\partial x^2}, & x \in \mathbb{R}, t > 0, \\ v(x, 0) = g(x), \end{cases}$$

dove la funzione $g(x)$ è assegnata e rappresenta la condizione iniziale. La soluzione di questo problema descrive l'evoluzione temporale del profilo iniziale $g(x)$ sotto l'effetto della diffusione.

Nel capitolo 4, si è arrivati alla generica equazione differenziale di BSM a cui applicare le condizioni iniziali. Ricordiamo la formula (3.3)

$$\frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf$$

Spostando rf a primo membro si ottiene un'equazione lineare del secondo ordine di tipo parabolico, definita per $S > 0$ e $t \in [0, T]$.

Per un'opzione europea, il valore del derivato è noto alla data di scadenza T . In particolare, per una call europea vale

$$c(S, T) = \max(S_T - K, 0),$$

mentre per una put europea

$$p(S, T) = \max(K - S_T, 0).$$

La determinazione del prezzo dell'opzione per $t < T$ consiste quindi nella risoluzione dell'equazione di Black-Scholes-Merton con una condizione assegnata al tempo finale. Si tratta pertanto di un problema di Cauchy posto all'indietro nel tempo.

Per ricondurre il problema a una formulazione standard sulle condizioni al tempo iniziale, si effettua la procedura seguente che analizzeremo a partire dal caso di una opzione call.

Consideriamo quindi l'equazione di BSM per una call:

$$\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 c}{\partial S^2} + rS \frac{\partial c}{\partial S} - rc = 0. (*)$$

Introduciamo due nuove variabili, u e z , così definite

$$u = \frac{2}{\sigma^2} \left(r - \frac{\sigma^2}{2} \right) \left[\log \frac{S}{K} - \left(r - \frac{\sigma^2}{2} \right) (t - T) \right],$$

$$z = -\frac{2}{\sigma^2} \left(r - \frac{\sigma^2}{2} \right)^2 (t - T).$$

Si assume $r \neq \frac{\sigma^2}{2}$.

Si effettua successivamente un cambiamento di variabili introducendo $Y(u, z)$ tale che $c(S, t) = e^{r(t-T)} Y(u, z)$.

Si applica ora il teorema delle funzioni composte per calcolare le varie derivate della call rispetto a S e t . Partiamo dalla derivata parziale di c rispetto a S :

$$\begin{aligned}\frac{\partial u}{\partial S} &= \frac{2}{\sigma^2} \left(r - \frac{\sigma^2}{2} \right) \frac{1}{S} = \\ &= \left(\frac{2r}{\sigma^2} - 1 \right) \frac{1}{S} \\ \frac{\partial u}{\partial t} &= \frac{2}{\sigma^2} \left(r - \frac{\sigma^2}{2} \right) \left[- \left(r - \frac{\sigma^2}{2} \right) \right] = \\ &= - \frac{2}{\sigma^2} \left(r - \frac{\sigma^2}{2} \right)^2 \\ \frac{\partial z}{\partial S} &= 0 \\ \frac{\partial z}{\partial t} &= - \frac{2}{\sigma^2} \left(r - \frac{\sigma^2}{2} \right)^2\end{aligned}$$

Si noti che le derivate temporale di u e z coincidono.

$$\frac{\partial c}{\partial S} = e^{r(t-T)} \left(Y_u \frac{\partial u}{\partial S} + Y_z \frac{\partial z}{\partial S} \right).$$

Poiché $\partial z / \partial S = 0$, segue

$$\frac{\partial c}{\partial S} = e^{r(t-T)} Y_u \left(\frac{2r}{\sigma^2} - 1 \right) \frac{1}{S}.$$

Deriviamo nuovamente rispetto a S

$$\frac{\partial^2 c}{\partial S^2} = e^{r(t-T)} \frac{\partial}{\partial S} \left[Y_u \left(\frac{2r}{\sigma^2} - 1 \right) \frac{1}{S} \right].$$

Applicando la regola di derivazione del prodotto e la regola della catena:

$$\frac{\partial^2 c}{\partial S^2} = e^{r(t-T)} \left[Y_{uu} \left(\frac{2r}{\sigma^2} - 1 \right)^2 \frac{1}{S^2} - Y_u \left(\frac{2r}{\sigma^2} - 1 \right) \frac{1}{S^2} \right].$$

Derivando poi c rispetto a t otteniamo

$$\frac{\partial c}{\partial t} = r e^{r(t-T)} Y + e^{r(t-T)} \left(Y_u \frac{\partial u}{\partial t} + Y_z \frac{\partial z}{\partial t} \right).$$

Sostituendo le derivate di u e z precedentemente calcolate si ha

$$\frac{\partial c}{\partial t} = r e^{r(t-T)} Y - e^{r(t-T)} \frac{2}{\sigma^2} \left(r - \frac{\sigma^2}{2} \right)^2 (Y_u + Y_z).$$

Sostituendo le espressioni ottenute in (*), si verifica che essa si riduce all'equazione

$$Y_z = Y_{uu},$$

che coincide con l'equazione del calore omogenea in due variabili. Si noti inoltre che quella che era la condizione finale è diventata iniziale, così da permettere la risoluzione come problema di Cauchy. Infatti

$$c(S_T, T) = Y(u_T, 0) = e^{r(T-T)} Y\left(\left(\frac{2r}{\sigma^2} - 1\right) \log \frac{S_T}{K}, 0\right) = \max(S_T - K, 0).$$

Si ha uno specifico teorema che definisce la soluzione del problema di Cauchy come segue:

$$v(x, t) \in C^{2,1}(\mathbb{R} \times (0, \infty)) \cap C(\mathbb{R} \times [0, \infty)) \text{ t.c}$$

$$v(x, t) = \int_{\mathbb{R}} K_t(x, y) g(y) dy$$

$$\forall (x, t) \in \mathbb{R} \times [0, \infty), v(x, 0) = g(x) \text{ e } K_t(x, y) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{(x-y)^2}{4t}}$$

Calcolando $v(x, t)$ tramite l'integrale appena citato e tornando alle variabili originali,

si ottiene l'equazione di Black-Scholes-Merton per la call europea

$$c(S, t) = e^{r(t-T)} Y(u(S, t), z(t)) = SN(d_1) - Ke^{-r(T-t)} N(d_2),$$

dove

$$d_{1,2} = \frac{\ln(S/K) + (r \pm \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}.$$

2. Codice Sorgente Python

In questa Appendice viene riportato lo script integrale utilizzato per la valutazione dell'opzione call tramite i modelli di Black-Scholes, binomiale, Monte Carlo e differenze finite (Capitolo 6).

```
1 import numpy as np
2 from scipy.stats import norm
3
4 # Parametri di input
5 S0, K, T, r, sigma = 42, 40, 0.5, 0.10, 0.20
6
7 # 1. MODELLO BLACK-SCHOLES-MERTON
8 def black_scholes(S, K, T, r, sigma):
9     d1 = (np.log(S / K) + (r + 0.5 * sigma**2) * T) / (sigma
10         * np.sqrt(T))
11     d2 = d1 - sigma * np.sqrt(T)
12     return S * norm.cdf(d1) - K * np.exp(-r * T) *
13         norm.cdf(d2)
14
15 # 2. MODELLO BINOMIALE (CRR)
16 def binomial_tree(S, K, T, r, sigma, steps=1000):
17     dt = T / steps
```

```

16     u = np.exp(sigma * np.sqrt(dt))
17     d = 1 / u
18     q = (np.exp(r * dt) - d) / (u - d)
19     prices = S * (u ** np.arange(steps, -1, -1)) * (d **
20         np.arange(0, steps + 1, 1))
21     values = np.maximum(prices - K, 0)
22     for i in range(steps - 1, -1, -1):
23         values = np.exp(-r * dt) * (q * values[:-1] + (1 - q)
24             * values[1:])
25     return values[0]
26
27 # 3. SIMULAZIONE MONTE CARLO
28 def monte_carlo(S, K, T, r, sigma, iterations=100000):
29     z = np.random.standard_normal(iterations)
30     ST = S * np.exp((r - 0.5 * sigma**2) * T + sigma *
31         np.sqrt(T) * z)
32     payoffs = np.maximum(ST - K, 0)
33     return np.exp(-r * T) * np.mean(payoffs)
34
35 # 4. DIFFERENZE FINITE (Z = ln(S))
36 def finite_differences(S0, K, T, r, sigma, q=0, M=100,
37     N=1000, method='implicit'):
38     # Parametri di griglia
39     S_max = S0 * 3
40     Z_max, Z_min = np.log(S_max), np.log(S0 / 3)
41     dz = (Z_max - Z_min) / M
42     dt = T / N
43     z_grid = np.linspace(Z_min, Z_max, M + 1)

```

```

41 # Condizione finale (payoff call)
42 v = np.maximum(np.exp(z_grid) - K, 0)
43
44 # Termine comune ai due metodi
45 drift = r - q - 0.5 * sigma**2
46
47 # Metodo implicito
48 if method == 'implicit':
49     alpha = (dt / (2 * dz)) * drift - (dt / (2 * dz**2))
50             * sigma**2
51     beta = 1 + (dt / dz**2) * sigma**2 + r * dt
52     gamma = -(dt / (2 * dz)) * drift - (dt / (2 * dz**2))
53             * sigma**2
54
55 # Costruzione Matrice
56 A = np.zeros((M - 1, M - 1))
57 for i in range(M - 1):
58     A[i, i] = beta
59     if i > 0: A[i, i - 1] = alpha
60     if i < M - 2: A[i, i + 1] = gamma
61
62 # Iterazione temporale all'indietro
63 for j in range(N):
64     rhs = v[1:-1].copy()
65     v[1:-1] = np.linalg.solve(A, rhs)
66
67 else: # Metodo esplicito
68     inv_disc = 1 / (1 + r * dt)
69     alpha_s = inv_disc * (-(dt / (2 * dz)) * drift + (dt

```

```

        / (2 * dz**2)) * sigma**2)
68     beta_s  = inv_disc * (1 - (dt / dz**2) * sigma**2)
69     gamma_s = inv_disc * ((dt / (2 * dz)) * drift + (dt /
        (2 * dz**2)) * sigma**2)

70
71     for j in range(N):
72         v_next = np.zeros(M + 1)
73         for i in range(1, M):
74             v_next[i] = alpha_s * v[i-1] + beta_s * v[i]
75                 + gamma_s * v[i+1]
76
77             # Condizioni al contorno (semplificate)
78             v_next[0] = 0
79             v_next[M] = np.exp(Z_max) - K * np.exp(-r * (j+1)
80                 * dt)
81             v = v_next
82
83     return np.interp(np.log(S0), z_grid, v)
84
85 # Esecuzione e Stampa
86 print(f"BSM: {black_scholes(S0, K, T, r, sigma):.4f}")
87 print(f"Binomiale: {binomial_tree(S0, K, T, r, sigma):.4f}")
88 print(f"Monte Carlo: {monte_carlo(S0, K, T, r, sigma):.4f}")
89 print(f"Diff. Finite (implicit): {finite_differences(S0, K,
90     T, r, sigma, method='implicit'):.4f}")
91
92 #Studio della convergenza
93 # =====
94 # VALORE DI RIFERIMENTO (BSM)
95 # =====

```

```

92 bsm_value = black_scholes(S0, K, T, r, sigma)
93 print(f"\nValore Black-Scholes-Merton: {bsm_value:.6f}")
94
95 # =====
96 # CONVERGENZA BINOMIALE
97 # =====
98 steps_list = [10, 25, 50, 100, 250, 500, 1000]
99
100 print("\n--- Convergenza Binomiale ---")
101 print("N\tPrezzo\t\tErrore")
102
103 binomial_results = []
104
105 for N in steps_list:
106     price = binomial_tree(S0, K, T, r, sigma, steps=N)
107     error = abs(price - bsm_value)
108     binomial_results.append((N, price))
109     print(f"{N}\t{price:.6f}\t{error:.6f}")
110
111 # =====
112 # CONVERGENZA MONTE CARLO
113 # =====
114 sim_list = [1000, 5000, 10000, 50000, 100000]
115
116 print("\n--- Convergenza Monte Carlo ---")
117 print("Sim\tPrezzo\t\tErrore")
118
119 mc_results = []
120

```

```

121 for M in sim_list:
122     price = monte_carlo(S0, K, T, r, sigma, iterations=M)
123     error = abs(price - bsm_value)
124     mc_results.append((M, price))
125     print(f"{M}\t{price:.6f}\t{error:.6f}")

```

Di seguito il codice relativo al download della serie storica del titolo Apple trattato nel capitolo 7, con calcolo del rendimento e volatilità.

```

1 import yfinance as yf
2 import pandas as pd
3 import numpy as np
4
5 # Scarica 5 anni di dati
6 df = yf.download("AAPL", period="5y")
7
8 # Calcolo log returns
9 df["log_return"] = np.log(df["Close"] / df["Close"].shift(1))
10
11 # Rimuove NaN
12 df = df.dropna()
13
14 # Volatilità annualizzata
15 vol = df["log_return"].std() * np.sqrt(252)
16
17 print(df.tail())
18 print("\nVolatilità annualizzata:", vol)

```

Si illustra il codice relativo al pricing della call europea scritta sul titolo Apple, con valutazione aggiuntiva della Vega.

```

1 import numpy as np
2 from scipy.stats import norm
3 import pandas as pd
4
5 # =====
6 # PARAMETRI DI INPUT
7 # =====
8
9 S0 = 293.320007
10 K = 290
11 T = 1
12 r = 0.0415
13 sigma = 0.273915
14
15 h = 0.01 # perturbazione volatilita'
16
17 # =====
18 # 1. BLACK-SCHOLES-MERTON
19 # =====
20
21 def black_scholes(S, K, T, r, sigma):
22
23     d1 = (
24         np.log(S / K) +
25         (r + 0.5 * sigma**2) * T
26     ) / (sigma * np.sqrt(T))
27
28     d2 = d1 - sigma * np.sqrt(T)

```

```

29
30     return (
31         S * norm.cdf(d1)
32         - K * np.exp(-r * T) * norm.cdf(d2)
33     )
34
35
36 def vega_bsm(S, K, T, r, sigma):
37
38     d1 = (
39         np.log(S / K) +
40         (r + 0.5 * sigma**2) * T
41     ) / (sigma * np.sqrt(T))
42
43     return S * norm.pdf(d1) * np.sqrt(T)
44
45
46 # =====
47 # 2. MODELLO BINOMIALE
48 # =====
49
50 def binomial_tree(S, K, T, r, sigma, steps=1000):
51
52     dt = T / steps
53
54     u = np.exp(sigma * np.sqrt(dt))
55     d = 1 / u
56
57     q = (np.exp(r * dt) - d) / (u - d)

```



```

58
59     prices = (
60         S *
61         (u ** np.arange(steps, -1, -1)) *
62         (d ** np.arange(0, steps + 1))
63     )
64
65     values = np.maximum(prices - K, 0)
66
67     for i in range(steps - 1, -1, -1):
68
69         values = np.exp(-r * dt) * (
70             q * values[: -1] +
71             (1 - q) * values[1:]
72         )
73
74     return values[0]
75
76
77 def vega_binomial(S, K, T, r, sigma, h=0.01):
78
79     up = binomial_tree(S, K, T, r, sigma + h)
80     down = binomial_tree(S, K, T, r, sigma - h)
81
82     return (up - down) / (2 * h)
83
84
85 # =====
86 # 3. MONTE CARLO

```

```

87 # =====
88
89 def monte_carlo(S, K, T, r, sigma, iterations=100000):
90
91     z = np.random.standard_normal(iterations)
92
93     ST = S * np.exp(
94         (r - 0.5 * sigma**2) * T +
95         sigma * np.sqrt(T) * z
96     )
97
98     payoffs = np.maximum(ST - K, 0)
99
100     return np.exp(-r * T) * np.mean(payoffs)
101
102
103 def vega_monte_carlo(S, K, T, r, sigma, h=0.01):
104
105     up = monte_carlo(S, K, T, r, sigma + h)
106     down = monte_carlo(S, K, T, r, sigma - h)
107
108     return (up - down) / (2 * h)
109
110
111 # =====
112 # 4. DIFFERENZE FINITE
113 # =====
114
115 def finite_differences(

```

```

116     S0,
117     K,
118     T,
119     r,
120     sigma,
121     q=0,
122     M=100,
123     N=1000
124 ):
125
126     S_max = S0 * 3
127
128     Z_max = np.log(S_max)
129     Z_min = np.log(S0 / 3)
130
131     dz = (Z_max - Z_min) / M
132     dt = T / N
133
134     z_grid = np.linspace(Z_min, Z_max, M + 1)
135
136     v = np.maximum(np.exp(z_grid) - K, 0)
137
138     drift = r - q - 0.5 * sigma**2
139
140     alpha = (
141         (dt / (2 * dz)) * drift
142         - (dt / (2 * dz**2)) * sigma**2
143     )
144

```

```

145     beta = (
146         1
147         + (dt / dz**2) * sigma**2
148         + r * dt
149     )
150
151     gamma = (
152         -(dt / (2 * dz)) * drift
153         - (dt / (2 * dz**2)) * sigma**2
154     )
155
156     A = np.zeros((M - 1, M - 1))
157
158     for i in range(M - 1):
159
160         A[i, i] = beta
161
162         if i > 0:
163             A[i, i - 1] = alpha
164
165         if i < M - 2:
166             A[i, i + 1] = gamma
167
168     for j in range(N):
169
170         rhs = v[1:-1].copy()
171
172         v[1:-1] = np.linalg.solve(A, rhs)
173

```

```

174     return np.interp(np.log(S0), z_grid, v)
175
176
177 def vega_fd(S0, K, T, r, sigma, h=0.01):
178
179     up = finite_differences(S0, K, T, r, sigma + h)
180     down = finite_differences(S0, K, T, r, sigma - h)
181
182     return (up - down) / (2 * h)
183
184
185 # =====
186 # CALCOLO VEGA
187 # =====
188
189 vega_results = {
190     "Black-Scholes": vega_bsm(S0, K, T, r, sigma),
191     "Binomiale": vega_binomial(S0, K, T, r, sigma),
192     "Monte_Carlo": vega_monte_carlo(S0, K, T, r, sigma),
193     "Diff_Finite": vega_fd(S0, K, T, r, sigma)
194 }
195
196 # Creazione DataFrame
197
198 df_vega = pd.DataFrame(
199     list(vega_results.items()),
200     columns=["Metodo", "Vega"]
201 )
202

```

```
203 print(df_vega)
```

Si allega il codice relativo al download delle options chain call scritte sul titolo Apple.

```
1 import yfinance as yf
2 import pandas as pd
3
4 # =====
5 # PARAMETRI
6 # =====
7
8 ticker_symbol = "AAPL"
9 target_strike = 290
10 expiry = "2027-06-17"
11
12 # =====
13 # DOWNLOAD DATI
14 # =====
15
16 ticker = yf.Ticker(ticker_symbol)
17
18 # Download option chain
19 chain = ticker.option_chain(expiry)
20
21 # Call options
22 calls = chain.calls
23
24 # Selezione strike piu' vicino a 290
25 selected_call = calls.iloc[
26     (calls['strike'] - target_strike).abs().argsort()[:1]
```

```

27 ]
28
29 print("\nCALL_SELEZIONATA")
30 print(selected_call[[
31     'contractSymbol',
32     'strike',
33     'lastPrice',
34     'bid',
35     'ask',
36     'impliedVolatility',
37     'volume',
38     'openInterest'
39 ]])
40
41 # =====
42 # ESTRAZIONE VALORI
43 # =====
44
45 market_price_call = selected_call['lastPrice'].values[0]
46 iv_call = selected_call['impliedVolatility'].values[0]
47
48 print("\n===== ")
49 print(f"Prezzo_mercato_CALL: {market_price_call:.4f} ")
50 print(f"Volatilita'_implicita_CALL: {iv_call:.4f}")
51 print("===== ")

```

Si mostra poi il codice legato al prezzaggio della put americana (approccio teorico e simulato - Capitolo 6).

```

1 import numpy as np

```

```

2 import matplotlib.pyplot as plt
3
4 # 1. PARAMETRI DI INPUT
5 S0 = 45          # Prezzo attuale
6 K = 50           # Strike Price
7 T = 0.75         # Scadenza (9 mesi)
8 r = 0.03         # Tasso risk-free
9 sigma = 0.25     # Volatilita'
10
11 # 2. MODELLO BINOMIALE (CRR) - BENCHMARK PER AMERICANE
12 def binomial_american_put(S, K, T, r, sigma, steps=2000):
13     dt = T / steps
14     u = np.exp(sigma * np.sqrt(dt))
15     d = 1 / u
16     q = (np.exp(r * dt) - d) / (u - d)
17
18     # Prezzi al terminale
19     prices = S * (u ** np.arange(steps, -1, -1)) * (d **
20         np.arange(0, steps + 1, 1))
21
22     # Payoff al terminale
23     values = np.maximum(K - prices, 0)
24
25     # Indietro nell'albero
26     for i in range(steps - 1, -1, -1):
27         values = np.exp(-r * dt) * (q * values[:-1] + (1 - q)
28             * values[1:])
29         prices = S * (u ** np.arange(i, -1, -1)) * (d **
30             np.arange(0, i + 1, 1))
31
32     # Vincolo esercizio anticipato

```



```

28         values = np.maximum(values, K - prices)
29
30     return values[0]
31
32 # 3. DIFFERENZE FINITE (LOG-PREZZO Z)
33 def finite_differences_american(S0, K, T, r, sigma, M=100,
34     N=2000, method='implicit'):
35     # Griglia sul log-prezzo Z
36     S_max, S_min = S0 * 3, S0 / 3
37     Z_max, Z_min = np.log(S_max), np.log(S_min)
38     dz = (Z_max - Z_min) / M
39     dt = T / N
40     z_grid = np.linspace(Z_min, Z_max, M + 1)
41     s_grid = np.exp(z_grid)
42
43     # Condizione finale (Payoff Put)
44     v = np.maximum(K - s_grid, 0)
45     drift = r - 0.5 * sigma**2
46
47     if method == 'implicit':
48         # Coefficienti matrice implicita
49         alpha = (dt / (2 * dz)) * drift - (dt / (2 * dz**2))
50             * sigma**2
51         beta = 1 + (dt / dz**2) * sigma**2 + r * dt
52         gamma = -(dt / (2 * dz)) * drift - (dt / (2 * dz**2))
53             * sigma**2
54
55         A = np.zeros((M - 1, M - 1))
56         for i in range(M - 1):

```

```

54         A[i, i] = beta
55         if i > 0: A[i, i - 1] = alpha
56         if i < M - 2: A[i, i + 1] = gamma
57
58     for j in range(N):
59         rhs = v[1:-1].copy()
60         v[1:-1] = np.linalg.solve(A, rhs)
61
62         # Condizioni al contorno e vincolo americano
63         v[0] = K - np.exp(Z_min)
64         v[-1] = 0
65         v = np.maximum(v, K - s_grid)
66
67     else: # Metodo Esplicito
68         inv_disc = 1 / (1 + r * dt)
69         alpha_s = inv_disc * (-(dt / (2 * dz)) * drift + (dt
70                               / (2 * dz**2)) * sigma**2)
71         beta_s = inv_disc * (1 - (dt / dz**2) * sigma**2)
72         gamma_s = inv_disc * ((dt / (2 * dz)) * drift + (dt /
73                               (2 * dz**2)) * sigma**2)
74
75     for j in range(N):
76         v_next = np.zeros(M + 1)
77         v_next[1:-1] = alpha_s * v[:-2] + beta_s *
78             v[1:-1] + gamma_s * v[2:]
79         v_next[0], v_next[M] = K - np.exp(Z_min), 0
80         v = np.maximum(v_next, K - s_grid)
81
82     return np.interp(np.log(S0), z_grid, v)

```

```

80
81 # --- ESECUZIONE ---
82 res_bin = binomial_american_put(S0, K, T, r, sigma)
83 res_imp = finite_differences_american(S0, K, T, r, sigma,
      method='implicit')
84 res_exp = finite_differences_american(S0, K, T, r, sigma,
      method='explicit')
85
86 print(f"---_Risultati_Pricing_Put_Americana_---")
87 print(f"Albero_Binomiale:{res_bin:.4f}")
88 print(f"Diff._Finite_(Implicito):{res_imp:.4f}")
89 print(f"Diff._Finite_(Esplicito):{res_exp:.4f}")
90
91 #Studio della convergenza
92 # =====
93 # VALORE DI RIFERIMENTO (BENCHMARK BINOMIALE)
94 # =====
95 benchmark = binomial_american_put(S0, K, T, r, sigma,
      steps=2000)
96 print(f"\nValore_benchmark_(Binomiale_2000_step):_
      {benchmark:.6f}")
97
98 # =====
99 # CONVERGENZA BINOMIALE
100 # =====
101 steps_list = [50, 100, 250, 500, 1000, 2000]
102
103 print(f"\n---_Convergenza_Binomiale_---")
104 print(f"N\tPrezzo\t\t\tErrore")

```

```

105
106 binomial_results = []
107
108 for N in steps_list:
109     price = binomial_american_put(S0, K, T, r, sigma, steps=N)
110     error = abs(price - benchmark)
111     binomial_results.append((N, price))
112     print(f"{N}\t{price:.6f}\t{error:.6f}")
113
114 # =====
115 # CONVERGENZA DIFFERENZE FINITE IMPLICITE
116 # =====
117 grid_list = [50, 100, 200, 400, 800]
118
119 print("\n--- Convergenza Diff. Finite (Implicit) ---")
120 print("Grid\tPrezzo\t\tErrore")
121
122 fd_results = []
123
124 for N in grid_list:
125     price = finite_differences_american(S0, K, T, r, sigma,
126                                         M=N, N=2000, method='implicit')
127     error = abs(price - benchmark)
128     fd_results.append((N, price))
129     print(f"{N}\t{price:.6f}\t{error:.6f}")

```

Segue codice relativo al download delle options chain relativo al caso della put americana.

Le opzioni scaricate tramite Yahoo Finance fanno riferimento alle equity options stan-

dardizzate negoziate sul mercato statunitense. Tali contratti, nel caso di opzioni scritte su singole azioni come Apple, risultano generalmente di tipo americano, ossia esercitabili anticipatamente in qualunque istante precedente la scadenza.

Di conseguenza, le quotazioni considerate risultano coerenti con il framework teorico della put americana sviluppato nel presente lavoro. Va sottolineato che nel caso della call non ci si è posti la problematica in quanto, in linea generale, non vengono mai esercitate anticipatamente.

```
1 import yfinance as yf
2 import pandas as pd
3
4 # =====
5 # PARAMETRI
6 # =====
7
8 ticker_symbol = "AAPL"
9 target_strike = 294
10 expiry = "2027-06-17"
11
12 # =====
13 # DOWNLOAD DATI
14 # =====
15
16 ticker = yf.Ticker(ticker_symbol)
17
18 # Download option chain
19 chain = ticker.option_chain(expiry)
20
21 # Put options
```

```

22 puts = chain.puts
23
24 # Selezione strike piu' vicino a 294
25 selected_put = puts.iloc[
26     (puts['strike'] - target_strike).abs().argsort()[0]
27 ]
28
29 print("\nPUT SELEZIONATA")
30 print(selected_put[[
31     'contractSymbol',
32     'strike',
33     'lastPrice',
34     'bid',
35     'ask',
36     'impliedVolatility',
37     'volume',
38     'openInterest'
39 ]])
40
41 # =====
42 # ESTRAZIONE VALORI
43 # =====
44
45 market_price_put = selected_put['lastPrice'].values[0]
46 iv_put = selected_put['impliedVolatility'].values[0]
47
48 print("\n===== ")
49 print(f"Prezzo mercato PUT: {market_price_put:.4f}$")
50 print(f"Volatilita' implicita PUT: {iv_put:.4f}")

```

```
51 print("=====")
```

In allegato il codice relativo al pricing della put europea.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.stats import norm
4
5 # =====
6 # 1. PARAMETRI DI INPUT
7 # =====
8
9 S0 = 45
10 K = 50
11 T = 0.75
12
13 r = 0.03
14 sigma = 0.25
15
16 # =====
17 # BLACK-SCHOLES PUT EUROPEA
18 # =====
19
20 def black_scholes_put(S, K, T, r, sigma):
21
22     d1 = (
23         np.log(S / K)
24         + (r + 0.5 * sigma**2) * T
25     ) / (sigma * np.sqrt(T))
26
```

```

27     d2 = d1 - sigma * np.sqrt(T)
28
29     return (
30         K * np.exp(-r * T) * norm.cdf(-d2)
31         - S * norm.cdf(-d1)
32     )
33
34     # =====
35     # MODELLO BINOMIALE (CRR)
36     # =====
37
38 def binomial_european_put(
39     S, K, T, r, sigma,
40     steps=2000
41 ):
42
43     dt = T / steps
44
45     u = np.exp(sigma * np.sqrt(dt))
46     d = 1 / u
47
48     q = (
49         np.exp(r * dt) - d
50     ) / (u - d)
51
52     prices = (
53         S
54         * (u ** np.arange(steps, -1, -1))
55         * (d ** np.arange(0, steps + 1))

```



```

56     )
57
58     values = np.maximum(K - prices, 0)
59
60     for _ in range(steps):
61
62         values = np.exp(-r * dt) * (
63             q * values[:-1]
64             + (1 - q) * values[1:]
65         )
66
67     return values[0]
68
69     # =====
70     # DIFFERENZE FINITE
71     # =====
72
73     def finite_differences_european(
74         S0, K, T, r, sigma,
75         M=100, N=2000,
76         method='implicit'
77     ):
78
79         S_max = S0 * 3
80         S_min = S0 / 3
81
82         Z_max = np.log(S_max)
83         Z_min = np.log(S_min)
84

```

```

85     dz = (Z_max - Z_min) / M
86     dt = T / N
87
88     z_grid = np.linspace(
89         Z_min,
90         Z_max,
91         M + 1
92     )
93
94     s_grid = np.exp(z_grid)
95
96     v = np.maximum(K - s_grid, 0)
97
98     drift = r - 0.5 * sigma**2
99
100     # =====
101     # IMPLICITO
102     # =====
103
104     if method == "implicit":
105
106         alpha = (
107             (dt/(2*dz))*drift
108             - (dt/(2*dz**2))*sigma**2
109         )
110
111         beta = (
112             1
113             + (dt/dz**2)*sigma**2

```

```

114         + r*dt
115     )
116
117     gamma = (
118         -(dt/(2*dz))*drift
119         - (dt/(2*dz**2))*sigma**2
120     )
121
122     A = np.zeros((M-1,M-1))
123
124     for i in range(M-1):
125
126         A[i,i] = beta
127
128         if i > 0:
129             A[i,i-1] = alpha
130
131         if i < M-2:
132             A[i,i+1] = gamma
133
134     for _ in range(N):
135
136         rhs = v[1:-1].copy()
137
138         v[1:-1] = np.linalg.solve(
139             A,
140             rhs
141         )
142

```

```

143         v[0] = (
144             K*np.exp(-r*(T))
145             - np.exp(Z_min)
146         )
147
148         v[-1] = 0
149
150         # =====
151         # ESPLICITO
152         # =====
153
154     else:
155
156         inv_disc = 1/(1+r*dt)
157
158         alpha_s = inv_disc * (
159             -(dt/(2*dz))*drift
160             + (dt/(2*dz**2))*sigma**2
161         )
162
163         beta_s = inv_disc * (
164             1
165             - (dt/dz**2)*sigma**2
166         )
167
168         gamma_s = inv_disc * (
169             (dt/(2*dz))*drift
170             + (dt/(2*dz**2))*sigma**2
171         )

```

```

172
173     for _ in range(N):
174
175         v_next = np.zeros(M+1)
176
177         v_next[1:-1] = (
178             alpha_s*v[:-2]
179             + beta_s*v[1:-1]
180             + gamma_s*v[2:]
181         )
182
183         v_next[0] = (
184             K*np.exp(-r*T)
185             - np.exp(Z_min)
186         )
187
188         v_next[-1] = 0
189
190         v = v_next
191
192     return np.interp(
193         np.log(S0),
194         z_grid,
195         v
196     )
197
198     # =====
199     # ESECUZIONE
200     # =====

```

```

201
202 bs_price = black_scholes_put(
203     S0, K, T, r, sigma
204 )
205
206 res_bin = binomial_european_put(
207     S0, K, T, r, sigma
208 )
209
210 res_imp = finite_differences_european(
211     S0, K, T, r, sigma,
212     method='implicit'
213 )
214
215 res_exp = finite_differences_european(
216     S0, K, T, r, sigma,
217     method='explicit'
218 )
219
220 print("\n---PricingPutEuropea---")
221
222 print(f"Black-Scholes: {bs_price:.4f}")
223 print(f"Binomiale: {res_bin:.4f}")
224 print(f"Diff. Finite Implicito: {res_imp:.4f}")
225 print(f"Diff. Finite Esplicito: {res_exp:.4f}")
226
227 benchmark = bs_price

```

3. Frontiera d'esercizio anticipato

La frontiera di esercizio anticipato $S^*(t)$ rappresenta il livello critico del prezzo del sottostante al di sotto del quale risulta ottimale esercitare anticipatamente un'opzione put americana.

Dal punto di vista teorico, essa emerge come soluzione di un problema a frontiera libera associato all'equazione di Black-Scholes, in cui la regione di esercizio e la regione di continuazione sono determinate in modo endogeno.

Nel presente lavoro, la curva $S^*(t)$ è stata stimata numericamente sia tramite il modello binomiale sia attraverso il metodo alle differenze finite implicito, evidenziando una forte coerenza tra i due approcci.

La forma crescente della frontiera rispetto al tempo alla scadenza riflette la progressiva riduzione del valore temporale dell'opzione: avvicinandosi alla scadenza, diventa ottimale esercitare per un intervallo sempre più ampio di valori del sottostante.

Tale comportamento è coerente con l'intuizione economica del problema e conferma la corretta implementazione numerica dei modelli utilizzati.

Di seguito il codice per visualizzare la frontiera d'esercizio anticipato:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # 1. PARAMETRI DI INPUT
5 S0 = 45
6 K = 50
7 T = 0.75
8 r = 0.03
9 sigma = 0.25
10
```

```

11 def get_corrected_boundaries(S0, K, T, r, sigma, M=150,
    N=1000):
12     # --- 1. FRONTIERA CON DIFFERENZE FINITE (IMPLICITO) ---
13     S_max, S_min = S0 * 3, S0 / 3
14     Z_max, Z_min = np.log(S_max), np.log(S_min)
15     dz = (Z_max - Z_min) / M
16     dt_fd = T / N
17     z_grid = np.linspace(Z_min, Z_max, M + 1)
18     s_grid_fd = np.exp(z_grid)
19     t_fd = np.linspace(0, T, N + 1)
20
21     v = np.maximum(K - s_grid_fd, 0)
22     boundary_fd = np.zeros(N + 1)
23     # Alla scadenza, la frontiera e' K
24     boundary_fd[-1] = K
25
26     drift = r - 0.5 * sigma**2
27     alpha = (dt_fd / (2 * dz)) * drift - (dt_fd / (2 *
        dz**2)) * sigma**2
28     beta = 1 + (dt_fd / dz**2) * sigma**2 + r * dt_fd
29     gamma = -(dt_fd / (2 * dz)) * drift - (dt_fd / (2 *
        dz**2)) * sigma**2
30
31     A = np.zeros((M - 1, M - 1))
32     for i in range(M - 1):
33         A[i, i] = beta
34         if i > 0: A[i, i - 1] = alpha
35         if i < M - 2: A[i, i + 1] = gamma
36

```



```

37 # Indietro nel tempo per differenze finite
38 for j in range(N - 1, -1, -1):
39     rhs = v[1:-1].copy()
40     v[1:-1] = np.linalg.solve(A, rhs)
41
42     # Condizioni al contorno e vincolo esercizio
43     v[0] = K - np.exp(Z_min)
44     v[-1] = 0
45     v = np.maximum(v, K - s_grid_fd)
46
47     # Check esercizio per determinare la frontiera
48     # Troviamo il primo indice da destra dove e' ottimale
49     # esercitare (prezzo piu' alto)
50     payoff = K - s_grid_fd
51     # v e' gia' il massimo tra payoff e continuazione,
52     # quindi controlliamo l'uguaglianza
53
54     # tolleranza numerica
55     exercise_idx = np.where(np.abs(payoff - v) < 1e-12)[0]
56
57     if len(exercise_idx) > 0:
58         boundary_fd[j] = s_grid_fd[exercise_idx[-1]]
59     else:
60         boundary_fd[j] = np.nan
61
62     # --- 2. FRONTIERA CON ALBERO BINOMIALE (CRR) ---
63     # Usiamo meno passi per il binomiale per visualizzare
64     # meglio la struttura discreta
65
66     steps_bin = 200
67     dt_bin = T / steps_bin

```

```

63     u = np.exp(sigma * np.sqrt(dt_bin))
64     d = 1 / u
65     q = (np.exp(r * dt_bin) - d) / (u - d)
66     t_bin = np.linspace(0, T, steps_bin + 1)
67     boundary_bin = np.zeros(steps_bin + 1)
68
69     tree_prices = []
70     for i in range(steps_bin + 1):
71         prices = S0 * (u ** np.arange(i, -1, -1)) * (d **
72             np.arange(0, i + 1, 1))
73         tree_prices.append(prices)
74
75     # Valori al terminale
76     current_prices = tree_prices[-1]
77     v_bin = np.maximum(K - current_prices, 0)
78
79     # Alla scadenza, la frontiera e' K
80     boundary_bin[-1] = K
81
82     # Indietro nell'albero per estrarre la frontiera
83     for i in range(steps_bin - 1, -1, -1):
84         # Valori di continuazione
85         continuation = np.exp(-r * dt_bin) * (q * v_bin[: -1]
86             + (1 - q) * v_bin[1:])
87
88         # Prezzi a questo passo temporale
89         current_prices = tree_prices[i]
90
91         # Payoff se esercitato

```

```

90     payoff = np.maximum(K - current_prices, 0)
91
92     # Identificazione frontiera: il prezzo piu' alto dove
93     payoff >= continuazione
94
95     exercise_mask = (payoff >= continuation) & (payoff >
96     0)
97
98     if np.any(exercise_mask):
99         # Il prezzo piu' alto in cui l'esercizio e'
100         ottimale
101         boundary_bin[i] =
102             np.max(current_prices[exercise_mask])
103     else:
104         # Nessun esercizio ottimale a questo t
105         boundary_bin[i] = np.nan
106
107     # Aggiorniamo i valori dell'opzione per il prossimo
108     passo (valido per pricing)
109     v_bin = np.maximum(continuation, payoff)
110
111     # --- PLOT FINALE ---
112     plt.figure(figsize=(12, 7))
113     plt.plot(t_fd, boundary_fd, 'b-', label='Diff. Finite
114     (Implicito)', alpha=0.9)
115
116     # Usiamo plt.step per mostrare la natura discreta
117     dell'albero binomiale
118
119     plt.step(t_bin, boundary_bin, 'r--', where='post',
120             label=f'Albero Binomiale ({steps_bin} step)',
121             alpha=0.9)

```

```

110
111     plt.axhline(K, color='black', linestyle=':',
112               label='Strike_K=50')
113
114     plt.title(f"Confronto_Corretto_Early_Exercise_Boundary_
115              (Put_Americana)\n$S_0={S0}, K={K}, \sigma={sigma},
116              r={r}, T={T}$")
117
118     plt.xlabel("Tempo_alla_scadenza_(Anni)")
119     plt.ylabel("Prezzo_Critico_$S^*$")
120     plt.legend()
121     plt.grid(True, which='both', linestyle='--', alpha=0.4)
122     plt.ylim(30, 55) # Limitiamo l'asse Y per ingrandire
123                       l'area d'interesse
124     plt.show()
125
126 # Esecuzione della funzione
127 get_corrected_boundaries(S0, K, T, r, sigma)

```

Si allega ora il codice relativo alla visualizzazione della frontiera d'esercizio anticipato per la put scritta sul titolo Apple. Da sottolineare la modifica applicata alla limitazione dell'asse Y per visionare meglio l'area d'interesse.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # 1. PARAMETRI DI INPUT (APPLE)
5 S0 = 293.320007
6 K = 294.0
7 T = 1.0
8 r = 0.0415
9 sigma = 0.273915

```

```

10
11 def get_corrected_boundaries(S0, K, T, r, sigma, M=150,
    N=1000):
12     # --- 1. FRONTIERA CON DIFFERENZE FINITE (IMPLICITO) ---
13     S_max, S_min = S0 * 3, S0 / 3
14     Z_max, Z_min = np.log(S_max), np.log(S_min)
15     dz = (Z_max - Z_min) / M
16     dt_fd = T / N
17     z_grid = np.linspace(Z_min, Z_max, M + 1)
18     s_grid_fd = np.exp(z_grid)
19     t_fd = np.linspace(0, T, N + 1)
20
21     v = np.maximum(K - s_grid_fd, 0)
22     boundary_fd = np.zeros(N + 1)
23     boundary_fd[-1] = K
24
25     drift = r - 0.5 * sigma**2
26     alpha = (dt_fd / (2 * dz)) * drift - (dt_fd / (2 *
        dz**2)) * sigma**2
27     beta = 1 + (dt_fd / dz**2) * sigma**2 + r * dt_fd
28     gamma = -(dt_fd / (2 * dz)) * drift - (dt_fd / (2 *
        dz**2)) * sigma**2
29
30     A = np.zeros((M - 1, M - 1))
31     for i in range(M - 1):
32         A[i, i] = beta
33         if i > 0: A[i, i - 1] = alpha
34         if i < M - 2: A[i, i + 1] = gamma
35

```

```

36     for j in range(N - 1, -1, -1):
37         rhs = v[1:-1].copy()
38         v[1:-1] = np.linalg.solve(A, rhs)
39         v[0] = K - np.exp(Z_min)
40         v[-1] = 0
41         v = np.maximum(v, K - s_grid_fd)
42
43         payoff = K - s_grid_fd
44         exercise_idx = np.where(np.abs(payoff - v) < 1e-12)[0]
45
46         if len(exercise_idx) > 0:
47             boundary_fd[j] = s_grid_fd[exercise_idx[-1]]
48         else:
49             boundary_fd[j] = np.nan
50
51     # --- 2. FRONTIERA CON ALBERO BINOMIALE (CRR) ---
52     steps_bin = 200
53     dt_bin = T / steps_bin
54     u = np.exp(sigma * np.sqrt(dt_bin))
55     d = 1 / u
56     q = (np.exp(r * dt_bin) - d) / (u - d)
57     t_bin = np.linspace(0, T, steps_bin + 1)
58     boundary_bin = np.zeros(steps_bin + 1)
59
60     tree_prices = []
61     for i in range(steps_bin + 1):
62         prices = S0 * (u ** np.arange(i, -1, -1)) * (d **
63             np.arange(0, i + 1, 1))
64         tree_prices.append(prices)

```

```

64
65     current_prices = tree_prices[-1]
66     v_bin = np.maximum(K - current_prices, 0)
67     boundary_bin[-1] = K
68
69     for i in range(steps_bin - 1, -1, -1):
70         continuation = np.exp(-r * dt_bin) * (q * v_bin[: -1]
71             + (1 - q) * v_bin[1:])
72         current_prices = tree_prices[i]
73         payoff = np.maximum(K - current_prices, 0)
74         exercise_mask = (payoff >= continuation) & (payoff >
75             0)
76
77         if np.any(exercise_mask):
78             boundary_bin[i] =
79                 np.max(current_prices[exercise_mask])
80         else:
81             boundary_bin[i] = np.nan
82
83         v_bin = np.maximum(continuation, payoff)
84
85     # --- PLOT FINALE ---
86     plt.figure(figsize=(12, 7))
87     plt.plot(t_fd, boundary_fd, 'b-', label='Diff. Finite
88         (Implicit)', alpha=0.9)
89     plt.step(t_bin, boundary_bin, 'r--', where='post',
90         label=f'Albero Binomiale ({steps_bin} step)',
91         alpha=0.9)

```

```

87     plt.axhline(K, color='black', linestyle=':',
            label=f'Strike_K={K}')
88     plt.title(f"Confronto_Corretto_Early_Exercise_Boundary_
            (Put_AAPL)\n$S_0={S0:.2f}, K={K}, \sigma={sigma:.4f},
            r={r}, T={T}$")
89     plt.xlabel("Tempo (Anni)")
90     plt.ylabel("Prezzo Critico $S^*$")
91     plt.legend()
92     plt.grid(True, which='both', linestyle='--', alpha=0.4)
93
94     # Adattare l'asse Y ai nuovi valori Apple
95     plt.ylim(K * 0.7, K * 1.1)
96     plt.show()
97
98     # Esecuzione
99     get_corrected_boundaries(S0, K, T, r, sigma)

```

Si conclude mostrando il codice per il calcolo della greca Theta, relativa alla sensibilità dell'opzione al trascorrere del tempo.

```

1  import numpy as np
2
3  # Parametri dell'opzione
4  S0 = 293.320007
5  K = 294
6  T = 1.0
7  r = 0.0415
8  sigma = 0.273915
9  dt_theta = 1/365 # Step temporale (1 giorno)
10

```



```

11 def binomial_american_put(S, K, T, r, sigma, N=1000):
12     """Modello Binomiale Cox-Ross-Rubinstein per Put
13         Americana"""
14
15     if T <= 0: return max(0, K - S)
16
17     dt = T / N
18
19     u = np.exp(sigma * np.sqrt(dt))
20     d = 1 / u
21     p = (np.exp(r * dt) - d) / (u - d)
22     disc = np.exp(-r * dt)
23
24     # Prezzi a scadenza
25     S_tree = S * (u ** np.arange(N, -1, -1)) * (d **
26         np.arange(0, N + 1))
27     v = np.maximum(K - S_tree, 0)
28
29     # Back-propagation con controllo esercizio anticipato
30     for i in range(N - 1, -1, -1):
31         v = disc * (p * v[:-1] + (1 - p) * v[1:])
32         S_tree = S * (u ** np.arange(i, -1, -1)) * (d **
33             np.arange(0, i + 1))
34         v = np.maximum(v, K - S_tree)
35     return v[0]

```

```

31 def implicit_fd_american_put(S, K, T, r, sigma, Smax=S0*2,
32     M=200, N=2000):
33
34     """Metodo differenze finite implicite per Put Americana"""
35
36     dt = T / N
37
38     dS = Smax / M
39     s_values = np.linspace(0, Smax, M + 1)

```

```

36     v = np.maximum(K - s_values, 0)
37
38     j = np.arange(1, M)
39     a = 0.5 * dt * (r * j - (sigma**2) * (j**2))
40     b = 1 + dt * (r + (sigma**2) * (j**2))
41     c = -0.5 * dt * (r * j + (sigma**2) * (j**2))
42
43     matrix = np.diag(a[1:], -1) + np.diag(b) +
44         np.diag(c[:-1], 1)
45
46     for n in range(N):
47         rhs = v[1:M]
48         rhs[0] -= a[0] * K
49         v_inner = np.linalg.solve(matrix, rhs)
50         v[1:M] = np.maximum(v_inner, K - s_values[1:M])
51         v[0], v[M] = K, 0
52
53     return np.interp(S, s_values, v)
54
55 # Calcolo Theta
56 # Modello Binomiale
57 v_bin_t = binomial_american_put(S0, K, T, r, sigma)
58 v_bin_dt = binomial_american_put(S0, K, T - dt_theta, r,
59     sigma)
60 theta_bin = (v_bin_dt - v_bin_t) / dt_theta
61
62 # Modello Differenze Finite
63 v_fd_t = implicit_fd_american_put(S0, K, T, r, sigma)
64 v_fd_dt = implicit_fd_american_put(S0, K, T - dt_theta, r,
65     sigma)

```

```

62 theta_fd = (v_fd_dt - v_fd_t) / dt_theta
63
64 # Visualizzazione Risultati
65 print(f"{'Metodo':<30}{'Theta':<15}")
66 print("-" * 45)
67 print(f"{'Binomiale':<30}{theta_bin:10.6f}")
68 print(f"{'Differenze Finite Implicite':<30}{theta_fd:10.6f}")

```

Bibliografia

- Black, F. & Scholes, M. (1973), 'The pricing of options and corporate liabilities', *Journal of Political Economy* **81**, 637–654.
- Boyle, P. P. (1977), 'Options: A monte carlo approach', *Journal of Financial Economics* **4**, 323–338.
- Boyle, P. P., Broadie, M. & Glasserman, P. (1997), 'Monte carlo methods for security pricing', *Journal of Economic Dynamics and Control* **21**, 1267–1322.
- Brennan, M. J. & Schwartz, E. S. (1977), 'The valuation of american options by finite difference methods', *Journal of Finance* **32**, 447–462.
- Broadie, M., Glasserman, P. & Jain, G. (1997), 'Enhanced monte carlo estimates for american option prices', *Journal of Derivatives* **5**(1), 25–44.
- Clelow, L. & Strickland, C. (1998), *Implementing Derivatives Models*, Wiley.
- Courant, R., Friedrichs, K. & Lewy, H. (1928), 'Über die partiellen differenzengleichungen der mathematischen physik', *Mathematische Annalen* **100**, 32–74.
- Cox, J. C., Ross, S. A. & Rubinstein, M. (1979), 'Option pricing: A simplified approach', *Journal of Financial Economics* **7**, 229–264.

- El-Nabulsi, R. A. & Anukool, W. (2026), 'Black-scholes equation in quantitative finance with variable parameters: A path to a generalized schrodinger equation', *Financial Innovation* **12**, 22.
- Evans, L. C. (2010), *Partial Differential Equations*, American Mathematical Society.
- Figlewski, S. & Gao, B. (1999), 'The adaptive mesh model: A new approach to efficient option pricing', *Journal of Financial Economics* **53**(3), 313–351.
- Haberman, R. (2013), *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*, Pearson.
- Hull, J. C. (2018), *Options, Futures, and Other Derivatives*, 10 edn, Pearson.
- Hull, J. C. & White, A. (1988), 'The use of the control variate technique in option pricing', *Journal of Financial and Quantitative Analysis* **23**, 237–251.
- Hull, J. C. & White, A. (1990), 'Valuing derivative securities using the explicit finite difference method', *Journal of Financial and Quantitative Analysis* **25**, 87–100.
- Merton, R. C. (1973), 'Theory of rational option pricing', *Bell Journal of Economics and Management Science* **4**, 141–183.
- Metropolis, N. & Ulam, S. (1949), 'The monte carlo method', *Journal of the American Statistical Association* **44**(247), 335–341.
- Rendleman, R. J. & Bartter, B. (1979), 'Two state option pricing', *Journal of Finance* **34**, 1092–1110.
- Smith, G. D. (1985), *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford University Press.
- Tychonov, A. N. & Samarski, A. A. (1964), *Partial Differential Equations of Mathematical Physics, Vol. I*, Holden-Day.

- Tychonov, A. N. & Samarski, A. A. (1967), *Partial Differential Equations of Mathematical Physics, Vol. II*, Holden-Day.
- Wilmott, P. (1998), *Derivatives: The Theory and Practice of Financial Engineering*, Wiley.
- Zachmanoglou, E. C. & Thoe, D. W. (1986), *Introduction to Partial Differential Equations with Applications*, Dover Publications.