Corso di Fisica Computazionale

- Terza Esercitazione -

L'EQUAZIONE DI BLACK-SCHOLES

Cescato Matteo, Garbi Luca, Libardi Gabriele

Issue: 1

19 agosto 2020

Università degli Studi di Trento
Dipartimento di Fisica
Via Sommarive 14, 38123
Povo (TN), Italia

Indice

Abstract

In questa esercitazione, viene studiato il modello finanziario di Black-Scholes-Merton. In particolare, viene ricavata e studiata l'equazione alle derivate parziali di Black-Scholes, che modellizza la dinamica di uno strumento derivato. Essa viene risolta sia analiticamente che numericamente, facendo uso di due metodi alle differenze finite, uno esplicito e l'altro implicito. Infine, vengono discussi i risultati ottenuti, ponendo a confronto le soluzioni numeriche con la soluzione analitica.

1 Il modello di Black-Scholes-Merton

Nella prima parte di questa sezione viene esposto il modello di Black-Scholes-Merton, fino a ricavare l'equazione differenziale alle derivate parziali di Black-Scholes (BS). Nella seconda parte viene risolta l'equazione utilizzando condizioni al contorno che permettano di fornire una formula esplicita per la dinamica dello strumento derivato.

1.1 L'equazione di Black-Scholes

Nella nostra trattazione ci concentreremo solamente sulle Opzioni *Call* (la soluzione è comunque analoga per Opzioni *Put*) di tipo Europeo, ovvero con la possibilità di esercitare il contratto di acquisto o meno solo alla data di scadenza. In particolare, il tipo di opzione analizzato è il *Plain Vanilla*, senza variazioni di tipo esotico. Per quanto riguarda la notazione: verrà utilizzato $S(t)$ per indicare il prezzo dell'*asset* sottostante al tempo t, con $V(S,\, t)$ si indicherà il prezzo dell'opzione come funzione di S al tempo t, mentre E sarà lo *strike price* dell'opzione, ovvero il prezzo di esercizio alla scadenza del contratto, che assumiamo avvenire al tempo $t = T$.
È necessario specificare le assunzioni che vengono fatte per lo sviluppo del modello:

- possibilità di acquisto o vendita di una qualsiasi quantità, anche frazionaria, di sottostante S e di un *asset* senza rischio R (tipicamente obbligazioni) con un tasso di interesse composto costante r;
- l'*asset* non paga alcun dividendo, se non il premio di vendita, e nessuna transazione è soggetta a commissione di sorta;
- l'andamento dell'*asset* in funzione del tempo è un processo markoviano e segue un moto browniano geometrico con *drift* e volatilità costanti;
- sul mercato non c'è opportunità di arbitraggio, ovvero non è possibile ottenere un profitto (con ritorno superiore a quello dato da r) senza rischi.

L'intuizione finanziaria dietro al modello di Black-Scholes-Merton è che sia possibile eliminare completamente il rischio dell'acquisto o vendita di un'opzione attraverso, rispettivamente, la vendita o l'acquisto dell'*asset* sottostante (fenomeno chiamato *hedging*). Di conseguenza, a causa dell'ultima assunzione elencata, ci dovrà essere un *fair price*, ossia un prezzo corretto per il derivato V tale da non permettere di avere un profitto (superiore a quello che si avrebbe investendo nell'asset R) senza rischi. Inoltre, dato il tipo di PDE e di condizioni al contorno che si trovano con questa trattazione, emerge che il *fair price*, per ogni istante di tempo t, è unico e determinato interamente da S, r, E e T.

Secondo l'assunzione che il prezzo del sottostante segua un moto geometrico browniano, possiamo descrivere la variazione di S in un intervallo infinitesimo $\mathrm{d}t$ come

$$\mathrm{d}S = \mu S \mathrm{d}t + \sigma S \mathrm{d}z \,. \tag{1}$$

Qui il parametro assunto noto μ è il *rate* di ritorno atteso per l'*asset* e identifica un *trend* del prezzo, σ invece è un parametro che rappresenta la volatilità del prezzo, ovvero la deviazione standard intorno al prezzo medio dato dal termine di *trend*, $\mathrm{d}z$ infine è un incremento infinitesimo di una variabile stocastica $z(t)$, ben modellizzata da una *random walk*. In definitiva, l'equazione (1) ci dice che, localmente, il tasso di ritorno dell'*asset* ha un valore d'aspettazione $\mu \mathrm{d}t$ ed una varianza $\sigma^2 \mathrm{d}t$.
A questo punto, sapendo che il derivato è una funzione deterministica di S e di t, possiamo utilizzare il lemma di Itô (un'applicazione dello sviluppo in serie di Taylor per funzioni di più variabili) per scrivere il differenziale di una funzione dipendente dal tempo per un processo stocastico:

$$\mathrm{d}V = \left(\frac{\partial V}{\partial S}\mu S + \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2 \right) \mathrm{d}t + \frac{\partial V}{\partial S}\sigma S \mathrm{d}z \,. \tag{2}$$

Con Π definiamo ora il nostro portafoglio di investimento misto, strutturato così:

$$\Pi = \alpha(t)R + \beta(t)S + V \,, \tag{3}$$

dove $\alpha(t)$ e $\beta(t)$ rappresentano, rispettivamente, le quantità di *asset* senza rischio a prezzo R e di sottostante S. Notiamo, en passant, che, per come è definito l'*asset* senza rischio, vale $\mathrm{d}R = rR\mathrm{d}t$.
Calcoliamo la variazione del portafoglio $\mathrm{d}\Pi$ dopo un tempo $\mathrm{d}t$:

$$\begin{aligned}
\mathrm{d}\Pi &= \alpha \mathrm{d}R + \beta \mathrm{d}S + \mathrm{d}V \\
&= \alpha r R \mathrm{d}t + \beta(\mu S \mathrm{d}t + \sigma S \mathrm{d}z) + \left(\frac{\partial V}{\partial S}\mu S + \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2 \right) \mathrm{d}t + \frac{\partial V}{\partial S}\sigma S \mathrm{d}z \\
&= \left(\alpha r R + \beta \mu S + \frac{\partial V}{\partial S}\mu S + \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2 \right) \mathrm{d}t + \left(\beta \sigma S + \frac{\partial V}{\partial S}\sigma S \right) \mathrm{d}z
\end{aligned}$$

Dall'ultima espressione risulta evidente come si possano eliminare gli effetti stocastici della volatilità, presenti nel differenziale $\mathrm{d}z$, scegliendo $\beta = -\frac{\partial V}{\partial S}$. Fatta questa sostituzione, il differenziale del portafoglio diventa

$$\mathrm{d}\Pi = \mathrm{d}(\alpha R + \beta S + V) = \left(\alpha r R + \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2 \right) \mathrm{d}t \,.$$

A questo punto, bisogna imporre la condizione di *fair value* che non permetta l'arbitraggio. Quindi imponiamo che il ritorno massimo che si possa avere investendo con il portafoglio misto (3) sia pari a quello che si avrebbe se si investisse in un portafoglio formato solamente da *asset* senza rischio (quindi con $\Pi = R$). Otteniamo la condizione $\mathrm{d}\Pi = r\Pi\mathrm{d}t$, quindi

$$r\Pi\mathrm{d}t = r(\alpha R + \beta S + V)\mathrm{d}t = \left(\alpha r R + \frac{\partial V}{\partial t} + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2\right)\mathrm{d}t\,,$$

che, elidendo i termini dipendenti da α, porta all'**equazione di Black-Scholes**:

$$\boxed{\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0} \tag{4}$$

La condizione finale data dall'opzione di esecuzione o meno del contratto al *payoff time* T è

$$V(S,\, T) = \max\{S - E,\, 0\}\,, \qquad \text{per } S \geq 0\,. \tag{5}$$

Ora, essendo presente una derivata parziale al secondo ordine rispetto ad S, sono necessarie due condizioni al contorno per questa variabile. Scegliamo le seguenti:

$$\begin{cases} V(0,\, t) = 0 & \text{per ogni } t \\ V(S,\, t) \to S & \text{per } S \to \infty \end{cases} \tag{6}$$

dove la prima rispecchia il fatto che il prezzo dello strumento derivato sia nullo quando lo è quello del sottostante; inoltre è chiaro come per la (5) il contratto non abbia alcun valore se al tempo finale vale $S = 0$. La seconda condizione, invece, deriva dal fatto che per $S \to \infty$ è sempre più probabile che $S(T)$ sia maggiore del prezzo di esercizio E e quindi che il valore del contratto, in questo limite, sia $V \sim S - E \sim S$. È opportuno notare come la seconda condizione, seppur scelta con cognizione di causa rispetto alla realtà, non sempre rispecchi la natura degli strumenti derivati.

1.2 Risoluzione analitica dell'equazione di Black-Scholes

Per un'Opzione Europea di tipo *Call* si può mostrare come l'equazione (4) risulti sostanzialmente un'equazione di diffusione.
Volendo trasformare l'equazione parabolica di tipo *backward* in una di tipo *forward*, più comoda nella risoluzione, procediamo effettuando le sostituzioni

$$\tau \equiv \frac{\sigma^2}{2}(T - t)\,, \qquad x \equiv \ln\left(\frac{S}{E}\right), \qquad v \equiv \frac{V}{E}\,,$$

con le quali otteniamo per le derivate

$$\frac{\partial V}{\partial t} = -\frac{E\sigma^2}{2}\frac{\partial v}{\partial \tau}\,,$$

$$\frac{\partial V}{\partial S} = \frac{E}{S}\frac{\partial v}{\partial x}\,,$$
$$\frac{\partial^2 V}{\partial S^2} = \frac{\partial}{\partial S}\left(\frac{\partial V}{\partial S}\right) = -\frac{E}{S^2}\frac{\partial v}{\partial x} + \frac{E}{S^2}\frac{\partial^2 v}{\partial x^2}\,,$$

e per la condizione finale (5)

$$V(S,\,T) = \max\{S - E,\,0\} = \max\{Ee^x - E,\,0\} \implies v(x,\,0) = \max\{e^x - 1,\,0\}\,.$$

Sostituendo le derivate all'interno dell'equazione BS si ottiene

$$-\frac{E\sigma^2}{2}\frac{\partial v}{\partial \tau} + \frac{1}{2}\sigma^2 S\left(-\frac{E}{S^2}\frac{\partial v}{\partial x} + \frac{E}{S^2}\frac{\partial^2 v}{\partial x^2}\right) + rs\left(\frac{E}{S}\frac{\partial v}{\partial x}\right) - rEv = 0\,,$$

che, semplificando e sostituendo $a \equiv (2r/\sigma^2) - 1$ e $b \equiv -(a+1)$, diventa

$$\frac{\partial v}{\partial \tau} = \frac{\partial^2 v}{\partial x^2} + a\frac{\partial v}{\partial x} + bv\,, \tag{7}$$

ovvero un'equazione di diffusione. Notiamo che, grazie al cambiamento di variabile, ora la variabile x può assumere tutti i valori della retta reale e l'equazione è a coefficienti costanti, determinati dal coefficiente di volatilità e dal tasso di interesse. Possiamo semplificare ulteriormente l'equazione introducendo un altro cambio di variabile:

$$u(x,\,\tau) \equiv \frac{v(x,\,\tau)}{e^{\lambda x + \nu\tau}}\,,$$

con λ e ν coefficienti da determinare. Le derivate temporali e spaziali di v ora risultano

$$\frac{\partial v}{\partial t} = \nu e^{\lambda x + \nu\tau} u + e^{\lambda x + \nu\tau}\frac{\partial u}{\partial \tau}\,,$$
$$\frac{\partial v}{\partial x} = \lambda e^{\lambda x + \nu\tau} u + e^{\lambda x + \nu\tau}\frac{\partial u}{\partial x}\,,$$
$$\frac{\partial^2 v}{\partial x^2} = \lambda^2 e^{\lambda x + \nu\tau} u + 2\lambda e^{\lambda x + \nu\tau}\frac{\partial u}{\partial x} + e^{\lambda x + \nu\tau}\frac{\partial^2 u}{\partial x^2}\,.$$

A questo punto, possiamo scegliere opportunamente i coefficienti. Prendendo $\lambda = -a/2$ e $\nu = \lambda^2 + a\lambda + b$, l'equazione (7) si riduce semplicemente a

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}\,, \tag{8}$$

ovvero un'equazione del calore con condizione iniziale

$$u(x,\,0) \equiv u_0(x) = e^{\lambda x + \nu 0} v(x,\,0) = e^{\lambda x}\max\{e^x - 1,\,0\} = \max\left\{2e^{x(a+1)/2}\sinh\left(\frac{x}{2}\right),\,0\right\}.$$

La soluzione generale per l'equazione del calore si ricava semplicemente sfruttando le trasformate di Fourier e risulta

$$u(x,\,\tau) = \frac{1}{2\sqrt{\pi\tau}} \int_{-\infty}^{+\infty} u_0(s) e^{-\frac{(x-s)^2}{4\tau}} \mathrm{d}s\,.$$

Riscriviamo l'integrale con il cambiamento di variabile $z \equiv (s-x)/\sqrt{2\tau}$:

$$u(x,\,\tau) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} u_0\left(z\sqrt{2\tau}+x\right) e^{-\frac{z^2}{2}} \mathrm{d}z\,.$$

Poiché la funzione u_0 si annulla per $x \le 0$, possiamo utilizzare come estremo inferiore dell'integrale $z = -x/\sqrt{2\tau}$. La funzione precedente allora diventa la somma di due integrali del tipo

$$\frac{1}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau}}^{+\infty} e^{\frac{q_{1,2}}{2}(x+z\sqrt{2\tau})} e^{-\frac{z^2}{2}} \mathrm{d}z\,,$$

dove $q_1 \equiv a+2$ e $q_2 \equiv a+1$. Questo tipo di integrale si trova tabulato e si riscrive come

$$\frac{e^{\frac{qx}{2}+\frac{\tau q^2}{4}}}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau}-q\sqrt{\tau/2}}^{+\infty} e^{-\frac{z^2}{2}} \mathrm{d}z\,.$$

Quindi l'integrale rimasto risulta essere la funzione di ripartizione di una distribuzione gaussiana; allora, definendo

$$\Phi(d) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d} e^{-\frac{z^2}{2}} \mathrm{d}z\,,$$

la soluzione per l'equazione del calore (8) risulta

$$u(x,\,\tau) = e^{\frac{(a+2)x}{2}+\frac{\tau(a+2)^2}{4}} \Phi(d_1) - e^{\frac{(a+1)x}{2}+\frac{\tau(a+1)^2}{4}} \Phi(d_2)\,,$$

con $d_{1,2} \equiv \frac{x}{\sqrt{2\tau}} + q_{1,2}\sqrt{\tau/2}$. Per concludere, torniamo alle variabili di partenza: ponendo

$$d_+ \equiv \frac{\log\left(\frac{S}{E}\right) + (T-t)(r+\sigma^2/2)}{\sigma\sqrt{T-t}}\,, \tag{9}$$

$$d_- \equiv \frac{\log\left(\frac{S}{E}\right) + (T-t)(r-\sigma^2/2)}{\sigma\sqrt{T-t}}\,, \tag{10}$$

la soluzione dell'equazione BS per la dinamica di un'Opzione Europea di tipo *Call* risulta

$$\boxed{V(S,\,t) = S\Phi(d_+) - Ee^{-r(T-t)}\Phi(d_-)} \tag{11}$$

2 Risoluzione numerica dell'equazione BS

In questa sezione vengono discusse le metodologie utilizzate per la risoluzione numerica dell'equazione BS (4). In particolare, vengono utilizzati due metodi alle differenze finite, uno esplicito e l'altro implicito, in cui il problema è ricondotto all'inversione di una matrice tridiagonale. Per l'applicazione di tali metodi si rende necessaria la suddivisione dell'intervallo temporale $[0,\,T]$ in una *mesh* di $M \in \mathbb{N}$ sottointervalli, ciascuno di ampiezza $\mathit{\Delta} T \equiv T/M$. L'intervallo $[0,\,\infty)$ su cui è definito il prezzo S dell'*asset*, invece, viene limitato superiormente introducendo il limite artificiale $S_{max} \equiv 3E$. Tale assunzione, nonostante permetta di ottenere risultati che, almeno per bassi valori di S, presentano un buon accordo con la soluzione analitica dell'equazione BS, rappresenta comunque un'approssimazione forte, i cui limiti risulteranno evidenti nel confronto tra soluzione esatta e numerica presentato nella sezione 3. L'intervallo $[0,\,S_{max}]$ su cui si limita la soluzione viene diviso in $N \in \mathbb{N}$ sottointervalli, ciascuno di ampiezza $\mathit{\Delta} S \equiv S_{max}/N$. Dunque, il piano $[0,\,S_{max}] \times [0,\,T]$ è approssimato dal reticolo di punti

$$(n\mathit{\Delta} S,\, m\mathit{\Delta} t) \in [0,\, N\mathit{\Delta} S] \times [0,\, M\mathit{\Delta} t]\,,$$

con $n = 0,\,1, ...,\,N$ e $m = 0,\,1, ...,\,M$.

Per semplificare la trattazione successiva, si introduce inoltre il cambio di variabile

$$\mathcal{T} \equiv T - t\,,$$

con cui l'equazione (4) diventa

$$\frac{\partial \tilde{V}}{\partial \mathcal{T}} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \tilde{V}}{\partial S^2} - rS\frac{\partial \tilde{V}}{\partial S} + r\tilde{V} = 0\,, \tag{12}$$

avendo definito $\tilde{V}(S,\,\mathcal{T}) \equiv V(S,\,T-\mathcal{T})$. Di conseguenza, la condizione finale (5) diventa

$$\tilde{V}(S,\,0) = \max\{S - E,\,0\}\,, \tag{13}$$

mentre le condizioni al bordo (6), tenendo conto del limite superiore S_{max} al valore di S, assumono la forma

$$\begin{cases} \tilde{V}(0,\,\mathcal{T}) = 0 \\ \tilde{V}(S_{max},\,\mathcal{T}) = S_{max} - E \end{cases} \tag{14}$$

per ogni $\mathcal{T} \in [0,\,T]$. Per alleggerire la notazione, in seguito si utilizzerà il simbolo v_n^m per indicare la quantità $\tilde{V}(n\mathit{\Delta} S,\,m\mathit{\Delta} t)$.

Nella nostra risoluzione numerica dell'equazione BS, per entrambi i metodi utilizzati, si fissano per i parametri i valori $T = 0.25$, $E = 10$, $r = 0.1$, $\sigma = 0.4$, e si considerano due casi distinti: $N = 200$, $M = 2000$, e $N = 1000$, $M = 41000$.

2.1 Metodo esplicito

Al fine di risolvere l'equazione BS riscritta nella forma (12), si introduce una scrittura discretizzata delle derivate parziali di $\tilde{V}$. La derivata temporale si approssima considerando una differenza finita in avanti (*forward difference*):

$$\frac{\partial \tilde{V}}{\partial \mathcal{T}}(n\Delta S,\, m\Delta t) = \frac{\mathrm{v}_n^{m+1} - \mathrm{v}_n^m}{\Delta t} + \mathcal{O}(\Delta t)\,.$$

Per le derivate rispetto a S si considera invece una differenza finita centrata simmetrica (*symmetric central difference*):

$$\begin{aligned}\frac{\partial \tilde{V}}{\partial S}(n\Delta S,\, m\Delta t) &= \frac{\mathrm{v}_{n+1}^m - \mathrm{v}_{n-1}^m}{2\Delta S} + \mathcal{O}\big((\Delta S)^2\big)\,,\\ \frac{\partial^2 \tilde{V}}{\partial S^2}(n\Delta S,\, m\Delta t) &= \frac{\mathrm{v}_{n+1}^m - 2\mathrm{v}_n^m + \mathrm{v}_{n-1}^m}{(\Delta S)^2} + \mathcal{O}\big((\Delta S)^2\big)\,.\end{aligned}$$

Sostituendo tali derivate approssimate nell'equazione (12), si trova

$$\frac{\mathrm{v}_n^{m+1} - \mathrm{v}_n^m}{\Delta t} - \frac{1}{2}\sigma^2 n^2 (\Delta S)^2 \frac{\mathrm{v}_{n+1}^m - 2\mathrm{v}_n^m + \mathrm{v}_{n-1}^m}{(\Delta S)^2} - rn\Delta S \frac{\mathrm{v}_{n+1}^m - \mathrm{v}_{n-1}^m}{2\Delta S} + r\mathrm{v}_n^m = 0\,, \quad (15)$$

con $n = 1, 2, ..., N-1$ e $m = 1, 2, ..., M-1$. L'errore complessivo è di ordine $\mathcal{O}\big(\Delta t + (\Delta S)^2\big)$. Isolando v_n^{m+1} nella (15) e definendo le quantità

$$\alpha \equiv \sigma^2 \Delta t\,, \qquad \beta \equiv r\Delta t\,, \quad (16)$$

si ottiene infine

$$\mathrm{v}_n^{m+1} = \frac{1}{2}(\alpha n^2 - \beta n)\mathrm{v}_{n-1}^m + (1 - \alpha n^2 - \beta)\mathrm{v}_n^m + \frac{1}{2}(\alpha n^2 + \beta n)\mathrm{v}_{n+1}^m\,.$$

Grazie a tale formula, conoscendo $\mathrm{v}_n^{m^*}$ per ogni $n \in [0,\, N] \cap \mathbb{N}$, con m^* fissato, è possibile ricavare un'approssimazione di $\mathrm{v}_n^{m^*+1}$, essendo note le quantità $\mathrm{v}_0^{m^*+1}$ e $\mathrm{v}_N^{m^*+1}$ per l'imposizione delle condizioni al bordo (14). Procedendo iterativamente a partire dalla condizione iniziale (13), è dunque possibile ottenere un'approssimazione numerica dell'intera funzione $\tilde{V}(S,\, \mathcal{T})$.

Si può dimostrare come il metodo esplicito appena presentato risulti stabile se è verificata la condizione[1]

$$0 < \Delta t < \frac{1}{\sigma^2(N-1) + \frac{1}{2}r}\,.$$

Per i valori scelti per i parametri, tale condizione risulta verificata per entrambe le coppie di valori $(N,\, M)$ considerate. Dunque, nel nostro caso, l'algoritmo esplicito

[1] Per la dimostrazione si veda ad esempio S. Ikonen, *Black-Scholes-yhtälöstä ja sen numeerisesta ratkaisemisesta differenssimenetelmällä*, MD Thesis, University of Jyväskylä, Finland, 2001.

analizzato risulta stabile.

2.2 Metodo implicito

Un altro modo per risolvere l'equazione BS nella forma (12) consiste nell'utilizzare quello che è noto come *full implicit method.* Anche in questo caso, le derivate parziali di $\tilde{V}$ si scrivono in modo discretizzato, considerando però una differenza finita all'indietro (*backward difference*) per la derivata temporale:

$$\begin{aligned}
\frac{\partial \tilde{V}}{\partial \mathcal{T}}\big(n\Delta S,\,(m+1)\Delta T\big) &= \frac{\mathrm{v}_n^{m+1}-\mathrm{v}_n^{m}}{\Delta T} + \mathcal{O}(\Delta T)\,,\\
\frac{\partial \tilde{V}}{\partial S}\big(n\Delta S,\,(m+1)\Delta t\big) &= \frac{\mathrm{v}_{n+1}^{m+1}-\mathrm{v}_{n-1}^{m+1}}{2\Delta S} + \mathcal{O}\big((\Delta S)^2\big)\,,\\
\frac{\partial^2 \tilde{V}}{\partial S^2}\big(n\Delta S,\,(m+1)\Delta t\big) &= \frac{\mathrm{v}_{n+1}^{m+1}-2\mathrm{v}_n^{m+1}+\mathrm{v}_{n-1}^{m+1}}{(\Delta S)^2} + \mathcal{O}\big((\Delta S)^2\big)\,.
\end{aligned}$$

Sfruttando tali approssimazioni, l'equazione (12) diventa

$$\frac{\mathrm{v}_n^{m+1}-v_n^{m}}{\Delta t} - \frac{1}{2}\sigma^2 n^2\left(\mathrm{v}_{n+1}^{m+1}-2\mathrm{v}_n^{m+1}+\mathrm{v}_{n-1}^{m+1}\right) - rn\left(\mathrm{v}_{n+1}^{m+1}-\mathrm{v}_{n-1}^{m+1}\right) + r\mathrm{v}_n^{m+1} = 0\,, \tag{17}$$

con $n = 1, 2, ..., N-1$ e $m = 0, 1, ..., M-1$. Anche qui l'errore complessivo è di ordine $\mathcal{O}\big(\Delta t + (\Delta S)^2\big)$. Utilizzando le definizioni (16), dall'equazione (17) si ottiene

$$\mathrm{v}_n^{m} = \frac{1}{2}\left(\beta n - \alpha n^2\right)\mathrm{v}_{n-1}^{m+1} + \left(1+\beta+\alpha n^2\right)\mathrm{v}_n^{m+1} - \frac{1}{2}\left(\beta n + \alpha n^2\right)\mathrm{v}_{n+1}^{m+1}\,. \tag{18}$$

Al fine di riscrivere quest'ultima relazione in forma matriciale, si definiscono la matrice tridiagonale

$$A \equiv \begin{pmatrix}
a_1 & c_1 & 0 & \cdots & & \cdots & 0\\
b_2 & a_2 & c_2 & 0 & & & \vdots\\
0 & b_3 & a_3 & c_3 & 0 & & \\
0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots\\
\vdots & & \ddots & & & \ddots & 0\\
\vdots & & & \ddots & \ddots & \ddots & c_{N-2}\\
0 & \cdots & & \cdots & 0 & b_{N-1} & a_{N-1}
\end{pmatrix}$$

con

$$\begin{aligned}
a_n &\equiv 1+\beta+\alpha n^2, & n &= 1, 2, ..., N-1,\\
b_n &\equiv \frac{1}{2}\left(\beta n - \alpha n^2\right), & n &= 2, 3, ..., N-1,\\
c_n &\equiv -\frac{1}{2}\left(\beta n + \alpha n^2\right), & n &= 1, 2, ..., N-2,
\end{aligned}$$

e i vettori

$$\mathbf{v}^m \equiv \begin{pmatrix} \mathrm{v}_1^m \\ \mathrm{v}_2^m \\ \vdots \\ \mathrm{v}_{N-1}^m \end{pmatrix}, \quad \mathbf{h}^m \equiv \begin{pmatrix} \mathrm{v}_1^m - \frac{1}{2}\left(\beta - \alpha\right) \mathrm{v}_0^{m+1} \\ \mathrm{v}_2^m \\ \vdots \\ \mathrm{v}_{N-2}^m \\ \mathrm{v}_{N-1}^m + \frac{1}{2}\left(\beta(N-1) + \alpha(N-1)^2\right) \mathrm{v}_N^{m+1} \end{pmatrix}.$$

Con tali definizioni, la relazione (18) può essere riscritta sotto forma della seguente equazione matriciale:

$$A\mathbf{v}^{m+1} = \mathbf{h}^m, \quad m = 0, 1, ..., M-1, \tag{19}$$

dove l'incognita è $\mathbf{v}^{m+1}$.

Per risolvere l'equazione (19) è necessario invertire la matrice A. A tale scopo, si effettua la cosiddetta decomposizione LU, cioè si decompone A nel prodotto di una matrice triangolare inferiore L con una matrice triangolare superiore U:

$$A = LU\,,$$

con

$$L \equiv \begin{pmatrix} 1 & 0 & \cdots & & & \cdots & 0 \\ \tilde{b}_2 & 1 & 0 & & & & \vdots \\ 0 & \tilde{b}_3 & 1 & 0 & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & & \cdots & 0 & \tilde{b}_{N-1} & 1 \end{pmatrix}, \quad U \equiv \begin{pmatrix} \tilde{a}_1 & \tilde{c}_1 & 0 & \cdots & & \cdots & 0 \\ 0 & \tilde{a}_2 & \tilde{c}_2 & 0 & & & \vdots \\ 0 & 0 & \tilde{a}_3 & \tilde{c}_3 & 0 & & \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & & & \ddots & 0 \\ \vdots & & & & \ddots & \ddots & \tilde{c}_{N-2} \\ 0 & \cdots & & & \cdots & 0 & \tilde{a}_{N-1} \end{pmatrix},$$

dove $\tilde{a}_1 \equiv a_1$, $\tilde{c}_1 \equiv c_1$, e

$$\begin{aligned} \tilde{a}_n &\equiv a_n - \frac{b_n c_{n-1}}{\tilde{a}_{n-1}}\,, \qquad && n = 2, 3, ..., N-1, \\ \tilde{b}_n &\equiv \frac{b_n}{\tilde{a}_{n-1}}\,, && n = 2, 3, ..., N-1, \\ \tilde{c}_n &\equiv c_n\,, && n = 2, 3, ..., N-2. \end{aligned}$$

Dunque, ponendo $\mathbf{x}^m \equiv U\mathbf{v}^m$, l'equazione (19) diventa

$$L\mathbf{x}^{m+1} = \mathbf{h}^m\,,$$

la cui soluzione, per componenti, è data da

$$x_1^{m+1} = h_1^m\,, \quad x_n^{m+1} = h_n^m - \tilde{b}_n x_{n-1}^{m+1} \quad \text{per } n = 2, 3, ..., N-1.$$

A questo punto, rimane solo da risolvere l'equazione

$$U\mathbf{v}^{m+1} = \mathbf{x}^{m+1}\,,$$

nell'incognita $\mathbf{v}^{m+1}$, che ha soluzione, per componenti,

$$\mathrm{v}_{N-1}^{m+1} = \frac{x_{N-1}^{m+1}}{\tilde{a}_{N-1}}\,, \qquad \mathrm{v}_n^{m+1} = \frac{1}{\tilde{a}_n}\left(x_n^{m+1} - \tilde{c}_n \mathrm{v}_{n+1}^{m+1}\right) \qquad \text{per } n = 1,\, 2, ...,\, N-2.$$

I passaggi effettuati permettono di ricavare una stima di v_n^{m+1} $\forall\, n \in [0,\, N] \cap \mathbb{N}$, dati v_n^m $\forall\, n \in [0,\, N] \cap \mathbb{N}$ e condizioni al bordo che fissino v_0^m e v_N^m $\forall\, m \in [0,\, M] \cap \mathbb{N}$. Procedendo iterativamente a partire dalla condizione iniziale (13) è dunque possibile, anche in questo caso, ricavare un'approssimazione numerica della funzione $\tilde{V}$. Inoltre, si può dimostrare come tale metodo implicito risulti stabile per qualsiasi scelta di N e M.

2.3 Implementazione degli algoritmi

I due metodi descritti nelle sezioni precedenti sono stati implementati in un programma in linguaggio C. La struttura del codice è molto semplice: la funzione principale *main*, per ogni coppia di valori $(N,\, M)$ scelta e ciascuno dei due algoritmi utilizzati, richiama ciclicamente in modo alternativo le funzioni *passo_ esplicito* e *passo_ implicito*, incaricate rispettivamente di eseguire un passo dei metodi esplicito e implicito. Entrambe le funzioni, data la funzione discretizzata $\tilde{V}(n\Delta S,\, m^*\Delta t)$ (con $n = 0,\, 1, ...,\, N$, m^* fissato) memorizzata in un *array* dichiarato come variabile globale, calcolano i valori $\tilde{V}(n\Delta S,\, (m^*+1)\Delta t)$, aggiornando conseguentemente l'*array*. I valori iniziali sono dati dalla condizione (13); i valori $\tilde{V}(0,\, m\Delta t)$ e $\tilde{V}(N\Delta S,\, m\Delta t)$ per ogni $m = 0,\, 1, ...,\, M$ sono fissati dalle condizioni al bordo (14).

I risultati ottenuti vengono salvati in un file di testo per la successiva elaborazione mediante l'ambiente MATLAB, con cui vengono tracciati alcuni grafici significativi. Il codice completo, compreso lo script MATLAB, è riportato in Appendice A.1.

3 Discussione dei risultati e confronto con la teoria

Di seguito si riportano alcuni grafici significativi rappresentanti le soluzioni numeriche dell'equazione BS (4) ricavate mediante i due algoritmi analizzati.
Nelle tre figure seguenti sono rappresentati i grafici tridimensionali della funzione V in funzione del tempo t e del prezzo dell'*asset* sottostante S. Il primo grafico (Fig.1) rappresenta la soluzione analitica data dalla relazione (11), mentre nel secondo (Fig.2) e nel terzo (Fig.3) sono rappresentate le soluzioni numeriche ricavate, rispettivamente, con il metodo esplicito ed implicito per $N = 1000$ e $M = 41000$.

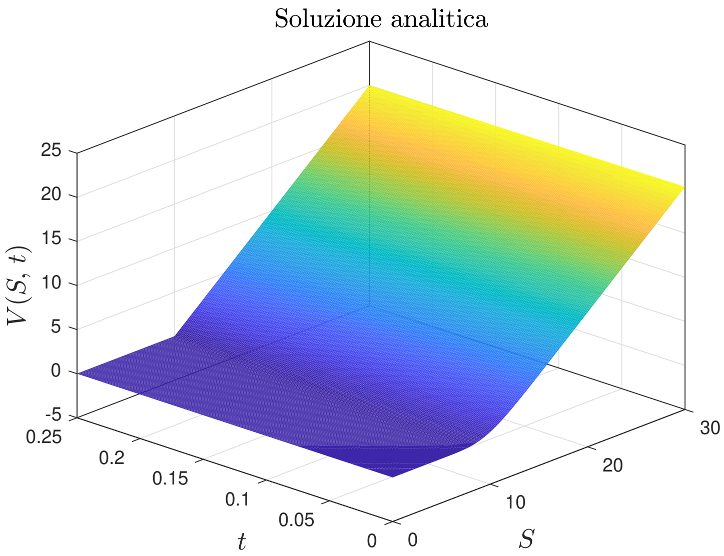

Figura 1: È rappresentata la soluzione analitica (11) dell'equazione BS (4).

Per permettere un miglior confronto tra le soluzioni approssimate e la soluzione esatta, conviene fissare la variabile temporale.

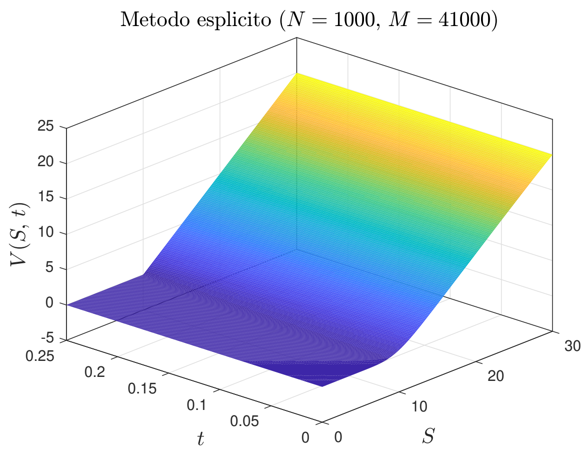

Figura 2: È rappresentata la soluzione numerica dell'equazione BS (4) ricavata con il metodo esplicito con $N = 1000$, $M = 41000$.

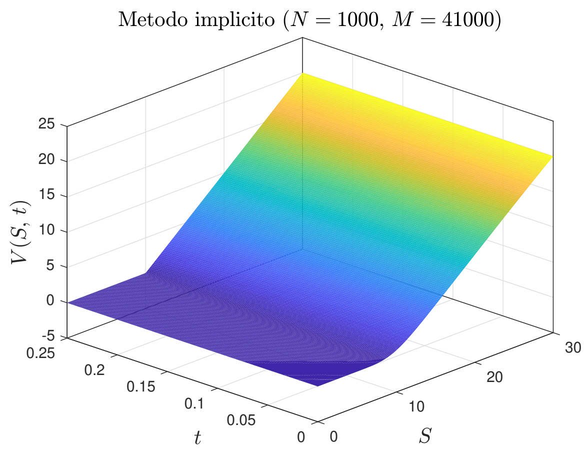

Figura 3: È rappresentata la soluzione numerica dell'equazione BS (4) ricavata con il metodo implicito con $N = 1000$, $M = 41000$.

In particolare, nei grafici riportati nelle figure 4 e 5 viene fissato $t = 0$. Si osserva come per valori di $S \lesssim 20$, le soluzioni numeriche presentino un buon accordo con la soluzione analitica. Per alti valori di S, invece, le soluzioni numeriche, indipendentemente dal metodo utilizzato per ottenerle, risultano sistematicamente inferiori rispetto alla soluzione esatta. Tale discrepanza è dovuta essenzialmente alla scelta di fissare un intervallo limitato $[0, S_{max}]$ su cui può variare S e alla conseguente sostituzione della condizione

$$V(S,t) \to S \quad \text{per } S \to \infty$$

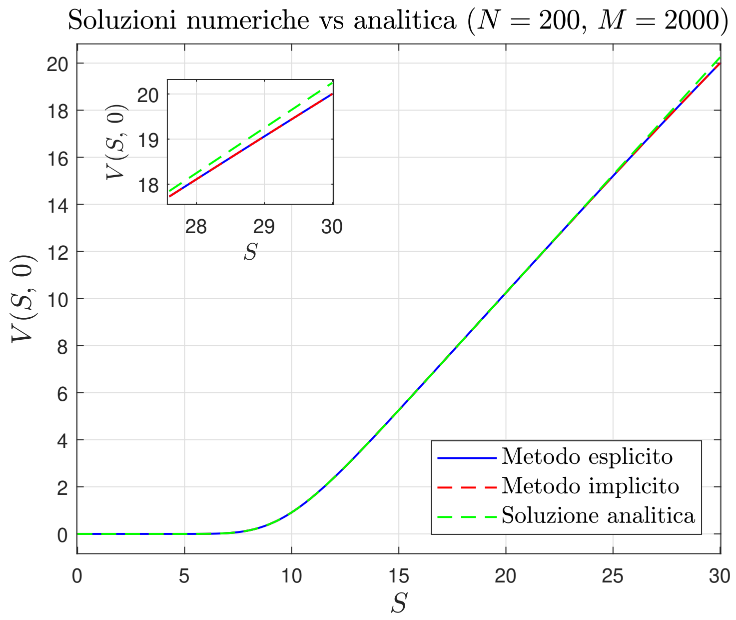

Figura 4: Sono poste a confronto le soluzioni numeriche (ottenute con $N = 200$, $M = 2000$) e la soluzione analitica (11) dell'equazione BS (4). Il grafico è realizzato a tempo $t = 0$ fissato, al variare di S.

con la condizione al bordo

$$V(S_{max}, t) = S_{max} - E$$

per ogni $t \in [0, T]$. Tale deviazione sistematica delle soluzioni numeriche da quella

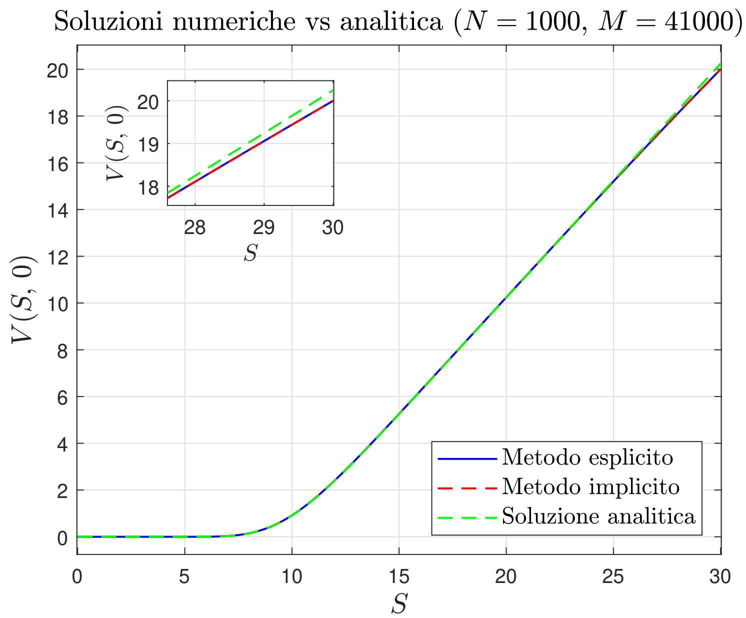

Figura 5: Come in figura 4, ma qui le soluzioni sono ottenute con $N = 1000$, $M = 41000$.

analitica è resa ancora più evidente dal grafico in figura 6, che rappresenta l'errore assoluto commesso nell'approssimazione numerica per ciascuno dei metodi implementati. Si osserva che gli errori ottenuti con i due metodi risultano dello stesso ordine di grandezza a parità di N e M scelti. Inoltre, come è lecito aspettarsi, le soluzioni numeriche, per entrambi i metodi utilizzati, risultano tanto migliori quanto

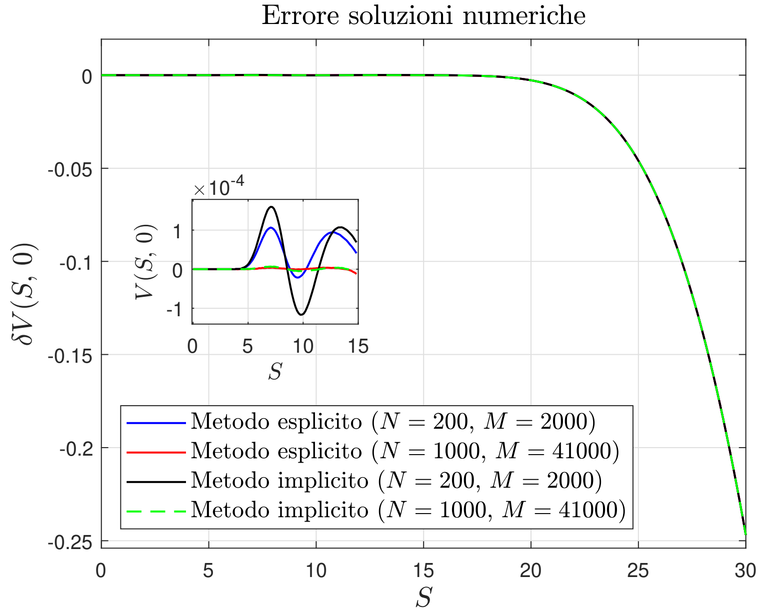

Figura 6: Sono rappresentati gli errori $\delta V(S, 0) \equiv V_{num}(S, 0) - V_{es}(S, 0)$ (con V_{num} soluzione numerica, V_{es} soluzione esatta) al tempo $t = 0$ fissato, al variare di S, per le quattro soluzioni numeriche ricavate.

più sono piccole le ampiezze Δt e ΔS dei sottointervalli in cui sono suddivisi gli intervalli $[0, T]$ e $[0, S_{max}]$ rispettivamente.

Infine, in tabella 1 è riportato $V(S, 0)$ per 4 diversi valori di S, ottenuto con ciascuno dei metodi numerici implementati, e calcolato mediante la formula analitica (11). Per $S \leq 20$, i risultati numerici sono in accordo con la soluzione analitica almeno entro la terza cifra decimale.

Tabella 1: Sono riportante le soluzioni numeriche e la soluzione analitica (11) dell'equazione BS (4), al tempo $t = 0$ fissato, per alcuni valori del prezzo S dell'*asset*. Si indicano con 'A' le soluzioni ottenute con $N = 200$, $M = 2000$, con 'B' quelle ottenute con $N = 1000$, $M = 41000$.

S	Metodo esplicito		Metodo implicito		Soluzione analitica
	A	B	A	B	
6	0.00385	0.00380	0.00388	0.00380	0.003795
12	2.41450	2.41441	2.41447	2.41441	2.414410
18	8.24719	8.24718	8.24719	8.24718	8.247704
24	14.21760	14.21759	14.21757	14.21759	14.24690

Per completezza, nella coppia di grafici seguente (Fig.7 e Fig.8) vengono raffigurate le soluzioni analitiche e numeriche $V(S, t)$, per alcuni valori di S fissati. Sia per il metodo esplicito che per quello implicito vengono rappresentati i risultati ottenuti con $N = 1000$ e $M = 41000$.

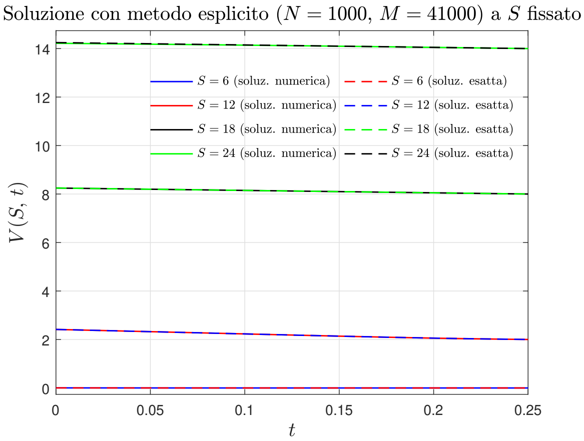

Figura 7: Confronto tra soluzione numerica e analitica dell'equazione BS (4) per alcuni valori di S fissati, ricavata con il metodo esplicito con $N = 1000$, $M = 41000$.

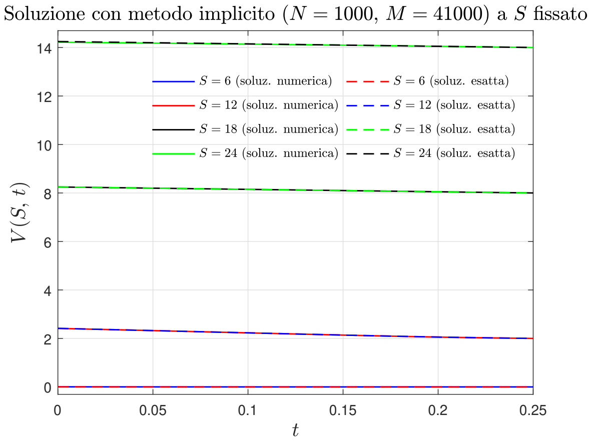

Figura 8: Confronto tra soluzione numerica e analitica dell'equazione BS (4) per alcuni valori di S fissati, ricavata con il metodo implicito con $N = 1000$, $M = 41000$.

Si nota che, coerentemente con quanto osservato in precedenza, solo per $S = 24$ si riescono ad apprezzare graficamente (lievi) divergenze tra risultati numerici e analitici.

A Appendice

A.1 Codice

Di seguito sono riportati codice e script utilizzati rispettivamente per la parte di risoluzione numerica e per la parte grafica.

A.1.1 Codice in linguaggio C

```c
#include <stdio.h>
#include <stdlib.h>

#define MAX(a, b) ((a) > (b) ? (a) : (b))

// Costanti
#define T 0.25 // Payoff time
#define E 10.0 // Strike price
#define r 0.1 // Tasso di interesse
#define sigma 0.4 // Volatilita' del prezzo
#define S_max 3*E // Prezzo massimo
#define NUM_PRINT_n 4 // Numero di valori di S per cui e' stampato a video V (a tempo fissato)
#define INT_PRINT_m 500 // Numero di passi tra una stampa a video e l'altra
#define NUM_SAVE_S 200 // (Numero-1) di valori di S per cui viene salvato V su file
#define NUM_SAVE_t 200 // (Numero-1) di valori di t per cui viene salvato V su file
#define LUN 2
const int N[LUN] = {200, 1000}; // Numero di intervalli in cui e' diviso [0, S_max]
const int M[LUN] = {2000, 41000}; // Numero di intervalli in cui e' diviso [0, T]

// File per salvataggio dati
const char nome_file[] = "C:\\Users\\Admin\\Documents\\MATLAB\\esercizio3_"; // Prima parte nome file
const char estens[] = ".txt"; // Estensione file

// Prototipi funzioni
void passo_esplicito(short); // Esegue un passo del metodo esplicito
void passo_implicito(short); // Esegue un passo del metodo implicito

// Variabili globali
```

```
double DeltaS; // Ampiezza intervalli in cui e' diviso [0,
    S_max]
double DeltaT; // Ampiezza intervalli in cui e' diviso [0, T]
double alpha; // sigma^2 * DeltaT
double beta; // r * DeltaT
double *V; // Puntatore a vettore che conterra' i valori V(n *
     DeltaS, m * DeltaT) (con m \in {0, 1, ..., M} fissato) per
     n = 0, 1, ..., N
double t; // Tempo (trasformato secondo t' = T - t)


int main(){
     FILE *file;

     short i, k;
     int j, m;
     int n_print[NUM_PRINT_n], n_save[NUM_SAVE_S+1], int_save_m
         ;
     char str[56];

     printf("RISOLUZIONE NUMERICA DELL'EQUAZIONE DI BLACK-
         SCHOLES\n\n");

     for(i = 0; i < LUN; i++){
          for(k = 0; k < 2; k++){ // k = 0: metodo esplicito, k
              = 1: metodo implicito
               /* File per salvataggio dati
                    '1': metodo esplicito, '2': metodo implicito;
                         la lettera identifica la coppia (N, M) */
               sprintf(str, "%s%d%c%s", nome_file, k+1, 'a'+i,
                    estens);
               file = fopen(str, "w");

               // Calcolo parametri
               DeltaS = S_max / N[i];
               DeltaT = T / M[i];
               alpha = sigma * sigma * DeltaT;
               beta = r * DeltaT;

               // Istanzio vettore V
               if(k == 0){
                    if(i == 0)
                         V = malloc(sizeof(double) * (N[i] + 1));
                    else
                         V = realloc(V, sizeof(double) * (N[i] + 1)
                              );
```

```
        }

        // Condizioni al bordo e condizioni iniziali
        V[0] = 0;
        V[N[i]] = S_max - E;
        for(j = 1; j < N[i]; j++)
            V[j] = MAX(j * DeltaS - E, 0);
        t = 0;

        // Stampo a video info e condizioni iniziali
        if(k == 0){
            printf("METODO ESPLICITO\n");
            for(j = 0; j < NUM_PRINT_n; j++)
                n_print[j] = N[i] / (NUM_PRINT_n + 1) * (j
                    + 1);

            int_save_m = M[i] / NUM_SAVE_t;
            for(j = 0; j <= NUM_SAVE_S; j++)
                n_save[j] = N[i] / NUM_SAVE_S * j;
        }
        else{
            printf("METODO IMPLICITO\n");
        }
        printf("N = %d, M = %d\n\n", N[i], M[i]);
        printf("Passo\tt");
        for(j = 0; j < NUM_PRINT_n; j++)
            printf("\t\tV(%1.0f,t)", n_print[j] * DeltaS);
        printf("\n0\t%.4f", t);
        for(j = 0; j < NUM_PRINT_n; j++)
            printf("\t\t%.5f", V[n_print[j]]);
        printf("\n");

        // Salvo su file condizioni iniziali
        fprintf(file, "0");
        for(j = 0; j <= NUM_SAVE_S; j++)
            fprintf(file, " %f", n_save[j] * DeltaS);
        fprintf(file, "\n");
        fprintf(file, "%f", t);
        for(j = 0; j <= NUM_SAVE_S; j++)
            fprintf(file, " %f", V[n_save[j]]);
        fprintf(file, "\n");

        // Evoluzione temporale
        for(m = 1; m <= M[i]; m++){
            if(k == 0)
```

```
                    passo_esplicito(i); // Passo algoritmo esplicito
                else
                    passo_implicito(i); // Passo algoritmo implicito

                // Stampo a video passo
                if(m % INT_PRINT_m == 0){
                    printf("%d\t%.5f", m, t);
                    for(j = 0; j < NUM_PRINT_n; j++)
                        printf("\t\t%.5f", V[n_print[j]]);
                    printf("\n");
                }

                // Salvo su file passo
                if(m % int_save_m == 0){
                    fprintf(file, "%f", t);
                    for(j = 0; j <= NUM_SAVE_S; j++)
                        fprintf(file, " %f", V[n_save[j]]);
                    fprintf(file, "\n");
                }
            }

            printf("\n----------------------------\n\n");

            fclose(file);
        }
    }

    return 0;
}


// Passo del metodo esplicito
void passo_esplicito(short index){
    int n;
    double V_prec = V[0], temp;

    for(n = 1; n < N[index]; n++){
        temp = V[n];
        V[n] = 0.5 * (alpha * n * n - beta * n) * V_prec + (1 - alpha * n * n - beta) * V[n] + 0.5 * (alpha * n * n + beta * n) * V[n+1];
        V_prec = temp;
    }

```

```
    t += DeltaT;
}


/* Passo del metodo implicito:
   risoluzione dell'equazione matriciale A * v = h (v
      incognito) */
void passo_implicito(short index){
    int n;
    double a[N[index]], b[N[index]], c[N[index]], x[N[index]];

    // A = LU decomposition
    a[1] = 1 + beta + alpha;
    c[1] = -0.5 * (beta + alpha);
    for(n = 2; n < N[index]; n++){
        a[n] = 1 + beta + alpha * n * n + 0.25 * (beta * n -
           alpha * n * n) * (beta * (n - 1) + alpha * (n - 1)
           * (n - 1)) / a[n-1];
        b[n] = 0.5 * (beta * n - alpha * n * n) / a[n-1];
        c[n] = -0.5 * (beta * n + alpha * n * n);
    }

    // Soluzione di L * x = h (x = U * v incognito)
    x[1] = V[1] - 0.5 * (beta - alpha) * V[0];
    for(n = 2; n < N[index]-1; n++)
        x[n] = V[n] - b[n] * x[n-1];
    x[N[index]-1] = V[N[index]-1] + 0.5 * (N[index] - 1) * (
       beta + alpha * (N[index] - 1)) * V[N[index]] - b[N[
       index]-1] * x[N[index]-2];

    // Soluzione di U * v = x
    V[N[index]-1] = x[N[index]-1] / a[N[index]-1];
    for(n = N[index]-2; n > 0; n--)
        V[n] = (x[n] - c[n] * V[n+1]) / a[n];

    // Incremento del tempo
    t += DeltaT;
}
```

A.1.2 Script MATLAB per i grafici

```matlab
close all;
format long;

colori = 'brkg';
colori2 = 'rbgk';
etichette = ['a', 'b'];
metodo = {'esplicito'; 'implicito'};
N = [200, 1000];
M = [2000, 41000];
E = 10;
T = 0.25;
r = 0.1;
sigma = 0.4;
S_max = 3 * E;
S_print = [6, 12, 18, 24];


%% Soluzione analitica: calcolo e grafico
S_e = (0:S_max/200:S_max)';
t_e = (0:T/200:T)';
V_e = zeros(length(t_e), length(S_e));

for k = 1:length(t_e)
    for h = 1:length(S_e)
        V_e(k, h) = S_e(h) * PHI((log(S_e(h) / E) + (r + sigma^2 / 2) * (T - t_e(k)))...
            / (sigma * sqrt(T - t_e(k)))) - E * exp(-r * (T - t_e(k))) * PHI((log(S_e(h) / E)...
            + (r - sigma^2 / 2) * (T - t_e(k))) / (sigma * sqrt(T - t_e(k))));
        if isnan(V_e(k, h))
            V_e(k, h) = 0;
        end
    end
end

% Grafico 3D
figure();
grid on;
hold on;
box on;
pl = surf(S_e, t_e, V_e);
set(pl, 'LineStyle', 'none');
tt = title('Soluzione analitica', 'Interpreter', 'latex');
xx = xlabel('$S$', 'Interpreter', 'latex');
```

```
yy = ylabel('$t$', 'Interpreter', 'latex');
zz = zlabel('$V(S,\,t)$', 'Interpreter', 'latex');
set(xx, 'FontSize', 14);
set(yy, 'FontSize', 14);
set(zz, 'FontSize', 14);
set(tt, 'FontSize', 14);

% Grafico a S fissato per vari valori di S
figure();
grid on;
hold on;
box on;
labels = {};
for i = 1:length(S_print)
    pl = plot(t_e, V_e(:, S_e == S_print(i)));
    set(pl, 'LineWidth', 1);
    labels = [labels, ['$S = ' num2str(S_print(i)) '$']];
end
tt = title('Soluzione analitica a $S$ fissato', 'Interpreter',
    'latex');
xx = xlabel('$t$', 'Interpreter', 'latex');
yy = ylabel('$V(S,\,t)$', 'Interpreter', 'latex');
ll = legend(labels, 'Interpreter', 'latex');
set(xx, 'FontSize', 14);
set(yy, 'FontSize', 14);
set(tt, 'FontSize', 14);
set(ll, 'FontSize', 12);


%% Soluzioni numeriche: upload dati e grafici
for j = 1:2
    for i = 1:length(etichette)
        %% Carico dati
        dati = importdata(['esercizio3_' num2str(j) etichette(
           i) '.txt']);
        S = (dati(1, 2:end))';
        V = dati(2:end, 2:end);
        t = dati(2:end, 1);

        t = T - t; % Cambio variabile

        %% Grafici
        % Grafico 3D
        figure();
        grid on;
        hold on;
```

```
        box on;
        pl = surf(S, t, V);
        set(pl, 'LineStyle', 'none');
        zlim([-5, 25]);
        tt = title(['Metodo ' char(metodo(j)) ' $(N = ' num2str(N(i))...
            '$, $M = ' num2str(M(i)) ')$'], 'Interpreter', 'latex');
        xx = xlabel('$S$', 'Interpreter', 'latex');
        yy = ylabel('$t$', 'Interpreter', 'latex');
        zz = zlabel('$V(S,\,t)$', 'Interpreter', 'latex');
        set(xx, 'FontSize', 14);
        set(yy, 'FontSize', 14);
        set(zz, 'FontSize', 14);
        set(tt, 'FontSize', 14);

        % Grafico a S fissato per vari valori di S e confronto con sol. analitica
        figure();
        grid on;
        hold on;
        box on;
        labels = {};
        for k = 1:length(S_print)
            pl = plot(t, V(:, S == S_print(k)), colori(k));
            set(pl, 'LineWidth', 1);
            labels = [labels, ['$S = ' num2str(S_print(k)) '$ (soluz. numerica)']];
        end
        for k = 1:length(S_print)
            pl = plot(t_e, V_e(:, S_e == S_print(k)), ['--' colori2(k)]);
            set(pl, 'LineWidth', 1);
            labels = [labels, ['$S = ' num2str(S_print(k)) '$ (soluz. esatta)']];
        end
        tt = title(['Soluzione con metodo ' char(metodo(j)) ' $(N = ' num2str(N(i))...
            '$, $M = ' num2str(M(i)) ')$ a $S$ fissato'], 'Interpreter', 'latex');
        xx = xlabel('$t$', 'Interpreter', 'latex');
        yy = ylabel('$V(S,\,t)$', 'Interpreter', 'latex');
        ll = columnlegend(2, labels, 'Interpreter', 'latex');
        set(xx, 'FontSize', 14);
        set(yy, 'FontSize', 14);
        set(tt, 'FontSize', 14);
```

```
        set(ll, 'FontSize', 14);

        % Confronto sol. analitica vs numeriche a t=0 fissato
        if j == 1
            fig(i) = figure();
            ax_fig(i) = gca();
            grid on;
            hold on;
            box on;
            pl = plot(S, V(end, :), colori(j));
            set(pl, 'LineWidth', 1);

            % Zoom
            ax_zoom(i) = axes('Position', [.24 .67 .20 .20]);
            grid on;
            hold on;
            box on;
            pl = plot(S(185:end), V(end, 185:end), colori(j));
            set(pl, 'LineWidth', 1);
        else
            figure(fig(i));
            axes(ax_fig(i));
            pl = plot(S, V(end, :), ['--' colori(j)]);
            set(pl, 'LineWidth', 1);
            pl = plot(S_e, V_e(1, :), '--g');
            set(pl, 'LineWidth', 1);
            tt = title(['Soluzioni numeriche vs analitica $(N
                = '...
               num2str(N(i)) '$, $M = ' num2str(M(i)) ')$'],
                  'Interpreter', 'latex');
            xx = xlabel('$S$', 'Interpreter', 'latex');
            yy = ylabel('$V(S,\,0)$', 'Interpreter', 'latex');
            ll = legend('Metodo esplicito', 'Metodo implicito'
                , 'Soluzione analitica');
            set(xx, 'FontSize', 14);
            set(yy, 'FontSize', 14);
            set(tt, 'FontSize', 14);
            set(ll, 'FontSize', 12);
            set(ll, 'Location', 'SouthEast');
            set(ll, 'Interpreter', 'latex');

            % Zoom
            axes(ax_zoom(i));
            pl = plot(S(185:end), V(end, 185:end), ['--'
                colori(j)]);
            set(pl, 'LineWidth', 1);
```

```
                pl = plot(S_e(185:end), V_e(1, 185:end), '--g');
                set(pl, 'LineWidth', 1);
                xx = xlabel('$S$', 'Interpreter', 'latex');
                yy = ylabel('$V(S,\,0)$', 'Interpreter', 'latex');
                set(xx, 'FontSize', 12);
                set(yy, 'FontSize', 12);
            end

            % Grafico errore
            if j == 1 && i == 1
                fig_err = figure();
                ax_err = gca();
                grid on;
                hold on;
                box on;
                ax_zoom_err = axes('Position', [.24 .47 .20 .20]);
                grid on;
                hold on;
                box on;
                num = 1;
                lab = {};
            end
            figure(fig_err);
            if num ~= 4
                axes(ax_err);
                pl = plot(S, V(end, :) - V_e(1, :), colori(num));
                axes(ax_zoom_err);
                pl2 = plot(S(1:100), V(end, 1:100) - V_e(1, 1:100)
                    , colori(num));
            else
                axes(ax_err);
                pl = plot(S, V(end, :) - V_e(1, :), ['--' colori(
                   num)]);
                axes(ax_zoom_err);
                pl2 = plot(S(1:100), V(end, 1:100) - V_e(1, 1:100)
                    , ['--' colori(num)]);
            end
            set(pl, 'LineWidth', 1);
            set(pl2, 'LineWidth', 1);
            lab = [lab, ['Metodo ' char(metodo(j)) ' $(N = '...
                    num2str(N(i)) '$, $M = ' num2str(M(i)) ')$']];
            num = num + 1;
        end
    end

    figure(fig_err);
```

```
axes(ax_err);
tt = title('Errore soluzioni numeriche', 'Interpreter', 'latex
    ');
xx = xlabel('$S$', 'Interpreter', 'latex');
yy = ylabel('$\delta V(S,\,0)$', 'Interpreter', 'latex');
ll = legend(lab, 'Interpreter', 'latex');
set(xx, 'FontSize', 14);
set(yy, 'FontSize', 14);
set(tt, 'FontSize', 14);
set(ll, 'FontSize', 12);
set(ll, 'Location', 'SouthWest');

axes(ax_zoom_err);
xx = xlabel('$S$', 'Interpreter', 'latex');
yy = ylabel('$V(S,\,0)$', 'Interpreter', 'latex');
set(xx, 'FontSize', 12);
set(yy, 'FontSize', 12);


%% Funzione per calcolo soluzione analitica
function [phi] = PHI(d)
    fun = @(x) exp(-x.^2./2);
    phi = 1 / sqrt(2 * pi) * integral(fun, -Inf, d);
end
```