



# Metodo a elementi finiti per il pricing di opzioni multi-asset con modelli di Lévy

Progetto di Programmazione Avanzata per il Calcolo Scientifico

Nahuel Foresta    Giorgio G. Re

Dipartimento di Matematica  
**Politecnico di Milano**

15 settembre 2014

# Indice

- 1 Introduzione
- 2 Il problema
- 3 Struttura del codice
- 4 Risultati
- 5 Conclusioni

# Il progetto

## Scopo

Lo scopo di questo progetto è creare una piccola libreria per il *pricing* di derivati finanziari con il metodo degli elementi finiti, utilizzando la libreria deal.ii. L'idea è che l'utente possa sia utilizzare gli strumenti presenti, sia aggiungerne altri nel caso di bisogno, con grande facilità (ereditarietà).

# Il progetto

## Scopo

Lo scopo di questo progetto è creare una piccola libreria per il *pricing* di derivati finanziari con il metodo degli elementi finiti, utilizzando la libreria deal.ii. L'idea è che l'utente possa sia utilizzare gli strumenti presenti, sia aggiungerne altri nel caso di bisogno, con grande facilità (ereditarietà).

## Motivazioni

La procedura più diffusa in finanza è l'utilizzo delle differenze finite. Gli elementi finiti, a fronte di una maggiore difficoltà implementativa, risultano essere più vantaggiosi.

# Indice

- 1 Introduzione
- 2 Il problema**
- 3 Struttura del codice
- 4 Risultati
- 5 Conclusioni

# L'equazione da risolvere

$$\begin{aligned} \frac{\partial C}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC + \\ + \int_{\mathbb{R}} \left( C(t, Se^y) - C(t, S) - S(e^y - 1) \frac{\partial C}{\partial S}(t, S) \right) \nu(dy) = 0 \end{aligned}$$

con opportune condizioni al contorno e condizione finale.

# L'equazione da risolvere

$$\frac{\partial C}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC + \int_{\mathbb{R}} \left( C(t, Se^y) - C(t, S) - S(e^y - 1) \frac{\partial C}{\partial S}(t, S) \right) \nu(dy) = 0$$

con opportune condizioni al contorno e condizione finale.

Possiamo scomporre il problema in due parti:

- la parte differenziale, trattata in modo usuale con l'aiuto della libreria deal.ii

# L'equazione da risolvere

$$\frac{\partial C}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC + \int_{\mathbb{R}} \left( C(t, Se^y) - C(t, S) - S(e^y - 1) \frac{\partial C}{\partial S}(t, S) \right) \nu(dy) = 0$$

con opportune condizioni al contorno e condizione finale.

Possiamo scomporre il problema in due parti:

- la parte differenziale, trattata in modo usuale con l'aiuto della libreria deal.ii
- la parte integrale, che necessita di un trattamento speciale



# L'equazione da risolvere

$$\frac{\partial C}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC + \int_{\mathbb{R}} \left( C(t, Se^y) - C(t, S) - S(e^y - 1) \frac{\partial C}{\partial S}(t, S) \right) \nu(dy) = 0$$

con opportune condizioni al contorno e condizione finale.

Possiamo scomporre il problema in due parti:

- la parte differenziale, trattata in modo usuale con l'aiuto della libreria deal.ii
- la parte integrale, che necessita di un trattamento speciale ed è separabile in due parti.

# L'equazione da risolvere

$$\frac{\partial C}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC + \int_{\mathbb{R}} \left( C(t, Se^y) - C(t, S) - S(e^y - 1) \frac{\partial C}{\partial S}(t, S) \right) \nu(dy) = 0$$

con opportune condizioni al contorno e condizione finale.

Possiamo scomporre il problema in due parti:

- la parte differenziale, trattata in modo usuale con l'aiuto della libreria deal.ii
- la parte integrale, che necessita di un trattamento speciale ed è separabile in due parti.

Trasformazioni *price* e *log-price*,  $x = \log(S/s_0)$ .

# L'equazione da risolvere

$$\frac{\partial u}{\partial t} + \left(r - \frac{\sigma^2}{2}\right) \frac{\partial u}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 u}{\partial x^2} - ru + \int_{\mathbb{R}} \left(u(t, x+y) - u(t, x) - (e^y - 1) \frac{\partial u}{\partial x}\right) \nu(dy) = 0$$

con opportune condizioni al contorno e condizione finale.

Possiamo scomporre il problema in due parti:

- la parte differenziale, trattata in modo usuale con l'aiuto della libreria deal.ii
- la parte integrale, che necessita di un trattamento speciale ed è separabile in due parti.

Trasformazioni *price* e *log-price*,  $x = \log(S/S_0)$ .

# Il problema con l'ostacolo: Opzioni Americane

Nelle Opzioni Americane:

$$P(S, t) \geq \max(K - S, 0) \quad \forall t \in [0, T]$$

# Il problema con l'ostacolo: Opzioni Americane

Nelle Opzioni Americane:

$$P(S, t) \geq \max(K - S, 0) \quad \forall t \in [0, T]$$

L'equazione diventa:

$$\begin{cases} \frac{\partial P}{\partial t} + rS \frac{\partial P}{\partial S} + \frac{1}{2} S^2 \frac{\partial^2 P}{\partial S^2} \\ + \int_{\mathbb{R}} \left( P(t, Se^y) - P(t, S) - S(e^y - 1) \frac{\partial P}{\partial S}(t, S) \right) \nu(dy) \leq rP, \\ P(S, t) \geq \max(K - S, 0) \end{cases}$$

# Scomposizione della parte integrale

Definendo nel modo seguente le quantità

$$\hat{\alpha} = \int_{\mathbb{R}} (e^y - 1) \nu(y) dy$$
$$\hat{\lambda} = \int_{\mathbb{R}} \nu(y) dy$$

l'equazione diventa

$$\frac{\partial C}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 C}{\partial S^2} + (r - \hat{\alpha}) S \frac{\partial C}{\partial S} - (r + \hat{\lambda}) C + \int_{\mathbb{R}} C(t, Se^y) \nu(y) dy = 0$$

# Scomposizione della parte integrale

Analogamente per la trasformazione *log-price* si ha

$$\hat{\alpha} = \int_{\mathbb{R}} (e^y - 1) \nu(y) dy$$

$$\hat{\lambda} = \int_{\mathbb{R}} \nu(y) dy$$

con rispettiva equazione

$$\frac{\partial u}{\partial t} + \left( r - \frac{\sigma^2}{2} - \hat{\alpha} \right) \frac{\partial u}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 u}{\partial x^2} - (r + \hat{\lambda})u + \int_{\mathbb{R}} u(t, x + y) \nu(y) dy = 0$$

# In due dimensioni

Con la trasformazione *price*

$$\begin{aligned}
 & \frac{\partial C}{\partial t} + (r - \hat{\alpha}_1)S_1 \frac{\partial C}{\partial S_1} + (r - \hat{\alpha}_2)S_2 \frac{\partial C}{\partial S_2} + \frac{\sigma_1^2}{2}S_1^2 \frac{\partial^2 C}{\partial S_1^2} + \frac{\sigma_2^2}{2}S_2^2 \frac{\partial^2 C}{\partial S_2^2} \\
 & \quad + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 C}{\partial S_1 \partial S_2} - (r + \lambda_1 + \lambda_2)C \\
 & \quad + \int_{\mathbb{R}} C(t, S_1 e^y, S_2) \nu_1(y) dy + \int_{\mathbb{R}} C(t, S_1, S_2 e^y) \nu_2(y) dy = 0
 \end{aligned}$$



# In due dimensioni

Con la trasformazione *log-price*

$$\begin{aligned}
 \frac{\partial u}{\partial t} + \frac{\sigma_1^2}{2} \frac{\partial^2 u}{\partial x_1^2} + \frac{\sigma_2^2}{2} \frac{\partial^2 u}{\partial x_2^2} + \rho \sigma_1 \sigma_2 \frac{\partial^2 u}{\partial x_1 \partial x_2} + \left( r - \frac{\sigma_1^2}{2} - \hat{\alpha}_1 \right) \frac{\partial u}{\partial x_1} \\
 + \left( r - \frac{\sigma_2^2}{2} - \hat{\alpha}_2 \right) \frac{\partial u}{\partial x_2} - (r + \hat{\lambda}_1 + \hat{\lambda}_2) u \\
 + \int_{\mathbb{R}} u(t, x_1 + y, x_2) \nu_1(y) dy + \int_{\mathbb{R}} u(t, x_1, x_2 + y) \nu_2(y) dy = 0
 \end{aligned}$$

# Discretizzazione

Data una griglia con nodi  $S_i$

- Per la parte differenziale, si scrive la formulazione variazionale e la discretizzazione nel modo usuale

# Discretizzazione

Data una griglia con nodi  $S_i$

- Per la parte differenziale, si scrive la formulazione variazionale e la discretizzazione nel modo usuale
- Per la parte integrale, si calcola il valore della parte integrale relativa al nodo  $S_i$

$$J^1(S_i) = \int_{\mathbb{R}} C(t, S_1 e^y, S_2) \nu_1(y) dy$$

ottenendo una vettore  $\mathbf{J}$  funzione di  $S_i$ . Tale funzione va poi scritta come elemento dello spazio a elementi finiti

# Discretizzazione

Data una griglia con nodi  $S_i$

- Per la parte differenziale, si scrive la formulazione variazionale e la discretizzazione nel modo usuale
- Per la parte integrale, si calcola il valore della parte integrale relativa al nodo  $S_i$

$$J^1(S_i) = \int_{\mathbb{R}} C(t, S_1 e^y, S_2) \nu_1(y) dy$$

ottenendo una vettore  $\mathbf{J}$  funzione di  $S_i$ . Tale funzione va poi scritta come elemento dello spazio a elementi finiti

- Per la discretizzazione temporale, viene applicato uno schema di Eulero Implicito, ma la parte integrale viene trattata in modo esplicito. Lo schema è stabile se  $1/\Delta t < \lambda$ .

# Discretizzazione

Data una griglia con nodi  $S_i$

Otteniamo dunque il seguente schema, con  $\mathbf{C}_h^k$  vettore componenti soluzione al tempo  $k$

$$M_1 \mathbf{C}_h^k = M_2 \mathbf{C}_h^{k+1} + M\mathbf{J}^{1,k} + M\mathbf{J}^{2,k}$$

# Discretizzazione

Data una griglia con nodi  $S_i$

Otteniamo dunque il seguente schema, con  $\mathbf{C}_h^k$  vettore componenti soluzione al tempo  $k$

$$M_1 \mathbf{C}_h^k = M_2 \mathbf{C}_h^{k+1} + M\mathbf{J}^{1,k} + M\mathbf{J}^{2,k}$$

Per la soluzione del sistema lineare abbiamo utilizzato un *solver* diretto della libreria. Per il problema con l'ostacolo abbiamo scritto un *solver* iterativo che impone ad ogni iterazione e per ogni punto della griglia che la soluzione stia sopra l'ostacolo.

# Calcolo dell'integrale in forma *price*

Ricordiamo l'integrale da calcolare

$$\int_{\mathbb{R}} C(t, Se^y) \nu(y) dy$$

al quale applichiamo il cambio di  
variabile

$$z = Se^y$$

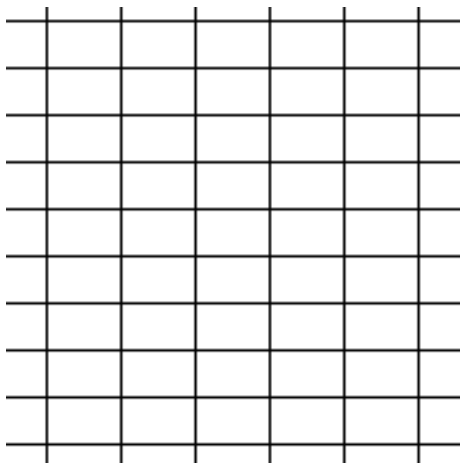


Figura: Una semplice griglia strutturata

# Calcolo dell'integrale in forma *price*

L'integrale diventa allora

$$\int_0^\infty \frac{C(t, z)}{z} \nu \left( \log \left( \frac{z}{S} \right) \right) dz$$

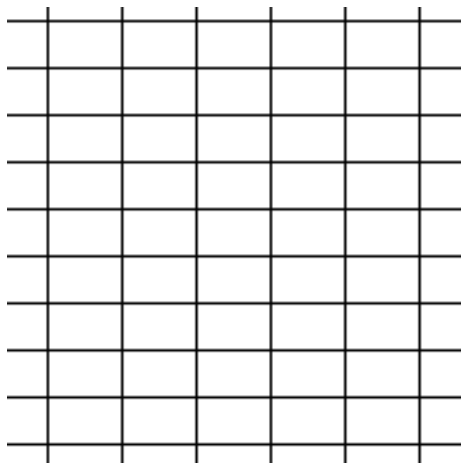


Figura: Una semplice griglia strutturata



# Calcolo dell'integrale in forma *price*

L'integrale diventa allora

$$\int_0^{\infty} \frac{C(t, z)}{z} \nu \left( \log \left( \frac{z}{S} \right) \right) dz$$

Quindi per ogni cella, si calcolano i contributi dovuti alla cella

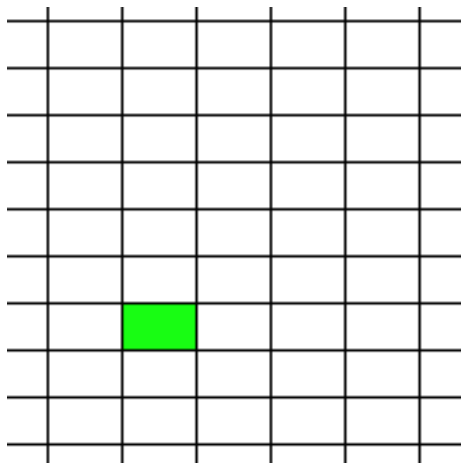


Figura: Poniamoci in una cella

# Calcolo dell'integrale in forma *price*

L'integrale diventa allora

$$\int_0^\infty \frac{C(t, z)}{z} \nu \left( \log \left( \frac{z}{S} \right) \right) dz$$

Quindi per ogni cella, si calcolano i contributi dovuti alla cella e si distribuiscono ai nodi di competenza:

- in 1d, a tutti i nodi
- in 2d, solo a quelli che giacciono sulla retta passante per la faccia selezionata. Prima sull'asse x.

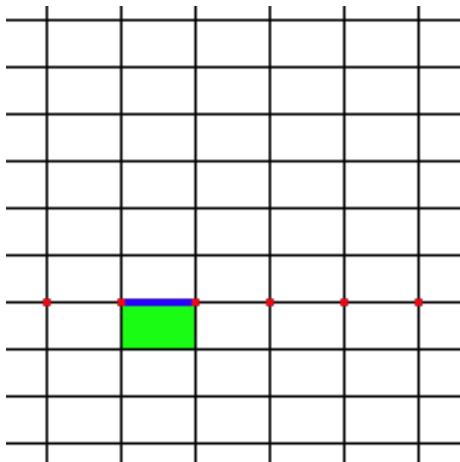


Figura: I contributi della cella ai nodi x

# Calcolo dell'integrale in forma *price*

L'integrale diventa allora

$$\int_0^\infty \frac{C(t, z)}{z} \nu \left( \log \left( \frac{z}{S} \right) \right) dz$$

Quindi per ogni cella, si calcolano i contributi dovuti alla cella e si distribuiscono ai nodi di competenza:

- in 1d, a tutti i nodi
- in 2d, solo a quelli che giacciono sulla retta passante per la faccia selezionata. Poi sull'asse  $y$ .

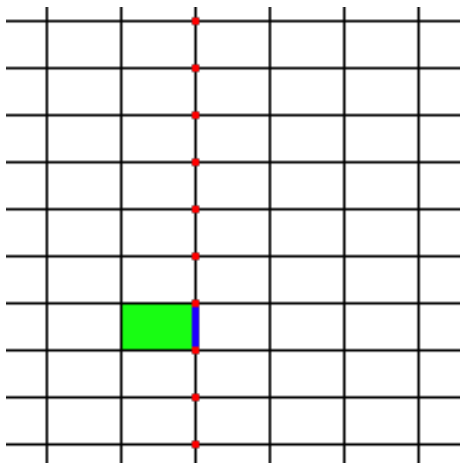


Figura: I contributi della cella ai nodi  $y$

# Calcolo dell'integrale in forma *log-price*

In questo caso l'integrale da calcolare è

$$\int_{-\infty}^{\infty} u(t, x + y) \nu(y) dy$$

su una griglia qualunque.

Osserviamo che si può uscire dal dominio a causa del termine  $x + y$ .

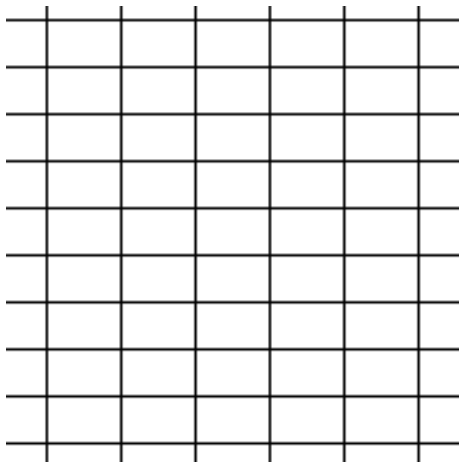


Figura: Una griglia

# Calcolo dell'integrale in forma *log-price*

In questo caso l'integrale da calcolare è

$$\int_{-\infty}^{\infty} u(t, x + y) \nu(y) dy$$

su una griglia qualunque.

Osserviamo che si può uscire dal dominio a causa del termine  $x + y$ .

Selezionato un vertice  $i$ ,

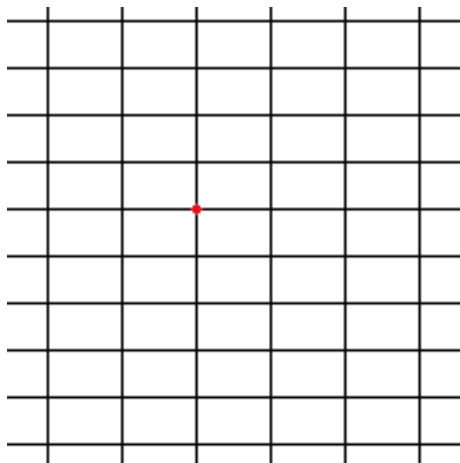


Figura: Poniamoci in un nodo

# Calcolo dell'integrale in forma *log-price*

In questo caso l'integrale da calcolare è

$$\int_{-\infty}^{\infty} u(t, x + y) \nu(y) dy$$

su una griglia qualunque.

Osserviamo che si può uscire dal dominio a causa del termine  $x + y$ .

Selezionato un vertice  $i$ , si avranno dei nodi di quadratura in direzione  $x$  e si quadra su  $x_i + z_l$  (in blu), e se la dimensione è due, anche su  $y$  lungo  $y_i + z_q$  (in verde).

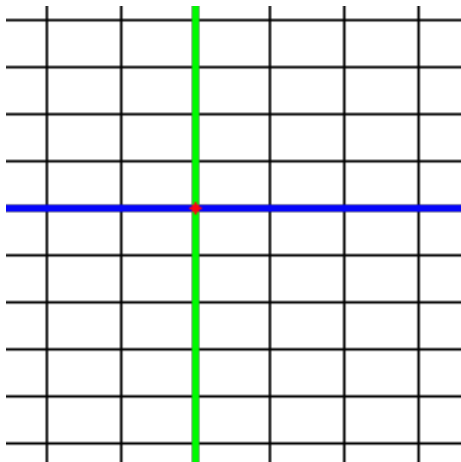


Figura: Calcolo lungo le direzioni

# Indice

- 1 Introduzione
- 2 Il problema
- 3 Struttura del codice**
- 4 Risultati
- 5 Conclusioni

# La libreria deal.ii

## Libreria deal.ii

Una potente libreria *open source* ad elementi finiti sui quadrilateri. Molto completa e semplice da utilizzare all'inizio, permette di risolvere problemi variazionali fino a 3 dimensioni con poche righe di codice.



# La libreria deal.ii

## Libreria deal.ii

Una potente libreria *open source* ad elementi finiti sui quadrilateri. Molto completa e semplice da utilizzare all'inizio, permette di risolvere problemi variazionali fino a 3 dimensioni con poche righe di codice.

## Vantaggi

- Documentazione molto ampia e chiara, a cui si aggiunge la presenza di oltre 50 *tutorial* che illustrano come usare la libreria per problemi più o meno tipici.
- Organizzata in moduli che coprono le diverse aree di un problema ad elementi finiti (*creazione griglie, algebra lineare, output risultati, etc*).

# La nostra implementazione

Tre strutture chiave per il problema

## Classi Opzione

Rappresentano il problema e gestiscono creazione griglia, assemblaggio sistema e soluzione.

## Classi Modello

I vari modelli utilizzati in finanza sono rappresentati con queste classi, la cui interfaccia è stabilita da una classe base astratta.

## Classi Integrali

Il calcolo della parte integrale è gestito da queste classi, e le Opzioni salvano un puntatore a un oggetto di questo tipo.

Tutte queste strutture sfruttano il meccanismo dell'ereditarietà al fine di coprire i diversi casi possibili.

# Le classi Opzione

Seguendo la linea di deal.ii, le classi opzione costituiscono il *core* del programma ad elementi finiti. Implementano i vari metodi necessari per la soluzione del problema.

Le classi foglia sono quelle effettivamente usate, in quanto implementano tutti i metodi.

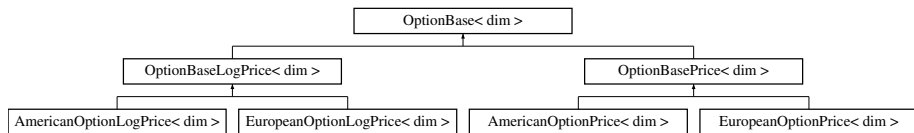


Figura: Schema delle classi Opzione

# Le classi Opzione

Seguendo la linea di deal.ii, le classi opzione costituiscono il *core* del programma ad elementi finiti. Implementano i vari metodi necessari per la soluzione del problema.

Le classi foglia sono quelle effettivamente usate, in quanto implementano tutti i metodi.

## Factory di Opzioni

Per facilitare la creazione di opzioni all'utente, è stata creata una *Factory* che permette di creare i vari oggetti **Opzione** con un'interfaccia comune.

# Le classi Opzione

Seguendo la linea di deal.ii, le classi opzione costituiscono il *core* del programma ad elementi finiti. Implementano i vari metodi necessari per la soluzione del problema.

Le classi foglia sono quelle effettivamente usate, in quanto implementano tutti i metodi.

## Factory di Opzioni

Per facilitare la creazione di opzioni all'utente, è stata creata una *Factory* che permette di creare i vari oggetti **Opzione** con un'interfaccia comune.

## Estensibili

L'utente può sia utilizzare le opzioni già esistenti, sia crearne delle nuove partendo dal secondo o dal terzo livello di ereditarietà.

# Le classi Integrable

Per calcolare la parte integrale, sono state create una serie di classi. Il secondo livello di ereditarietà distingue fra *price* e *log-price*, mentre le classi foglia implementano quadrature specifiche per i modelli.

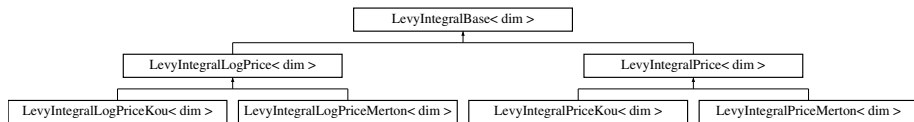


Figura: Schema delle classi LevyIntegral

# Le classi Integrare

Per calcolare la parte integrale, sono state create una serie di classi. Il secondo livello di ereditarietà distingue fra *price* e *log-price*, mentre le classi foglia implementano quadrature specifiche per i modelli.

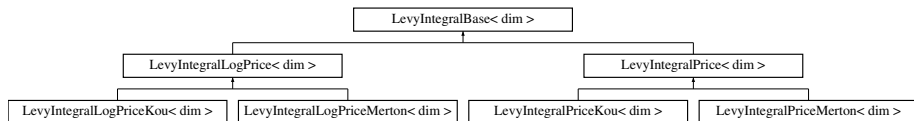


Figura: Schema delle classi LevyIntegral

## Generiche

Le classi intermedie non sono astratte e implementano formule di calcolo generiche (cioè, Gauss) per risolvere il problema con modelli qualsiasi.

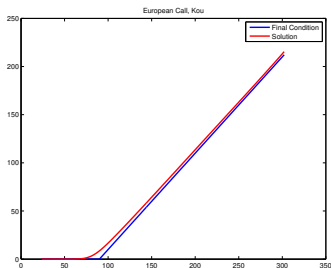
# Indice

- 1 Introduzione
- 2 Il problema
- 3 Struttura del codice
- 4 Risultati**
- 5 Conclusioni

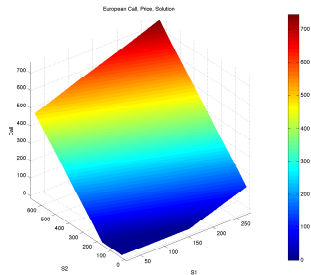


# Opzioni Europee

Alcuni esempi di risultati ottenuti:



(a) Call europea, modello di Kou

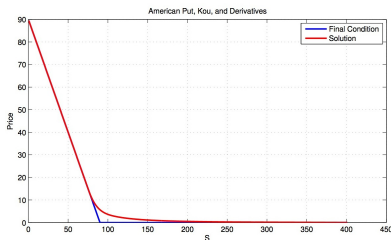


(b) Call Basket europea, modello di Black&Scholes

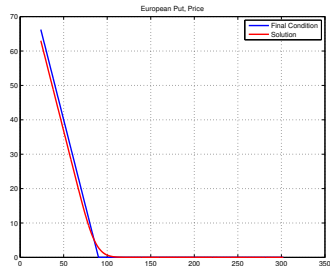
I risultati, confrontati con la soluzione analitica (nei pochi casi in cui esiste) o con altri *software* di simulazione (differenze finite o Montecarlo), sono corretti.

# Opzioni Americane

Risultati ottenuti nel problema con ostacolo:



(a) Put americana, modello di Black&Scholes



(b) Put europea, modello di Black&Scholes

Come possiamo notare, il *solver* dell'americana impone che la soluzione stia sempre sopra l'ostacolo, nonostante essa tenda ad andare sotto il *payoff*.

# Price v.s Log-Price

Con entrambi i metodi si ottengono risultati corretti e soddisfacenti:

<i>Price</i>	<i>Log-Price</i>
In 1d molto veloce	In 1d mediamente veloce
In 2d performance ottime	In 2d lento
Non parallelizzabile	Facilmente parallelizzabile (quindi più veloce di <i>price</i> 1d con almeno 4 <i>cores</i> )
No <i>mesh adapting</i> in 2d per PIDE	<i>Mesh adapting</i> anche in 2d, migliorando le prestazioni
Il troncamento del dominio può introdurre problemi	Nessun problema con il tronca- mento del dominio

**Tabella:** Confronto fra *Price* e *Log-Price*

# Price v.s Log-Price

Con entrambi i metodi si ottengono risultati corretti e soddisfacenti:

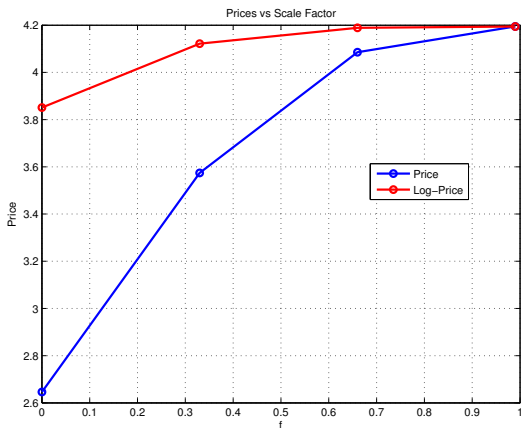
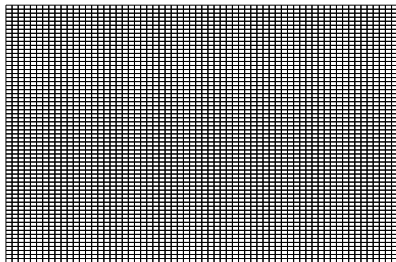


Figura: Convergenza del prezzo per una put al variare dello *scaling factor*

# Mesh adaptivity

Utilizzando le funzioni della libreria deal.ii, è facile adattare la griglia:

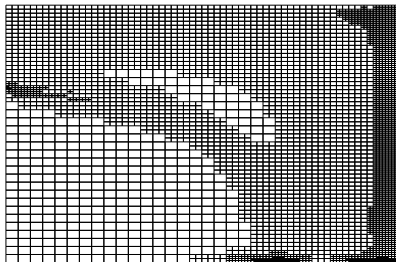


(a) Griglia iniziale

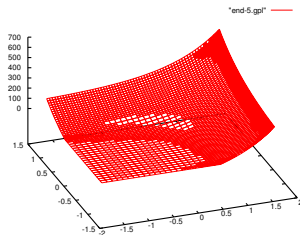
**Figura:** Adattamento di griglia per una Call Europea 2d in forma *Log-Price*

# Mesh adaptivity

Utilizzando le funzioni della libreria deal.ii, è facile adattare la griglia:



(a) Griglia adattata



(b) Soluzione con griglia adattata

**Figura:** Adattamento di griglia per una Call Europe 2d in forma *Log-Price*

# Indice

- 1 Introduzione
- 2 Il problema
- 3 Struttura del codice
- 4 Risultati
- 5 Conclusioni**

# Bilancio del progetto

Il programma finale si configura come una piccola ma solida libreria per il *pricing* di derivati finanziari di base, con la possibilità di prezzarne altri facilmente. Inoltre lascia molta libertà all'utente (più trasformazioni, scelta di parametri, *mesh adapting*).

## FEM vs. FDM

Grazie agli elementi finiti è possibile avere un'approssimazione più pregiata della soluzione, permettendo di calcolare in modo più preciso anche le derivate.

## Prestazioni rispetto ad altri *software*

Le prestazioni sono migliori se comparate con altri *software*, specialmente nella risoluzione di problemi con ostacolo.



# Possibilità di estensioni

Il progetto, avendo una struttura aperta, si presta molto facilmente ad estensioni. Alcune idee:

- aggiunta di altri derivati finanziari simili

# Possibilità di estensioni

Il progetto, avendo una struttura aperta, si presta molto facilmente ad estensioni. Alcune idee:

- aggiunta di altri derivati finanziari simili
- aggiunta di altri modelli finanziari

# Possibilità di estensioni

Il progetto, avendo una struttura aperta, si presta molto facilmente ad estensioni. Alcune idee:

- aggiunta di altri derivati finanziari simili
- aggiunta di altri modelli finanziari
- parallelizzazione in memoria distribuita

# Possibilità di estensioni

Il progetto, avendo una struttura aperta, si presta molto facilmente ad estensioni. Alcune idee:

- aggiunta di altri derivati finanziari simili
- aggiunta di altri modelli finanziari
- parallelizzazione in memoria distribuita
- estensione al caso con tre sottostanti.

**Vi ringraziamo dell'attenzione.**