Riassunto Iniziale Progetto Pacs

Nahuel Foresta, Giorgio Re

21 aprile 2014

1 Riassunto Obiettivi

L'obiettivo del progetto è di creare un programma per prezzare una serie di opzioni utilizzando dei metodi basati a elementi finiti. Il valore di un opzione al variare del sottostante (che supponiamo evolvere secondo un modello Jump-Diffusion) può essere quasi sempre trovato come soluzione di un equazione integro differenziale del tipo

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

su $[0, T] \times [0, +\infty]$ con opportune condizioni al bordo e condizione finale $C(T, S) = g(S)$ payoff dell'opzione. k è un nucleo con una forte massa nell'intorno dello zero e code esponenziali. In due dimensioni, supponendo l'indipendenza delle componenti di salto dei due sottostanti, tale equazione diventa:

$$\begin{aligned}\frac{\delta C}{\delta t} + \frac{\sigma_1^2}{2} S_1^2 \frac{\delta^2 C}{\delta S_1^2} + \frac{\sigma_2^2}{2} S_2^2 \frac{\delta^2 C}{\delta S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\delta^2 C}{\delta S_1 \delta S_2} + r\frac{\delta C}{\delta S_1} + r\frac{\delta C}{\delta S_2} - rC + \\ + \int_{\mathbb{R}} \left(C(t, S_1 e^y, S_2) - C(t, S_1, S_2) - S_1(e^y - 1)\frac{\delta C}{\delta S}(t, S_1, S_2) \right) k_1(y)dy \\ + \int_{\mathbb{R}} \left(C(t, S_1, S_2 e^y) - C(t, S_1, S_2) - S_2(e^y - 1)\frac{\delta C}{\delta S}(t, S_1, S_2) \right) k_2(y)dy = 0 \\ (2)\end{aligned}$$

su $[0, T] \times [0, +\infty]^2$ con opportune B.C. e valore finale.
Al variare delle condizioni finali e delle condizioni al contorno si possono descrivere altri tipi di opzioni. Un esempio interessante è il caso dell'opzione asiatica, che dipende dalla media del sottostante nel tempo. Infatti se consideriamo la media come seconda variabile, si ottiene un equazione simile a quella precedente.

2 Strumenti

2.1 La libreria deal II

Per realizzare questo progetto, l'idea è la libreria deal II per gli elementi finiti. Tale libreria permette un approccio pulito alle diverse parti necessarie alla costruzione di un programma ad elementi finiti. Sono presenti classi per le griglie, le matrici, le funzioni di base, la quadratura, i solutori e quant'altro necessario alla soluzione di un problema EF classico. Il punto principale sarà di aggiungere il termine integrale.

2.2 Altri strumenti

Abbiamo per ora incluso la libreria gsl per l'interpolazione del valore soluzione in punti non appartenenti alla mesh. Siccome è una parte piuttosto pesante, si cercano modi di evitare l'interpolazione (vedere in seguito). La libreria deal II utilizza CMake di default per generare i makefile, e quindi abbiamo adottato tale metodo, adattandolo alle nostre necessità.

3 Cosa è stato fatto

In questa prima parte del progetto, abbiamo iniziato a fare i primi esperimenti con la libreria deal II. In particolare abbiamo:

- Costruito un programma che risolve l'equazione PDE (senza parte integrale) in una dimensione in un caso semplice utilizzando unicamente gli strumenti forniti dalla libreria.
- Abbiamo risolto lo stesso problema nel caso bi-dimensionale. Sebbene il comportamento qualitativo della soluzione è quello aspettato, i valori esatti non sono ancora giusti (confrontati con dei risultati dati da tool che risolvono l'equazione con metodi alle differenze finite).
- Abbiamo provato a risolvere l'equazione Integrodifferenziale in una dimensione in diversi modi (per il dettaglio vedere sotto). In due casi siamo riusciti ad ottenere un risultato corretto, ma non soddisfacenti.

3.1 Metodologia

La PDE (e la PIDE) in questione è trasformabile in un equazione a coefficienti costanti con la trasformazione $x = \ln S$ (o $x = \ln {}^{S}\!/_{S_0}$) e $C(t, S) = u(t, x)$. In tal caso diventa (1D):

$$\frac{\delta u}{\delta t} + \frac{\sigma^2}{2}\frac{\delta^2 u}{\delta x^2} + \left(r - \frac{\sigma^2}{2}\right)\frac{\delta u}{\delta x} - ru + \int_{\mathbb{R}} \left(u(t, x+y) - u(t, x) - (e^y - 1)\frac{\delta u}{\delta x}\right) k(y)dy = 0 \tag{3}$$

In molti casi, è possibile separare i tre pezzi dell'integrale, e trattare gli ultimi due addendi separatamente. Definendo

$$\hat{\lambda} = \int_{\mathbb{R}} u(t, x)k(y)dy \qquad \hat{\alpha} = \int_{\mathbb{R}} (e^y - 1)\frac{\delta u}{\delta x}k(y)dy$$

L'equazione (3) diventa:

$$\frac{\delta u}{\delta t} + \frac{\sigma^2}{2}\frac{\delta^2 u}{\delta x^2} + \left(r - \frac{\sigma^2}{2} - \hat{\alpha}\right)\frac{\delta u}{\delta x} - r\hat{\lambda}u + \int_{\mathbb{R}} u(t, x+y)k(y)dy = 0 \tag{4}$$

Per la parte temporale abbiamo utilizzato una discretizzazione differenze finite con schema di eulero implicito, eccetto per la parte integrale che viene trattata esplicitamente e messa nel *rhs*.Tale schema è stabile purché $\Delta t < 1/\hat{\lambda}$. Gli elementi finiti scelti sono Q1, polinomi lineari sui quadrati, offerti gentilmente dalla libreria deal II. Il dominio su S viene troncato all'intervallo (S_{min}, S_{max}), opportunamente scelto.
La difficoltà si riduce dunque a integrare $u(t, x + y)k(y)$.
Senza entrare nei dettagli, otteniamo una discretizzazione del tipo :

$$M_1 u^k = M_2 u^{k+1} + J^{k+1} \qquad \text{per } k = M \ldots 1 \qquad \text{e} \qquad u^M(S) = g(S) \tag{5}$$

Dove M_1 è la somma delle matrici date dagli elementi finiti (stiffnes, etc, etc) e M_2 è la matrice di massa divisa per il passo temporale. Il termine esplicito J può essere calcolato in principio in diversi modi.

3.2 La parte integrale J

$k(y)$ è un nucleo che decresce rapidamente, è quindi possibile troncare il dominio d'integrazione ottenendo una soluzione di poco diversa (esistono stime a riguardo, qua non citate). L'integrale è allora da fare sull'intervallo (B_l, B_u), con B_l e B_u opportunamente scelti. Si presentano due problemi con questo termine:

- Nel caso generale $(S_{min}, S_{max}) \subset (B_l, B_u)$, quindi il termine integrale è non locale. Si sceglie di estendere il valore di u utilizzando la condizione al bordo, pratica comune in questi casi.
- Il termine $u(t, x + y)$ non è facilmente trattabile in quanto il fatto di sommare $x + y$ introduce un possibile shift al di fuori dai nodi di griglia al quale bisogna stare attenti

3.2.1 Possibili approcci

Una prima differenza negli approcci ahl'ultim termine integrale è la scelta di scrivere l'incognita u subito come appartenente allo spazio V_h dove sono ambientati gli elementi finiti, o prima integrare ottenendo una funzione $J(x)$ e scrivere quest'ultima come oggetto di V_h. Partiamo dal secondo approccio.

3.2.2 Integrazione prima di discretizzazione

Questa è una tecnica a quanto pare più diffusa (nell'approccio a differenze finite) In questo caso, si tratta di calcolare:

$$J_i = J(x_i) = \int_{B_l}^{B_u} u(t, x_i + y)k(y)dy$$

nei diversi nodi x_i della griglia. In seguito si potra scrivere dunque

$$\sum_{j=1}^{N} J_j \int_{x_{min}}^{x_{max}} \phi_i(x)\phi_j(x)dx$$

scrivibile come $M\underline{J}$, con M matrice di massa, e aggiungere tale termine all'*rhs*.

L'eq da risolvere a ogni passo temporale è dunque:

$$M_1 u^k = M_2 u^{k+1} + M\underline{J}^{k+1} \qquad \text{per } k = M \dots 1$$

In tal caso ci sono due modi di calcolare il vettore $\underline{J}$:

- Utilizzare una griglia qualunque d'integrazione, interpolando il valore di u in $x_i + y$ se $x_i + y$ cade all'interno del dominio, o imponendo le condizioni al bordo se fuori dal dominio. Tale operazione, abbastanza semplice, è molto lenta. Per l'interpolazione abbiamo usato le funzioni della *gsl*, utilizzando un metodo a spline.

- Allineare le griglie d'integrazione e del dominio in modo che $x_i + y$ coincida sempre con un nodo quando dentro il dominio. Questa procedura è semplice in 1D se il passo della griglia è costante. Questa procedura è più veloce, ma suppone che i nodi siano ordinati.

In questi due casi abbiamo ottenuto risultati corretti in 1D, ma stiamo cercando di capire come si può fare in più dimensioni.
Una strada che abbiamo provato in questo caso è di usare il cambio di variabili $x + y = z$, e il termine da integrare diventa

$$J_i = J(x_i) = \int_{B_l}^{B_u} u(t, x_i + y)k(y)dy = \int_{B_l + x_i}^{B_u + x_i} u(t, z)k(z - x_i)dz$$

La griglia d'integrazione cambia passo a passo, ma anche in questo caso deve essere allineata con la griglia del dominio ed è necessario sfruttare l'ordinamento.

3.2.3 Discretizzazione e in seguito integrazione

Un altra strada più naturale è di scrivere prima la funzione u come appartente allo spazio V_h. In tal caso si ha al passo k:

$$\int_{x_{min}}^{x_{max}} \phi_i(x) \left(\int_{B_l}^{B_u} \sum_{j=1}^{N} u_j^k \phi_i(x + y)k(y)dy \right) dx =$$
$$\sum_{j=1}^{N} \int_{x_{min}}^{x_{max}} \int_{B_l}^{B_u} \phi_i(x)\phi_i(x + y)k(y)dxdy$$

che si potrebbe anche scrivere, tramite un cambio di variabile prima della moltiplicazione per ϕ_i come

$$\sum_{j=1}^{N} \int_{x_{min}}^{x_{max}} \int_{B_l + x}^{B_u + x} \phi_i(x)\phi_i(z)k(z - x)dzdx$$

In tal caso, si potrebbe scrivere come una matrice densa A e l'equazione da risolvere a ogni passo temporale sarebbe:

$$M_1 u^k = M_2 u^{k+1} + A u^{k+1} \qquad \text{per } k = M \dots 1$$

Sebbene si abbia una moltiplicazione matrice densa per vettore a ogni passo, la matrice va costruita un unica volta.

Per ora non siamo riusciti a ottenere risultati corretti con questa tecnica, ne con cambio di variabili, ne con quadrature ne cercando di calcolare esattamente. Questo approccio sembra prestarsi a una costruzione della matrice ffacendo un ciclo sui quadrati, che sarebbe estensibile più facilmente al 2D.

3.2.4 Conclusioni parziali e futuri sviluppi

Riassumendo, dei due approcci per calcolare l'integrale, il primo ha dato riultati corretti ma sembra più difficile da estendere a due dimensioni in modo efficente. Il secondo fin'ora non ha dato risultati, probabilmente per errore negli algoritmi.

Un terzo approccio da esplorare è il cambio di variabili $Se^y = z$ a livello equazione. In tal caso si ottiene un equazione a coefficienti non costanti, ma la parte integrale potrebbe essere più semplice.