

# RELAZIONE PROJECT WORK “OutOfBounds”

FINANCIAL RISK MANAGEMENT



*Scuola di Economia e Management*

Lavoro svolto da:

Giulia Gasparini  
Brian Signori  
Anna Tiddia  
Ilenia Zendri

Anno accademico 2019/2020

In questo project work si è analizzato come mitigare l'esposizione creditizia di un portafoglio di strumenti finanziari, nel nostro caso composto da Swap su tassi di interesse, mettendo in evidenza l'impatto della collateralizzazione e del netting.

Inizialmente, al fine di formalizzare il contratto tra le due controparti, si è creata la classe CreditSupportAnnex, contenente tutti i requisiti per la delineazione di garanzie tra le controparti, in modo da permettere lo scambio di strumenti derivati, mitigando il rischio di credito. Tra gli elementi più rilevanti stabiliti tra le parti, utilizzati in questo project work, si trovano i Threshold e i Minimum Transfer Amount, sia in versione Posting sia Receiving.

In seguito alla creazione, all'interno della classe ExposureTest, del markToMarketCube (MTMC), che rappresenta lo stato complessivo del portafoglio, è stata calcolata l'esposizione creditizia netta del portafoglio  $V_t$  all'interno della classe CounterpartyExposureAggregation.

Per calcolare  $V_t$ , sono state richiamate, per mezzo delle FactorKey, tutte le Surface che contengono le simulazioni del valore dei singoli strumenti finanziari, salvandole in un nuovo array somma[] di tipo RandomVariable. Questo vettore contiene, per ogni istante temporale, l'esposizione creditizia di portafoglio, per tutti i possibili scenari ( $\omega$ ) e per tutti gli istanti temporali  $t$  di ogni trade, calcolata secondo la sommatoria:

$$V_t(\omega) := \sum_{i=1}^N V_t^i(\omega)$$

Nel caso in cui sia presente un unico strumento finanziario, il valore contenuto in somma[] coincide con lo strumento stesso.

Successivamente, si è proceduto al calcolo dell'evoluzione del collaterale, ovvero il margine di garanzia da applicare all'esposizione netta di portafoglio precedentemente calcolata, al fine di gestire l'esposizione creditoria a cui si è esposti, e quindi attenuare il rischio di credito.

Sono stati calcolati, a tal fine, il Collateral Balance (C), il Positive Collateral Account Balance (PCAB) e il Negative Collateral Account Balance (NCAB), salvando i valori trovati in tre array di tipo RandomVariable, che abbiamo chiamato con i medesimi nomi. Per determinare questi valori si sono seguite procedure simili, ma con delle differenze per ciascuno:

- Per il calcolo del PCAB<sub>t</sub>, è stata creata una RandomVariable sommaPos, che contiene i valori risultanti, per ogni istante temporale e per ogni scenario ( $\omega$ ), dalla somma del valore del portafoglio, del Threshold Posting e dell'MTA Posting, confrontando i valori di sommaPos con lo 0, al fine di trovare il valore minimo.
- Analogamente, per il calcolo della NCAB<sub>t</sub>, è stato svolto un procedimento simile, che differisce dal primo per il fatto che il confronto viene effettuato ricercando il massimo tra lo 0 e i valori contenuti nella RandomVariable sommaNeg, che contiene la differenza tra il valore del portafoglio, del Threshold Receiving e dell'MTA Receiving.
- Ct, invece, è stato calcolato, semplicemente, come somma di PCAB<sub>t</sub> e NCAB<sub>t</sub>.

Per effettuare questo confronto sono stati creati due vettori di supporto (valoriSommaPos e valoriSommaNeg), che, inizialmente, contenevano i valori assunti dalle RandomVariable sommaPos e sommaNeg, che sono stati, successivamente, sovrascritti con i valori risultanti dal confronto, eseguito con la funzione Math.min() per PCAB<sub>t</sub> e Math.max() per NCAB<sub>t</sub>.

Infine, sono state create tre Surface, a cui ad ognuna è stato assegnato uno dei vettori di RandomVariable appena creati, per inserirle nel SurfaceContainer.

Durante la revisione del progetto, ci si è resi conto che lo stesso procedimento poteva essere eseguito sfruttando i metodi cap() e floor(), presenti nella libreria di finmath. Dal momento che entrambe le soluzioni portano allo stesso risultato, si è ritenuto preferibile mantenere la nostra implementazione, ritenendola più ottimizzata. Dopo attente riflessioni sul codice, si ritiene che questa soluzione sia più efficiente in termini di tempo di esecuzione, dato che si sono utilizzati i metodi Math.max() e Math.min() direttamente all'interno delle classi CollateralSimulation e PostCollateral al momento del confronto. Se, invece, si avesse implementato l'altra soluzione, si avrebbe dovuto richiamare un metodo della libreria finmath, al cui interno sono presenti sempre i medesimi metodi (Math.max() e Math.min()). In questo modo si è evitato di richiamare un metodo esterno per effettuare il confronto, producendo un codice che risulta computazionalmente meno costoso.

Dopo aver calcolato il valore del portafoglio e del collaterale, si è calcolata, all'interno della classe `PostCollateralExposureCalculation`, l'esposizione netta, cioè la `Post Collateral Exposure (PCE)`, per ogni istante di tempo  $t$  e per ogni  $\omega \in \Omega$ , che è determinata dalla differenza tra l'esposizione creditizia di portafoglio e il collaterale.

Come nel caso precedente, sono stati definiti tre nuovi vettori di `RandomVariable`, che rappresentano la `Positive Exposure (PE)`, la `Negative Exposure (NE)` e l'`Exposure (E)`, ed è stata richiamata la `Surface` "somma" creata in precedenza.

Nel caso in cui sia presente il collaterale, si richiama la corrispondente `Surface` e si definiscono tre variabili aleatorie, che rappresentano l'esposizione creditizia, il collaterale e la loro differenza. Per trovare i valori della `PE` e della `NE` si è ricercato (per ogni istante di tempo e per ogni scenario) nel primo caso, il massimo tra i valori di questa differenza e 0, mentre, nel secondo caso, il valore minimo tra gli stessi valori con segno negativo e 0. Invece, nel caso in cui le due parti non avessero previsto lo scambio di un collaterale, si ricorre allo stesso procedimento prendendo in considerazione solamente l'esposizione netta.

Il confronto è stato nuovamente svolto utilizzando dei vettori di supporto.

Per entrambi i casi, sono state create le rispettive `Surface`, chiamate `positiveExposureSurface`, `negativeExposureSurface` e `postCollateralExposureSurface`, che contengono le esposizioni creditizie appena calcolate, in seguito aggiunte al container.

Successivamente, si è calcolata la `Potential Future Exposure (PFE)` per i vari istanti di tempo, che indica la massima esposizione creditizia attesa, ovvero il rischio massimo a cui ci si può esporre per un dato livello di confidenza fissato, nel nostro caso al 95%. La simulazione della `PFE` risulta di cruciale importanza al fine di poter meglio allocare il rischio di credito per una banca, dato che consente di tracciare il comportamento della vita della transazione. La `PFE` è stata calcolata nella classe `ExposureTest` all'interno del metodo `compare()`, che restituisce una matrice contenente i risultati ottenuti.

Come sempre si ricavano dal `MTMC` le `Surface` relative alla somma e alla somma collateralizzata, che saranno utili in seguito per il calcolo dei quantili. Per ogni istante temporale, si popola la matrice con il valore ottenuto dal metodo `getQuantile`, applicato prima all'esposizione creditizia netta ( $V_t$ ) e poi all'esposizione collateralizzata ( $V_t - C_t$ ).

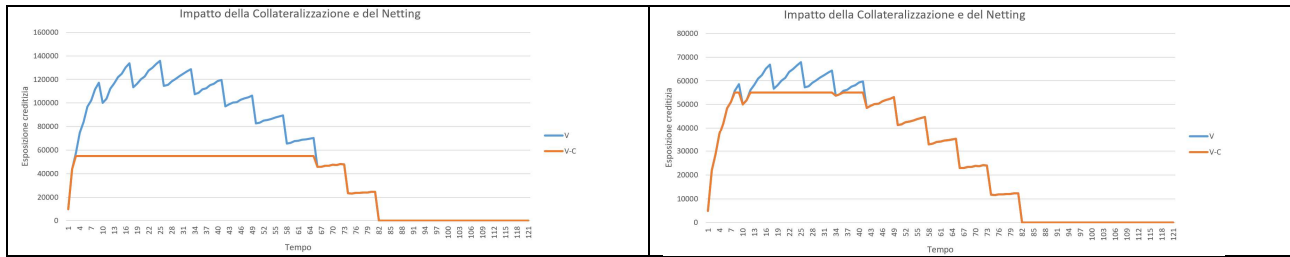
Al fine di studiare ed evidenziare meglio i valori trovati, per poter accertare la correttezza del nostro codice si è, in un primo momento, sfruttata la funzione di debugging, mentre, in un secondo momento, si è generato il grafico a partire dai valori trovati, che consente all'utente di analizzare più facilmente i risultati.

Per creare il grafico si è sfruttata la libreria `Apache POI`, richiamando l'oggetto `Workbook` di tipo `XSSF` per poter scrivere i dati risultanti dalla simulazione in un file Excel con estensione `.xls` (grafico.xlsx). Per il passaggio dei valori si è eseguito lo scorrimento della matrice creata in precedenza, stampando i valori (che sono 120, proprio come il numero degli istanti temporali che si considerano) nelle prime due colonne del file `.xlsx`. Queste due colonne sono state preimpostate per essere la base di dati utilizzata per la generazione del grafico all'interno di Excel.

Si è notato come, nel caso non collateralizzato, l'esposizione creditizia sia guidata solamente dall'andamento del valore del `MTM` dello `Swap`. Invece, nel caso collateralizzato, è presente una soglia massima, data dal collaterale, che pone un limite all'esposizione creditizia del portafoglio. In prossimità della maturazione dello strumento finanziario, l'esposizione creditizia presenta livelli più contenuti e si nota che le due esposizioni giungono a maturità assumendo i medesimi valori.

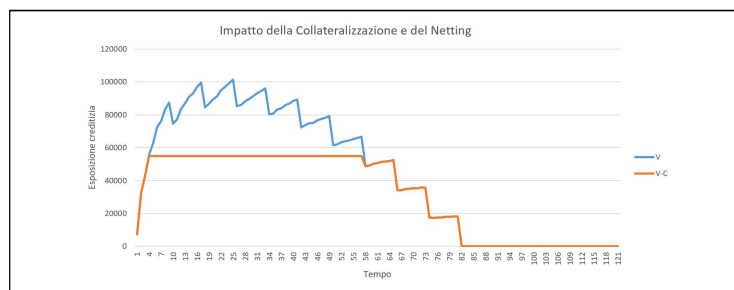
Un altro studio effettuato ha riguardato l'impatto che il netting può avere sulle esposizioni creditizie. Il netting è un sistema di compensazione per cui due controparti, che hanno esposizioni tra loro contrarie, prevedono di scambiarsi, in seguito ad un accordo, solamente un valore netto anziché l'intero debito o credito vantato. In questo modo il rischio che la controparte vada in default diminuisce.

Per effettuare questa analisi, si è inserito nel portafoglio, che fino ad ora era esclusivamente composto da un singolo `Swap`, che riceve il tasso fisso e paga il tasso variabile, un altro `Swap` con comportamento opposto rispetto al precedente. Per creare un effetto di netting con un risultato apprezzabile si è inserito nel portafoglio uno `Swap`, che riceve il variabile e paga il fisso, e che ha metà del nozionale originale, per differenziarlo dallo `Swap` precedente.



Nel grafico di sinistra è rappresentato un portafoglio con un singolo titolo, mentre si può notare come, nel grafico di destra, aggiungendogli un secondo swap, l'esposizione creditizia si abbassi per effetto del netting.

Al fine di simulare uno scenario più realistico, si è costruito un nuovo portafoglio composto da più Swap, le cui gambe e il nozionale sono assegnate randomicamente.



Come si può notare dal grafico, si può apprezzare l'impatto del netting, ossia come l'esposizione creditizia di un portafoglio composto da 10 titoli sia più bassa rispetto a quella del portafoglio composto da un solo titolo.

Il codice è implementato per costruire una simulazione triviale, ovvero di un portafoglio composto da uno o due titoli, oppure di un portafoglio più vicino alla realtà, composto da un numero arbitrario di strumenti.

Lato utente:

Il programma consente l'interazione dell'utente. Infatti, l'utente può scegliere, tramite input da tastiera, se eseguire la simulazione in un mercato reale o fittizio e con quanti titoli realizzarla. In questo modo, all'interno dello stesso file grafico.xlsx, si possono ottenere due risultati diversi.

Sono stati gestiti i possibili casi criticità, come la non esistenza del file o il fatto che il file sia già aperto al momento dell'esecuzione o la ricezione di un input errato da tastiera, segnalandole all'utente tramite popup implementati con la libreria Swing di java, consentendogli di effettuare un nuovo tentativo.