XVA and its Greeks Calculation

森本 裕介

三菱 UFJ 銀行

2018/11/23 - 11/25
数理ファイナンス合宿セミナー

Contents

Definition of XVA

- V : **従来のデリバティブの価値**
  - Counter party risk(**自社のクレジットリスク**) **を考慮しない**
  - **担保を勘案しない** (Variation margin, Initial Margin)
  - **自社の** Funding **コストを考慮しない**
- $\hat{V}$: **上記に付随する金利のスプレッド**, Cash flow **を勘案したデリバティブの価値**

$$\hat{V} = V + \sum_{X=C,D,F,M,\dots} XVA$$

- CVA: counterparty credit risk adjustment
- DVA: Own credit risk adjustment
- FVA: Funding cost
- MVA: Margin cost

$\hat{V}$ の導出には色々なやり方があり、それに応じて XVA の表現にも色々な流派がある。そもそも、XVA の分け方は当然色々なものがあり、各社マネージのしやすさに応じて自社独自のコストの仕分けを行なっている。

- Piterburg
- Burgard and Kjaer
- Albanese and Andersen
- Fuji and Takahashi
- Takada

Semi-Replication Theory

Burgard-Kjaer

- S : Spot asset price $dS = \mu(t, S)Sdt + \sigma_s S dW_s$
- r : risk free rate $dr = \mu_r(t, r)dt + \sigma_r(t, r)dW_r$
- P_r: default free bond : $dP_r = rdP_r dt$
- P_c, P_F: counterpraty and bank bond

$$dP_* = r_* P_*(t-)dt - (1 - R_*)P_*(t-)dJ_*, * = c, F$$

- β_* : repo account $d\beta_* = r_*\beta_* dt, * = S, r, c, F$
- $r_* = r + s_*,\ s_* = (1 - R_*)\lambda_*,\ * = c, F$
- $C = C_v + C_i$: collateral (Variation margin, initial margin)

Bank から見た Derivative の価値 $\hat{V}$

$$\hat{V} = \hat{V}(t, r, S, J_F, J_c),$$
$$\hat{V}(T, r, S, 0, 0) = H(r, S),$$
$$\hat{V}(t, R, S, 1, 0) = g_F, \hat{V}(t, R, S, 0, 1) = g_c.$$

Hedging portfolio value Π

$$\Pi = \delta_s S + \delta_c P_c + \delta_r P_r + \delta_F P_F + \alpha_s \beta_s + \alpha_c \beta_c + \alpha_r \beta_r - C,$$
$$\Pi = \hat{V}.$$

Hedge asset の Repo 調達を仮定

$$\delta_s S + \alpha_s \beta_s = 0,$$
$$\delta_c P_c + \alpha_c \beta_c = 0,$$
$$\delta_r P_r + \alpha_r \beta_r = 0.$$

Funding condition

$$\hat{V} + \delta_F P_F - C = 0.$$

Semi-Replication 1: **自社の** default risk **意外を全て** hedge

$$
\begin{aligned}
&d(\hat{V}+\Pi) \\
=&\{(\frac{\partial}{\partial t}+\mathcal{A})\hat{V}+r_F\delta_F P_F+\lambda_c(g_c-\hat{V})-r_v C_v-r_i C_i\}dt \\
&+\epsilon_h dJ_B
\end{aligned}
$$

$$
\hat{V}=V_{c_v}+CVA+FVA+MVA
$$

$$
\begin{aligned}
CVA &= -E[\int_0^T \lambda_c(t)D^{r_F+\lambda_c}(V_{c_v}-g_c)\ dt] \\
FVA &= -E[\int_0^T (r_F-r_{c_V})D^{r_F+\lambda_c}(V_{c_V}-g_c)\ dt] \\
MVA &= -E[\int_0^T (r_F-r_{c_i})D^{r_F+\lambda_c}C_i\ dt]
\end{aligned}
$$

FVA **は自社の** default risk **を完全に** hedge **できないことから出てくる**

$$FVA = DVA + FCA,$$
$$DVA = -E[\int_0^T \lambda_B(t) D^{r_F+\lambda_c}(V_{c_v} - g_B)\ dt]$$
$$FCA = -E[\int_0^T \lambda_B(t) D^{r_F+\lambda_c}\epsilon_h(t)\ dt].$$

FVA **に関する様々な議論** (Hull, White, Albanese, Andersen, Crepy, Duffie, etc.)

- Pricing **は** Funding **と切り離して行われるべきだ**
- Modiliani-Millar **の定理**: **企業価値は** Funding strategy **に依存しない に矛盾**
- Shareholder value, Firm Value

Semi-Replication 2： Martingale Part 以外をすべて hedge

$$
\begin{aligned}
=&\{(\frac{\partial}{\partial t} + \mathcal{A} - r)\hat{V} + \lambda_B(g_B - \hat{V}) + \lambda_c(g_c - \hat{V}) \\
&+(r - r_v)C_v + (r - r_i)C_i\}dt \\
&+ \epsilon_h(dJ_B - \lambda_B dt).
\end{aligned}
$$

$$\hat{V} = V_{c_v} + CVA + DVA + VMVA + VMVA + IMVA$$

$$CVA = -E[\int_0^T \lambda_c(t) D^{r+\lambda_B+\lambda_c}(V_{c_v} - g_c)\ dt]$$

$$DVA = -E[\int_0^T \lambda_B(t) D^{r+\lambda_B+\lambda_c}(V_{c_v} - g_B)\ dt]$$

$$VMVA = -E[\int_0^T (r - r_{c_V}) D^{r+\lambda_B+\lambda_c}(V_{c_v} - C_v)\ dt]$$

$$IMVA = -E[\int_0^T (r - r_{c_i}) D^{r+\lambda_B+\lambda_c} C_i\ dt]$$

XVA の計算法

- Counterpraty c に対する XVA は概ね以下のような計算になる.

$$\text{XVA}(c) = \sum_{j=1}^{J} E\left[\int_0^T Z(t)\left(\sum_{k\in N_j(c)} V_k(t) - C(t)\right)^{(+)} dt\right]$$

  - $V_k(t)$: 取引 k の Exposure(t 時点の時価)

$$V_k(t) = E\left[\sum_{t_i>t} F_k(t_i, X(t_i))|\mathcal{F}_t\right]$$

  - $N_j(c)$: c との取引契約はいくつかの Netting Set(CSA 条件などによって) に分かれている。

$$Trade(c) = \cup_{j=1}^{J} N_j(c)$$

  - Z : discount, spread, hazard, recovery rate, etc.
  - T : Trade(c) に含まれる取引の最長満期

実際の計算は大きく２つに大別される。

- Approximate Exposure function
  V_k **を原資産の関数として、その関数系を計算する。**

$$\tilde{V}_k : \mathbf{R}^{\mathbf{N}} \to \mathbf{R},$$

$$V_k(t) \approx \tilde{V}_k(X(t))$$

- Aggregation
  - $\tilde{V}_k(X(t))$ **から、**Collateral $\tilde{C}$ **を計算**.
  - collateral **勘案後の** Exposure **に** spread **をかけて積分を実施**.

$$E\left[\int_0^T \left(\tilde{V}_k(X(t)) - \tilde{C(t)}\right) D(t)\ dt\right].$$

Collateral 計算法

- VM(Variation Margin)
  - H: Threshold. Exposure **がこの額を超えていないと担保なし。**
  - M: MTA(minimum transfer amount). **支払い担保の最低額。**
  - δ: MPOR(mergin period of risk). Default **時点と最後に担保が支払われた時点のバッファ期間**. e.g. 14day.

$$C(t) = (V(t-\delta) - H)^+ 1_{\{V(t-\delta)-H>M\}}$$

- IM(Initial Margin)
  - MPOR **期間の** Gap Risk **をカバーする。** e.g. 10day 99% Var.
  - CCP(central counter party) **や、** SIMM(standard initial margin model) **ごとに** Greeks **を使った** Var **の近似公式が決められている。**

$$IM(t) = \sum \phi_{i,j} \frac{\partial V(t)}{\partial X^i(t)} \frac{\partial V(t)}{\partial X^j(t)}$$

XVA の数値計算手法

- 大量の risk facotor, product の計算を同時に行う必要があるため (Netting), Monte-Carlo 法が通常使われる。
- Exposure 計算には American Monte Carlo 法が使われる。多くの場合、LSM(Least square monte carlo) が用いられる。

$$\tilde{V}(x) = \sum_{i=1}^{n} a_i \phi_i(x), \ \ \phi_i : \text{polynomial}.$$

- LSM のメリット
  - 計算が高速かつ汎用的: model や商品性によらず、一律に適応できる
  - Data がシンプルになる: 複雑な取引情報を多項式の係数で表現できる
  - Netting が容易: path を発生させなくても、多項式の係数の和で netting できる。
- LSM のデメリット
  - 近似精度の問題

Implicit method

無担保の場合は、Exposure の近似精度の悪さを補う次のような手法がある。

$$XVA = E\left[\int_0^T Z(t)\left(\sum_{k\in N_j(c)} V_k(t)\right)^+ dt\right]$$

$$= E\left[\int_0^T Z(t)\left(\sum_{k\in N_j(c)} F_k(t_k, X(t_k))\right) 1_{\{\sum_{k\in N_j(c)} V_k(t)\geq 0\}}\right]$$

$$\approx E\left[\int_0^T Z(t)\left(\sum_{k\in N_j(c)} F_k(t_k, X(t_k))\right) 1_{\{\sum_{k\in N_j(c)} \tilde{V}_k(t)\geq 0\}}\right]$$

- Exposure **の近似は正負の符号判定のみに用いられる。**
- **値がどんなに離れていても符号さえ正しければ、誤差は発生しない。**

$$|\hat{c}_1(\omega) - c_0| \leqq C_{p,\delta} L^{-\frac{3/2(1-\delta)^2}{2+(1+\delta)(\tilde{N}+1)\ell_0/2}},$$

for any $\omega \in \Omega_L$ and $L \geqq 1$. If there exists $\gamma \in (0,1]$ such that

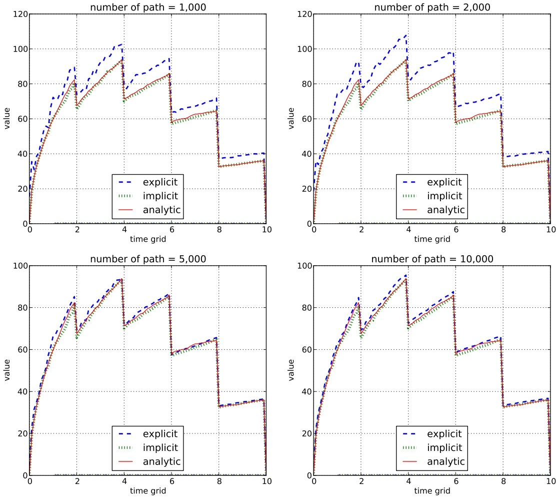
$$\int_0^T E^{\mu}[1_{|\sum_{k:T_k \geqq t_{i+1}} (P^{(k)}_{T_k - t} F_k)(\pi_k(X(t)))| \leqq \theta\}}] dt = O(\theta^{\gamma}), \text{as } \theta \downarrow 0,$$

$$|\hat{c}_2(\omega) - c_0| \leqq C_{p,\delta} L^{-\frac{3(\gamma+1)(1-\delta)^2}{(\gamma+2)(2+(1+\delta)(\tilde{N}+1)\ell_0/2)}},$$

特に $\gamma = 1$ **の場合、**

$$|\hat{c}_1 - c_0| \leqq C_{p,\delta} L^{-\frac{3(1-\delta)^2}{2(2+(1+\delta)(N+1)\ell_0/2)}}$$

$$|\hat{c}_2 - c_0| \leqq C_{p,\delta} L^{-\frac{3(1-\delta)^2}{2+(1+\delta)(N+1)\ell_0/2}}$$


number of path = 1,000
number of path = 2,000
number of path = 5,000
number of path = 10,000
value
time grid
explicit
implicit
analytic
0
20
40
60
80
100
120
0
2
4
6
8
10

XVA Greeks の種類

XVA は多くの Market Sensitivity(Greeks) を持っている。

- IR
  - **金利スワップ** (JPY 6m roll IRS, 1Y, 2Y, ...)
  - Tenor basis swap (JPY 3m roll IRS, JPY 3m roll IRS, ...)
  - **通貨スワップ** (USD LIBOR vs JPY LIBOR $+ \alpha$)
- IR Vol
  - Option **満期** $\times$ Swap Tenor
- FX
- FX Vol
- Infration Swap, CDS, etc.

基本的には Greeks **計算は有限差分法を使って求める。**

$$Greeks(\theta) = \frac{XVA(\theta + \epsilon) - XVA(\theta - \epsilon)}{2\epsilon}$$

XVA Greeks 計算 Tecnnique

愚直に差分を一つ一つ計算する以外に、以下のような Technique **が提案されている。**

- Freezing method
  - $XVA(\theta + (-)\epsilon)$ **を計算する際に、**Regression **の結果は** $XVA(\theta)$ **のものを使い回す方法**
  - American Option **の計算時にはうまくいくらしい。**
- Automatic Differentiation
  - **関数を関数を計算する際に、微分も同時に求められるような実装テクニック**
  - Neural Network **のパラメータにも広く利用されている** (Back Propagation)

Freezing 法の検証

- Parameter: $\theta = x_0 \ or \ \sigma$
- model : BS model :
  $dX(t, \theta) = X(t, \theta)(\mu dt + \sigma dW(t)), \ X(0, \theta) = x_0.$
- Pay off function :$F(x) = (x - K_1)^+ - K_2.$
- Exposure:

$$V(x, \theta) = E[F(X(T, \theta))|X(t, \theta) = x] \approx \tilde{V}(x),$$

- Approximated Exposure: $\tilde{V}(x, \theta) \approx V(x, \theta)$
- EPE: $EPE(t) = E[V(X(t))^+]$

この設定の下、Delta $\frac{\partial}{\partial x}EPE(t)$, Vega $\frac{\partial}{\partial \sigma}EPE(t)$, **を計算する。**

$$Explicit = \frac{\tilde{V}(X(t, \theta + \epsilon), \theta) - \tilde{V}(X(t, \theta - \epsilon), \theta)}{2\epsilon}$$

Explicit 法を用いた Freezing method

Table: Test result of Delta

method	mean	stdev
analytic	5.19	0.13
Non Freezing	5.53	0.13
Freezing	5.22	0.21

Table: Test result of Vega

method	mean	stdev
analytic	1264.77	61.08
Non Freezing	1480.38	74.98
Freezing	522.30	60.15

Delta は Freezing しても誤差はほとんどないが、Vega は Freezing すると誤差がとても大きい

Test 結果の分析 (Explicit 法)

$$\frac{\partial}{\partial\theta}E[V\left(X(t,\theta),\theta\right)^{+}] = E[1_{\{V(X(t,\theta),\theta)\geqq 0\}}\frac{\partial}{\partial\theta}V\left(X(t,\theta),\theta\right)]$$

$$= E[1_{\{V(X(t,\theta),\theta)\geqq 0\}}\left(\frac{\partial V}{\partial x_1}\left(X(t,\theta),\theta\right)\frac{\partial X(t,\theta)}{\partial\theta} + \frac{\partial V}{\partial x_2}\left(X(t,\theta),\theta\right)\right)]$$

- $\theta = x_0$ の場合、$\frac{\partial V}{\partial x_2}\left(X(t,\theta),\theta\right) = 0$ なので, θ 変化後に Regression を再度やる必要はなかった。
- $\theta = \sigma$ の場合、$\frac{\partial V}{\partial x_2}\left(X(t,\theta),\theta\right) \neq 0$ なので、θ 変化後に Regression をやり直さないと大きな誤差が生じる

一般に Markov 過程でモデル化された原資産の初期値パラメータに関する Greeks であれば、Freezing の誤差はないが、そうでない場合は Freezing により誤差が生じうる。

Implicit 法を用いた Freezing method

Table: Test result of Delta

method	mean	stdev
analytic	5.19	0.13
Non Freezing	5.16	0.12
Freezing	5.16	0.12

Table: Test result of Vega

method	mean	stdev
analytic	1264.77	61.08
Non Freezing	1262.38	60.99
Freezing	1262.38	60.99

Implicit method を使った場合、Delta も Vega もうまくいく理由の分析

$$\frac{\partial}{\partial\theta}E[F(X(T,\theta)1_{\{V(X(t,\theta),\theta)\geqq 0\}}] = E[\frac{\partial F}{\partial x}(X(T,\theta))\frac{\partial X(t,\theta)}{\partial\theta}]$$

$$+E[F(X(T,\theta))\delta_0(V\left(X(t,\theta),\theta\right))\frac{\partial V}{\partial x_2}\left(X(t,\theta),\theta\right)]$$

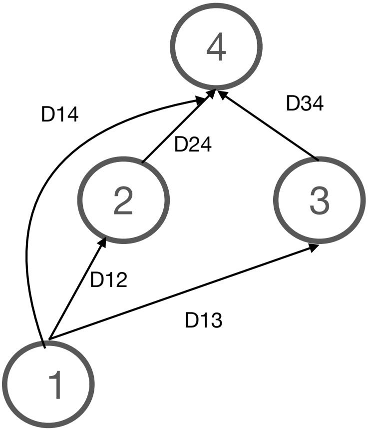
第二項目が Exposure **関数自体の** θ **に関する変動を含んだ式になっている。ただし、次のように変形すると** 0 **になることがわかる。**

$$\textbf{(第二項)} = E[E[F(X(T,\theta))|\mathcal{F}_t]\delta_0(V\left(X(t,\theta),\theta\right))\frac{\partial V}{\partial x_2}\left(X(t,\theta),\theta\right)]$$

$$= E[V(X(t,\theta))\delta_0(V\left(X(t,\theta),\theta\right))\frac{\partial V}{\partial x_2}\left(X(t,\theta),\theta\right)] = 0.$$

Automatic Differentiation の仕組み

- 基本的な関数の合成を計算グラフで表現し、プログラム内にその構造を保持
- 基本的な関数の微分は既知なので、その関数もプログラムしておく
- 最終的に得られる関数は基本関数の合成なので、Tree をたどって、各ノードの微分を掛け合わせて行くことが、Chain rule に対応しており、最終的な Greeks が求められる



$$\frac{dx_4}{dx1} = D14 + D12 * D24 + D13 * D34$$

XVA 計算における AD のメリット、デメリット

メリット

- 通常の値の計算の数倍（10 未満?）で大量の Greeks 計算が出来る。
- 真の関数の微分を実装しているので、差分法のようなバイアスがない。

デメリット

- 大量のメモリが必要
- On memory で一気通貫で計算しないと厳しい。途中で中間データを出力しながらだと、巨大なヤコビ行列を計算する必要が生じる。
- 途中に滑らかでない関数が入ると一気に依存するポイントの Greeks がおかしくなる
- path ごとの微分を行なっていくので、熱核で積分して滑らかになる要素が含まれていない。

AAD の最近の進展

- Stochastic AAD
- MVA への応用
- ccp