NATIXIS

Financial Mathematics + Scientific Computation
"AAD" applications in Finance
Local correlation + XVA

2016

Speaker:

- Adil Reghai – NATIXIS 07/07/2016

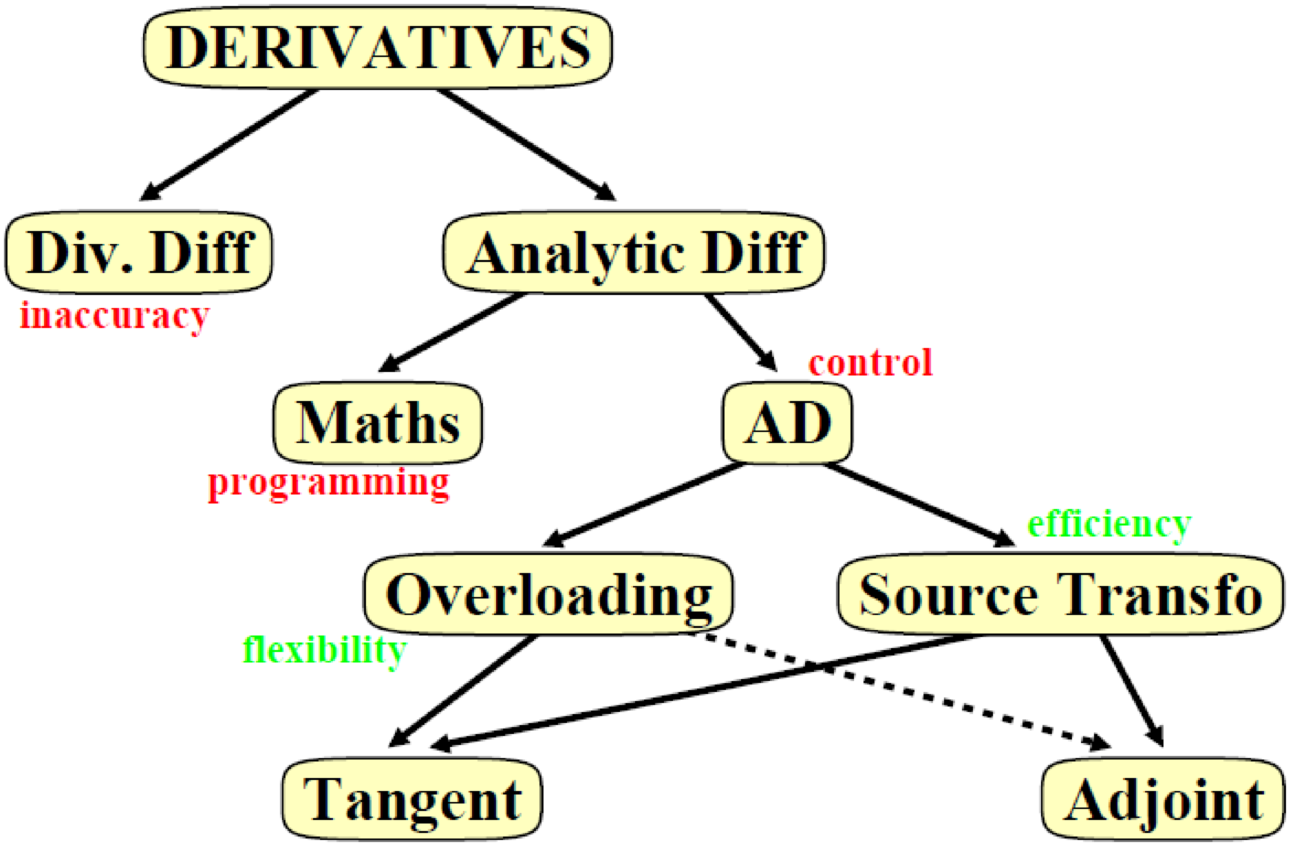
Acknowledgement

Thanks to the help of the Quant and the IT Team. To name few Marouen Messaoud, Adel Ben Haj Yedder, Florian Deyspesse, Sebastien Mollaret, Carole Camozzi, José Luu, Rida Mahi, Nicolas Venuti.

Table of contents

1a Algorithmic differentiation

Many strategies to compute derivatives (differentials)



Source:Olivier Hascöet

NATIXIS

AD references & tools

Industry:

Meteo France,
Dassault,
EDF,

Finance :

Luca Caprioti
Mike Gilles
Paul Glasserman
Christian Homescu
Olivier Pirroneau
Laurent Hascöet & al
Mark Joshi & al
Uwe Nauman

Software:

- **C++ framework (ADOL…)**
- **Code generation : tapenade**
- **DIY do it yourself methodology**

What is AD ?

What is AD ?

A set of techniques to numerically evaluate the derivative of a function specified by a computer program

- Automatic methodology
- Computes any number of derivatives

Financial applications – AD in pricing

Using AD in pricing

$$Price : \mathbb{R}^n \longrightarrow \mathbb{R}$$
$$\nabla Price : (\dots\dots\dots\dots\dots.)$$

Linear form (n)

- Calculating derivative makes the problem linear

Idea of **Pontryagin**

NATIXIS

Adjoint : Ordering the calculations

$$A(i)\colon\ \mathbb{R}^{nxn} and\ x\mathbb{R}^{nx1}$$

$$\left(\prod_{i=1,\dots,m} A(i)\right)x = A(1)\left(\dots\left(A(m-1)(A(m)x)\right)\ \right)$$

- Calculate the matrix x vector from the end is computationally more efficient than computing the product of all matrices

- Price of efficiency : Need of memory of the jacobian at each step

Algorithmic differentiation

Tangent linear model

- **Forward propagation of the chosen derivate**
  - More stable computation
  - Automatic method, naturally object oriented

Adjoint model

- **Transposed differentiation problem**
  - **Fast, Constant cost (Worst case: 4X the problem complexity)**
  - Important source code modification
    - Richer Forward
    - Non-generic Backward
    - Tremendous human cost

Algorithmic differentiation

- Price variation

$$p = F(x)$$
$$dp = F'(x)dx$$
$$=< \nabla F(x) | dx >$$

- Price function

$$\boxed{F = PoGoH}$$

- Tangent model

output ← input

$$dp =< (\nabla P) \mid (\nabla G) \mid (\nabla H) \mid dx >$$

- Adjoint model

output → input

$$=< (\nabla P) \mid (\nabla G) \mid (\nabla H) \mid dx >$$

Algorithmic differentiation - Adjoint model

$$
\begin{aligned}
\bar{x} &= <\bar{y}|\,\nabla F(x)> \\
&= <(\nabla F(x))^t|\bar{y}>
\end{aligned}
$$

// Forward sweep

For i = 1..n

$v_i = f_i(v_{i-1})$

$push(v_i)$

// Backward sweep

$v_n = Id(\dim(n))$

For i = n..1

$v_i = pop(\quad)$

$\overline{v_{i-1}} = (\nabla f_i(v_i))^t . \overline{v_i}$

(©J.Lotz)

NATIXIS

1b Algorithmic differentiation Pricing

Financial applications – AD in pricing

$$Price = E(Payoff \ o \ Diffusion) \ o \ Calibration$$

$$< \nabla Price| = < \underbrace{\nabla Payoff}_{\text{FD}} \ | \underbrace{\nabla Diffusion}_{\text{AD}} \ | \underbrace{\nabla \ Calibration}_{\text{Implicit function}} \ |$$

FD AD Implicit function

Finite differences

$$\boxed{\frac{\partial Price}{\partial x_1} = \frac{Price(x_1 + \varepsilon) - Price(x_1 - \varepsilon)}{2\varepsilon}}$$

Cega – Multi Terminal Skew diffusion

The model

$$C(K,T) = B(0,T)\mathbb{E}^{\mathbb{Q}}\left((S_T - K)^+\right)$$

$$F_T(K) = 1 + \frac{1}{B(0,T)}\frac{\partial C(K,T)}{\partial K}$$

- Sampling of $\tilde{Z}$
- Correlation of samples : , $Z \equiv L.\tilde{Z}$
- $U = \phi(Z)$
- $X = F^{-1}(U)$
- $P = P(X)$

Copula

NATIXIS

Cega – TSKEW Theoretical calculus

- Computing $\overline{X_i(t)} = \frac{\partial P}{\partial X_i(t)}$
- Computing
  - $\overline{Z_i(t)} = \frac{\partial P}{\partial Z_i(t)}$
    - $\bar{U} = \bar{X} \frac{1}{f(F^{-1}(U))}$
    - $\bar{Z} = \bar{U}\varphi(Z)$
  - $\bar{L} = \sum_t \bar{Z}\,\tilde{Z}^T$

- Finite Diff
- Vibrato
- Smoothing

Analytical
Finite Differences

Cega – Problematic & technical chalenges

One of the most costly calculations in MonteCarlo pricing

- **For a basket option with 10 underlying**
  - Cost of computation: $\frac{n(n-1)}{2} * CorrelTermStructureCount$
  - 45 = 10*(10-1)/2 price computation using asymmetrical differentiation
  - Local correlation model -> twice as much computation

Calculation time reduction by a factor of 50. More than 80000 hours saved.

NATIXIS

Cega – Finite Diff & AD

Finite differences

Barycentric bump

$$\nabla f(x) \approx \frac{f(x+\delta) - f(x-\delta)}{2\delta}$$

$$C_\varepsilon^+ = (1-\varepsilon)C + \varepsilon \begin{bmatrix} 1 & 1 & 0\ldots & 0 \\ 1 & 1 & 0\ldots & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V(C_\varepsilon^+) = V(C) + \varepsilon(1-\rho_{12})\frac{\partial V}{\partial \rho_{12}} - \varepsilon \sum_{i,j \neq 1,2} \rho_{ij} \frac{\partial V}{\partial \rho_{ij}} + o(\varepsilon)$$

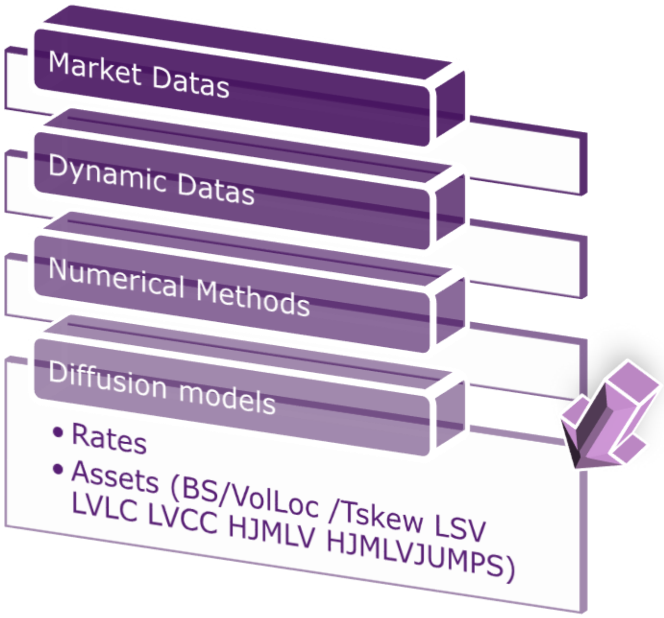
$$V(C_\varepsilon^-) = V(C) - \varepsilon(1+\rho_{12})\frac{\partial V}{\partial \rho_{12}} - \varepsilon \sum_{i,j \neq 1,2} \rho_{ij} \frac{\partial V}{\partial \rho_{ij}} + o(\varepsilon)$$

Estimator

$$\frac{\partial V}{\partial \rho_{12}} = \frac{V(C_\varepsilon^+) - V(C_\varepsilon^-)}{2\varepsilon} + o(1)$$

Library Architecture

Surgical coding DIY



Cega – BS & VolLoc diffusion

Model BS & VolLoc

- Sampling of $\tilde{Z}$
- Correlation of the samples : , $Z \equiv L.\tilde{Z}$
- Computation of martingale and spot :
  - $M_t = exp\left(\frac{{\sigma_{i,t-1}}^2(M_{t-1})}{2}(t-(t-1)) - \sqrt{(t-(t-1))}\sigma_{i,t-1}(M_{t-1})Z\right)M_{t-1}$
  - $X_t = A_t.M_t + B_t$ pour $t \in \|1,T\|$
- Payout computation $P = P(X)$

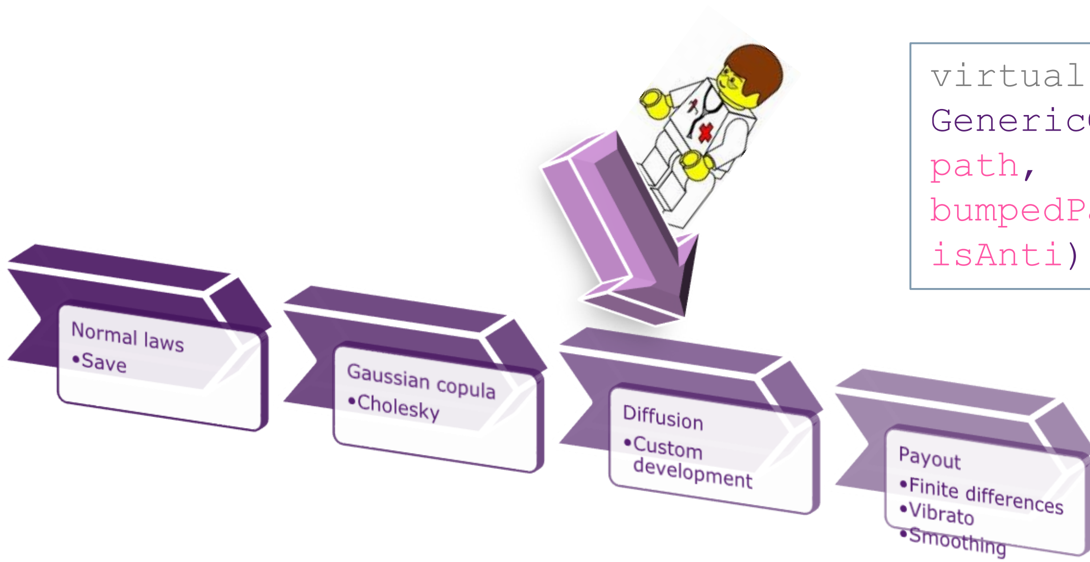
NATIXIS

Cega – Technical tricks

$$Price = Payoff\ o\ Diffusion\ o\ Calibration$$

$$< \nabla Price| = < \nabla Payoff\ | \nabla Diffusion\ | \nabla\ Calibration\ |$$

FD — AD — Not Needed




```
virtual void
GenericCalculateCega(
path,
bumpedPaths,
isAnti);
```

Cega – BS & VolLoc Theoretical calculus

- Computing $\overline{X_i(t)} = \frac{\partial P}{\partial X_i(t)}$

- Computing
  - $\overline{Z_i(t)} = \frac{\partial P}{\partial Z_i(t)}$
  - $= \frac{\partial P}{\partial X_i(t)} \cdot \frac{\partial X_i(t)}{\partial Z_i(t)} + \sum_{t'>t, t' date\ de\ constatation} \frac{\partial P}{\partial X_i(t')} \cdot \frac{\partial X_i(t')}{\partial M_i(t)} \cdot \frac{\partial M_i(t)}{\partial Z_i(t)}$
  - $= \boxed{\sigma_{i,t-1}\sqrt{t-(t-1)}.\left(\overline{X_i(t)}.A_i(t).M_i(t) + \sum_{t'>t, t' date\ de\ constatation} \overline{X_i(t')}.\frac{\partial X_i(t')}{\partial M_i(t)}.M_i(t)\right)}$
  - $\bar{L} = \sum_t \overline{Z}\, \tilde{Z}^T$

DeltaKT

VolLoc approximation
To get rid of numerical noise

NATIXIS

Cega – BS & VolLoc Theoretical calculus

Cega carried by BS

$$\frac{\partial X_i(t')}{\partial M_i(t)} = \frac{\partial M_i(t')}{\partial M_i(t)}.A_i(t')$$

$$= A_i(t') \prod_{t''=t}^{t''=t'-1} \frac{\partial M_i(t''+1)}{\partial M_i(t'')}$$

$$= A_i(t') \prod_{t''=t}^{t''=t'-1} exp\left(\frac{{\sigma_{i,t''-1}}^2(M_i(t''))}{2}((t''+1)-t-\sqrt{((t''+1)-t)}\sigma_{i,t''-1}(M_i(t''))Z_i(t'')\right)$$

$$+M_i(t'').\frac{\partial exp\left(\frac{\sigma_{i,t''-1}(M_i(t''))}{2}((t''+1)-t)-\sqrt{((t''+1)-t)}\sigma_{i,t''-1}(M_i(t''))Z_i(t'')\right)}{\partial M_i(t'')}$$

Cega carried by local volatility

Is it the end of grid computations?

- Not at all !!
- Finite Differences is a necessary benchmark
- New Regulations such as FRTB, MIFIDII, HIRE ACT II, PRIIPS, UK Prd Governance are demanding in terms of direct computation time

2 Greeks Duality

Gamma Vega in a Black Scholes Model

$$\partial_\sigma p = T\sigma S^2 \partial_{SS} p$$

- In a Black Sholes Model, **vega** and **gamma** are related by the formula above
- P. Carr & F. Mercurio & al showed many similar formula for **Homogeneous** models (stochastic volatility and jumps)
- Can be interpreted as a relationship between a **parameter sensitivity** (vega) and a **greek** (gamma)

- **What about local volatility type models?**

Gamma local Vega local in a Local Vol Model

$$\frac{dS}{S} = \sigma(t,S)dB$$

- local volatility and drift processes are local,

- **We have a local link between local vega (parameter sensitivity) and a local greek gamma**

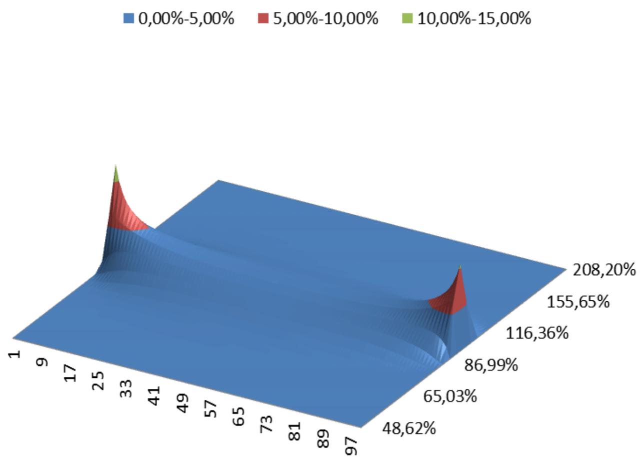
$$\frac{\partial p}{\partial \sigma(t,S)} = \varphi(t,S)\sigma(t,S)S^2\partial_{SS}p(t,S)$$

$\varphi(t,S)$ Density at point S at time t
Calculated using a forward pde

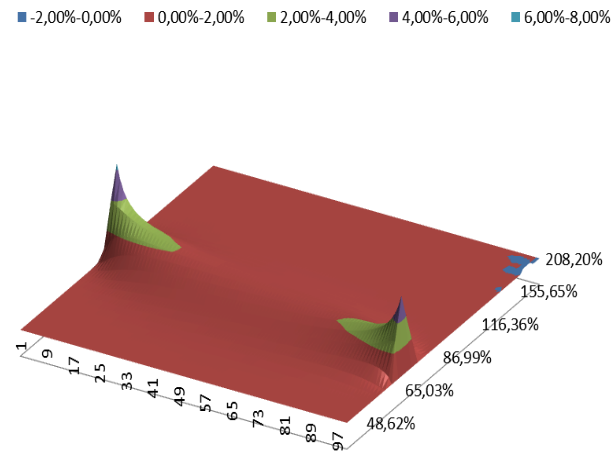
NATIXIS

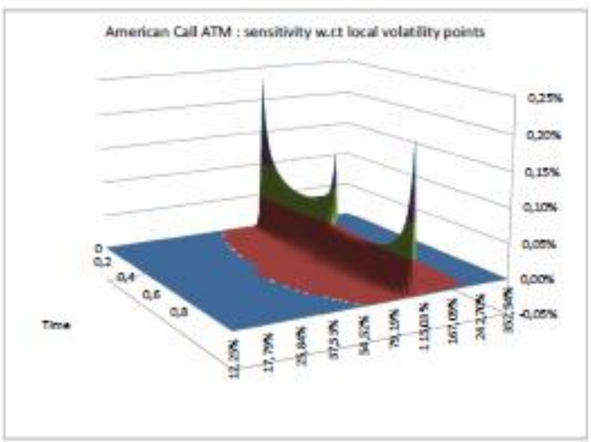
local vol sensitivity : Vanillas

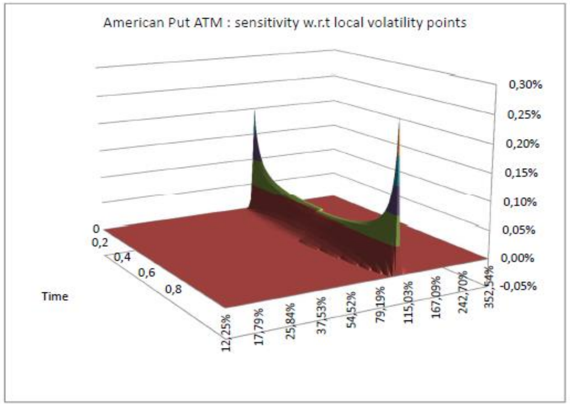
Call K=100, T=1y
Matrix sensitivity to all local volatility points



Call K=75, T=1y
Matrix sensitivity to all local volatility points







NATIXIS

We improve the Most Likely Path technique by introducing some convexity:

$$\Sigma_{TK}^2 \cong \frac{1}{T}\int_0^T E_{K,T}\sigma^2(t,S_t)dt$$

$$\cong \frac{1}{T}\int_0^T \sigma^2\left(t,E_{K,T}(S_t)\right)dt + \frac{1}{T}\int_0^T \frac{1}{2}Var_{K,T}(S_t)\frac{\partial\sigma^2}{\partial S^2}\left(t,E_{K,T}(S_t)\right)dt$$

$w(t,S_t) =$ is the result of one calculation

Cross Gamma vs local correlation in a Local Vol local correlation Model

$$\frac{dS_i}{S_i} = \sigma_i(t, S_i)dB_i$$

- local parameters including local correlation

- **We have also a local link between local correlation sensitivity (parameter sensitivity) and a local greek cross gamma**

$$\frac{\partial p}{\partial \rho_{kl}(t, \vec{S})} = \varphi(t, \vec{S}) S_k S_l \sigma_k(t, S_k) \sigma_l(t, S_l) \partial_{S_k S_l} p(t, S)$$

NATIXIS

Drift sensitivity vs local Delta

$$\frac{dS}{S} = \mu(t,S)dt + \sigma(t,S)dB$$

- local parameters

- **We have also a local link between local drift sensitivity (parameter sensitivity) and a local delta**

$$\frac{\partial p}{\partial \mu(t,S)} = \varphi(t,S)S\partial_S p(t,S)$$

3 Perturbation Techniques

AD for pricing – Fudge VolLoc 1/2

$$\partial_t p + \frac{1}{2}\sigma_{loc}^2 S^2 \partial_{SS} p = 0$$

Can be interpreted in terms of perturbations

$$\partial_t p + \frac{1}{2}\sigma_{BS}^2 S^2 \partial_{SS} p = -\frac{1}{2}\varepsilon(\sigma_{loc}^2 - \sigma_{BS}^2) S^2 \partial_{SS} p$$

Solution of the form: $p = p_0 + \varepsilon p_1$

$$\begin{cases} \partial_t p_0 + \frac{1}{2}\sigma_{BS}^2 S^2 \partial_{SS} p_0 = 0 \quad (1) \\ \partial_t p_1 + \frac{1}{2}\sigma_{BS}^2 S^2 \partial_{SS} p_1 = -\frac{1}{2}(\sigma_{loc}^2 - \sigma_{BS}^2) S^2 \partial_{SS} p_0 \quad (2) \end{cases}$$

AD for pricing – Fudge VolLoc 2/2

Using Feymann-Kac

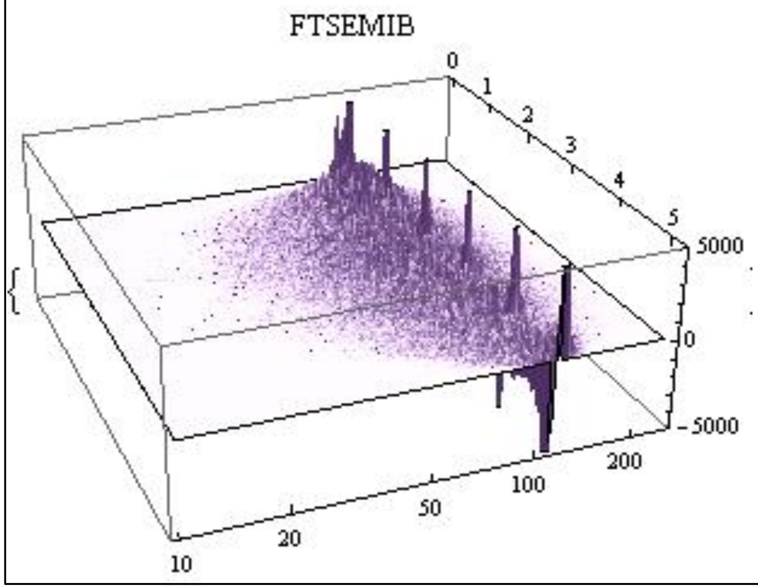
$$p = p_{BS} + \iint \frac{1}{2}\left(\sigma_{loc}^2 - \sigma_{BS}^2\right)\varphi(S)\partial_{SS}p_{BS}S^2 dtdS \ \ (*)$$

Same for $u = \frac{\partial p_{BS}}{\partial \sigma_{BS}^2(S,T)}, u = 0\ (boundaries)$

$$p_{LV} \approx p_{BS} + \iint \left(\sigma_{loc}^2 - \sigma_{BS}^2\right)\frac{\partial p_{BS}}{\partial \sigma_{loc}^2(S,T)} dSdT$$

NATIXIS

VegaKT LV and PnL Explain



A-Gamma Map

Vega KT Loc vol is useful to understand **where** the risk is located and its **nature:**

B-PnL Explain

$$p_{LVnew} - p_{LVold} \approx \iint \left(\sigma^2_{loc,new} - \sigma^2_{loc,old}\right) \frac{\partial p_{loc,old}}{\partial \sigma^2_{loc}(S,T)} dS dT$$

NATIXIS

LCM: variety of appraoches

- **Dupire :** Ito expansion normal dynamics
- **El Karoui-Durrelemann :** local regression
- **Avellaneda** : the most likely configuration
- **Langnau** : local moment matching approach
- **Sbai-Jourdan** : no explicit local correlation → deduce stocks vols from index and not index from stocks
- **Reghaï** : Fixed Point approach (could be slow)
- **Guyon/PHL/Piterbarg** : Iterative approach and Dupire formula
- **Bouchaud & al** : regression/data/Limit theorems
- **Delanoe** : mixing of Reghaï and Guyon/PHL/Piterbarg
- **Luci :** copula techniques

AD for pricing – Local correlation 1/3

The model

$$\frac{dS_i}{S_i} = \sigma_{i,loc}(t,S_i)\left(\sqrt{1-\varepsilon\lambda(t,S)}dW^{\rho} + \sqrt{\varepsilon\lambda}dW^{\rho\perp}\right)$$

With $\boldsymbol{\lambda} = f(t, S_1, \dots, S_n)$

Satisfies the following PDE

$$\partial_t p + \frac{1}{2}\sum_{i,j}\sigma_i\sigma_j S_i S_j\left((1-\varepsilon\lambda)\rho_{i,j} + \varepsilon\lambda\right)\partial_{S_i,S_j}p = 0$$

Solution of the form $p = p_0 + \varepsilon p_1$

NATIXIS

AD for pricing – Local correlation 2/3

$$\begin{cases} \partial_t p_0 + \frac{1}{2}\sum_{i,j} S_i S_j \sigma_i \sigma_j \rho_{i,j} \partial_{S_i,S_j} p_0 = 0 \quad (1) \\ \partial_t p_1 + \frac{1}{2}\sum_{i,j} S_i S_j \sigma_i \sigma_j \rho_{i,j} \partial_{S_i,S_j} p_1 = \frac{1}{2}\sum_{i,j} S_i S_j \sigma_i \sigma_j \big(\lambda(1-\rho_{i,j})\big) \partial_{S_i,S_j} p_0 \end{cases}$$

Which leads to

$$p_{LV,LC} = p_{LV,CC} - E\left(\int \frac{1}{2}\sum_{i,j} S_i S_j \sigma_i \sigma_j \lambda\left(1-\rho_{i,j}\right) \partial_{S_i,S_j} p_{VL,CC}\, dt\right)$$

AD for pricing – Local correlation 3/3

Let $u = \frac{\partial p_1}{\partial \lambda}$

With Feymann-Kac we obtain
for $\lambda = 0$

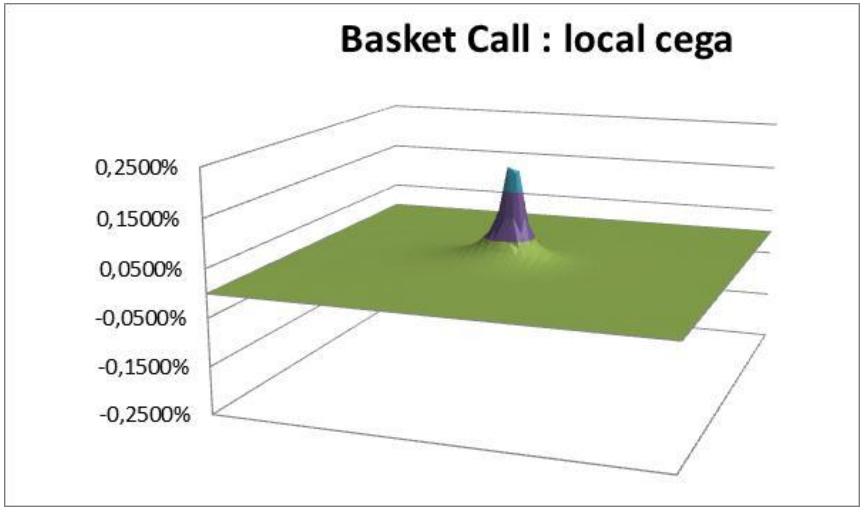
$$u = -\frac{1}{2}\sum_{i,j}\varphi(S_i)\varphi(S_j)S_iS_j\sigma_i\sigma_j\left(1-\rho_{i,j}\right)\partial_{S_i,S_j}p_{VL,CC}$$

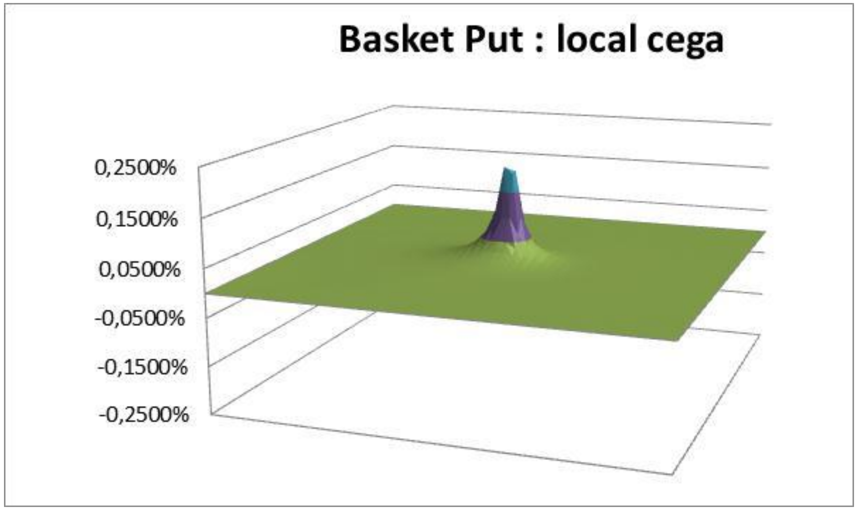
Finally

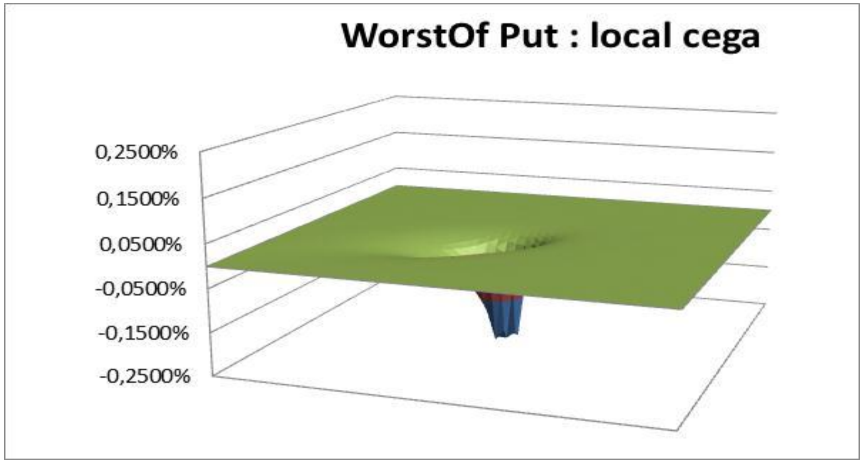
$$p_{LV,LC} \approx p_{LV,CC} + E\left(\int_0^T \lambda \frac{\partial p_{LV,CC}}{\partial \lambda} dt\right)$$

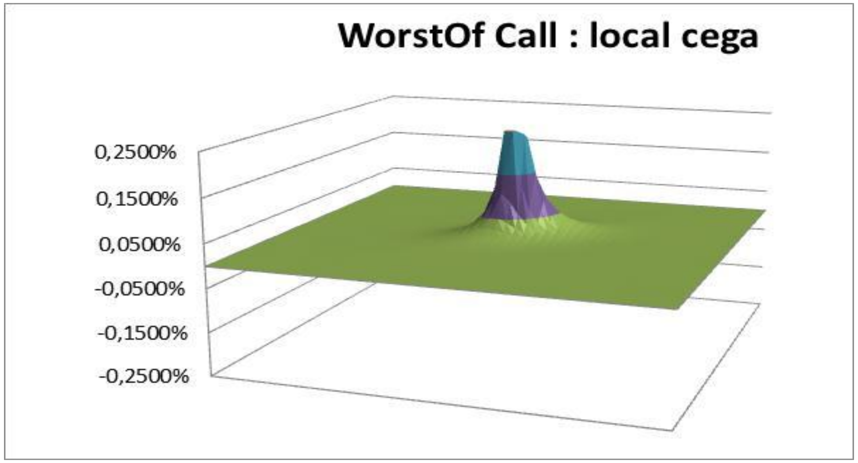
NATIXIS

2D exemple : Basket/Worst of









NATIXIS

4

Financial Applications

Adjust Prices to better prices

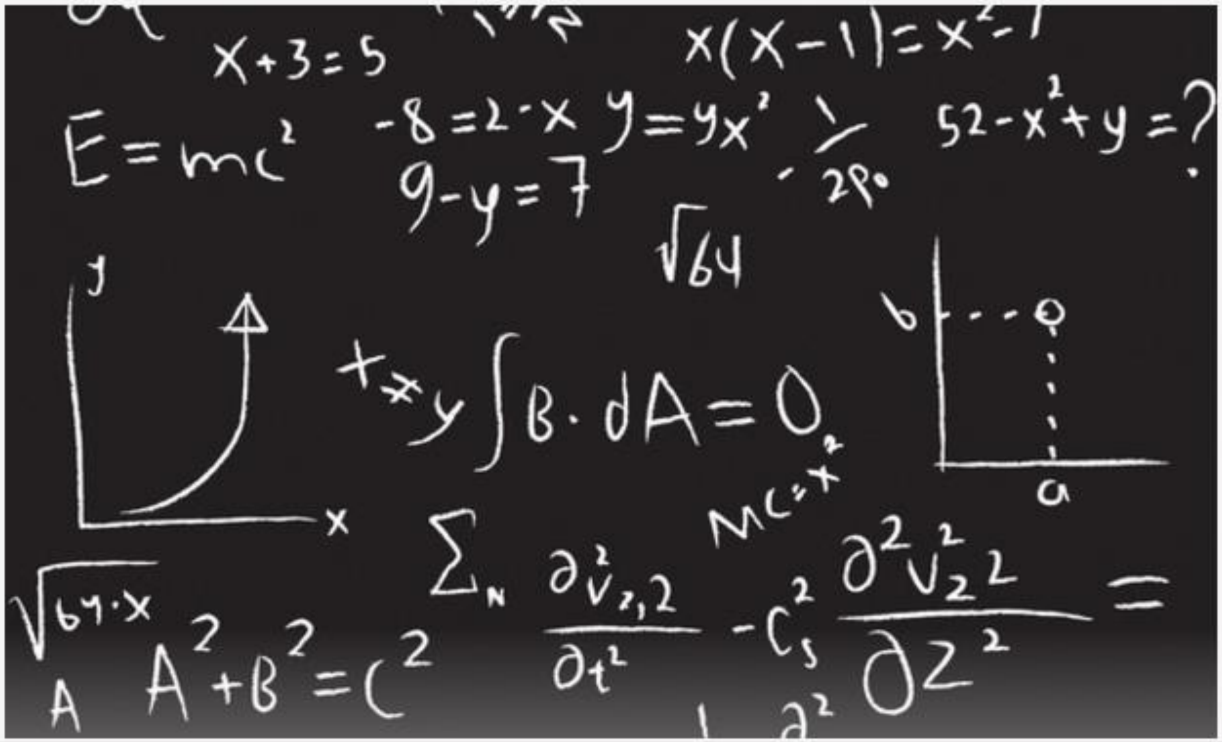
Example : CVA

NATIXIS

Reference Risk Nov. 2015 – subm. in 2012

CVA with Greeks and AAD

Reghai, Kettani and Messaoud present new technique to calculate CVA using adjoints



NATIXIS

CVA Problem

- Formula to calculate CVA adjustment:

$$CVA = \mathbb{E}(1_{\tau \leq T}(1-R)(\mathbb{E}_\tau \pi_T)^+)$$

- This can be solved using a non linear PDE

$$CVA = p_\beta(t,S) - p_0(t,S) \quad (3)$$

where $p(t,S)$ satisfies a non-linear PDE which can be written in a normal form. For more details [PHL].

$$\begin{aligned} \partial_t p + \mu S \partial_S p + \tfrac{1}{2}\sigma^2 S^2 \partial_{SS} p + \beta(p^+ - p) &= 0 \\ p(T) &= \pi_T(S) \end{aligned} \quad (4)$$

where $\beta = \lambda(1-R)$

NATIXIS

CVA Monte Carlo

Approach

- Exposure is calculated with the zeroth order contract price, This means that $\int_t^T \mathbb{E}_{t,S}[p^+(u,S_u)]\beta e^{-\beta(u-t)}du$ is approximated with $\int_t^T \mathbb{E}_{t,S}[p_0^+(u,S_u)]\beta e^{-\beta(u-t)}du$,
- The price in the future is the 0 order price plus the Ito integral:

$$p_0\left(t,S_t\right) = \texttt{price} + \int_0^t \frac{\partial p_0}{\partial S}\left(s,S_s\right)dS_s$$

- Pathwise delta are computed thanks to AAD and a link between computational sensitivities with respect to local drift of the process. Precisely, we obtain a relationship stating a link between the delta pathwise

$$\frac{\partial p_0}{\partial S}\left(t,S\right)$$

and the following sensitivity

$$\frac{\partial p_0}{\partial \mu\left(t,S\right)}$$

.

NATIXIS

CVA Monte Carlo

Estimator

A Monte Carlo estimator is given by the following equations:

$$
\begin{aligned}
p - p_0 &= \int_t^T \frac{1}{N_{paths}} \sum_{p>0} p\beta e^{-\beta(u-t)} du \\
&= \int_t^T \frac{1}{N_{paths}} \sum p 1_{p>0} \beta e^{-\beta(u-t)} du \\
&= \frac{1}{N_{paths}} \sum \int_t^T p 1_{p>0} \beta e^{-\beta(u-t)} du
\end{aligned}
$$

Duality greeks AAD

$$\frac{\partial p_0}{\partial t} + S\mu \frac{\partial p_0}{\partial S} + \frac{1}{2} S^2 \sigma_{\text{loc}}^2(t,S) \frac{\partial^2 p_0}{\partial S^2} = 0 \tag{14}$$

p_0 can be seen as a function of a whole surface of parameters $\mu(t,S)$. We can therefore, using adjoint techniques, produce all the sensitivities with respect to these inputs at a very small cost, not related to the number of points of discretisation. This means that for every point t_1, S_1 in the future, we have obtained numerically the quantity $\overline{p_0}\,(t_1,S_1) = \frac{\partial p_0}{\partial \mu(t_1,S_1)}$.

If we derive formally the previous 0 order PDE with respect to the parameters $\mu(t_1,S_1)$ we obtain:

$$\frac{\partial \overline{p_0}}{\partial t} + S\mu \frac{\partial \overline{p_0}}{\partial S} + \frac{1}{2} S^2 \sigma_{\text{loc}}^2(t,S) \frac{\partial^2 \overline{p_0}}{\partial S^2} = -\delta_{t-t_1, S-S_1} S_1 \frac{\partial p_0}{\partial S} \tag{15}$$

At this stage, we introduce the density function ϕ of the equity process (forward Kolmogorov)

$$\frac{\partial \phi}{\partial T} = \frac{1}{2} \frac{\partial^2 \sigma_{\text{loc}}^2(t,S) S^2 \phi}{\partial S^2} - \frac{\partial \mu S \phi}{\partial S} \tag{16}$$

and obtain from the previous equations the following relationship:

$$\overline{p_0} = \frac{\partial p_0}{\partial \mu\,(t_1,S_1)} = \phi\,(t_1,S_1)\, S_1 \frac{\partial p_0}{\partial S}\,(t_1,S_1) \tag{17}$$

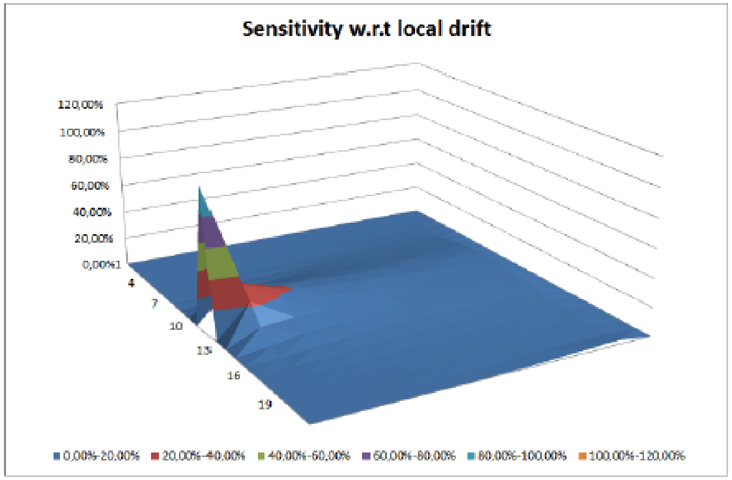
Drift sensitivity : AAD Rapid calculation

(12) $$logX_{t+dt} = logX_t + \mu(t, X_t)dt + \sigma(t, X_t)\sqrt{dt}\epsilon_t - \frac{1}{2}\sigma^2(t, X_t)dt$$

Where ϵ_t is a standard normal distribution.

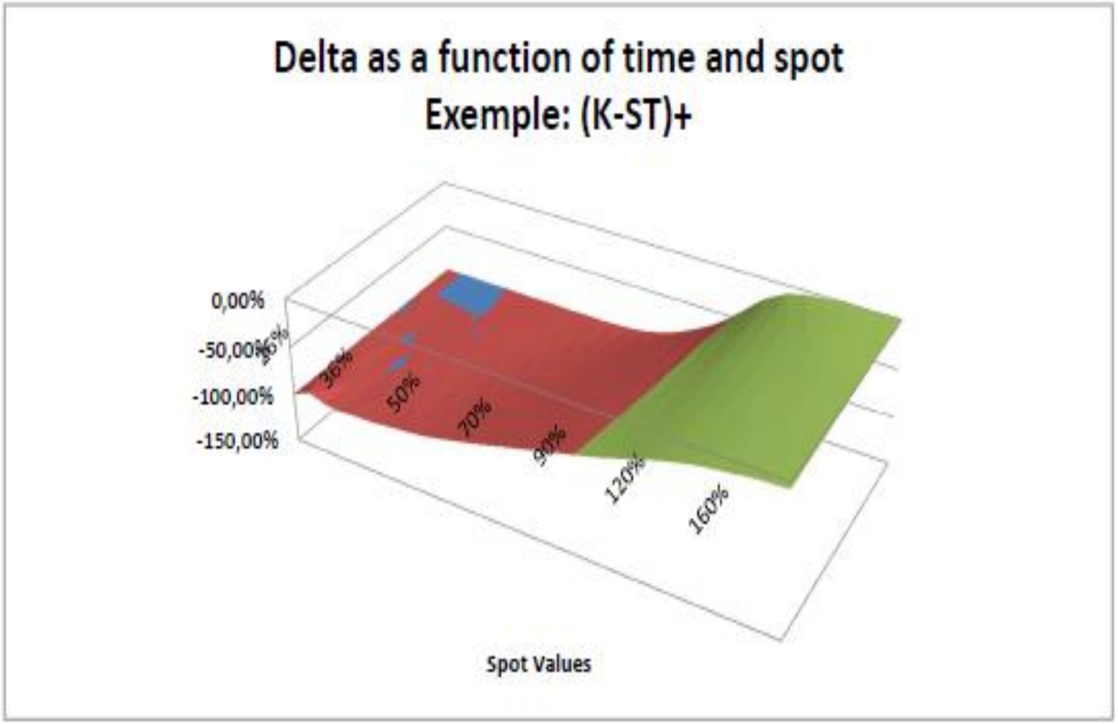
The AAD version of this code which takes into account the drift component can be written as follows:

(13) $$\overline{\mu}(t, X_t) = \overline{logX}_{t+dt}dt$$



NATIXIS

CVA Duality Numerical Verification



CVA martingale representation

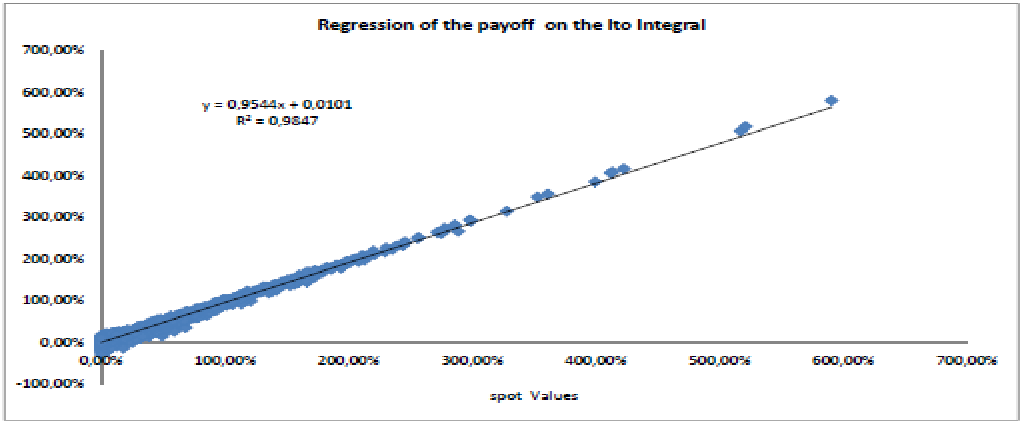
Martingale representation theorem

4.4. **Ito price reconstruction.** Finally, we can reconstruct the price thanks to the Ito integral:

$$p_0(t, S_t) = \texttt{price} + \int_0^t \frac{\partial p_0}{\partial S}(s, S_s)\, dS_s \tag{18}$$

We can also use the backward version:

$$p_0(t, S_t) = \psi(T, S_T) + \int_t^T \frac{\partial p_0}{\partial S}(s, S_s)\, dS_s \tag{19}$$



2 important consequences

- Automatic Control Variate

4.5. **Automatic Control Variate.** In this subsection, the martingale $\int_0^t \frac{\partial p_0}{\partial S}(u, S_u)\, dS_u$ is used as a control variate. We record the Monte Carlo speed up using this zero mean variable on some classical payoffs:

Payoff	Monte Carlo Speed Up
Forward	18000
Call	56
Put	15
Cliquet	4
Asian	26

TABLE 1. Results

- Alternative to LSM

Compute European product + AAD Drift + Duality → $(t, S) \to \frac{\partial p_E}{\partial S}(t, S)$

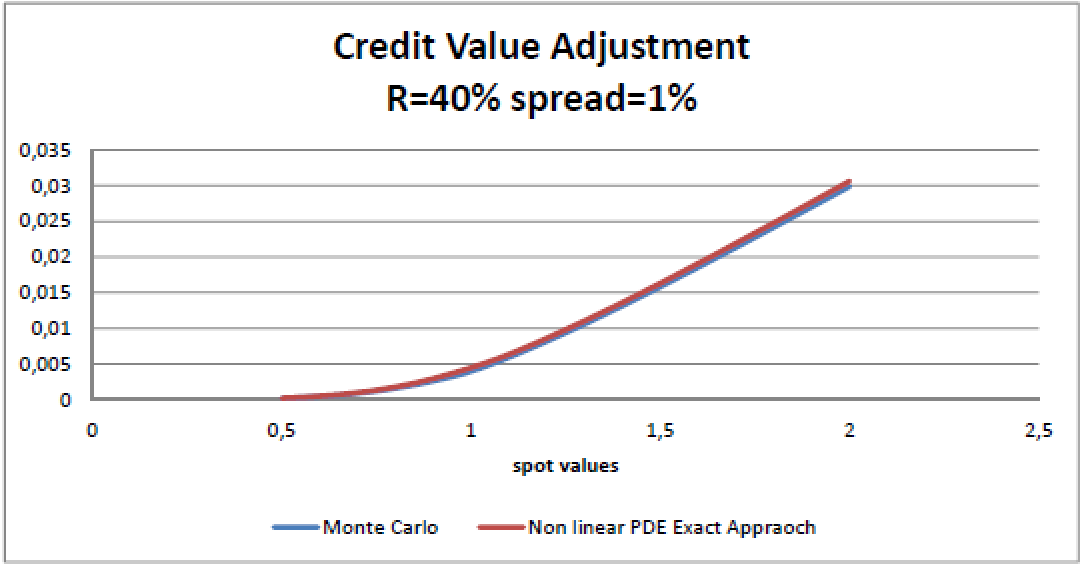
Estimate future prices using the mart. rep. → $p_E(t, S) = price + \int_0^t \frac{\partial p_E}{\partial S}(u, S) du$

Early exercise approach → $Max(p_E(t, S), early\ exercise) \to p_A(t, S)$

NATIXIS

CVA numerical application

- **Comparing the Non linear PDE with the proposed approach shows excellent results**



NATIXIS

5 Conclusion

AD – Conclusions

- Benefits of this Revolution
  - Implementation is an engineering task
  - Cega : Very good computation time (+10% of a single pricing for complete structure)
  - AD Combine different techniques (finite diff, tangent, adjoint) and library needs to evolve
  - New techniques for Automatic Control Variates / Early exercise value
  - Perturbation techniques → Adjust prices to better prices (improved price and its greeks at the same time)

Questions