Comparison of some improved methods to calculate american option in high dimension

Seminaire FiME

Xavier Warin

EDF R&D/ Laboratoire FIME

7 may 2010

joint work with Bruno Bouchard

Plan

1. Motivation
2. Regression based method
3. Malliavin method
3. Quantization method
4. Numerical results

Motivation

Many numerical methods to evaluate american options. Among them :

- Regression methods (Tsiltiklis Van Roy, Longstaff Schwarz)
- Quantization (Pages-Bally-Printemps)
- Malliavin (Bouchard-Touzi)

Question :

- How to use them efficiently in high dimension,
- Which one to choose depending on the dimension of the problem ?

Problematic

Assets following BS model under risk neutral probability :

$$S_t^i = S_0^i \exp\left(rt - \frac{1}{2}\sigma^{i,2} t + \sigma^i W_t^i \right), \quad i = 1, L \tag{1}$$

with $\rho_{i,j} = corr(W_t^i, W_t^j)$.
Valuation of

$$V = \sup_{\tau \in [0,T]} \mathbb{E}\left(exp(-r\tau(\omega) f(S^1_{\tau(\omega)}, ..., S^L_{\tau(\omega)}, K) \right) \tag{2}$$

with f pay off function depending on strike

Example of classical pay off

Basket Put Option :

$$f(S_t^1, .., f_t^L, K) = (K - \frac{1}{L}\sum_{i=1}^{L} S_t^i)^+ \tag{3}$$

Difficulties when $L > 3$ for classical deterministic methods (binary trees, PDE)

Classical algorithm based on dynamic programming

Based on the discretized values of $S_t^k, k = 1, L$ obtained by MonteCarlo, recombining trees etc and using some time discretization $t_i = i\Delta t, i = 1$ to N.

$V_N^j = f(S_T^{1,j}, ..., S_T^{N,j}, K) \quad \forall j \in (1, M)$

for $i = N \;\; 1$ // Backward resolution **do**

  for $j = 1 \;\; M$ // iterate on discretized assets values **do**

$$V_{i-1}^j = \max\Big(f(S_{(i-1)\Delta t}^{1,j}, ..., S_{(i-1)\Delta t}^{L,j}, K), \; exp(-r\Delta t)\,\mathbb{E}\left(V_i|\mathcal{F}_{(i-1)\Delta}\right)\Big)$$

  end for

end for

return $V = V_0^j$ j quelconque

Modification

Possibility to approximate stopping time :

- while optimizing with Longstaff Schwarz method with Monte Carlo methods,
- with a forward simulation with quantization (using continuation values calculated in a first step).

Conditional expectation need to be calculated

Due to markovian properties of S^j_{i-1}

$$\mathbb{E}\left(V_i|\mathcal{F}_{(i-1)\Delta}\right) = \mathbb{E}\left(V_i|S^1_{(i-1)\Delta t}, ..., S^L_{(i-1)\Delta t}\right) = \mathbb{E}^i V_i \qquad (4)$$

Conditional expectation need to be calculated in dimension L. Use

- Regression
- Quantization
- Malliavin.

Regression basis function

- X_t markovian process, $\mathbb{E}^i(f(X_{(i+1)\Delta t}))$ function F of $X_{i\Delta t}$
- Conditional expectation = projection :

$$F = \arg \min_{g \in L_2(\Omega)} \mathbb{E}((f(X_{i+1}) - g(X_i))^2) \tag{5}$$

- Use function basis to approximate F

$$F(x) = \sum_{j=0}^{J} \alpha_j \psi_j(x)$$

where α_j is the coordinate associated to the j basis function

Regression classical (II)

Using a Monte Carlo method , paths $X_i^{(k)}$ have been generated for $k \in [1, M]$. At each time step i, noting $A \in \mathbb{R}^M \times \mathbb{R}^J$ such that $A_{k,j} = \psi_j(X_i^{(k)})$ and $B \in \mathbb{R}^M$ such that $B_k = f(X_{i+1}^{(k)})$ the equation 5 discretized can be rewritten as

$$\min_{\alpha \in \mathbb{R}^J} \|A\alpha - B\|^2 \tag{6}$$

Use :

- Normal equation $A^* A\alpha = A^* B$
- QR factorization Q rotation, R upper triangular, $\mathbb{R}\alpha = Q^* B$
- Single Value Decomposition $A = UWV^*$, U, V rotations ,W diagonal $\alpha = V[diag(1/w_i)]U^* B$

Usual basis function

Hermite, Hyperbolic and Chebyshev polynomials usually used

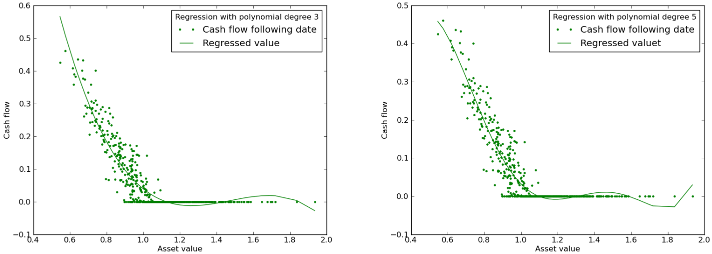
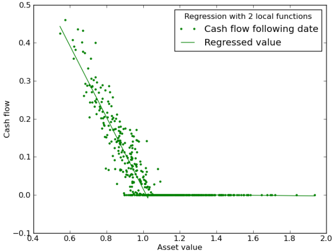


Figure: Regression with global function : regression of a put final pay-off against previous asset value

Choosing the right basis

- Use local basis function with linear approximation (non conform finite element)
- Support are choosen so that they contain roughly the same number of particules (quantization)



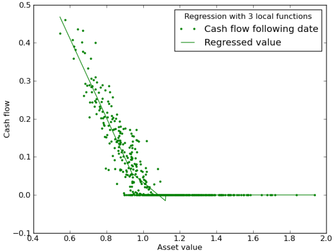


Figure: Regression with local function : regression of a final put pay off against previous asset value

Example of function basis support

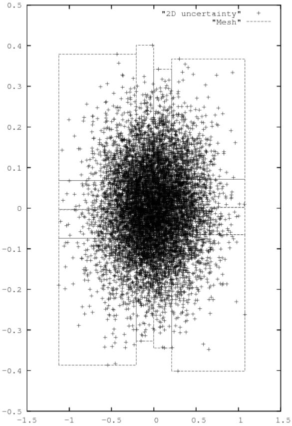


Figure: function basis support

Choosing the right basis

Advantage :

- Totally adaptative,
- No special guess of function basis
- Does not degrade as you increase the function number reasonnabily
- Give stable normal equation very sparse.
- Only way to have very accurate conditional expectation (SOBSDE)

Extra cost : need a sorting method to create the support. Using KD tree, achieved in $N \log N$. Complexity in $O(S(JM + MlogM)))$ for basis function regression

Conditionnal expectation by Malliavin for gaussian process

$$\mathbb{E}\left(f(X_{i+1})|X_i = x\right) = \frac{\mathbb{E}(\delta_x(X_i)f(X_{i+1}))}{\mathbb{E}(\delta_x(X_i))}$$

Integrating by part first in dimension l and then in all direction successively using :

- X_i^l gaussian , density $g_i^l(y)$
- $X_{i+1}^l - X_i^l$ gaussian, density $g_{i,i+1}^l(y)$

$$\begin{aligned}
\mathbb{E}(\delta_x(X_i)f(X_{i+1})) &= \int\int \delta_x(y)f(y+z)g_i(y)g_{i,i+1}(z)dydz \\
&= \mathbb{E}(1_{X_i>x}f(X_{i+1})\prod_l(\frac{X_i^l-\mu^l t_i}{\sigma^{2,l}t_i} - \frac{X_{i+1}^l-X_i^l-\mu^l(t_{i+1}-t_i)}{\sigma^{2,l}(t_{i+1}-t_i)})) \\
&= \mathbb{E}(1_{X_i>x}f(X_{i+1})\prod_l(\frac{W_i^l}{t_i} - \frac{W_{i+1}^l-W_i^l}{t_{i+1}-t_i})) \\
\mathbb{E}(\delta_x(X_i)) &= \mathbb{E}(1_{X_i>x}\prod_l(\frac{W_i^l}{t_i})) = \mathbb{E}(1_{X_i>x}\prod_l(\frac{W_i^l}{t_i} - \frac{W_{i+1}^l-W_i^l}{t_{i+1}-t_i}))
\end{aligned}$$

Monte Carlo for Malliavin

$$\mathbb{E}\left(f(X_{i+1})|X_i = x\right) \simeq \frac{\displaystyle\sum_{k=1}^{M} 1_{X_i^{(k)} > x} f(X_{i+1}^{(k)}) \prod_l \left(\frac{W_i^{l,(k)}}{t_i} - \frac{W_{i+1}^{l,(k)} - W_i^{l,(k)}}{t_{i+1} - t_i} \right)}{\displaystyle\sum_{k=1}^{M} 1_{X_i^{(k)} > x} \prod_l \left(\frac{W_i^{l,(k)}}{t_i}\right)}$$

1. Convergence rate in $N^{-\frac{1}{2}}/\pi^{d/4}$ explodes as the time step goes to zero.
2. Weights do not depend on process parameters,
3. Naive cost :
   - $O(M)$ operations for $x = X_i^p$,
   - Global cost in $O(M^2)$.

Variance reduction by localization (Bouchard Eykeland Touzi)

Reduce the variance by localization method . Use optimal localization function $\phi(y) = exp(-\eta y)$ with $\phi(0) = 1$,

$$\begin{aligned} \mathbb{E}(\delta_x(X_i) f(X_{i+1})) &= \int\int \delta_x(y)\phi(y-x) f(y+z) g_i(y) g_{i,i+1}(z) dydz \\ &= \mathbb{E}(1_{X_i > x} f(X_{i+1}) \textstyle\prod_l \left(\phi(X_i^l - x^l)(\frac{W_i^l}{t_i} - \frac{W_{i+1}^l - W_i^l}{t_{i+1} - t_i}) - \phi'(X_i^l - x^l)\right)) \end{aligned}$$

$$\mathbb{E}\left(f(X_{i+1}) | X_i = x\right) \simeq \frac{\sum_{k=1}^{M} 1_{X_i^{(k)} \geq x_i} f(X_{i+1}^{(k)}) \prod_l exp(-\eta X_i^{l,(k)})(\frac{W_i^{l,(k)}}{t_i} - \frac{W_{i+1}^{l,(k)} - W_i^{l,(k)}}{t_{i+1} - t_i} + \eta)}{\sum_{k=1}^{M} 1_{X_i^{(k)} \geq x_i} \prod_l exp(-\eta X_i^{l,(k)})(\frac{W_i^{l,(k)}}{t_i} - \frac{W_{i+1}^{l,(k)} - W_i^{l,(k)}}{t_{i+1} - t_i} + \eta)}$$

At each step, calculate for $x = X_i^{(k)}$ for $k = 1, M$.

Algorithm for dominance problem

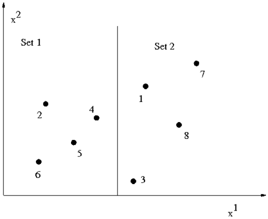
Need to calculate efficiently :

$$g(X^{(p)}) = \sum_{k=1}^{M} 1_{X^{(k)} \geq X^{(p)}} f^{(k)}$$

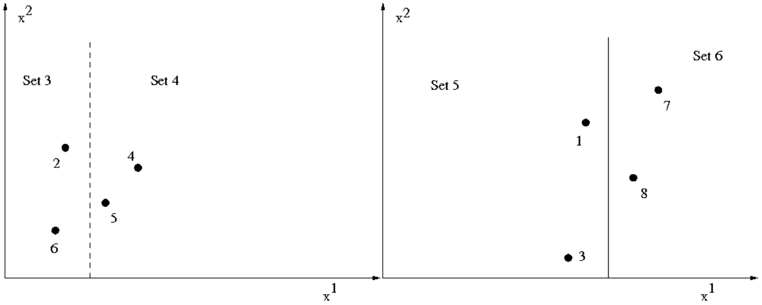
Crude cost of calculating $O(M^2)$
In 1D, sort $X^{(1)} \leq X^{(2)} \leq ... \leq X^{(M)}$ in $O(MlogM)$, then use partial sommations

$psum = 0$
for $p = M$ to 1 **do**
  $psum = psum + f^{(p)}$
  $g(X^{(p)}) = psum$
end for

Dominance problem in 2D



- Sort according to x coordinate,
- Split the points into two groups with the same cardinality Set 1 = (6,2,5 4) and Set 2 = (3,1,8,7) :
- Compare the points from Set 1 and 2 according to y coordinate while keeping the partial summation :
  - set *psum* to zero, point 7 (highest y coordinate of set 2) dominates all points of set 1 , add $f^{(7)}$ to *psum*.
  - Point 1 dominates points all points of set 1 : add $f^{(1)}$ to *psum* .
  - Point 8 doesn't dominate points 2 and 4 : add *psum* to $g(x^2)$,$g(x^4)$, add $f^{(8)}$ to *psum*.
  - The last one point 3 doesn't dominate any points of set 1 : add *psum* to $g(x^5)$ and $g(x^6)$.



- Split recursively set 1 into set 3 and 4 and set 2 into set 5 and 6.
- We apply the same procedure as before.

For set 3

- $psum = 0$.
- Point 4 with (highest y coordinate of set 4) doesn't dominate point 2 of the first set 3 : add $f^{(4)}$ to $psum$.
- Point 5 does the same : add $f^{(5)}$ to $psum$.
- Add $psum$ to the g value of remaining point 6 of set 3.

Idem set 4,5,6. Recursively

Complexity of Malliavin conditional expectation

In 2D :

- 2 sort in X and y $O(MlogM)$,
- Main recursive procedure in $O(MlogM)$.
  - Binary tree with depth $O(log(M))$
  - At each level $O(M)$ calculation.

Recursive algorithm in $d > 2$ dimension, with cost $Mlog(M)^{d-1}$

Number of particules	1D	2D	3D	4D	5D	6D	7D	8D	9D
10.000	0.	0.01	0.07	0.22	0.48	0.78	1.08	1.32	1.52
100.000	0.01	0.13	1.05	3.94	9.85	18.95	29.96	41.04	50.3
1.000.000	0.17	1.92	15.2	62.24	178.45	396	717	1110	1518

Table: Time spend in seconds to calculate g function for a given set of points

For d$=$ 4, numerically cost between $O(Mlog(M)^2)$ and $O(Mlog(M)^3)$. For d$=$ 9, numerically cost in $O(MlogM^6)$.

Quantization principle (Bally, Pages, Printems)

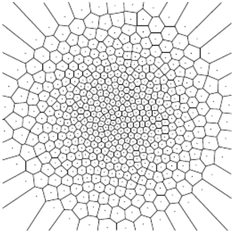


Figure 1: *A 500-tuple with its Voronoi tessellation with the lowest quadratic quantization error for the bi-variate normal distribution.*

- find best approximation of X random vector taking at most M fixed values $x^1, ..x^M \in \mathbb{R}^d$.

- $x = (x^1, .., x^M) \in (\mathbb{R}^d)^M$ given , $(C_i(x))_{i=1,M}$ partition of $\mathbb{R}^d$ s.t.

$$C_i(x) \subset \{\xi \in \mathbb{R}^d, \\ \|\xi - x^i\| \leq \min_{i \neq j} |\xi - x^j|\}$$

- $E(f(X)) \simeq \sum_{i=1}^M f(x^i) P(X \in C_i(x))$,
- X gaussian : optimal points available on web site *www.quantize.maths-fi.com* for different point numbers.

Use for american options (Bally, Pages, Printems)

- Choose number of time step,
- Choose number of points at each time step $M_k = \left[\frac{t_k^{\frac{d}{2(d+1)}}(M-1)}{t_1^{\frac{d}{2(d+1)}} + \dots t_S^{\frac{d}{2(d+1)}}} \right]$ so that $M_0 = 1$ and $M \leq M_0 + M_1 + ..M_S \leq M + S$
- Calculate the transition probability $P(X \in C_j(x)/Y \in C_i(y))$:
  - Use Monte Carlo to estimate,
  - Accelerate with Principal Axis Tree method (McNames)

Benchmarks for bermudean options

- d asset with same caracteristics $S_0^i = 1$, $\sigma^i = 0.2$, no correlation
- Maturity $T = 1$, interest rate r,
- Exercice dates $j\Delta t$, $j = 0$, $T/\Delta t = 10$,
- pay off :
  - $(K - \prod_{i=1}^{d} S_T^i)^+$ geometric american put , with strike $K = 1.$, reference with 1D tree method,
  - $1_{K \geq \prod_{i=1}^{d} S_T^i}$ digital american put, with strike $K = 0.9$ reference with 1D PDE method,
  - $(K - \frac{1}{d}\sum_{i=1}^{d} S_T^i)^+$ basket american put with no reference given

General consideration on the benchmark

- no specific use of some controle variate method (can be used by each method),
- no use of specific knowlegde of the pay off (don't use pay off as regressor in regression method.

Malliavin

- β parameter of malliavin fonction equal to $1/\sqrt{\Delta t}$,
- plot option value with Longstaff Schwarz (stopping time), with Tsiltiklis Van Roy depending on number of particules

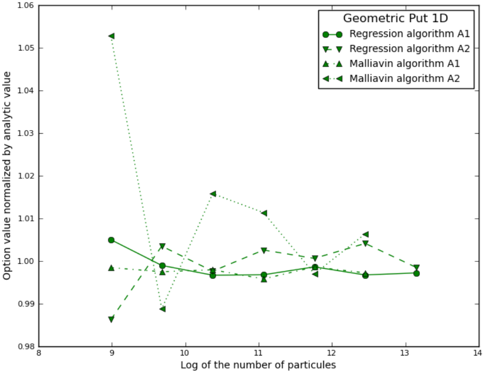
Fonction regression

- 8^d function basis,
- plot option value with Longstaff Schwarz (stopping time), with Tsiltiklis Van Roy depending on number of particules

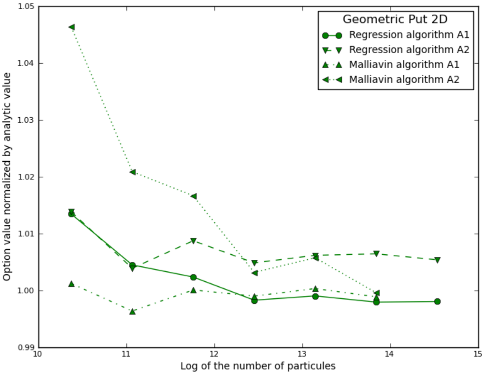
Quantization

- plot the value obtained for different points, 4000000 particules taken for probability calculations,
- plot the forward value obtained with 4000000 others MonteCarlo realizations.

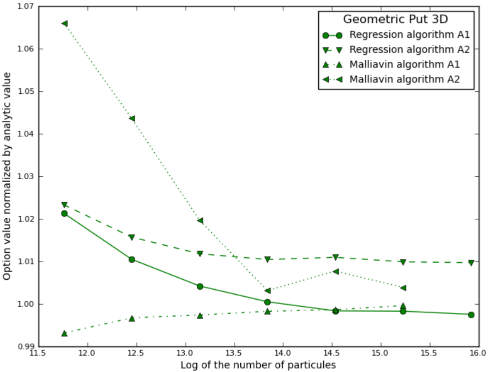
Comparison between Monte Carlo methods for american geometric put option 1D (algo A 1 = Longstaff Schwarz, algo A 2 = Tsiltiklis Van Roy)



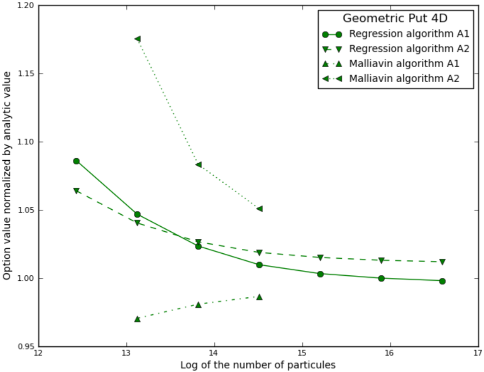
Comparison between Monte Carlo methods for american geometric put option 2D (algo A 1 = Longstaff Schwarz, algo A 2 = Tsiltiklis Van Roy)



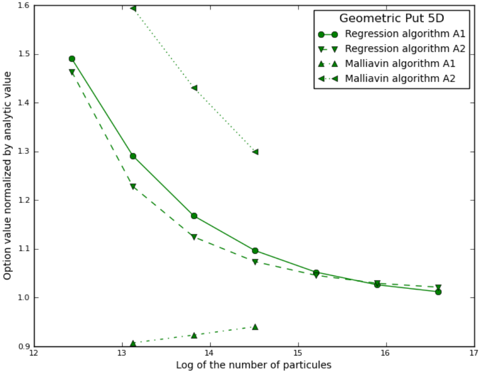
Comparison between Monte Carlo methods for american geometric put option 3D (algo A 1 = Longstaff Schwarz, algo A 2 = Tsiltiklis Van Roy)



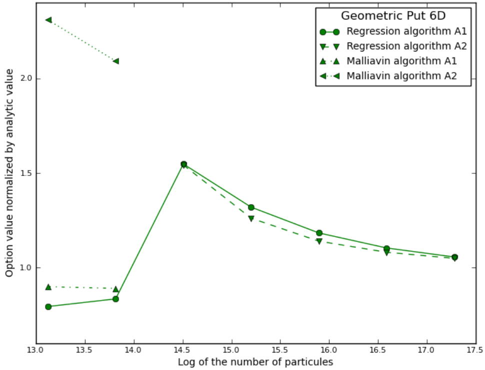
Comparison between Monte Carlo methods for american geometric put option 4D (algo A 1 = Longstaff Schwarz, algo A 2 = Tsiltiklis Van Roy)



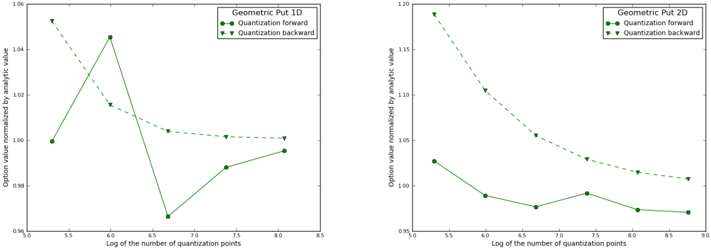
Comparison between Monte Carlo methods for american geometric put option 5D (algo A 1 = Longstaff Schwarz, algo A 2 = Tsiltiklis Van Roy)



Comparison between Monte Carlo methods for american geometric put option 6D (algo A 1 = Longstaff Schwarz, algo A 2 = Tsiltiklis Van Roy)



Convergence for quantization

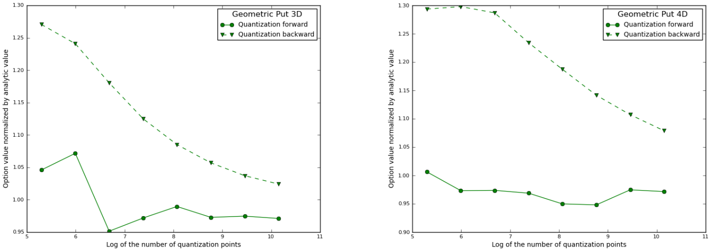


(a) 1D

(b) 2D

Figure: Quantization for american geometric put option 1D, 2D

Convergence for quantization

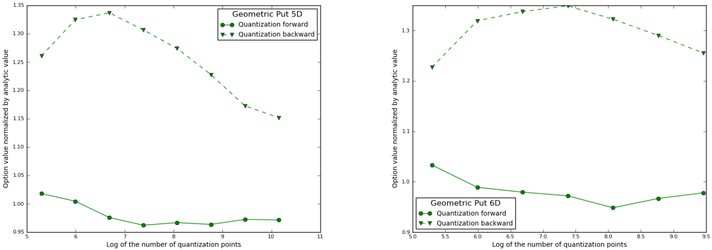


(a) 3D

(b) 4D

Figure: Quantization for american geometric put option 3D, 4D

Convergence for quantization



(a) 5D (b) 6D

Figure: Quantization for american geometric put option 5D, 6D

Time spend for calculation of Malliavin and Regression based approach for different number of particule

Dimension 10^3 particules log of particules	1D 8 8.98	1D 256 12.45	2D 256 12.45	2D 1024 13.84	3D 256 12.45	3D 2000 14.50	6D 1000 13.81	6D 2000 14.50
Regression	0.025	0.80	0.38	6.6	2.45	20.8	36.	65.
Malliavin	0.020	0.95	1.03	23.5	31.	360.	3650.	9080.

Efficiency for a given accuracy

- Dimension 1 Malliavin clearly superior to regression, Dimension 2 Malliavin seems to be superior to regression
- Quantization very quick when probabilities calculated BUT need far more points for dimension > 2,
- Dimension ≥ 3 only regression method can compete,

Similar results with digital options.

Delta with Longstaff Schwarz for Monte Carlo methods :

Two representations for Delta

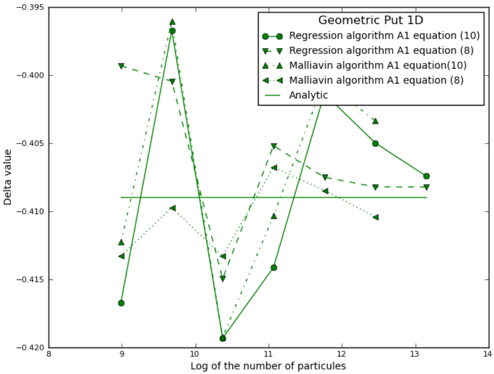
- Tangent process (equation (8)) not usable for digital options :

$$\phi^l = \mathbb{E}\left(exp(-r\tau(\omega)) \frac{\partial f(S^1_{\tau(\omega)}, ..., S^L_{\tau(\omega)}, K)}{\partial S^l} \frac{\partial S^l_{\tau(\omega)}}{\partial S^l_0} \right) \quad (7)$$

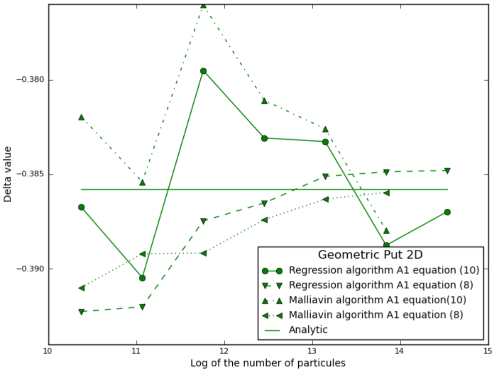
- Malliavin : consider an option with pay off $V_{\Delta t}$ at maturity Δt. Using martingale property of V_t value (equation (10))

$$\phi^l = \mathbb{E}\left(exp(-r\tau(\omega)) f(S^1_{\tau(\omega)}, ..., S^L_{\tau(\omega)}, K) \frac{W^l_{\Delta t}}{\Delta t \sigma^l S^l_0} \right) \quad (8)$$

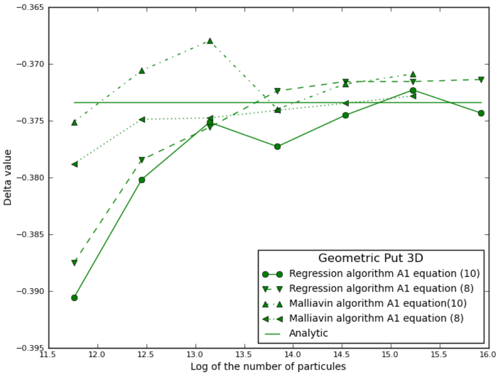
Delta results for geometric option 1D



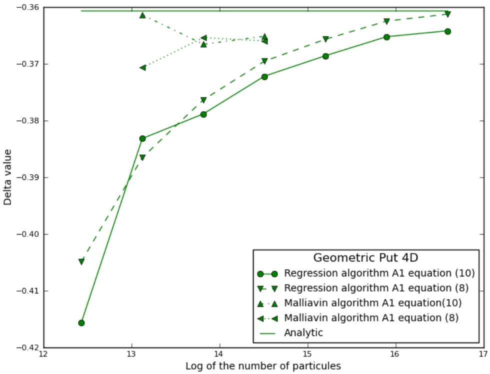
Delta results for geometric option 2D



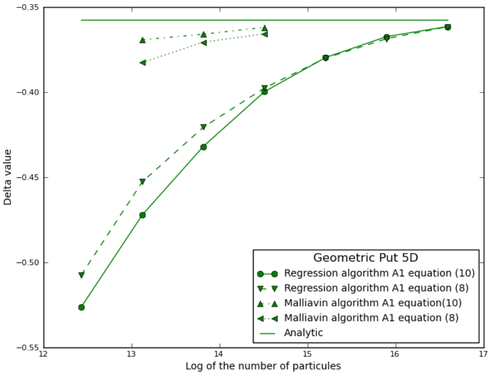
Delta results for geometric option 3D



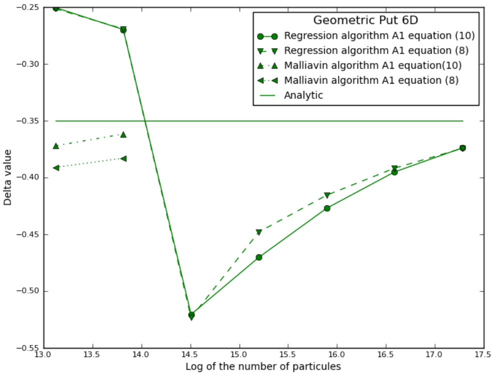
Delta results for geometric option 4D



Delta results for geometric option 5D



Delta results for geometric option 6D



Conclusion on Delta

- Both Delta representation give very good results
- In dimension 1 and 2, very similar results for Malliavin and regression for each Delta representation,
- In dimension 3 and 4 Malliavin converge faster for this option (not seen with digital options)
- The cost of calculation confirm the results observed for value options.
- As seen in dimension 2, Delta Malliavin (equation 10) seems to have a higher variance.