Using a PDE solver for pricing multi-assets options

Olivier Bokanowski*

In this projet a multidimensional PDE solver (ROC-HJ solver [2]) will be used in order to approximate a multi-asset option, such as European options. The aim of the project is to study the limit of a direct full grid approximation for some basic options in finance.

In this project a particular fully explicit semi-Lagrangian method (a kind of finite difference method) is considered.

It is prefered that a linux OS (or mac OS with linux) architecture is used with the Library.

1) In a first part we consider a multi-dimensional PDE of the form

$$-v_t - \frac{1}{2}Tr(\sigma\sigma^T D^2 v) - \sum_i r x_i v_{x_i} + rv \equiv -v_t + \mathcal{A}v = 0 \tag{1}$$

and

$$v(T,x) = \varphi(\sum \alpha_i x_i).$$

where $\alpha_i > 0$ are given, and $\varphi(y) := \max(K-y,0)$ (put like form). First find coefficients (σ_{ij}) such that for u an analytical solution is known, of the form $v(t,x_1,\ldots,x_n) = w(t,y)$ where $y = \sum_i \alpha_i x_i$, and where v is solution of a one-dimensioal Black and Scholes equation (the coefficients $\sigma_{ij}(x) = x_i\beta_{ij}$ where β_{ij} are constants to be determined so that the above property is true).

This particular problem is then used as a reference test case for multi-dimensional approximation.

2) Let $u^n(x)$ represents an approximation of the value $v(t_n,x)$, where $t_n = nh$ and $h = T/N$ is a time step.
(i) We first look for u^n in the form:

$$u^N(x) = \varphi(x) \tag{2a}$$

and, for $n = N-1,\ldots,0$:

$$u^n(x) = \frac{e^{-rh}}{2d}\sum_{k=1,\ldots,d}\sum_{\epsilon=\pm1} u^{n+1}(x + \bar{b}_k(x)h + \epsilon\bar{\sigma}_k(x)\sqrt{h}) \tag{2b}$$

Show that for well chosen vectors $\bar{\sigma}_k > 0$ and vectors $\bar{b}_k(x)$ (give simple equations they have to satisfy, in relationship with σ_k and $b_k(x)$) we obtain a first order scheme in the sense that the following consistency error estimate holds, for any $v \in C^4([0,T]\times\mathbb{R}^+)$, with $v^n(x) = v(t_n,x)$, solution of (1),

$$|\frac{v^n(x) - (Sv^{n+1})(x)}{h}| \le Ch,$$

*boka@math.jussieu.fr

where the constant C depends of the derivatives of v (to be made precise). Deduce then that the true error is of first order in the sense that (assuming v sufficiently regular):

$$|u^n(x) - v^n(x)| \leq Ch.$$

(ii) The implemented scheme will be of the form

$$u^n(x) = \frac{e^{-rh}}{2d} \sum_{k=1,\ldots,d} \sum_{\epsilon=\pm 1} [u^{n+1}](x + b_k(x)h + \epsilon\sigma_k(x)\sqrt{\gamma_k h}) \tag{3}$$

where $[\psi]$ denotes a P_1 or Q_1 interpolation on a cartesian mesh (in particular $|[\psi](x) - \psi(x)| \leq \Delta x^2$ for any C^2 regular function ψ, and $|[\psi](x) - \psi(x)| \leq L\Delta x$ for any Lipschitz-continuous function ψ). Analyse again the consistency error as well a the true error of the scheme in this case, assuming that $\Delta x > 0$ is a uniform mesh step in any direction for the mesh grid (Show a supplementary error term of the form $+\frac{\Delta x^2}{h}$ in the regular case).

3) By using the ROC-HJ library, implement and test this scheme in the two-dimensional case. Check the accuracy order (typically using $\Delta t \equiv \Delta x \to 0$), CPU times.

4) Test the scheme on higher-dimensions, (In particular one may look for the limit of full grid approximation, in terms of CPU computational time, and in terms of memory limit capacity).

5) If time allows, higher order schemes can be studied or approximation of american options, or error approximation in the only Lipschitz continuous case.

References

[1] K. Debrabant et Epsen. R. Jakobsen, Preprint 2009.
Voir `http://www.math.ntnu.no/~erj/publications.imf`

[2] The "ROC-HJ" solver: a parallel c++ library for Reachability and Optimal Control software using the Hamilton-Jacobi approach. `http://itn-sadco.inria.fr/software/ROC-HJ`