Multi-asset option pricing using Black-Scholes PDE in Tensor-Train Format

Leonid Shartser

March 28, 2021

Abstract

We provide a detailed reference on how TT-Format can be used for pricing multi-asset options. The goal is to present alternative methods to Monte-Carlo simulations for the high dimensional setting. The paper is written from a practitioner point of view.

1 Introduction

There are two main parameters to control the accuracy of the solution.

1. The tolerance of approximation of payoff
2. (a) smoothing parameter
   (b) epsilon for rounding
   (c) number of approximating iterations
3. The tolerance of the linear solver (solving for a solution at time step t_{i-1} from the solution at time t_i).

2 Multivariate Black-Scholes PDE

3 Tenosrs and TT-Format

4 Non smooth approximation

Low rank tensor approximation method for functions don't work well for non smooth function. Potential approachs to deal with that are:

1. Introduce a parameter α to the function f, yielding a parametric function f_α, so that

$$g_\alpha(x) := \int_0^\alpha f_s(x)ds$$

is smooth in x. Then for linear problems, we could use g_α in place of f and differentiate the result with respect to α at the end. For example, $f(x) = \max(x, 0)$, $f_\alpha(x) = \max(x - \alpha, 0)$.
if $x < \alpha$ then,

$$\begin{aligned} g_\alpha(x) &= \int_0^\alpha \max(x - s, 0) ds \\ &= \int_x^\alpha (x - s) ds \\ &= x(\alpha - x) - 1/2 s^2|_x^\alpha \\ &= x(\alpha - x) - 1/2(\alpha^2 - x^2) \end{aligned}$$

else, $x \geq \alpha$, in this case $g_\alpha(x) = 0$. Let check the derivatives with respect to x and α.

$$\frac{dg_\alpha}{dx} = \begin{cases} a - x & x < \alpha \\ 0 & x \geq \alpha \end{cases}$$

Important point to note, $\frac{dg_\alpha}{dx}$ it is continuous.

$$\frac{dg_\alpha}{d\alpha} = \begin{cases} x - \alpha & x < \alpha \\ 0 & x \geq \alpha \end{cases}$$

which is continous as well. Moreover, $\frac{dg_\alpha}{d\alpha} = f_\alpha(x)$ and thus in particular, $f(x) = \frac{dg_0}{d\alpha}$. The conclusion is that for a linear apporiximation problem, i.e. $\mathcal{A}f = \hat{f}$, where $\mathcal{A}$ is linear operator from $C^1(\mathbb{R}^n)$ to some finite dimensional vector sapce (e.g. $\hat{f}$ is a discretized tensor approximation of f). If our f is not smooth, i.e. not in $C^1(\mathbb{R}^n)$, then we can apply the parametr trick. That is, define g_α as above, next solve $\mathcal{A}g_\alpha = \hat{g_\alpha}$. Finally, $\hat{f} = \frac{d\hat{g_\alpha}}{d\alpha}$.

2. Use resolution of singularities. If f is not a smooth function, we may extend the value to multivalued function that is smooth.??????

3. using finite difference step $\Delta x = 1$,40 iterations, rounding to 1e-7, PDE accuracy 1e-6 produced reasonable results

4. $Lu_1 = g_1$, $Lu_2 = g_2$, $u_1(x_{max}, t) = u_2(x_{min}, t)$

5. Using Fourier approximation of payoff, tt-cross aproximating every term, and add up the approximation (rounding every time).

6. Using Fourier approximation, then solving PDE for each Fuorier term and combining the solution at the end. That is, if Furier approximation has 100 terms, create 100 final conditions, solve for each, add up the solutions.

We see an incerease in error in the same spot regardless of smoothing technique for payoff. For example, when we use 1e-12 accuracy payoff (as a sum of 1e-12 accurate payoffs) we see the error in the same plt as in the case where we sum up the payoffs using 1e-3 accuracy truncation. This suggests one of two problems:

(a) Boundary conditions at the upper end of the grid.

(b) Inaccuracy of the time stepping solver, currently using 1e-6

Need to examine cases where T is large 1Y and cases when T is very short. For longer T, there are less issues due to error in approximation of the payoff. Shorter maturities have more issues with pricing of non smooth payoffs.

Need to compare the different smoothing methods in a table with:

- Calculation time,
- Total memomry consumption,
- Accuracy

5 Implicit finite difference method in TT format

5.1 Implementation

5.2 Experiments

6 Value formula as a parametrized expection

For longer maturity payoffs (e.g. at 1Y) the value of epsilon 1e-4 is enoough for payoff, smoothing 0.1 and 1e-5 for the solver. For shorter maturity looks like payoff has to be better approximated or be smoother. For non smooth payoff, we smoothing. The higher the smoothness the higher the apporixmation rate for the same tolerance levels.

References

[1] Andrei Chertkov, Ivan Oseledets. *Solution of the Fokker-Planck equation by cross approximation method in the tensor train format.* arXiv:2102.08143 [math.NA], 2021.

[2] Oseledets IV, Tyrtyshnikov EE. *Tt-cross approximation for multidimensional arrays.* Linear Algebra and its Applications 432 (2010) 70–88.

[3] I. V. Oseledets. *Tensor-Train Decomposition* SIAM J. Sci. Comput., 33(5), 2295–2317.