Course project
"Risk analysis and high-dimensional integrals"

Nikita Puchkin, Igor Silin,Alexander Podkopaev

December 19, 2016

1 Team members

- Igor Silin (leader)

  Igor implemented a solver for numerical solution of Black-Scholes partial differential equation. He described a background, formulated a problem and described, how to reduce it to reaction-diffusion equation as well.

- Nikita Puchkin

  Nikita implemented a solver for numerical solution of reaction-diffusion equation. He prepared description of method and algorithm in this report as well.

- Aleksandr Podkopaev

  Alexandr found different options and tested solver for Black-Scholes partial differential equation. He prepared a section with data and results in this report as well.

2 Background

Many stock prices can modelled using Brownian motion, i.e. the next state depends on the previous state plus some random event. Important statistical options (for example, option calls) can be then estimated as expectations of random process. Option is a contract which gives the owner of the option the right, but not the obligation, to buy or sell an underlying asset at a specified price on a specified date, depending on the form of the option. Two types of options are considered: call options that give the holder the right to buy an asset at a specified price on a specified date and put options that give the holder the right to sell an asset at a specified price on a specified date. The straightforward way for estimation of options is to do Monte-Carlo simulation, but it can not give high accuracy. Another way is to write down the expectation as a path integral and then approximate it by a multi-dimensional integral. More detailed description of the problem which is solved is given in the next section.

3 Problem formulation

Denote by s the stock price and by $c(s,t)$ general option value. Then consider Black-Scholes pricing model. It is known that in that case c must satisfy the following partial differential

equation:

$$\begin{aligned}
&\frac{\partial c(s,t)}{\partial t} + rs\frac{\partial c(s,t)}{\partial s} + \frac{1}{2}\gamma^2 s^2 \frac{\partial^2 c(s,t)}{\partial s^2} = rc(s,t), \quad s \in \mathbb{R}_+, \;\; t \in [0;\; T'], \\
&c(s,T') = g(s), \quad s \in \mathbb{R}_+, \\
&c(0,t) = 0, \quad t \in [0;\; T'].
\end{aligned} \tag{1}$$

Here r is the risk-free interest rate and γ is the volatility of the stock. This is called Black-Scholes partial differential equation. Different terminal conditions $g(s)$ corresponds to different types of derivatives (e.g. European call option). We are interested in methods of solving given PDE for different terminal conditions which are described in the next section.

4 Data

We've chosen several types of options when setting terminal conditions allows to solve Black-Scholes PDE analytically. Let us consider these cases.

4.1 European call option

The corresponding terminal function is:

$$g(s) = \max\{s - E, 0\} \tag{2}$$

For introducing the analytical solution, we firstly denote:

$$d_+^{T'}(t) = \frac{\log(\frac{s}{E}) + (r + \frac{\gamma^2}{2})(T' - t)}{\gamma\sqrt{T' - t}} \tag{3}$$

$$d_-^{T'}(t) = \frac{\log(\frac{s}{E}) + (r - \frac{\gamma^2}{2})(T' - t)}{\gamma\sqrt{T' - t}} \tag{4}$$

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{y^2}{2}} dy \tag{5}$$

Then the analytical solution for the case of european call is:

$$c(s,t) = s\Phi(d_+^{T'}) - Ee^{-r(T'-t)}\Phi(d_-^{T'}) \tag{6}$$

4.2 European put option

The corresponding terminal function is:

$$g(s) = \max\{E - s, 0\} \tag{7}$$

Then the analytical solution for the case of european put option is:

$$c(s,t) = Ee^{-r(T'-t)}\Phi(-d_-^{T'}) - s\Phi(-d_+^{T'}) \tag{8}$$

4.3 Cash-or-nothing call option

The corresponding terminal function is:

$$g(s) = \begin{cases} B, & s \geq E \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Then the analytical solution for the case of cash-or-nothing call option is:

$$c(s,t) = Be^{-r(T'-t)}\Phi(d_-^{T'}) \tag{10}$$

4.4 Cash-or-nothing put option

The corresponding terminal function is:

$$g(s) = \begin{cases} 0, & s \geq E \\ B, & \text{otherwise} \end{cases} \tag{11}$$

Then the analytical solution for the case of case of cash-or-nothing put option is:

$$c(s,t) = Be^{-r(T'-t)}\Phi(-d_-^{T'}) \tag{12}$$

4.5 Asset-or-nothing call option

The corresponding terminal function is:

$$g(s) = \begin{cases} s, & s \geq E \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Then the analytical solution for the case of asset-or-nothing call option is:

$$c(s,t) = s\Phi(d_+^{T'}) \tag{14}$$

4.6 Asset-or-nothing put option

The corresponding terminal function is:

$$g(s) = \begin{cases} 0, & s \geq E \\ s, & \text{otherwise} \end{cases} \tag{15}$$

Then the analytical solution for the case of case of asset-or-nothing put option is:

$$c(s,t) = s\Phi(-d_+^{T'}) \tag{16}$$

5 Related work

We implement the method based on one described in [1].

6 Scope

The solution is based on two important steps: firstly, we reduce the initial problem to one described in related paper, secondly, we implement the method proposed in that paper to get numerical solution.

6.1 Reduction to one-dimensional reaction-diffusion equation

Consider variable substitution of the following type: $x = \ln s$ and $w(x,t) = c(s,t)$. Then after sequence of elementary transformations one obtains:

$$\begin{aligned} &\frac{\partial w(x,t)}{\partial t} + \left(r - \frac{\gamma^2}{2}\right)\frac{\partial w(x,t)}{\partial x} + \frac{1}{2}\gamma^2\frac{\partial^2 w(x,t)}{\partial x^2} = rw(x,t), \quad x \in \mathbb{R}, \;\; t \in [0;\, T'], \\ &w(x,T') = g(e^x), \quad x \in \mathbb{R}, \\ &w(-\infty,t) = 0, \quad t \in [0;\, T']. \end{aligned} \tag{17}$$

To get rid of the drift term $\left(r - \frac{\gamma^2}{2}\right)\frac{\partial w(x,t)}{\partial x}$ we introduce

$$v(x,t) = e^{\frac{1}{\gamma^2}(r-\frac{\gamma^2}{2})x} w(x,t), \tag{18}$$

and obtain

$$\begin{aligned} &\frac{\partial v(x,t)}{\partial t} + \frac{1}{2}\gamma^2\frac{\partial^2 v(x,t)}{\partial x^2} = \left(r + \frac{1}{2\gamma^2}\left(r - \frac{\gamma^2}{2}\right)^2\right) v(x,t), \quad x \in \mathbb{R}, \;\; t \in [0;\, T'], \\ &v(x,T') = e^{\frac{1}{\gamma^2}(r-\frac{\gamma^2}{2})x} g(e^x), \quad x \in \mathbb{R}, \\ &v(-\infty,t) = 0, \quad t \in [0;\, T']. \end{aligned} \tag{19}$$

Introducing $\sigma = \frac{1}{2}\gamma^2$, $V(x,t) = V = r + \frac{1}{2\gamma^2}\left(r - \frac{\gamma^2}{2}\right)^2$ and $f(x) = e^{\frac{1}{\gamma^2}(r-\frac{\gamma^2}{2})x} g(e^x)$ we have

$$\begin{aligned} &-\frac{\partial v(x,t)}{\partial t} = \sigma\frac{\partial^2 v(x,t)}{\partial x^2} - V(x,t)v(x,t), \quad x \in \mathbb{R}, \;\; t \in [0;\, T'], \\ &v(x,T') = f(x), \quad x \in \mathbb{R}, \\ &v(-\infty,t) = 0, \quad t \in [0;\, T']. \end{aligned} \tag{20}$$

Finally, we set $u(x,t) = v(x, T' - t)$ and obtain

$$\begin{aligned} &\frac{\partial u(x,t)}{\partial t} = \sigma\frac{\partial^2 u(x,t)}{\partial x^2} - V(x,t)u(x,t), \quad x \in \mathbb{R}, \;\; t \in [0;\, T'], \\ &v(x,0) = f(x), \quad x \in \mathbb{R}, \\ &v(-\infty,t) = 0, \quad t \in [0;\, T']. \end{aligned} \tag{21}$$

The condition $v(-\infty,t) = 0$ will be satisfied automatically, so our partial differential equation is

$$\begin{aligned} &\frac{\partial u(x,t)}{\partial t} = \sigma\frac{\partial^2 u(x,t)}{\partial x^2} - V(x,t)u(x,t), \quad x \in \mathbb{R}, \;\; t \in [0;\, T'], \\ &u(x,0) = f(x), \quad x \in \mathbb{R}. \end{aligned} \tag{22}$$

This form is exactly one that is considered in Oseledets's paper.

So, when we have $g(s)$, γ, r, T' from our financial model, we just set

$$f(x) = e^{\frac{1}{\gamma^2}(r-\frac{\gamma^2}{2})x} g(e^x), \;\; \sigma = \frac{1}{2}\gamma^2, \quad V(x,t) = V = r + \frac{1}{2\gamma^2}\left(r - \frac{\gamma^2}{2}\right)^2, \tag{23}$$

solve (22) with this parameters, and reconstruct the desired function $c(s,t)$ as

$$c(s,t) = w(e^x,t) = e^{-\frac{1}{\gamma^2}(r-\frac{\gamma^2}{2})\ln s} v(\ln s, t) = e^{-\frac{1}{\gamma^2}(r-\frac{\gamma^2}{2})\ln s} u(\ln s, T' - t). \tag{24}$$

6.2 Description of algorithm and PDE solver

We implement a solver for numerical solution of PDE

$$\frac{\partial}{\partial t}u(x,t) = \sigma\frac{\partial^2}{\partial x^2}u(x,t) - V(x,t)u(x,t), \quad t \in [0,T],\, x \in \mathbb{R}$$
$$u(x,0) = f(x)$$

In this section we use the same idea as described in [1]. The exact solution is given by the Feynman-Kac formula

$$u(x,T) = \int\limits_{C\{x,0;T\}} f(\xi(T)) \exp\left\{-\int\limits_0^T V(\xi(\tau), T-\tau)d\tau\right\}\mathcal{D}_\xi,$$

where the integration is done over the set $C\{x,0;T\}$ of all continuous paths $\xi(T) : [0,T] \to \mathbb{R}$ from the Banach space $\Xi([0,T],\mathbb{R})$ starting at $\xi(0) = x$ and stopping at arbitrary endpoints at time T. $\mathcal{D}_\xi$ is the Wiener measure and $\xi(t)$ is the Wiener process.

For numerical computation one can break the time range $[0,T]$ into n intervals by points

$$\tau_k = k\delta t, \qquad 0 \le k < n, \qquad n : \tau_n = T$$

The average path of a Brownian particle $\xi(\tau_k)$ after k steps is defined as

$$\xi^{(k)} = \xi(\tau_k) = x + \xi_1 + \cdots + \xi_k,$$

where every random step ξ_i, $1 \le i \le k$, is independently taken from a normal distribution $\mathcal{N}(0, 2\sigma\delta t)$. By definition $\xi^{(0)} = x$.

On the introduced uniform grid one approximates

$$\Lambda(T) = \int\limits_0^T V(\xi(\tau), T-\tau)d\tau \simeq \sum_{i=0}^{n} w_i V_i^{(n)}\delta t, \qquad V_i^{(n)} \equiv V(\xi(\tau_i), \tau_{n-i}),$$

where the set of weights $\{w_i\}_{i=0}^n$ is taken according to trapezoid rule or Simpson rule. Then

$$\exp\{-\Lambda(T)\} \simeq \prod_{i=0}^{n} \exp\{-w_i V_i^(n)\delta t$$

The Wiener measure transforms to n-dimensional measure

$$\mathcal{D}_\xi^{(n)} = \left(\frac{\lambda}{\pi}\right)^{\frac{n}{2}} \prod_{k=1}^{n} \exp\{-\lambda\xi_k^2\}d\xi_k, \qquad \lambda = \frac{1}{4\sigma\delta t},$$

and a numerical approximation of the exact solution can be written in the form

$$u^{(n)}(x,T) = \int\limits_{-\infty}^{\infty} \mathcal{D}_{xi} f(\xi^{(n)}) \prod_{i=0}^{n} e^{-w_i v_i^{(n)}\delta t}$$

The multidimensional integral can be represented in terms of n one-dimensional convolutions. Define

$$F_k^{(n)}(x) = \sqrt{\frac{\lambda}{\pi}} \int\limits_{-\infty}^{\infty} \Phi_{k+1}^{(n)}(x+\xi)e^{-\lambda\xi^2}d\xi, \qquad x \in \mathbb{R}, \quad k = n, n-1, \ldots, 1,$$

where

$$\Phi_{k+1}^{(n)}(x) = F_{k+1}^{(n)}(x)\exp\{-w_k V(x,\tau_{n-k})\delta t,$$

and

$$F^{(n)}(x)_{n+1} = f(x)$$

Then the numerical solution is given by formula

$$u^{(n)}(x,T) = F_1^{(n)}(x)e^{-w_0 V(x,T)\delta t}$$

Since $F_k^{(n)}(x)$ is represented as integral over all real values, it is replaced by an integral over a segment

$$F_k^{(n)}(x) \simeq \tilde{F}_k^{(n)}(x) = \sqrt{\frac{\lambda}{\pi}} \int\limits_{-a_x}^{a_x - h_x} \Phi_{k+1}^{(n)}(x+\xi)e^{-\lambda\xi^2}d\xi$$

The function $F_k^{(n)}(x)$ is computed on the uniform mesh

$$x_i^{(k)} = -ka_x + ih_x, \qquad 0 \le i \le kM, \qquad h_x = \frac{a_x}{N_x}, \qquad M = 2N_x$$

and the integration mesh is taken with the same step h_x

$$\xi_j = -a_x + jh_x, \qquad 0 \le j < M$$

Then

$$\tilde{F}_k^{(n)}(x_i^{(k)}) \simeq \sum_{j=0}^{M-1} \mu_j \Phi_{k+1}^{(n)}(x_{i+j}^{(k+1)}p(\lambda,\xi_j), \qquad p(\lambda,\xi) = \sqrt{\frac{\lambda}{\pi}e^{-\lambda\xi^2}}$$

$$\Phi_{k+1}^{(n)}(x_i^{(k+1)} = \tilde{F}_{k+1}^{(n)}(x_i^{(k+1)}\exp\{-w_k V(x_i^{(k+1)},\tau_{n-k})\delta t$$

or in the matrix form

$$\tilde{F}_k^{(n)} = \Phi_{k+1}^{(n)} \circ \tilde{\mu}, \qquad \tilde{\mu}_j = \mu_j p(\lambda,\xi_j)$$

$$\Phi_{k+1}^{(n)} = \tilde{F}_{k+1}^{(n)}\exp\{-w_k V(x^{(k+1)},\tau_{n-k})\delta t,$$

where $a \circ b$ denotes a convolution of vectors $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^k$, i. e. a vector $c \in \mathbb{R}^{m+k-1}$, such that

$$c_i = \sum_{j=0}^{k-1} a_{i+j}b_j, \quad a_i = 0, \forall i : (i<0)\bigvee(i \ge m)$$

The algorithm for computation of $u^{(n)}(x,T)$ is following.

1. Given T, choose a time step δt and the number of steps $n = \frac{T}{\delta t}$.
2. Create 1D array τ of size n, $\tau_i = i\delta t$.
3. Create 1D array w of size $(n+1)$, corresponding to the set of weights in trapezoid or Simpson rule.
4. Choose a_x, size of coordinate grid $M = 2N_x$ and a coordinate step $h_x = \frac{a_x}{N_x}$.
5. Initialize 1D array $\tilde{F}_{n+1}^{(n)}$ of size $(n+1)M$, where $F_{n+1}^{(n)} = f(x)$ and $x_i = -(n+1)a_x + ih_x$, $0 \le i < (n+1)M$.
6. For $k = n, n-1, \ldots, 1$ do

(a) Create 1D array $x^{(k+1)}$ of size $(k+1)M$, $x_i^{(k+1)} = -(k+1)a_x + ih_x$, $0 \leq i < (k+1)M$

(b) Create 1D array of size $(k+1)M$ $e^{-w_k V(x^{(k)}, \tau_{n-k})\delta t}$.

(c) Create 1D array $\Phi_{k+1}^{(n)} = F_{k+1}^{(n)} \odot e^{-w_k V(x^{(k+1)}, \tau_{n-k})\delta t}$.

(d) Create 1D array μ of size $M+1$, corresponding to the set of weights.

(e) Create 1D array ξ of size $M+1$, where $\xi_j = -a_x + jh_x$, $0 \leq j < M+1$.

(f) Create 1D array $\tilde{\mu} = \mu \odot p(\lambda, \xi)$.

(g) Compute convolution $\tilde{F}_k^{(n)} = \Phi_{k+1}^{(n)} \circ \tilde{\mu}$. $\tilde{F}_k^{(n)}$ is 1D array of size kM.

7. Create 1D array $x^{(1)}$ of size M, $x_i^{(1)} = -a_x + ih_x$, $0 \leq i < M$

8. Create 1D array of size M $e^{-w_0 V(x^{(1)}, T)\delta t}$.

9. Compute the numerical solution $u^{(n)} = F_1^{(n)} \odot e^{-w_0 V(x^{(1)}, T)\delta t}$.

The proposed algorithm is implemented in class 'PDE_Solver'.

7 Evalution

To measure the quality of the proposed method, we consider cases when Black-Scholes PDE can be solved analytically. Hence, to measure the quality of the obtained solution, we will simply compare numerical solutions given by the method and the analytical ones.

8 Results

For each of the cases we present 3 plots: the plot for the numerical solution, the plot for analytical solution and, finally, the plot which describes the difference between numerical and analytical solution. The red line on the plots stands for terminal condition $g(s)$ in each case.

8.1 European call option

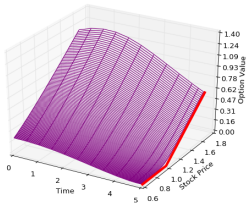

Figure 1: Numerical solution

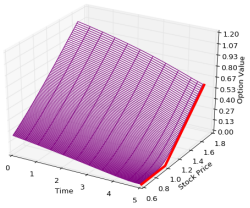

Figure 2: Analytical solution

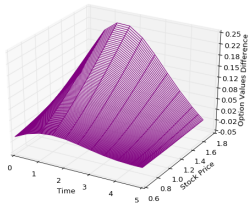

Figure 3: Difference between solutions

8.2 European put option

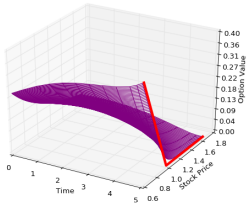

Figure 4: Numerical solution

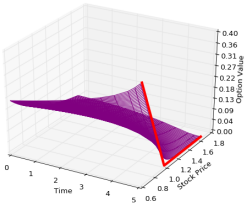

Figure 5: Analytical solution

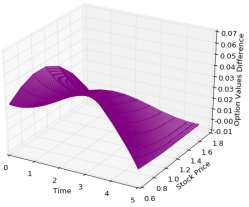

Figure 6: Difference between solutions

8.3 Cash-or-nothing call option

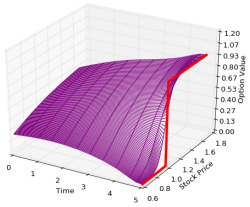

Figure 7: Numerical solution

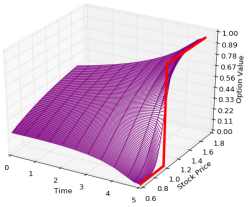

Figure 8: Analytical solution

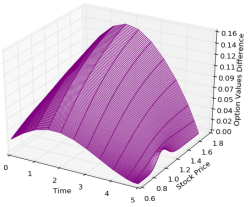

Figure 9: Difference between solutions

8.4 Cash-or-nothing put option

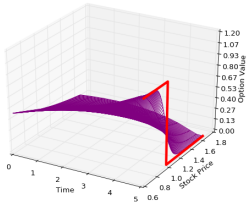

Figure 10: Numerical solution

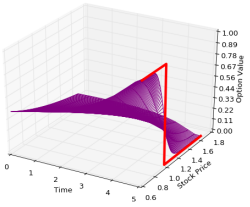

Figure 11: Analytical solution

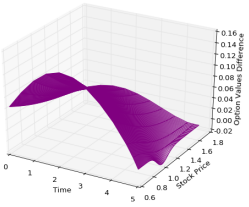

Figure 12: Difference between solutions

8.5 Asset-or-nothing call option

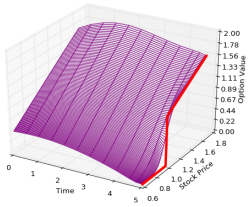

Figure 13: Numerical solution

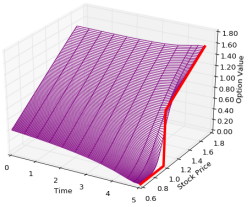

Figure 14: Analytical solution

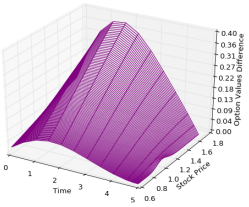

Figure 15: Difference between solutions

8.6 Asset-or-nothing put option

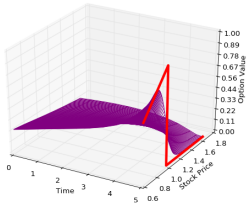

Figure 16: Numerical solution

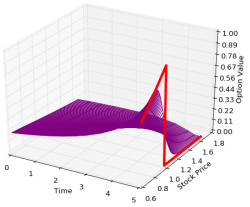

Figure 17: Analytical solution

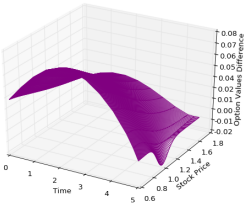

Figure 18: Difference between solutions

9 Conclusion

During this work we faced the following problem: when we used substitution $s = e^x$, a function in initial conditions became to grow exponentially in x, while it grew linearly in s. When we took a large number of time steps n, at the first step we had to find value of initial conditions at points with large coordinates and obtained an overflow. Thus, we were prohibited to take large numbers of steps and, consequently, accuracy of our computations suffered. Moreover, since n was small we did not use low-rank approximation technique, because it was aimed to accelerate computations with insignificant loss of accuracy, while we were mostly interested in accurate solution. We think, that this restriction is caused by specificity of our problem and the method proposed in [1] is inappropriate for this problem. We think, it will be better to build a stable difference scheme for Black-Scholes PDE and not to make the substitution $s = e^x$.

References

[1] A low-rank approach to the computation of path integrals. M.S. Litsarev, I.V. Oseledets, 2015.