INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 00, Number 0, Pages 000–000

JUMPS WITHOUT TEARS:
A NEW SPLITTING TECHNOLOGY FOR BARRIER OPTIONS

ANDREY ITKIN AND PETER CARR

Abstract. The market pricing of OTC FX options displays both stochastic volatility and stochastic skewness in the risk-neutral distribution governing currency returns. To capture this unique phenomenon Carr and Wu developed a model (SSM) with three dynamical state variables. They then used Fourier methods to value simple European-style options. However pricing exotic options requires numerical solution of 3D unsteady PIDE with mixed derivatives which is expensive. In this paper to achieve this goal we propose a new splitting technique. Being combined with another method of the authors, which uses pseudo-parabolic PDE instead of PIDE, this reduces the original 3D unsteady problem to a set of 1D unsteady PDEs, thus allowing a significant computational speedup. We demonstrate this technique for single and double barrier options priced using the SSM.

Key words. barrier options, pricing, stochastic skew, jump-diffusion, finite-difference scheme, numerical method, the Green function, general stable tempered process.

1. Introduction

Every 6 months, the Bank for International Settlements (BIS) publishes an overview of OTC derivatives market activity. The report covers OTC derivatives written on credit, interest rate, currencies, commodities, and equities. Based on that as of the end of June 2009, the total notional amount outstanding in OTC derivatives across the above asset classes stood at about 604 trillion US dollars. Just over one twelfth of this figure is attributable to OTC derivatives on foreign exchange (FX), which includes forwards, swaps, and options. The notional in OTC FX options stands at $11 trillion, which is roughly fifty times the notional in exchange-traded FX contracts.

If one wishes to understand how FX options are priced, it becomes important to access OTC FX options data. Unfortunately, these data is not as readily available as its more liquid exchange-traded counterpart. As a consequence, almost all academic empirical research on FX options has focussed on the exchange-traded market. An exception is a paper by Carr and Wu [6] (henceforth CW), who examine OTC FX options on dollar-yen and on dollar-pound (cable). CW document an empirical phenomenon that is unique to FX options markets. Specifically, at every maturity, the sensitivity of implied volatility to moneyness switches signs over calendar time. This contrasts with the pricing of say equity index options, for which the sensitivity of implied volatility to moneyness is consistently negative over (calendar) time. Since practitioners routinely refer to the sensitivity of implied volatility to moneyness as skew, CW term this time-varying sensitivity ”stochastic skew”. Using time-changed Lévy processes, they develop a class of option pricing models which can accommodate stochastic skew. While their models can, in principle, be used to price any FX exotic, CW only develop their methodology for plain vanilla OTC FX options, which are European-style. In the OTC FX arena, there is a thriving market for barrier options, whose pricing is not covered by CW. The purpose of this paper is to show that barrier options can be efficiently priced in the

2000 *Mathematics Subject Classification.* 60J75, 35M99, 65L12, 65L20, 34B27, 65T50.

stochastic skew model of CW. This model has 3 stochastic state variables, which evolve as a time-homogeneous Markov process. While Monte Carlo can be used to price barrier options in these models, this paper focusses on the use of finite difference methods. More specifically, we show that operator splitting can be used to price a wide variety of barrier options. In particular, we examine the valuation of down-and-out calls, up-and-out calls, and double barrier calls.

Our goal was to propose a splitting technique that could reduce the SSM model original 3D unsteady partial integro-differential equation (PIDE) to a set of simple equations. It turned out that this set contains just 1D unsteady partial differential equations (PDEs) that could be efficiently solved using well-known finite difference schemes. Providing second order of accuracy in time and space was the second important point to meet when building the corresponding numerical methods. Unconditional stability of the method was the third important criterion. So in this paper we present an algorithm, which consists of the following steps:

(1) Split the original 3D unsteady PIDE to 2 independent 2D unsteady PIDEs. This is an exact result with no splitting error.
(2) Split each 2D unsteady PIDE to 1D unsteady PIDE with no drift and diffusion and 2D unsteady PDE with mixed derivatives.
(3) Split the 2D unsteady PDE with mixed derivatives to a set of 1D unsteady PDE using technique of [17].
(4) Using our approach in [18], transform 1D unsteady PIDE with no drift and diffusion to a pseudo-parabolic PDE which then could be efficiently solved by using finite difference schemes for 1D unsteady parabolic PDEs.

We implement this algorithm, providing the second order of approximation in time and all space directions, both at each step of the algorithm and for the entire algorithm as well. Also, our scheme is unconditionally stable in time.

New results presented in the paper are:

- Proved a theorem on how to exactly split 3D PIDE derived under the SSM model into two 2D PIDEs.
- Used a new method to compute the integral term with a linear complexity. The foundation of the method were given in another our paper. And as applied to real multi-dimensional pricing problem it was never described before in the literature.
- Proposed a new approach which reduces solution of the above described 3D PIDE to the solution of a set of 1D unsteady PDEs. The algorithm is of second order of approximation over all space and time coordinates and unconditionally stable.
- Obtained new numerical results on prices of barrier options under the SSM model.

The rest of the paper is organized as follows. The next section lays out assumptions and notation of the SSM model and develops a PIDE that governs the arbitrage-free value of any barrier option under this model. It also discusses boundary conditions for the 3 types of barrier options that we cover. Section 3 validates that the matrix of second derivatives of the model is semi-positive definite. The next three sections show how operator splitting can be applied to the resulting boundary value problems. The penultimate section shows our numerical results, and the final section concludes.

2. Pricing barrier options under SSM

We assume frictionless markets and no arbitrage. Carr and Wu [6] further assume that under an equivalent martingale measure $\mathbb{Q}$, the dynamics of the spot exchange rate and the two activity rates are given by the following system of stochastic differential equations:

$$\begin{aligned} dS_t &= (r_d - r_f)S_{t-}dt \\ &\quad + \sigma\sqrt{V_t^R}S_{t-}dW_t^R + \int_0^\infty S_{t-}(e^x-1)\left[\mu^R(dx,dt) - \lambda\frac{e^{-|x|\nu}}{|x|^{1+\alpha}}\sqrt{V_t^R}dxdt\right] \\ &\quad + \sigma\sqrt{V_t^L}S_{t-}dW_t^L + \int_{-\infty}^0 S_{t-}(e^x-1)\left[\mu^L(dx,dt) - \lambda\frac{e^{-|x|\nu}}{|x|^{1+\alpha}}\sqrt{V_t^L}dxdt\right] \\ dV_t^R &= \kappa(1-V_t^R)dt + \sigma_V\sqrt{V_t^R}dZ_t^R \\ dV_t^L &= \kappa(1-V_t^L)dt + \sigma_V\sqrt{V_t^L}dZ_t^L \\ dW_t^R dW_t^L &= 0, \quad dZ_t^R dZ_t^L = 0, \quad dW_t^R dZ_t^L = 0, \quad dW_t^L dZ_t^R = 0, \\ dW_t^R dZ_t^R &= \rho^R dt, \quad dW_t^L dZ_t^L = \rho^L dt, \end{aligned} \tag{1}$$

for $t \in [0, \Upsilon]$, where $r_d, r_f, \sigma, \lambda, \sigma_V, \kappa$ are nonnegative constants, S_0, V_0^R, V_0^L, ν are positive constants, $\alpha < 2$ is constant, $\rho^R, \rho^L \in [-1, 1]$ are constant, and Υ is some arbitrarily distant time horizon.

Since the spot exchange rate can jump, $S_{t\_}$ denotes the spot price just prior to any jump at t. The processes W^R, W^L, Z^L, Z^R are all $\mathbb{Q}$ standard Brownian motions. The random measures $\mu^R(dx,dt)$ and $\mu^L(dx,dt)$ are used to count the number of up jumps and down jumps of size x in the log spot FX rate at time t. The processes $\int_0^t\int_0^\infty S_{s-}(e^x-1)\lambda\frac{e^{-|x|\nu}}{|x|^{1+\alpha}}\sqrt{V_s^R}dxds$ and $\int_0^t\int_{-\infty}^0 S_{s-}(e^x-1)\lambda\frac{e^{-|x|\nu}}{|x|^{1+\alpha}}\sqrt{V_s^L}dxds$ respectively compensate the driving jump processes
$\int_0^t\int_0^\infty S_{t-}(e^x-1)\mu^R(dx,dt)$ and $\int_0^t\int_{-\infty}^0 S_{t-}(e^x-1)\mu^L(dx,dt)$.

As a result, the last terms in each line of the first equation in (1) are the increments at t of a $\mathbb{Q}$ jump martingale.

When calibrating, we assume that S_0, r_d, and r_f are directly observable. The parameter $\alpha < 2$ is pre-specified. This leaves the two state variables V_t^R, V_t^L and the 7 free parameters $\sigma, \lambda, \sigma_V, \kappa, \nu, \rho^R, \rho^L$ to be identified from the time series of option prices across multiple maturities and moneyness levels.

The vector process $[S_t, V_t^R, V_t^L, t]$ is Markovian in itself on the state space $S > 0, V_R > 0, V_L > 0, t \in [0, T)$. Let:

$$C(S, V_R, V_L, t) \equiv e^{-r(T-t)}E^{\mathbb{Q}}\{(S_T - K)^+ | [S_t, V_t^R, V_t^L, t] = [S, V_R, V_L, t]\} \tag{2}$$

be the smooth function relating the arbitrage-free value of a European call option at time t to the vector of state variables. This function is governed by the following

PIDE:

$$
\begin{aligned}
r_d C(S, \mathbf{\Omega}, t) &= \frac{\partial}{\partial t} C(S, \mathbf{\Omega}, t) + (r_d - r_f) S \frac{\partial}{\partial S} C(S, \mathbf{\Omega}, t) \\
&+ \kappa(1 - V_R) \frac{\partial}{\partial V_R} C(S, \mathbf{\Omega}, t) + \kappa(1 - V_L) \frac{\partial}{\partial V_L} C(S, \mathbf{\Omega}, t) \\
&+ \frac{\sigma^2 S^2 (V_R + V_L)}{2} \frac{\partial^2}{\partial S^2} C(S, \mathbf{\Omega}, t) + \sigma \rho^R \sigma_V S V_R \frac{\partial^2}{\partial S \partial V_R} C(S, \mathbf{\Omega}, t) \\
&+ \sigma \rho^L \sigma_V S V_L \frac{\partial^2}{\partial S \partial V_L} C(S, \mathbf{\Omega}, t) + \frac{\sigma_V^2 V_R}{2} \frac{\partial^2}{\partial V_R^2} C(S, \mathbf{\Omega}, t) + \frac{\sigma_V^2 V_L}{2} \frac{\partial^2}{\partial V_L^2} C(S, \mathbf{\Omega}, t) \\
&+ \sqrt{V_R} \int_0^\infty \left[C(S e^x, \mathbf{\Omega}, t) - C(S, \mathbf{\Omega}, t) - \frac{\partial}{\partial S} C(S, \mathbf{\Omega}, t) S (e^x - 1) \right] \lambda \frac{e^{-|x| \nu}}{|x|^{1+\alpha}} dx \\
&+ \sqrt{V_L} \int_{-\infty}^0 \left[C(S e^x, \mathbf{\Omega}, t) - C(S, \mathbf{\Omega}, t) - \frac{\partial}{\partial S} C(S, \mathbf{\Omega}, t) S (e^x - 1) \right] \lambda \frac{e^{-|x| \nu}}{|x|^{1+\alpha}} dx,
\end{aligned}
\tag{3}
$$

where $\mathbf{\Omega}$ is a vector of V_R, V_L, and the PIDE is defined on the domain $S > 0, V_R > 0, V_L > 0$ and $t \in [0, T]$.

The same PIDE and domain holds for European put values. For American and barrier put and call values, the above PIDE holds on the continuation region (only).

Further it is convenient to introduce a backward time $\tau = T - t$. In the new time the Eq. (3) reads

$$
\begin{aligned}
\frac{\partial}{\partial \tau} C(S, \mathbf{\Omega}, \tau) &= -r_d C(S, \mathbf{\Omega}, \tau) + (r_d - r_f) S \frac{\partial}{\partial S} C(S, \mathbf{\Omega}, \tau) \\
&+ \kappa(1 - V_R) \frac{\partial}{\partial V_R} C(S, \mathbf{\Omega}, \tau) + \kappa(1 - V_L) \frac{\partial}{\partial V_L} C(S, \mathbf{\Omega}, \tau) \\
&+ \frac{\sigma^2 S^2 (V_R + V_L)}{2} \frac{\partial^2}{\partial S^2} C(S, \mathbf{\Omega}, \tau) + \sigma \rho^R \sigma_V S V_R \frac{\partial^2}{\partial S \partial V_R} C(S, \mathbf{\Omega}, \tau) \\
&+ \sigma \rho^L \sigma_V S V_L \frac{\partial^2}{\partial S \partial V_L} C(S, \mathbf{\Omega}, \tau) + \frac{\sigma_V^2 V_R}{2} \frac{\partial^2}{\partial V_R^2} C(S, \mathbf{\Omega}, \tau) + \frac{\sigma_V^2 V_L}{2} \frac{\partial^2}{\partial V_L^2} C(S, \mathbf{\Omega}, \tau) \\
&+ \sqrt{V_R} \int_0^\infty \left[C(S e^x, \mathbf{\Omega}, \tau) - C(S, \mathbf{\Omega}, \tau) - \frac{\partial}{\partial S} C(S, \mathbf{\Omega}, \tau) S (e^x - 1) \right] \lambda \frac{e^{-|x| \nu}}{|x|^{1+\alpha}} dx \\
&+ \sqrt{V_L} \int_{-\infty}^0 \left[C(S e^x, \mathbf{\Omega}, \tau) - C(S, \mathbf{\Omega}, \tau) - \frac{\partial}{\partial S} C(S, \mathbf{\Omega}, \tau) S (e^x - 1) \right] \lambda \frac{e^{-|x| \nu}}{|x|^{1+\alpha}} dx,
\end{aligned}
\tag{4}
$$

Boundary conditions for the above problem are as follows.

Down and Out Calls. Let $L < S$ be a lower barrier. On the domain $S > L, V_R > 0, V_L > 0$ and $t \in [0, T]$, the down-and-out call value function solves the PIDE Eq. (4). On the domain $S < L, V_R > 0, V_L > 0$ and $t \in [0, T], C(S, V_R, V_L, t) = 0$.

The terminal condition for the European call value is:

$$
C(S, V_R, V_L, T) = (S - K)^+, \qquad S \in \mathbb{R}, V_R > 0, V_L > 0. \tag{5}
$$

We impose a delta boundary condition at extremely high return levels:

$$
\lim_{S \uparrow \infty} \frac{\partial}{\partial S} C(S, V_R, V_L, \tau) = e^{r_f \tau}, \qquad V_R > 0, V_L > 0, \tau \in [0, T]. \tag{6}
$$

There exist a wide discussion in the literature on how to impose boundary conditions at extreme values of the activities (see, for instance, [31, 20, 11] and discussion at forums at `http://www.wilmott.com`) [1]. The correct boundary conditions at $V_R, V_L = 0$ are determined by the speed of the diffusion term going to zero as we approach the boundary in a direction normal to the boundary. Suppose we have the PDE

$$C_r = a(r)C_{rr} + b(r)C_r - rC,$$

then, as given in [25], no boundary condition is required at $r = 0$ if $\lim_{r\to 0}(b - a_r) \geq 0$. That seems to be reasonable if the convection at $V = 0$ in the V direction is flowing outside (that is true for the described model as could be seen from the Eq. (4)). To avoid conditions on the coefficients in the Eq. (4) as $V_i \to 0, i = R, L$ one could assume that

$$dV_{i,t} = \kappa(1 - V_{i,t})dt + \sigma_V \sqrt{(V_{i,t})^{1+\epsilon}} dZ_t^i$$

for any $\epsilon > 0$. Obviously, if $\epsilon \ll 1$, this will make no practical difference in the solution, but now, no boundary condition is required at $V = 0$, and this is completely mathematically rigorous. To make it clear no boundary condition means that instead of the boundary condition at $V_i \to 0, i = R, L$ we use the Eq. (4) itself substituting $V_R = 0$ or $V_L = 0$ at the corresponding boundary and taking into account that all the normal diffusion terms are zero. That was also suggested by T. Kluge [20] and motivated by the stability result of the numerical scheme. So he kept the variance boundary at $V = 0$ and did not impose any artificial boundary condition. Instead he discretized the PDE at the grid's boundary points by using one-sided finite differences (from the right) as this is usually done, for instance, in upwind schemes. The accurate numerical implementation of the left variance boundary is required [2]. That is because the numerical solution at any internal point is decisive for errors in the boundary condition approximation since the boundary $V = 0$ influence the value at this internal point.

As, $V_i \to \infty, i = R, L$ it is common to make the following argument. In this case the diffusion term will become very large, and the solution will become very flat. So, $C_{V_i} \approx 0$ as $V_i \to \infty, i = R, L$. This boundary condition has been used for Heston-type models for many years. A relatively recent paper [16], which discusses some methods for the Heston model, outlines the use of this boundary condition.

Up-and-Out Calls. The terminal condition for the European call value is still given by the Eq. (5). Let $H < S$ be a higher barrier. On the domain $H > S, V_R > 0, V_L > 0$ and $t \in [0, T]$, the call value function solves the PIDE Eq. (4). On the domain $H < S, V_R > 0, V_L > 0$ and $t \in [0, T], C(S, V_R, V_L, t) = 0$. Boundary conditions at extreme values of V_R and V_L are same as for the Down-and-Out call.

Double barrier Calls. The terminal condition for the European call value is still given by the Eq. (5). On the domain $H > S > L, V_R > 0, V_L > 0$ and $t \in [0, T]$, the call value function solves the PIDE Eq. (4). On the domain $S < L, V_R > 0, V_L > 0$ or $S < L, V_R > 0, V_L > 0$ and $t \in [0, T], C(S, V_R, V_L, t) = 0$. Boundary conditions at extreme values of V_R and V_L are same as for the Down-and-Out call.

[1] We also thank an anonymous referee who provided us with a further detailed analysis

[2] for instance, if the whole scheme is of the second order in space, then the boundary conditions have to be approximated with the same accuracy

3. A sufficient condition for the matrix of second derivatives to be semi-positive-definite

In principle, any convergent numerical scheme could be applied to the above equation. To guarantee convergence matrix of the second derivatives of the Eq. (4) has to be symmetric and positive definite. This also provides the matrix diagonal dominance.

Proposition 3.1. *Matrix of second derivatives of the Eq. (3) is semi-definite if* $|\rho_L| < 1$ *and* $|\rho_R| < 1$*. It is positive definite if* $V_R \neq 0$ *and* $V_L \neq 0$*, and semi-definite if* $V_R = 0$ *or* $V_L = 0$*.*

Proof. Necessary and sufficient conditions for the matrix of coefficients $(a_{ij})_{3\times 3}$ to be positive definite are [15]:

$$\begin{aligned} &a_{11}a_{22} - a_{12}^2 > 0, \qquad a_{11}a_{33} - a_{13}^2 > 0, \qquad a_{22}a_{33} - a_{23}^2 > 0, \\ &a_{11}a_{22}a_{33} - a_{11}a_{23}^2 - a_{22}a_{13}^2 - a_{33}a_{12}^2 + 2a_{12}a_{13}a_{23} > 0 \end{aligned} \tag{7}$$

For Eq. (4) and the vector of independent variables $\mathbf{x} = (S, V_L, V_R)$ the matrix $(a_{ij})_{3\times 3} \equiv a(\mathbf{x})$ is

$$a(\mathbf{x}) = \frac{1}{2}\begin{vmatrix} \sigma^2 S^2(V_L + V_R) & S V_R \sigma \sigma_V \rho_R & S V_L \sigma \sigma_V \rho_L \\ S V_R \sigma \sigma_V \rho_R & \sigma_V^2 V_R & 0 \\ S V_L \sigma \sigma_V \rho_L & 0 & \sigma_V^2 V_L \end{vmatrix} \tag{8}$$

so direct substitution shows that if $V_R \neq 0$ and $V_L \neq 0$

$$\begin{aligned} &a_{11}a_{22} - a_{12}^2 = \frac{1}{4} S^2\sigma^2\sigma_V^2 V_R\left[V_L + V_R(1-\rho_R^2)\right] > 0, \quad \text{if} |\rho_R| < 1; \\ &a_{11}a_{33} - a_{13}^2 = \frac{1}{4} S^2\sigma^2\sigma_V^2 V_L\left[V_R + V_L(1-\rho_L^2)\right] > 0, \quad \text{if} |\rho_L| < 1; \\ &a_{22}a_{33} - a_{23}^2 = \frac{1}{4} V_L V_R \sigma_V^4 > 0; \\ &a_{11}a_{22}a_{33} - a_{11}a_{23}^2 - a_{22}a_{13}^2 - a_{33}a_{12}^2 + 2a_{12}a_{13}a_{23} = \\ &\frac{1}{8} S^2\sigma^2\sigma_V^4 V_L V_R\left[V_L(1-\rho_L^2) + V_R(1-\rho_R^2)\right] > 0, \quad \text{if} |\rho_L| < 1 \;\&\&\; |\rho_R| < 1; \end{aligned} \tag{9}$$

□

As follows from linear algebra any non-degenerate coordinate transformation also keeps the positive definiteness of the matrix.

4. Our splitting method

To solve the Eq. (4) we utilize splitting. This technique was also known under some different names, like the method of fractional steps [32, 27, 12]; it is also cited in the financial literature as Russian splitting. It is also known as locally one-dimensionally schemes (LOD).

The method of fractional steps reduces the solution of the original N-dimensional unsteady problem to the solution of N 1D equations per time step. For the diffusion equation after a standard discretization is applied, the matrix of each such 1D equation is tri-diagonal. Marchuk [22] and then Strang [28] extended this idea for complex physical processes (for instance, diffusion in the chemically reacting gas, or advection-diffusion problem). In addition to (or instead of) splitting on spatial coordinates, they also proposed to split the equation into physical processes different in nature, for instance, convection and diffusion. Such an approach becomes especially efficient if characteristic times of evolution (relaxation time) of such processes are significantly different.

For additional details on how to construct a splitting algortihm in general we refer the reader to [21].

To use our method of splitting, we first need to rewrite the Eq. (3) using a new variable $x = \ln S/Q$, where Q is a certain constant. That gives

$$
\begin{aligned}
\frac{\partial}{\partial \tau} C(x, \mathbf{\Omega}, \tau) &= -r_d C(x, \mathbf{\Omega}, \tau) \\
&+ \left[r_d - r_f - \frac{\sigma^2}{2}(V_L + V_R) - a_R \sqrt{V_R} - a_L \sqrt{V_L} \right] \frac{\partial}{\partial x} C(x, \mathbf{\Omega}, \tau) \\
&+ \kappa(1 - V_R) \frac{\partial}{\partial V_R} C(x, \mathbf{\Omega}, \tau) + \kappa(1 - V_L) \frac{\partial}{\partial V_L} C(x, \mathbf{\Omega}, \tau) \\
&+ \frac{\sigma^2 (V_R + V_L)}{2} \frac{\partial^2}{\partial x^2} C(x, \mathbf{\Omega}, \tau) + \sigma \rho^R \sigma_V V_R \frac{\partial^2}{\partial x \partial V_R} C(x, \mathbf{\Omega}, \tau) \\
&+ \sigma \rho^L \sigma_V V_L \frac{\partial^2}{\partial x \partial V_L} C(x, \mathbf{\Omega}, \tau) + \frac{\sigma_V^2 V_R}{2} \frac{\partial^2}{\partial V_R^2} C(x, \mathbf{\Omega}, \tau) + \frac{\sigma_V^2 V_L}{2} \frac{\partial^2}{\partial V_L^2} C(x, \mathbf{\Omega}, \tau) \\
&+ \sqrt{V_R} \int_0^\infty \left[C(x+y, \mathbf{\Omega}, \tau) - C(x, \mathbf{\Omega}, \tau) - \frac{\partial}{\partial x} C(x, \mathbf{\Omega}, \tau) y \right] \lambda \frac{e^{-|y|\nu}}{|y|^{1+\alpha}} dy \\
&+ \sqrt{V_L} \int_{-\infty}^0 \left[C(x+y, \mathbf{\Omega}, \tau) - C(x, \mathbf{\Omega}, \tau) - \frac{\partial}{\partial x} C(x, \mathbf{\Omega}, \tau) y \right] \lambda \frac{e^{-|y|\nu}}{|y|^{1+\alpha}} dy,
\end{aligned}
\tag{10}
$$

where

$$
\begin{aligned}
a_R &= \int_0^\infty (e^y - 1 - y) \lambda \frac{e^{-|y|\nu}}{|y|^{1+\alpha}} dy \\
a_L &= \int_{-\infty}^0 (e^y - 1 - y) \lambda \frac{e^{-|y|\nu}}{|y|^{1+\alpha}} dy
\end{aligned}
$$

4.1. First step of splitting. Now let us represent the Eq. (10) in the form

$$
\frac{\partial}{\partial \tau} C(x, \mathbf{\Omega}, \tau) = (L_R + L_L) C(x, \mathbf{\Omega}, \tau), \tag{11}
$$

where

$$
\begin{aligned}
L_R C(x, \mathbf{\Omega}, \tau) &= \Bigg[-\frac{1}{2} r_d + \left(\frac{1}{2}(r_d - r_f) - \frac{1}{2}\sigma^2 V_R - a_R \sqrt{V_R} \right) \frac{\partial}{\partial x} \\
&+ \kappa(1 - V_R) \frac{\partial}{\partial V_R} + V_R \frac{\sigma^2}{2} \frac{\partial^2}{\partial x^2} + \sigma \rho^R \sigma_V V_R \frac{\partial^2}{\partial x \partial V_R} + \frac{\sigma_V^2 V_R}{2} \frac{\partial^2}{\partial V_R^2} \Bigg] C(x, \mathbf{\Omega}, \tau) \\
&+ \sqrt{V_R} \int_0^\infty \left[C(x+y, \mathbf{\Omega}, \tau) - C(x, \mathbf{\Omega}, \tau) - \frac{\partial}{\partial x} C(x, \mathbf{\Omega}, \tau) y \right] \lambda \frac{e^{-|y|\nu}}{|y|^{1+\alpha}} dy
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
L_L C(x,\mathbf{\Omega},\tau) = \Bigg[&-\frac{1}{2} r_d + \left(\frac{1}{2}(r_d - r_f) - \frac{1}{2}\sigma^2 V_L - a_L\sqrt{V_L} \right) \frac{\partial}{\partial x} \\
&+ \kappa(1 - V_L)\frac{\partial}{\partial V_L} + V_L \frac{\sigma^2}{2}\frac{\partial^2}{\partial x^2} + \sigma\rho^L \sigma_V V_L \frac{\partial^2}{\partial x \partial V_L} + \frac{\sigma_V^2 V_L}{2}\frac{\partial^2}{\partial V_L^2} \Bigg] C(x,\mathbf{\Omega},\tau) \\
&+ \sqrt{V_L}\int_{-\infty}^{0} \Big[C(x+y,\mathbf{\Omega},\tau) - C(x,\mathbf{\Omega},\tau) - \frac{\partial}{\partial x}C(x,\mathbf{\Omega},\tau) y \Big] \lambda \frac{e^{-|y|\nu}}{|y|^{1+\alpha}} dy,
\end{aligned}
$$

The following result is easily to obtain.

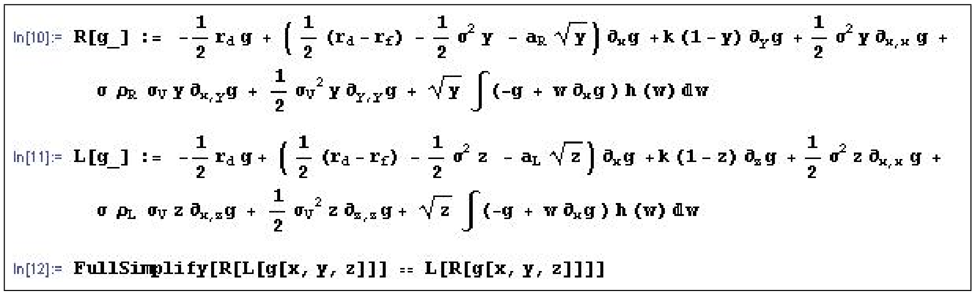
Proposition 4.1. *Operators $L_i, i = R, L$ defined in the Eq. (12) commute.*

Proof. As the commutator of these operators can be derived in a closed form, it can be verified directly that it vanishes. Let us consider first the operators L_R and L_L without the term $C(x+y, V_R, V_L, \tau)$ under the integrals. The coefficients of all terms of L_R are either constants or functions of V_R. They also contain only partial derivatives on x and V_R. Suppose first that $\alpha < 0$, so the integral is well-defined even without term $C(x+y, V_R, V_L, \tau)$. Then without this term the last two terms under the integral can be explicitly integrated. This gives the first term being proportional to $C(x,\mathbf{\Omega},\tau)$, and the other - to $\frac{\partial}{\partial x}C(x,\mathbf{\Omega},\tau)$ both with constant coefficients. Then such L_R can be represented as $\mathcal{L}_{\mathcal{R}}(V_R, \frac{\partial}{\partial x}, \frac{\partial}{\partial V_R})$.

Similarly, the coefficients of all terms of L_L are either constants or functions of V_L. Therefore, it can be represented as $\mathcal{L}_{\mathcal{R}}(V_L, \frac{\partial}{\partial x}, \frac{\partial}{\partial V_L})$.

It is clear then that these reduced operators commute.

Users of the Wolfram Mathematica package could run the following commands to validate this statement (see Fig. 1).



In[10]:= $\mathbf{R[g\_] := -\frac{1}{2} r_d\, g + \left(\frac{1}{2}(r_d - r_f) - \frac{1}{2}\sigma^2 y - a_R\sqrt{y}\right)\partial_x g + k\,(1-y)\,\partial_y g + \frac{1}{2}\sigma^2 y\, \partial_{x,x} g + \sigma\, \rho_R\, \sigma_V\, y\, \partial_{x,y} g + \frac{1}{2}\sigma_V{}^2 y\, \partial_{y,y} g + \sqrt{y}\int(-g + w\, \partial_x g)\, h(w)\, dw}$

In[11]:= $\mathbf{L[g\_] := -\frac{1}{2} r_d\, g + \left(\frac{1}{2}(r_d - r_f) - \frac{1}{2}\sigma^2 z - a_L\sqrt{z}\right)\partial_x g + k\,(1-z)\,\partial_z g + \frac{1}{2}\sigma^2 z\, \partial_{x,x} g + \sigma\, \rho_L\, \sigma_V\, z\, \partial_{x,z} g + \frac{1}{2}\sigma_V{}^2 z\, \partial_{z,z} g + \sqrt{z}\int(-g + w\, \partial_x g)\, h(w)\, dw}$

In[12]:= **FullSimplify[R[L[g[x, y, z]]] == L[R[g[x, y, z]]]]**

FIGURE 1. Mathematica commands to verify that the differential operators $L_{R,L}$ commute as well as the integral operators.

In the general case, when $\alpha < 2$ and the first term under the integral in the Eq. (12) is taken into account, we can formally expand $C(x+y,\mathbf{\Omega},\tau)$ in power series on y to obtain

$$I_L = \sqrt{V_L}\int_{-\infty}^{0}\left[C(x+y,\mathbf{\Omega},\tau) - C(x,\mathbf{\Omega},\tau) - \frac{\partial}{\partial x}C(x,\mathbf{\Omega},\tau)y\right]\lambda\frac{e^{-|y|\nu}}{|y|^{1+\alpha}}dy =$$
$$\sqrt{V_L}\sum_{n=2}^{\infty}a_n\frac{\partial^n}{\partial x^n}C(x,\mathbf{\Omega},\tau),$$
$$I_R = \sqrt{V_R}\int_{0}^{\infty}\left[C(x+y,\mathbf{\Omega},\tau) - C(x,\mathbf{\Omega},\tau) - \frac{\partial}{\partial x}C(x,\mathbf{\Omega},\tau)y\right]\lambda\frac{e^{-|y|\nu}}{|y|^{1+\alpha}}dy =$$
$$\sqrt{V_R}\sum_{n=2}^{\infty}b_n\frac{\partial^n}{\partial x^n}C(x,\mathbf{\Omega},\tau),$$
$$a_n = \int_{-\infty}^{0}\frac{y^{n+1}}{n!}\lambda\frac{e^{-|y|\nu}}{|y|^{1+\alpha}}dy, \qquad b_n = \int_{0}^{\infty}\frac{y^{n+1}}{n!}\lambda\frac{e^{-|y|\nu}}{|y|^{1+\alpha}}dy. \tag{13}$$

Again all coefficients of I_R are just functions of V_R, and all coefficients of I_L are just functions of V_L. Therefore, the whole integral in L_R commutes with both the whole integral in L_L and the differential part of L_L. Similarly, for the whole integral in L_L. Thus, the operators L_R and L_L commute. □

Thus, according to the analysis of the previous section, this splitting scheme does not introduce any splitting error [3]. In other words, the local error of the method is determined by the local errors of each step with no additional error coming up from splitting.

Another important advantage of our splitting scheme is that operators L_R and L_L are two-dimensional integro-differential operators while the original problem Eq. (3) contains a three-dimensional integro-differential operator. Thus, we managed to reduce the dimensionality of the problem.

Note, that alternatively we could extract the part $\frac{1}{2}\left[-r_d + (r_d - r_f)\frac{\partial}{\partial x}\right]C(x,\mathbf{\Omega},\tau)$ from each operator L_R and L_L and combine them as the third operator L_3. In other words, we could split our original operator L into three operators

$$\frac{\partial}{\partial \tau}C(x,\mathbf{\Omega},\tau) = (L_R + L_L + L_3)C(x,\mathbf{\Omega},\tau), \tag{14}$$

where now in contrast to the Eq. (12) L_R and L_L don't have the above part which moved to the operator L_3. It could be again shown that all three operators commute. Therefore this splitting algorithm also does not bring any numerical error. However, the operator L_3 is of the first order. Despite the equation

$$\frac{\partial}{\partial \tau}C(x,\mathbf{\Omega},\tau) = L_3 C(x,\mathbf{\Omega},\tau)$$

can be solved analytically, we expect to face a problem with the boundary conditions. To improve this situation, one can try to add a second order derivative $\frac{\partial^2}{\partial x^2}$ to the operator L_3 and subtract a half of it from the operators L_R and L_L. However, it results in losing the diagonal dominance property of the matrices of second derivatives of the operators L_R, L_L. On the contrary the matrix of second derivatives for the Eq. (11) is still diagonally dominant if $\rho_i \neq 1$ because

$$\frac{1}{2}(a_{11}a_{22} - a_{12}^2) = \sigma^2 S^2 V_i^2 \sigma_V^2 (1 - \rho_i^2). \tag{15}$$

[3] This becomes obvious if one uses the Baker-Campbell-Hausdorff formula (see, for instance, [21]).

Based on the above the whole numerical scheme for one time step θ could be written as

$$
\begin{aligned}
(16)\frac{\partial}{\partial\tau}C^a(x,\mathbf{\Omega},\tau) &= L_R C^a(x,\mathbf{\Omega},\tau), \quad \tau\in[0,\theta], \quad C^a(x,\mathbf{\Omega},0) = C(x,\mathbf{\Omega},0) \\
\frac{\partial}{\partial\tau}C(x,\mathbf{\Omega},\tau) &= L_L C(x,\mathbf{\Omega},\tau), \quad \tau\in[0,\theta], \quad C(x,\mathbf{\Omega},0) = C^a(x,\mathbf{\Omega},\theta)
\end{aligned}
$$

As the L operators commute, in principle, we could solve correspondent equations in an arbitrary order.

Also note that the structure of our boundary conditions allows to naturally split them as well. So for the first equation in Eq. (16) we use boundary conditions at x and V_R boundaries, as they were defined before. And same is true for the second equation at boundaries x and V_R. Actually, these boundaries conditions now coincide with what is used in the literature when pricing barrier options under the Heston model.

4.2. Second step. Strange splitting of jumps. In the previous section the original 3D unsteady PIDE was reduced to a pair of simpler 2D unsteady PIDEs. Each of these PIDEs is similar in structure to that obtained from the well-known Bates jump-diffusion model [2] which is a combination of the Heston model and jumps. Therefore, there exists a wide literature on how to solve these equations, and various numerical methods were proposed (see [14]). A general idea is to either further split the integral and differential terms, or to treat the integral term explicitly to avoid inversion of the dense matrix (because the integral operator is non-local). A good survey of this technique is given in [8], and regarding computation of the integral term - in [5].

Making an analogy between mathematical finance and physics note that Marchuk's idea of splitting on physical processes applied to jump-diffusion models has been already proposed by Cont and Voltchkova in [9]. Their method is based on splitting the operator L into two parts:

$$
\frac{\partial}{\partial\tau}C(S,\mathbf{\Omega},\tau) = [\mathcal{D}+\mathcal{J}]\,C(S,\mathbf{\Omega},\tau), \tag{17}
$$

where $\mathcal{D}$ and $\mathcal{J}$ stand for the differential and integral parts of L respectively. They replaced $\mathcal{D}C(S,V_R,V_L,\tau)$ with a finite difference approximation D, $\mathcal{J}C(S,V_R,V_L,\tau)$ with a certain finite approximation of the integral J and used the following explicit-implicit time stepping scheme:

$$
\frac{C(S,\mathbf{\Omega},\tau)^{n+1} - C(S,\mathbf{\Omega},\tau)^n}{\Delta\tau} = DC(S,\mathbf{\Omega},\tau)^{n+1} + JC(S,\mathbf{\Omega},\tau)^n \tag{18}
$$

Cont and Volchkova treat the integral part explicitly in order to avoid inversion of the non-sparse matrix J. However, this scheme is only conditionally stable; i.e. it brings limitations on the size of the time steps.

In this paper, we intend to use a different approach proposed by us in [18]. The idea is to represent a Lévy measure as the Green's function of some yet unknown differential operator $\mathcal{A}$. If we manage to find an explicit form of such an operator then the original PIDE reduces to a new type of equation - so-called pseudo-parabolic equation. These equations are known in mathematics (see, for instance, [4]) but are new for mathematical finance. Let us underline that this could not be done for an arbitrary Lévy model. However, General tempered stable processes (GTSP) do allow such a transformation and SSM model considered in this paper is just an example of the GTSP.

Further, we rely on two important observations: a) the inverse operator $\mathcal{A}^{-1}$ exists, and b) the obtained pseudo parabolic equation could be formally solved analytically via a matrix exponent. Having that we proposed a numerical method of how to compute this matrix exponent. We show that we can do this computation using a finite difference scheme (FD) similar to that used for solving parabolic PDEs; moreover, the matrix of this FD scheme is banded. We demonstrate this approach in detail for general tempered stable processes (GTSP) with an integer damping exponent α.

Alternatively for some class of Lévy processes, known as GTSP/KoBoL/SSM models, with the real dumping exponent α we show how to transform the corresponding PIDE to a fractional PDE (method 2). Fractional PDEs for the Lévy processes with finite variation were derived in [3] and later in [7] using a characteristic function technique. Numerical solution of these equations was investigated by [7] and [23]. In [18] we derive similar equations in all cases, including processes with infinite variation using a different technique - shift operators. Then to solve them we apply a new method, namely: having results computed for $\alpha \in \mathbb{I}$ we then interpolate them with the second order in α to obtain the solution at any $\alpha \in \mathbb{R}$.

We also show that despite it is a common practice to integrate out all Lévy compensators in the integral term when one considers jumps with finite activity and finite variation, this breaks the stability of the scheme (at least for the fractional PDE). Therefore, in order to construct the unconditionally stable scheme one must keep all the terms under the integrals. To resolve this, in Cartea (2007) the authors were compelled to change their definition of the fractional derivative.

An important local conclusion at this point is that limitations on the use of implicit approximation of the integral term could be overcome by application of our new method. Moreover, this method allows second order approximation in time and space to be achieved straightforward. Therefore, here we approximate the integral term implicitly, so unconditional stability of this approximation could be proved (see [10]).

In order to get the second order approximation, the higher-order operator splitting algorithm can be applied. We use the Strang splitting method of the second order [28] where each time step consists of three sub-steps:

$$
\begin{aligned}
\frac{\partial C^*(S,\mathbf{\Omega},\theta)}{\partial\theta} &= DC^*(S,\mathbf{\Omega},\theta), & C^*(S,\mathbf{\Omega},0) &= C^n(S,\mathbf{\Omega}), & \theta &\in [0,\tau/2] \\
\frac{\partial C(S,\mathbf{\Omega},\theta)^{**}}{\partial\theta} &= JC^{**}(S,\mathbf{\Omega},\theta), & C^{**}(S,\mathbf{\Omega},0) &= C^*(S,\mathbf{\Omega},\tau/2), & \theta &\in [0,\tau] \\
\frac{\partial C^{n+1}(S,\mathbf{\Omega},\theta)}{\partial\theta} &= DC^{**}(S,\mathbf{\Omega},\theta), & C^{n+1}(S,\mathbf{\Omega},0) &= C^{**}(S,\mathbf{\Omega},\tau), & \theta &\in [0,\tau/2]
\end{aligned}
\tag{19}
$$

Usually, for parabolic equations with constant coefficients this composite algorithm is second order accurate in time provided a numerical procedure, which solves a corresponding equation at each splitting step, is at least second order accurate. Thus, now instead of 2D unsteady PIDE we obtain one PIDE with no drift and diffusion (the second equation in the Eq. (19)) and two 2D unsteady PDE (the first and third equations in the Eq. (19)).

4.3. Splitting the 2D PDE. Recall that one of the equations obtained in the previous section by applying the Strang splitting scheme is the first (or the third) equation in the Eq. (19)) which in the explicit form reads

$$\begin{aligned}
(20)\qquad & \frac{\partial}{\partial\theta}C^{**}(x,V_R,V_L,\theta) = (F_0+F_1+F_2)C^{**}(x,V_R,V_L,\theta)\\
& C^{**}(S,V_R,V_L,0) = C^{*}(S,V_R,V_L,\tau/2), \quad \theta\in[0,\tau]\\
& F_0 = \sigma\rho^i\sigma_V V_i \frac{\partial^2}{\partial x\partial V_i}\\
& F_1 = -\frac{1}{2}r_d + \left[\frac{1}{2}(r_d-r_f) - \frac{1}{2}\sigma^2 V_i - a_{1i}\sqrt{V_i}\right]\frac{\partial}{\partial x} + V_i\frac{\sigma^2}{2}\frac{\partial^2}{\partial x^2}\\
& F_2 = \kappa(1-V_i)\frac{\partial}{\partial V_i} + \frac{\sigma_V^2 V_i}{2}\frac{\partial^2}{\partial V_i^2},\\
& a_{1R} = \int_0^\infty (e^y-1-y)\lambda\frac{e^{-|y|\nu}}{|y|^{1+\alpha}}dy, \quad a_{1L} = \int_{-\infty}^0 (e^y-1-y)\lambda\frac{e^{-|y|\nu}}{|y|^{1+\alpha}}dy
\end{aligned}$$

Here index $i\in(R,L)$ is used to mark R and L states and related coefficients.

One can observe that this equation is very similar to the PDE which appears in stochastic volatility models; for instance, in the familiar Heston model. Therefore, there is a wide literature on how to solve this PDE using various numerical methods, and, particularly, finite difference. But in general, all these methods have to solve a problem of the mixed derivative term F_0.

We, however, propose a different approach. We follow Hout and Welfert [17], who considered the unconditional stability of second-order ADI schemes in the numerical solution of finite difference discretizations of multi-dimensional diffusion problems containing mixed spatial-derivative terms. They investigated the ADI scheme proposed by Craig and Sneyd (see references in the paper), the ADI scheme that is a modified version thereof, and the ADI scheme introduced by Hundsdorfer and Verwer. Both necessary and sufficient conditions are derived on the parameters of each of these schemes for unconditional stability in the presence of mixed derivative terms. Their main result is that, under a mild condition on its parameter θ, the second-order Hundsdorfer and Verwer scheme is unconditionally stable when applied to semi-discretized diffusion problems with mixed derivative terms in arbitrary spatial dimensions $k>2$.

Following [17], consider the initial-boundary value problems for two-dimensional diffusion equations, which after the spatial discretization lead to initial value problems for huge systems of ordinary differential equations (ODEs)

$$U'(t) = F(t,U(t)) \quad t\geq 0, \quad U(0)=U_0, \tag{21}$$

with given vector-valued function F and initial vector U_0. Hout and Welfert consider splitting schemes for the numerical solution of the Eq. (21). They assume that F is decomposed into the sum

$$F(t,U) = F_0(t,U) + F_1(t,U) + ... + F_k(t,U), \tag{22}$$

of $k+1$ terms $F_j,\ j=0...k$ that are easier to handle than F itself. The term F_0 contains all contributions to F stemming from the mixed derivatives in the diffusion equation, and this term is always treated explicitly in the numerical time integration. Next, for each $j\geq 1, F_j$ represents the contribution to F stemming from the second-order derivative in the j -th spatial direction, and this term is always treated implicitly.

Further, Hout and Welfert consider two splitting schemes, and one of them is a modified Craig and Sneyd scheme. This scheme defines an approximation

$U_n \approx U(t_n),\ n = 1, 2, 3...$ by

$$
\begin{aligned}
(23) \qquad & Y_0 = U_{n-1} + \Delta t F(t_{n-1}, U_{n-1}), \\
& Y_j = Y_{j-1} + \theta \Delta t \left[F_j(t_n, Y_j) - F_j(t_{n-1}, U_{n-1}) \right],\ j = 1, 2, ..., k \\
& \hat{Y}_0 = Y_0 + \theta \Delta t \left[F_0(t_n, Y_k) - F_0(t_{n-1}, U_{n-1}) \right], \\
& \tilde{Y}_0 = \hat{Y}_0 + \left(\frac{1}{2} - \theta \right) \Delta t \left[F(t_n, Y_k) - F(t_{n-1}, U_{n-1}) \right], \\
& \tilde{Y}_j = \tilde{Y}_{j-1} + \theta \Delta t \left[F_j(t_n, \tilde{Y}_j) - F_j(t_n, Y_k) \right],\ j = 1, 2, ..., k \\
& U_n = \tilde{Y}_k
\end{aligned}
$$

This scheme is of order two in time step for any value of θ, so this parameter can thus be chosen to meet additional requirements. Further Hout and Welfert investigate the stability of this scheme using von Neumann analysis. Accordingly, stability is always considered in the L2-norm and, in order to make the analysis feasible, all coefficients in the Eq. (23) are assumed to be constant and the boundary condition to be periodic. Under these assumptions, the matrices $A_0, A_1, ..., A_k$ obtained by finite difference discretization of operators F_k [4] are constant and form Kronecker products of circulant matrices. Hence, they are normal and commute with each other. This implies that stability can be analyzed by considering the linear scalar ODE

$$
(24) \qquad U'(t) = (\lambda_0 + \lambda_1 + ... + \lambda_k) U(t)
$$

where λ_j denotes an eigenvalue of the matrix $A_j,\ 0 \le j \le k$. Then analyzing this equation Hout and Welfert prove some important theorems to show unconditional stability of their splitting scheme if $\theta \ge 1/3$.

An important property of this scheme is that the mixed derivative term in the first equation of the Eq. (23) is treated explicitly while all further implicit steps contain only derivative in time and derivative in a one space coordinate. In other words, the whole 2D unsteady problem is reduced to a set of four 1D unsteady equations and 2 explicit equations.

For the semi-discretization of the Eq. (23) the authors consider finite differences. All spatial derivatives are approximated using second-order central differences on a rectangular grid with constant mesh width $\Delta x_i > 0$ in the x_i direction ($1 \le i \le k$). Details of this scheme implementation are discussed in a recent paper [29]. Their experiments show that for the Heston model choice of $\theta = 1/3$ is good. They also demonstrate that this scheme has a stiff [5] order of convergence in time equal to two.

It can be easily observed that the first and third equations in the Eq. (19) are of the Heston type, therefore we will apply the above described scheme to the solution of these equations.

4.4. From PIDE to pseudo-parabolic PDE and further.

This section shortly describes our method of solving the second equation in the Eq. (19). Readers interesting in getting more details are referred to [18]. Here we will also modify a bit the SSM model by using in the jump terms two different activities ν_R and ν_L instead of just one ν, α_R and α_L instead of one α, and λ_R and λ_L instead of one

[4] An explicit discretization of F_k in our case is discussed below.

[5] I.e. the order of convergency doesn't fluctuate significantly with time step change, and is always very close to 2.

λ. This increases the number of free parameters of the model and allows its better calibration to market data.

Actually, for the whole algorithm we have to solve two similar PIDEs. One appears at the second step of splitting of the operator L_R and has the explicit form (25)

$$\frac{\partial C(x,\mathbf{\Omega},\theta)}{\partial\theta} = \sqrt{V_R}\int_0^\infty \Big[C(x+y,\mathbf{\Omega},\theta) - C(x,\mathbf{\Omega},\theta) - \frac{\partial}{\partial x}C(x,\mathbf{\Omega},\theta)y\Big]\lambda_R\frac{e^{-|y|\nu_R}}{|y|^{1+\alpha_R}}dy$$

The other one appears at the second step of splitting of the operator L_L and has the explicit form

$$\frac{\partial C(x,\mathbf{\Omega},\theta)}{\partial\theta} = \sqrt{V_L}\int_{-\infty}^0 \Big[C(x+y,\mathbf{\Omega},\theta) - C(x,\mathbf{\Omega},\theta) - \frac{\partial}{\partial x}C(x,\mathbf{\Omega},\theta)y\Big]\lambda_L\frac{e^{-|y|\nu_L}}{|y|^{1+\alpha_L}}dy.$$

In what follows where it is not confusing we will use the same notation for the price $C(x,\mathbf{\Omega},\theta)$.

For the moment, assume that $\alpha < 0$ and therefore we can integrate out two last compensating terms under the integrals. These terms could be added to the diffusion part considered in the previous section. Later we will relax this assumption and consider the whole integral with $\alpha < 2$.

Assuming $z = x + y$ we rewrite the first equation in the Eq. (25) in the form

$$\frac{\partial}{\partial\theta}C(x,\theta) = \sqrt{V_R}\int_x^\infty C(z,\theta)\lambda_R\frac{e^{-\nu_R|z-x|}}{|z-x|^{1+\alpha_R}}\mathbf{1}_{z-x>0}dz \tag{26}$$

To achieve our goal we have to solve the following problem. We need to find a differential operator $\mathcal{A}_y^+$ which Green's function is the kernel of the integral in the Eq. (26), i.e.

$$\mathcal{A}_y^+\left[\lambda_R\frac{e^{-\nu_R|y|}}{|y|^{1+\alpha_R}}\mathbf{1}_{y>0}\right] = \delta(y). \tag{27}$$

Proposition 4.2. *Assume that in the Eq. (27) $\alpha_R \in \mathbb{I}$ (α_R is integer), and $\alpha_R < 0$. Then the solution of the Eq. (27) with respect to $\mathcal{A}_y^+$ is*

$$\mathcal{A}_y^+ = \frac{1}{\lambda_R p!}\left(\nu_R + \frac{\partial}{\partial y}\right)^{p+1} \equiv \frac{1}{\lambda_R p!}\left[\sum_{i=0}^{p+1} C_i^{p+1}\nu_R^{p+1-i}\frac{\partial^i}{\partial y^i}\right], \quad p \equiv -(1+\alpha_R) \geq 0,$$

where C_i^{p+1} are the binomial coefficients.

The proof is given in [18].

For the second equation in the Eq. (25) it is possible to elaborate on an analogous approach. Again assuming $z = x + y$ we rewrite it in the form

$$\frac{\partial}{\partial\theta}C(x,\theta) = \sqrt{V_L}\int_{-\infty}^x C(z,\theta)\lambda_L\frac{e^{-\nu_L|z-x|}}{|z-x|^{1+\alpha_L}}\mathbf{1}_{z-x<0}dz \tag{28}$$

Now we need to find a differential operator $\mathcal{A}_y^-$ which Green's function is the kernel of the integral in the Eq. (28), i.e.

$$\mathcal{A}_y^-\left[\lambda_L\frac{e^{-\nu_L|y|}}{|y|^{1+\alpha_L}}\mathbf{1}_{y<0}\right] = \delta(y) \tag{29}$$

In [18], we prove the following proposition.

Proposition 4.3. *Assume that in the Eq. (29) $\alpha_L \in \mathbb{I}$, and $\alpha_L < 0$. Then the solution of the Eq. (29) with respect to $\mathcal{A}^-_y$ is*

$$\mathcal{A}^-_y = \frac{1}{\lambda_L p!}\left(\nu_L - \frac{\partial}{\partial y}\right)^{p+1} \equiv \frac{1}{\lambda_L p!}\left[\sum_{i=0}^{p+1}(-1)^i C_i^{p+1}\nu_L^{p+1-i}\frac{\partial^i}{\partial y^i}\right], \ p \equiv -(1+\alpha_L),$$

To proceed we need two technical propositions.

Proposition 4.4. *Let us denote the kernels as*

$$g^+(z-x) \equiv \lambda_R \frac{e^{-\nu_R|z-x|}}{|z-x|^{1+\alpha_R}}\mathbf{1}_{z-x>0}. \tag{30}$$

Then

$$\mathcal{A}^-_x g^+(z-x) = \delta(z-x). \tag{31}$$

Proposition 4.5. *Let us denote the kernels as*

$$g^-(z-x) \equiv \lambda_L \frac{e^{-\nu_L|z-x|}}{|z-x|^{1+\alpha_L}}\mathbf{1}_{z-x<0}. \tag{32}$$

Then

$$\mathcal{A}^+_x g^-(z-x) = \delta(z-x). \tag{33}$$

Again, for the details see [18]. We now apply the operator $\mathcal{A}^-_x$ to both parts of the Eq. (26) to obtain

$$\begin{aligned} \mathcal{A}^-_x \frac{\partial}{\partial \theta}C(x,\theta) &= \sqrt{V_R}\mathcal{A}^-_x\int_x^\infty C(z,\theta)g^+(z-x)dz \\ &= \sqrt{V_R}\left\{\int_x^\infty C(z,\theta)\mathcal{A}^-_x g^+(z-x)dz + \mathcal{R}\right\} \\ &= \sqrt{V_R}\left\{\int_x^\infty C(z,\theta)\delta(z-x)dz + \mathcal{R}\right\} = \frac{1}{2}\sqrt{V_R}C(x,\theta) - \sqrt{V_R}\mathcal{R} \end{aligned} \tag{34}$$

Here

$$\mathcal{R} = \sum_{i=0}^{p} a_i \left(\frac{\partial^{p-i}}{\partial x^{p-i}}V(x)\right)\left(\frac{\partial^i}{\partial x^i}g(z-x)\right)\Big|_{z-x=0}, \tag{35}$$

and a_i are some constant coefficients. As from the definition in the Eq. (30) $g(z-x) \propto (z-x)^p$, the only term in the Eq. (35) which does not vanish is that at $i=p$. Thus

$$\mathcal{R} = V(x)\left(\frac{\partial^p}{\partial x^p}g(z-x)\right)\Big|_{z-x=0} = V(x)p!\mathbf{1}_{(0)} = 0; \tag{36}$$

With allowance for this expression from the Eq. (34) we obtain the following pseudo-parabolic equation for $C(x,t)$

$$\mathcal{A}^-_x \frac{\partial}{\partial \theta}C(x,\theta) = \frac{1}{2}\sqrt{V_R}C(x,\theta) \tag{37}$$

Applying the operator $\mathcal{A}^+_x$ to both parts of the second equation in the Eq. (28) and doing in the same way as in the previous paragraph we obtain the following pseudo parabolic equation for $C(x,\theta)$

$$\mathcal{A}^+_x \frac{\partial}{\partial \theta}C(x,t) = -\frac{1}{2}\sqrt{V_L}C(x,\theta) \tag{38}$$

We will solve these equations with the same initial and boundary conditions as were described before.

4.4.1. Solution of the pseudo parabolic equation. Assuming that the inverse operator $\mathcal{A}^{-1}$ exists (see discussion later) we can represent, for instance, the Eq. (37) in the form

$$\frac{\partial}{\partial \theta} C(x,\theta) = \mathcal{B} C(x,\theta), \quad \mathcal{B} \equiv \frac{1}{2}\sqrt{V_R}(\mathcal{A}_x^-)^{-1}, \tag{39}$$

This equation can be formally solved analytically to give

$$C(x,\theta) = e^{\mathcal{B}\theta} C(x,0), \tag{40}$$

Below we consider numerical methods which allow one to compute this operator exponent with a prescribed accuracy. First, we consider a straightforward approach when $\alpha \in \mathbb{I}$. Then for $\alpha \in \mathbb{R}$ we use interpolation; i.e. having the results computed for $\alpha \in \mathbb{I}$ we interpolate them with the second order spline in α space to obtain the solution at any $\alpha \in \mathbb{R}$.

To make sure this is consistent we prove the following proposition.

Proposition 4.6. *Both integrals in the Eq. (12) are continuous in α at $\alpha < 2$.*

Proof. To prove this we use a series representation of the integrals derived in the Eq. (13). Then at $\alpha < 2$ coefficients a_n and b_n, $n \ge 2$ are regular functions of α. So the integrand kernels in the definition of a_n, b_n are continuous functions of α as well as a_n and b_n. □

Next suppose that the whole time space is uniformly divided into N steps, so the time step $\theta = T/N$ is known. Assuming that the solution at time step $k, 0 \le k < N$ is known and we go backward in time, we could rewrite the Eq. (40) in the form

$$C^{k+1}(x) = e^{\mathcal{B}\theta} C^k(x), \tag{41}$$

where $C^k(x) \equiv C(x, k\theta)$. To get representation of the rhs of the Eq. (41) with given order of approximation in θ, we can substitute the whole exponential operator with its Padé approximation of the corresponding order m.

First, consider the case $m = 1$. A symmetric Padé approximation of the order $(1,1)$ for the exponential operator is

$$e^{\mathcal{B}\theta} = \frac{1 + \mathcal{B}\theta/2}{1 - \mathcal{B}\theta/2} \tag{42}$$

Substituting this into the Eq. (41) and affecting both parts of the equation by the operator $1 - \mathcal{B}\theta/2$ gives

$$\left(1 - \frac{1}{2}\mathcal{B}\theta\right) C^{k+1}(x) = \left(1 + \frac{1}{2}\mathcal{B}\theta\right) C^k(x). \tag{43}$$

This is a discrete equation which approximates the original solution given in the Eq. (41) with the second order in θ. One can easily recognize in this scheme the famous Crank-Nicolson scheme.

We do not want to invert the operator $\mathcal{A}_x^-$ in order to compute the operator $\mathcal{B}$ because $\mathcal{B}$ is an integral operator. Therefore, we will apply the operator $\mathcal{A}_x^-$ to the both sides of the Eq. (43). The resulting equation is a pure differential equation and reads

$$\left(\mathcal{A}_x^- - \frac{\sqrt{V_R}}{4}\theta\right) C^{k+1}(x) = \left(\mathcal{A}_x^- + \frac{\sqrt{V_R}}{4}\theta\right) C^k(x). \tag{44}$$

Let us work with the operator $\mathcal{A}_x^-$ (for the operator $\mathcal{A}_x^+$ all corresponding results can be obtained in a similar way). The operator $\mathcal{A}_x^-$ contains derivatives in x up to the order $p+1$. If one uses a finite difference representation of these derivatives the resulting matrix in the rhs of the Eq. (44) is a band matrix. The number of diagonals in the matrix depends on the value of $p = -(1+\alpha_R) > 0$. For central difference approximation of derivatives of order d in x with the order of approximation q the matrix will have at least $l = odd(d+q, d+q-1)$ [13] [6]. Therefore, if we consider a second order approximation in x, i.e. $q = 2$ in our case the number of diagonals is $l = p + 3 = 2 - \alpha_R$.

As the rhs matrix $\mathcal{D} \equiv \mathcal{A}_x^- - \sqrt{V_R}\theta/4$ is a banded matrix, the solution of the corresponding system of linear equations in the Eq. (44) could be efficiently obtained using a modern technique (for instance, using a ScaLAPACK package[7]). The computational cost for the LU factorization of an N-by-N matrix with lower bandwidth P and upper bandwidth Q is $2NPQ$ (this is an upper bound) and storage-wise - $N(P+Q)$. So in our case of the symmetric matrix the cost is $(1-\alpha_R)^2 N/2$ performance-wise and $N(1-\alpha_R)$ storage-wise. This means that the complexity of our algorithm is still $O(N)$ while the constant $(1-\alpha_R)^2/2$ could be large.

A typical example could be if we solve our PDE using an x-grid with 300 nodes, so $N = 300$. Suppose $\alpha_R = -10$. Then the complexity of the algorithm is $60N = 18000$. Compare this with the FFT algorithm complexity which is $(34/9)2N\log_2(2N) \approx 20900$ [8], one can see that our algorithm is of the same speed as the FFT.

The case $m = 2$ could be achieved either using symmetric (2,2) or diagonal (1,2) Padé approximations of the operator exponent. The (1,2) Padé approximation reads

$$e^{\mathcal{B}\theta} = \frac{1+\mathcal{B}\theta/3}{1-2\mathcal{B}\theta/3+\mathcal{B}^2\theta^2/6}, \tag{45}$$

and the corresponding finite difference scheme for the solution of the Eq. (41) is

$$\left[(\mathcal{A}_x^-)^2 - \frac{1}{3}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{24}V_R\theta^2\right]C^{k+1}(x) = \mathcal{A}_x^-\left[\mathcal{A}_x^- + \frac{1}{6}\sqrt{V_R}\theta\right]C^k(x). \tag{46}$$

which is of the third order in θ. The (2,2) Padé approximation is

$$e^{\mathcal{B}\theta} = \frac{1+\mathcal{B}\theta/2+\mathcal{B}^2\theta^2/12}{1-\mathcal{B}\theta/2+\mathcal{B}^2\theta^2/12}, \tag{47}$$

and the corresponding finite difference scheme for the solution of the Eq. (41) is

$$\left[(\mathcal{A}_x^-)^2 - \frac{1}{4}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{48}V_R\theta^2\right]C^{k+1}(x) = \left[(\mathcal{A}_x^-)^2 + \frac{1}{4}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{48}V_R\theta^2\right]C^k(x), \tag{48}$$

which is of the fourth order in θ.

Matrix of the operator $(\mathcal{A}_x^-)^2$ has $2l-1$ diagonals, where l is the number of diagonals of the matrix $\mathcal{A}_x^-$. Thus, the finite difference equations Eq. (46) and Eq. (48) still have banded matrices and could be efficiently solved using the appropriate technique.

[6] For instance, approximation of the second derivative ($d = 2$) with the second order ($q = 2$) gives odd(4,3) = 3.

[7] see `http://www.netlib.org/scalapack/scalapack_home.html`

[8] We use $2N$ instead of N because in order to avoid undesirable wrap-round errors a common technique is to embed a discretization Toeplitz matrix into a circulant matrix. This requires to double the initial vector of unknowns

4.4.2. Stability analysis. For the derived finite difference schemes this could be provided using the standard von-Neumann method. Suppose that operator $\mathcal{A}_x^-$ has eigenvalues ζ which belong to continuous spectrum. Any finite difference approximation of the operator $\mathcal{A}_x^- \to FD(\mathcal{A}_x^-)$ - transforms this continuous spectrum into some discrete spectrum, so we denote the eigenvalues of the discrete operator $FD(\mathcal{A}_x^-)$ as $\zeta_i, i = 1, N$, where N is the total size of the finite difference grid.

Now let us consider, for example, the Crank-Nicolson scheme given in the Eq. (44). It is stable if in some norm $\| \cdot \|$

$$\left\| \left(\mathcal{A}_x^- - \frac{\sqrt{V_R}}{4}\theta \right)^{-1} \left(\mathcal{A}_x^- + \frac{\sqrt{V_R}}{4}\theta \right) \right\| < 1. \tag{49}$$

It is easy to see that this inequality holds when all eigenvalues of the operator $\mathcal{A}_x^-$ are negative. However, based on the definition of this operator given in the Proposition 4.3, it is clear that the central finite difference approximation of the first derivative does not give rise to a full negative spectrum of eigenvalues of the operator $FD(\mathcal{A}_x^-)$. So below we define a different approximation.

Case $\alpha_R < 0$**.** In this case we will use a one-sided forward approximation of the first derivative which is a part of the operator $\left(\nu_R - \dfrac{\partial}{\partial x}\right)^{\alpha_R}$. Define $h = (x_{max} - x_{min})/N$ to be the grid step in the x-direction, N is the total number of steps, x_{min} and x_{max} are the left and right boundaries of the grid. Also define $c_i^k = C^k(x_i)$. To make our method to be of the second order in x we use the numerical approximation

$$\frac{\partial C^k(x)}{\partial x} = \frac{-C_{i+2}^k + 4C_{i+1}^k - 3C_i^k}{2h} + O(h^2) \tag{50}$$

The matrix of this discrete difference operator has the form

$$M_f = \frac{1}{2h} \begin{pmatrix} -3 & 4 & -1 & 0 & ...0 \\ 0 & -3 & 4 & -1 & ...0 \\ 0 & 0 & -3 & 4 & ...0 \\ .. & .. & .. & .. & .. \\ 0 & 0... & 0 & 0 & -3 \end{pmatrix} \tag{51}$$

All eigenvalues of M_f are equal to $-3/(2h)$.

To get a power of the matrix M we use its spectral decomposition, i.e. we represent it in the form $M = EDE'$, where D is a diagonal matrix of eigenvalues $d_i, i = 1, N$ of the matrix M, and E is a matrix of eigenvectors of the matrix M. Then $M^{p+1} = ED^{p+1}E'$, where the matrix D^{p+1} is a diagonal matrix with elements $d_i^{p+1}, i = 1, N$. Therefore, the eigenvalues of the matrix $\left(\nu_R - \frac{\partial}{\partial x}\right)^{\alpha_R}$ are $[\nu_R + 3/(2h)]^{\alpha_R}$. And, consequently, the eigenvalues of the operator $\mathcal{B}$ in this representation are

$$\zeta_{\mathbb{B}} = \sqrt{V_R}\lambda_R\Gamma(-\alpha_R)\left\{[\nu_R + 3/(2h)]^{\alpha_R} - \nu_R^{\alpha_R}\right\}. \tag{52}$$

As $\alpha_R < 0$ and $\nu_R > 0$ it follows that $\zeta_{\mathbb{B}} < 0$. Rewriting the Eq. (43) in the form

$$C^{k+1}(x) = \left(1 - \frac{1}{2}\mathcal{B}\theta\right)^{-1} \left(1 + \frac{1}{2}\mathcal{B}\theta\right) C^k(x), \tag{53}$$

and taking into account that $\zeta_{\mathbb{B}} < 0$ we arrive at the following result

$$\left\| \left(1 - \frac{1}{2}\mathcal{B}\theta\right)^{-1} \left(1 + \frac{1}{2}\mathcal{B}\theta\right) \right\| < 1. \tag{54}$$

Thus, our numerical method is unconditionally stable.

Case $\alpha_L < 0$. In this case we will use a one-sided backward approximation of the first derivative in the operator $\left(\nu_L + \dfrac{\partial}{\partial x}\right)^{\alpha_L}$ which reads

$$\frac{\partial C^k(x)}{\partial x} = \frac{3C_i^k - 4C_{i-1}^k + C_{i-2}^k}{2h} + O(h^2) \tag{55}$$

The matrix of this discrete difference operator has the form

$$M_b = \frac{1}{2h}\begin{pmatrix} 3 & 0 & 0 & 0 & ...0 \\ -4 & 3 & 0 & 0 & ...0 \\ 1 & -4 & 3 & 0 & ...0 \\ .. & .. & .. & .. & .. \\ 0 & 0... & 1 & -4 & 3 \end{pmatrix} \tag{56}$$

All eigenvalues of M_b are equal to $3/(2h)$. Proceeding as above, we can show that the eigenvalues of the operator $\mathcal{B}$ read

$$\zeta_{\mathbb{B}} = \sqrt{V_L}\lambda_L\Gamma(-\alpha_L)\left\{[\nu_L + 3/(2h)]^{\alpha_L} - \nu_L^{\alpha_L}\right\}. \tag{57}$$

As $\alpha_L < 0$ and $\nu_L > 0$ it follows that $\zeta_{\mathbb{B}} < 0$, and the numerical method in this case is unconditionally stable.

In [18], we also consider two special cases $\alpha = 0, 1$ and show how to extend the proposed method to these values of α. For $1 < \alpha < 2$, instead of interpolation we suggest to use extrapolation. That is because our method, as it was described, is not applicable at $\alpha = 2$. Therefore, we can not use interpolation in this case, and this is a shortcoming of the method. In [18], we also provide a comparison of our method with the FFT method used to compute the jump integral in many papers. Andersen and Andreasen [1] were apparently the first who suggest to use the method for this purpose, while a detailed description of the method is given in [33]).

From the numerical point of view the proposed approach has an advantage as compared with the above FFT methods. Indeed, first we managed to reduce the original evolutionary integral equation to a pure differential equation. Second, this equation could be formally solved analytically. To compute the operator exponent we applied a Padé approximation technique. This eventually allowed us to derive finite difference equations which approximate the original solution with the necessary order. These equations could be solved at the same grid as the diffusion part of the original PIDE, thus eliminating problems inherent to the FFT methods (interpolation of the FFT solution to the original grid). In addition, despite the original integral term is non-local, the rhs matrix $\mathcal{D}$ of the system of linear equations, obtained by applying our approach, is a band matrix in case of integer $\alpha_{R,L}$; i.e. it corresponds to a local approximation of the option price. Also, as it is shown in [18], at $\alpha \leq 0$ the complexity of our algorithm is much lower (almost 30 times faster than that of the FFT), while the accuracy is much better (for practical purposes we can use second or third order FD approximation, while getting same approximation within the FFT method significantly increases the complexity).

The complexity of the solution at $\alpha = 1$ is higher than that of the FFT. This in part is compensated by few factors: our algorithm provides the second order approximation in both space and time, and it does not require to re-interpolate the FFT results to the FD grid which was previously used to find solution for the diffusion part of the original PIDE. Also, it doesn't require a special treatment of the point $x = 0$ at $\alpha > 0$, as this was done in [9].

4.5. Boundary conditions. To recap, in the previous sections we transformed the solution of the original PIDE Eq. (10) to the following problems.

Method.

(1) Solve the equation

$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = L_R C(x, V_R, V_L, \tau), \tag{58}$$

where the operator L_R is defined in Eq. (12). To do that

(a) Solve the Eq. (20) using the scheme Eq. (23). Thus, we solve the following equations:

$$\begin{aligned}
Y_0 &= U_{n-1} + \Delta t F(t_{n-1}, U_{n-1}), \\
Y_j &= Y_{j-1} + \theta \Delta t \left[F_j(t_n, Y_j) - F_j(t_{n-1}, U_{n-1}) \right], \ j = 1, 2 \\
\hat{Y}_0 &= Y_0 + \theta \Delta t \left[F_0(t_n, Y_k) - F_0(t_{n-1}, U_{n-1}) \right], \\
\tilde{Y}_0 &= \hat{Y}_0 + \left(\frac{1}{2} - \theta \right) \Delta t \left[F(t_n, Y_k) - F(t_{n-1}, U_{n-1}) \right], \\
\tilde{Y}_j &= \tilde{Y}_{j-1} + \theta \Delta t \left[F_j(t_n, \tilde{Y}_j) - F_j(t_n, Y_k) \right], \ j = 1, 2 \\
U_n &= \tilde{Y}_k,
\end{aligned} \tag{59}$$

where

$$\begin{aligned}
F_0 &= \sigma \rho^R \sigma_V V_R \frac{\partial^2}{\partial x \partial V_R} \\
F_1 &= -\frac{1}{2} r_d + \left[\frac{1}{2}(r_d - r_f) - \frac{1}{2}\sigma^2 V_R - a_{1R}\sqrt{V_R} \right] \frac{\partial}{\partial x} + V_R \frac{\sigma^2}{2} \frac{\partial^2}{\partial x^2} \\
F_2 &= \kappa(1 - V_R) \frac{\partial}{\partial V_R} + \frac{\sigma_V^2 V_R}{2} \frac{\partial^2}{\partial V_R^2}, \\
a_{1R} &= \int_0^\infty (e^y - 1 - y) \lambda \frac{e^{-|y|\nu_R}}{|y|^{1+\alpha}} dy \\
&= \begin{cases} \lambda \Gamma(-\alpha) \left[(\nu_R - 1)^\alpha + (\alpha - \nu_R)\nu_R^{\alpha - 1} \right], & \alpha < 2, \ \mathbb{R}(\nu_R) > 1 \\ \lambda \left[-\frac{1}{\nu_R} + \log\left(\frac{\nu_R}{\nu_R - 1} \right) \right], & \alpha = 0 \\ \lambda \left[1 + (\nu_R - 1) \log\left(\frac{\nu_R - 1}{\nu_R} \right) \right], & \alpha = 1 \end{cases} \\
F &\equiv F_0 + F_1 + F_2
\end{aligned} \tag{60}$$

We use this representation if $\alpha_R < 0$. Otherwise we assume $a_{1R} = 0$. The initial data for this step are taken from the previous time level, or from the terminal condition at the first level.

(b) Solve the second equation in Eq. (19) using the method of the previous section. In case $\alpha_R < 1$ we use a reduced integral (the second and third terms under the integral are integrated out to produce a_{1R} term in the previous step). Then we have to solve 3 one-dimensional unsteady equations of the type Eq. (43) or Eq. (46) for integer values of α closest to real α_R, and use these results to interpolate to the real value of α_R. Each solution for an integer $\alpha < 1$ requires just one sweep, and for

$\alpha = 1$ - multiple such steps. The initial data are taken from the previous step in time.
In case $\alpha_R > 0$ we keep all terms under the integral (so $a_{1R} = 0$) and compute the price for 3 closest integer values of α. We then use interpolation if real $\alpha_R \le 1$, or extrapolation if $1 < \alpha_R < 2$.
As the initial data use the solution of the step 1a.

(c) Repeat step 1a and as the initial data use the solution of the step 1b.

(2) Solve the equation

$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = L_L C(x, V_R, V_L, \tau), \tag{61}$$

where the operator L_L is defined in Eq. (12). To do that

(a) Solve the Eq. (20) using the scheme Eq. (23). Thus, we solve the Eq. (59) where

$$\begin{aligned}
F_0 &= \sigma \rho^L \sigma_V V_L \frac{\partial^2}{\partial x \partial V_L} \\
F_1 &= -\frac{1}{2} r_d + \left[\frac{1}{2}(r_d - r_f) - \frac{1}{2}\sigma^2 V_L - a_{1R}\sqrt{V_L}\right] \frac{\partial}{\partial x} + V_L \frac{\sigma^2}{2} \frac{\partial^2}{\partial x^2} \\
F_2 &= \kappa(1 - V_L) \frac{\partial}{\partial V_L} + \frac{\sigma_V^2 V_L}{2} \frac{\partial^2}{\partial V_L^2}, \\
a_{1L} &= \int_{-\infty}^{0} (e^y - 1 - y) \lambda \frac{e^{-|y|/\nu_l}}{|y|^{1+\alpha}} dy \\
&= \begin{cases} \lambda \Gamma(-\alpha) \left[(\nu_L + 1)^\alpha - (\alpha + \nu_L)\nu_R^{\alpha - 1}\right], & \alpha < 2, \ \mathbb{R}(\nu_L) > 0 \\ \lambda \left[\frac{1}{\nu_L} + \log\left(\frac{\nu_L}{\nu_L + 1}\right)\right], & \alpha = 0 \\ \lambda \left[-1 + (\nu_L + 1)\log\left(\frac{\nu_L + 1}{\nu_L}\right)\right], & \alpha = 1 \end{cases} \\
F &\equiv F_0 + F_1 + F_2
\end{aligned} \tag{62}$$

We use this representation if $\alpha_L < 0$. Otherwise we assume $a_{1L} = 0$. The initial data for this step is the solution of the step 1c.

(b) Solve the second equation in Eq. (19) using the method of the previous section. In case $\alpha_L < 1$ we use a reduced integral (the second and third terms under the integral are integrated out to produce a_{1L} term in the previous step). Then we have to solve 3 one-dimensional unsteady equations of the type Eq. (43) or Eq. (46) for integer values of α closest to real α_L, and use these results to interpolate to the real value of α_L. Each solution for an integer $\alpha < 1$ requires just one sweep, and for $\alpha = 1$ - multiple such steps. The initial data are taken from the previous step in time.
In case $\alpha_L > 0$ we keep all terms under the integral (so $a_{1L} = 0$) and compute the price for 3 closest integer values of α. We then use interpolation if real $\alpha_L \le 1$, or extrapolation if $1 < \alpha_L < 2$.
As the initial data use the solution of the previous step 2a.

(c) Repeat step 2a and as the initial data use the solution of the step 2b.

■

As is easy to see, all the equations we have to solve are either explicit, or 1D implicit equations in space. This determines the boundary conditions we need to impose at every step of our splitting method. In more detail, this is

(1) First equation in Eq. (59): no boundary conditions are required.

(2) Second equation in Eq. (59) at $j = 1$: This equation is in a log-return space. Therefore, we impose same boundary conditions that were discussed at the end of section 2.

(3) Second equation in Eq. (59) at $j = 2$: This equation is in V_R space. At $V_R \to \infty$ according to the analysis of section 2 we use the boundary condition $\frac{\partial}{\partial V_R} = 0$ which means $F_2 = 0$. At $V_R \to 0$ the ratio of the convection to diffusion term in F_2 is positive, therefore as the boundary condition we have to us the equation itself. In other words, at $V_R = 0$ we obtain $F_2 = \kappa \frac{\partial}{\partial V_R}$, and therefore the second equation in Eq. (59) becomes hyperbolic. Therefore we have to address two issues. First, we need to approximate the first derivatives (at $V_R \to \infty$ and at $V_R = 0$) with the second order of approximation to preserve the second order of the whole scheme. Second we have to chose a correct approximation (downward or upward) to preserve the stability of the scheme. It can be verified that as $\kappa > 0$ and we solve the PDE backward in time one must use a forward approximation of the first derivative at $V_R = 0$ and backward approximation at $V_R \to \infty$. If then she uses a 3-points approximation of the first derivative, the matrix of the rhs remains three-diagonal while at the first and the last row it has three elements instead of two. However, a slight modification of the LU solver can still be applied.

(4) Third equation in Eq. (59): no boundary conditions are required.

(5) Fourth equation in Eq. (59) at $j = 1$: This equation is in a log-return space. Therefore we impose same boundary conditions that were discussed at the end of section 2.

(6) Fourth equation in Eq. (59) at $j = 2$: This equation is in V_R space. So se can impose the same boundary conditions that were discussed in item 3.

(7) Step 1b: This is a pure jump equation transformed to a PDE according to our method. Thus, this PDE is defined in the log-return space. Therefore, we impose same boundary conditions that were discussed at the end of section 2.

For the next set of steps of the above described method the boundary conditions are imposed by analogy.

5. Finite-difference scheme

To solve one-dimensional unsteady equations given in the previous section we need a reliable finite difference scheme. We have already mentioned some crucial requirements that our method should obey, namely: this scheme has to be at least of the second order of approximation in time and space, be fast, stable with respect to the discontinuities in the initial data at $S = K$ (payoff) and at the barrier(s). In general in our numerical experiments we used the same approach as in [29] with some changes that are described below.

5.1. Order of approximation. Aside from a detailed discussion just mention that we would prefer to work with a high order compact (HOC) scheme which provides a fourth order accuracy in space $O(h^4)$, h is the space step. This would provide us with a necessary relative accuracy to compute option values with an error up to 0.5 cents (rawly for a 500$ stock this is a relative error of 10^{-5}) while still preserving a finite difference grid from using very small steps in space. On the other hand a high order approximation in time would be also very desirable. As shown in [30] the HOC scheme for the heat equation with smooth initial conditions attains clear fourth-order convergence but fails if non-smooth payoff conditions are used. Therefore it is impossible to resolve this problem using just the HOC scheme in space as some authors claimed.

For homogeneous parabolic partial differential equations with non-smooth initial data (like we have) a family of higher order accurate smoothing schemes has recently been developed (see a literature survey in [19]). Convergence results and numerical experiments show that these schemes can be more robust than the well known Rannacher smoothing schemes [26] with respect to spurious oscillations generated through high frequency components in non-smooth initial or boundary data. Usually these schemes are constructed based on Pade approximation of the evolutionary operator. Unfortunately, to the best of our knowledge these schemes mostly operate with a second order discretization in space. Using a HOC scheme in space together with the Pade scheme brings some extra complexity that we would prefer to eliminate because of the speeding requirement. The other problem is that the Pade scheme to be efficient should be parallelized which results in necessity to work with a complex arithmetic. Fortunately, CUBLAS library does support complex arithmetic for CUDA, therefore working with the HOC-PADE schemes could be a possible challenge for the future.

5.2. Grid. As mentioned in [29] and many others papers it is useful to apply non-uniform meshes in all spatial directions such that relatively many mesh points lie in the neighborhood of $S = K$ and $v = 0$. This, first, greatly improves the accuracy of the scheme as compared to the uniform meshes. Indeed, the Eq. (4) is very sensitive to localization errors when S is in the vicinity of K, since the first derivative of the payoff doesn't exist at this point. Therefore, to increase accuracy it would be reasonable to use an adaptive mesh with high concentration of the mesh points around $S = K$, while a rarefied mesh could be used far away from this area. For the barrier options the situation is even more complicated [31]. Here we consider only continuously sampled barriers, so it is sufficient to place the barriers on the boundaries of the grid and enforce a boundary condition of zero option value. The gradient of the option price is discontinuous at the barriers because we never solve the pricing equation there. Therefore it is reasonable to concentrate the grid cells in the vicinity of the barriers as well [9]. Also at $v = 0$ the Eqs. (58, 61) become convection dominated so it reasonable to concentrate meshes at this point as well as at the initial level of v.

We build a non-uniform grid using a coordinate transformation. Let us choose a "factorize" coordinate transformation in sense that we transform coordinate $x = \log S$ independently of the other coordinates. In other words we use a map $x \leftrightarrow X, V_R \leftrightarrow v_r, V_L \leftrightarrow v_l, t \leftrightarrow \tau$ of the form

$$x = x(X), \quad v_r = V_R(v_r), \quad v_l = V_L(v_l), \quad \tau = T - t. \tag{63}$$

We use a transformation that has been proposed in [31] with the idea of concentrating finite difference grid points near critical points such as barrier prices and strike. We define the Jacobian of this transformation

$$J(X) = \frac{dx(X)}{dX}, \tag{64}$$

[9] For the discretely sampling barriers this could result in some problems, see discussion in [31]

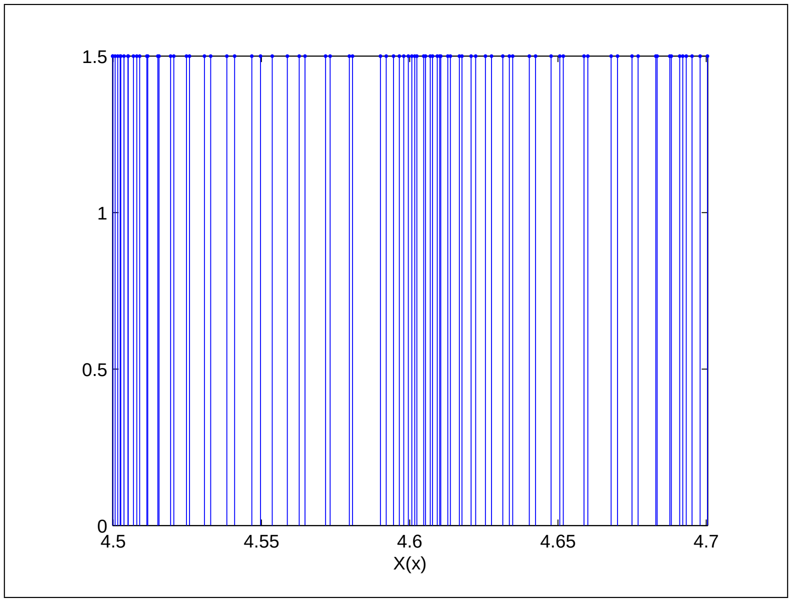


FIGURE 2. New grid obtained from the uniform grid in x with the transformation Eq. (65)

.

as

$$J(X) = A\left[\sum_{k=1}^{k=3} J_k(z)^{-2}\right]^{-1/2} \tag{65}$$
$$J_k(X) = \left[\alpha_k^2 + (x(X) - B_k)^2\right]^{1/2}$$

Parameters B_k correspond to the critical points, i.e. in our case $B_1 = \mathcal{L} \equiv \log L, B_2 = \mathcal{H} \equiv \log H, B_3 = \mathcal{K} \equiv \log K$. Parameters A and $\alpha_k, k = 1, 2, 3$ are adjustable. Setting $\alpha_k \ll \mathcal{H} - \mathcal{L}$ yields a highly nonuniform grid while $\alpha_k \gg \mathcal{H} - \mathcal{L}$ yields a uniform grid.

For the transformation given by the Eq. (65) near the strike and barriers the global Jacobian $J(X)$ is dominated by the behavior of the local $J_k(X)$, but the influence of nearby critical points ensures that the transitions between them are smooth. In general the global Jacobian must be integrated numerically to yield the transformation $x(X)$. Any standard ODE integrator (for instance, Matlab ode45) could be used for that using the initial condition $x(0) = \mathcal{L}$. To obey the second boundary condition $x(X_{max}) = \mathcal{H}$ one can vary the adjustable parameter A. Since $x(X_{max})$ is monotonically increasing with A the numerical iterations are guaranteed to converge.

In Fig. 2 we present a map of the new grid obtained from the original uniform x grid by using the transformation Eq. (65). The new grid in x contains 41 nodes distributed from $\log(L)$ to $\log(H)$. Value of parameters used in this example are: $H = 110, L = 90, K = 100, \alpha_H = \alpha_L = (\log H - \log L)/30, \alpha_K = (\log H - \log K)/10$.

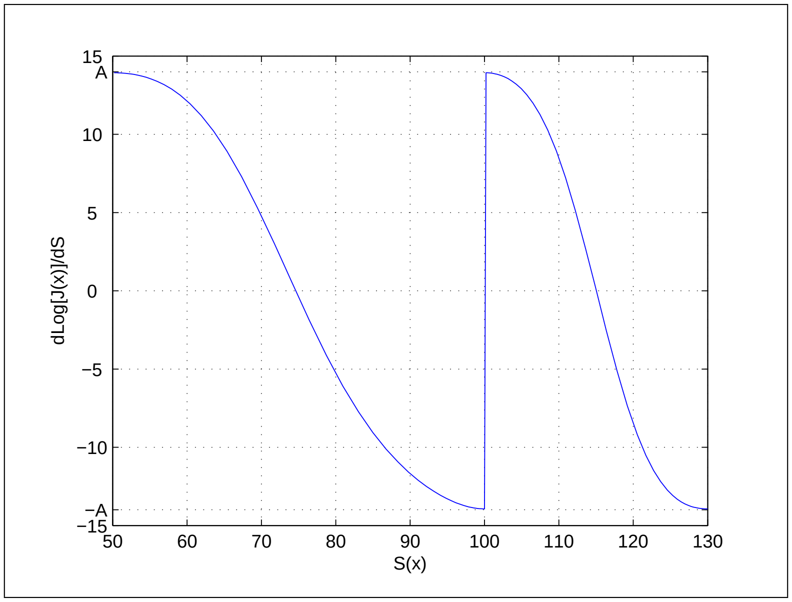


FIGURE 3. $d\ln J(X)/dX$ as a function of $S(X)$. Parameters for this test are given after the Fig. 2 while the barriers moved to $H = 130, L = 50$.

Note that for the transformation Eq. (65)

$$\frac{d\ln J(x)}{dx} = \frac{dJ}{dx} = A\left\{\sum_{k=1}^{3}\frac{x-B_k}{\left[\alpha_k^2+(x-B_k)^2\right]^2}\right\}\left[\sum_{k=1}^{3}\frac{1}{\alpha_k^2+(x-B_k)^2}\right]^{-3/2}$$

$$\approx A\left[\frac{1}{(x-\log H)^3}+\frac{1}{(x-\log K)^3}+\frac{1}{(x-\log L)^3}\right]\cdot \tag{66}$$

$$\left[\frac{1}{(x-\log H)^2}+\frac{1}{(x-\log K)^2}+\frac{1}{(x-\log L)^2}\right]^{-3/2}.$$

This function is bounded and changes within the range $(-A, A)$ as can be seen in Fig. 3. Thus $|d\ln J(x)/dx|$ is bounded.

To preserve monotonicity of the grid as well as monotonicity of the grid steps $h_i,\ i = 1, N$, after A and the Jacobian are computed we make smoothing of the grid running a robust local regression with a moving average which uses weighted linear least squares and a second degree polynomial model. So the new grid steps are $hh_i,\ i = 1, N$). We then re-normalize the grid steps to have the grid fitted the original boundaries, i.e. we compute $C = (\sum_i h_i)/(\sum_i hh_i)$ and then reassign $h_i = C \cdot hh_i,\ i = 1, N$. The span for the moving average is equal to 10.

Further for the variables v_r and v_l we use same type of the transformation as in the Eq. (65). Here as the critical points B_k we choose $B_1 = 0$ and $B_2 = V_{i0},\ i = R, L$, where the later is the initial level of the activity.

In Fig. 4 we present a map of the new grid obtained from the original uniform $x - V_R$ grid by using the transformation Eq. (65). The new grid contains 101 nodes in X distributed from $\mathcal{L}$ to $\mathcal{H}$, and 61 nodes in v_R distributed from 0 to V_{max}. Value of parameters used in this example are: $H = 110, L = 90, K = 100, \alpha_H = \alpha_L =$

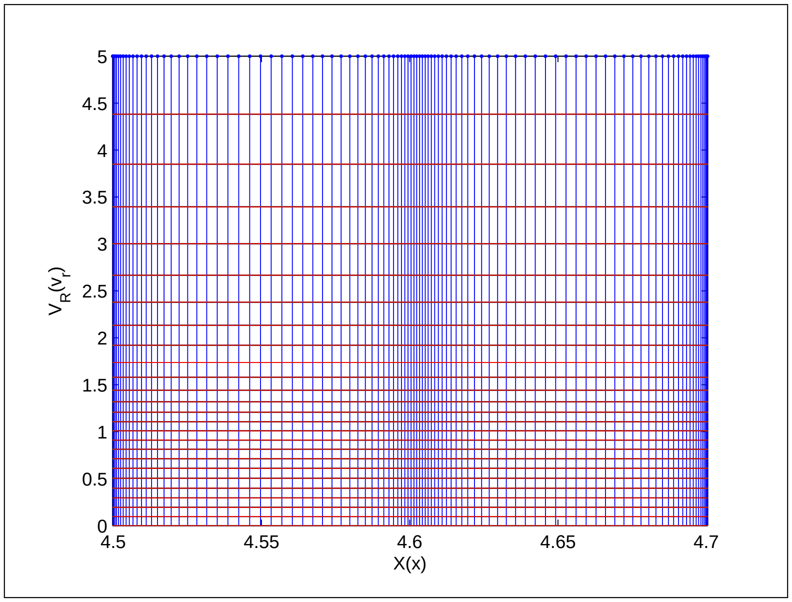


FIGURE 4. New grid obtained from the uniform grid in x, v_r with the transformation Eq. (65)

.

$(\log H - \log L)/60, \alpha_K = (\log H - \log K)/20, \alpha_0 = \alpha_{v0} = V_{max}/20, V_{max} = 1.5$ is a maximum value of V_R and V_L on the grid, and $V_{R0} = 0.3$.

6. Numerical experiments

First, to make comparison with the results given in [29] we used their case 4, i.e. we priced double barrier call option with the initial data given in Tab. 1.

S	K	L	H	T	r_d	r_f
100	100	80	150	0.25	0.0507	0.0469

TABLE 1. Option parameters used in our numerical experiments

For down-and-out call options the grid described in the previous section worked well. However, for double barrier call options at $t = 0$ there exists a discontinuity in the option price at the upper barrier. At $t > 0$ this discontinuity decays in the direction of the lower barrier (downward), while the price field close to the upper barrier is still characterized by high gradients which exist for some characteristic time of decay. Therefore compressing grid cells close to this barrier produces high gradients in the numerical solution as well. That is not a problem if one uses a fully implicit 2d finite difference scheme for approximation of the Eq. (20). However, under the splitting approach of [29] which includes some explicit steps this could require a very small initial step in time in order to avoid negative prices in the solution.

To address this we used just a minor compression of the grid close to the upper barrier, i.e. $\alpha_H = 1$, while still compressed the grid at $S = K$ by choosing $\alpha_K = 5$. We also used $\alpha_0 = \alpha_{V0} = 10$. The number of steps in x directions was 50, and

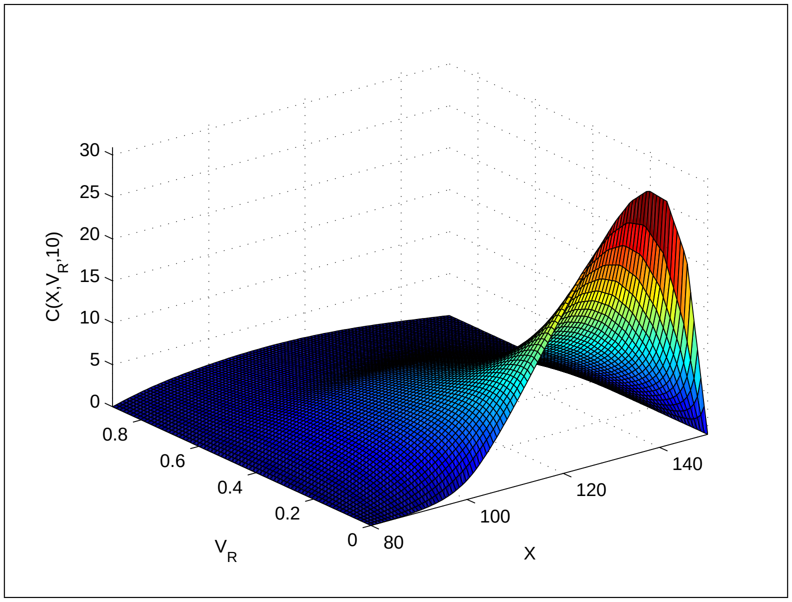


FIGURE 5. Price of a double barrier call option obtained within the Heston model with no jumps. Initial data are given in Tab. 1.

in $V_i, i = L, R$ direction was 25. As the upper bound of the grid in $V_i, i = L, R$ direction we used $V_{max} = 0.9$.

Also at the first 3 steps in time instead of using the ADI scheme of [29] we applied a fully implicit approximation of 2D Eq. (20) and solved it in time using the Euler scheme (thus, using Rannacherś approach to smooth initial discontinuities).[10]

Certainly, as gradients of the flow decrease with time, the time step could be increased. Therefore we also applied a smooth increase of the time step based on the analysis of the price gradient field over the whole region. We used the initial time step to be $\tau = 0.001$ yrs, and the maximum time step $\tau_{max} = 0.008$ yrs.

For the chosen ADI scheme matrices of all first and second derivatives could be pre-computed. Following the analysis in [20, 29], in a special case when the flow is inwind, i.e. directed from the upper boundary to the internal area (this occurs when the condition $0.5\sigma_V^2 + \kappa(\Theta - V_i) < 0$ is true [20]), we used an one-sided approximation of the first derivative in V in order to avoid spurious oscillations observed in the numerical solution when the ration $\kappa/\sigma_V \gg 1$.

In our first test we used the described scheme to price a double barrier call option within the Heston model with no jumps. We compared the results with analogous results kindly provided by Prof. K. Hout. Our results are given in Fig. 5 and with a high accuracy are identical to that of K. Hout. This is clear because we used the same ADI approach so the difference was basically in how we built the non -uniform grid. Also we experimented with another ADI scheme proposed by Hundsdorfer and Verwer (which as well is described in detail in [29]) with the recommended values of parameters $\theta = 1/2 + \sqrt{3}/6,\ \mu = 1/2$, and did found a very good agreement with the solution obtained using the Hout and Welfert scheme. The total number of steps in time was 250 which in our example took 91 sec at 3.2 Ghz PC to calculate the solution.

[10] A more sophisticated approach described in [19] could also be applied here.

As we basically use same discretization on a uniform grid as in the reference above [11] we didn't make special tests to investigate the order of convergency of the scheme. Details on this subject could be obtained from [20, 29].

Having ensured that the finite difference scheme provides the correct results in the test cases, we ran a series of tests in which the solution of the SSM model for double barrier call option with the initial parameters given in Tab. 1 was studied. We describe the results of three tests. For each test we computed the solution first with no jumps, and then with jumps. Also as our model is 3D and unsteady we present numerical results of every test in three figures. The first figure is representing the option price in coordinates (x, V_R) while the third coordinate V_L is fixed. We then present 6 different plots corresponding to the different values of V_L, where V_L runs from 0 to V_{max}. Namely, we use V_L-grid nodes 4, 8, 12, 16, 20, 24, while the total number of the grid nodes in this direction is 25. The second figure displays the results in coordinates (x, V_L) while V_R is the running coordinate, and the V_R-grid nodes are 4, 8, 12, 16, 20, 24 out of total 25 nodes. The third figure contains 6 plots in coordinates (V_R, V_L), and X is the running coordinate. We display plots related to the X grid nodes 2, 10, 20, 30, 40, 49 out of the total 50 nodes.

Test 1. The initial parameters of the SSM model used in this test are given in Tab. 2.

σ	σ_V	ρ_R	ρ_L	κ	Θ	V_{R0}
1	0.5	-0.1	-0.1	2.5	0.06	0.5
ν_R	ν_L	α_R	α_L	λ_R	λ_L	V_{L0}
2.5	2.5	-0.1	-0.1	3	3	0.5

Table 2. Model parameters used in Test 1.

This test uses fully symmetric parameters for positive and negative jumps. The results of calculation with no jumps in coordinates V_L, V_R are given in Fig. (6) [12], and with jumps - in Fig. (7-9). Also Tab. 3 displays call option prices $V(S, V_R, V_L)$ at several selected points.

The total time of calculation is 0.74*250 = 185 sec in case of no jumps, and 8.6*250 = 2150 sec with jumps. This is because in our splitting procedure we make one "no-jumps" step for the "R" equation and one "no-jumps" step for the "L" equation. Thus, the total time is twice as big as in our previous example with the Heston model. In case of jumps our splitting algorithm uses 4 "no-jumps" sweeps as well as two "jumps" sweeps, and in addition it uses a quadratic interpolation in α. Therefore, at every time step we solve the pure jump equation for three different values of α, and then interpolate as that was described in the previous sections. This means that the pure calculation time of one sweep for the pure jump equation is (8.6 - 2*0.74)/2/3 = 1.18 sec. This is almost 3 times more than it is necessary to solve a no-jump equation (a convective-diffusion equation). To understand this let us remind that the matrix of the rhs part obtained by applying

[11] But a different transformation, therefore our non-uniform grid differs from that of K. Hout, but not much.

[12] For the sake of brevity we omit the results in the other pairs of coordinates since they are of less interest

S	V_R	V_L	C (no jumps)	$C(jumps)$
90.39	0.24	0.24	1.887	2.329
100.39	0.24	0.24	3.512	4.014
110.64	0.24	0.24	4.495	4.793
120.31	0.24	0.24	4.552	4.416
132.01	0.24	0.24	3.453	2.962
90.39	0.103	0.24	2.263	3.477
100.39	0.103	0.24	4.516	6.536
110.64	0.103	0.24	6.231	8.491
120.31	0.103	0.24	6.699	8.304
132.01	0.103	0.24	5.348	5.833
90.39	0.24	0.103	2.268	2.891
100.39	0.24	0.103	4.532	5.070
110.64	0.24	0.103	6.262	6.165
120.31	0.24	0.103	6.739	5.721
132.01	0.24	0.103	5.385	3.836
90.39	0.103	0.103	2.083	3.756
100.39	0.103	0.103	5.089	7.820
110.64	0.103	0.103	8.528	11.140
120.31	0.103	0.103	10.593	11.495
132.01	0.103	0.103	9.540	8.321

Table 3. Call option prices at several selected points obtained in Test 1.

the finite difference approximation to the pure jump equation is banded but not tridiagonal. This increase in the number of bands results in a corresponding overhead of the computational time.

At the same time performance of the algorithm could be significantly improved by using parallel calculations. Indeed, calculations of the option prices at different α are independent and could be done in parallel. For instance, one can make use of a new Matlab parallel toolbox and run Matlab at a multicore box, or in any other suitable way. If we do that the total computational time in case of jumps drops down to 3.84 sec per time step (i.e. almost 2.2 times faster), and the total time becomes 560 sec.

The results of the symmetric test with no jumps at every plain V_R = const or V_L = const look pretty similar to what is known from modeling prices of barrier options within the Heston model. In coordinates V_L, V_R at x = const the plots seem to be less trivial. In these coordinates the option price surfaces close to the lower and upper barriers have a different shape and significantly different values. Accounting for symmetric jumps just slightly changes the shape of the option price surfaces in coordinates x, V_L and x, V_R at given jumps activities. However, in coordinates V_R, V_L even such jumps change the surface shape close the lower barrier.

Test 2. The initial parameters of the SSM model used in this test are given in Tab. 4.

This test uses $\alpha_R = \alpha_L = 0$, and therefore corresponds to a well-known Variance Gamma model of D. Madan and E. Seneta [24]. This is the jump model with infinite activity but finite variation. We also consider asymmetric jump parameters ν_R and ν_L as well as asymmetric correlations ρ_R and ρ_L. The results of calculation with

σ	σ_V	ρ_R	ρ_L	κ	Θ	V_{R0}
1	0.5	-0.7	0.1	2.5	0.06	0.5
ν_R	ν_L	α_R	α_L	λ_R	λ_L	V_{L0}
2.5	3.5	0	0	3	3	0.5

Table 4. Model parameters used in Test 2.

no jumps in coordinates V_R, V_L are given in Fig. (10) (again the results in other pairs of coordinates are omitted for brevity), and with jumps - in Fig. (11-13).

The total time of calculations is 188 sec with no jumps, 575 sec with jumps and no parallel calculations, and 382 sec using parallel calculations. This is because at $\alpha = 0$ we use a special algorithm which is faster than a general one (see Section 4.4).

The results show that the asymmetric in ν jumps at $\alpha = 0$ change both the level of prices (the "no-jumps" case versus that with jumps) and the shape of the surface in coordinates V_R, V_L (compare Fig. (10) and Fig. (13)).

Test 3. The initial parameters of the SSM model used in this test are given in Tab. 5.

σ	σ_V	ρ_R	ρ_L	κ	Θ	V_{R0}
1	0.5	-0.7	0.1	2.5	0.06	0.5
ν_R	ν_L	α_R	α_L	λ_R	λ_L	V_{L0}
2.5	3.5	-0.7	0	3	3	0.5

Table 5. Model parameters used in Test 3.

In Test 3 negative jumps again are modeled using the VG model, whereas positive jumps follows the CGMY model with $\alpha_R = -0.7$. All other parameters are same as in Test 2. The results of calculation with no jumps in coordinates V_R, V_L are given in Fig. (14), and with jumps - in Fig. (15-17).

It is interesting to see that asymmetric $\alpha_i, i = R, L$ produce a qualitatively new effect. It consists in the appearance of a big dome close to ATM at moderate values of V in addition to a standard arc which is also close to ATM at small values of V.

The total time of calculations is 190 sec with no jumps, 950 sec with jumps and no parallel calculations, and 444 sec using parallel calculations.

7. Conclusion

In this paper we considered a problem of pricing barrier options within the SSM model proposed by Carr and Wu. They derived a 3D unsteady PIDE which describes evolution of the option price in time and 3D space. The 3D space here is the underlying spot price S, the activity of the positive jumps V_R, and that of the negative jumps V_R. This equation makes it difficult to solve it numerically in the case of exotic options because of high dimensionality and a presence of jumps. Monte Carlo is one of the approaches, however it is slow.

Instead, we proposed a numerical algorithm which significantly uses splitting of the original equation in a set of one-dimensional unsteady equations of the Black-Scholes type. We also showed how each integral in the above PIDE can be replaced with another one-dimensional unsteady equation using the results presented in our paper [18]. We also constructed our scheme to be of the second order of approximation in time and all spatial coordinates. In a series of tests presented at the end of the paper we prove a computational efficiency of the proposed algorithm.

Another advantage of the scheme is that for all non-jump and jump sweeps we use the same non-uniform grid, in contrast to the well-known FFT method of computations of the jump integral. The latter approach first uses one grid to solve the convective-diffusion part of the PIDE, and then uses another grid to compute the jump part. Thus, to proceed after the second sweep it needs to re-interpolate the obtained results to the original finite-difference grid used at the first step.

In all calculations we used Matlab 2009b and represented banded matrices as sparse. It is known, however, that it takes some time from Matlab to analyze a structure of the sparse matrix in order to apply an efficient LU factorization algorithm. Thus, performance of our scheme could be further improved, because we know in advance the exact banded structure of the matrices.

We also want to underline that, as follows from [18], at $\alpha > 0$ our algorithm becomes less efficient because at $\alpha = 1$ we need to do multiple steps to integrate a pure jump equation. This results in the increase of computational time for one jump sweep, approximately by factor 60 (from 3.8 sec (parallel case) to 228 sec). Therefore, this is our goal for the future to improve this result, perhaps by considering various alternative approaches in this particular case.

Acknowledgments

We thank Peter Forsyth, Karel In't Hout, Dmitry Kreslavsky, attendees of the "Global Derivatives and Risk, 2006" and "Computational Methods in Finance, 2007" conferences for useful comments. We assume full responsibility for any remaining errors.

References

[1] L. Andersen and J. Andreasen. Jump diffusion processes: volatility smile fitting and numerical methods for option pricing. *Review of Derivatives Research*, 4:231–262, 2000.

[2] D. S. Bates. The crash of '87: Was it expected? The evidence from option markets. *Journal of Finance*, 45(3):1009–1044, 1991.

[3] S. Boyarchenko and S. Levendorskii. *Non-Gaussian Merton-Black-Scholes Theory.* World Scientific, 2002.

[4] J. Cannon and Y. Lin. Classical and weak solutions for one-dimensional pseudo-parabolic equations with typical boundary data. *Annali di Matematica Pura ed Applicata*, 152(1):375–385, 1988.

[5] P. Carr and A. Mayo. On the numerical evaluation of option prices in jump diffusion processes. *The European Journal of Finance*, 13(4):353–372, 2007.

[6] P. Carr and L. Wu. Time-changed levy processes and option pricing. *Journal of Financial Economics*, 71:113–141, 2004.

[7] A. Cartea and D. del Castillo-Negrete. Fractional diffusion models of option prices in markets with jumps. *Physica A*, 374:749–763, 2007.

[8] R. Cont, editor. *Frontiers In Quantitative Finance: Volatility And Credit Risk Modeling.* Wiley Finance Press, 2009.

[9] R. Cont and E. Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential Lévy models. Technical Report 513, Rapport Interne CMAP, 2003.

[10] Y. dH́alluin, P. A. Forsyth, and K. R. Vetzal. Robust numerical methods for contingent claims under jump diffusion processes. *IMA J. Numerical Analysi*, 25:87–112, 2005.

[11] D.J. Duffy. *Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach.* The Wiley Finance Series, 2006.

[12] E.G. Dyakonov. Dierence schemes with a separable operator for general second order parabolic equations with variable coecient. *Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki*, 4(2):278–291, 1964.

[13] David Eberly. Derivative approximation by finite differences, March 2 2008. available online at http://www.geometrictools.com/Documentation/FiniteDifferences.pdf.

[14] L. Feng and V. Linetsky. Pricing options in jump-diffusion models: An extrapolation approach. *Operations Researsch*, 56(2):304–325, 2008.

[15] R.A. Fraser, W.J. Duncan, and A.R.Collar. *Elementary matrices*. Cambridge Univarsity Press, 1963.

[16] S. Ikonen and J. Toivanen. Efficient numerical methods for pricing american options under stochastic volatility. *Num. Meth. PDEs*, 24(1):104–126, 2007.

[17] K. J. In 't Hout and B. D. Welfert. Stability of ADI schemes applied to convection-diffusion equations with mixed derivative terms. *Applied Numerical Mathematics*, 57:19–35, 2007.

[18] A. Itkin and P. Carr. Using pseudo-parabolic and fractional equations for option pricing in jump diffusion models. *submitted to Appied Mathematical Finance*, 2010. available at `http://arxiv.org/PS_cache/arxiv/pdf/1002/1002.1995v1.pdf`.

[19] A.Q.M. Khaliq, B.A. Wade, M. Yousuf, and J. Vigo-Aguiar. High order smoothing schemes for inhomogeneous parabolic problems with applications in option pricing. *Numerical Methods for Partial Differential Equations*, 23(5):1249–1276, 2007.

[20] T. Kluge. *Pricing derivatives in stochastic volatility models using the finite difference method.* PhD thesis, Technische UniversitÄat Chemnitz, 2002.

[21] D. Lanser and J.G. Verwer. Analysis of operator splitting for advection-diusion-reaction problems from air pollution modelling. *Journal of Computational and Applied Mathematics*, 111(1-2):201–216, 1999.

[22] G.I. Marchuk. *Methods of Numerical Mathematics*. Springer-Verlag, 1975.

[23] O. Marom and E. Momoniat. A comparison of numerical solutions of fractional diffusion models in finance. *Nonlinear Analysis: Real World Applications*, 10:3435–3442, 2009.

[24] D. Madan and E. Seneta. The variance gamma (V.G.) model for share market returns. *Journal of Business*, 63(4):511–524, 1990.

[25] O. A. Oleinik and E. V. Radkevich. *Second order equations with non-negative characteristic form*. Kluwer Academic Publishers, 1973.

[26] R. Rannacher. Finite element solution of diffusion equation with irregular data,. *Numerical mathematics*, 43:309–327, 1984.

[27] A.A. Samarski. Economical dierence schemes for parabolic equations with mixed derivatives. *Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki*, 4(4):753–759, 1964.

[28] G. Strang. On the construction and comparison of dierence schemes. *SIAM J. Numerical Analysis*, 5:509–517, 1968.

[29] K. J. In' t Hout and S. Foulon. ADI finite difference schemes for option pricing in the Heston model with correlation. *International journal of numerical analysis and modeling*, 7(2):303–320, 2010.

[30] D.Y. Tangman, A. Gopaul, and M. Bhuruth. Numerical pricing of options using high-order compact finite difference schemes. *Journal of Computational and Applied Mathematics*, 218:27–0280, 2008.

[31] D. Tavella and C. Randall. *Pricing Financial Instruments. The Finite-Difference method.* Wiley series in financial engineering. John Wiley & Sons, Inc., New York,, 2000.

[32] N.N. Yanenko. *The method of fractional steps.* Springer-Verlag, 1971.

[33] K. Zhang and S. Wang. A computational scheme for options under jump diffusion processes. *International journal of numerical analysis and modeling*, 6(1):110–123, 2009.

Department of Finance and Risk Engineering, New York University Politechnic Institute, Brooklyn, New York, 11201, USA
E-mail: `aitkin@poly.edu`

Morgan Stanley and New York University
E-mail: `Peter.P.Carr@morganstanley.com`

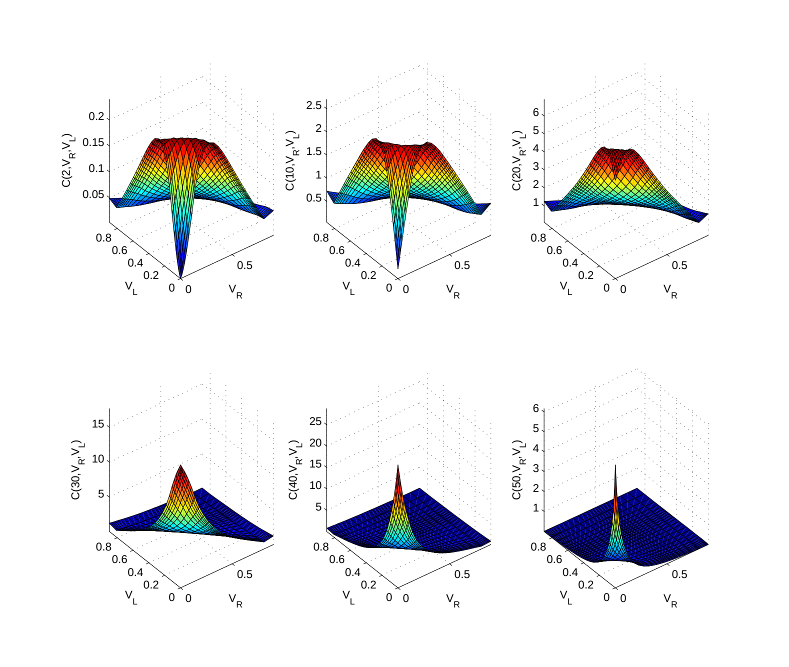


FIGURE 6. Test 1 with no jumps. Call option price in V_R, V_L coordinates, X runs from plot 1 to 6.

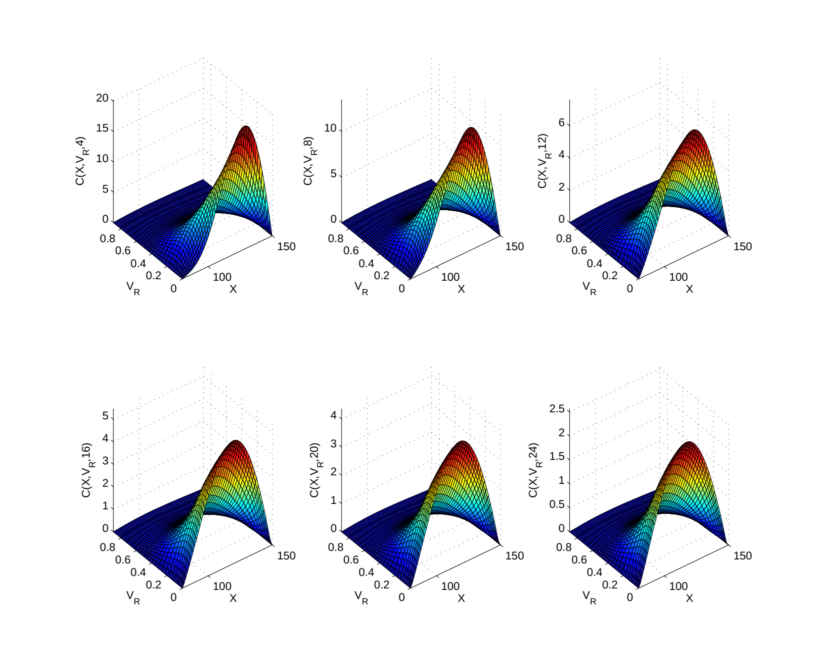


FIGURE 7. Test 1 with jumps. Call option price in X, V_R coordinates, V_L runs from plot 1 to 6.

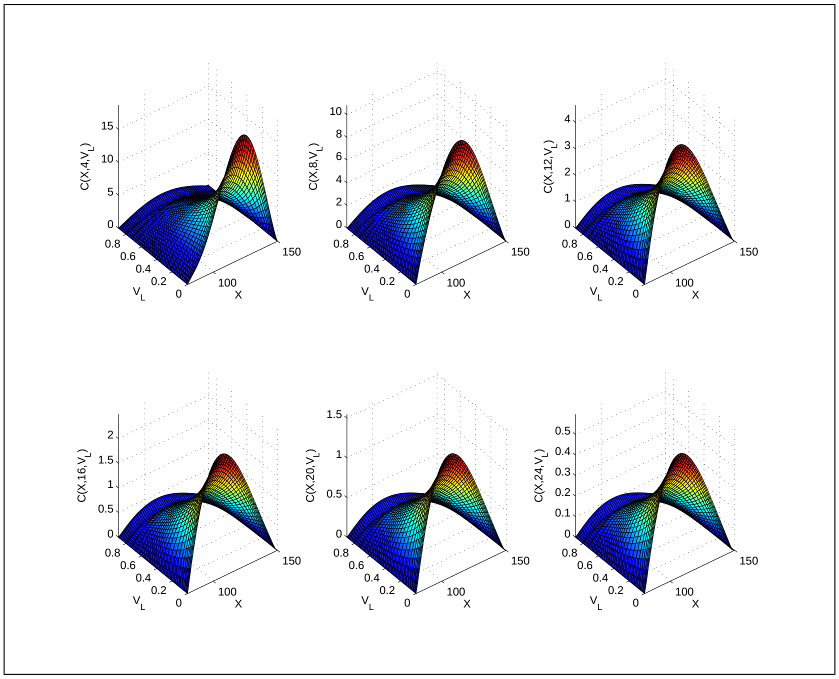


FIGURE 8. Test 1 with jumps. Call option price in X, V_L coordinates, V_R runs from plot 1 to 6.

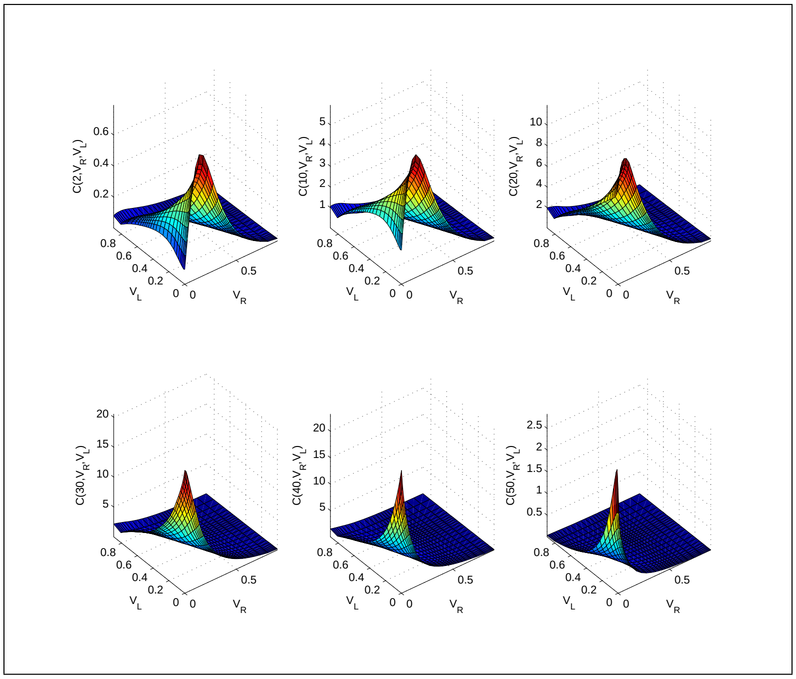


FIGURE 9. Test 1 with jumps. Call option price in V_R, V_L coordinates, X runs from plot 1 to 6.

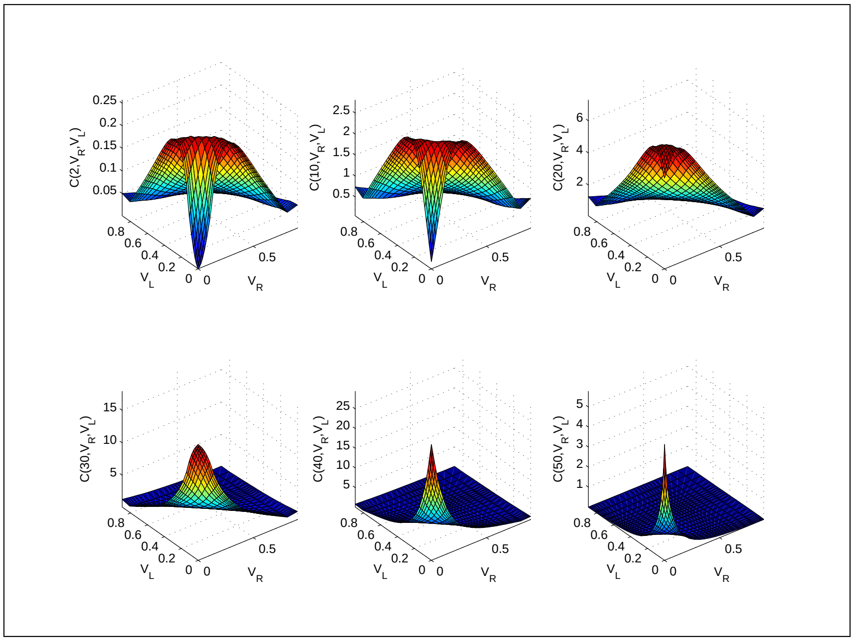


FIGURE 10. Test 2 with no jumps. Call option price in V_R, V_L coordinates, X runs from plot 1 to 6.

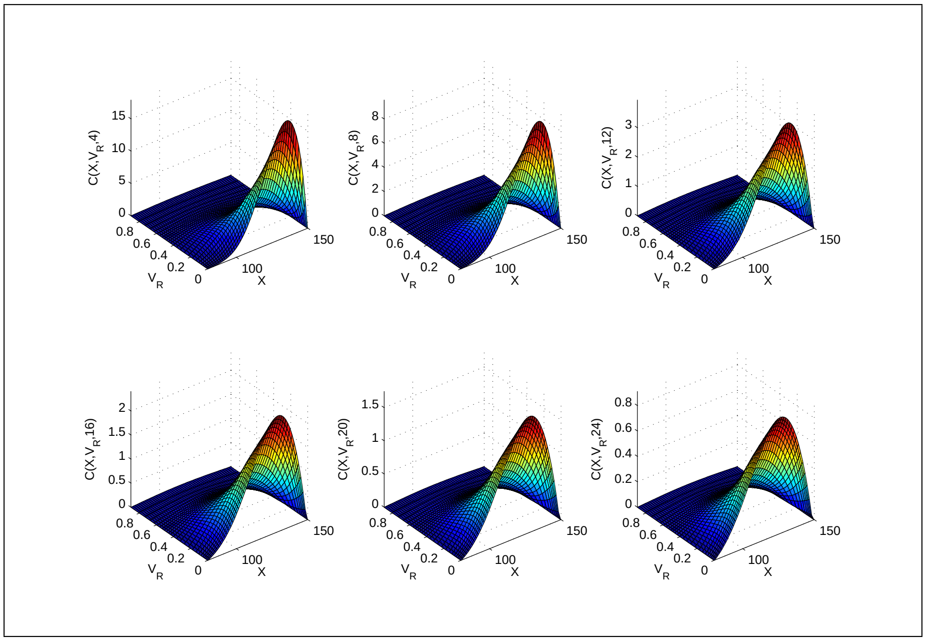


FIGURE 11. Test 2 with jumps. Call option price in X, V_R coordinates, V_L runs from plot 1 to 6.

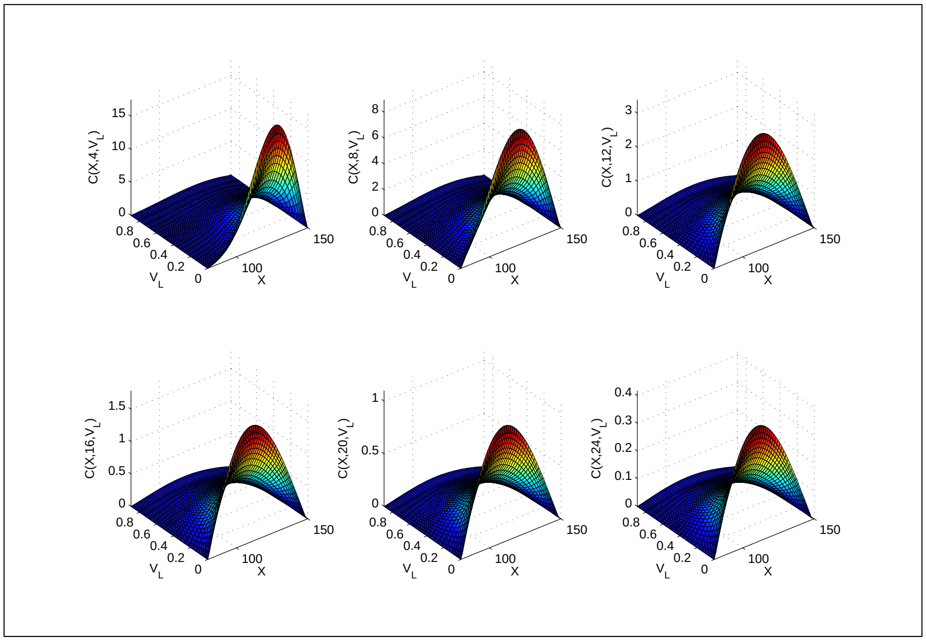


Figure 12. Test 2 with jumps. Call option price in X, V_L coordinates, V_R runs from plot 1 to 6.

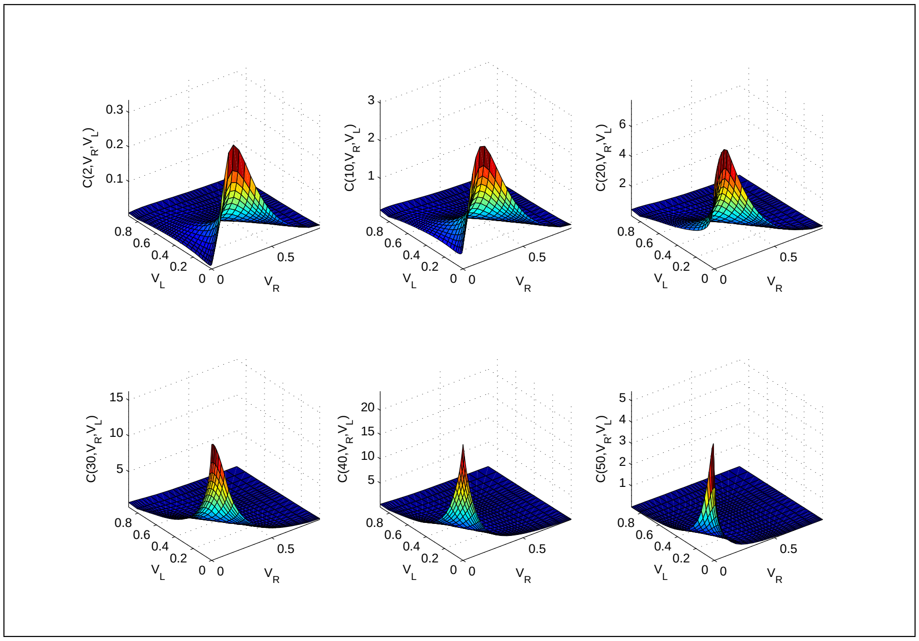


Figure 13. Test 2 with jumps. Call option price in V_R, V_L coordinates, X runs from plot 1 to 6.

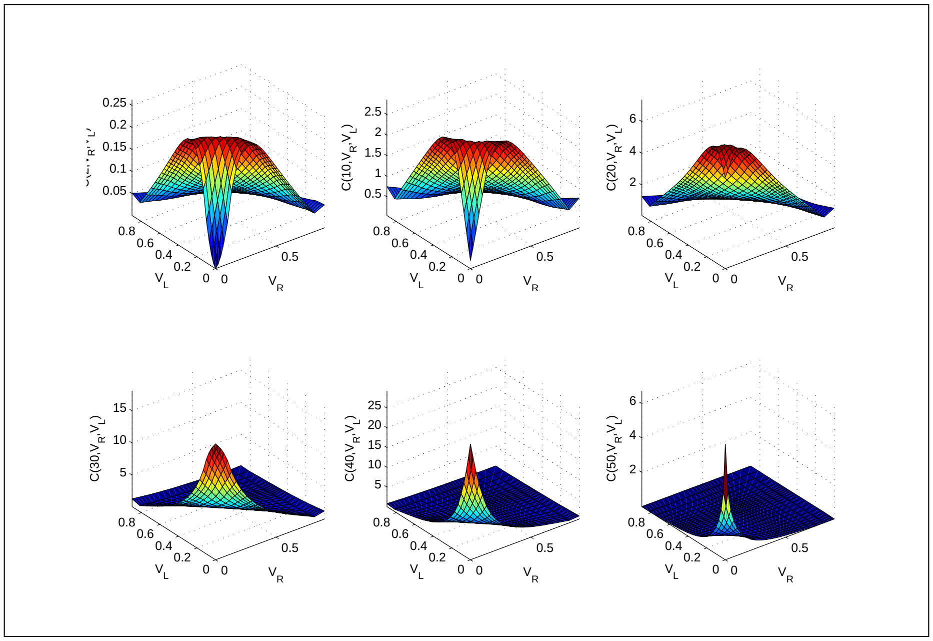


FIGURE 14. Test 3 with no jumps. Call option price in V_R, V_L coordinates, X runs from plot 1 to 6.

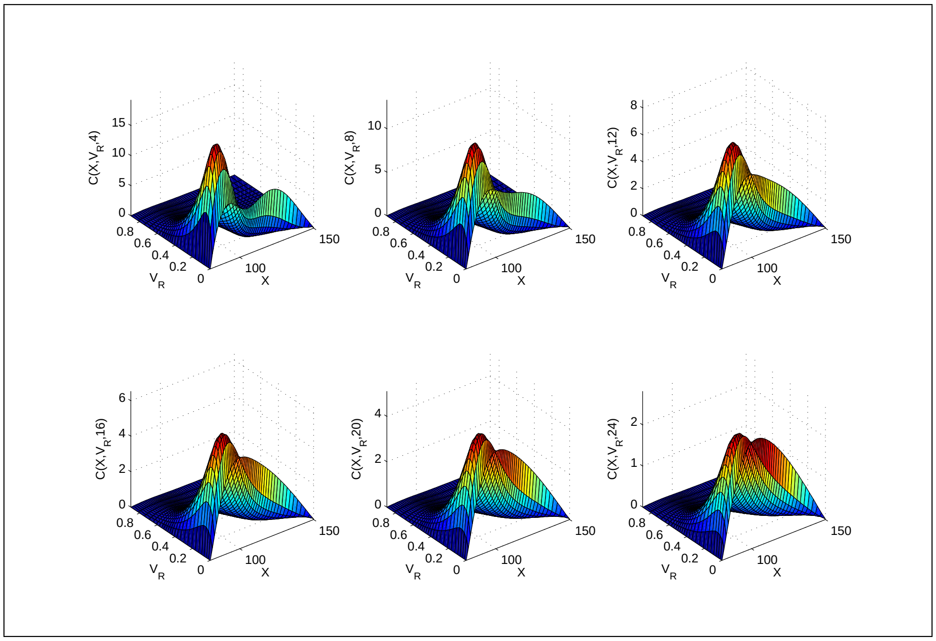


FIGURE 15. Test 3 with jumps. Call option price in X, V_R coordinates, V_L runs from plot 1 to 6.

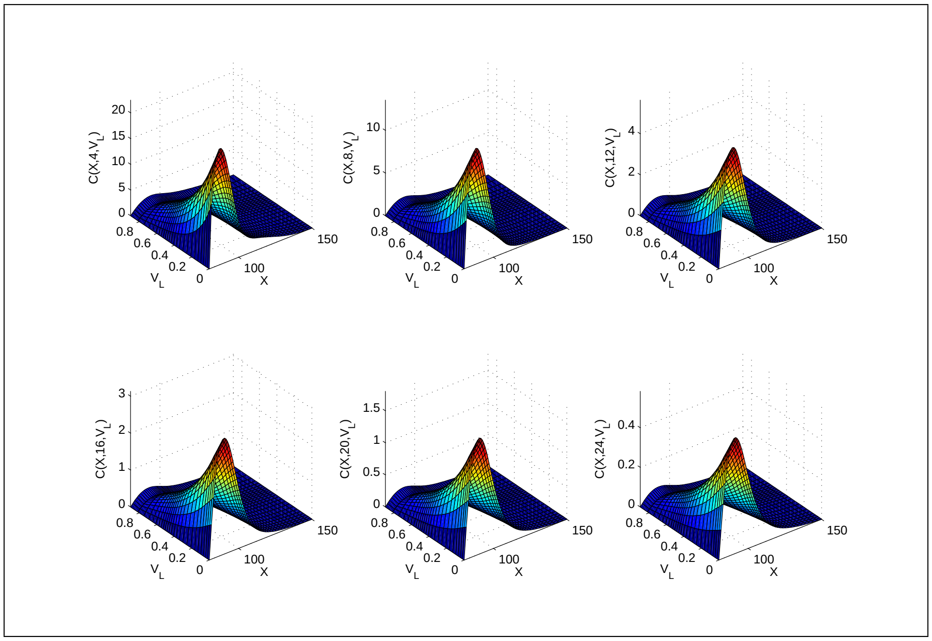


FIGURE 16. Test 3 with jumps. Call option price in X, V_L coordinates, V_R runs from plot 1 to 6.

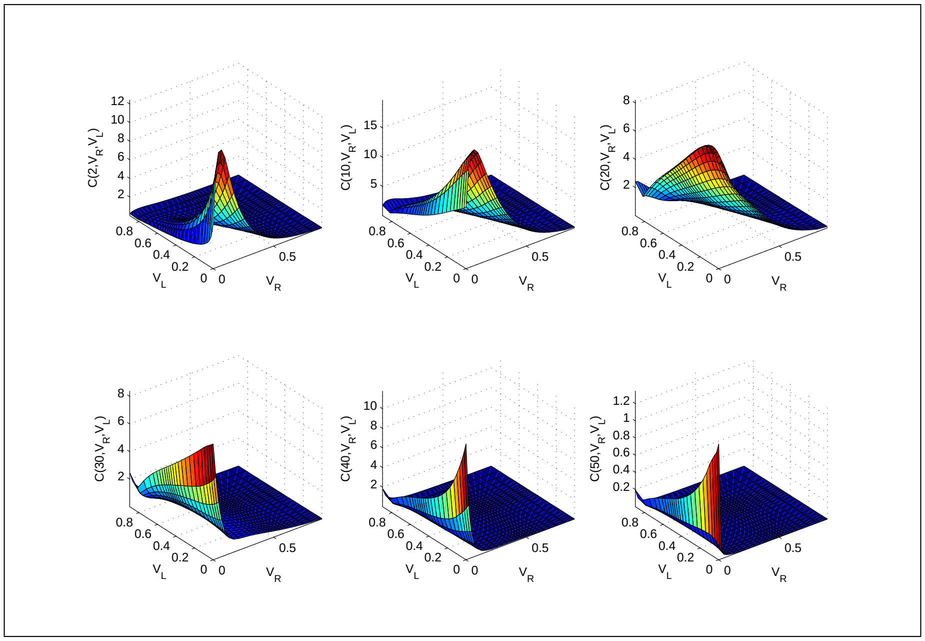


FIGURE 17. Test 3 with jumps. Call option price in V_R, V_L coordinates, X runs from plot 1 to 6.