Using pseudo-parabolic and fractional equations for option pricing in jump diffusion models

Andrey Itkin[1]

[1]HAP Capital LLC and NYU Poly

(joined work with Peter Carr)

6-th World Bachelier Congress, Toronto, 2010

Outline

In mathematical finance a popular approach for pricing options under some Lévy model is to consider underlying that follows a Poisson jump diffusion process. As it is well known this results in a partial integro-differential equation (PIDE) that usually does not allow an analytical solution while numerical solution brings some problems. In this work we elaborate a new approach on how to transform the PIDE to some class of either so-called pseudo-parabolic equations or fractional equations. They both are known in mathematics but are relatively new for mathematical finance. As an example we discuss several jump-diffusion models which Lévy measure allows such a transformation.

Existing approaches to solve PIDE

1. Amim (1993) - explicit multinomial tree
2. Tavella and Randall (2000) - Picard iterations

Existing approaches to solve PIDE

1. Amim (1993) - explicit multinomial tree
2. Tavella and Randall (2000) - Picard iterations
3. Andersen and Andreasen (2000) - forward equation, European call, second order accurate unconditionally stable operator splitting (ADI), FFT

Existing approaches to solve PIDE

1. Amim (1993) - explicit multinomial tree
2. Tavella and Randall (2000) - Picard iterations
3. Andersen and Andreasen (2000) - forward equation, European call, second order accurate unconditionally stable operator splitting (ADI), FFT
4. Cont and Voltchkova (2003) - discretization that is implicit in the differential terms and implicit in the integral term, it converges to a viscosity solution.

Existing approaches to solve PIDE

1. Amim (1993) - explicit multinomial tree
2. Tavella and Randall (2000) - Picard iterations
3. Andersen and Andreasen (2000) - forward equation, European call, second order accurate unconditionally stable operator splitting (ADI), FFT
4. Cont and Voltchkova (2003) - discretization that is implicit in the differential terms and implicit in the integral term, it converges to a viscosity solution.
5. D'Halluin et al. (2004, 2005)- implicit methods for evaluating vanilla European options, barrier options, and American options

Existing approaches to solve PIDE

1. Amim (1993) - explicit multinomial tree
2. Tavella and Randall (2000) - Picard iterations
3. Andersen and Andreasen (2000) - forward equation, European call, second order accurate unconditionally stable operator splitting (ADI), FFT
4. Cont and Voltchkova (2003) - discretization that is implicit in the differential terms and implicit in the integral term, it converges to a viscosity solution.
5. D'Halluin et al. (2004, 2005)- implicit methods for evaluating vanilla European options, barrier options, and American options
6. D'Halluin et al. (2005) - semi-Lagrangian approach for pricing American Asian options

Existing approaches to solve PIDE

1. Amim (1993) - explicit multinomial tree
2. Tavella and Randall (2000) - Picard iterations
3. Andersen and Andreasen (2000) - forward equation, European call, second order accurate unconditionally stable operator splitting (ADI), FFT
4. Cont and Voltchkova (2003) - discretization that is implicit in the differential terms and implicit in the integral term, it converges to a viscosity solution.
5. D'Halluin et al. (2004, 2005)- implicit methods for evaluating vanilla European options, barrier options, and American options
6. D'Halluin et al. (2005) - semi-Lagrangian approach for pricing American Asian options

Existing approaches - cont.

1. Because of the integrals in the equations the methods have proven to be relatively expensive. Quadrature methods are expensive since the integrals must be evaluated at every point of the mesh. Though less so, Fourier methods are also computationally intensive since in order to avoid wrap around effects they require enlargement of the computational domain. They are also slow to converge when the parameters of the jump process are not smooth, and for efficiency require uniform meshes. And low order of accuracy.
2. Carr and Mayo (2007) proposed a different and more efficient class of methods which are based on the fact that the integrals often satisfy differential equations. Completed only for Merton and Kou models.

Existing approaches - cont.

1. Because of the integrals in the equations the methods have proven to be relatively expensive. Quadrature methods are expensive since the integrals must be evaluated at every point of the mesh. Though less so, Fourier methods are also computationally intensive since in order to avoid wrap around effects they require enlargement of the computational domain. They are also slow to converge when the parameters of the jump process are not smooth, and for efficiency require uniform meshes. And low order of accuracy.
2. Carr and Mayo (2007) proposed a different and more efficient class of methods which are based on the fact that the integrals often satisfy differential equations. Completed only for Merton and Kou models.

Existing approaches - cont.

Our approach:

1. 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 this operator then the original PIDE reduces to a new type of equation - so-called pseudo-parabolic equation.
2. 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. Fractional PDEs for the Lévy processes with finite variation were derived by Boyarchenko and Levendorsky (2002) and later by Cartea (2007) using a characteristic function technique. Here we derive them in all cases including processes with infinite variation using a different technique - shift operators. Then to solve them we use a shifted Grunwald-Letnikov approximation scheme proven to be unconditionally stable. First and second order of approximation in space and time are considered.

Existing approaches - cont.

Our approach:

1. 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 this operator then the original PIDE reduces to a new type of equation - so-called pseudo-parabolic equation.
2. 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. Fractional PDEs for the Lévy processes with finite variation were derived by Boyarchenko and Levendorsky (2002) and later by Cartea (2007) using a characteristic function technique. Here we derive them in all cases including processes with infinite variation using a different technique - shift operators. Then to solve them we use a shifted Grunwald-Letnikov approximation scheme proven to be unconditionally stable. First and second order of approximation in space and time are considered.

GTSP/KoBoL/SSM models.

1. Stochastic skew model (SSM) has been proposed by Carr and Wu for pricing currency options. It makes use of a Lévy model also known as generalized tempered stable processes (GTSP) for the dynamics of stock prices which generalize the CGMY processes proposed by Carr, Geman, Madan and Yor. A similar model was independently proposed by Koponen and then Boyarchenko and Levendorsky. The processes are obtained by specifying a more generalized Lévy measure with two additional parameters. These two parameters provide control on asymmetry of small jumps and different frequencies for upward and downward jumps. Results of Zhou, Hagan and Schleiniger show that this generalization allows for more accurate pricing of options.
2. Generalized Tempered Stable Processes (GTSP) have probability densities symmetric in a neighborhood of the origin and exponentially decaying in the far tails. After this exponential softening, the small jumps keep their initial stable-like behavior, whereas the large jumps become exponentially tempered. The Lévy measure of GTSP reads

$$\mu(y) = \lambda_- \frac{e^{-\nu_- |y|}}{|y|^{1+\alpha_-}} \mathbb{1}_{y<0} + \lambda_+ \frac{e^{-\nu_+ |y|}}{|y|^{1+\alpha_+}} \mathbb{1}_{y>0}, \qquad (1)$$

where $\nu_\pm > 0$, $\lambda_\pm > 0$ and $\alpha_\pm < 2$. The last condition is necessary to provide

$$\int_{-1}^{1} y^2 \mu(dy) < \infty, \quad \int_{|y|>1} \mu(dy) < \infty. \qquad (2)$$

GTSP/KoBoL/SSM models.

1. Stochastic skew model (SSM) has been proposed by Carr and Wu for pricing currency options. It makes use of a Lévy model also known as generalized tempered stable processes (GTSP) for the dynamics of stock prices which generalize the CGMY processes proposed by Carr, Geman, Madan and Yor. A similar model was independently proposed by Koponen and then Boyarchenko and Levendorsky. The processes are obtained by specifying a more generalized Lévy measure with two additional parameters. These two parameters provide control on asymmetry of small jumps and different frequencies for upward and downward jumps. Results of Zhou, Hagan and Schleiniger show that this generalization allows for more accurate pricing of options.

2. Generalized Tempered Stable Processes (GTSP) have probability densities symmetric in a neighborhood of the origin and exponentially decaying in the far tails. After this exponential softening, the small jumps keep their initial stable-like behavior, whereas the large jumps become exponentially tempered. The Lévy measure of GTSP reads

$$\mu(y) = \lambda_- \frac{e^{-\nu_- |y|}}{|y|^{1+\alpha_-}} \mathbb{1}_{y<0} + \lambda_+ \frac{e^{-\nu_+ |y|}}{|y|^{1+\alpha_+}} \mathbb{1}_{y>0}, \tag{1}$$

   where $\nu_\pm > 0$, $\lambda_\pm > 0$ and $\alpha_\pm < 2$. The last condition is necessary to provide

$$\int_{-1}^{1} y^2 \mu(dy) < \infty, \ \int_{|y|>1} \mu(dy) < \infty. \tag{2}$$

3. The case $\lambda_+ = \lambda_-, \alpha_+ = \alpha_-$ corresponds to the CGMY process. The limiting case $\alpha_+ = \alpha_- = 0, \lambda_+ = \lambda_-$ is the special case of the Variance Gamma process. As Hagan at al mentioned, six parameters of the model play an important role in capturing various aspects of the stochastic process. The parameters $\lambda_\pm$ determine the overall and relative frequencies of upward and downward jumps. If we are interested only in jumps larger than a given value, these two parameters tell us how often we should expect such events. $\nu_\pm$ control the tail behavior of the Lévy measure, and they tell us how far the process may jump. They also lead to skewed distributions when they are unequal. In the special case when they are equal, the Lévy measure is symmetric. Finally, $\alpha_\pm$ are particularly useful for the local behavior of the process. They determine whether the process has finite or infinite activity, or variation.

GTSP/KoBoL/SSM models.

1. Stochastic skew model (SSM) has been proposed by Carr and Wu for pricing currency options. It makes use of a Lévy model also known as generalized tempered stable processes (GTSP) for the dynamics of stock prices which generalize the CGMY processes proposed by Carr, Geman, Madan and Yor. A similar model was independently proposed by Koponen and then Boyarchenko and Levendorsky. The processes are obtained by specifying a more generalized Lévy measure with two additional parameters. These two parameters provide control on asymmetry of small jumps and different frequencies for upward and downward jumps. Results of Zhou, Hagan and Schleiniger show that this generalization allows for more accurate pricing of options.

2. Generalized Tempered Stable Processes (GTSP) have probability densities symmetric in a neighborhood of the origin and exponentially decaying in the far tails. After this exponential softening, the small jumps keep their initial stable-like behavior, whereas the large jumps become exponentially tempered. The Lévy measure of GTSP reads

$$\mu(y) = \lambda_- \frac{e^{-\nu_- |y|}}{|y|^{1+\alpha_-}} \mathbb{1}_{y<0} + \lambda_+ \frac{e^{-\nu_+ |y|}}{|y|^{1+\alpha_+}} \mathbb{1}_{y>0}, \tag{1}$$

where $\nu_\pm > 0$, $\lambda_\pm > 0$ and $\alpha_\pm < 2$. The last condition is necessary to provide

$$\int_{-1}^{1} y^2 \mu(dy) < \infty, \ \int_{|y|>1} \mu(dy) < \infty. \tag{2}$$

3. The case $\lambda_+ = \lambda_-, \alpha_+ = \alpha_-$ corresponds to the CGMY process. The limiting case $\alpha_+ = \alpha_- = 0, \lambda_+ = \lambda_-$ is the special case of the Variance Gamma process. As Hagan at al mentioned, six parameters of the model play an important role in capturing various aspects of the stochastic process. The parameters $\lambda_\pm$ determine the overall and relative frequencies of upward and downward jumps. If we are interested only in jumps larger than a given value, these two parameters tell us how often we should expect such events. $\nu_\pm$ control the tail behavior of the Lévy measure, and they tell us how far the process may jump. They also lead to skewed distributions when they are unequal. In the special case when they are equal, the Lévy measure is symmetric. Finally, $\alpha_\pm$ are particularly useful for the local behavior of the process. They determine whether the process has finite or infinite activity, or variation.

Original model - SSM

Using this model of jumps Carr and Wu (2004) derived the following PIDE which governs an arbitrage-free value of a European call option at time t

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

on the domain $S > 0$, $V_R > 0$, $V_L > 0$ and $t \in [0, T]$, where S, V_R, V_L are state variables (spot price and stochastic variances). For the following we make some critical assumptions.

1. This PIDE could be generalized with allowance for GTSP processes, which means we substitute α in Eq. (3) with α_R, α_L, and λ with λ_R, λ_L correspondingly.
2. The obtained PIDE could be solved by using a splitting technique similar to that proposed in Itkin, Carr (2006).

Original model - SSM

Using this model of jumps Carr and Wu (2004) derived the following PIDE which governs an arbitrage-free value of a European call option at time t

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

on the domain $S > 0$, $V_R > 0$, $V_L > 0$ and $t \in [0, T]$, where S, V_R, V_L are state variables (spot price and stochastic variances). For the following we make some critical assumptions.

1. This PIDE could be generalized with allowance for GTSP processes, which means we substitute α in Eq. (3) with α_R, α_L, and λ with λ_R, λ_L correspondingly.
2. The obtained PIDE could be solved by using a splitting technique similar to that proposed in Itkin, Carr (2006).
3. We assume $\alpha_R < 0$, $\alpha_L < 0$ which means we consider only jumps with finite activity. Therefore, each compensator under the integral could be integrated out.

Original model - SSM

Using this model of jumps Carr and Wu (2004) derived the following PIDE which governs an arbitrage-free value of a European call option at time t

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

on the domain $S > 0$, $V_R > 0$, $V_L > 0$ and $t \in [0, T]$, where S, V_R, V_L are state variables (spot price and stochastic variances). For the following we make some critical assumptions.

1. This PIDE could be generalized with allowance for GTSP processes, which means we substitute α in Eq. (3) with α_R, α_L, and λ with λ_R, λ_L correspondingly.
2. The obtained PIDE could be solved by using a splitting technique similar to that proposed in Itkin, Carr (2006).
3. We assume $\alpha_R < 0$, $\alpha_L < 0$ which means we consider only jumps with finite activity. Therefore, each compensator under the integral could be integrated out.

Original model - PIDE

As a result we consider just that steps of splitting which deals with the remaining integral term. The corresponding equation reads

1. $$\frac{\partial}{\partial t} C(S, V_R, V_L, t) = -\sqrt{V_R} \int_0^\infty C(Se^y, V_R, V_L, t) \lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} dy \qquad (4)$$

   for positive jumps and

2. $$\frac{\partial}{\partial t} C(S, V_R, V_L, t) = -\sqrt{V_L} \int_{-\infty}^0 C(Se^y, V_R, V_L, t) \lambda_L \frac{e^{-\nu_L |y|}}{|y|^{1+\alpha_L}} dy \qquad (5)$$

   for negative jumps.

Original model - PIDE

As a result we consider just that steps of splitting which deals with the remaining integral term. The corresponding equation reads

1. $$\frac{\partial}{\partial t} C(S, V_R, V_L, t) = -\sqrt{V_R} \int_0^\infty C(Se^y, V_R, V_L, t)\lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} dy \tag{4}$$

   for positive jumps and

2. $$\frac{\partial}{\partial t} C(S, V_R, V_L, t) = -\sqrt{V_L} \int_{-\infty}^0 C(Se^y, V_R, V_L, t)\lambda_L \frac{e^{-\nu_L |y|}}{|y|^{1+\alpha_L}} dy \tag{5}$$

   for negative jumps.

3. Now an important note is that in accordance with the definition of these integrals we can rewrite the kernel as

$$\begin{aligned} \frac{\partial}{\partial t} C(x, t) &= -\sqrt{V_R} \int_0^\infty C(x+y, t)\lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} \mathbb{1}_{y>0} dy \\ \frac{\partial}{\partial t} C(S, t) &= -\sqrt{V_L} \int_{-\infty}^0 C(x+y, t)\lambda_L \frac{e^{-\nu_L |y|}}{|y|^{1+\alpha_L}} \mathbb{1}_{y<0} dy \end{aligned} \tag{6}$$

   This two equations are still PIDE or evolutionary integral equations. We want to apply our new method to transform them to a certain pseudo parabolic equations at $\alpha \in \mathbb{I}$.

Original model - PIDE

As a result we consider just that steps of splitting which deals with the remaining integral term. The corresponding equation reads

1. $$\frac{\partial}{\partial t} C(S, V_R, V_L, t) = -\sqrt{V_R} \int_0^{\infty} C(Se^y, V_R, V_L, t) \lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} dy \tag{4}$$

   for positive jumps and

2. $$\frac{\partial}{\partial t} C(S, V_R, V_L, t) = -\sqrt{V_L} \int_{-\infty}^{0} C(Se^y, V_R, V_L, t) \lambda_L \frac{e^{-\nu_L |y|}}{|y|^{1+\alpha_L}} dy \tag{5}$$

   for negative jumps.

3. Now an important note is that in accordance with the definition of these integrals we can rewrite the kernel as

$$\begin{aligned} \frac{\partial}{\partial t} C(x, t) &= -\sqrt{V_R} \int_0^{\infty} C(x+y, t) \lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} \mathbb{1}_{y>0} dy \\ \frac{\partial}{\partial t} C(S, t) &= -\sqrt{V_L} \int_{-\infty}^{0} C(x+y, t) \lambda_L \frac{e^{-\nu_L |y|}}{|y|^{1+\alpha_L}} \mathbb{1}_{y<0} dy \end{aligned} \tag{6}$$

   This two equations are still PIDE or evolutionary integral equations. We want to apply our new method to transform them to a certain pseudo parabolic equations at $\alpha \in \mathbb{I}$.

Transformation - $\alpha \in \mathbb{I}$

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. (6), i.e.

$$\mathcal{A}_y^+ \left[\lambda \frac{e^{-\nu |y|}}{|y|^{1+\alpha}} \mathbb{1}_{y>0} \right] = \delta(y) \tag{7}$$

We prove the following proposition.

Proposition

Assume that in the Eq. (7) $\alpha \in \mathbb{I}$, and $\alpha < 0$. Then the solution of the Eq. (7) with respect to $\mathcal{A}_y^+$ is

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

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

Transformation - $\alpha \in \mathbb{I}$ - cont.

For the second equation in the Eq. (6) it is possible to elaborate an analogous approach. Again assuming $z = x + y$ we rewrite it in the form

$$\frac{\partial}{\partial t} C(x,t) = -\sqrt{V_L} \int_{-\infty}^{x} C(z,t) \lambda_R \frac{e^{-\nu_R |z-x|}}{|z-x|^{1+\alpha_R}} \mathbb{1}_{z-x<0} dz \tag{8}$$

Now we need to find a differential operator $\mathcal{A}_y^-$ which Green's function is the kernel of the integral in the Eq. (8), i.e.

$$\mathcal{A}_y^- \left[\lambda \frac{e^{-\nu |y|}}{|y|^{1+\alpha}} \mathbb{1}_{y<0} \right] = \delta(y) \tag{9}$$

We prove the following proposition.

Proposition

Assume that in the Eq. (9) $\alpha \in \mathbb{I}$, and $\alpha < 0$. Then the solution of the Eq. (9) with respect to $\mathcal{A}_y^-$ is

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

Transformation - $\alpha \in \mathbb{I}$ - cont.

To proceed we need to prove two other statements.

Proposition

Let us denote the kernels as

$$g^{+}(z-x) \equiv \lambda_R \frac{e^{-\nu_R |z-x|}}{|z-x|^{1+\alpha_R}} \mathbb{1}_{z-x>0}. \tag{10}$$

Then

$$\mathcal{A}_x^{-} g^{+}(z-x) = \delta(z-x). \tag{11}$$

Proposition

Let us denote the kernels as

$$g^{-}(z-x) \equiv \lambda_L \frac{e^{-\nu_L |z-x|}}{|z-x|^{1+\alpha_L}} \mathbb{1}_{z-x<0}. \tag{12}$$

Then

$$\mathcal{A}_x^{+} g^{-}(z-x) = \delta(z-x). \tag{13}$$

Transformation - $\alpha \in \mathbb{I}$ - cont.

1. We now apply the operator $\mathcal{A}_x^-$ to both parts of the Eq. (6) to obtain

$$\mathcal{A}_x^- \frac{\partial}{\partial t} C(x,t) = -\sqrt{V_R}\mathcal{A}_x^- \int_x^\infty C(z,t) g^+(z-x) dz = \sqrt{V_R}\left\{\int_x^\infty C(z,t)\mathcal{A}_x^- g^+(z-x)dz + \mathcal{R}\right\} \tag{14}$$
$$= \sqrt{V_R}\left\{\int_x^\infty C(z,t)\delta(z-x)dz + \mathcal{R}\right\} = \frac{1}{2}\sqrt{V_R}C(x,t) + \sqrt{V_R}\mathcal{R}$$

Here

$$\mathcal{R} = \sum_{i=0}^p a_i \left(\frac{\partial^{p-i}}{\partial x^{p-i}} V(x)\right)\left(\frac{\partial^i}{\partial x^i} g(z-x)\right)\Big|_{z-x=0}, \tag{15}$$

and a_i are some constant coefficients. As from the definition in the Eq. (10) $g(z-x) \propto (z-x)^p$, the only term in the Eq. (15) which does not vanish is that at $i = p$. Thus

$$\mathcal{R} = V(x)\left(\frac{\partial^p}{\partial x^p} g(z-x)\right)\Big|_{z-x=0} = V(x)p!\mathbb{1}_{(0)} = 0; \tag{16}$$

2. With allowance for this expression from the Eq. (14) we obtain the following pseudo parabolic equation for $C(x,t)$

$$\boxed{\mathcal{A}_x^- \frac{\partial}{\partial t} C(x,t) = -\frac{1}{2}\sqrt{V_R}C(x,t)} \tag{17}$$

Transformation - $\alpha \in \mathbb{I}$ - cont.

1. We now apply the operator $\mathcal{A}_x^-$ to both parts of the Eq. (6) to obtain

$$\mathcal{A}_x^- \frac{\partial}{\partial t} C(x,t) = -\sqrt{V_R}\mathcal{A}_x^- \int_x^\infty C(z,t) g^+(z-x)dz = \sqrt{V_R}\left\{\int_x^\infty C(z,t)\mathcal{A}_x^- g^+(z-x)dz + \mathcal{R}\right\} \quad (14)$$
$$= \sqrt{V_R}\left\{\int_x^\infty C(z,t)\delta(z-x)dz + \mathcal{R}\right\} = \frac{1}{2}\sqrt{V_R}C(x,t) + \sqrt{V_R}\mathcal{R}$$

Here

$$\mathcal{R} = \sum_{i=0}^{p} a_i \left(\frac{\partial^{p-i}}{\partial x^{p-i}} V(x)\right)\left(\frac{\partial^i}{\partial x^i} g(z-x)\right)\Big|_{z-x=0}, \quad (15)$$

and a_i are some constant coefficients. As from the definition in the Eq. (10) $g(z-x) \propto (z-x)^p$, the only term in the Eq. (15) which does not vanish is that at $i = p$. Thus

$$\mathcal{R} = V(x)\left(\frac{\partial^p}{\partial x^p} g(z-x)\right)\Big|_{z-x=0} = V(x)p!\mathbb{1}_{(0)} = 0; \quad (16)$$

2. With allowance for this expression from the Eq. (14) we obtain the following pseudo parabolic equation for $C(x,t)$

$$\boxed{\mathcal{A}_x^- \frac{\partial}{\partial t} C(x,t) = -\frac{1}{2}\sqrt{V_R}C(x,t)} \quad (17)$$

3. Applying the operator $\mathcal{A}_x^+$ to both parts of the second equation in the Eq. (8) and doing in the same way as in the previous paragraph we obtain the following pseudo parabolic equation for $C(x,t)$

$$\boxed{\mathcal{A}_x^+ \frac{\partial}{\partial t} C(x,t) = -\frac{1}{2}\sqrt{V_L}C(x,t)} \quad (18)$$

Transformation - $\alpha \in \mathbb{I}$ - cont.

1. We now apply the operator $\mathcal{A}_x^-$ to both parts of the Eq. (6) to obtain

$$\mathcal{A}_x^- \frac{\partial}{\partial t} C(x,t) = -\sqrt{V_R}\mathcal{A}_x^- \int_x^\infty C(z,t) g^+(z-x)dz = \sqrt{V_R}\left\{\int_x^\infty C(z,t)\mathcal{A}_x^- g^+(z-x)dz + \mathcal{R}\right\} \quad (14)$$
$$= \sqrt{V_R}\left\{\int_x^\infty C(z,t)\delta(z-x)dz + \mathcal{R}\right\} = \frac{1}{2}\sqrt{V_R}C(x,t) + \sqrt{V_R}\mathcal{R}$$

Here

$$\mathcal{R} = \sum_{i=0}^{p} a_i \left(\frac{\partial^{p-i}}{\partial x^{p-i}} V(x)\right)\left(\frac{\partial^i}{\partial x^i} g(z-x)\right)\Big|_{z-x=0}, \quad (15)$$

and a_i are some constant coefficients. As from the definition in the Eq. (10) $g(z-x) \propto (z-x)^p$, the only term in the Eq. (15) which does not vanish is that at $i=p$. Thus

$$\mathcal{R} = V(x)\left(\frac{\partial^p}{\partial x^p} g(z-x)\right)\Big|_{z-x=0} = V(x)p!\mathbb{1}_{(0)} = 0; \quad (16)$$

2. With allowance for this expression from the Eq. (14) we obtain the following pseudo parabolic equation for $C(x,t)$

$$\boxed{\mathcal{A}_x^- \frac{\partial}{\partial t} C(x,t) = -\frac{1}{2}\sqrt{V_R}C(x,t)} \quad (17)$$

3. Applying the operator $\mathcal{A}_x^+$ to both parts of the second equation in the Eq. (8) and doing in the same way as in the previous paragraph we obtain the following pseudo parabolic equation for $C(x,t)$

$$\boxed{\mathcal{A}_x^+ \frac{\partial}{\partial t} C(x,t) = -\frac{1}{2}\sqrt{V_L}C(x,t)} \quad (18)$$

Solution of PPE

Exact solution:

Assume that the inverse operator $\mathcal{A}^{-1}$ exists we can represent, for instance, the Eq. (17) in the form

$$\frac{\partial}{\partial t} C(x,t) = -\mathcal{B} C(x,t), \quad \mathcal{B} \equiv \frac{1}{2}\sqrt{V_R}(\mathcal{A}_x^-)^{-1}, \tag{19}$$

This equation can be formally solved analytically to give

$$C(x,t) = e^{\mathcal{B}(T-t)} C(x,T), \tag{20}$$

where T is the time to maturity and $C(x,T)$ is payoff. Switching to a new variable $\tau = T - t$ to go backward in time we rewrite the Eq. (20) as

$$C(x,\tau) = e^{\mathcal{B}\tau} C(x,0), \tag{21}$$

Numerical solution:

1. 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. (20) in the form
$$C^{k+1}(x) = e^{\mathcal{B}\theta} C^k(x), \tag{22}$$
where $C^k(x) \equiv C(x, k\theta)$. To get representation of the rhs of the Eq. (22) with given order of approximation in θ, we can substitute the whole exponential operator with its Padé approximation of the corresponding order m.
2. First, consider the case $m = 1$. A symmetric Padé approximation of the order $(1,1)$ for the exponential operator is
$$e^{\mathcal{B}\theta} = \frac{1 + \mathcal{B}\theta/2}{1 - \mathcal{B}\theta/2} \tag{23}$$

Solution of PPE

Exact solution:

Assume that the inverse operator $\mathcal{A}^{-1}$ exists we can represent, for instance, the Eq. (17) in the form

$$\frac{\partial}{\partial t} C(x,t) = -\mathcal{B} C(x,t), \quad \mathcal{B} \equiv \frac{1}{2}\sqrt{V_R}(\mathcal{A}_x^-)^{-1}, \tag{19}$$

This equation can be formally solved analytically to give

$$C(x,t) = e^{\mathcal{B}(T-t)} C(x,T), \tag{20}$$

where T is the time to maturity and $C(x,T)$ is payoff. Switching to a new variable $\tau = T - t$ to go backward in time we rewrite the Eq. (20) as

$$C(x,\tau) = e^{\mathcal{B}\tau} C(x,0), \tag{21}$$

Numerical solution:

1. 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. (20) in the form

$$C^{k+1}(x) = e^{\mathcal{B}\theta} C^k(x), \tag{22}$$

   where $C^k(x) \equiv C(x, k\theta)$. To get representation of the rhs of the Eq. (22) with given order of approximation in θ, we can substitute the whole exponential operator with its Padé approximation of the corresponding order m.
2. First, consider the case $m = 1$. A symmetric Padé approximation of the order $(1,1)$ for the exponential operator is

$$e^{\mathcal{B}\theta} = \frac{1 + \mathcal{B}\theta/2}{1 - \mathcal{B}\theta/2} \tag{23}$$

Numerical solution - cont.

1. Substituting this into the Eq. (22) and affecting both parts of the equation by the operator $1 - \mathcal{B}\theta/2$ gives

$$\left(1 - \frac{1}{2}\mathcal{B}\theta\right) C^{k+1}(x) = \left(1 + \frac{1}{2}\mathcal{B}\theta\right) C^{k}(x). \tag{24}$$

   This is a discrete equation which approximates the original solution given in the Eq. (22) with the second order in θ. One can easily recognize in this scheme a 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. (24). The resulting equation is a pure differential equation and reads

$$\left(\mathcal{A}_x^- - \frac{\sqrt{V_R}}{4}\theta\right) C^{k+1}(x) = \left(\mathcal{A}_x^- + \frac{\sqrt{V_R}}{4}\theta\right) C^{k}(x). \tag{25}$$

2. 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. (25) 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 $l = d + q$ diagonals, where it appears that $d + q$ is necessarily an odd number. 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 band matrix the solution of the corresponding system of linear equations in the Eq. (25) could be efficiently obtained using a modern technique (for instance, using a ScaLAPACK package). 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.

Numerical solution - cont.

1. Substituting this into the Eq. (22) and affecting both parts of the equation by the operator $1 - \mathcal{B}\theta/2$ gives

$$\left(1 - \frac{1}{2}\mathcal{B}\theta\right) C^{k+1}(x) = \left(1 + \frac{1}{2}\mathcal{B}\theta\right) C^{k}(x). \tag{24}$$

   This is a discrete equation which approximates the original solution given in the Eq. (22) with the second order in θ. One can easily recognize in this scheme a 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. (24). The resulting equation is a pure differential equation and reads

$$\left(\mathcal{A}_x^- - \frac{\sqrt{V_R}}{4}\theta\right) C^{k+1}(x) = \left(\mathcal{A}_x^- + \frac{\sqrt{V_R}}{4}\theta\right) C^{k}(x). \tag{25}$$

2. 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. (25) 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 $l = d + q$ diagonals, where it appears that $d + q$ is necessarily an odd number. 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 band matrix the solution of the corresponding system of linear equations in the Eq. (25) could be efficiently obtained using a modern technique (for instance, using a ScaLAPACK package). 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.

Numerical solution - cont.

1. Example: 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 $(34/9)2N\log_2(2N) \approx 20900$ (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.), one can see that our algorithm is of the same speed as the FFT.
2. 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

$$e^{\mathcal{B}\theta} = \frac{1 + \mathcal{B}\theta/3}{1 - 2\mathcal{B}\theta/3 + \mathcal{B}^2\theta^2/6}, \tag{26}$$

and the corresponding finite difference scheme for the solution of the Eq. (22) is

$$\left[(\mathcal{A}_x^-)^2 - \frac{1}{3}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{24}V_R\theta^2\right] C^{k+1}(x) = \mathcal{A}_x^- \left[\mathcal{A}_x^- + \frac{1}{6}\sqrt{V_R}\theta\right] C^k(x). \tag{27}$$

which is of the third order in θ.

Numerical solution - cont.

1. Example: 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 $(34/9)2N\log_2(2N) \approx 20900$ (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.), one can see that our algorithm is of the same speed as the FFT.
2. 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

$$e^{\mathcal{B}\theta} = \frac{1 + \mathcal{B}\theta/3}{1 - 2\mathcal{B}\theta/3 + \mathcal{B}^2\theta^2/6}, \tag{26}$$

and the corresponding finite difference scheme for the solution of the Eq. (22) is

$$\left[(\mathcal{A}_x^-)^2 - \frac{1}{3}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{24}V_R\theta^2\right] C^{k+1}(x) = \mathcal{A}_x^- \left[\mathcal{A}_x^- + \frac{1}{6}\sqrt{V_R}\theta\right] C^k(x). \tag{27}$$

which is of the third order in θ.

3. The (2,2) Padé approximation is

$$e^{\mathcal{B}\theta} = \frac{1 + \mathcal{B}\theta/2 + \mathcal{B}^2\theta^2/12}{1 - \mathcal{B}\theta/2 + \mathcal{B}^2\theta^2/12}, \tag{28}$$

and the corresponding finite difference scheme for the solution of the Eq. (22) is

$$\left[(\mathcal{A}_x^-)^2 - \frac{1}{4}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{48}V_R\theta^2\right] C^{k+1}(x) = \left[(\mathcal{A}_x^-)^2 + \frac{1}{4}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{48}V_R\theta^2\right] C^k(x), \tag{29}$$

which is of the fourth order in θ.

Numerical solution - cont.

1. Example: 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 $(34/9)2N\log_2(2N) \approx 20900$ (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.), one can see that our algorithm is of the same speed as the FFT.
2. 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

$$e^{\mathcal{B}\theta} = \frac{1 + \mathcal{B}\theta/3}{1 - 2\mathcal{B}\theta/3 + \mathcal{B}^2\theta^2/6}, \tag{26}$$

and the corresponding finite difference scheme for the solution of the Eq. (22) is

$$\left[(\mathcal{A}_x^-)^2 - \frac{1}{3}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{24}V_R\theta^2\right]C^{k+1}(x) = \mathcal{A}_x^-\left[\mathcal{A}_x^- + \frac{1}{6}\sqrt{V_R}\theta\right]C^k(x). \tag{27}$$

which is of the third order in θ.

3. The (2,2) Padé approximation is

$$e^{\mathcal{B}\theta} = \frac{1 + \mathcal{B}\theta/2 + \mathcal{B}^2\theta^2/12}{1 - \mathcal{B}\theta/2 + \mathcal{B}^2\theta^2/12}, \tag{28}$$

and the corresponding finite difference scheme for the solution of the Eq. (22) is

$$\left[(\mathcal{A}_x^-)^2 - \frac{1}{4}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{48}V_R\theta^2\right]C^{k+1}(x) = \left[(\mathcal{A}_x^-)^2 + \frac{1}{4}\sqrt{V_R}\theta\mathcal{A}_x^- + \frac{1}{48}V_R\theta^2\right]C^k(x), \tag{29}$$

which is of the fourth order in θ.

Stability analysis

1. Stability analysis of the derived finite difference schemes could be provided using a standard von-Neumann method. Suppose that operator $\mathcal{A}_x^-$ has eigenvalues λ which belong to continuous spectrum. Any finite difference approximation of the operator $\mathcal{A}_x^-$ - $FD(\mathcal{A}_x^-)$ - transforms this continuous spectrum into some discrete spectrum, so we denote the eigenvalues of the discrete operator $FD(\mathcal{A}_x^-)$ as λ_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. (25). It is stable if in some norm $\| \cdot \|$

   $$\left\| \left(\mathcal{A}_x^- - \frac{\sqrt{V_R}}{4}\theta \right)^{-1} \left(\mathcal{A}_x^- + \frac{\sqrt{V_R}}{4}\theta \right) \right\| < 1. \tag{30}$$

   It is easy to see that this inequality obeys when all eigenvalues of the operator $\mathcal{A}_x^-$ are negative. However, based on the definition of this operator given in the Proposition 2, 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 $FD(\mathcal{A}_x^-)$.

2. **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 $\left(\nu_R - \frac{\partial}{\partial x}\right)^{\alpha_R}$. Define $h = (x_{max} - x_{min})/N$ to be the grid step in the x-direction. Also define $c_i^k = C^k(x_i)$. To make our method to be of the second order in x we use the following numerical approximation

   $$\frac{\partial C^k(x)}{\partial x} = \frac{-C_{i+2}^k + 4C_{i+1}^k - 3C_i^k}{2h} + O(h^2) \tag{31}$$

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

Stability analysis

1. Stability analysis of the derived finite difference schemes could be provided using a standard von-Neumann method. Suppose that operator $\mathcal{A}_x^-$ has eigenvalues λ which belong to continuous spectrum. Any finite difference approximation of the operator $\mathcal{A}_x^-$ - $FD(\mathcal{A}_x^-)$ - transforms this continuous spectrum into some discrete spectrum, so we denote the eigenvalues of the discrete operator $FD(\mathcal{A}_x^-)$ as λ_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. (25). It is stable if in some norm $\| \cdot \|$

$$\left\| \left(\mathcal{A}_x^- - \frac{\sqrt{V_R}}{4}\theta \right)^{-1} \left(\mathcal{A}_x^- + \frac{\sqrt{V_R}}{4}\theta \right) \right\| < 1. \tag{30}$$

It is easy to see that this inequality obeys when all eigenvalues of the operator $\mathcal{A}_x^-$ are negative. However, based on the definition of this operator given in the Proposition 2, 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 $FD(\mathcal{A}_x^-)$.

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

$$\frac{\partial C^k(x)}{\partial x} = \frac{-C_{i+2}^k + 4C_{i+1}^k - 3C_i^k}{2h} + O(h^2) \tag{31}$$

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

Stability analysis - cont.

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 matrix $\mathbb{B}$ are

$$\lambda_{\mathbb{B}} = \sqrt{V_R}\lambda_R \Gamma(-\alpha_R)\left\{[\nu_R + 3/(2h)]^{\alpha_R} - \nu_R^{\alpha_R}\right\}. \tag{32}$$

As $\alpha_R < 0$ and $\nu_R > 0$ it follows that $\lambda_{\mathbb{B}} < 0$. Taking into account that $\lambda_{\mathbb{B}} < 0$ we arrive at the following result

$$\left\|\left(1 - \frac{1}{2}\mathcal{B}\theta\right)^{-1}\left(1 + \frac{1}{2}\mathcal{B}\theta\right)\right\| < 1. \tag{33}$$

We also obey the condition $\mathbb{R}\left(\nu_R - \frac{\partial}{\partial x}\right) > 0$. Thus, our numerical method is unconditionally stable.

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

$$\frac{\partial C^k(x)}{\partial x} = \frac{3C_i^k - 4C_{i-1}^k + C_{i-2}^k}{2h} + O(h^2) \tag{34}$$

Same proof that the resulting FD scheme is unconditionally stable and of the second order of approximation.

Stability analysis - cont.

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 matrix $\mathbb{B}$ are

$$\lambda_{\mathbb{B}} = \sqrt{V_R}\lambda_R\Gamma(-\alpha_R)\left\{[\nu_R + 3/(2h)]^{\alpha_R} - \nu_R^{\alpha_R}\right\}. \tag{32}$$

As $\alpha_R < 0$ and $\nu_R > 0$ it follows that $\lambda_{\mathbb{B}} < 0$. Taking into account that $\lambda_{\mathbb{B}} < 0$ we arrive at the following result

$$\left\| \left(1 - \frac{1}{2}\mathcal{B}\theta\right)^{-1}\left(1 + \frac{1}{2}\mathcal{B}\theta\right)\right\| < 1. \tag{33}$$

We also obey the condition $\mathbb{R}\left(\nu_R - \frac{\partial}{\partial x}\right) > 0$. Thus, our numerical method is unconditionally stable.

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

$$\frac{\partial C^k(x)}{\partial x} = \frac{3C_i^k - 4C_{i-1}^k + C_{i-2}^k}{2h} + O(h^2) \tag{34}$$

   Same proof that the resulting FD scheme is unconditionally stable and of the second order of approximation.

Comparison with FFT

To apply an FFT approach we first select a domain in x space where the values of function $C(x, \tau$ are of our interest. Suppose this is $x \in (-x_*, x_*)$. We define a uniform grid in this domain which contains N points: $x_1 = -x_*, x_2, \dots x_{N-1}, x_N = x_*$ such that $x_i - x_{i-1} = h$, $i = 2 \dots N$. We then approximate the integral in the rhs with the first order of accuracy in h as

$$\int_0^\infty C(x+y,\tau)\lambda_R \frac{e^{-\nu_R|y|}}{|y|^{1+\alpha_R}} dy = h \sum_{j=1-i}^{N-i} C_{i+j}(\tau) f_j, \quad f_j \equiv \lambda_R \frac{e^{-\nu_R|x_j|}}{|x_j|^{1+\alpha_R}} + O(h^2). \tag{35}$$

This approximation means that we have to extend our computational domain to the left up to $x_{1-N} = x_1 - hN$.
The matrix $|f|$ is a Toeplitz matrix. Using FFT directly to compute a matrix-vector product in the Eq. (35) will produce a wrap-round error that significantly lowers the accuracy. Therefore a standard technique is to embed this Toeplitz matrix into a circulant matrix $\mathcal{F}$ which is defined as follows. The first row of F is

$$F_1 = (f_0, f_1, \dots, f_{N-1}, 0, f_{1-N}, \dots, f_{-1}),$$

and others are generated by permutation (see, for instance Zhang & Wang 2009). We also define a vector

$$\hat{C} = [C_1(\tau), \dots C_N(\tau), \underbrace{0, \dots, 0}_{N}]^T.$$

Then the matrix-vector product in the rhs Eq. (35) is given by the first N rows in the vector $V = \mathrm{ifft}(\mathrm{fft}(F_1) * \mathrm{fft}(\hat{C}))$, where fft and ifft are the forward and inverse discrete Fourier transforms as they are defined, say in Matlab. In practice, an error at edge points close to x_1 and x_N is higher, therefore it is useful first to add some points left to x_1 and right to x_N and then apply the above described algorithm to compute the integral. We investigated some test problems, for instance, where the function C was chosen as $C(x) = x$ so the integral can be computed analytically. Based on the obtained results we found that it is useful to extend the computational domain adding $N/2$ points left to x_1 and right to x_N that provides an accurate solution in the domain $x_1, \dots, x_N$. The drawback of this is that the resulting circulant matrix has $4N$ x $4N$ elements that increases the computational work by 4 times ($4N \log_2(4N) \approx 4(N \log_2 N)$).

Comparison with FFT - FD setup

1. In our calculations we used $x_* = 20$, $h = 2x_*/N$ regardless of the value of N which varies in the experiments. Then we extended the domain to $x_1 = -x_* - h(N/2 - 1)$, $x_N = x_* + h(N/2 + 1)$, and so this doubles the originally chosen value of N, i.e. $N_{new} = 2N$. But the final results were analyzed at the domain $x \in (-x_*, x*)$.
   Integrating the PIDE in time we use an explicit Euler scheme of the first order which is pretty fast. This is done in order to provide the worst case scenario for the below FD scheme. Thus, if our FD scheme is comparable in speed with FFT in this situation it will even better if some other more accurate integration schemes are applied together with the FFT.
2. **FD:** We build a fixed grid in the x space by choosing $S_{min} = 10^{-8}$, $S_{max} = 500$, $x_1 = \log(S_{min})$, $x_N = \log(S_{max})$, $h = (x_N - x_1)/N$, $N = 256$. The Crank-Nicolson scheme was applied.

Comparison with FFT - FD setup

1. In our calculations we used $x_* = 20$, $h = 2x_*/N$ regardless of the value of N which varies in the experiments. Then we extended the domain to $x_1 = -x_* - h(N/2-1)$, $x_N = x_* + h(N/2+1)$, and so this doubles the originally chosen value of N, i.e. $N_{new} = 2N$. But the final results were analyzed at the domain $x \in (-x_*, x*)$.
   Integrating the PIDE in time we use an explicit Euler scheme of the first order which is pretty fast. This is done in order to provide the worst case scenario for the below FD scheme. Thus, if our FD scheme is comparable in speed with FFT in this situation it will even better if some other more accurate integration schemes are applied together with the FFT.
2. **FD:** We build a fixed grid in the x space by choosing $S_{min} = 10^{-8}$, $S_{max} = 500$, $x_1 = \log(S_{min})$, $x_N = \log(S_{max})$, $h = (x_N - x_1)/N$, $N = 256$. The Crank-Nicolson scheme was applied.
3. The first series of tests was provided when $\alpha_R \in \mathbb{I}$ and $\nu_R = 1$, $\lambda_R = 0.2$.

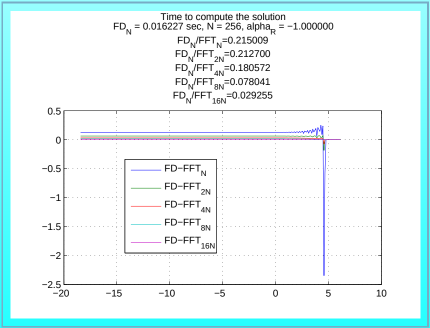


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x obtained using our finite-difference method (FD) and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -1$.

Comparison with FFT - FD setup

1. In our calculations we used $x_* = 20$, $h = 2x_*/N$ regardless of the value of N which varies in the experiments. Then we extended the domain to $x_1 = -x_* - h(N/2-1)$, $x_N = x_* + h(N/2+1)$, and so this doubles the originally chosen value of N, i.e. $N_{new} = 2N$. But the final results were analyzed at the domain $x \in (-x_*, x*)$.
   Integrating the PIDE in time we use an explicit Euler scheme of the first order which is pretty fast. This is done in order to provide the worst case scenario for the below FD scheme. Thus, if our FD scheme is comparable in speed with FFT in this situation it will even better if some other more accurate integration schemes are applied together with the FFT.
2. **FD:** We build a fixed grid in the x space by choosing $S_{min} = 10^{-8}$, $S_{max} = 500$, $x_1 = \log(S_{min})$, $x_N = \log(S_{max})$, $h = (x_N - x_1)/N$, $N = 256$. The Crank-Nicolson scheme was applied.
3. The first series of tests was provided when $\alpha_R \in \mathbb{I}$ and $\nu_R = 1$, $\lambda_R = 0.2$.

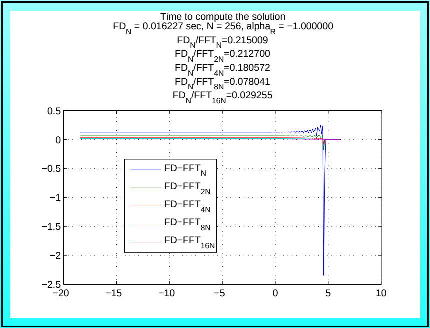


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x obtained using our finite-difference method (FD) and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -1$.

Comparison FD/FFT - cont.

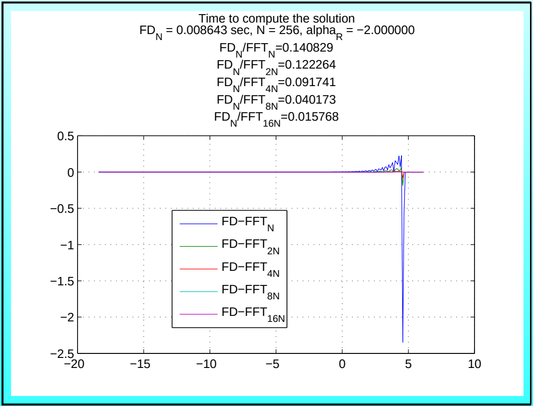


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x obtained using our finite-difference method (FD) and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -2$.

Comparison FD/FFT - cont.

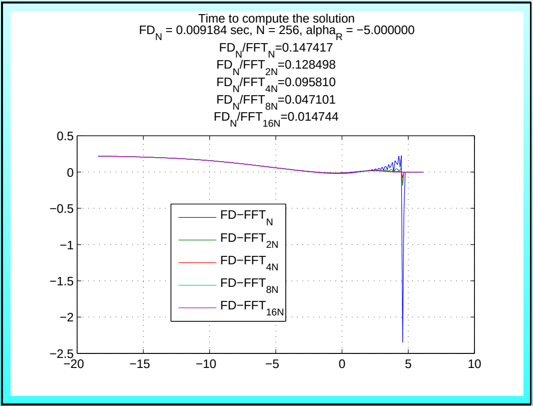


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x obtained using our finite-difference method (FD) and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -5$.

Comparison FD/FFT - cont.

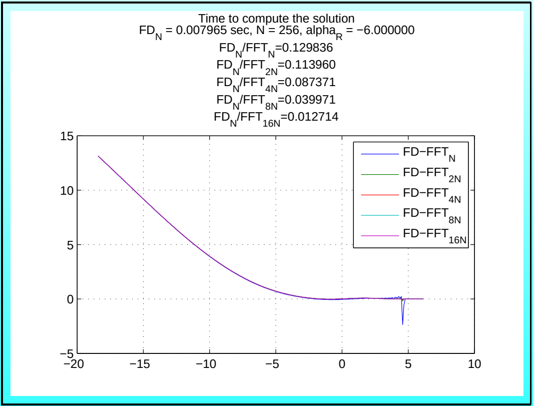


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x obtained using our finite-difference method (FD) and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -6$.

Comparison FD/FFT - cont.

- In case $\alpha_R = -1$ in Fig. 1 the FFT solution computed with $N = 256$ provides a relatively big error which disappears with N increasing. It is clear, because the Crank-Nicolson scheme is of the second order in h while the approximation Eq. (35) of the integral is of the first order in h. Numerical values of the corresponding steps in the described experiments are given in Tab. 1.

	FD_{256}	FFT_{256}	FFT_{512}	FFT_{1024}	FFT_{2048}	FFT_{4096}
h	0.096	0.1563	0.078	0.039	0.0195	0.00977

Table: Grid steps h used in the numerical experiments

  Therefore, $h^2_{FD} \approx h_{FFT_{16}}$. Actually, the difference between the FD solution with $N_{FD} = 256$ and the FFT one with $N = 4N_{FD}$ is almost negligible. However, the FD solution is computed almost 13 times faster. Even the FFT solution with $N = N_{FD}$ is 10 times slower than the FD one (It actually uses $4N$ points as it was already discussed).

- For $\alpha_R = -2$ in Fig. 4 we see almost the same picture. For $\alpha_R = -5$ speed characteristics of both solutions are almost same while the accuracy of the FD solution decreases. This is especially pronounced for $\alpha_R = -6$ in Fig. **??** at low values of x. The problem is that when α_R decreases the eigenvalues of matrix $\mathcal{B}$ grow significantly (in our tests at $\alpha_R = -6$ the eigenvalues are of order of 10^7), so the norm of matrix is very close to 1. Thus the FD method becomes just an A-stable. However, a significant difference is observed mostly at very low values of x which correspond to the spot price $S = \exp(x)$ close to zero. For a boundary problem this effect is partly dumped by the boundary condition at the low end of the domain.

Comparison FD/FFT - cont.

- In case $\alpha_R = -1$ in Fig. 1 the FFT solution computed with $N = 256$ provides a relatively big error which disappears with N increasing. It is clear, because the Crank-Nicolson scheme is of the second order in h while the approximation Eq. (35) of the integral is of the first order in h. Numerical values of the corresponding steps in the described experiments are given in Tab. 1.

	FD_{256}	FFT_{256}	FFT_{512}	FFT_{1024}	FFT_{2048}	FFT_{4096}
h	0.096	0.1563	0.078	0.039	0.0195	0.00977

  Table: Grid steps h used in the numerical experiments

  Therefore, $h^2_{FD} \approx h_{FFT_{16}}$. Actually, the difference between the FD solution with $N_{FD} = 256$ and the FFT one with $N = 4N_{FD}$ is almost negligible. However, the FD solution is computed almost 13 times faster. Even the FFT solution with $N = N_{FD}$ is 10 times slower than the FD one (It actually uses $4N$ points as it was already discussed).

- For $\alpha_R = -2$ in Fig. 4 we see almost the same picture. For $\alpha_R = -5$ speed characteristics of both solutions are almost same while the accuracy of the FD solution decreases. This is especially pronounced for $\alpha_R = -6$ in Fig. **??** at low values of x. The problem is that when α_R decreases the eigenvalues of matrix $\mathcal{B}$ grow significantly (in our tests at $\alpha_R = -6$ the eigenvalues are of order of 10^7), so the norm of matrix is very close to 1. Thus the FD method becomes just an A-stable. However, a significant difference is observed mostly at very low values of x which correspond to the spot price $S = \exp(x)$ close to zero. For a boundary problem this effect is partly dumped by the boundary condition at the low end of the domain.

PIDE, $\alpha \in \mathbb{R}$

First idea for $\alpha < 0$ - use interpolation.

The second series of tests deals with $\alpha_R \in \mathbb{R}$ using the same parameters $\nu_R = 1$, $\lambda_R = 0.2$. Four point cubic interpolation is used to compute the value of $C(x, \tau)$ at real α_R using the closest four integer values of α_R.

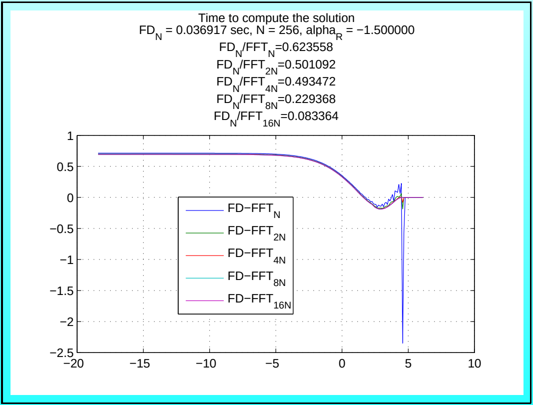


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x at $\alpha_R \in \mathbb{R}$ obtained using our finite-difference method (FD) and interpolation and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -1.5$.

PIDE, $\alpha \in \mathbb{R}$ - interpolation, cont.

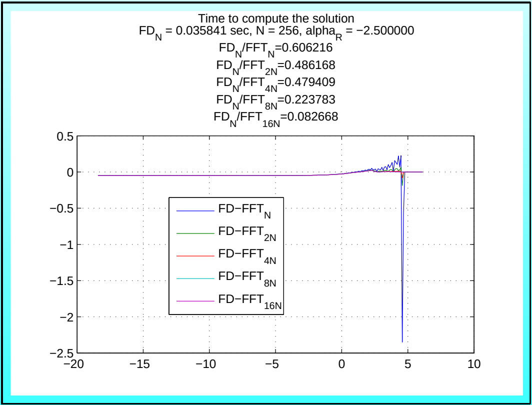


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x at $\alpha_R \in \mathbb{R}$ obtained using our finite-difference method (FD) and interpolation and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -2.5$.

PIDE, $\alpha \in \mathbb{R}$ - interpolation, cont.

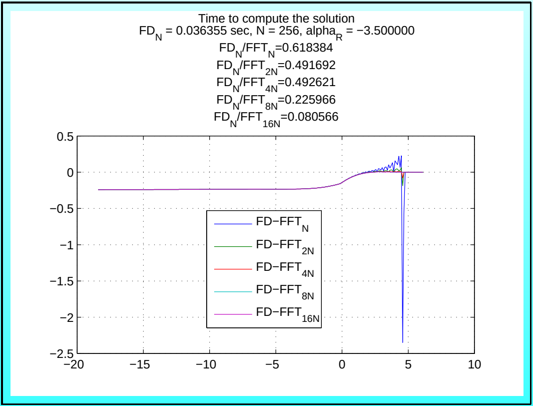


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x at $\alpha_R \in \mathbb{R}$ obtained using our finite-difference method (FD) and interpolation and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -3.5$.

PIDE, $\alpha \in \mathbb{R}$ - interpolation, cont.

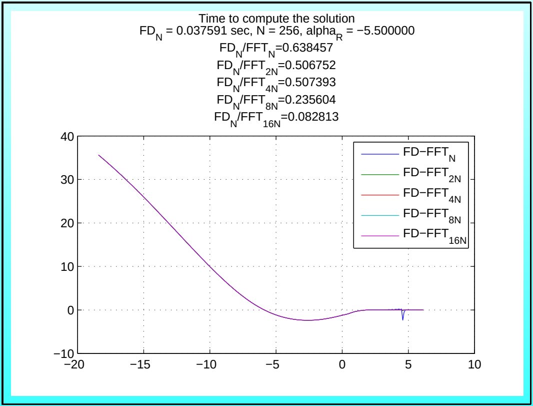


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x at $\alpha_R \in \mathbb{R}$ obtained using our finite-difference method (FD) and interpolation and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -5.5$.

PIDE, $\alpha \in \mathbb{R}$ - interpolation, cont.

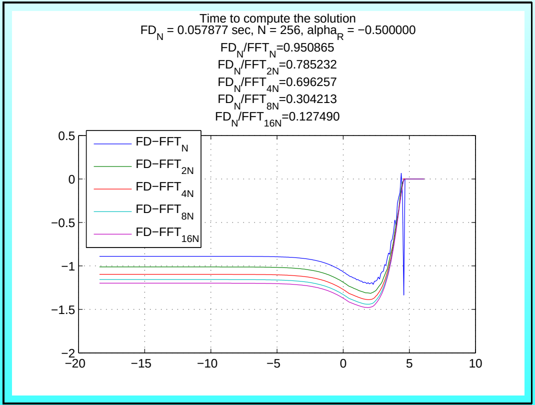


Figure: Difference (FD-FFT) in solutions of the PIDE as a function of x at $\alpha_R \in \mathbb{R}$ obtained using our finite-difference method (FD) and interpolation and an explicit Euler scheme in time where the jump integral is computed using FFT. $\alpha_R = -0.5$.

Actually, here the FFT to be accurate requires $N > 8192$.

Special cases, $\alpha = 0$

- This extreme case corresponds to the familiar Variance Gamma model. Operators $\mathcal{B}_R, \mathcal{B}_L$ have a special form.

$$\begin{aligned} \mathcal{B}_R &= \sqrt{V_R}\lambda_R \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right) \right\} \\ \mathcal{B}_L &= \sqrt{V_L}\lambda_L \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right) \right\} \end{aligned} \tag{36}$$

- Therefore, the PPE could be integrated to obtain an explicit form of the Eq. (22)

$$\begin{aligned} C^{k+1}(x) &= \left(1 - \frac{1}{\nu_R}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_R}\lambda_R\theta > 0, \\ C^{k+1}(x) &= \left(1 + \frac{1}{\nu_L}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_L}\lambda_L\theta, \end{aligned} \tag{37}$$

Special cases, $\alpha = 0$

- This extreme case corresponds to the familiar Variance Gamma model. Operators $\mathcal{B}_R, \mathcal{B}_L$ have a special form.

$$\begin{aligned} \mathcal{B}_R &= \sqrt{V_R}\lambda_R \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right)\right\} \\ \mathcal{B}_L &= \sqrt{V_L}\lambda_L \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right)\right\} \end{aligned} \tag{36}$$

- Therefore, the PPE could be integrated to obtain an explicit form of the Eq. (22)

$$\begin{aligned} C^{k+1}(x) &= \left(1 - \frac{1}{\nu_R}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_R}\lambda_R\theta > 0, \\ C^{k+1}(x) &= \left(1 + \frac{1}{\nu_L}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_L}\lambda_L\theta, \end{aligned} \tag{37}$$

- Exploit a modification of our interpolation method.

Special cases, $\alpha = 0$

- This extreme case corresponds to the familiar Variance Gamma model. Operators $\mathcal{B}_R, \mathcal{B}_L$ have a special form.

$$\begin{aligned}\mathcal{B}_R &= \sqrt{V_R}\lambda_R \left\{\log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right)\right\} \\ \mathcal{B}_L &= \sqrt{V_L}\lambda_L \left\{\log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right)\right\}\end{aligned} \tag{36}$$

- Therefore, the PPE could be integrated to obtain an explicit form of the Eq. (22)

$$\begin{aligned}C^{k+1}(x) &= \left(1 - \frac{1}{\nu_R}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_R}\lambda_R\theta > 0, \\ C^{k+1}(x) &= \left(1 + \frac{1}{\nu_L}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_L}\lambda_L\theta,\end{aligned} \tag{37}$$

- Exploit a modification of our interpolation method.

1. Typical values of λ_R, λ_L as well as V_R, V_L are limited, i.e. normally $\lambda_R < M$, $\lambda_L < M$, $V_R < M$, $V_L < M$ where M could be chosen in the range, say 3-5.

Special cases, $\alpha = 0$

- This extreme case corresponds to the familiar Variance Gamma model. Operators $\mathcal{B}_R, \mathcal{B}_L$ have a special form.

$$\begin{aligned} \mathcal{B}_R &= \sqrt{V_R}\lambda_R \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right)\right\} \\ \mathcal{B}_L &= \sqrt{V_L}\lambda_L \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right)\right\} \end{aligned} \tag{36}$$

- Therefore, the PPE could be integrated to obtain an explicit form of the Eq. (22)

$$\begin{aligned} C^{k+1}(x) &= \left(1 - \frac{1}{\nu_R}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_R}\lambda_R\theta > 0, \\ C^{k+1}(x) &= \left(1 + \frac{1}{\nu_L}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_L}\lambda_L\theta, \end{aligned} \tag{37}$$

- Exploit a modification of our interpolation method.

1. Typical values of λ_R, λ_L as well as V_R, V_L are limited, i.e. normally $\lambda_R < M, \lambda_L < M, V_R < M, V_L < M$ where M could be chosen in the range, say 3-5.
2. The time step of integration θ in the Eq. (37) is determined by the time step used at the integration of the diffusion part. So θ is usually small.

Special cases, $\alpha = 0$

- This extreme case corresponds to the familiar Variance Gamma model. Operators $\mathcal{B}_R, \mathcal{B}_L$ have a special form.

$$\begin{aligned}\mathcal{B}_R &= \sqrt{V_R}\lambda_R \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right)\right\} \\ \mathcal{B}_L &= \sqrt{V_L}\lambda_L \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right)\right\}\end{aligned} \qquad (36)$$

- Therefore, the PPE could be integrated to obtain an explicit form of the Eq. (22)

$$\begin{aligned}C^{k+1}(x) &= \left(1 - \frac{1}{\nu_R}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_R}\lambda_R\theta > 0, \\ C^{k+1}(x) &= \left(1 + \frac{1}{\nu_L}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_L}\lambda_L\theta,\end{aligned} \qquad (37)$$

- Exploit a modification of our interpolation method.

1. Typical values of λ_R, λ_L as well as V_R, V_L are limited, i.e. normally $\lambda_R < M, \lambda_L < M, V_R < M, V_L < M$ where M could be chosen in the range, say 3-5.
2. The time step of integration θ in the Eq. (37) is determined by the time step used at the integration of the diffusion part. So θ is usually small.
3. Thus, it is reasonable to assume that $m < 2$.

Special cases, $\alpha = 0$

- This extreme case corresponds to the familiar Variance Gamma model. Operators $\mathcal{B}_R, \mathcal{B}_L$ have a special form.

$$\begin{aligned} \mathcal{B}_R &= \sqrt{V_R}\lambda_R \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right)\right\} \qquad (36) \\ \mathcal{B}_L &= \sqrt{V_L}\lambda_L \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right)\right\} \end{aligned}$$

- Therefore, the PPE could be integrated to obtain an explicit form of the Eq. (22)

$$\begin{aligned} C^{k+1}(x) &= \left(1 - \frac{1}{\nu_R}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_R}\lambda_R\theta > 0, \qquad (37) \\ C^{k+1}(x) &= \left(1 + \frac{1}{\nu_L}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_L}\lambda_L\theta, \end{aligned}$$

- Exploit a modification of our interpolation method.

1. Typical values of λ_R, λ_L as well as V_R, V_L are limited, i.e. normally $\lambda_R < M, \lambda_L < M, V_R < M, V_L < M$ where M could be chosen in the range, say 3-5.
2. The time step of integration θ in the Eq. (37) is determined by the time step used at the integration of the diffusion part. So θ is usually small.
3. Thus, it is reasonable to assume that $m < 2$.
4. As follows from the definition of the fractional derivatives, the operators in the Eq. (37) are continuous in m.

Special cases, $\alpha = 0$

- This extreme case corresponds to the familiar Variance Gamma model. Operators $\mathcal{B}_R, \mathcal{B}_L$ have a special form.

$$\begin{aligned} \mathcal{B}_R &= \sqrt{V_R}\lambda_R \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right) \right\} \\ \mathcal{B}_L &= \sqrt{V_L}\lambda_L \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right) \right\} \end{aligned} \tag{36}$$

- Therefore, the PPE could be integrated to obtain an explicit form of the Eq. (22)

$$\begin{aligned} C^{k+1}(x) &= \left(1 - \frac{1}{\nu_R}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_R}\lambda_R\theta > 0, \\ C^{k+1}(x) &= \left(1 + \frac{1}{\nu_L}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_L}\lambda_L\theta, \end{aligned} \tag{37}$$

- Exploit a modification of our interpolation method.

1. Typical values of λ_R, λ_L as well as V_R, V_L are limited, i.e. normally $\lambda_R < M, \lambda_L < M, V_R < M, V_L < M$ where M could be chosen in the range, say 3-5.
2. The time step of integration θ in the Eq. (37) is determined by the time step used at the integration of the diffusion part. So θ is usually small.
3. Thus, it is reasonable to assume that $m < 2$.
4. As follows from the definition of the fractional derivatives, the operators in the Eq. (37) are continuous in m.
5. Therefore, we could solve the Eq. (37) for $m = 0, 1, 2$ and then use quadratic interpolation to get the solution given the real value of m, and the condition $m < 2$. Note, that $m = 0$ is a trivial case so the solution $C^{k+1}(X) = C^k(x)$ is already known.

Special cases, $\alpha = 0$

- This extreme case corresponds to the familiar Variance Gamma model. Operators $\mathcal{B}_R, \mathcal{B}_L$ have a special form.

$$\begin{aligned}
\mathcal{B}_R &= \sqrt{V_R}\lambda_R \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right)\right\} \\
\mathcal{B}_L &= \sqrt{V_L}\lambda_L \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right)\right\}
\end{aligned} \tag{36}$$

- Therefore, the PPE could be integrated to obtain an explicit form of the Eq. (22)

$$\begin{aligned}
C^{k+1}(x) &= \left(1 - \frac{1}{\nu_R}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_R}\lambda_R\theta > 0, \\
C^{k+1}(x) &= \left(1 + \frac{1}{\nu_L}\frac{\partial}{\partial x}\right)^{-m} C^k(x), \quad m = \sqrt{V_L}\lambda_L\theta,
\end{aligned} \tag{37}$$

- Exploit a modification of our interpolation method.

1. Typical values of λ_R, λ_L as well as V_R, V_L are limited, i.e. normally $\lambda_R < M, \lambda_L < M, V_R < M, V_L < M$ where M could be chosen in the range, say 3-5.
2. The time step of integration θ in the Eq. (37) is determined by the time step used at the integration of the diffusion part. So θ is usually small.
3. Thus, it is reasonable to assume that $m < 2$.
4. As follows from the definition of the fractional derivatives, the operators in the Eq. (37) are continuous in m.
5. Therefore, we could solve the Eq. (37) for $m = 0, 1, 2$ and then use quadratic interpolation to get the solution given the real value of m, and the condition $m < 2$. Note, that $m = 0$ is a trivial case so the solution $C^{k+1}(X) = C^k(x)$ is already known.

Numerical details, $\alpha = 0$

- Construct a stable FD scheme to solve the first equation in the Eq. (37). A forward second order approximation of the first derivative has to be chosen. Then the eigenvalues of the discrete operator $\left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m}$ are

$$\zeta = \left(1 + \frac{3}{2h\nu_R}\right)^{-m}. \qquad (38)$$

- We need to guarantee that $\| \left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m} \| < 1$. Thus, if $\nu_R < 1$ this FD scheme is stable at $h < 3/[2(1-\nu_R)]$, and if $\nu_R \geq 1$ - it is unconditionally stable. As follows from the Proposition Eq. (5) $\mathbb{R}(\nu_R) > 1$, therefore the scheme is unconditionally stable.

Numerical details, $\alpha = 0$

- Construct a stable FD scheme to solve the first equation in the Eq. (37). A forward second order approximation of the first derivative has to be chosen. Then the eigenvalues of the discrete operator $\left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m}$ are

$$\zeta = \left(1 + \frac{3}{2h\nu_R}\right)^{-m}. \tag{38}$$

- We need to guarantee that $\| \left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m} \| < 1$. Thus, if $\nu_R < 1$ this FD scheme is stable at $h < 3/[2(1-\nu_R)]$, and if $\nu_R \geq 1$ - it is unconditionally stable. As follows from the Proposition Eq. (5) $\mathbb{R}(\nu_R) > 1$, therefore the scheme is unconditionally stable.
- After this discretization the matrix of the lhs operator becomes one-sided tridiagonal if $m = 1$, and one-sided pentadiagonal if $m = 2$.

Numerical details, $\alpha = 0$

- Construct a stable FD scheme to solve the first equation in the Eq. (37). A forward second order approximation of the first derivative has to be chosen. Then the eigenvalues of the discrete operator $\left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m}$ are

$$\zeta = \left(1 + \frac{3}{2h\nu_R}\right)^{-m}. \tag{38}$$

- We need to guarantee that $\| \left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m} \| < 1$. Thus, if $\nu_R < 1$ this FD scheme is stable at $h < 3/[2(1-\nu_R)]$, and if $\nu_R \geq 1$ - it is unconditionally stable. As follows from the Proposition Eq. (5) $\mathbb{R}(\nu_R) > 1$, therefore the scheme is unconditionally stable.
- After this discretization the matrix of the lhs operator becomes one-sided tridiagonal if $m = 1$, and one-sided pentadiagonal if $m = 2$.
- To preserve monotonicity of the solution for the second equation in the Eq. (37) a backward second order approximation of the first derivative has to be chosen.

Numerical details, $\alpha = 0$

- Construct a stable FD scheme to solve the first equation in the Eq. (37). A forward second order approximation of the first derivative has to be chosen. Then the eigenvalues of the discrete operator $\left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m}$ are

$$\zeta = \left(1 + \frac{3}{2h\nu_R}\right)^{-m}. \tag{38}$$

- We need to guarantee that $\| \left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m} \| < 1$. Thus, if $\nu_R < 1$ this FD scheme is stable at $h < 3/[2(1-\nu_R)]$, and if $\nu_R \geq 1$ - it is unconditionally stable. As follows from the Proposition Eq. (5) $\mathbb{R}(\nu_R) > 1$, therefore the scheme is unconditionally stable.
- After this discretization the matrix of the lhs operator becomes one-sided tridiagonal if $m = 1$, and one-sided pentadiagonal if $m = 2$.
- To preserve monotonicity of the solution for the second equation in the Eq. (37) a backward second order approximation of the first derivative has to be chosen.
- Based on these results we extend our numerical test described in the previous section to the case $\alpha_R = 0$. However, to preserve convergence of the integral now we have to use the extended equation

$$\frac{\partial}{\partial \tau} C(x,\tau) = \int_0^\infty \left[C(x+y,\tau) - C(x,\tau)\right] \lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} dy, \quad \alpha_R < -1 \tag{39}$$

Numerical details, $\alpha = 0$

- Construct a stable FD scheme to solve the first equation in the Eq. (37). A forward second order approximation of the first derivative has to be chosen. Then the eigenvalues of the discrete operator $\left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m}$ are

$$\zeta = \left(1 + \frac{3}{2h\nu_R}\right)^{-m}. \tag{38}$$

- We need to guarantee that $\| \left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m} \| < 1$. Thus, if $\nu_R < 1$ this FD scheme is stable at $h < 3/[2(1-\nu_R)]$, and if $\nu_R \geq 1$ - it is unconditionally stable. As follows from the Proposition Eq. (5) $\mathbb{R}(\nu_R) > 1$, therefore the scheme is unconditionally stable.
- After this discretization the matrix of the lhs operator becomes one-sided tridiagonal if $m = 1$, and one-sided pentadiagonal if $m = 2$.
- To preserve monotonicity of the solution for the second equation in the Eq. (37) a backward second order approximation of the first derivative has to be chosen.
- Based on these results we extend our numerical test described in the previous section to the case $\alpha_R = 0$. However, to preserve convergence of the integral now we have to use the extended equation

$$\frac{\partial}{\partial \tau} C(x,\tau) = \int_0^\infty \left[C(x+y,\tau) - C(x,\tau)\right] \lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} dy, \quad \alpha_R < -1 \tag{39}$$

- It should be underlined that the presented simple FFT algorithm completely loses its accuracy when $\alpha_R \to 0$.

Numerical details, $\alpha = 0$

- Construct a stable FD scheme to solve the first equation in the Eq. (37). A forward second order approximation of the first derivative has to be chosen. Then the eigenvalues of the discrete operator $\left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m}$ are

$$\zeta = \left(1 + \frac{3}{2h\nu_R}\right)^{-m}. \tag{38}$$

- We need to guarantee that $\| \left(1 - \frac{1}{\nu_R}\frac{\Delta}{\Delta x}\right)^{-m} \| < 1$. Thus, if $\nu_R < 1$ this FD scheme is stable at $h < 3/[2(1-\nu_R)]$, and if $\nu_R \geq 1$ - it is unconditionally stable. As follows from the Proposition Eq. (5) $\mathbb{R}(\nu_R) > 1$, therefore the scheme is unconditionally stable.
- After this discretization the matrix of the lhs operator becomes one-sided tridiagonal if $m = 1$, and one-sided pentadiagonal if $m = 2$.
- To preserve monotonicity of the solution for the second equation in the Eq. (37) a backward second order approximation of the first derivative has to be chosen.
- Based on these results we extend our numerical test described in the previous section to the case $\alpha_R = 0$. However, to preserve convergence of the integral now we have to use the extended equation

$$\frac{\partial}{\partial \tau} C(x,\tau) = \int_0^\infty \left[C(x+y,\tau) - C(x,\tau)\right] \lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} dy, \quad \alpha_R < -1 \tag{39}$$

- It should be underlined that the presented simple FFT algorithm completely loses its accuracy when $\alpha_R \to 0$.

Special cases, $\alpha = 1$

- Let us remind that as follows from the Proposition 5 in this case the original PIDE Eq. (49) is equivalent to the PIDE

$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = \sqrt{V_R} \lambda_R \Big\{ - \nu_R \log \nu_R + (\nu_R - \frac{\partial}{\partial x}) \log \left(\nu_R - \frac{\partial}{\partial x} \right) \tag{40}$$

$$+ [\nu_R \log \nu_R - (\nu_R - 1) \log(\nu_R - 1)] \frac{\partial}{\partial x} \Big\} C(x, V_R, V_L, \tau)$$

$$\mathbb{R}(\partial / \partial x) < 0, \mathbb{R}(\nu_R) > 1,$$

- and

$$-\nu_R \log \nu_R + (\nu_R - \frac{\partial}{\partial x}) \log \left(\nu_R - \frac{\partial}{\partial x} \right) + [\nu_R \log \nu_R - (\nu_R - 1) \log(\nu_R - 1)] \frac{\partial}{\partial x}$$

$$= \int_{\nu}^{\infty} \left\{ \log \nu_R - \log \left(\nu_R - \frac{\partial}{\partial x} \right) + \left(\log \frac{\nu_R - 1}{\nu_R} \right) \frac{\partial}{\partial x} \right\} d\nu \tag{41}$$

Special cases, $\alpha = 1$

- Let us remind that as follows from the Proposition 5 in this case the original PIDE Eq. (49) is equivalent to the PIDE

$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = \sqrt{V_R}\lambda_R \Big\{ -\nu_R \log \nu_R + (\nu_R - \frac{\partial}{\partial x}) \log \left(\nu_R - \frac{\partial}{\partial x}\right) \tag{40}$$
$$+ [\nu_R \log \nu_R - (\nu_R - 1)\log(\nu_R - 1)] \frac{\partial}{\partial x} \Big\} C(x, V_R, V_L, \tau)$$
$$\mathbb{R}(\partial/\partial x) < 0, \mathbb{R}(\nu_R) > 1,$$

- and

$$-\nu_R \log \nu_R + (\nu_R - \frac{\partial}{\partial x}) \log \left(\nu_R - \frac{\partial}{\partial x}\right) + [\nu_R \log \nu_R - (\nu_R - 1)\log(\nu_R - 1)] \frac{\partial}{\partial x}$$
$$= \int_\nu^\infty \left\{ \log \nu_R - \log\left(\nu_R - \frac{\partial}{\partial x}\right) + \left(\log \frac{\nu_R - 1}{\nu_R}\right) \frac{\partial}{\partial x} \right\} d\nu \tag{41}$$

- So to construct a FD numerical method for solving the Eq. (40) we rewrite it in the form

$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = \mathbb{L}_R C(x, V_R, V_L, \tau) \tag{42}$$
$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = \mathbb{L}_L C(x, V_R, V_L, \tau)$$

$$\mathbb{L}_R \equiv \sqrt{V_R}\lambda_R \int_\nu^\infty \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right) + \left(\log \frac{\nu_R - 1}{\nu_R}\right) \frac{\partial}{\partial x} \right\} d\nu$$
$$\mathbb{L}_L \equiv \sqrt{V_L}\lambda_L \int_\nu^\infty \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right) + \log\left(\frac{\nu_L + 1}{\nu_L}\right) \frac{\partial}{\partial x} \right\} d\nu$$

We already know how to solve these equations if the operators $\mathbb{L}_R$ and $\mathbb{L}_L$ do not contain the integrals.

Special cases, $\alpha = 1$

- Let us remind that as follows from the Proposition 5 in this case the original PIDE Eq. (49) is equivalent to the PIDE

$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = \sqrt{V_R}\lambda_R \Big\{ -\nu_R \log \nu_R + (\nu_R - \frac{\partial}{\partial x}) \log\left(\nu_R - \frac{\partial}{\partial x}\right) \tag{40}$$
$$+ [\nu_R \log \nu_R - (\nu_R - 1)\log(\nu_R - 1)] \frac{\partial}{\partial x}\Big\} C(x, V_R, V_L, \tau)$$
$$\mathbb{R}(\partial/\partial x) < 0, \mathbb{R}(\nu_R) > 1,$$

- and

$$-\nu_R \log \nu_R + (\nu_R - \frac{\partial}{\partial x}) \log\left(\nu_R - \frac{\partial}{\partial x}\right) + [\nu_R \log \nu_R - (\nu_R - 1)\log(\nu_R - 1)] \frac{\partial}{\partial x}$$
$$= \int_\nu^\infty \left\{ \log \nu_R - \log\left(\nu_R - \frac{\partial}{\partial x}\right) + \left(\log \frac{\nu_R - 1}{\nu_R}\right) \frac{\partial}{\partial x} \right\} d\nu \tag{41}$$

- So to construct a FD numerical method for solving the Eq. (40) we rewrite it in the form

$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = \mathbb{L}_R C(x, V_R, V_L, \tau) \tag{42}$$
$$\frac{\partial}{\partial \tau} C(x, V_R, V_L, \tau) = \mathbb{L}_L C(x, V_R, V_L, \tau)$$
$$\mathbb{L}_R \equiv \sqrt{V_R}\lambda_R \int_\nu^\infty \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x}\right) + \left(\log \frac{\nu_R - 1}{\nu_R}\right) \frac{\partial}{\partial x} \right\} d\nu$$
$$\mathbb{L}_L \equiv \sqrt{V_L}\lambda_L \int_\nu^\infty \left\{ \log(\nu_L) - \log\left(\nu_L + \frac{\partial}{\partial x}\right) + \log\left(\frac{\nu_L + 1}{\nu_L}\right) \frac{\partial}{\partial x} \right\} d\nu$$

We already know how to solve these equations if the operators $\mathbb{L}_R$ and $\mathbb{L}_L$ do not contain the integrals.

Numerical details, $\alpha = 1$

1. First truncate the upper limit in the integral to some ν_*. This could be done because the integral in the Eq. (42) is well-defined and at $\nu_R \to \infty$ the integral kernel tends to zero as

$$\lim_{\nu_R \to \infty} \mathbb{L}_R C(x, V_R, V_L, \tau) = \sqrt{V_R}\lambda_R \frac{1}{2\nu_R^2}\left(-\frac{\partial}{\partial x} + \frac{\partial^2}{\partial x^2}\right) + O(1/\nu_R^3) \tag{43}$$

At the interval (ν, ν_*) we approximate the integral in ν using some quadrature formula or even adaptive quadratures. So we partition the interval (ν, ν_*) into an even number of intervals M all of the same width $h = (\nu_* - \nu)/M$.

2. Each operator $\mathcal{B}_R, \mathcal{B}_L$ now becomes a sum of M operators which commute with each other. Therefore, the solution of the Eq. (42) reads

$$C(x, V_R, V_L, \tau) = \exp\left[\sum_{i=0}^{M} \mathbb{L}_{i,R}\tau\right] C(x, V_R, V_L, 0) = \prod_{i=1}^{M} e^{\mathbb{L}_{i,R}\tau} C(x, V_R, V_L, 0) \tag{44}$$

$$C(x, V_R, V_L, \tau) = \exp\left[\sum_{i=0}^{M} \mathbb{L}_{i,L}\tau\right] C(x, V_R, V_L, 0) = \prod_{i=1}^{M} e^{\mathbb{L}_{i,L}\tau} C(x, V_R, V_L, 0)$$

Using a splitting technique (Lanser, Verwer, Yoshida) we can represent this equation in the form

$$\begin{aligned} C_1(x, V_R, V_L, \theta) &= e^{\mathbb{L}_{1,R}\tau} C(x, V_R, V_L, 0) \\ C_2(x, V_R, V_L, \theta) &= e^{\mathbb{L}_{2,R}\tau} C_1(x, V_R, V_L, \theta) \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ C_M(x, V_R, V_L, \theta) &= e^{\mathbb{L}_{M,R}\tau} C_{M-1}(x, V_R, V_L, \theta) \\ C(x, V_R, V_L, \theta) &= C_M(x, V_R, V_L, \theta) \end{aligned} \tag{45}$$

Numerical details, $\alpha = 1$

1. First truncate the upper limit in the integral to some ν_*. This could be done because the integral in the Eq. (42) is well-defined and at $\nu_R \to \infty$ the integral kernel tends to zero as

$$\lim_{\nu_R \to \infty} \mathbb{L}_R C(x, V_R, V_L, \tau) = \sqrt{V_R}\lambda_R \frac{1}{2\nu_R^2}\left(-\frac{\partial}{\partial x} + \frac{\partial^2}{\partial x^2}\right) + O(1/\nu_R^3) \tag{43}$$

At the interval (ν, ν_*) we approximate the integral in ν using some quadrature formula or even adaptive quadratures. So we partition the interval (ν, ν_*) into an even number of intervals M all of the same width $h = (\nu_* - \nu)/M$.

2. Each operator $\mathcal{B}_R, \mathcal{B}_L$ now becomes a sum of M operators which commute with each other. Therefore, the solution of the Eq. (42) reads

$$\begin{aligned} C(x, V_R, V_L, \tau) &= \exp\left[\sum_{i=0}^{M} \mathbb{L}_{i,R}\tau\right] C(x, V_R, V_L, 0) = \prod_{i=1}^{M} e^{\mathbb{L}_{i,R}\tau} C(x, V_R, V_L, 0) \\ C(x, V_R, V_L, \tau) &= \exp\left[\sum_{i=0}^{M} \mathbb{L}_{i,L}\tau\right] C(x, V_R, V_L, 0) = \prod_{i=1}^{M} e^{\mathbb{L}_{i,L}\tau} C(x, V_R, V_L, 0) \end{aligned} \tag{44}$$

Using a splitting technique (Lanser, Verwer, Yoshida) we can represent this equation in the form

$$\begin{aligned} C_1(x, V_R, V_L, \theta) &= e^{\mathbb{L}_{1,R}\tau} C(x, V_R, V_L, 0) \\ C_2(x, V_R, V_L, \theta) &= e^{\mathbb{L}_{2,R}\tau} C_1(x, V_R, V_L, \theta) \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ C_M(x, V_R, V_L, \theta) &= e^{\mathbb{L}_{M,R}\tau} C_{M-1}(x, V_R, V_L, \theta) \\ C(x, V_R, V_L, \theta) &= C_M(x, V_R, V_L, \theta) \end{aligned} \tag{45}$$

Numerical details, $\alpha = 1$

1. Each equation in the Eq. (45) is very similar to that corresponding to the case $\alpha = 0$. The only difference is that the operators $\mathbb{L}_{i,R}$ now contain an extra term $\mathbb{L}_{3,i,R} = \left(\log \dfrac{\nu_{i,R} - 1}{\nu_{i,R}}\right) \frac{\partial}{\partial x}$, and the operators $\mathbb{L}_{i,L}$ now contain an extra term $\mathbb{L}_{3,i,L} = \left(\log \dfrac{\nu_{i,L} + 1}{\nu_{i,L}}\right) \frac{\partial}{\partial x}$. We can apply splitting to these operators similar to as we did in the above. Further thee terms $e^{\mathbb{L}_{3,i,R}\theta}$ and $e^{\mathbb{L}_{3,i,L}\theta}$ could be approximated with the second order of accuracy in θ by using Pade.
2. Finally, each equation in the Eq. (45) reads

$$
\begin{aligned}
C_{-1}^{k+1}(x) &= C^k(x) \\
C_{i*}^{k+1}(x) &= \frac{1 + \dfrac{m_i}{2}\mathbb{L}_{3,i,R}\theta}{1 - \dfrac{m_i}{2}\mathbb{L}_{3,i,R}\theta} C_{i-1}^k(x) \\
C_i^{k+1}(x) &= \left(1 - \frac{1}{\nu_{1,R}} \frac{\partial}{\partial x}\right)^{-m_i} C_{i*}^k(x), \quad i = 0, \dots, M, \qquad m_i \equiv a_i \sqrt{V_R}\lambda_R \frac{\nu_* - \nu}{3M}\theta \\
C^{k+1}(x) &= C_M^{k+1}(x)
\end{aligned}
\tag{46}
$$

We can chose the number M to guarantee that the value of m_i is less than 2 and then use interpolation solving the above equations at $m_i = 0, 1, 2$.

Numerical details, $\alpha = 1$

1. Each equation in the Eq. (45) is very similar to that corresponding to the case $\alpha = 0$. The only difference is that the operators $\mathbb{L}_{i,R}$ now contain an extra term $\mathbb{L}_{3,i,R} = \left(\log \dfrac{\nu_{i,R}-1}{\nu_{i,R}}\right)\frac{\partial}{\partial x}$, and the operators $\mathbb{L}_{i,L}$ now contain an extra term $\mathbb{L}_{3,i,L} = \left(\log \dfrac{\nu_{i,L}+1}{\nu_{i,L}}\right)\frac{\partial}{\partial x}$. We can apply splitting to these operators similar to as we did in the above. Further thee terms $e^{\mathbb{L}_{3,i,R}}\theta$ and $e^{\mathbb{L}_{3,i,L}}\theta$ could be approximated with the second order of accuracy in θ by using Pade.
2. Finally, each equation in the Eq. (45) reads

$$
\begin{aligned}
C_{-1}^{k+1}(x) &= C^k(x) \qquad (46)\\
C_{i*}^{k+1}(x) &= \frac{1+\dfrac{m_i}{2}\mathbb{L}_{3,i,R}\theta}{1-\dfrac{m_i}{2}\mathbb{L}_{3,i,R}\theta} C_{i-1}^k(x)\\
C_i^{k+1}(x) &= \left(1-\frac{1}{\nu_{1,R}}\frac{\partial}{\partial x}\right)^{-m_i} C_{i*}^k(x), \quad i=0,\dots,M, \qquad m_i \equiv a_i\sqrt{V_R}\lambda_R\frac{\nu_*-\nu}{3M}\theta\\
C^{k+1}(x) &= C_M^{k+1}(x)
\end{aligned}
$$

We can chose the number M to guarantee that the value of m_i is less than 2 and then use interpolation solving the above equations at $m_i = 0, 1, 2$.

3. The matrix in the rhs of the second equation in the Eq. (46) is upper tridiagonal. The matrix in the rhs of the third equation in the Eq. (46) is lower tridiagonal at $m_i = 1$ and lower pentadiagonal at $m_i = 2$. The total complexity of the algorithm as compared with the case $\alpha = 0$ is: one extra equation at each step, M steps instead of just one in the case $\alpha = 0$. Therefore, using our numerical tests we can expect that at $M = 30$ this algorithm is about 3 times slower than the FFT. On the other hand it provides the second order approximation in both space and time, and 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.

Numerical details, $\alpha = 1$

1. Each equation in the Eq. (45) is very similar to that corresponding to the case $\alpha = 0$. The only difference is that the operators $\mathbb{L}_{i,R}$ now contain an extra term $\mathbb{L}_{3,i,R} = \left(\log \dfrac{\nu_{i,R} - 1}{\nu_{i,R}}\right) \frac{\partial}{\partial x}$, and the operators $\mathbb{L}_{i,L}$ now contain an extra term $\mathbb{L}_{3,i,L} = \left(\log \dfrac{\nu_{i,L} + 1}{\nu_{i,L}}\right) \frac{\partial}{\partial x}$. We can apply splitting to these operators similar to as we did in the above. Further thee terms $e^{\mathbb{L}_{3,i,R}\theta}$ and $e^{\mathbb{L}_{3,i,L}\theta}$ could be approximated with the second order of accuracy in θ by using Pade.
2. Finally, each equation in the Eq. (45) reads

$$
\begin{aligned}
C^{k+1}_{-1}(x) &= C^k(x) \\
C^{k+1}_{i*}(x) &= \frac{1 + \dfrac{m_i}{2}\mathbb{L}_{3,i,R}\theta}{1 - \dfrac{m_i}{2}\mathbb{L}_{3,i,R}\theta} C^k_{i-1}(x) \\
C^{k+1}_{i}(x) &= \left(1 - \frac{1}{\nu_{1,R}}\frac{\partial}{\partial x}\right)^{-m_i} C^k_{i*}(x), \quad i = 0, \dots, M, \qquad m_i \equiv a_i \sqrt{V_R}\lambda_R \frac{\nu_* - \nu}{3M}\theta \\
C^{k+1}(x) &= C^{k+1}_M(x)
\end{aligned}
\tag{46}
$$

We can chose the number M to guarantee that the value of m_i is less than 2 and then use interpolation solving the above equations at $m_i = 0, 1, 2$.

3. The matrix in the rhs of the second equation in the Eq. (46) is upper tridiagonal. The matrix in the rhs of the third equation in the Eq. (46) is lower tridiagonal at $m_i = 1$ and lower pentadiagonal at $m_i = 2$. The total complexity of the algorithm as compared with the case $\alpha = 0$ is: one extra equation at each step, M steps instead of just one in the case $\alpha = 0$. Therefore, using our numerical tests we can expect that at $M = 30$ this algorithm is about 3 times slower than the FFT. On the other hand it provides the second order approximation in both space and time, and 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.

Numerical details, $\alpha = 1$

To verify this we provided two numerical experiments. In the first experiment ν_* varied while $h = (\nu_* - \nu_R)/M$ was chosen to be constant. At $\nu_* = 5$ we chose $M = 30$. The other parameters are same as in the previous numerical experiments reported in the above.

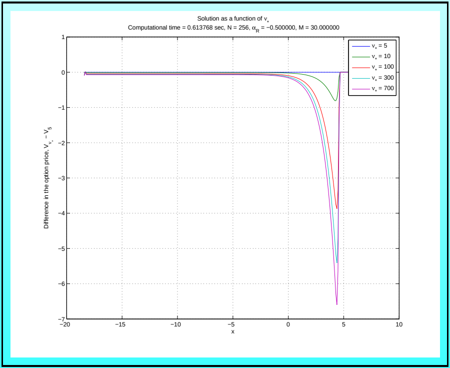


Figure: Difference in solutions obtained at various ν_* and that at $\nu_* = 5$ at $M = 30$ and $\alpha_R = 1$.

The computational time rawly increases by the factor $M/2$, i.e. for $M = 30$ it is almost same as for the corresponding FFT. It is seen that an appropriate value of ν_* should be more than 300.

Numerical details, $\alpha = 1$

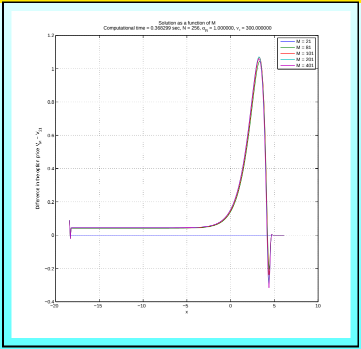
- 

Figure: Difference in solutions obtained at various M and that at $M = 21$ at $\nu_* = 300$ and $\alpha_R = 1$.

In the second experiment we fixed the value $\nu_* = 300$ and varied M to see at which M one could expect to get convergency. As it is seen $M = 81$ seems to be sufficient to obtain the convergency. The computational time in the case $M = 81$ is 1.4 sec which 3.6 times more than that for the FFT. Thus, in this case our algorithm is almost 4 times slower than the FFT.

- As it was already mentioned this could be compensated a) by the second order of accuracy in space and time, and b) no need for re-interpolation of the FFT results to the FD grid. One more advantage is that we don't need to treat the point $y = 0$ in a special way as it was done, say in Cont, Voltchkova 2003.

Numerical details, $\alpha = 1$

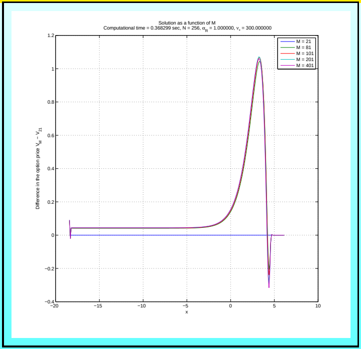
- 

  Figure: Difference in solutions obtained at various M and that at $M = 21$ at $\nu_* = 300$ and $\alpha_R = 1$.

  In the second experiment we fixed the value $\nu_* = 300$ and varied M to see at which M one could expect to get convergency. As it is seen $M = 81$ seems to be sufficient to obtain the convergency. The computational time in the case $M = 81$ is 1.4 sec which 3.6 times more than that for the FFT. Thus, in this case our algorithm is almost 4 times slower than the FFT.

- As it was already mentioned this could be compensated a) by the second order of accuracy in space and time, and b) no need for re-interpolation of the FFT results to the FD grid. One more advantage is that we don't need to treat the point $y = 0$ in a special way as it was done, say in Cont, Voltchkova 2003.
- Note, that as we use M steps in the splitting scheme, the error in time becomes $O(M\theta^2)$ that could kill the second order of approximation. Therefore, for instance, it is better to use a third order approximation in time. This scheme increases the total computational time by about 10%, however the accuracy in time increases to $O(M\theta^3)$.

Numerical details, $\alpha = 1$

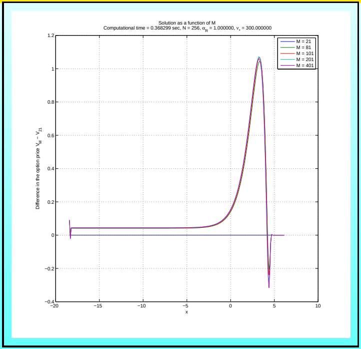
- 

  Figure: Difference in solutions obtained at various M and that at $M = 21$ at $\nu_* = 300$ and $\alpha_R = 1$.

  In the second experiment we fixed the value $\nu_* = 300$ and varied M to see at which M one could expect to get convergency. As it is seen $M = 81$ seems to be sufficient to obtain the convergency. The computational time in the case $M = 81$ is 1.4 sec which 3.6 times more than that for the FFT. Thus, in this case our algorithm is almost 4 times slower than the FFT.
- As it was already mentioned this could be compensated a) by the second order of accuracy in space and time, and b) no need for re-interpolation of the FFT results to the FD grid. One more advantage is that we don't need to treat the point $y = 0$ in a special way as it was done, say in Cont, Voltchkova 2003.
- Note, that as we use M steps in the splitting scheme, the error in time becomes $O(M\theta^2)$ that could kill the second order of approximation. Therefore, for instance, it is better to use a third order approximation in time. This scheme increases the total computational time by about 10%, however the accuracy in time increases to $O(M\theta^3)$.

Alternative methods for real α

1. Suppose we consider GTSP/KoBoL/SSM class of models. We will transform the corresponding PIDE to a fractional PDE. Fractional PDEs for Lévy processes with finite variation were obtained by Boyarchenko and Levendorsky (2002) and later by Cartea (2007) using a characteristic function technique.
2. We derive it in all cases including processes with infinite variation using a different technique - shift operators.

$$\mathfrak{S}_a = \exp\left(a\frac{\partial}{\partial x}\right), \quad \text{so} \quad \mathfrak{S}_a f(x) = f(x+a). \tag{47}$$

Alternative methods for real α

1. Suppose we consider GTSP/KoBoL/SSM class of models. We will transform the corresponding PIDE to a fractional PDE. Fractional PDEs for Lévy processes with finite variation were obtained by Boyarchenko and Levendorsky (2002) and later by Cartea (2007) using a characteristic function technique.
2. We derive it in all cases including processes with infinite variation using a different technique - shift operators.

$$\mathfrak{S}_a = \exp\left(a\frac{\partial}{\partial x}\right), \quad \text{so} \quad \mathfrak{S}_a f(x) = f(x+a). \tag{47}$$

3. and a pure jump PIDE (could be always obtained by using splitting) reads (positive jumps, but negative - by analogy)

$$\begin{aligned} &\frac{\partial}{\partial\tau}C(x,\tau) = \mathcal{B}_1 C(x,\tau) \\ &\mathcal{B}_1 \equiv \int_0^\infty \left[\exp\left(y\frac{\partial}{\partial x}\right) - 1 - (e^y - 1)\frac{\partial}{\partial x}\right]\lambda_R \frac{e^{-\nu_R|y|}}{|y|^{1+\alpha_R}}dy \end{aligned} \tag{48}$$

Formal integration could be fulfilled if we treat a differential operator $\partial/\partial x$ as a parameter.

Alternative methods for real α

1. Suppose we consider GTSP/KoBoL/SSM class of models. We will transform the corresponding PIDE to a fractional PDE. Fractional PDEs for Lévy processes with finite variation were obtained by Boyarchenko and Levendorsky (2002) and later by Cartea (2007) using a characteristic function technique.
2. We derive it in all cases including processes with infinite variation using a different technique - shift operators.

$$\mathfrak{S}_a = \exp\left(a\frac{\partial}{\partial x}\right), \quad \text{so} \quad \mathfrak{S}_a f(x) = f(x+a). \tag{47}$$

3. and a pure jump PIDE (could be always obtained by using splitting) reads (positive jumps, but negative - by analogy)

$$\begin{aligned} &\frac{\partial}{\partial \tau} C(x,\tau) = \mathcal{B}_1 C(x,\tau) \\ &\mathcal{B}_1 \equiv \int_0^\infty \left[\exp\left(y\frac{\partial}{\partial x}\right) - 1 - (e^y - 1)\frac{\partial}{\partial x}\right] \lambda_R \frac{e^{-\nu_R|y|}}{|y|^{1+\alpha_R}} dy \end{aligned} \tag{48}$$

Formal integration could be fulfilled if we treat a differential operator $\partial/\partial x$ as a parameter.

Main theorem

Theorem (1)

The PIDE

$$\frac{\partial}{\partial \tau} C(x,\tau) = \int_0^\infty \left[C(x+y,\tau) - C(x,\tau) - \frac{\partial}{\partial x} C(x,\tau)(e^y - 1) \right] \lambda_R \frac{e^{-\nu_R |y|}}{|y|^{1+\alpha_R}} dy \tag{49}$$

is equivalent to the fractional PDE

$$\begin{gathered}\frac{\partial}{\partial \tau} C(x,\tau) = \lambda_R \Gamma(-\alpha_R) \left\{ \left(\nu_R - \frac{\partial}{\partial x} \right)^{\alpha_R} - \nu_R^{\alpha_R} + \left[\nu_R^{\alpha_R} - (\nu_R - 1)^{\alpha_R} \right] \frac{\partial}{\partial x} \right\} C(x,\tau), \\ \mathbb{R}(\alpha_R) < 2, \ \mathbb{R}(\nu_R - \partial/\partial x) > 0, \ \mathbb{R}(\nu_R) > 1. \end{gathered} \tag{50}$$

In special cases this equation changes to

$$\begin{gathered}\frac{\partial}{\partial \tau} C(x,\tau) = \lambda_R \left\{ \log(\nu_R) - \log\left(\nu_R - \frac{\partial}{\partial x} \right) + \log\left(\frac{\nu_R - 1}{\nu_R} \right) \frac{\partial}{\partial x} \right\} C(x,\tau) \\ \alpha_R = 0, \mathbb{R}(\nu_R - \partial/\partial x) > 0, \mathbb{R}(\nu_R) > 1, \end{gathered} \tag{51}$$

and

$$\begin{gathered}\frac{\partial}{\partial \tau} C(x,\tau) = \lambda_R \Big\{ -\nu_R \log \nu_R + (\nu_R - \frac{\partial}{\partial x}) \log\left(\nu_R - \frac{\partial}{\partial x} \right) + [\nu_R \log \nu_R - (\nu_R - 1) \log(\nu_R - 1)] \frac{\partial}{\partial x} \Big\} C(x,\tau) \\ \alpha_R = 1, \mathbb{R}(\partial/\partial x) < 0, \mathbb{R}(\nu_R) > 1, \end{gathered}$$

Main theorem - cont.

Theorem (2)

The PIDE

$$\frac{\partial}{\partial \tau} C(x,\tau) = \int_{-\infty}^{0} \left[C(x+y,\tau) - C(x,\tau) - \frac{\partial}{\partial x} C(x,\tau)(e^y - 1) \right] \lambda_L \frac{e^{-\nu_L |y|}}{|y|^{1+\alpha_L}} dy \tag{52}$$

is equivalent to the fractional PDE

$$\begin{gathered} \frac{\partial}{\partial \tau} C(x,\tau) = \lambda_L \Gamma(-\alpha_L) \left\{ \left(\nu_L + \frac{\partial}{\partial x} \right)^{\alpha_L} - \nu_L^{\alpha_L} + \left[\nu_L^{\alpha_L} - (\nu_L + 1)^{\alpha_L} \right] \frac{\partial}{\partial x} \right\} C(x,\tau), \\ \mathbb{R}(\alpha_L) < 2, \ \mathbb{R}(\nu_L + \partial/\partial x) > 0, \ \mathbb{R}(\nu_L) > 0. \end{gathered} \tag{53}$$

In special cases this equation changes to

$$\begin{gathered} \frac{\partial}{\partial \tau} C(x,\tau) = \lambda_L \left\{ \log \left(\nu_L + \frac{\partial}{\partial x} \right) - \log(\nu_L) - \log \left(\frac{\nu_L + 1}{\nu_L} \right) \frac{\partial}{\partial x} \right\} \\ \alpha_L = 0, \ \mathbb{R}(\nu_L + \partial/\partial x) > 0, \ \mathbb{R}(\nu_L) > 0, \end{gathered} \tag{54}$$

and

$$\begin{aligned} &\frac{\partial}{\partial \tau} C(x,\tau) = \lambda_L \Big\{ - \nu_L \log \nu_L + [\nu_L \log \nu_L - (\nu_L + 1) \log(\nu_L + 1)] \frac{\partial}{\partial x} + (\nu_L + \frac{\partial}{\partial x}) \log \left(\nu_L + \frac{\partial}{\partial x} \right) \Big\} C(x,\tau) \\ &\alpha_R = 1, \ \mathbb{R}(\partial/\partial x) < 0, \ \mathbb{R}(\nu_L) > 0, \end{aligned}$$

History

Similar representations were obtained first by Boyarchenko and Levendorsky (2002) and later Cartea (2007) using a characteristic function approach. For instance, the latter authors considered several Lévy processes with known characteristic function, namely LS, CGMY or KoBoL. Then using Fourier transform they managed to convert the governing PIDE (same type as the Eq. (3) but for the Black-Scholes model with jumps) to a fractional PDE. In their notation our operator $\mathcal{A}_1$ is represented as

$$\mathcal{A}_1 \propto (-1)^{\alpha R} e^{\nu R} \; {}_x\mathbb{D}_{\infty}^{\alpha R} \left(e^{-\nu R} C(x,t) \right), \tag{55}$$

and operator $\mathcal{A}_2$ as

$$\mathcal{A}_2 \propto e^{\nu R} \; {}_{\infty}\mathbb{D}_x^{\alpha R} \left(e^{-\nu R} C(x,t) \right), \tag{56}$$

So to compare we have to note that aside of the different method of how to derive these equations our main contribution is:

1. Special cases $\alpha_r = 0, 1$, $\alpha_l = 0, 1$ are not considered by Cartea. in BL (2002) a corresponding characteristic function of the KoBoL process was obtained in all cases for $\alpha \leq 1$. However, the authors did not consider numerical solution of the fractional PDE. In this paper we derive a fractional PDE for all $\alpha < 2$ and propose a numerical method for their solution.
2. Jumps up and down are considered separately so the model in use (SSM) is slightly different from the model considered by Cartea.

History

Similar representations were obtained first by Boyarchenko and Levendorsky (2002) and later Cartea (2007) using a characteristic function approach. For instance, the latter authors considered several Lévy processes with known characteristic function, namely LS, CGMY or KoBoL. Then using Fourier transform they managed to convert the governing PIDE (same type as the Eq. (3) but for the Black-Scholes model with jumps) to a fractional PDE. In their notation our operator $\mathcal{A}_1$ is represented as

$$\mathcal{A}_1 \propto (-1)^{\alpha_R} e^{\nu_R} \ {}_x\mathbb{D}_\infty^{\alpha_R} \left(e^{-\nu_R} C(x,t) \right), \tag{55}$$

and operator $\mathcal{A}_2$ as

$$\mathcal{A}_2 \propto e^{\nu_R} \ {}_\infty\mathbb{D}_x^{\alpha_R} \left(e^{-\nu_R} C(x,t) \right), \tag{56}$$

So to compare we have to note that aside of the different method of how to derive these equations our main contribution is:

1. Special cases $\alpha_r = 0, 1$, $\alpha_l = 0, 1$ are not considered by Cartea. in BL (2002) a corresponding characteristic function of the KoBoL process was obtained in all cases for $\alpha \le 1$. However, the authors did not consider numerical solution of the fractional PDE. In this paper we derive a fractional PDE for all $\alpha < 2$ and propose a numerical method for their solution.
2. Jumps up and down are considered separately so the model in use (SSM) is slightly different from the model considered by Cartea.
3. In Cartea (2007) a Crank-Nicolson type numerical scheme was proposed to solve the obtained FPDE in time while discretization in space was done using the Grunwald-Letnikov approximation which is of the first order in space. Here we propose high-order schemes in both time and space.

History

Similar representations were obtained first by Boyarchenko and Levendorsky (2002) and later Cartea (2007) using a characteristic function approach. For instance, the latter authors considered several Lévy processes with known characteristic function, namely LS, CGMY or KoBoL. Then using Fourier transform they managed to convert the governing PIDE (same type as the Eq. (3) but for the Black-Scholes model with jumps) to a fractional PDE. In their notation our operator $\mathcal{A}_1$ is represented as

$$\mathcal{A}_1 \propto (-1)^{\alpha_R} e^{\nu_R} \, {}_x\mathbb{D}^{\alpha_R}_{\infty} \left(e^{-\nu_R} C(x,t) \right), \tag{55}$$

and operator $\mathcal{A}_2$ as

$$\mathcal{A}_2 \propto e^{\nu_R} \, {}_{\infty}\mathbb{D}^{\alpha_R}_{x} \left(e^{-\nu_R} C(x,t) \right), \tag{56}$$

So to compare we have to note that aside of the different method of how to derive these equations our main contribution is:

1. Special cases $\alpha_r = 0, 1$, $\alpha_l = 0, 1$ are not considered by Cartea. in BL (2002) a corresponding characteristic function of the KoBoL process was obtained in all cases for $\alpha \leq 1$. However, the authors did not consider numerical solution of the fractional PDE. In this paper we derive a fractional PDE for all $\alpha < 2$ and propose a numerical method for their solution.
2. Jumps up and down are considered separately so the model in use (SSM) is slightly different from the model considered by Cartea.
3. In Cartea (2007) a Crank-Nicolson type numerical scheme was proposed to solve the obtained FPDE in time while discretization in space was done using the Grunwald-Letnikov approximation which is of the first order in space. Here we propose high-order schemes in both time and space.
4. As it is known from recent papers (Abu,Saman (2007), Meerschaert,Tadjeran (2004), Tadjeran (2006), Meerschaert, Tadjeran (2006), Sousa (2008)), a standard Grunwald-Letnikov approximation leads to unconditionally unstable schemes. To improve this a shifted Grunwald-Letnikov approximation was proposed which allows construction of the unconditionally stable scheme of the first order in space. Here we use a shifted approximation to derive the unconditionally stable scheme of higher order.

History

Similar representations were obtained first by Boyarchenko and Levendorsky (2002) and later Cartea (2007) using a characteristic function approach. For instance, the latter authors considered several Lévy processes with known characteristic function, namely LS, CGMY or KoBoL. Then using Fourier transform they managed to convert the governing PIDE (same type as the Eq. (3) but for the Black-Scholes model with jumps) to a fractional PDE. In their notation our operator $\mathcal{A}_1$ is represented as

$$\mathcal{A}_1 \propto (-1)^{\alpha_R} e^{\nu_R} \, {}_x\mathbb{D}_{\infty}^{\alpha_R} \left(e^{-\nu_R} C(x,t) \right), \tag{55}$$

and operator $\mathcal{A}_2$ as

$$\mathcal{A}_2 \propto e^{\nu_R} \, {}_{\infty}\mathbb{D}_{x}^{\alpha_R} \left(e^{-\nu_R} C(x,t) \right), \tag{56}$$

So to compare we have to note that aside of the different method of how to derive these equations our main contribution is:

1. Special cases $\alpha_r = 0, 1$, $\alpha_l = 0, 1$ are not considered by Cartea. in BL (2002) a corresponding characteristic function of the KoBoL process was obtained in all cases for $\alpha \le 1$. However, the authors did not consider numerical solution of the fractional PDE. In this paper we derive a fractional PDE for all $\alpha < 2$ and propose a numerical method for their solution.
2. Jumps up and down are considered separately so the model in use (SSM) is slightly different from the model considered by Cartea.
3. In Cartea (2007) a Crank-Nicolson type numerical scheme was proposed to solve the obtained FPDE in time while discretization in space was done using the Grunwald-Letnikov approximation which is of the first order in space. Here we propose high-order schemes in both time and space.
4. As it is known from recent papers (Abu,Saman (2007), Meerschaert,Tadjeran (2004), Tadjeran (2006), Meerschaert, Tadjeran (2006), Sousa (2008)), a standard Grunwald-Letnikov approximation leads to unconditionally unstable schemes. To improve this a shifted Grunwald-Letnikov approximation was proposed which allows construction of the unconditionally stable scheme of the first order in space. Here we use a shifted approximation to derive the unconditionally stable scheme of higher order.
5. We show that when considering jumps with finite activity and finite variation despite it is a common practice to integrate out all Lévy compensators in the integral terms 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 some other terms under the integrals. To resolve this in Cartea (2007) the authors were compelled to change their definition of the fractional derivative.

History

Similar representations were obtained first by Boyarchenko and Levendorsky (2002) and later Cartea (2007) using a characteristic function approach. For instance, the latter authors considered several Lévy processes with known characteristic function, namely LS, CGMY or KoBoL. Then using Fourier transform they managed to convert the governing PIDE (same type as the Eq. (3) but for the Black-Scholes model with jumps) to a fractional PDE. In their notation our operator $\mathcal{A}_1$ is represented as

$$\mathcal{A}_1 \propto (-1)^{\alpha_R} e^{\nu_R} \; {}_x\mathbb{D}_{\infty}^{\alpha_R} \left(e^{-\nu_R} C(x,t) \right), \tag{55}$$

and operator $\mathcal{A}_2$ as

$$\mathcal{A}_2 \propto e^{\nu_R} \; {}_{\infty}\mathbb{D}_{x}^{\alpha_R} \left(e^{-\nu_R} C(x,t) \right), \tag{56}$$

So to compare we have to note that aside of the different method of how to derive these equations our main contribution is:

1. Special cases $\alpha_r = 0, 1$, $\alpha_l = 0, 1$ are not considered by Cartea. in BL (2002) a corresponding characteristic function of the KoBoL process was obtained in all cases for $\alpha \le 1$. However, the authors did not consider numerical solution of the fractional PDE. In this paper we derive a fractional PDE for all $\alpha < 2$ and propose a numerical method for their solution.
2. Jumps up and down are considered separately so the model in use (SSM) is slightly different from the model considered by Cartea.
3. In Cartea (2007) a Crank-Nicolson type numerical scheme was proposed to solve the obtained FPDE in time while discretization in space was done using the Grunwald-Letnikov approximation which is of the first order in space. Here we propose high-order schemes in both time and space.
4. As it is known from recent papers (Abu,Saman (2007), Meerschaert,Tadjeran (2004), Tadjeran (2006), Meerschaert, Tadjeran (2006), Sousa (2008)), a standard Grunwald-Letnikov approximation leads to unconditionally unstable schemes. To improve this a shifted Grunwald-Letnikov approximation was proposed which allows construction of the unconditionally stable scheme of the first order in space. Here we use a shifted approximation to derive the unconditionally stable scheme of higher order.
5. We show that when considering jumps with finite activity and finite variation despite it is a common practice to integrate out all Lévy compensators in the integral terms 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 some other terms under the integrals. To resolve this in Cartea (2007) the authors were compelled to change their definition of the fractional derivative.

Approximation

First, we remind that for integer coefficients $\alpha_R \in \mathbb{I},\ \alpha_L \in \mathbb{I}, \alpha_R < 0, \alpha_L < 0$ the FD scheme could be written in the form

$$\left(\mathbb{L}_1^- - \frac{1}{2}\mathbb{L}_2^- \theta\right) C^{k+1}(x) = \left(\mathbb{L}_1^- + \frac{1}{2}\mathbb{L}_2^- \theta\right) C^k(x), \tag{57}$$

$$\mathbb{L}_1^- \equiv \left(\nu_R - \frac{\partial}{\partial x}\right)^{-\alpha_R}, \qquad \mathbb{L}_2^- \equiv 1 - \nu_R^{\alpha_R}\mathbb{L}_1^-$$

Finite difference operators $\mathbb{L}_1^-$ and $\mathbb{L}_2^-$ have upper band matrices, and therefore the Eq. (57) can be simply solved by a backward substitution. Computational cost of such an algorithm is $C(N) \propto N(2\alpha + 1)$.

In the general case of real coefficients $\alpha_R \in \mathbb{R},\ \alpha_L \in \mathbb{R}, \alpha_R < 0, \alpha_L < 0$ it is still convenient to represent the discrete PIDE in the form of the Eq. (57). Approximation of the operator $\mathbb{L}_1^-$ should now rely on a proper definition of the fractional derivative. It is known (see, for instance, Podlubniy (2009)) that the left and right sided Riemann-Liouville derivatives are defined by

$$_aD_x^\mu \phi(x) = \frac{1}{\Gamma(m-\mu)}\left(\frac{d}{dx}\right)^m \int_a^x \frac{\phi(\xi)d\xi}{(x-\xi)^{\mu-m+1}}, \quad m-1 < \mu \le m, \tag{58}$$

$$_xD_b^\mu \phi(x) = \frac{1}{\Gamma(m-\mu)}\left(-\frac{d}{dx}\right)^m \int_x^b \frac{\phi(\xi)d\xi}{(\xi-x)^{\mu-m+1}}, \quad m-1 < \mu \le m,$$

It is also known that the left-sided Riemann-Liouville fractional derivative $_aD_x^\mu$ can be approximated in all nodes of the equidistant space discretization net simultaneously with the help of the upper(lower) triangular strip matrix F_N^α. As applied to our problem, we have to use this approximation for the operator $\mathbb{B}^-$ with $a = 0$ and $\mathbb{B}^+$ with $b = 0$. A useful approximation to the left-sided and right-sided fractional derivatives is given by the Grunwald-Letnikov formula

$$\frac{d^\alpha f(x)}{d_+x^\alpha} = \lim_{M_+\to\infty} \frac{1}{h^\alpha}\sum_{k=0}^{M_+}(-1)^k C_k^\alpha f(x-kh), \tag{59}$$

$$\frac{d^\alpha f(x)}{d_-x^\alpha} = \lim_{M_-\to\infty} \frac{1}{h^\alpha}\sum_{k=0}^{M_-}(-1)^k C_k^\alpha f(x+kh),$$

Conclusion:

1. Pseudo-differential equations can be used instead of some PIDE in mathematical finance. For GTSP/CGMY/KoBoL family of processes at $\alpha \in \mathbb{I}$ our numerical FD schemes seem to be more efficient than the standard FFT approach.
2. At $\alpha \in \mathbb{R}, \alpha < 0$ interpolation can be used together with the FD schemes still be more efficient than the FFT.

Conclusion:

1. Pseudo-differential equations can be used instead of some PIDE in mathematical finance. For GTSP/CGMY/KoBoL family of processes at $\alpha \in \mathbb{I}$ our numerical FD schemes seem to be more efficient than the standard FFT approach.
2. At $\alpha \in \mathbb{R}, \alpha < 0$ interpolation can be used together with the FD schemes still be more efficient than the FFT.
3. Fractional derivatives approach combined with a Grunwald-Letnikov scheme of higher-order can be used for any $\alpha \in \mathbb{R}, \alpha < 2$. Despite the speed becomes $O(N^2)$ vs $O(N \log(N))$ of the FFT, the accuracy of the numerical approximation could be much higher.

Conclusion:

1. Pseudo-differential equations can be used instead of some PIDE in mathematical finance. For GTSP/CGMY/KoBoL family of processes at $\alpha \in \mathbb{I}$ our numerical FD schemes seem to be more efficient than the standard FFT approach.
2. At $\alpha \in \mathbb{R}, \alpha < 0$ interpolation can be used together with the FD schemes still be more efficient than the FFT.
3. Fractional derivatives approach combined with a Grunwald-Letnikov scheme of higher-order can be used for any $\alpha \in \mathbb{R}, \alpha < 2$. Despite the speed becomes $O(N^2)$ vs $O(N \log(N))$ of the FFT, the accuracy of the numerical approximation could be much higher.