Lévy series representation: notes.

Matteo Gardini*

September 3, 2023

Abstract

In this paper, we briefly summarize some results regarding the Lévy series representation of infinitely divisible random variables and of Lévy processes. We also report some important examples and we provide some Python code.

1 Preliminaries notions and observations

Definition 1.1. *An random variable X is said to have an infinite divisible law if $\forall n \in \mathbb{N}$, there exists $\{Y_i\}_{i=1}^n$ iid random variable such that:*

$$X \stackrel{d}{=} \sum_{i=1}^{n} Y_i.$$

Definition 1.2. *The Laplace–Stieltjes transform of a random variable X is given by:*

$$\varphi_X(s) = \mathbb{E}\left[e^{-sX}\right], \quad s \in \mathbb{R}.$$

Recall the important Lévy-Khintchine formula.

Theorem 1.1. *Lévy-Khintchine A probability law μ of a real-valued random variable X is infinitely divisible with characteristic exponent ψ_X,*

$$e^{-\psi(u)} = \int_{\mathbb{R}} e^{i\theta x}\mu(dx), \quad \theta \in \mathbb{R},$$

if and only if there exists a triple (a, σ, N), where $a \in \mathbb{R}$, $sigma \geq 0$ and N is a measure concentrate in $\mathbb{R} \setminus \{0\}$ satisfying:

$$\int_R (1 \wedge y)\, N(dy) < \infty,$$

such that:

$$\psi(\theta) = ia\theta - \frac{\sigma^2\theta^2}{2} + \int_{\mathbb{R}} \left(e^{i\theta x} - 1 - i\theta x \mathbb{1}_{|x|\leq 1}\right) N(dx), \quad \theta \in \mathbb{R}.$$

*Eni Plenitude, Via Ripamonti 85, 20136, Milan, Italy, email matteo.gardini@eniplenitude.com, Department of Mathematics, University of Genoa, Via Dodecaneso 45, 16146, Genoa, Italy, email gardini@dima.unige.it

From this result (as shown in Cont and Tankov [3, Corollary 3.1]) if we consider a random variable X with infinitely divisible law and such that X assumes only non-negative values, its Laplace-Stieltjes transform is given by:

$$\varphi_X(s) = \exp\left\{-as + \int_{(0\infty)} \left(e^{-sy} - 1\right) N(dy)\right\}, \tag{1}$$

with $a \geq 0$ and N a Lévy measure. In the sequel, we assume for simplicity that $a = 0$.

1.1 A simple example: simulation from an infinitely divisible law with $N(\mathbb{R}^+) < \infty$.

Refer, for example, to Bondesson [2]. Assume that Y is a random variable assuming only non negative values with an infinite divisible law such that $a = 0$ in Equation (1). Assume that we want to find an algorithm to simulate from its distribution. Assume that:

$$\lambda = N((0, \infty)) < \infty.$$

Defining $G(dy) = N(dy)/\lambda$ and $\varphi_G(s) = \mathbb{E}\left[e^{sG}\right]$, we get:

$$\varphi_Y(s) = \exp\left\{\lambda \int_{(0,\infty)} \left(e^{-sy} - 1\right) G(dy)\right\} = \exp\{\lambda\left(\varphi_G(s) - 1\right)\} \tag{2}$$

On the other hand, assume now that $N \sim \mathcal{P}(\lambda)$ and that $\{G_i\}_{i\geq 1}$ are *iid* random variables with *cdf* given by G. Define:

$$X = \sum_{i=1}^{N} G_i.$$

If we compute the Laplace–Stieltjes transform of X we get:

$$\begin{aligned}
\mathbb{E}\left[e^{-sX}\right] &= \mathbb{E}\left[\mathbb{E}\left[e^{-s\sum_{i=1}^{n} G_i} \mid N = n\right]\right] = \mathbb{E}\left[\prod_{i=1}^{N} \varphi_G(s)\right] \\
&= \mathbb{E}\left[\varphi_G(s)^N\right] = \exp\left\{\lambda\left(\varphi_G(s) - 1\right)\right\},
\end{aligned}$$

which is the same as Equation (2). It follows that if one has to sample from the distribution of Y can use Algorithm 1.

Algorithm 1 Simulation from an infinitely divisible random variable with finite Lévy measure

1: Define $\lambda = N((0, \infty))$
2: Sample $n \sim \mathcal{P}(\lambda)$.
3: Sample $\{G_i\}_{i=1}^{n}$ *iid* random variables with law G.
4: Return $X = \sum_{i=1}^{n} G_i$.

We have that $X \stackrel{d}{=} Y$.

2 The more general case when $N((0,\infty)) = \infty$.

Assume we have a random variable Y such that its law in infinitely divisible with Laplace-Stieltjes transformation $\varphi_Y(s)$ given by:

$$\varphi_Y(s) = \exp\left\{\int_{(0,\infty)} \left(e^{-sy} - 1\right) N(dy)\right\}. \tag{3}$$

Consider a collection of independent, positive random variables $\{Z(u)\}_{u>0}$. Let be $\{T_i\}$ a sequence of jump times of a Poisson process with intensity λ. We have the following proposition:

Proposition 2.1. *Let $\{Z(u)\}_{u>0}$ be a collection of independent, positive random variables $\{Z(u)\}_{u>0}$. Let be $\{T_i\}$ a sequence of jump times of a Poisson process with intensity λ. Define:*

$$X = \sum_{i=1}^{\infty} Z(T_i).$$

Then:

$$\varphi_X(s) = \exp\left\{\lambda \int_{(0,\infty)} \left(\zeta(s;u) - 1\right) du\right\},$$

where $\zeta(s;u) = \mathbb{E}\left[\exp\{-sZ(u)\}\right]$.

Now assume that $Z(u)$ has *cdf* given by $H(y;u) = \mathbb{P}(Z(u) \le y)$. Then we have:

$$\begin{aligned}\varphi_X(s) &= \exp\left\{\lambda \int_{(0,\infty)} \left(\mathbb{E}\left[\exp\{-sZ(u)\}\right] - 1\right) du\right\}\\ &= \exp\left\{\lambda \int_{(0,\infty)} \left(\int_{(0,\infty)} \left(e^{-sy} - 1\right) H(dy;u)\right) du\right\}\\ &= \exp\left\{\lambda \int_{(0,\infty)} \left(e^{-sy} - 1\right) \left(\int_{(0,\infty)} H(dy;u) du\right)\right\}\end{aligned} \tag{4}$$

Now compare Equation (3) with Equation (4). Assume that for a given *cdf* $H(y;u)$ on $[0,\infty)$ and $\lambda > 0$ we have:

$$N(dy) = \lambda \int_{(0,\infty)} H(dy;u) du. \tag{5}$$

This is equivalent to require:

$$\bar{N}(x) = \lambda \int_0^\infty \bar{H}(x;u) du, \quad x \ge 0, \tag{6}$$

where $\bar{H} = 1 - H$. Indeed:

$$\begin{aligned}\bar{N}(x) &= \int_x^\infty N(dy) = \int_x^\infty \lambda \int_0^\infty H(dy;u) du = \lambda \int_0^\infty \left(\int_x^\infty H(dy;u) du\right)\\ &= \lambda \int_0^\infty \left(1 - H(x)\right) du = \lambda \int_0^\infty \bar{H}(x;u) du\end{aligned}$$

Since X and Y have the same Laplace-Stiltjes transform they are equal in distribution. Assume that $Z(u)$ has *pdf* given by $H(y;u)$ and that we are able to sample from the distribution of $Z(u)$. Since:

$$X = \sum_{i=1}^{\infty} Z(T_i),$$

we can sample generate a sequence of jumps-time for a Poisson process with intensity λ, $\{T_i\}_{i\geq 1}$, hence simulate from $Z(T_i)$ with *pdf* $H(y;u)$ and sum $Z(T_i)$ up.

For example, if $H(y;u) = 1 - e^{-uy}$ we are simulating $Z(T_i)$ as a random variable with exponential law with parameter T_i. Algorithm 2 formalize this procedure.

Algorithm 2 Simulation from an infinitely divisible random variable with infinite Lévy measure

---

1: Let $\{H(y;u)\}_{u>0}$ and $\lambda > 0$ be given.
2: Simulate $\{T_i\}_{i\geq 1}$, jump-times of a Poisson process with intensity $\lambda > 0$.
3: Sample $Z(T_i)$ with *cdf* $H(x;T_i)$.
4: Return $X = \sum_{i\geq 1} Z(T_i)$.

X is a random variable with infinitely divisible law and Lévy measure $N(y)$.

We have shown that in order to sample from X with an *id* law with Lévy measure $N(y)$ we must be able to find $H(y;u)$ such that Equation (5) holds and, moreover, we must be able to sample in a easy way from the *cdf* $H(y;u)$.

3 A general (theoretical) algorithm to sample from *id* law

In this section we assume that $N(dy)$ is given and that:

$$N(dy) = \lambda \int_{(0,\infty)} H(dy;u)du,$$

for some $H(y;u)$. Now assume that $H(y;u) = H(y/g(u))$ for some $g : [0,\infty) \to [0,\infty)$ and that g is non-increasing.

Let $\{V_u\}_{u>0}$ be a collection of *iid* random variables with *cdf* $H(x) = \mathbb{P}(V_u \leq x)$. If we set $Z_u = g(u)V_u$ we have that:

$$H(y;u) = \mathbb{P}(Z_u \leq y) = \mathbb{P}(g(u)V_u \leq y) = \mathbb{P}(V_u \leq y/g(u)) = H(y/g(u)).$$

Hence if $H(y;u) = H(y/g(u))$ is the *cdf* of Z_u, you can sample from it by:

- Simulate V_u with *cdf* given by $H(x)$.
- Set $Z_u = g(u)V_u$.

The function g is called the *response* and the distribution H is called the *shot-distribution*.

It is easy to show that, using a change of variables and by Equation (6):

$$\bar{N}(x) = \lambda \int_{(0,\infty)} \bar{H}(x/g(u))\, du = \lambda \int_{(0,\infty)} \bar{H}(x/v)\, g^{-1}(dv).$$

Introduce another hypothesis and assume that $V = 1$ a.s. namely that

$$H(x) = \begin{cases} 1, & \text{if } x \geq 1 \\ 0, & \text{if } x < 1. \end{cases}$$

Then:

$$\bar{N}(x) = \lambda \int_{\{u:g(u)>x\}} du = \lambda\mu \{u : g(u) > x\},$$

where μ is the Lebesgue measure. Hence:

$$\frac{\bar{N}(x)}{\lambda} = \mu \{u : g(u) > x\}.$$

We also have that:

$$g^{-1}(x) = \mu \{u : g(u) > x\} = \sup \{u : g(u) > x\}. \tag{7}$$

Remember that λ and $N(dy)$ are given and that we have to find a g satisfying Equation (7). You can do so by imposing:

$$g(u) = \mu \left\{x : \bar{N}(x)/\lambda > u\right\} = \sup \left\{x : \bar{N}(x)/\lambda > u\right\}.$$

Summing up the previous results we have the following proposition:

Proposition 3.1. *Every random variable X with infinitely divisible law on $[0, \infty)$ can be written as a shot noise distribution H degenerated at 1 with response function $g(u) = \sup \left\{x : \bar{N}(x)/\lambda > u\right\}$.*

$$X \stackrel{d}{=\!=} \sum_{i=1}^{\infty} g(T_i),$$

where $\{T_i\}_{i\geq 1}$ is a sequence of jump-times of a Poisson process with intensity λ.

Summarizing you can sample from X by:

- Simulate $\{T_i\}_{i\geq 1}$ jump-times of a Poisson process with intensity λ.
- Set:

$$X = \sum_{i\geq 1} g(T_i) = \sum_{i\geq 1} \sup \left\{x : \bar{N}(x)/\lambda > T_i\right\} = \sum_{i\geq 1} \sup \left\{x : \bar{N}(x) > T_i\lambda\right\}$$

$\bar{N}(x)$ is clearly decreasing. If we assume that $\{T_i\}_{i\geq 1}$ are increasing then the sum is a sequence of decreasing numbers.

In Cont and Tankov [3] the notation is slightly different, since it defines the tail-integral $(N\bar(x))$ as:

$$U(x) = \int_x^{\infty} N(dy),$$

and its *generalized inverse* as:

$$U^{-1}(y) = \sup \left\{x : \bar{N}(x) > y\right\}.$$

All these results together show us how to simulate from a infinitely divisible law, once that the Lévy measure is know. Unfortunately, from a practical point of view, computing the function g might be very expensive from a numerical point of view. For this reason, even if this method works in theory it is hard to apply in practice.

4 Particular forms of H and g

Remember the goal: We have a random variable with infinitely divisible law and we want to write:

$$X \stackrel{d}{=} \sum_{i=1}^{\infty} Z(T_i),$$

where $Z(u) = g(u)V_u$ with $\{V_u\}_{u \geq 0}$ *iid* with *cdf* $H(x)$. This leads to a special form of $H(y;u) = H(y/g(u))$.

Assume $\bar{H}(x)$ has a general form and that $g(u) = Ce^{-\rho u}$. Remember also that we are writing $Z(u) = g(u)V_u$, where V_u are *iid* random variables with *cdf* given by H. Assume $\rho = C = 1$ and also assume that $N(dy)$ admits a density, namely $N(dy) = n(y)dy$.

What types of Lévy measures does these choices of H and g creates (if any)?

$$\bar{N}(x) = \lambda \int_{(0,\infty)} \bar{H}\left(x/g(u)\right) du = \lambda \int_{(0,\infty)} \bar{H}\left(xe^{u}\right) du = \lambda \int_{(x,\infty)} y^{-1}\bar{H}(y)dy.$$

Moreover:

$$\bar{N}(x) = \int_x^{\infty} N(dy) = \int_x^{\infty} n(y)dy = \int_x^{\infty} \lambda y^{-1}\bar{H}(y)dy.$$

Hence we can conclude that if $g(u) = e^{-u}$ then the Lévy density must satisfy the following relation:

$$yn(y) = \lambda H(y).$$

5 A very particular case: simulating from a gamma distribution

Consider a random variable X with gamma distribution $X \sim \Gamma\left(\alpha, \beta\right)$ with *pdf*

$$f_X\left(x\right) = \frac{\beta^{\alpha}}{\Gamma\left(\alpha\right)} x^{\alpha-1}e^{-\beta x}.$$

Its Laplace-Stiltjes transform is given by:

$$\varphi_X(s) = \mathbb{E}\left[e^{-sX}\right] = \left(1 + \frac{s}{\beta}\right)^{-\alpha} = \exp\left\{\int_0^{\infty}\left(e^{-sy} - 1\right)\underbrace{\alpha y^{-1}e^{-\beta y}dy}_{N(dy)}\right\}$$

Now let be $Y = \sum_{i \geq 1} Z(T_i)$ with *cdf* given by:

$$\varphi_Y(s) = \exp\left\{\lambda \int_{(0,\infty)}\left(e^{-sy} - 1\right)\left(\int_{(0,\infty)} H(dy;u)du\right)\right\}. \tag{8}$$

Can I choose λ and H so that Equation (5) and Equation (8) have the same expression? If so, at least theoretically, I can use:

$$Y = \sum_{i \geq 1} Z(T_i), \tag{9}$$

to sample from $Y \stackrel{d}{=} X$.

5.1 Simulation algorithm

Let be $Z(u) = g(u)V_u$ where V_u are *iid* random variables with n*cdf* given by $H(x)$. Take $\bar{H}(x) = e^{-\beta x}$ and $g(u) = e^{-u}$. Hence $H(x; u) = H(x/g(u))$.

Since $g(u) = e^{-u}$, as seen in Section 4, the Lévy density must satisfy the relation:

$$yn(y) = \lambda H(y) = \lambda e^{-\beta y},$$

hence:

$$n(y) = y^{-1}\lambda e^{-\beta y}.$$

We have shown that:

$$\begin{aligned}\bar{N}(x) &= \int_x^\infty N(dy) = \int_x^\infty n(y)dy = \int_x^\infty \alpha e^{-\beta y} y^{-1} dy \\ &= \int_0^\infty \lambda \bar{H}\left(y; u\right) du = \int_0^\infty \lambda \bar{H}\left(y/g(u)\right) du = \int_x^\infty \lambda y^{-1} \bar{H}(y) dy\end{aligned}$$

which is equivalent to have shown that $N(dy) = \lambda \int_{(0,\infty)} H(dy; u)du$. Simulation from a distribution with *cdf* equal to $\bar{H}$ is easy to perform, $g(u) = e^{-u}$ hence we can use series representation to simulate from a $\Gamma(\alpha, \lambda)$ random variable. The Algorithm is given in Algorithm 3.

Algorithm 3 Simulation from gamma distribution

1: Set $\lambda = \alpha$ and simulate jump times $\{T_i\}_{i\geq 1}$ of a Poisson process of intensity λ.
2: $g(u) = e^{-u}$
3: Let $(V_u)_{u>0}$ *iid* random variables exponentially distributed with parameter β.
4: Set $Z(u) = g(u)V_u$.
5: Set:

$$Y = \sum_{i\geq 1} g(T_i)V_i = \sum_{i\geq 1} e^{-T_i} V_i \sim \Gamma\left(\alpha, \beta\right).$$

For considerations on infinitely distributions on $\mathbb{R}$ and for detailed explanation about simulations from infinitely divisible law, including how to truncate the infinite sum in Equation (9), see Bondesson [1].

5.2 Python code for simulation from a gamma $\Gamma(\alpha, \beta)$ distribution

The Python code[1] that one can use to generate from a Gamma distribution using the general method of Section 3 or the one from Section 5.1 is the following.

```
import scipy.special as sp
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gamma
```

[1]Please note that this code aims not to be a perfect exercise on object oriented programming nor aims at showing how to implement the algorithm following Python best practices. It is just a didactic example, which try to help to reader to move from theory to the numerical implementation.

```


# Define the function whose inverse you want to find
def func(x, alpha, beta):
    return alpha*sp.expn(1, beta*x)


def func_derivative(x, alpha, beta):
    """
    Fist order derivatible of exponential integral
    :param x:
    :return:
    """
    return -np.exp(-beta*x)*alpha/x


def func_sec_derivative(x, alpha, beta):
    """
    Second order derivative of the exponential integal
    :param x:
    :return:
    """
    return alpha*np.exp(-beta*x)*(beta*x+1)/x**2.0

def exponential_integral_inversion_halley(y,alpha,beta, x_guess):
    """
    Invert the exponential integral using the Halley method
    :param y: value of the exponential integral
    :param x_guess: starting guess value
    :return:
    """

    # Define the value of the function you want to invert
    # Set a tolerance for convergence
    tolerance = 1e-6

    # Maximum number of iterations
    max_iterations = 100

    x_next = None

    # Newton-Raphson iteration
    for i in range(max_iterations):
        f_x_guess = func(x_guess, alpha, beta)
        f_prime_x_guess = func_derivative(x_guess, alpha, beta)
        f_second_x_guess = func_sec_derivative(x_guess, alpha, beta
    )
        x_next = np.abs(x_guess - (2*(f_x_guess - y)*
    f_prime_x_guess)/(2*f_prime_x_guess**2.0 -

        (f_x_guess - y)*f_second_x_guess))

```

```
        # Check for convergence
        if abs(x_next - x_guess) < tolerance:
            break

        x_guess = x_next

    if np.isinf(x_guess) or np.isnan(x_guess):
        return 0
    else:
        return x_next


if __name__ == "__main__":

    np.random.seed(5)

    # Define the parameters of the gamma distribution
    alpha = 3.0  # Shape parameter (k)
    beta = 2.0  # Scale parameter (theta)

    # Truncation time of the Poisson process
    T = 50
    n_sim = 5000
    Y_bondesson = np.zeros((n_sim, 1))
    Y_general_method = np.zeros((n_sim, 1))

    # Enable to simulate from a gamma variable using the standard
   methodology
    # random_number_gamma = np.random.gamma(shape=alpha, scale=1/
   beta, size=n_sim)

    x_guess = 0.05  # Starting point for the inversion

    # Simulate using Bondesson method
    for i in range(n_sim):

        poisson_random_number = np.random.poisson(alpha*T)
        jump_times = np.random.uniform(0, T, poisson_random_number)

        for j in range(poisson_random_number):
            V = np.random.exponential(scale=1/beta)
            Y_bondesson[i, 0] = Y_bondesson[i, 0] + np.exp(-
   jump_times[j]) * V
            Y_general_method[i, 0] = Y_general_method[i, 0] +\

   exponential_integral_inversion_halley(jump_times[j], alpha=1,

            beta=beta, x_guess=x_guess)

    # Create a Boolean mask to filter values based on the threshold
   : indeed the inversion of the tail integral might not
```

```
    # converge (You can try different starting point to try to achieve the convergence)
    threshold = np.max(Y_bondesson)
    mask = Y_general_method <= threshold

    # Apply the mask to the original array to get the filtered array
    Y_general_method = Y_general_method[mask]

    # Generate random numbers from the gamma distribution
    data = Y_bondesson
    fig = plt.figure(figsize=(1920 / 100, 1080 / 100), dpi=100)
    ax = fig.add_subplot(111)

    # Create histograms of distributions
    hist, bins, _ = ax.hist(data, bins=30, edgecolor='k', density=True, label="Bondesson")
    _, _, _ = ax.hist(Y_general_method, bins=30, edgecolor='k', density=True, label="General method")

    # Calculate the PDF for each x
    x = np.linspace(np.min(data), np.max(data), 500)
    pdf = gamma.pdf(x, a=alpha, scale=1/beta)

    # Plot the gamma distribution PDF
    ax.plot(x, pdf, 'r-', lw=2, label='Gamma PDF')

    # Set labels and title
    ax.set_xlabel('Value')
    ax.set_ylabel('Frequency / PDF')
    ax.set_title('Gamma Distribution')

    # Add a legend
    ax.legend()
```

In Figure 1 we compare the gamma *pdf* with histograms obtained by using the general method and the one proposed by Bondesson [2].

6 Lévy processes: series representation

In this section we show how the technique presented in the previous sections can be adapted to simulate Lévy processes. We refer to Cont and Tankov [3, Chapter 6].

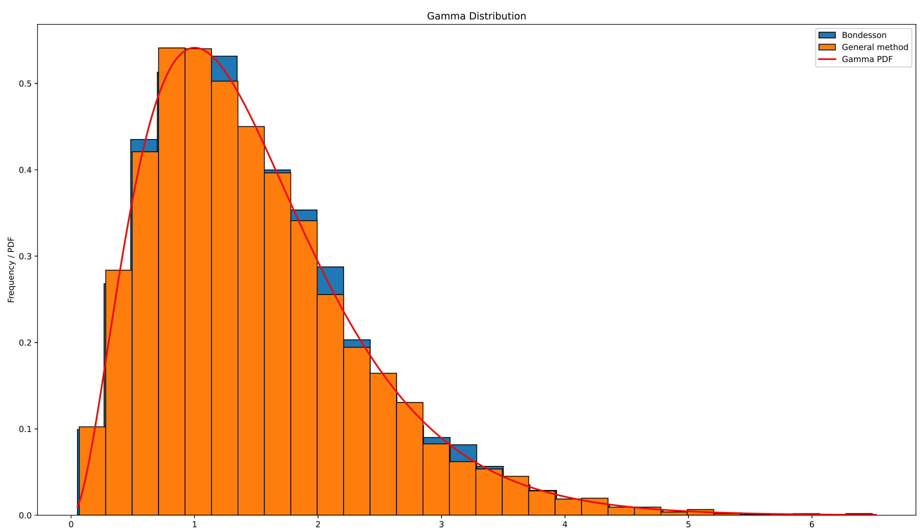

Figure 1: Probability density function of gamma random variable with parameters $\alpha = 3$ and $\beta = 2$ and histogram of the simulations obtained by using the method of Bondesson and the general one.

References

[1] L. Bondesson. *Generalized Gamma Convolutions and Related Classes of Distributions and Densities*. Springer-Verlag New York, 1992.

[2] Lennart Bondesson. On simulation from infinitely divisible distributions. *Advances in Applied Probability*, 14(4):855–869, 1982. doi: 10.2307/1427027.

[3] R. Cont and P. Tankov. *Financial Modelling with Jump Processes*. Chapman and Hall, 2003.