Powering up Fourier valuation to any dimension

Laura Ballotta[a]

1st July 2022

Fast, accurate and robust computations for derivatives prices and their sensitivities (the Greeks) are fundamental ingredients for the valuation of financial and insurance products. Efficiency of the computations is in particular crucial for model calibration. Over the last 20 years or so, Fourier transform-based approaches have shown their power for this task. Indeed they are efficient and simple to implement. All you need is the characteristic function of the driving process, the Fourier transform of the payoff function of the contract, and a good numerical scheme for the computation of the integral which will return the contract price.

Broadly speaking, there are three different popular approaches for Fourier-based valuation. The first one was proposed by Carr & Madan (1999); it starts from the Fourier transform of the option price with respect to the log-strike variable, and then relies on Fast Fourier Transform for the approximation of the resulting integral. The second approach is due to Eberlein et al. (2010). It applies the Fourier transform directly to the payoff function in the log-returns variable, and obtains the price by direct integration. The magic step is the complete separation of the payoff function from the underlying process in the resulting integral. Both approaches require an exponential damping factor to ensure integrability. The third approach, put forward by Fang & Oosterlee (2008), relies instead on the Fourier cosine series expansion of the contract payoff, and a combination of suitable truncation schemes for the approximation of the relevant integration range, and the resulting infinite summation.

Although the three approaches are essentially equivalent in terms of performance in the case of 1-dimensional vanilla contracts, and have been adapted for the pricing of a number of path-dependent instruments, such as Asian and Barrier options, the second method seems the most amenable for extensions to multi-asset contracts. I will stay with this one for the purpose of this article.

Let $X(t)$ be a stochastic process with characteristic function $\phi_{X(t)}(u)$, and assume that under the pricing measure the underlying has form

$$S(t) = S(0)e^{X(t)},$$

then the price at time 0 of a contract on S with payoff $f(X(T) - s)$ at maturity T, $s = -\ln S(0)$, can be represented as

$$V_f(X; s) = e^{-rT}\frac{e^{-Rs}}{\pi}\int_0^{\infty} \Re\left(e^{-\mathrm{i}us}\phi_{X(T)}(u - \mathrm{i}R)\hat{f}(\mathrm{i}R - u)\right) du. \tag{1}$$

[a]Corresponding Author: L.Ballotta@city.ac.uk, Faculty of Finance, Bayes Business School (formerly Cass), City, University of London, UK

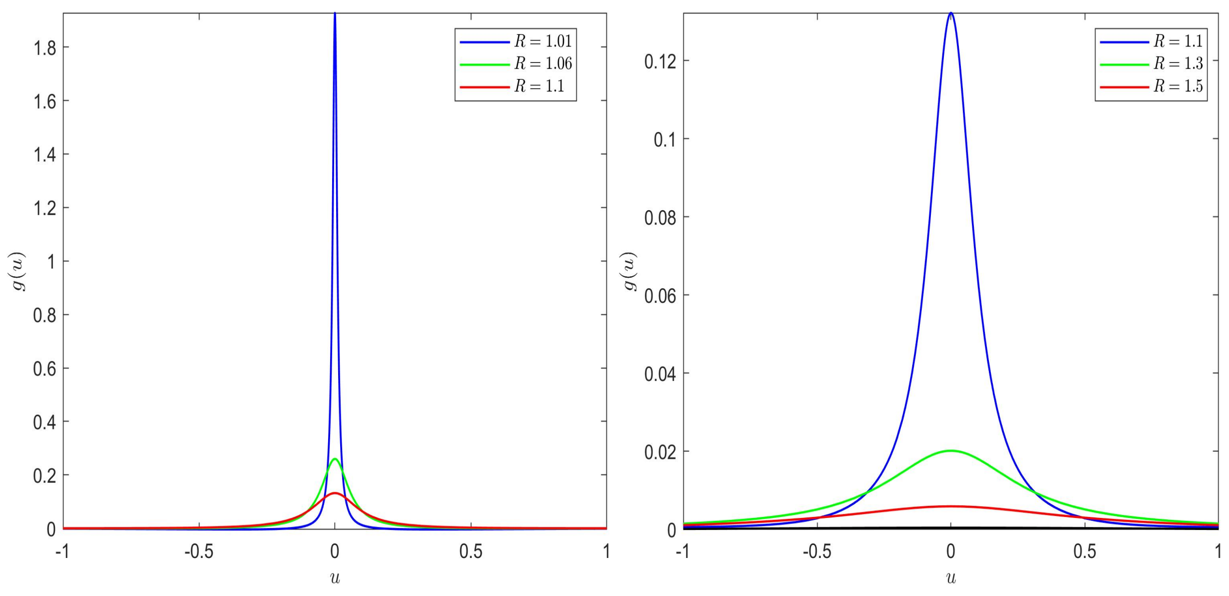

Figure 1: Impact of damping on the integrand function in equation (1): the case of a European call option. Driving process: CGMY process. Parameters as in Ballotta & Kyriakou (2014).

In the above equation $\hat{f}$ is the Fourier transform of f, r is the risk free rate of interest, R is the damping parameter ensuring sufficient integrability and $\Re$ denotes the real part of a complex number. The pricing formula (1) has a natural extension for d-dimensional contracts, like for example a basket option, which is

$$V_f(\mathbf{X};\mathbf{s}) = e^{-rT}\frac{e^{-<\mathbf{R},\mathbf{s}>}}{(2\pi)^d}\int_{\mathbb{R}^d} e^{-\mathrm{i}<\mathbf{u},\mathbf{s}>}\phi_{\mathbf{X}(T)}(\mathbf{u}-\mathrm{i}\mathbf{R})\hat{f}(\mathrm{i}\mathbf{R}-\mathbf{u})\,d\mathbf{u}. \tag{2}$$

In this case $\mathbf{X}(T)$, $\mathbf{s}$ and $\mathbf{R}$ are all d-dimensional vectors.

Eberlein et al. (2010) provide several combinations of assumptions under which the above representations hold together with an elegant proof based on Fubini's theorem. For the practical implementation of these formulas, as said before, we require a model with known characteristic function (pick your favourite stochastic volatility model or Lévy model), the Fourier transform of the payoff function and an integration routine as well as a value for the damping parameters.

Choosing such value represents in general the main hurdle for a shy beginner in Fourier pricing methods. But does it really matter which value we assign to these parameters? As long as we are careful at choosing a value in the admissible region, i.e. the region ensuring the required integrability, then yes and no. Yes because it can impact the time your computer takes to return a value. No because it does not affect the value itself.

To illustrate the point, I consider the case of a European vanilla call option on a stock price driven by the CGMY process of Carr et al. (2002). In this case, for complex z, we have

$$\hat{f}(z) = \int_{\mathbb{R}} e^{\mathrm{i}zx}\,(e^x - K)^+\,dx = \frac{K^{1+\mathrm{i}z}}{iz(1+\mathrm{i}z)},$$

if $\Im(z) > 1$, that is for $z = -u + \mathrm{i}R$ we must choose $R > 1$. Using the built-in MATLAB routine `integral` we obtain for an illustrative example the results reported in Figure 1 and in Panel A of Table 1. Figure 1 shows the different shapes of the integrand function

Table 1: Impact of damping on CPU time. Driving process: CGMY process. Parameters as in Ballotta & Kyriakou (2014).

Panel A	European Call		
	Reference Price	20.8557	
	R	CPU time (sec.)	Price difference
	1.01	0.0238	-
	1.1	0.0137	-7.58E-12
	1.3	0.0093	-5.80E-12
	1.5	0.0183	-7.52E-12
	2.1	0.0207	-7.08E-12

Panel B	$2d$ Basket Put		
	Reference Price	17.7845	
	R_1, R_2	CPU time (sec.)	Price difference
	(-0,1,-0,1)	0.3862	-
	(-1.1,-1,1)	0.0478	2.96E-07
	(-2.1,-2.1)	0.0796	2.26E-07

$$g(u) = \Re\left(e^{-\mathrm{i}us}\phi_{X(T)}(u - \mathrm{i}R)\hat{f}(\mathrm{i}R - u)\right)$$

in (1) for different values of $R > 1$ - note the different scale on the y-axis in the two plots: the damping parameter R impacts on the peakedness of the integrand function.

Table 1 reports the corresponding CPU time and the difference in values with respect to the case in which $R = 1.01$. The procedure is very fast regardless of the value of the damping parameter. Nevertheless, there are some differences in CPU time. These become more evident when tackling high dimensional contracts. We repeat the previous experiment with a basket put option. The Fourier transform of the payoff with unit strike price and unit weights is (see Hurd & Zhou, 2010, for the proof)

$$\hat{f}(\mathbf{z}) = \int_{\mathbb{R}^d} e^{\mathrm{i}<\mathbf{z},\mathbf{x}>}\left(1 - \sum_{j=1}^{d} e^{x_j}\right)^+ d\mathbf{x} = \frac{\prod_{j=1}^{d}\Gamma(\mathrm{i}z_j)}{\Gamma(\mathrm{i}\sum_{j=1}^{d} z_j + 2)}, \quad \mathbf{z} \in \mathbb{C}^d,$$

where $\Gamma(\cdot)$ is the Gamma function. Convergence is ensured if $\Re(\mathrm{i}z_j) > 0$, $j = 1, \ldots, d$; therefore, for $z_j = \mathrm{i}R_j - u_j$, we require $R_j < 0$, $j = 1, \ldots, d$. The extension to the general case is achieved by appropriate scaling. The multidimensional CGMY process is constructed following Ballotta & Bonfiglioli (2016) and Ballotta et al. (2017).

Figure 2 and Panel B of Table 1 report the results for a 2-dimensional basket. We note again the different shapes (and scale on the z-axis) of the real part of the integrand function for different values of R_1, R_2 (the integral of the imaginary part cancels out). The differences in the computational time are now more significant. Here I have used the built-in MATLAB routine `integral2`, however results would not change if we used other platforms (Python or VBA, for example).

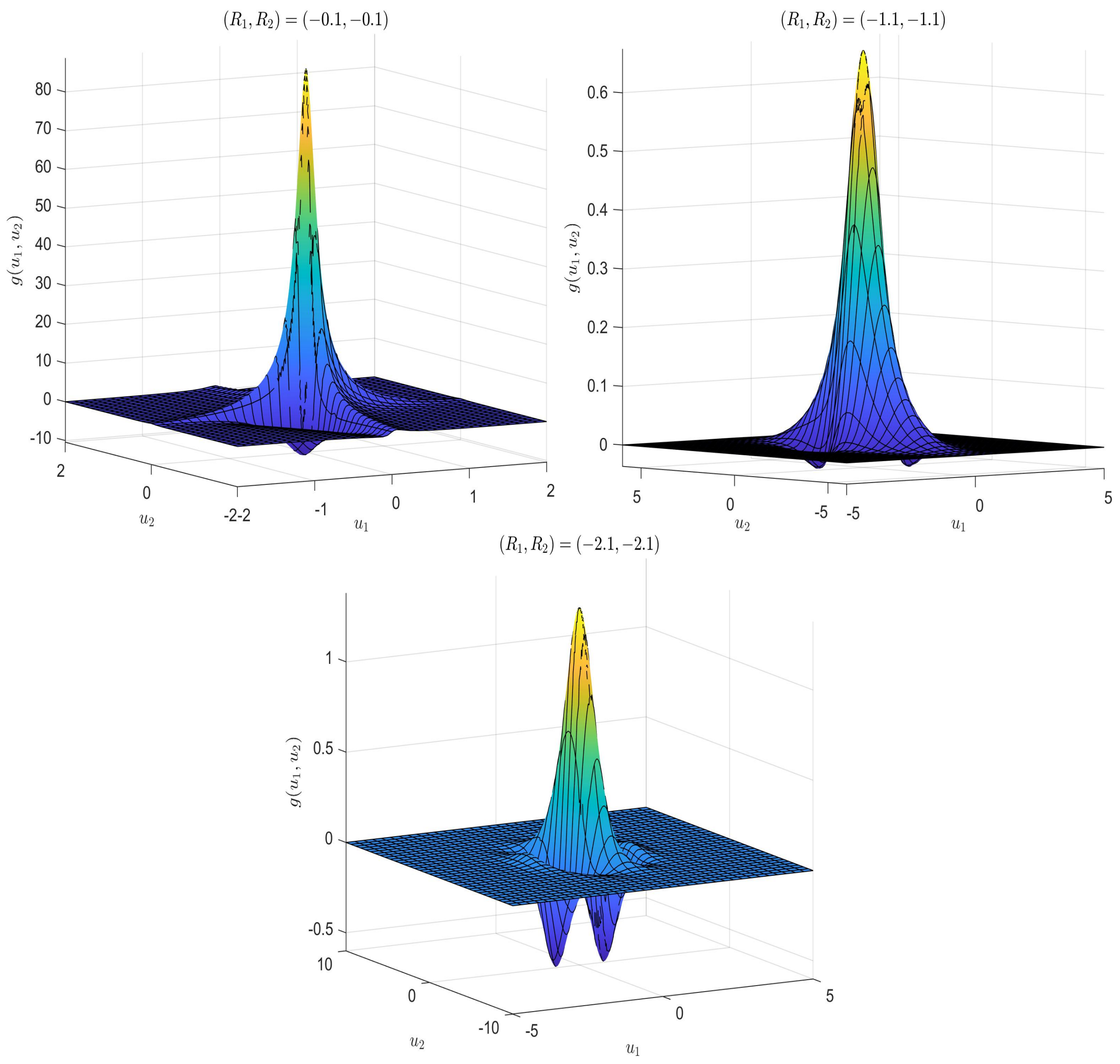

Figure 2: Impact of damping on the real part of the integrand function in equation (2): the case of a $2d$ basket put option. Driving process: CGMY process. Parameters as in Ballotta & Kyriakou (2014).

Extending the application to higher dimensions poses another implementation problem even to a more experienced user of Fourier transform techniques. Although deterministic quadrature rules are theoretically possible, practice tells us otherwise as integration beyond dimension 3 is unfeasible. A valid alternative is represented by random quadrature methods, in other words Monte Carlo integration (MCi). Given a specific d-dimensional domain for the integral, we fill in this domain with points which are randomly scattered, rather than placed on a grid by means of a chosen rule.

Assume we generate M of these random points from a multivariate distribution with joint density $p(\mathbf{u})$, then the d-dimensional integral in (2) can be approximated by

$$e^{-rT}\frac{e^{-<\mathbf{R},\mathbf{s}>}}{(2\pi)^d}\left(\frac{1}{M}\sum_{m=1}^{M}\frac{g(\mathbf{u}_m)}{p(\mathbf{u}_m)}\right), \quad \mathbf{u}_m \in \mathbb{R}^d, m = 1, \ldots, M,$$

Table 2: Computing high dimensional integrals: deterministic vs random quadrature. Driving process: CGMY process. Parameters as in Ballotta & Kyriakou (2014).

	Det. Quadrature		Random Quadrature (MCi)		
Basket Put	Price	CPU (sec.)	Price	Std. Error	CPU (sec.)
$2d, K = 90$	17.7845	0.3921	17.7843	0.0058	108.1086
$3d, K = 100$	8.2983	9.2796	8.2975	0.0043	133.7345

with

$$g(\mathbf{u}_m) = \Re\left(e^{-\mathrm{i}<\mathbf{u}_m,\mathbf{s}>}\phi_{\mathbf{X}(T)}(\mathbf{u}_m - \mathrm{i}\mathbf{R})\hat{f}(\mathrm{i}\mathbf{R} - \mathbf{u}_m)\right).$$

The procedure is already embedded in an importance sampling setting as $p(\mathbf{u})$ can be chosen to improve convergence. As illustrated in Figure 2, the integrand function is oscillatory, which can slow the convergence of the MCi procedure significantly. This means that $p(\mathbf{u})$ has to be chosen wisely. How do we do this exactly? For practical purposes, we should lean towards distributions with simple (and therefore fast) random generators, and density functions which are relatively similar to the integrand function in terms of shape. In our example, the Gaussian distribution comes to mind and indeed proves to be effective. The sample size M instead, as usual with Monte Carlo methods, has to be set to a value which achieves the desired accuracy.

In Table 2 we compare the two implementations in the case of a 2-dimensional and a 3-dimensional basket put option. The built-in deterministic quadrature routines are faster thanks to the fact that the contracts are relatively simple. More complex payoff structures, such as for example the ones found in insurance contracts like Variable Annuities, can affect significantly the CPU time for these deterministic quadrature routines. For an illustration of this impact, we refer to Ballotta et al. (2020) and Ballotta et al. (2021), in which it is shown that for Variable Annuities MCi is more efficient than deterministic quadrature routines even in dimension 3. Actually, in these papers integrals up to 9 dimensions are tackled; however, due to the long maturity of insurance policies, 20 or even 30 dimensions would not be unusual.

But then, if Monte Carlo has to be, why not resorting to traditional Monte Carlo simulation, and avoid worrying about deriving the Fourier transform of complex payoff functions, or choosing suitable damping parameters and importance sampling distributions?

As all textbooks point out, the convergence rate of Monte Carlo simulation schemes is insensitive to the underlying dimensionality. The CPU time though is not, especially when generating random numbers from the underlying (joint) distribution is non-trivial, like in the examples considered so far. The CGMY process is a pure jump Lévy process, thus a process with independent and stationary increments, a feature which makes the process 'simulation-friendly'. However, the distribution of the increments is unknown, and the process representation as a subordinated Brownian motion is not particularly helpful as the distribution of the subordinator is not known either (in stark contrast to easier examples such as the Variance Gamma or the Normal inverse Gaussian process). Nevertheless, its well known characteristic function can be used to tabulate the cumulative distribution function via inversion of the Fourier transform. In this way, random numbers from the CGMY distribution can be generated by interpolation as suggested in Ballotta & Kyriakou (2014).

Fourier meets Monte Carlo again. Which combination is better? Fourier valuation imple-

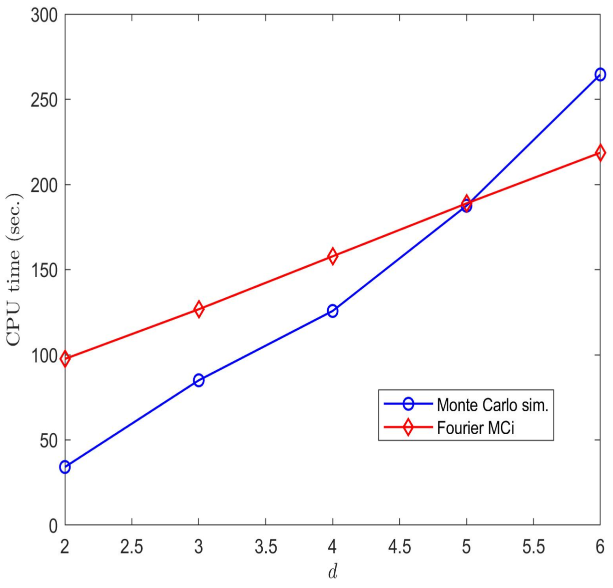

Figure 3: Fourier transform & Monte Carlo integration vs Monte Carlo simulation. The case of d-dimensional basket put options. Driving process: CGMY process. Parameters as in Ballotta & Kyriakou (2014). Importance sampling distribution: zero mean multivariate Gaussian distribution with independent components, and variance set to 1 for all components for $d = 2, 3$, 1.3 for $d = 4$, and 1.1 for $d = 5, 6$. Monte Carlo sampling of the CGMY process: Ballotta & Kyriakou (2014).

mented via Monte Carlo integration, or Monte Carlo simulation with sampling operated via Fourier inversion?

A comparison of the CPU times for the case of the basket option is offered in Figure 3: I use 10^8 iterations for both the MCi and the Monte Carlo simulation schemes; the standard errors of the prices are comparable (0.01%–0.06% of the contract price). Beyond dimension 5, the valuation formula (2) implemented by means of MCi is faster.

It has to be noticed that the CGMY process - in spite of the lack of known density function - is still simple to implement in a Monte Carlo simulation setting. The large majority of Lévy processes is based on distributions which are not closed under convolution (for example, the Hyperbolic process of Eberlein & Keller (1995) and Eberlein et al. (1998)): in these cases, the generation of random numbers from the relevant distribution can be rather demanding. The generation of random numbers can be very time consuming also for many popular stochastic volatility models, due to the need to accurately capture the value of the integrated variance, and interest rate models, in which the risk drivers are integrals of Lévy processes and not just the processes themselves.

In summary, Fourier transform approaches for the valuation of derivative structures are very powerful, regardless of the dimensionality involved. Even the powerful though need a hand to achieve their full potential: in this respect Monte Carlo integration methods can offer support.

References

Ballotta, L., & Bonfiglioli, E. 2016. Multivariate asset models using Lévy processes and applications. *The European Journal of Finance*, *22*(13), 1320–1350. doi: 10.1080/1351847X.2013.870917

Ballotta, L., Deelstra, G., & Rayée, G. 2017. Multivariate FX models with jumps: triangles, quantos and implied correlation. *European Journal of Operational Research*, *260*(3), 1181–1199. doi: 10.1016/j.ejor.2017.02.018

Ballotta, L., Eberlein, E., Schmidt, T., & Zeineddine, R. 2020. Variable Annuities in a Lévy-based hybrid model with surrender risk. *Quantitative Finance*, *20*(5), 867–886. doi: 10.1080/14697688.2019.1687929

Ballotta, L., Eberlein, E., Schmidt, T., & Zeineddine, R. 2021. Fourier based methods for the management of complex life insurance products. *Insurance: Mathematics and Economics*, *101*(B), 320–341. doi: 10.1016/j.insmatheco.2021.08.009

Ballotta, L., & Kyriakou, I. 2014. Monte Carlo simulation of the CGMY process and option pricing. *Journal of Futures Markets*, *34*(12), 1095-1121. doi: 10.1002/fut.21647

Carr, P., Geman, H., Madan, D. B., & Yor, M. 2002. The fine structure of asset returns: An empirical investigation. *Journal of Business*, *75*(2), 305–332.

Carr, P., & Madan, D. B. 1999. Option valuation using the fast Fourier transform. *Journal of Computational Finance*, *2*, 61-73.

Eberlein, E., Glau, K., & Papapantoleon, A. 2010. Analysis of Fourier transform valuation formulas and applications. *Applied Mathematical Finance*, *17*(3), 211–240.

Eberlein, E., & Keller, U. 1995. Hyperbolic distributions in finance. *Bernoulli*, *1*(3), 281–299. doi: 10.3150/bj/1193667819

Eberlein, E., Keller, U., & Prause, K. 1998, July. New insights into smile, mispricing, and Value at Risk: The Hyperbolic model. *The Journal of Business*, *71*(3), 371–405.

Fang, F., & Oosterlee, C. W. 2008. A novel pricing method for European options based on Fourier-Cosine series expansions. *SIAM Journal on Scientific Computing*, *31*(2), 826–848. doi: 10.1137/080718061

Hurd, T. R., & Zhou, Z. 2010. A Fourier transform method for spread option pricing. *SIAM Journal of Financial Mathematics*, *1*, 142-157. doi: 10.1137/090750421