Chebyshev Greeks: smoothing gamma without bias

The fast and accurate evaluation of sensitivities can be a hard task. Recent advances, such as adjoint algorithmic differentiation, are unfortunately ineffective on second-order Greeks, such as gamma, which are plagued by the most significant instabilities, so that a viable alternative to standard finite differences is still lacking. Andrea Maran, Andrea Pallavicini and Stefano Scoleri apply Chebyshev techniques to the computation of spot Greeks in Monte Carlo simulations, showing how to improve their stability in a simple and general way

The evaluation of sensitivities of financial derivatives with respect to specific market and model parameters (so-called Greeks) is a fundamental task for both the front office and the risk management department of a financial institution. Greeks are used on a daily basis by derivatives traders to hedge their books against movements in the market. Moreover, regulators are increasingly pointing towards a sensitivity based representation of financial risks (see, for example, the Fundamental Review of the Trading Book, the Standard Initial Margin Model, and the Standardised Approach to credit valuation adjustment). These all call for a fast and accurate computation of Greeks.

The standard approach to the numerical evaluation of derivatives is via finite differences (FDs). This approach implies that the pricing function is called multiple times, for each bumped scenario: this could be particularly expensive, considering that complex payoffs are usually priced via Monte Carlo simulation. Moreover, when FDs are coupled with Monte Carlo, the bias-variance problem can lead to numerical instabilities, especially for second-order Greeks: if we try to decrease the size of the bump in order to reduce the bias coming from the approximation of the derivative with an FD, then the variance of the Monte Carlo estimator of the derivative increases, making the result noisy. In practice, a trade-off with the choice of bump must be found empirically. As a result, computing Greeks in a fast and accurate way turns out to be a demanding task, possibly threatening the reliability of calculated hedge ratios.

In the last decade, the introduction of adjoint algorithmic differentiation (AAD) in the financial industry solved both the speed and accuracy problems for first-order Greeks (see, for example, Savine (2018) and the references therein): unbiased estimates of an arbitrary number of derivatives of a scalar function can be obtained at a cost that is approximately four times that of evaluating the function alone, independently of the gradient dimension. Unfortunately, this result does not generalise to second-order derivatives. Moreover, adjoint techniques may suffer from numerical instabilities, particularly for payoffs with discontinuities. All trivial tricks, such as increasing the bump or smoothing the discontinuity (eg, replacing indicator functions by tight call spreads), are able to reduce the Monte Carlo noise but invariably add a bias to the result. Therefore, if an accurate gamma is needed, we are usually forced to use a huge number of Monte Carlo paths, thus reducing computational performance.

In this work, we discuss an application of Chebyshev interpolation to the computation of Greeks of arbitrary order, aiming to improve the performance of FDs. Chebyshev interpolation techniques have recently gained increased interest in finance and, in particular, in risk management, because of their ability to boost computational performance (Gaß *et al* 2018; Glau *et al* 2019, 2020; Zeron & Ruiz 2021a,b). Their application to computation of Greeks has been addressed in a few works, such as Gaß *et al* (2018), where some theoretical convergence results are established, and Zeron & Ruiz (2021b), where an application to the computation of dynamical sensitivities for initial margin purposes is presented.

The underlying idea is to approximate, under suitable regularity conditions, the original pricing function $f(x)$ with a polynomial $p(x)$, by interpolating the values of f on a grid of n points $\{x_i\}_{i=1,\dots,n}$ in a given interval $[a, b]$ for the parameter x to be varied. If the points $\{x_i\}$ are chosen to be the Chebyshev points and f is analytical, the approximated function p_n converges exponentially to the original one for increasing n. Remarkably, this is still true for the derivatives; therefore, in the case of Chebyshev interpolation, the derivatives of the polynomial interpolant, $p^{(m)}$, are also a good approximation to the actual derivatives $f^{(m)}$. These derivatives can be computed effectively owing to the barycentric formula. In particular, we provide some heuristic rules for choosing n and $[a, b]$ so that the approximation error is minimised and possible singularities are correctly handled. We highlight that the details of our approach differ from the approach adopted in Zeron & Ruiz (2021b), where sensitivities are computed directly on Chebyshev nodes and then interpolated on generic points.

The paper is organised as follows. In the next section we introduce the theoretical framework and the proposed methodology, and then in the subsequent section we present some tests on real payoffs and assess the performance of the proposed methodology with respect to standard FDs. In the last section we state our conclusions and suggest some directions for future work. Some technical considerations on the errors of polynomial interpolation techniques applied to Monte Carlo prices and Greeks are provided in the appendices (see Maran *et al* (2021) for an extended version of the paper).

Chebyshev methods for price and Greeks approximation

This section is devoted to some theoretical considerations that form the basis of our proposal for an effective computation of Greeks. We briefly review standard FD techniques below, with a particular focus on their interaction with Monte Carlo simulations. We recall some key results on polynomial interpolation and introduce our methodology based on Chebyshev interpolation techniques for computation of Greeks.

■ **Finite differences.** Consider a function $f : U(x_0) \to \mathbb{R}$, defined on some neighbourhood $U(x_0) = [x_0-a, x_0+a]$ of x_0. Let f be differentiable at least twice in x_0.[1] In most pricing applications, when FDs are chosen to approximate Greeks, three-point central differences are used, as they provide

[1] *It is easy to extend the discussion to higher derivatives, but for financial applications we are only interested in derivatives up to second order.*

second-order approximations of derivatives in the bump h, with only two additional function evaluations:

$$f'(x_0) = \frac{1}{2h}[-f(x_0-h) + f(x_0+h)] + \mathcal{O}(h^2) \quad (1)$$

$$f''(x_0) = \frac{1}{h^2}[f(x_0-h) - 2f(x_0) + f(x_0+h)] + \mathcal{O}(h^2) \quad (2)$$

If the function f is computed by a Monte Carlo simulation, (1) and (2) can still be used, but the variance of the estimation of the derivatives increases when reducing h, which is in contrast with the need to choose a small value for h to reduce the FD bias. In particular, if the same Monte Carlo seed is used for all function evaluations, it can be shown that:

$$\text{Bias}[f', f''] = \mathcal{O}(h^2), \quad \text{Var}[f'] = \mathcal{O}(h^{-1}), \quad \text{Var}[f''] = \mathcal{O}(h^{-3}) \quad (3)$$

Equation (3) gives a clear hint why second-order derivatives are so hard to compute with FD in a Monte Carlo approach. This is the well known bias-variance problem. One way to tackle this problem is to switch to n-point central differences, for $n > 3$. In this case the bias is $\mathcal{O}(h^{n-1})$, while the variance is unaffected. For example, the seven-point differences are given by the following formulas:

$$f'(x_0) = \frac{1}{60h}[-f_{-3} + 9f_{-2} - 45f_{-1} + 45f_1 - 9f_2 + f_3] + \mathcal{O}(h^6) \quad (4)$$

$$f''(x_0) = \frac{1}{180h^2}[2f_{-3} - 27f_{-2} + 270f_{-1} - 490f_0 + 270f_1 - 27f_2 + 2f_3] + \mathcal{O}(h^6) \quad (5)$$

where $f_{\pm k} := f(x_0 \pm kh)$. Therefore, we can increase the bump size to decrease the variance without too much impact on the bias. This comes at the cost of additional revaluations of the pricing function.

■ **Polynomial interpolation.** In this section, we consider functions f defined on $[-1, 1]$. The general case can easily be obtained after an affine transformation, and the following results will be unaffected. We refer the reader to Trefethen (2020) for a complete introduction to polynomial interpolation methods.

The formulas for n-point central differences are usually derived by Taylor-expanding the function f around x_0 up to order $n-1$. However, they can also be obtained computing the derivatives, at x_0, of the Lagrange polynomial interpolating points $f(x_k)$ on a uniform grid of n points around x_0, with spacing h.

The Lagrange interpolant is the unique polynomial p_{n-1} of degree at most $n-1$ that satisfies $f(x_k) = p_{n-1}(x_k)$ for each x_k in the interpolation grid $\{x_i\}_{i=0,\dots,n-1}$. It is given by:

$$p_{n-1}(x) = \sum_{k=0}^{n-1} f(x_k)\ell_k(x) \quad (6)$$

where:

$$\ell_k(x) = \prod_{j\neq k}^{n-1} \frac{x - x_j}{x_k - x_j}$$

are the Lagrange polynomials, ie, the unique polynomials of degree $n-1$ that take a value of 1 at x_k and 0 at the other points x_i of the grid. We note that (6) is not limited to uniform grids, but also holds for generic grids.

The Lagrange interpolant can be evaluated quickly and in a numerically stable way at any point $x \in [-1, 1] \setminus \{x_i\}$ through the barycentric formula:

$$p_{n-1}(x) = \frac{\sum_{k=0}^{n-1} \frac{w_k f(x_k)}{x - x_k}}{\sum_{k=0}^{n-1} \frac{w_k}{x - x_k}}, \qquad w_k := \frac{1}{\prod_{j\neq k}^{n-1}(x_k - x_j)} \quad (7)$$

The evaluation of derivatives of a Lagrange interpolant is very easy and does not require any additional evaluation of the function f; differentiating (6) at grid points $\{x_i\}$ yields:

$$p_{n-1}^{(m)}(x_i) = \sum_{k=0}^{n-1} f(x_k)\ell_k^{(m)}(x_i) = \sum_{k=0}^{n-1} D_{ik}^{(m)} f(x_k) \quad (8)$$

which is simply the multiplication of an $n \times n$ differential matrix $D_{ik}^{(m)} := \ell_k^{(m)}(x_i)$ with the vector containing the values of the function f at the grid points. The differential matrices depend only on the grid points $\{x_i\}$ and are given by the following recursive formula (see Welfert 1997):

$$\left.\begin{aligned} D_{ik}^{(0)} &= \delta_{ik} \\ D_{ik}^{(m)} &= \begin{cases} \dfrac{m}{x_i - x_k}\left(\dfrac{w_k}{w_j} D_{ii}^{(m-1)} - D_{ik}^{(m-1)}\right) & \text{if } i \neq k \\ -\displaystyle\sum_{j\neq i}^{n-1} D_{ij}^{(m)} & \text{if } i = k \end{cases} \end{aligned}\right\} \quad (9)$$

The value of $p_{n-1}^{(m)}(x)$ at a generic point can then be obtained via the barycentric formula (7), replacing $f(x_k)$ with $p_{n-1}^{(m)}(x_k)$ as computed with (8).

It is known that for generic grids, including the uniform grid, polynomial interpolations have bad convergence properties. On the other hand, Lagrangian interpolation on the so-called Chebyshev points enjoys optimal convergence properties, at least for some classes of functions.

Let $\{z_i\}_{i=0,\dots,n-1}$ be n equally spaced points on the upper unit circle in the complex plane. Chebyshev points are defined as their projections on the real line:

$$x_k = \text{Re}[z_k] = \cos\left(\frac{k\pi}{n-1}\right), \quad k = 0, \dots, n-1 \quad (10)$$

The following result shows that exponential convergence of the Chebyshev interpolant and all its derivatives is guaranteed for analytic functions (see Trefethen 2020, chapter 21).

THEOREM 1 *Let f be an analytic function on $[-1, 1]$ that is analytically continuable to the closed Bernstein ellipse $\bar{E}_\rho$ of radius $\rho > 1$. Then, for each integer $m > 0$, there exists a constant $C > 0$ such that:*

$$\| f^{(m)} - p_{n-1}^{(m)} \|_\infty \leqslant C\rho^{-n}$$

An extension to the multivariate case is described in Gaß *et al* (2018), where specific convergence results for the partial derivatives in the $\mathcal{C}^m$-norm were also derived on weighted Sobolev spaces.

Chebyshev interpolants can be expressed in the basis of Chebyshev polynomials $\{T_k(x)\}$, with coefficients $\{c_k\}$ given as fast Fourier transforms of $\{f(x_k)\}$. However, the best way to evaluate Chebyshev interpolants and their derivatives is via the barycentric formula, where the weights can be analytically evaluated as:

$$w_k = \begin{cases} \frac{1}{2}(-1)^k & \text{if } k = 0, n-1 \\ (-1)^k & \text{otherwise} \end{cases} \quad (11)$$

■ **Chebyshev Greeks with adaptive domains.** We are now ready to formulate our proposal for the computation of Greeks. Let $f(x)$ be the price of a financial product as a function of the parameter x. We want to evaluate its derivatives numerically at some point x_0. In the applications of the present work, x_0 will be the spot price of one of the underlying assets, $f'(x_0)$ will be the delta and $f''(x_0)$ will be the gamma.

We approximate the Greeks $f^{(m)}(x_0)$ with the derivatives $p^{(m)}(x_0)$ of a Chebyshev interpolant of $f(x)$ in some region around x_0. We pick Chebyshev interpolation because of its optimal properties (described earlier). The detailed steps of the methodology are as follows:

(1) Choose the interpolation domain $H = [x_0 - a, x_0 + a]$ and the number n of Chebyshev points. It is convenient to choose n as an odd integer, so that the point x_0 is included in the Chebyshev grid and the price $f(x_0)$ will be obtained while building the interpolator (see step (2)) without additional evaluations.

(2) Build the Chebyshev interpolator as follows:
(a) Compute the Chebyshev points $\{x_k\}_{k=0,\dots,n-1}$ via (10) and map them from the unit interval to H with the appropriate affine transformation.
(b) Compute the barycentric weights via (11).
(c) Compute the differential matrices up to the desired-order m via (9).
(d) Evaluate the original pricing function on the Chebyshev points to obtain the interpolation nodes $\{f(x_k)\}_{k=0,\dots,n-1}$. This is usually the most expensive step. If f is evaluated through Monte Carlo simulation, we should fix the seed at each revaluation, so as not to add spurious discontinuities.

(3) Obtain the desired Greeks as $p^{(m)}(x_0)$ using (8) or, if n is even, (7), replacing $f(x_k)$ with $p^{(m)}(x_k)$. This step is almost instantaneous.

The above procedure depends on the choice of two adjustable parameters:[2] the number of nodes n and the domain size a. The number of nodes n should be greater than 3 (otherwise the method would degenerate to standard central differences) but not too high, in order to keep the build time of the interpolator low. We find empirically that $n = 7$ is a good compromise in most situations.

Having fixed n, we are left with the choice of parameter a. The choice of a small value for n is justified when the conditions of theorem 1 (basically, the analyticity of f inside H) are satisfied; therefore, the choice of a should be guided by the possible presence of singularities either in the price function or in its derivatives. Note that, even when the underlying function f is analytic, if it is estimated via Monte Carlo simulation, then the estimator has an error $\varepsilon_{\mathrm{MC}}$ and any approximation of f based on this estimator will only work up to, at most, this level of precision. We expect theorem 1 to be only valid up to this error; the results in appendix A show empirically that this is the case. Apart from this fact, true singularities are usually present only at known fixing dates and are located at known levels (eg, barriers or strikes). Measurable singularities can be removed with independent techniques. Even though actual singularities are formally present only at specific dates, interpolation may struggle when the singularity date approaches, showing significant oscillations, even though the pricing function is smooth; the interpolation domain should, therefore, capture this behaviour, avoiding regions where the function is almost flat. One possible solution is to adjust the size of H according to the 'time to next singularity' τ and underlying volatility σ. This is motivated by the following argument: assuming that singularities are generated by digital features (indicators present in the payoff), for digital options in the Black model[3] the scale of the singularity is given by $\sigma\sqrt{\tau}$. As we move away from the singularity, we can use larger domain sizes, thus exploiting the good properties of the Chebyshev interpolator to reduce the Monte Carlo variance of standard FDs. A possible implementation of this time- and state-adaptive strategy is the following:

(1) Let τ be the time to the next singularity date T, let $\{b_i\}_{i=1,\dots,B}$ be the positions of the singularities at time T, let σ be the at-the-money volatility of the underlying asset, estimated directly from the market quotes of plain vanilla options, and let x_0 be its spot price.

(2) Define:
(a) $a_\tau := \alpha x_0 \sigma \sqrt{\tau}$ for some $\alpha \in [1, 2]$;
(b) $d_i := |x_0 - b_i|$ for all $i = 1, \dots, B$; and
(c) $a_b := \min_{i=1,\dots,B} \frac{1}{2}(d_i - a_\tau)^+$.

(3) Set the size of the interpolation domain H equal to:

$$a = \min(\max(a_b + a_\tau, a_{\min}), a_{\max}) \tag{12}$$

where $a_{\min}$ and $a_{\max}$ are appropriate bounds. For example, we can set $a_{\min} = \lfloor n/2 \rfloor \cdot h$, where h is the characteristic bump of standard three-point central differences, and we can set $a_{\max}$ large enough to span all relevant features of the payoff (strikes, barriers, etc).

Numerical investigations

In this section, we perform some numerical experiments to assess the effectiveness of the Chebyshev methodology introduced in the previous section and compare it to standard FDs. We aim to show that, using Monte Carlo simulations, more stable Greeks can be obtained at a reduced computational cost and without significant biases. Indeed, as explained in the previous section, meaningful results for second-order Greeks can be achieved with standard techniques only by resorting to a huge number of Monte Carlo paths. On the other hand, with Chebyshev Greeks, while the number of repricings is slightly increased, the number of Monte Carlo paths for each revaluation can be dramatically reduced while preserving accuracy.

We consider two types of exotic derivatives under complex pricing models: foreign exchange target redemption forwards (TARFs) under the stochastic local volatility model by Tataru & Fisher (2010) and equity autocallables under a multi-asset local volatility model. Both payoffs can show singularities due to the presence of different types of barriers, so we can test the performance of the adaptive method outlined earlier. Additionally, in appendix B we consider a digital option under the Black model as a textbook example that allows us to perform a better error analysis.

■ **Target redemption forwards.** We consider a TARF on EUR/USD exchange rate S, with weekly put-like coupons with strike $K = 1.15$, which are paid until a maximum payout θ (target) is reached. Negative coupon payments are triggered by a knock-in barrier set at $B_{\mathrm{KI}} = 1.19$. Additionally, a knock-out barrier is present at $B_{\mathrm{KO}} = 1.135$. At each coupon fixing date

[2] *We note that standard FD techniques also depend on the choice of two parameters: the number n of points, often set to 3, and the bump size h, which is also related to the size of the approximation domain.*

[3] *The Black model is only used to estimate Chebyshev parameters, but the method works with general dynamics; in particular, we consider models with local and stochastic volatility in the 'Numerical investigations' section.*

T_i, the TARF payoff can be written as:

$$\Pi(T_i) = \frac{K - S_{T_i}}{K S_{T_i}} (\mathbf{1}_{\{S_{T_i} \leqslant K\}} + \mathbf{1}_{\{S_{T_i} > B_{\mathrm{KI}}\}}) \times \mathbf{1}_{\{\sum_{j=1}^{i} (K - S_{T_j})^+ < \theta\}} \left(1 - \min\left\{1, \sum_{j=1}^{i} \mathbf{1}_{\{S_{T_j} < B_{\mathrm{KO}}\}}\right\}\right) N_i \tag{13}$$

where N_i are coupon notionals. There are 70 remaining coupons, and the residual target is 0.2. Figure 1 shows the results of gamma for different spot levels and evaluation dates. The details of the computation are summarised in table A. The improved stability of the Chebyshev methodology is evident. We highlight that FD Greeks were obtained with 1,000,000 simulation paths for each call to the pricing function, while Chebyshev Greeks were obtained with 300,000 paths for each call; since seven Chebyshev points were employed versus three points for central FD, overall we also reduced the computational time by one-third. Despite the superior accuracy of the Chebyshev result, there is still room for further computational time savings, depending on the desired accuracy threshold.

In order to assess the consistency of our methodology, we proceed as follows. We check how the computed gamma is good at explaining the actual change in delta, with the different methods. For this purpose, we define the following 'explanation error':

$$\varepsilon_M = \max_p |\Gamma_M(S_p) \cdot \mathrm{d}S_p - \mathrm{d}\Delta_M(S_p)| \tag{14}$$

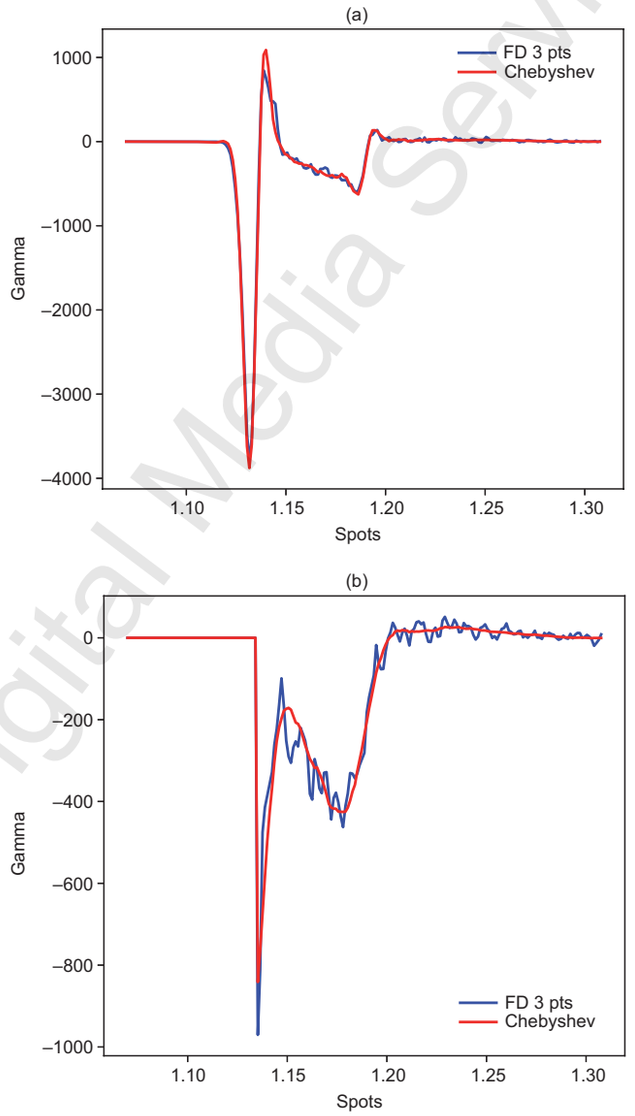
where the spots S_p run over the grid of points used for the tests, and Δ_M, Γ_M are delta and gamma evaluated with the numerical method M, respectively. Note that this explanation error says nothing about numerical errors built in the computations (the latter are discussed in appendix B for a simpler test case). They are simply used to check the self-coherence of Chebyshev Greeks: this is worth doing, since at each spot S_p a different interpolator is built. We show the results in table A: the explanation errors are comparable for all the methods analysed.

Autocallables. We now consider an autocallable option with memory, on a basket of two stocks: Telecom Italia and Vodafone. The option pays a stream of coupons at times T_i, provided that the performance of the basket, with respect to a past strike date t_{ref}, is above $B_{\mathrm{coup}} = 90\%$. Additionally, an early termination feature is present that gets activated if the basket performance is over $B_{\mathrm{call}} = 100\%$ at some T_i. Finally, there is a 'capital guarantee' barrier at $B_{\mathrm{guar}} = 60\%$ on the last fixing date, T_N. The basket performance is of the 'worst-of' type. At each coupon fixing date, the autocallable payoff can be written as:

$$\Pi(T_i) = \mathbf{1}_{\{\tau > T_i\}} \times \left(\left[N_i + \sum_{j=1}^{i-1} (N_j - \Pi(T_j)) \right] \mathbf{1}_{\{P(T_i) \geqslant B_{\mathrm{coup}}\}} + \delta_{iN} (P(T_N) - 1) \mathbf{1}_{\{P(T_N) < B_{\mathrm{guar}}\}} \right) \tag{15}$$

where N_i are coupon notionals, $P(t) = \min\{S_t^{\mathrm{tel}}/S_{t_{\mathrm{ref}}}^{\mathrm{tel}}, S_t^{\mathrm{vod}}/S_{t_{\mathrm{ref}}}^{\mathrm{vod}}\}$ is the basket performance and $\tau = \min\{T_i : P(T_i) \geqslant B_{\mathrm{call}}\}$ is the early termination time. There are seven coupons remaining every three months. The initial fixings of the underlying assets, for the computation of the performances, are $S_{t_{\mathrm{ref}}}^{\mathrm{tel}} = €0.48$ for Telecom Italia and $S_{t_{\mathrm{ref}}}^{\mathrm{vod}} = £1.3$ for Vodafone. The current spot price of Vodafone is kept fixed to $S_0^{\mathrm{vod}} = £1.35$.

1 Gamma of a TARF for 200 spot levels for (a) one day to the next fixing date and (b) one week to the next fixing date



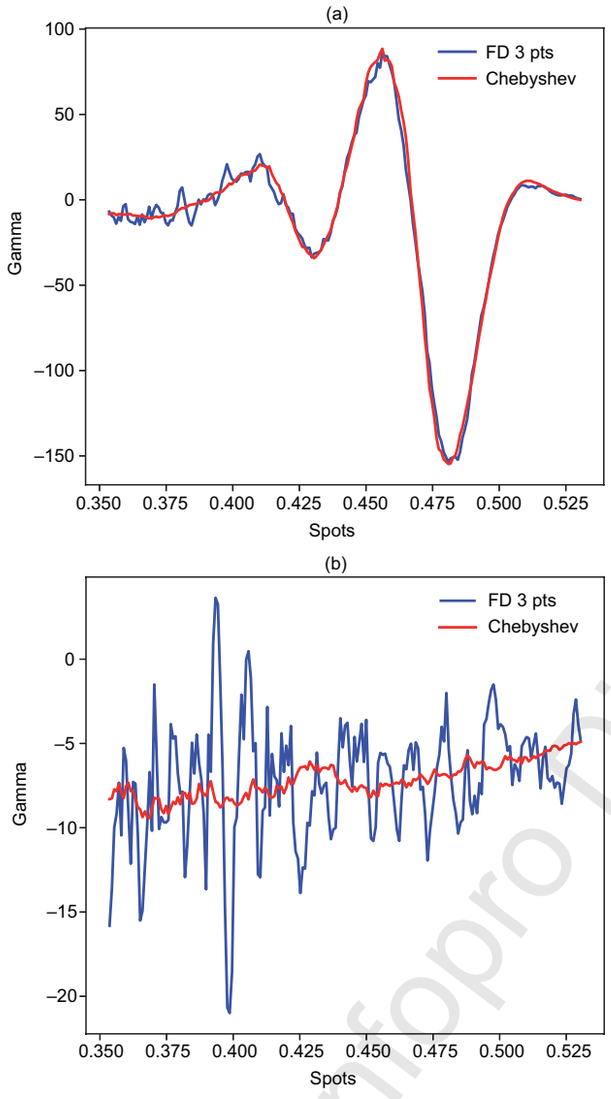
For the three-point central differences (blue) 1,000,000 Monte Carlo paths are used, while for the seven-point Chebyshev Greeks with time- and state-adaptive interpolation domain (red) 300,000 Monte Carlo paths are used. Singularities show up at \$/€1.135 and \$/€1.19 spot levels

In figure 2 we show the results of gamma for different evaluation dates and spot levels of Telecom Italia. The details of the computation are summarised in table A. Again, 300,000 Monte Carlo paths were used with Chebyshev Greeks, against 1,000,000 paths with FDs. The same caveats as for TARF results apply here and confirm the effectiveness of Chebyshev method.

Conclusions and further developments

In this work we presented a simple method to numerically evaluate the delta and gamma of arbitrarily complex payoffs. The method is based on Chebyshev interpolation over a suitable domain around the spot price. The degrees of freedom available in the choice of the interpolation domain enable us to

2 Gamma of a worst-of autocallable on Telecom Italia and Vodafone, for 200 spot levels of Telecom Italia for (a) three days to the next fixing date and (b) three weeks to the next fixing date



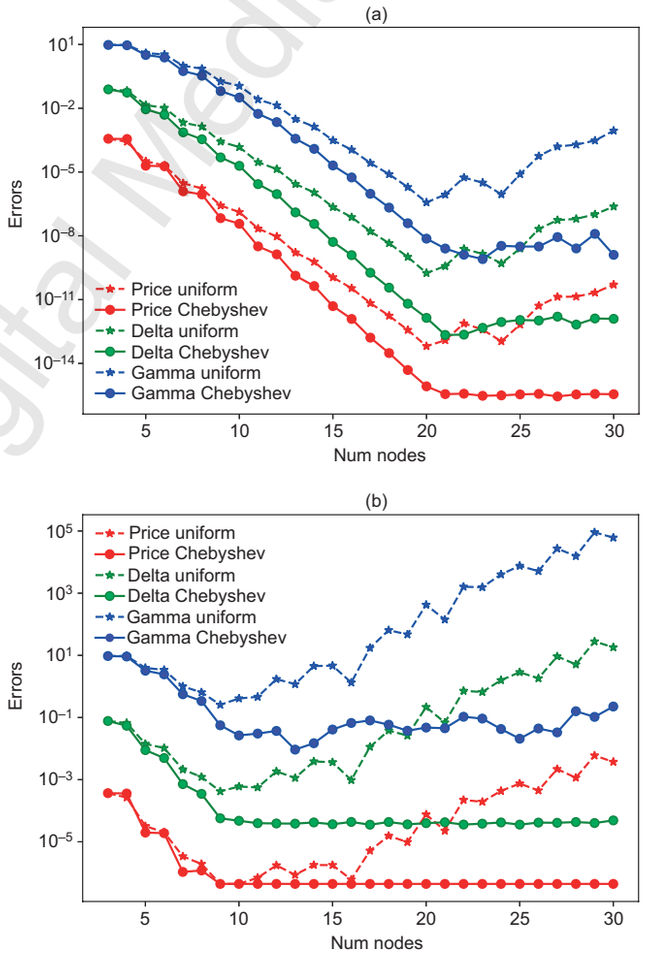
For the three-point central differences (blue) 1,000,000 Monte Carlo paths are used, while for the seven-point Chebyshev Greeks time- and state-adaptive interpolation domain (red) 300,000 Monte Carlo paths are used. Singularities show up at €0.432 and €0.48 spot levels

A. Numerical details of the computations

Payoff	MC paths	Greeks method	# nodes	$b_{\min}$	$b_{\max}$	ϵ
TARF	1,000,000	Finite differences	3	0.0025	0.0025	18.1
TARF	300,000	Adaptive Chebyshev	7	0.0075	0.05	17.1
AUTOC	1,000,000	Finite differences	3	0.01	0.01	0.02
AUTOC	300,000	Adaptive Chebyshev	7	0.03	0.1	0.03

$b_{\min}$ and $b_{\max}$ are given by h/x_0 for FD Greeks, and by $a_{\min}/x_0$ and $a_{\max}/x_0$ for the adaptive cases (see (12)). The worst case on all evaluation dates is shown. The errors were computed over a grid of 200 points

3 Errors in L^∞ distance versus number of interpolation nodes for polynomial uniform and Chebyshev interpolants and their derivatives



The true values are given by exact Black formulas. Errors are evaluated over a grid of 1,000 points around the strike. The polynomial interpolators were built using both (a) the analytical pricer and (b) the Monte Carlo pricer. The maximum Monte Carlo error on the prices computed at nodal points is 2×10^{-6}. Results shown are for a call with strike $K = 1$, time to maturity $T = 0.1$, volatility $\sigma = 0.07$, risk-free rate $r = 0$ and interpolation domain $H = [0.94, 1.01]$

use a low number of interpolation nodes, where the original pricer must be called. In order to do this, it is essential to adapt the size of the interpolation domain to the positions, in space and time, of possible singularities in the price or its derivatives. By considering some particularly exotic test cases, we showed that our methodology is able to substantially reduce the computational burden of standard techniques (based on FDs) for gamma while improving its numerical stability in Monte Carlo simulations. This is due to the optimal convergence properties of Chebyshev interpolation.

The theory presented in this work is limited to the one-dimensional case. However, Chebyshev techniques can easily be extended to d dimensions with the use of Chebyshev tensors; appropriate tensor compression algorithms should be used to handle the high-dimensional cases (see Behnam & Trefethen 2017; Glau *et al* 2019; Townsend & Trefethen 2013; Zeron & Ruiz 2021b). This is not strictly necessary in the applications presented here; after all, Greeks are partial derivatives with respect to single parameters. Nevertheless, it would be interesting to explore the multi-dimensional extension,

which could offer the possibility of effectively computing cross-gammas as well as gammas.

Appendix A: convergence of polynomial interpolants with Monte Carlo errors

Here, we analyse the convergence properties of polynomial interpolations when the pricing function f is evaluated through a Monte Carlo estimate $\bar{f}$. In this situation, the values to be interpolated will be $\bar{f}(x_k) = f(x_k) + \varepsilon_k$, where $\varepsilon_k \sim \mathcal{N}(0, \varepsilon_{\mathrm{MC}}^{(k)})$ and $\varepsilon_{\mathrm{MC}}^{(k)}$ is the standard error of the estimate $\bar{f}$ at the node x_k. Assume that f satisfies the conditions of theorem 1. Then theorem 2.5 in Gaß *et al* (2018) implies the following stability result for the Chebyshev approximation $\bar{p}$, which interpolates the noisy points $\{\bar{f}(x_k)\}$:

$$\mathbb{E}[\|f - \bar{p}_{n-1}\|_\infty] \leqslant C\rho^{-n} + D_n \max_k \varepsilon_{\mathrm{MC}}^{(k)} \tag{16}$$

where C is some constant and:

$$D_n = \left(1 + \frac{2}{\pi}\log n\right)\sqrt{2\log(2n)}$$

This result is stated for prices and shows that the Monte Carlo impact on the approximation error is comparable with the worst Monte Carlo error on the interpolation nodes if n is low (as in our applications). We now investigate empirically whether this property extends to derivatives. Let us consider a call option under the Black model and study the convergence of polynomial interpolants, say $\bar{p}_n^{(m)}$, of price $\bar{f}$ and Greeks $\bar{f}^{(m)}$ to the true values f and $f^{(m)}$ as the number on interpolation nodes n increases. As a measure of approximation error, we consider the L^∞ distance between $\bar{p}_n^{(m)}$ and $f^{(m)}$, the latter being given by analytical Black formulas, for $m = 0, 1, 2$. We compare uniform polynomial interpolators, based on equidistant points in the interpolation interval, with Chebyshev approximation over a domain with a size comparable with that provided by (12). The results shown in figure 3 imply that Chebyshev interpolants converge exponentially up to the Monte Carlo error, and then they remain quite stable. Conversely, after an initial convergence, uniform interpolators diverge from true values (Trefethen 2020).

Appendix B: error analysis for polynomial approximation of Greeks

Let us consider a digital option under a Black model whose payoff is defined as:

$$\Pi(T) = \mathbf{1}_{\{S_T > K\}} \tag{17}$$

We repeat the same tests as those described in the 'Numerical investigations' section. Since the analytical results for Δ_{BS}, Γ_{BS} are available for Greeks in this simple case, we can define the errors in our numerical approximations as follows:

$$\left.\begin{aligned}\varepsilon_{n,\Delta}(S) &= |p'_{n-1}(S) - \Delta_{\mathrm{BS}}(S)|\\ \varepsilon_{n,\Gamma}(S) &= |p''_{n-1}(S) - \Gamma_{\mathrm{BS}}(S)|\end{aligned}\right\} \tag{18}$$

We aim to quantitatively measure the stability of the Chebyshev Greeks introduced earlier with respect to FDs. To this end, we compute the errors (18) for different levels of the spot S and provide some statistics. The results are summarised in table B. It turns out that three-point FDs display high variance or bias, depending on the choice of the bump. Both seven-point FDs and Chebyshev approximation significantly reduce the bias. The largest variance reduction is obtained with the adaptive Chebyshev method, especially for gamma, as is evident from the standard deviation and the maximum of the errors. ■

B. The averages, standard deviations and maximums of errors (18) for a digital call with strike $K = 1$, time to maturity $T = 0.1$, volatility $\sigma = 0.07$ and risk-free rate $r = 0$ under the Black model

Method	# nodes	Bump	Size	$\epsilon_\Delta^{\mathrm{avg}}$	$\epsilon_\Delta^{\mathrm{std}}$	$\epsilon_\Delta^{\max}$	$\epsilon_\Gamma^{\mathrm{avg}}$	$\epsilon_\Gamma^{\mathrm{std}}$	$\epsilon_\Gamma^{\max}$
FD	3	0.25%	—	0.04	0.05	0.3	30.3	39.8	275.4
FD	3	1%	—	0.17	0.17	0.65	6.6	8.6	43.9
FD	7	1%	—	0.03	0.03	0.17	5.19	6.75	40.3
Cheb.	7	—	3.32%	0.03	0.04	0.18	3.03	3.93	20.6

The results for 2,000 spot levels around the strike are shown, while 300,000 Monte Carlo paths were used to build the interpolators. Bumps h and domain size a are given as a percentage of the spot

At the time of writing, Andrea Maran worked as quantitative analyst at Exprivia Spa. Andrea Pallavicini is the head of equity, forex and commodity models at Intesa Sanpaolo; and Stefano Scoleri is a quantitative analyst at Be Management Consulting. All authors are based in Milan.
Email: andre.maran95@gmail.com, andrea.pallavicini@intesasanpaolo.com, s.scoleri@be-tse.it.

REFERENCES

Behnam H and LN Trefethen, 2017
Chebfun in three dimensions
SIAM Journal on Scientific Computing 39(5), pages C341–C363

Gaß M, K Glau, M Mahlstedt and M Mair, 2018
Chebyshev interpolation for parametric option pricing
Finance and Stochastics 22(3), pages 701–731

Glau K, P Herold, D Madan and C Pötz, 2019
The Chebyshev method for the implied volatility
Journal of Computational Finance 23(3), pages 1–31

Glau K, D Kressner and F Statti, 2020
Low-rank tensor approximation for Chebyshev interpolation in parametric option pricing
SIAM Journal on Financial Mathematics 11(3), pages 897–927

Maran A, A Pallavicini and S Scoleri, 2021
Chebyshev Greeks: smoothing gamma without bias
Preprint, arXiv:2106.12431

Savine A, 2018
Modern Computational Finance: AAD and Parallel Simulations
Wiley

Tataru G and T Fisher, 2010
Stochastic local volatility
Report, Quantitative Development Group, Bloomberg

Townsend A and LN Trefethen, 2013
An extension of Chebfun to two dimensions
SIAM Journal on Scientific Computing 35(6), pages C495–C518

Trefethen LN, 2020
Approximation Theory and Approximation Practice
Society for Industrial and Applied Mathematics

Welfert BD, 1997
Generation of pseudospectral differentiation matrices
SIAM Journal on Numerical Analysis 34(4), pages 1,640–1,657

Zeron M and I Ruiz, 2021a
Denting the FRTB IMA computational challenge via orthogonal Chebyshev sliding technique
Wilmott Magazine 111, pages 74–93

Zeron M and I Ruiz, 2021b
Tensoring dynamic sensitivities and dynamic initial margin
Risk April, pages 76–81