

## Optimal approximations of power-laws with exponentials: application to volatility models with long memory

Challet, Damien; Bochud, Thierry

Postprint / Postprint

Zeitschriftenartikel / journal article

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:

[www.peerproject.eu](http://www.peerproject.eu)

### Empfohlene Zitierung / Suggested Citation:

Challet, D., & Bochud, T. (2007). Optimal approximations of power-laws with exponentials: application to volatility models with long memory. *Quantitative Finance*, 7(6), 585-589. <https://doi.org/10.1080/14697680701278291>

### Nutzungsbedingungen:

Dieser Text wird unter dem "PEER Licence Agreement zur Verfügung" gestellt. Nähere Auskünfte zum PEER-Projekt finden Sie hier: <http://www.peerproject.eu>. Gewährt wird ein nicht exklusives, nicht übertragbares, persönliches und beschränktes Recht auf Nutzung dieses Dokuments. Dieses Dokument ist ausschließlich für den persönlichen, nicht-kommerziellen Gebrauch bestimmt. Auf sämtlichen Kopien dieses Dokuments müssen alle Urheberrechtshinweise und sonstigen Hinweise auf gesetzlichen Schutz beibehalten werden. Sie dürfen dieses Dokument nicht in irgendeiner Weise abändern, noch dürfen Sie dieses Dokument für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, aufführen, vertreiben oder anderweitig nutzen.

Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

**gesis**  
Leibniz-Institut  
für Sozialwissenschaften

### Terms of use:

This document is made available under the "PEER Licence Agreement". For more information regarding the PEER-project see: <http://www.peerproject.eu>. This document is solely intended for your personal, non-commercial use. All of the copies of this documents must retain all copyright information and other information regarding legal protection. You are not allowed to alter this document in any way, to copy it for public or commercial purposes, to exhibit the document in public, to perform, distribute or otherwise use the document in public.

By using this particular document, you accept the above-stated conditions of use.

Mitglied der  
**Leibniz**  
Leibniz-Gemeinschaft

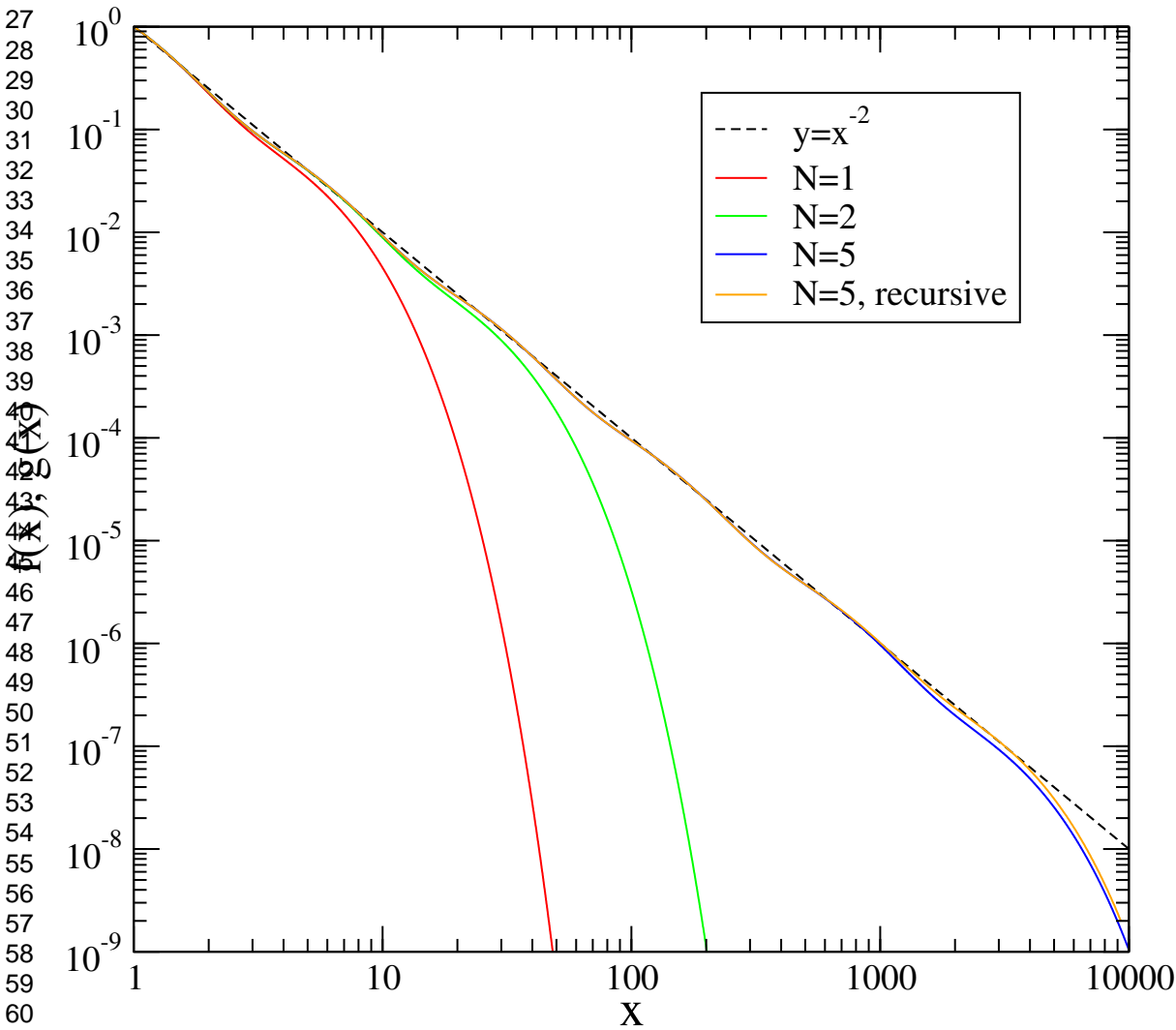
## Optimal approximations of power-laws with exponentials

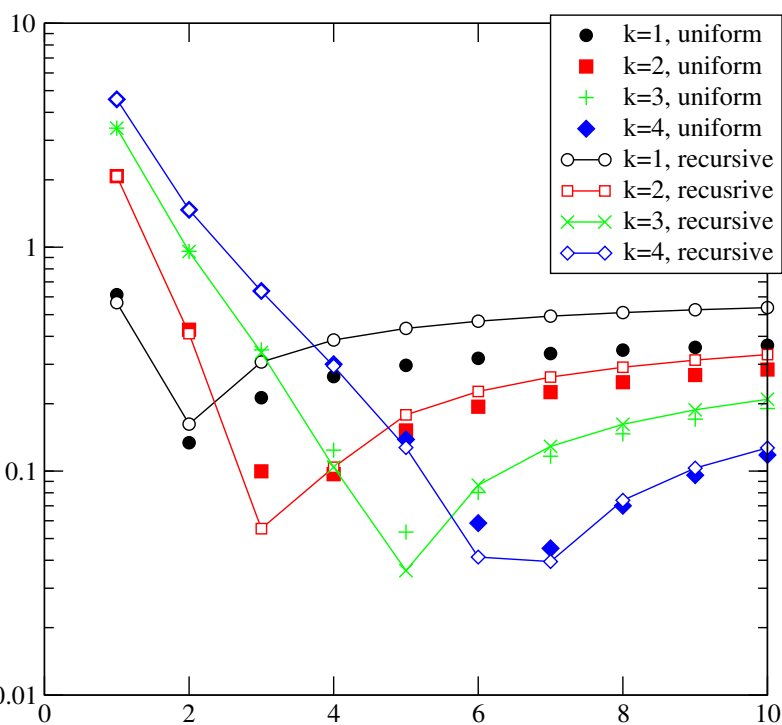
Journal:	<i>Quantitative Finance</i>
Manuscript ID:	RQUF-2006-0102.R2
Manuscript Category:	Research Paper
Date Submitted by the Author:	13-Feb-2007
Complete List of Authors:	Challet, Damien; OCIAM, Mathematical Institute Bochud, Thierry; Nestlé, Nestec
Keywords:	Stochastic Volatility, Time Series Analysis, Volatility Modelling, Exponential moving averages
JEL Code:	C63 - Computational Techniques < C6 - Mathematical Methods and Programming < C - Mathematical and Quantitative Methods, G10 - General < G1 - General Financial Markets < G - Financial Economics
Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.	
approxpw.bbl approxpw.tex	

SCHOLARONE™  
Manuscripts

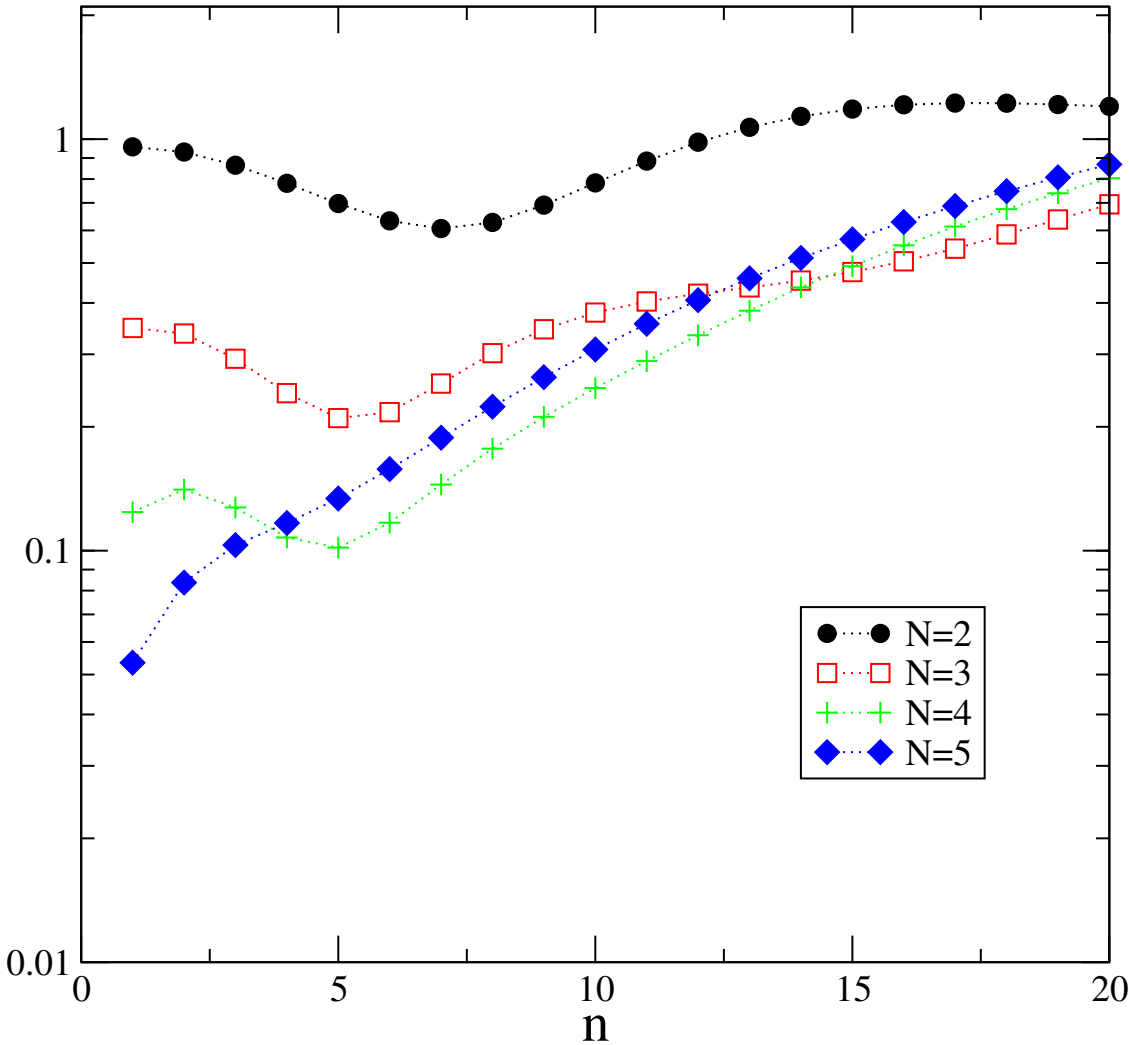
Only

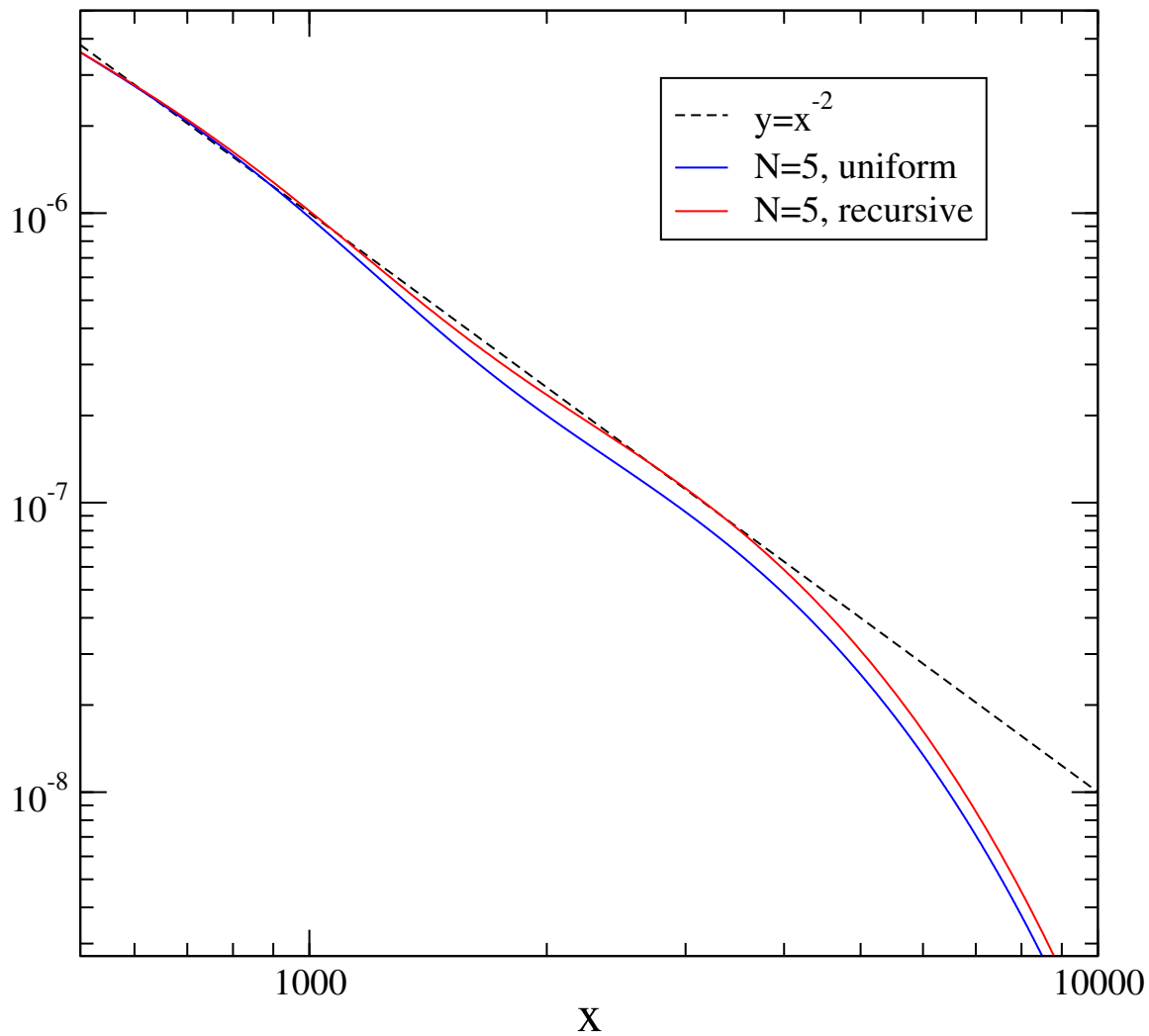
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60





1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60





# Optimal approximations of power-laws with exponentials: application to volatility models with long memory

Thierry Bochud

*Nestlé Capital Advisers, Av. Nestlé 55, 1800 Vevey, Switzerland\**

Damien Challet

*Nomura Centre for Quantitative Finance, Mathematical Institute,  
Oxford University, 24–29 St Giles', Oxford OX1 3LB, United Kingdom†*

(Dated: February 13, 2007)

We propose an explicit recursive method to approximate a power-law with a finite sum of weighted exponentials. Applications to moving averages with long memory are discussed in relationship with stochastic volatility models.

Exponential moving averages are widely used as tool for computing efficiently averages of time-changing quantities such as volatility and price. Their main advantage resides in their recursive definition that allows for easy numerical implementation, or remarkably simple models of stochastic volatility, such as GARCH [1]. Their use is however conceptually questionable when the process in question has long memory, as the volume and volatility do [2–4]. One should rather consider a power-law kernel; this requires however considerably more computing power as one must keep track of all the data points. Some authors approximate a power-law with a sum of exponentials in the literature, the record being held by Ref. [5], which uses 600 exponentials for 2 decades but notices that only a few have a significant contribution to the final function.

While the principle of economy should dictate to fit power-law-looking data with nothing else than a power-law (see for instance the controversy in the June 2001 issue of Quantitative Finance), computing real-time averages with a power-law kernel is much eased by the use of a sum of exponentials. Recent stochastic volatility models for instance use a sum of exponentials [6–8] (5, 12 and an infinity, respectively) with algebraically decreasing weights and algebraically increasing characteristic times, thereby respecting the long-memory of the volatility, which might explain in part their forecasting performance[24]. It is clear that only a handful of exponentials are required in order to approximate a power-law up to a given order of magnitude, as many practitioners are aware (see for instance [2, 6]). Since financial market data time series do not extend over an infinite period, such approximation will be good enough for application to financial time-correlations. How many exponentials should be used and with what parameters seem never discussed in the literature. Here, we aim to derive an explicit and new simple scheme that improves the often used approximation; in addition we show that the usual assumption of independent contribution from each exponential implies the existence of an optimal number of exponentials.

Let  $f(x) = x^{-\alpha}$  and  $g(x) = \sum_{i=0}^N g_i(x)$  where  $g_i(x) = w_i \exp(-\lambda_i x)$ . Assume that one would like to approximate  $f$  with  $g$  from  $x = 1$  to  $x = 10^k$ , that is, over  $k$  decades. The standard approach (see for instance [9]) consists in defining a cost function per decade that is the integral of some measure of the difference between  $f$  and  $g$ , i.e.

$$C = \frac{1}{k} \left( \int_1^{10^k} [-\alpha \log x - \log g(x)]^2 d \log x \right)^{1/2}, \quad (1)$$

and to minimize  $C$  with respect to  $w_i$  and  $\lambda_i$ , so as to obtain  $2(N + 1)$  coupled non-linear equations. Ad-hoc numerical methods have been investigated a long time ago, that solve the resulting set of equations by using the Gram-Schmidt orthonormalisation of exponentials [9]; working with an orthonormal basis is crucial in order to avoid the famously ill-posed problem of fitting a sum of exponentials [10, 11], most recently shown to be an example of sloppy model [12]. Our aim here is to obtain a sub-optimal (with respect to  $C$ ) but explicit set of  $w_i$  and  $\lambda_i$ .

The proposed method relies on a simple ansatz for  $w_i$  and  $\lambda_i$ . Instead of trying to solve an intricate set of non-linear equations, one observes that the nature of a power-law is to be scale-free, whereas an exponential has a well defined scale. Therefore, the role of each exponential is to approximate a given region of the  $k$  decades. In particular, one wishes that the  $i$ -th exponential approximates correctly  $f(x)$  at  $x_i = \beta^i$  where  $\beta > 1$  is a constant. This already suggests that  $\lambda_i \propto \beta^{-i}$ , which is both intuitive and well-known. Then one matches  $g$  to  $f$  and its first derivative  $g'$  to  $f'$  at  $x_i = \beta^i$ . However, once again, this would yield  $2(N + 1)$  coupled non-linear equations. The key observation is that, provided that  $\beta$  is large enough (see below), only  $g_i$  contributes significantly to  $g$  at  $x_i$ , i.e.  $g(x_i) \simeq g_i(x_i)$ . We therefore solve  $g_i(x_i) = f(x_i)$  and  $g'_i(x_i) = f'(x_i)$ , which gives

$$\lambda_i = \alpha \beta^{-i} \quad (2)$$

$$w_i = \left( \frac{e}{\beta^i} \right)^\alpha. \quad (3)$$

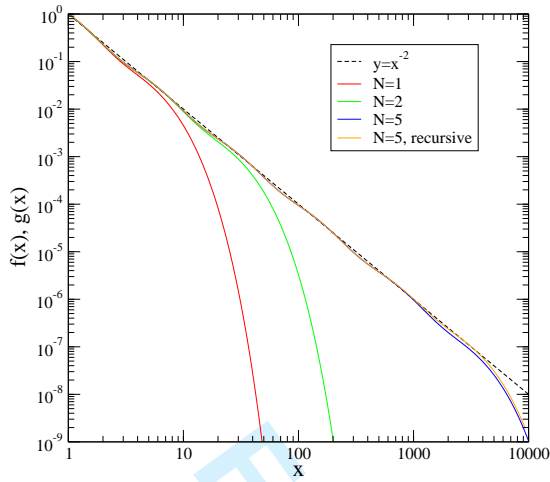


FIG. 1: Convergence of the approximation function  $g(x)$  to  $f(x)$  for the uniform ansatz with 2 (red line), 3 (green line) and 6 exponentials (blue line), and for the recursive ansatz with 6 exponentials (orange line);  $\alpha = 2$ ,  $\beta = 5$

However,  $g(x_i) > f(x_i)$  because the contribution of the exponentials other than the  $i$ -th cannot be totally ignored. Therefore, one must correct the above over-optimistic assumption by considering that  $g$  is a weighted sum of  $g_i(x)$

$$g(x) = \sum_{i=0}^N c_i \beta^{-i\alpha} \exp(\alpha) \exp(-\alpha/\beta^i x), \quad (4)$$

where  $\{c_i\}$  is a set of correction factors. The last step is to solve  $g(\beta^j) = f(\beta^j)$ , which is a set of  $N+1$  linear equations with variables  $c_i$ . The complexity of the problem has been greatly reduced. One can solve numerically this set of equations. In order to obtain explicit expressions for  $c_i$ , one has to resort to another approximation.

The simplest ansatz for  $c_i$  already gives a high degree of accuracy and is equivalent to the one currently in use in the literature. Taking uniform  $c_i = c$  given by  $1/c = \sum_{i=0}^N \beta^{-i\alpha} \exp(\alpha) \exp(-\alpha/\beta^i)$  ensures the equality  $g(1) = f(1)$ . With this choice the factor  $\exp(\alpha)$  disappears from  $g(x)$  and

$$g(x) = \left( \sum_{i=0}^N \beta^{-i\alpha} \exp(-\alpha/\beta^i) \right)^{-1} \sum_{i=0}^N \beta^{-i\alpha} \exp(-\alpha/\beta^i x) \quad (5)$$

Fig. 1 shows how the approximation works for increasing  $N$ : each additional exponential extends the range that is well approximated by a factor  $\beta$ . The value of  $\beta$  was chosen large enough so as to emphasise the oscillations of  $g(x)$  at each  $\beta^j$ . The uniform ansatz implies that while  $g(1) = f(1) = 1$ ,  $g(\beta^j) > f(\beta^j)$  for  $0 < j < N$  since the contribution of each  $g_k$  is asymmetric with respect to  $\beta^j$ ; when  $j = N$ , since there are no additional exponentials from  $i > j$  to contribute to  $g$ ,  $g(\beta^N) < \beta^{-\alpha N}$  (see

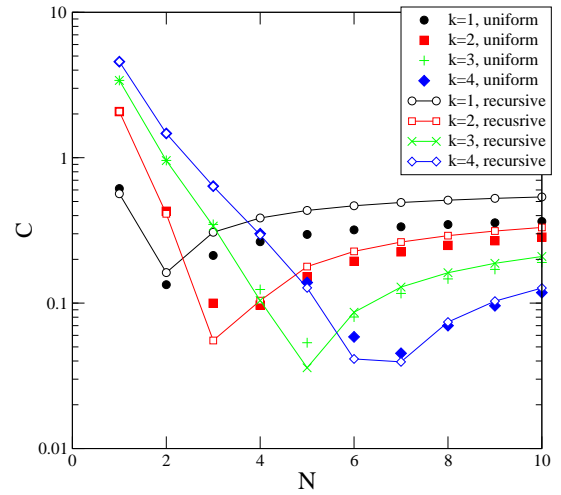


FIG. 2: Error per decade  $C$  as a function of  $N$  for various  $k$ ;  $\alpha = 2$ ;  $\beta = 10^{k/N}$  for the uniform and recursive ansatz (full and empty symbols respectively). Lines are for eye guidance only.

Fig. 1). This problem is of course negligible when a very large number of exponentials is used; however, since our aim is to use as few exponentials as possible it needs to be addressed.

The parameter  $\beta$  tunes how much of a decade is approximated by a single exponential. When  $k$  and  $N$  are fixed, it is sensible to take  $\beta^N = 10^k$ . The cost function  $C$  is plotted in Fig. 2 as a function of  $N$  at fixed  $k$  for several values of  $k$ . For small  $N$ ,  $C$  decreases exponentially as a function  $N$ . Then, strikingly,  $C$  has a minimum at  $N_m(k)$  and increases slightly before stabilising; the smaller  $\alpha$ , the smaller the subsequent increase. One would have naively expected that  $C$  decreased monotonically as a function of  $N$ ; however, since  $\beta$  decreases when  $N$  increases at fixed  $k$ , the assumption that the exponentials give independent contributions to  $g$  is not valid any more at  $N \simeq N_m$ , and becomes clearly incorrect when  $N > N_m$ . The consequence is that  $g(x)$  becomes too large except at  $x = 1$ . This is not problematic, however, since in practice, one prefers large  $\beta$  to small ones, so as to use as few exponentials as possible. As expected,  $N_m$  increases linearly with  $k$ , implying that for  $\alpha = 2$ , the optimal  $N = N_m(k) \simeq 1.7k$ , or equivalently  $\beta \simeq 10^{1/1.7} \simeq 3.87$ . Another feature of this figure is that  $C(N_m(k))$  decreases as function of  $k$ : this due to the vanishing influence of the deviation caused by the downwards shift of the last exponential.

It is possible to improve the precision of the approximation for  $N < N_m$  by modifying the scale of  $x$ , or equivalently by taking into account derivatives of  $g$  of higher orders. The second order yields  $\lambda_i = \sqrt{\alpha(\alpha+1)}\beta^{-i}$ . From the conditions on the first derivatives and on the equality of functions,  $w_i \propto \beta^{-\alpha i} \exp(\sqrt{\alpha(\alpha+1)})$ . This reasoning can be extended to match the derivatives up



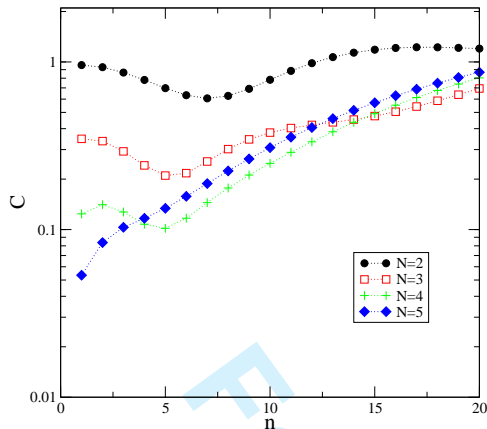


FIG. 3: Error per decade  $C$  as a function of  $n$  for various  $N \leq N_m = 5$ ;  $k = 3$ ,  $\alpha = 2$ ;  $\beta = 10^{k/N}$ . Dotted lines are for eye guidance only.

to order  $n$ , resulting in

$$g(x) = \left( \sum_{i=0}^N \beta^{-i\alpha} \exp(-\mu/\beta^i) \right)^{-1} \sum_{i=0}^N \beta^{-i\alpha} \exp(-\mu x/\beta^i) \quad (6)$$

with

$$\mu = \left[ \prod_{j=0}^{n-1} (\alpha + j) \right]^{\frac{1}{n}} = \left[ \frac{\Gamma(\alpha + n)}{\Gamma(\alpha)} \right]^{\frac{1}{n}} \quad (7)$$

Since  $\mu$  does not depend on  $i$  it modifies the scale of  $x$ , which can be used to adjust the position in log-space of  $g$  relative to  $f$ . For large  $n$ ,  $\mu \simeq (n + \alpha - 1)/e$ , therefore shifting  $g(x)$  to larger  $x$ . According to Fig. 3, as long as  $N < N_m$ , there is an optimal  $n$ . This comes from the fact that  $g(\beta^N) < f(\beta^N)$ : it is more advantageous to shift  $x$  to larger values so as to avoid the too small value of  $g$  at  $\beta^N$ . It also emphasises once again the need to solve the problem of the last exponential.

The solution comes from a close examination of Fig. 1: the first exponentials do not contribute much to the value of  $g(\beta^N)$  for  $N$  not too small. This suggests that the contribution of  $g_i(\beta^j)$  to  $g(\beta^j)$  can be neglected if  $i < j$ . As a consequence,  $g(\beta^N) \simeq g_N(\beta^N)$ , and  $c_N = 1$ . Thus

$$c_{N-1} = 1 - \beta^{-\alpha} e^{\alpha(1-1/\beta)}. \quad (8)$$

More generally,

$$c_{N-k} = 1 - \sum_{i=0}^{k-1} c_{N-i} \beta^{-\alpha(k-i)} e^{\alpha(1-\beta^{i-k})} \quad (9)$$

$c_0$  is the same with both ansätze, since there is no exponential on the left of  $\beta^0$ . Table I gives an example set of

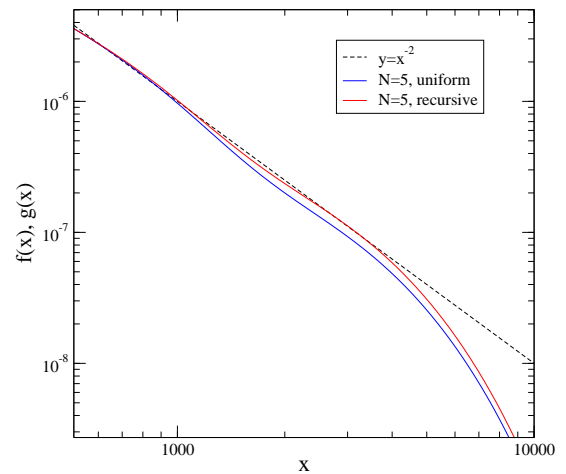


FIG. 4: Zoom of Fig 1 on the last two exponentials.  $\alpha = 2$ ;  $\beta = 4$

$c_{N-k}$ . It is noticeable that  $c_{N-k}$  display oscillations which are damped as  $k$  increases: since  $c_N = 1$  is large in order to compensate for the absence of further exponentials,  $c_{N-1}$  must be smaller than  $c_0$ ; next,  $c_{N-2}$  will be slightly larger than  $c_0$  so as to satisfy  $g(\beta^{N-1}) = f(\beta^{N-1})$ , etc.

TABLE I: Correction coefficients given by the recursive ansatz.  $\alpha = 2$ ,  $N = 8$ ,  $\beta = 4$

$k$	0	1	2	3	4	5	6	7	8
$c_{N-k}$	1.000	0.720	0.773	0.763	0.765	0.765	0.765	0.765	0.765

The recursive ansatz always gives a better result than the uniform one, as it ensures that  $g(\beta^i)$  is closer to  $f(\beta^i)$  for all  $i$ , and particularly for large  $i$ ;  $g$  approximates  $f$  remarkably well at  $x_i = \beta^i$  provided that  $\beta$  is not too small. The differences are most perceptible for  $x \simeq \beta^N$ , where the recursive scheme gives a much better approximation (see Fig. 4), which explains why it is most advantageous for  $k \leq 4$  where it decreases  $C$ , at  $N_m$  by a factor 2 for  $k = 2$  and 1.5 for  $k = 3$ ; larger  $k$ , hence larger  $N_m$ , will not bring much improvement since the weight of the discrepancy caused by the uniform ansatz at  $\beta^N$  decreases. Improving the precision further is possible by taking more exponentials from the left hand side of  $\beta^j$  into account in the calculus of  $c_i$  at the price of heavier and probably non-explicit computations. Finally, if solving the full set of linear equations for  $c_i$  does not give enough precision, the remaining possibility is to minimise numerically  $C$  [9].

#### Application to volatility models with long memory

Volatility modeling and forecasting is carried out in discrete time by autoregressive models such as GARCH

and its numerous variants (see for instance [13, 14]). They are all based on a recursive moving average of the volatility  $\sigma^2$

$$\sigma^2(t) = \sigma^2(t)\Lambda + (1 - \Lambda)v(t - \delta) \quad (10)$$

where  $v(t)$  is some measure of the instantaneous volatility (e.g. daily volatility) over  $\delta t$  units of time, and  $\Lambda = e^{-\lambda}$  is the memory. RiskMetrics recommends  $\Lambda_1 = 0.98$  or  $\Lambda_2 = 0.94$ . While this is an efficient way of computing an average, it implicitly assumes a choice of a single time scale  $1/|\ln \Lambda| \simeq 1/(1 - \Lambda)$  for  $\Lambda$  close to 1.

Unfortunately, the volatility is a process with no obvious time scale, as its autocorrelation function decreases slowly; fitting it with a power-law gives an exponent  $\nu \simeq 0.3$  [2, 3]. In other words, any choice of  $\Lambda$  is a compromise between smoothness and reactivity. To our knowledge, the first paper to use a power-law kernel for measuring volatilities is from the Olsen group [15]. One possible reason of this particular functional form of the volatility memory is that the market is made of heterogeneous participants [16]. For instance the variety of time-scales of people taking part into financial markets is obvious to any practitioner, hence a choice of a single  $\Lambda$  selects the categories of traders that the resulting average volatility incorporates. Direct measure on high-frequency data revealed five characteristic time scales [8].

Extending GARCH to include multiple time scales can be done in a simple way [25] by a weighted sum of volatilities at increasing time scales  $\delta t_k$ ,  $k = 1, \dots, k_{max}$ , denoted by  $\sigma_k^2$

$$\sigma^2(t) = \sum_{k=1}^{k_{max}} w_k \sigma_k^2(t) \quad (11)$$

where  $\sigma_k^2$  is the estimated volatility at time scale  $\delta t_k$ , whose dynamics is given by

$$\sigma_k^2(t) = \Lambda_k \sigma_k^2(t - \delta t_k) + (1 - \Lambda_k) r_{\delta t_k}(t) \quad (12)$$

$$r_{\delta t_k}(t) = \frac{\log p(t) - \log p(t - \delta t_k)}{\sqrt{\tau_k}}. \quad (13)$$

Several works considered similar setups [6–8, 17, 18]. Because of the power-law decay of the auto-correlation of the volatility, the weights  $w_k$  should decrease algebraically. Indeed, Ref [8] fitted such a stochastic volatility model with five time-scales and found them to be 0.18, 1.4, 2.8, 7, 28 business days, with respective weights of 0.39, 0.20, 0.18, 0.12, 0.11; the time scales span about 2.2 decades, and the weights decreases algebraically as the timescale grows with an exponent of about  $\alpha = 0.3$ . Other work considered  $\alpha = 2$  [6, 15]. Generally speaking,  $2\alpha - 2 = \nu$ , which gives  $\alpha = 1.15$  if  $\nu = 0.3$  (see e.g. [7]). Using the results from the previous section, for  $\alpha = 1.15$ , five exponentials approximate best three decades with corrections  $\vec{c} = (0.704, 0.702, 0.714, 0.647, 1)$ . As stated

in the introduction, the number of timescales to take into account in this kind of modeling varies from paper to paper: 5 [8], 12 [6] and an infinity [7] for instance (in the latter case, this was motivated by mathematical tractability). Our first result implies that the optimal number of time scales (hence of exponentials) is about 1.7 per decade if  $\alpha = 2$ ; if one uses 5-minute data and if one contents oneself with a cut-off of 6 months, as in Ref. [6], there are 4.7 decades, which are optimally approximated by 8 exponentials. One sees therefore that long-memory generalisations of GARCH do not require many exponentials in order to implement at relatively long-term cut-off.

However, there is still a computational problem with this approach: according to Eq (13), one must keep the returns over a time horizon equal to the longest time scale; this is barely economical and defeats the initial aim of the approximation. An ingenious solution resides in nested exponential moving averages. Indeed, applying  $n$  times such an operator to a price return time series is equivalent to averaging the last  $n$  price returns [19]. Since by definition a price return over a time horizon of  $n\delta t$  is nothing else but the sum of  $n$  returns at time scale  $\delta t$ , one can abstain from keeping any price return in memory, at the price of using more computing power, which is not a problem nowadays; even better, the first section of this paper explicitly specifies how to minimise the number of exponentials so as to save memory and processor time.

## CONCLUSIONS

We have provided a simple method to use efficiently a sum of weighted exponentials as a parsimonious approximation of a power-law with any exponent. In particular, we have shown the existence of an optimal number of exponentials when one neglects the contribution of some exponentials in the determination of the coefficients. The recursive ansatz is probably precise enough for most applications. This method can be directly applied to determine the minimal number of components in long-memory volatility forecasting.

We thank Gilles Zumbach for useful discussions.

---

\* Electronic address: [thierry.bochud@nestle.com](mailto:thierry.bochud@nestle.com)

† Electronic address: [challet@maths.ox.ac.uk](mailto:challet@maths.ox.ac.uk)

[1] R. Engle, *Econometrica* **50**, 987 (1982).

[2] M. M. Dacorogna, R. Gencay, U. A. Müller, R. B. Olsen, and O. V. Pictet, *An Introduction to High-Frequency Finance* (Academic Press, London, 2001).

[3] J.-P. Bouchaud and M. Potters, *Theory of Financial Risks* (Cambridge University Press, Cambridge, 2000).

[4] P. Gopikrisnan, V. Plerou, X. Gabaix, and H. Stanley, *Phys. Rev. E* **p. R4493** (2000).

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

[5] B. Quian and R. A. Goldstein, *PROTEINS: Structure, Functions, and Genetics* **45**, 102 (2001).

[6] G. Zumbach, *Quant. Fin.* **4**, 70 (2004).

[7] L. Borland and J.-P. Bouchaud (2005), preprint physics/0507073.

[8] P. Lynch and G. Zumbach, *Quantitative Finance* **3**, 320 (2003).

[9] T. Svensson, *IEEE Trans. Circuit Theory* p. 142 (1973).

[10] C. Lanczos, *Applied Analysis* (Prentice-Hall, 1956).

[11] A. van den Bos and J. H. Swarte, *IEEE Trans. on Sign. Proc.* **41**, 313 (1993).

[12] J. J. Water, F. P. Casey, R. N. Gutenjunt, K. S. Brown, C. R. Myers, P. W. Brouwer, V. Elser, and J. P. Sethna, *Phys. Rev. Lett.* **97**, 150601 (2006).

[13] T. Bollerslev, R. Y. Chou, and K. F. Kroner, *J of Econometrics* **52**, 5 (1992).

[14] A. K. Bera and M. L. Higgins, *J. of Economic Surveys* **7**, 305 (1993).

[15] G. O. Zumbach, M. M. Dacorogna, J. L. Olsen, and R. B. Olsen, *International Journal of Theoretical and Applied Finance* **3**, 347 (2000).

[16] R. B. Olsen, M. M. Dacorogna, U. A. Müller, and O. V. Pictet, Tech. Rep. BRO.1992-09-07, Olsen (1992).

[17] C. Granger and Z. Ding, *J. of Econometrics* **73**, 61 (1996).

[18] Z. Ding and C. Granger, *J. of Econometrics* **73**, 185 (1996).

[19] G. Zumbach and U. Müller, *Int. J. Th. and Appl. Fin.* **4**, 147 (2001).

[20] L. Calvet and A. Fisher, *J. of Econometrics* **105**, 27 (2001).

[21] J. Muzy, J. Delour, and E. Bacry, *Eur. J. Phys B* **3**, 537 (2000).

[22] T. Lux (2003), working paper 2003-13.

[23] R. T. Baillie, T. Bollerslev, and H. O. Mikkelsen, *J. of Econometrics* **74**, 3 (1996).

[24] Models with long memory (see also [20, 21]) appear to be intrinsically better for forecasting [22].

[25] The FIGARCH family [23] is a possible route with some non-trivial implementation problems (see e.g. [6])