COMPUTATIONAL METHODS IN APPLIED MATHEMATICS, Vol. 8 (2008), No. 2, pp. 187–200

EFFICIENT AND PRECISE COMPUTATION OF CONVOLUTIONS: APPLYING FFT TO HEAVY TAILED DISTRIBUTIONS

P. SCHALLER[1] AND G. TEMNOV[2]

Abstract — In this paper we estimate and analyze the errors associated with the use of the discrete (fast) Fourier transformation for the numerical calculation of convolutions. We suggest and compare methods to reduce these errors without loosing the computational efficiency of the calculation scheme. A typical field of application of our findings is the calculation of aggregate loss distributions for, e.g., losses from insurance cases or operational risk losses in the finance industry.

2000 Mathematics Subject Classification: 42A38, 44A35, 60E10, 65R10, 90–08.

Keywords: compound probability distributions, characteristic functions; discrete Fourier transform, exponential window.

1. Introduction: the problem

Loss aggregation is a well-known problem in insurance. Usually, one needs to compute with sufficient precision the cumulative distribution of the losses aggregated for some fixed period (normally, one year). The task reduces to the problem of summing up a random number of random variables. In the present work we address this classical problem, supposing that r.v.'s having the sense of single losses are iid with a common heavy tailed distribution, and that the counting process for losses is characterized by a generating function. Specifically, the variable we are interested in is

$$S_N = \sum_{k=1}^{N} X_k, \quad \mathbf{P}(X_k < x) =: F_X(x), \quad \mathbf{P}(N = k) =: \alpha_k \tag{1.1}$$

where r.v.'s X_k are assumed to be iid, and N is the number of events within a selected period, generated by a process $N(t)$ of occurences, α_k are the coefficients in the power series expansion of the generating function $h(x) = \sum_k \alpha_k x^k$.

Examples are provided by the Poisson or mixed Poisson process, for which one has $\alpha_k = E_\lambda[(\lambda t)^k e^{-\lambda t}/k!]$, where the intensity λ is constant in the Poisson case, and is a r.v. having gamma distribution in the mixed Poisson case. In the first case, the expectation value $E_\lambda[.]$ is trivial and the generating function is given by $h(x) = e^{-\lambda[1-x]}$. In the second case the loss counting process follows a negative binomial distribution. The corresponding

[1] *Bank Austria, Strategic Risk Managemnet department*, Julius Tandler Platz, A-1090 Vienna, Austria. E-mail: peter@ca-risc.co.at

[2] *Vienna University of Technology, Institute for Mathematical Methods in Economics*, Wiedner Hauptstraße 8-10/105-1, A-1040 Vienna, Austria. E-mail: gtemnov@fam.tuwien.ac.at

generating function reads $h(x) = 1/[1+\beta-\beta x]^r$, where β and r are the parameters of the gamma distribution associated with the intensity.

The use of characteristic functions for the calculation of compound distributions is a widely applied approach. Using the notation

$$\widehat{f}(u) = \int\limits_{-\infty}^{\infty} e^{iux}\, dF(x)$$

for the characteristic function (chf) of a probability distribution F, we can write down the well-known representation of the chf of the aggregate distribution

$$\widehat{f}_{S_N}(u) = h(\widehat{f}_X(u)). \tag{1.2}$$

In terms of probability density functions the representation turns into

$$f_{S_N}(x) = \sum_k \alpha_k f_X^{*k}(x) \tag{1.3}$$

with the upper index $*k$ denoting the k-fold convolution.

Clearly, direct calculations with (1.3) for arbitrary f_X are not feasible. With the help of chf's, the task of the calculation of the aggregate loss distribution is reduced to the following operations:

- calculate the chf $\widehat{f}_X$ of a single-loss distribution,
- calculate the aggregate loss chf $\widehat{f}_{S_N}$ by (1.2),
- by inverting $\widehat{f}_{S_N}$ calculate the pdf f_{S_N} and the cdf F_{S_N} of the aggregate loss.

The discrete Fourier transform (DFT) is a well-known efficient method for calculating chf's. The so-called fast Fourier transform (FFT), as a natural algorithm based on the DFT, is probably the most powerful application for calculating chf's. It nevertheless produces some typical errors, which can appear crucial for the final result, especially when one works with heavy tailed distributions and higher quantiles of the aggregate loss distribution are of interest. In the following paragraph we analyze the source of errors in the FFT.

Note that the use of the discrete Fourier transform (FFT in particular) and its error analysis have been amply studied in the literature since the middle of the XX century (see, e.g., [4] or [1] for detailed overview), but the attention was mostly focused on the application in harmonic analysis, images and signal processing. Our aim is to concentrate on the application to the task of precise calculation of the sum of random variables, which may be rather heavy tailed, and to investigate the efficiency of the FFT within the framework of this problem.

To our knowledge, there are no clear suggestions in the literature on operational risk and other branches of risk theory to use exponential windows with adjusted parameters, which we investigate here. Usually (see, e.g., [10]) authors suggest to use the FFT with padding by zeros, or to select the right endpoint of the distribution high enough, which are not the most efficient techniques to handle heavy tailed distributions, especially with infinite expectations. The search for optimal techniques within the framework of this particular task motivated our research.

Let us note that methodologies similar to the one that we use have been implemented for various problems of applied mathematics. A short review on similar tasks connected with financial mathematics can be found in Section 6 in this paper, where we also compare them with our task.

2. Discrete convolution, errors classification, anti-aliasing

2.1. Convolution theorem in the discrete case. Consider two functions $f(x)$ and $g(x)$, both having positive support. Together with a uniform grid $x_k = k/N,\ k = 0, \ldots, N-1$ they define vectors $\{f_k\}$ and $\{g_k\}$ via $f_k = f(x_k)$, $g_k = g(x_k)$. The approximate values of the convolution $c := f * g$ can then be expressed by

$$c_n = \frac{1}{N}\sum_{k=0}^{n} f_k g_{n-k}. \tag{2.1}$$

The approximate chf is the result of the application of the discrete Fourier transform[1]:

$$\widehat{f}_l = \sum_{k=0}^{N-1} e^{2\pi ikl/N} f_k. \tag{2.2}$$

The following chain of equalities is in fact the well-known proof of the convolution theorem in the discrete case:

$$\frac{1}{N}\sum_{l=0}^{N-1} \widehat{f}_l\widehat{g}_l e^{-2i\pi ln/N} = \frac{1}{N}\sum_{l=0}^{N-1}\left[\sum_{k=0}^{N-1} f_k\, e^{2i\pi kl/N}\right]\widehat{g}_l\, e^{-2i\pi ln/N} =$$

$$\sum_{k=0}^{N-1} f_k\left(\frac{1}{N}\sum_{l=0}^{N-1}\widehat{g}_l\, e^{-2i\pi l(n-k)/N}\right) = \sum_{k=0}^{N-1} f_k g_{n-k}. \tag{2.3}$$

Note that the range of summation in (2.3) is different from the one in (2.1). Indeed, it implies the appearance of negative vector indices $n-k$ in some of the summands. This can be understood as a consequence of the periodicity of the Fourier transformation: For $n-k < 0$, g_{n-k} is to be identified with g_{n-k+N}, as is most easily seen by applying the identity

$$e^{-2i\pi l(n-k)/N} = e^{-2i\pi l(n-k+N)/N} \tag{2.4}$$

to the left hand side of the last equation in (2.3). To avoid ambiguities we may wish to restrict our notation to indices in the range $0, \ldots, N-1$. We may then rewrite (2.3) as

$$\sum_{l=1}^{N-1} \widehat{f}_l\widehat{g}_l\, e^{2i\pi ln/N} = c_n + \sum_{k=n}^{N-1} f_k g_{N+n-k}. \tag{2.5}$$

The quantity

$$A_n := \sum_{k=n}^{N-1} f_k g_{N+n-k} \tag{2.6}$$

is called *aliasing error* in the following. With the restriction to positive indices it is given by a sum over terms of the type $f_k g_l$ with $k + l = n + N \geqslant N$.

[1] The usual notation is $\sum \cdot\, e^{-2\pi ikl/N}$ for the direct DFT and $\sum \cdot\, e^{2\pi ikl/N}$ for the inverse DFT. We will use these notations vice versa just for convenience, as there is no difference in this case which of the transforms is applied first.

2.2. Error classification.

2.2.1. Aliasing error. Let us try to generalize this observation to the n-fold convolution in order to find an upper bound for the aliasing error. The n-fold convolution of some probability distribution with pdf f can be represented by

$$F^{*n}(x) = \int\limits_{\{x_i \,|\, \sum x_i \leqslant x\}} \prod_i f(x_i)\, dx_i. \tag{2.7}$$

Assume that we calculate the n–fold convolution of the d.f. $F(x)$ via the DFT on a grid with a right endpoint x_m. Let us analyze the aliasing error for $F^{*n}(x)$, which is A_n. By a straightforward generalization of the above considerations for the two–fold convolution, it is clear that A_n consists of a (discretized) integral with the same integrand as in the r.h.s. of (2.7), however integration is over a region with $\sum x_i \geqslant x_m$. As the integrand is positive, we have

$$A_n \leqslant \int\limits_{\{x_i \,|\, \sum x_i > x_m\}} \prod_i f(x_i) dx_i = 1 - F^{*n}(x_m). \tag{2.8}$$

2.2.2. Discretization error. Given the values of some monotonic (non surjective) function $F_0 : [0, \infty] \to \mathbb{R}$ (e.g., the cumulated probabilities of some distribution) on a uniform grid $\{x_k\}$, $k \in \mathbb{N}_0$, $x_k < x_{k+1}$ with step size $x_{k+1} - x_k = \Delta_x$, the value of the function on some point x is always between the values on the neighboring grid points:

$$x_k \leqslant x \leqslant x_{k+1} \quad \Rightarrow \quad F_0(x_k) \leqslant F_0(x) \leqslant F_0(x_{k+1}) \tag{2.9}$$

Define by F_+ and F_- the step functions given by

$$F_-(x) = F_0(x_k)\,, \quad F_+(x) = F_0(x_{k+1}) \tag{2.10}$$

with x_k as in (2.9). Then these step functions provide lower and upper bounds for $F_0(x)$

$$F_-(x) \leqslant F_0(x) \leqslant F_+(x). \tag{2.11}$$

Now let F_0 be the cumulated distribution function of some random variable X . Then random variables $X_\pm$ with cdf's $F_\pm$ are given by $X_\pm = F_\pm^{-1}(F_0(X))$ (Here the inverse of the step functions $F_\pm$ is defined, as usual, via $F_\pm^{-1}(q) = \sup\{x \,|\, F_\pm(x) < q\}$). Due to the monotonicity of the functions involved we have $X_+ \leqslant X \leqslant X_-$ and thus for the cdf's of the n-fold convolutions we have

$$F_-^{*n} \leqslant F_0^{*n} \leqslant F_+^{*n}. \tag{2.12}$$

Moreover, this inequality holds for any distribution function F, whose values on the grid are identical with the values of F_0:

$$F(x_k) = F_0(x_k) \;\; \forall k \in \mathbb{N}_0 \quad \Rightarrow \quad F_-^{*n} \leqslant F^{*n} \leqslant F_+^{*n}. \tag{2.13}$$

As F_- and F_+ differ by a translation of the argument by Δ_x, a random variable with distribution F_- is obtained from X_+ via a constant shift $X_- = X_+ + \Delta_x$. In the n-fold convolution the shift parameter is multiplied by n. So the quantiles of F_-^{*n} and F_+^{*n} differ by $n\Delta_x$. Due to the monotonicity of the distribution functions the inequality (2.13) transfers to the quantile functions and we have

$$(F_+^{*n})^{-1} \leqslant (F^{*n})^{-1} \leqslant (F_+^{*n})^{-1} + n\Delta_x. \tag{2.14}$$

So the discretization causes an error in the quantile function of the n-fold convolution, which is limited by $n\Delta_x$.

Note that this error is not specific to the DFT. It applies to any numerical convolution technique which is based on a discretization on a uniform grid.

From the error in the quantile we may calculate the error in the distribution function by multiplying it with the corresponding density

$$\Delta_{F^{*n}}(x) = n\Delta_x \, \partial F^{*n}(x)/\partial x \,. \tag{2.15}$$

2.2.3. Numerical integration error. One should mention another source of numerical inaccuracy, which stems from the discrete nature of the algorithm. Indeed, calculating the DFT we replace the integral $\int_0^{x_m} e^{itx} f(x)\, dx$ by the approximation $\sum_{k \geqslant 1} e^{itx_k} f_k \cdot (x_k - x_{k-1}) = \Delta x \sum_k e^{itx_k} f_k$.

Consider now the case where the FFT is applied to the set representing the values of the probability density function and the analytical form of not only the density, but also of the corresponding cdf F is given. Then as an input for the FFT we can use, instead of the density values $\{f_k\}$, the sequence $\{\Delta F_k\} = F(x_k) - F(x_{k-1})$. Clearly, operating with such values one obtains a precise value of the chf. The corresponding estimate of the convolution cdf will also be precise (up to the errors described above, of course).

2.3. Techniques for precision improvement

2.3.1. Padding by zeros. This is, perhaps, the simplest tool for the reduction of the aliasing error. In the case of the two-fold convolution the aliasing error could be completely eliminated in the following way. If one considers the set $\{\widetilde{f}_k\}$, $k := 0, \ldots, 2N-1$, where the first N elements are equal to the corresponding $\{f_k\}$, and the remaining N elements are equal to zero, then the first N elements of the set $\widetilde{c}_n \equiv \Delta x \sum_{j=0}^{2N-1} \widetilde{f}_j \widetilde{f}_{n-j} = \Delta x \sum_{j=0}^{N-1} \widetilde{f}_j \widetilde{f}_{n-j}$ are equal to the values of the corresponding elements of $\{c_k\}$. This is most easily seen by noting that $\widetilde{f}_{-1} \ldots \widetilde{f}_{-N}$ is zero, as the Fourier transformation treats $\{\widetilde{f}_k\}$ as a periodic quantity with period length $2N$. For the n-fold convolution we would have to use $\{\widetilde{f}_k\}$, $k := 0, \ldots, (nN-1)$ with $\widetilde{f}_k = 0$ for $k > N-1$ to eliminate completely the aliasing error. However, an effective error reduction will usually be reached by increasing the grid size by a factor of 2 or 4. The method has an obvious drawback: In order to calculate the density after convolution up to some point x_{N-1} we need a multiple of N grid points and this will slow down the calculation. Alternatively, we could increase the step size at the price of increasing the discretization error.

2.3.2. Increasing the speed of tending to zero. A natural way to reduce the aliasing error in the FFT algorithm is to apply a transformation which decreases the function in use at the right end of the grid and which can be easily inverted right after the convolution procedure. Probably the most simple of such transformations is the exponential one. Obviously, if $f_\tau(x) = f(x) \cdot e^{-x/\tau}$, then

$$(f * g)(t) = e^{x/\tau} f_\tau * g_\tau. \tag{2.16}$$

We shall call this transformation the *exponential window* following the terminology used mainly in signal analysis theory. Exponential windows are widely used, e.g., in visual signal processing. Some works to mention on the use of exponential windows: [3, 9].

Within the scope of our work the point about the exponential window we are interested in is the adjustment of the parameter τ. As we will see soon, this parameter plays a dual role — it reduces the aliasing error and at the same time influences the numerical precision of the whole procedure.

Let us note again at this point that increasing the speed of tending a function to zero is a common method in Fourier analysis. It has been mainly used to limit the domain of the integration of a ch.f dealing with the Fourier-inversion problem. We make a further note on these issues in Section 6. In our investigation, we are interested in increasing the speed with which a prototype of a chf tends to zero as we tend to reduce the aliasing error, keeping all other errors associated with the FFT under control.

3. Analysis of the errors

Aliasing error revisited. When the exponential window $e^{-x/\tau}$ is applied, each contribution to the aliasing error is multiplied by a factor $e^{-\alpha x_m/\tau}$ for some $\alpha \in \mathbb{N}$.
As $e^{-\alpha x_m/\tau} \leqslant e^{-x_m/\tau}$

$$A_n^\tau \leqslant e^{-x_m/\tau}(1 - F^{*n}(x_m)). \tag{3.1}$$

From the example with n–fold convolution we pass to the more general case, namely to the generating function

$$h(t) = \sum_k \mathbf{P}(N = k)t^k = \sum_k a_k t^k, \tag{3.2}$$

so that the compound probability d.f. is

$$F^h(x) = \sum_k a_k F^{*k}(x). \tag{3.3}$$

Hence, the error of the whole procedure turns into

$$A_\tau^h = \sum_k a_k A_k^\tau \leqslant \left[\sum_k a_k \left(1 - F^{*k}(x_m)\right)\right] e^{-x_m/\tau}. \tag{3.4}$$

Finally,

$$A_\tau^h \leqslant e^{-x_m/\tau}\left(1 - F^h(x_m)\right). \tag{3.5}$$

Numerical precision limits. Starting a discussion of the efficiency of the exponential window in use, one has to pay special attention to the problem of the numerical (*machine*) precision of calculations. Applying an exponential window we face, at certain stages of the algorithm, the problem of operating with very small numbers. Those are the points where loss of precision is probable. That is why it is important to analyze the change in the numerical precision with a change in the exponential window parameter and other important parameters.

Assume that the machine precision can be represented as $\varepsilon = 10^{-p_\varepsilon}$, where ε is to be interpreted as a relative error. More precisely speaking, in the computer the real number u_2 may have the same internal representation as another number u_1, if $|u_2 - u_1| < \varepsilon \cdot |u_1|$. E.g., in the widely used IEEE-standard representation of double precision floating point numbers we have $p_\varepsilon = 16$ (for an overview of the IEEE-standards see, e.g., [5]).

Using the notation introduced in the previous sections, the discrete Fourier transform of some d.f. f (or likewise the product of a distribution function and an exponential window, respectively) is given by

$$\widehat{f_k} = \sum_{l=1}^{N} f_l \cdot e^{2\pi ikl/N}. \tag{3.6}$$

As $e^{-2\pi ikl/N} \sim O(1)$ and $\sum_{l=1}^{N} f_l \leqslant 1$, the numerical error in the calculation of $\widehat{f_k}$ is of order $O(10^{-p_\varepsilon})$. Here the symbol $O(\cdot)$ denotes the order of magnitude.

For the compound distribution we have

$$\widehat{f_l^h} = h(\widehat{f_l}) = \sum_{n=1}^{N} a_n (\widehat{f_l})^n. \tag{3.7}$$

Using the expansion

$$(\widehat{f_l})^n \to \left[\widehat{f_l} + O\left(10^{-p_\varepsilon}\right)\right]^n = (\widehat{f_l})^n + O(10^{-p_\varepsilon}) n (\widehat{f_l})^{n-1}. \tag{3.8}$$

and noting that $\sum a_n n = \overline{n}$ and that the $\widehat{f_l}$ are of order $O(1)$ at most, we can expect the error in the DFT of the compound distribution to be of order $O\left(\overline{n} 10^{-p_\varepsilon}\right)$ at most. Since

$$f_l^h = \frac{1}{N} \sum_{k=1}^{N} \widehat{f_k^h} e^{-2\pi ikl/N}, \tag{3.9}$$

and, as mentioned above, $e^{-2\pi ikl/N} \sim O(1)$ the numerical error of the calculation of f_l^h has the order of $N^{-1} \sum O(\overline{n}\, 10^{-p_\varepsilon}) \sim N^{-1/2} O(\overline{n}\, 10^{-p_\varepsilon})$. In the last estimate we assumed that the errors in the summands were independent and thus the size of the error is proportional to the square root of the number of summands.

To calculate the cumulated distribution function upon convolution, the inverse of the exponential window has to be applied to f^{*n}, and then the sum has to be taken:

$$F^{*N}(x) = \sum_{x_l < x} f_l^{*n} \cdot e^{l\Delta x/\tau}. \tag{3.10}$$

So the error in the cumulated distribution function upon convolution is represented by

$$\sum \frac{1}{\sqrt{N}} O(\overline{n}\, 10^{-p_\varepsilon}) e^{l\Delta x/\tau}. \tag{3.11}$$

Again assuming the independence of errors, we may apply the rule that the size of the error of the sum is of the same order of magnitude as the square root of the sum of the squared error sizes. Simplifying the sum of exponentials as

$$\sum_l e^{2l\Delta x/\tau} = \frac{e^{2x_m/\tau} - 1}{e^{2\Delta x/\tau} - 1} \sim \frac{e^{2x_m/\tau}}{2\Delta x/\tau}. \tag{3.12}$$

we find the following: The numerical error of the whole procedure in the case of the exponential window application can be finally represented as

$$\frac{1}{\sqrt{N}} \overline{n}\, 10^{-p_\varepsilon} \sqrt{\frac{e^{2x_m/\tau} N}{2x_m/\tau}} \leqslant \overline{n} 10^{-p_\varepsilon} e^{x_m/\tau} / \sqrt{2x_m/\tau}. \tag{3.13}$$

4. Performance of the algorithm and adjustment of the parameters

4.1. Two-fold convolution example. We start the analysis of the efficiency of the proposed method using a simple example of the two-fold convolution of the exponential distribution. In this case, the resulting distribution has an explicit analytical form — it is a gamma-distribution: if $f(x) = \alpha e^{-\alpha x}$, then $\widehat{f}(u) = 1/(1 - iu/\alpha) =: \widehat{\gamma}_{1,\alpha}(x)$ and thus $f^{*2}(x) = \gamma_{2,\alpha}(x) = \alpha^2 x \mathrm{e}^{-\alpha x}$.

We consider $x_k := k;\ k := 1, \dots, K;\ K = 2^{12}$, $\alpha = 10^{10}$. We are interested in the value of F_K^{*2}, i.e., the computed cumulative distribution of the convolution function, and compare it to the precise value. The latter is computed as $q_{\text{exact}} = 0.9084218$, while the naive application of the FFT scheme gives a value of 0.9637042, leaving us with an error of 0.055282. This is much larger than the maximal discretization error of some 1.4e-4 given by (2.15).

We might try to improve the result by extending the grid to include points up to x_{2K}, leading, of course, to a reduction in the performance of the algorithm, as the calculation time increases with the number of grid points. As a result, the error will reduce to 2.3642e-3 .

While this is a significant improvement, we can do better by setting the density for all points between 2^{12} and 2^{13} to zero. For a two-fold convolution, this should give us the exact result up to the discretization error. Indeed, the error of 3.575e-5 we obtain is lower than the maximal possible discretization error as estimated above.

As mentioned above, we have doubled the number of grid points in the last calculation, at the price of the computational performance. We could try to estimate the quantile, however, with a grid reaching to 2^{13} and 2^{12} grid points by doubling the step size to a value of two. This would allow us to set all densities for all points between 2^{12} and 2^{13} to zero without loss of performance. The error of 7.1498e-5 we would obtain this way is twice as large as the error in the previous calculation. This is not surprising, as the results are subject to the discretization error only, which can be expected to increase by a factor of two upon doubling the step size. Table 4.1 summarizes the numerical comparison of different methods (see the first four rows of it).

Table 4.1. **Comparison of the FFT with an exponential window to the naive FFT and to the one padded by zeros for the case of 2-fold convolution**

τ	$x_{\max}$	$\exp(-x_{\max}/\tau)$	step size	grid size*	padding	error
∞	2^{12}	1	1	2^{12}	no	5.528e-2
∞	2^{13}	1	1	2^{13}	no	2.364e-3
∞	2^{13}	1	1	2^{13}	yes	3.575e-5
∞	2^{13}	1	2	2^{12}	yes	7.150e-5
1024	2^{12}	1.83e-02	1	2^{12}	no	1.048e-3
512	2^{12}	3.35e-04	1	2^{12}	no	5.429e-5
256	2^{12}	1.13e-07	1	2^{12}	no	3.576e-5
230	2^{12}	1.83e-08	1	2^{12}	no	3.576e-5
128	2^{12}	1.27e-14	1	2^{12}	no	2.680e-4

* We remark that the size of the grid in the case where the padding by zeros was done is the size **after the padding**.

The question arises: Does the application of the exponential window allow to obtain a precision not worse than that obtained by padding with zeros on a grid with 2^{13} points, at the cost of a poorer performance, also on the smaller grid with 2^{12} points only?

This can be achieved if one applies an exponential window, as illustrated by Table 4.1, where – besides the above results – also the errors for different choices of the exponential parameter τ are given. (Note that in our notation the value of $\tau = \infty$ corresponds to the case where an exponential window is not used at all. The error is defined as the difference between the FFT result and the exact result. The performance of the calculation largely depends on the grid size).

The performance of the exponential window, however, crucially depends on a careful choice of τ. We find that for a τ value of 1024 the exponential window reduces the error significantly, however, we are far from the optimal result. For a value of 128 the error is already higher than the optimal result due to the numerical precision limits.

The empirical adjustment of τ leads to a certain optimal value of this parameter $\tau_{opt} \sim 230$. With this value for τ, the calculation on a grid with 2^{12} points and a step size of 1 gives us an error of 3.576e-5, which is slightly larger than the error due to the padding by zeros on a grid with 2^{13} points and much smaller than the result from the padding by zeros on a grid with 2^{12} points and grid size of 2.

The bottom part of Table 4.1 summarizes the results for the exponential window.

Note that all calculations have been done on a machine with a precision of 16 significant digits.

4.2. Adjustment of the exponent parameter. In order to use the exponential window as an effective tool for loss aggregation, one might have a notion of the optimal value of the exponent parameter τ. Obviously, it should be a value that minimizes the aliasing error, but without a crucial loss of numerical precision. Recall now inequality (3.5), which represents the bound for the aliasing error and compare it to the latter inequality (3.13). Both inequalities include the dependence on the exponent parameter τ, and while the aliasing error decreases with decreasing τ, the numerical precision error simultaneously increases. Consequently, one has to take such a value of τ, which minimizes the difference between the aliasing and numerical precision errors. In other words, we make the r.h.s's of the two equalities proportional

$$e^{-x_m/\tau}\left(1 - F^h(x_m)\right) \sim \overline{n}10^{-p_\varepsilon}e^{x_m/\tau}/\sqrt{2x_m/\tau}$$

and conclude that the solution of the balance equation

$$2x_m/\tau = \log(1 - F^h(x_m)) + p_\varepsilon \log(10) - \log(\overline{n}) + \frac{1}{2}\log(2x_m/\tau) \tag{4.1}$$

should give us the optimal value of the product $2\tau x_m$, i.e., of the parameter τ itself.

4.3. Performance of the algorithm — further analysis.

4.3.1. A case with exponential distribution. If the loss severity has an exponential distribution, its characteristic function has the form

$$\widehat{f}_{exp}(t) = \frac{1}{1 - it/a}, \tag{4.2}$$

where a is the parameter of the distribution. As $\widehat{f}^k_{exp}(t) = 1/(1 - it/a)^k$ for $k = 2, 3, \ldots$ is the chf of the gamma distribution with parameters (a, k), the cdf of the aggregate loss distribution can be represented as

$$G_S(x) = e^{-\lambda t}\sum_{k=0}^{\infty}\frac{(\lambda t)^k \widehat{\gamma}_{k,a}(x)}{k!}, \tag{4.3}$$

which can be calculated for each x with the needed precision (as soon as the series in the r.h.s. of the above equality converges fast enough).

We consider the exponential distribution with the parameter $a = 0.001$ and are interested in calculating the 0.9–quantile of the aggregate loss distribution with the Poisson intensity equal to 2. The exact result is equal to $q^{0.9} = 4728.411$. Table 4.2 illustrates the difference between methods. T he optimal value of the exponent parameter is equal $\tau = 561.29$. The right endpoint is chosen at 10^4.

Table 4.2. **Comparison of the exponential window results to the naive FFT and to that obtained via padding by zeros for the case of exponential distribution. The differences between the true result and these obtained by the three methods are given**

Grid size*	naive FFT	1x padding	2x padding	exp. window	discr. error
2^{15}	−70.513	0.628	0.628	0.628	0.610
2^{16}	−70.208	0.323	0.323	0.323	0.305
2^{17}	−70.132	0.170	0.095	0.095	0.153
2^{18}	−70.093	0.132	0.056	0.056	0.076

[*] In this table, unlike the previous one, the grid size indicated is the one **before the padding by zeros**.

Naturally, increasing the grid size one wishes to improve the precision of the result by decreasing the discretization error (which is two times the grid step for the selected example and is indicated in the right column of Table 4.2). We observe that up to a certain value of the grid size N the use of the exponential window gives the same result as the use of the padding by a set of size N of zeros. But for $N \geqslant 2^{17}$ the 1-fold padding by zeros (denoted as "1x" in the table) is no longer enough to ensure that the error in the result is less than the discretization error. At the same time, the use of the exponential window provides the same precision of the result as the use of the 2x padding by zeros, though the computation time is three times less that the 2x padding takes.

4.3.2. Real data example. We proceed with numerical examples, based on the Pareto distribution with parameters having the usual order in the OpRisk data. We work with the generalized Pareto distribution (GPD) in the form

$$G_{\xi,\mu,\beta}(x) = 1 - \left(1 + \xi \frac{x-\mu}{\beta}\right)^{-1/\xi}, \tag{4.4}$$

where μ, β and ξ are called the location, scale and shape parameters, respectively.

For the following example, we took the GPD parameters $\mu = 7000$, $\beta = 12000$ and $\xi = 1.0$. The counting process is the Poisson process with intensity $\lambda = 18$. The optimal value of the exponent parameter τ here is $\tau_{\text{opt}} = 149\,688.5$. We truncate the distribution at the point $x_m = 5 \cdot 10^6$ (and do not normalize it). Table 4.3 gives the absolute values of the 0.9-quantile of the aggregate loss distribution. The grid size indicated in the table is the one before the padding by zeros.

The dynamics of the difference between the methods can be seen from Fig. 4.1. For this example, we again use the GPD parameters ($\xi = 1.0, \mu = 7000, \beta = 12000$) as in the previous example, but the Poisson intensity is now equal to 4. The exponent parameter is adjusted to $\tau_{\text{opt}} = 286\,293.7$. The right endpoint is $x_m = 10^6$.

Figure 4.1 illustrates the convergence of the results obtained with differently arranged FFT methods to the precise result with the increase in the grid volume. Note that the result obtained with 3x padding by zeros and that obtained with an exponential window both converge to the true result with a decrease in the discretization error, but the 3x padding proves to be at least four times slower.

Table 4.3. **Comparison of the FFT with an exponential window to the naive FFT and to the padding by zeros for the case of the GPD distribution**

Grid size	Naive FFT	1x padding	3x padding	$e^{-x/\tau}$
2^{17}	2 854 461	3 130 684	3 132 324	3 132 324
2^{18}	2 854 595	3 130 893	3 132 496	3 132 496
2^{19}	2 854 661	3 130 380	3 132 591	3 132 591
2^{20}	2 854 695	3 131 027	3 132 643	3 132 643

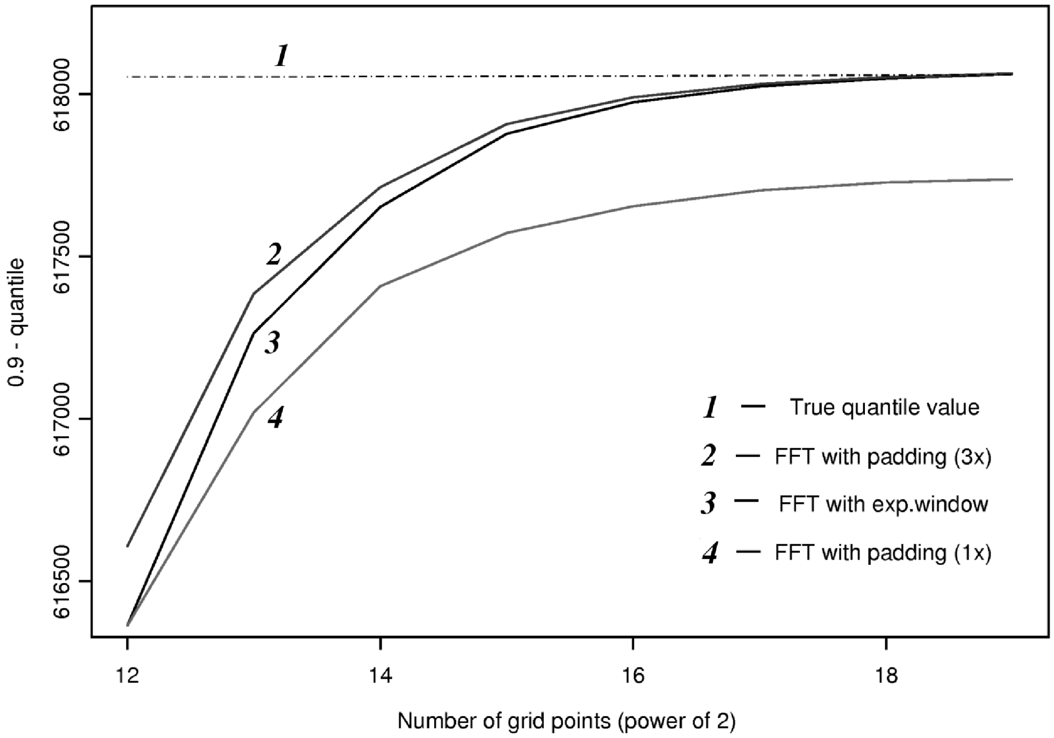

Fig. 4.1. Change in the 0.9-quantile of aggregate losses caused by a change in the number of grid points from 2^{12} up to 2^{19}. The models used: GPD with $(1.0, 7000, 12000)$ and a Poisson process with intensity $\lambda = 4$

Figure 4.2 gives another illustration of the interplay between the error in the result obtained via the exponential window and the error associated with the padding by zeros. We trace the change in these errors with the change of the right truncation point of the distribution we use. The rough breakdown of the precision of the result obtained with the padding is due to the aliasing error, while the precision associated with the exponential window stays within the limits of the discretization error. In fact, Fig. 4.2 depicts the interplay between the aliasing and the discretization error.

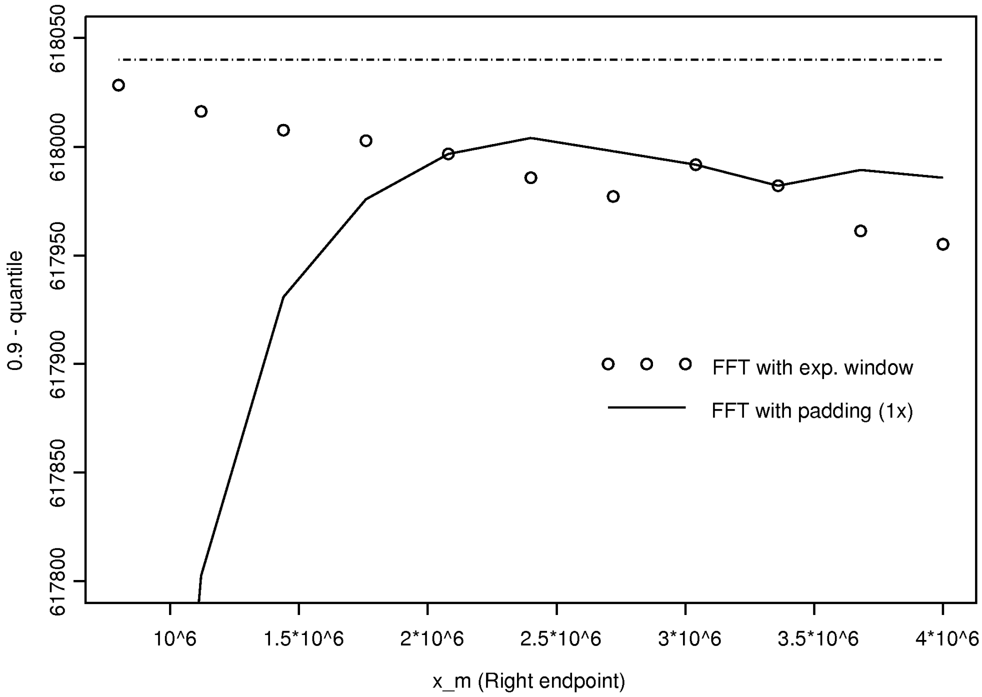

Fig. 4.2. Change in the 0.9-quantile of aggregate losses caused by a change in the right endpoint. Solid line — results calculated with 1-fold padding. Circles — results computed with an exponential window. Dotted line — true quantile value. The models used: GPD with $(1.0, 7000, 12000)$ and a Poisson process with intensity $\lambda = 4$. The grid size used for the calculations is 2^{17}

5. Comparison to other methods

Among the alternative techniques for the calculating aggregate loss recursive methods based on the so called *Panjer recursion* should be mentioned in the first place. This methodology is straightforward and allows one to calculate the quantile of the aggregate loss distribution with a precision only restricted by the discretization error (see [6] for details). However, as the analysis performed in [11] shows, the recursive methods are significantly slower than the FFT when a sufficient precision of the estimated quantile is needed. The advantages of the FFT compared to Monte Carlo techniques are obvious.

6. A note on FFT methods with damping in finance

In the applications of Fourier and Laplace analysis, one of the major problems is the numerical inversion of a chf or of a Laplace image. Considering this problem, the basic difficulty is that integration, in general, should be made over an infinite domain.

Some general notes on the methodology of managing the problems of the numerical inversion of Fourier and Laplace images by increasing the speed of their tending to zero can be found in, e.g., [7] or other classical literature on the characteristic functions and Laplace transforms.

Concerning the application of those methods, a reference should be made to such a problem of financial mathematics as the valuation of options using the FFT. This problem was addressed in [2] and considered in more detail in a series of works including [8].

Usually, dealing with the application of Fourier analysis for option pricing, it is assumed that the chf associated with an option price (often, with the log of an option price) is known analytically. Thus, the problem reduces to the Fourier-inversion of a chf, and that is where the exponential tranformation comes into play, similarly to our case. Specifically, let one be interested in the terminal spot price S_T for a European call of maturity T, with log-price $s_T = \ln(S_T)$ having density q_T and the corresponding chf

$$\phi_T(z) = \mathbf{E}[\exp(izs_T)].$$

Dealing with the inversion of the chf, one faces the problem of the integration over the infinite domain, which leads to the truncation error in the numerical implementation. To reduce this difficulty, an alternative representation of the option price was proposed in [2], particularly by considering the *exponentially damped* price $\widehat{S}_T(\alpha) = C\exp(\alpha k)$, where k is the log of the corresponding strike K.

This transformation allows one to reduce the inversion of the corresponding chf to the calculation of the integral having the form

$$\int\limits_{-\infty-i\alpha}^{\infty-i\alpha} e^{-izk}\psi(z-i)\,dz,$$

where the function $\psi(z)$ is related to chf $\phi_T(z)$. Thus, one can shift the contour of the integration by changing the parameter α. Note that this is related to the so called *saddlepoint approximation* for the Fourier inversion, having exactly the specified form.

The problem is reduced to the optimal choice of the parameter α, which would keep the balance between the truncation error and the discretization error. A method for searching the solution to this problem w.r.t. α described in [8] allows one to reduce it to the following easier problem:

$$\operatorname*{argmin}_{\alpha} \mid e^{-\alpha k}\psi(-(\alpha+1)i) \mid .$$

Shortly summarizing: In the context of option pricing the key problem related to Fourier analysis is the reduction of the truncation error, while in our case we deal with the aliasing problem. The methods for choosing an optimal exponential parameter α described in [2] and [8] also differ from those used by us to justify the optimal choice of the exponential parameter τ.

7. Conclusions

Our investigation confirms the FFT to be a convenient and effective tool for precise calculation of multifold convolutions and, particularly, of the distributions of random sums of iid random variables. We have found that the use of exponential windows may considerably increase the speed of calculation by the FFT with the required accuracy. Instead of padding by zeros, which is demanding to the machine time, one can estimate the optimal parameter for the exponential window and then perform the calculations by the FFT with a precision only restricted by discretization error.

The key point of our investigation is the analysis of the balance equation that allows one to find the optimal value of the exponential parameter used in our FFT procedure.

We find this simple methodology useful for numerical problems of insurance concerning multiple calculations of the quantiles of aggregate loss distributions with a high precision.

Acknowledgments. The authors thank the anonymous referee for his or her useful remarks and comments that helped to significantly improve this article.

Financial support by Christian Doppler Laboratory for Portfolio Risk Management (PRisMa Lab) http://www.prismalab.at/ is gratefully acknowledged.

References

1. R. Bracewell, *The Fourier transform and its applications*, Springer, 2000.

2. P. P. Carr and D. B. Madan, *Option valuation using the fast Fourier transform*, J. Computational Finance. **2** (1999), pp. 61–73.

3. F. J. Harris, *On the use of windows for harmonic analysis with the discrete Fourier transform*, Proc. IEEE. **66** (1978), pp. 51–83.

4. A. Jerri, *Integral and discrete transforms with applications and error analysis*, Marcel Dekker, Inc., New York, 1992.

5. W. Kahan, *Lecture notes on the status of IEEE standard 754 for binary floating-point arithmetic*, UC Berkeley, online version http://www.cs.berkeley.edu/∼wkahan/ieee754status/ieee754.ps (1996).

6. S. Klugman, H. Panjer, and G. Willmot, *Loss models. From data to decisions (2nd edition)*, Wiley–Interscience, Hoboken, NJ, USA, 2004.

7. V. I. Krylov and N. S. Skoblya, *A Handbook of Methods of Approximate Fourier Transformation and Inversion of the Laplace Transformation.* Mir Publishers, Moscow, 1977 (in Russian).

8. R. Lord and C. Kahl, *Optimal Fourier inversion in semi-analytical option pricing*, J. Computational Finance. **10** (2007), no. 4, pp. 1–30.

9. A. H. Nuttall, *Some windows with very good sidelobe behaviour*, IEEE Transactions of Acoustics, Speech, and Signal Processing (1981), no. 1, pp. 84–91.

10. H. Panjer, *Operational risk. Modelling analytics*, Wiley Series in Probability and Statistics. Wiley-Interscience, Hoboken, NJ, 2006.

11. G. Temnov and R. Warnung, *A comparison of loss aggregation methods for Operational risk*, Journal of Operational risk. **3** (2008), no. 1, pp. 3–23.