



## **Libstable: Fast, Parallel and High-Precision Computation of $\alpha$ -Stable Distributions in C/C++ and MATLAB**

**Javier Royuela-del-Val**  
University of Valladolid

**Federico Simmross-Wattenberg**  
Universidad de Valladolid

**Carlos Alberola-López**  
Universidad de Valladolid

---

### **Abstract**

$\alpha$ -stable distributions are a wide family of probability distributions used in many fields where probabilistic approaches are taken. However, the lack of closed analytical expressions is a major drawback for their application. Currently, several tools have been developed to numerically evaluate their density and distribution functions or estimate their parameters, but available solutions either do not reach sufficient precision on their evaluations or are too slow for several practical purposes. Moreover, they do not take full advantage of the parallel processing capabilities of current multi-core machines. Other solutions work only on a subset of the  $\alpha$ -stable parameter space. In this paper we present a C/C++ library and a MATLAB front-end that allows fully parallelized, fast and high precision evaluation of density, distribution and quantile functions (PDF, CDF and  $CDF^{-1}$  respectively), random variable generation and parameter estimation of  $\alpha$ -stable distributions in their whole parameter space. The library provided can be easily integrated on third party developments.

*Keywords:*  $\alpha$ -stable distributions, numerical calculation, parallel processing.

---

## **1. Introduction**

$\alpha$ -stable distributions are a wide family of probability distributions that allow adjustable levels of heavy tails and skewness and include family includes Gaussian, Cauchy and Lévy distributions as particular cases. They were first introduced by Lévy (1925). Since then,  $\alpha$ -stable distributions have been applied in many fields where probabilistic approaches are

applied, such as hydrology (Mandelbrot and Wallis 1968), finances (Fama 1965; Borak *et al.* 2011), noise in communications channels (Arce 2005), network traffic analysis (Simmross-Wattenberg *et al.* 2011) or medical image segmentation (Salas-Gonzalez *et al.* 2013).

The increasing interest on  $\alpha$ -stable distributions is due, on the one hand, to the empirical evidence that they properly describe the behavior of real data exhibiting impulsiveness or strong asymmetries. On the other hand, the *generalized central limit theorem* (Samorodnitsky and Taqqu 1994) states that the normalized sum of independent and identically distributed (i.i.d.) random variables with finite or infinite variance converges, if it does, to an  $\alpha$ -stable distribution. This result provides theoretical support when the data under study can be interpreted as the superposition of many independent sources.

The major drawback for the application of  $\alpha$ -stable models is the lack of closed analytical expressions for their PDF or CDF (except in particular cases), which makes the application of numerical methods necessary to evaluate them. Besides, the non-existence of moments of order two or higher (except in the Gaussian case) increases the difficulty in estimating their parameters to fit real data. Several authors have addressed both the numerical evaluation of the PDF or CDF of  $\alpha$ -stable distributions (Nolan 1997; Mittnik *et al.* 1999a; Belov 2005; Menn and Rachev 2006; Górska and Penson 2011) and the estimation of their parameters (Fama and Roll 1971; Koutrouvelis 1980, 1981; McCulloch 1986; Mittnik *et al.* 1999b; Nolan 2001; Fan 2006; ?) and have proposed different methods and algorithms for these purposes (details are discussed in section 2). Some authors also provide implementations of their own or others' methods, offering software solutions or packages in various common computer languages (Nolan 2006; Veillete 2010; Liang and Chen 2013). However, some of these solutions are not fully operational in their public domain versions and cannot take advantage of multi-core processors (Nolan 2006), have limited accuracy (Belov 2005; Menn and Rachev 2006) or take too long doing computations for some practical applications (Veillete 2010; Liang and Chen 2013).

In this paper we present a C/C++ library and a MATLAB (Inc. 2012) front-end that allows parallelized, fast and high precision evaluation of density and distribution functions, random variables generation and parameter estimation of  $\alpha$ -stable distributions. The library can be easily integrated on third party developments to be used by practitioners. Its utilization in MATLAB is straightforward and allows taking advantage of both the high efficiency of the compiled C/C++ code and the user friendly graphical interface and statistical tools of the MATLAB environment.

The rest of the paper is organized as follows. In section 2, currently proposed algorithms and methods to numerically evaluate the density and distribution functions of  $\alpha$ -stable distributions are described, as well as reported methods to estimate their parameters. In section 3, the developed library and its functionalities are presented. The results obtained with the library are discussed in section 4. In section 5.2, the usage the proposed library is described. Finally, section 6 describes main conclusions and further work possibilities derived from this work.

## 2. Numerical methods in $\alpha$ -stable distributions

Since, generally speaking, there are not closed expressions for them,  $\alpha$ -stable distributions are frequently described by their characteristic function  $\Phi(t) = \exp[\Psi(t)]$  as (Samorodnitsky and

Taqqu 1994):

$$\Psi(t) = \begin{cases} -|\sigma t|^\alpha [1 - i\beta \tan(\frac{\pi\alpha}{2}) \text{sign}(t)] + i\mu t, & \alpha \neq 1, \\ -|\sigma t| [1 + i\beta \frac{2}{\pi} \text{sign}(t) \ln(|t|)] + i\mu t, & \alpha = 1, \end{cases} \quad (1)$$

where

$$\text{sign}(t) = \begin{cases} 1, & t > 0, \\ 0, & t = 0, \\ -1, & t < 0. \end{cases}$$

The four parameters that govern the  $\alpha$ -stable distributions appear in equation (1) and are usually denoted as follows: the stability index  $\alpha \in (0, 2]$ , the skewness parameter  $\beta \in [-1, 1]$ , the scale parameter  $\sigma > 0$  and the location parameter  $\mu \in \mathbb{R}$ . An  $\alpha$ -stable distribution is said *standard* if  $\sigma = 1$  and  $\mu = 0$ . When  $\alpha = 2$  the distribution becomes normal with standard deviation  $\sigma/\sqrt{2}$  and mean  $\mu$  ( $\beta$  becomes irrelevant). The Cauchy distribution results from setting  $\alpha = 1$  and  $\beta = 0$  with scale parameter  $\sigma$  and locate parameter  $\mu$ , and the Lévy distribution when  $\alpha = 0.5$  and  $\beta = 1$ . These are the only cases for which the PDF can be expressed analytically. In all other cases, PDF must be calculated numerically.

Next subsections discuss different methods proposed in the literature to obtain the PDF and CDF, estimate the parameters of  $\alpha$ -stable distributions and to generate samples of an  $\alpha$ -stable random variable.

## 2.1. Numerical computation of $\alpha$ -stable distributions

The PDF and characteristic function of any probability function are related via the Fourier inversion formula given by

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \phi(t) e^{-itx} dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{\psi(t) - itx} dt. \quad (2)$$

When substituting expression (1) in (2) the resulting integral cannot be, in general, solved analytically; therefore it must be evaluated by numerical methods. To this end, the well-known *Fast Fourier Transform* (FFT) provides a fast algorithm to efficiently evaluate the previous integral. Mittnik *et al.* (1999a), for instance, apply the FFT directly to calculate the PDF. However, this approach suffers from several important drawbacks. First, the algorithm provides the value of the integral on a set of evenly spaced points of evaluation. This is not valid for some applications, where the PDF or CDF must be evaluated at some specific set of points. A posterior step of interpolation is then required, which introduces additional computational costs and reduces precision. Second, the method is only suitable for  $\alpha$  close to 2, for which the tails of the distribution decay more quickly. When  $\alpha$  is small, the tails decay very slowly and the *aliasing* effect becomes more noticeable, thus reducing the achievable precision. Experimentally, the committed absolute error is in the order of  $10^{-5}$ , but relative error goes as high as  $10^{-2}$ . Menn and Rachev (2006) propose a method based on a refinement of the FFT to increase precision in the central part of the PDF. The tails of the distribution are calculated via the Bergström asymptotic series expansion (Zolotarev 1986), which provides an alternative expression as an infinite sum of decaying terms. This way they achieve relative precision of about  $10^{-4}$ , but this precision strongly depends on the values of  $\alpha$  and  $\beta$  and the method is only applicable when  $\alpha > 1$ .

Górska and Penson (2011) follow a different approach and obtain explicit expressions for the PDF and CDF as series of generalized hypergeometric functions. However, the expressions

involved in the calculation are expensive to evaluate and the results are only valid for some rational values of the parameters  $\alpha$  and  $\beta$ , so they are not valid for the whole parameters space.

When compared with the FFT, direct integration of the expression in (2) by numerical quadrature initially implies a higher computational cost, but evaluation can be performed at any desired set of points without the need of additional interpolation steps and there is no restriction on the values of the parameters of the distribution. This method has been implemented by Nolan (1999)<sup>1</sup>. However when  $\alpha$  is small the integrand oscillates very quickly and its amplitude decays slowly along the infinite integration interval, which limits the achievable precision although many evaluations of the integrand are used. According to the author, the method results are valid only for  $\alpha > 0.75$  and it achieves a precision in the order of  $10^{-6}$ . Similar results have been obtained later by Below (2005), where the infinite integrand interval is divided in two (one bounded and the other infinite) and two quadrature formulae are applied.

In order to overcome the previous difficulties, Nolan (1997) obtains a new set of equations from the original ones by means of a convenient analytic extension of the integrand to the complex plane. This way, a continuous, bounded, non-oscillating integrand is obtained. Moreover, the integration interval becomes finite. The expressions obtained allows the author to achieve, by numerical quadrature, a relative accuracy in the order of  $10^{-14}$  in most of the parameter space. For numerical convenience, a slight different parametrization of the distribution is employed, based on Zolotarev (1986) M parametrization . The modification introduced consists in a shift of the distribution along the abscissa axis in order to avoid the discontinuity of the distribution at  $\alpha = 1$ . In this paper, we denote the change in parametrization by the subindex 0 and the resulting characteristic function is given by  $\Phi_0(t) = \exp[\Psi_0(t)]$  where

$$\Psi_0(t) = \begin{cases} -|\sigma t|^\alpha [1 + i\beta \tan(\frac{\pi\alpha}{2}) \text{sign}(t) (|\sigma t|^{1-\alpha} - 1)] + i\mu_0 t, & \alpha \neq 1, \\ -|\sigma t| [1 + i\beta \frac{2}{\pi} \text{sign}(t) \ln(|\sigma t|)] + i\mu_0 t, & \alpha = 1. \end{cases} \quad (3)$$

The parameters  $\alpha$ ,  $\beta$  and  $\sigma$  keep their previous meaning while the original and modified location parameters  $\mu$  and  $\mu_0$  are related according to

$$\mu = \begin{cases} \mu_0 - \beta \tan(\frac{\alpha\pi}{2}) \sigma, & \alpha \neq 1, \\ \mu_0 - \beta \frac{2}{\pi} \sigma \ln(\sigma), & \alpha = 1. \end{cases} \quad (4)$$

With this modification, the resulting distribution is continuous in its four parameters, which is convenient when estimating the parameters of the distribution or approximating its PDF or CDF.

## 2.2. Parameter estimation of $\alpha$ -stable distributions

As previously explained, the lack of closed expressions for the PDF or CDF of  $\alpha$ -stable distributions implies a major drawback to estimate their parameters. Therefore, the technique employed for the estimation usually rely on numerical evaluations of these functions or, alternatively, are based on other characteristics of the distributions.

Several methods can be found in the related literature. Hill (1975) proposes the estimation of the  $\alpha$  parameter by linear regression on the right tail of the data empirical distribution.

<sup>1</sup>Although finally published in 1999, this article is already cited in Nolan (1997) as a "to appear in" reference, so the described software therein is earlier to 1997

However, in many practical cases the method is not applicable because of the high number of samples needed to detect the tail behavior.

McCulloch (1986) proposes an algorithm to estimate the four parameters of the  $\alpha$ -stable distribution simultaneously from sample quantiles and tabulated values. The resulting estimator has a very low computational cost but a low accuracy as well. However, it can be conveniently used as an initial estimation for other methods.

Koutrouvelis (1981) departs from the empirical characteristic function to, by means of recursive linear regressions on its log-log plot, obtain estimators for  $\alpha$  and  $\sigma$  in a first step and for  $\beta$  and  $\mu$  on a second one. The method implies a higher computational cost than the quantile method, but yields more accurate results.

Maximum Likelihood (ML) estimation is considered the most accurate estimator available for  $\alpha$ -stable distributions (Borak *et al.* 2011). However, numerical methods to both approximate the PDF and to maximize the likelihood of the sample must be used, which implies a very high computational cost due to the numerous PDF evaluations required to maximize the likelihood in the four-dimensional parameter space. However, increasing computer capabilities and the use of precalculated values of the PDF allows the application of this method on certain applications (Nolan 2001).

### 2.3. Generation of $\alpha$ -stable random variables

Given the lack of closed expressions for the CDF or its inverse (the quantile function), the simulation of  $\alpha$ -stable distributed random variables cannot be achieved easily from a uniformly distributed random variable. Chambers *et al.* (1976) provided a direct method to generate an  $\alpha$ -stable random variable by means of the transformation of an exponential and a uniform random variable. The method proposed lacks theoretical demonstration until Weron (1996) gave an explicit proof and slightly modified the original expressions. The resulting method is regarded as the fastest and the most accurate (Weron 2004).

Based on the methods described so far, currently several software tools are available. The program STABLE (Nolan 2006) employs Nolan's expressions (Nolan 1997) for the high precision computation of PDF, CDF and quantile function and maximum likelihood parameter estimation. However, a fully operational version of the software is not publicly available. Veillete (2010) has developed a MATLAB package that also applies Nolan's expressions for high precision PDF and CDF evaluation and Koutrouvelis (1981) method for parameters estimation based on the characteristic function. However, the performance obtained is low when trying to achieve high precision or when fitting a large data sample. A package with similar features and characteristics has been reported by Liang and Chen (2013).

## 3. Algorithms and implementation

One purpose of the proposed library is to have a fast and accurate tool to numerically evaluate the PDF and CDF of  $\alpha$ -stable distributions. From the previous study, it may be concluded that those methods that employ Nolan's expressions gives the most accurate results. Therefore, the developed library makes use them as well. However, these expressions have some aspects than must be addressed. Firstly, the mathematical expressions involved in the calculation are, in computational terms, expensive to evaluate. In second place, although the integral involved in the calculations (a convenient transformation of equation (2) has some

desirable properties, it is generally hard to approximate. Therefore some strategies have to be elaborated in order to evaluate it accurately and efficiently.

### 3.1. Fast and accurate evaluation of Nolan's expressions

Nolan's expressions allow to calculate the PDF of a standard  $\alpha$ -stable distribution with  $\alpha \neq 1$  as

$$f_{\mathbf{X}}(x; \alpha, \beta) = \begin{cases} \frac{\alpha}{\pi(x - \zeta)|\alpha - 1|} \int_{-\theta_0}^{\pi/2} h_{\alpha, \beta}(\theta; x) d\theta & x > \zeta \\ \frac{\Gamma(1 + \frac{1}{\alpha}) \cos(\theta_0)}{\pi(1 + \zeta^2)^{\frac{1}{2\alpha}}} & x = \zeta \\ f_{\mathbf{X}}(-x; \alpha, -\beta) & x < \zeta \end{cases} \quad (5)$$

where

$$\begin{aligned} \zeta(\alpha, \beta) &= -\beta \tan\left(\frac{\pi\alpha}{2}\right) \\ \theta_0(\alpha, \beta) &= \frac{1}{\alpha} \arctan\left(\beta \tan\left(\frac{\pi\alpha}{2}\right)\right) \\ h_{\alpha, \beta}(\theta; x) &= (x - \zeta)^{\frac{\alpha}{\alpha-1}} V_{\alpha, \beta}(\theta) e^{-(x-\zeta)^{\frac{\alpha}{\alpha-1}} V_{\alpha, \beta}(\theta)} \\ V_{\alpha, \beta}(\theta) &= (\cos(\alpha\theta_0))^{\frac{1}{\alpha-1}} \left(\frac{\cos(\theta)}{\sin(\alpha(\theta_0 + \theta))}\right)^{\frac{\alpha}{\alpha-1}} \frac{\cos(\alpha\theta_0 + (\alpha - 1)\theta)}{\cos(\theta)} \end{aligned} \quad (6)$$

When  $\alpha = 1$ , the definition of the expressions changes to

$$f_{\mathbf{X}}(x; 1, \beta) = \begin{cases} \int_{-\frac{\pi}{2}}^{\pi/2} h_{1, \beta}(\theta; x) d\theta & \beta \neq 0 \\ \frac{1}{\pi(1 + x^2)} & \beta = 0 \end{cases} \quad (7)$$

where

$$\begin{aligned} h_{1, \beta}(\theta; x) &= e^{-\frac{\pi x}{2\beta} V_{1, \beta}(\theta)} \\ V(\theta; 1, \beta) &= \frac{2}{\pi} \left(\frac{\frac{\pi}{2} + \beta\theta}{\cos(\theta)}\right) e^{\frac{1}{\beta}(\frac{\pi}{2} + \beta\theta) \tan(\theta)} \end{aligned} \quad (8)$$

It is worth noticing that the change in the definition of the expressions above does not imply a discontinuity in the PDF when  $\alpha = 1$ . It can be demonstrated that the expressions for  $\alpha \neq 1$  tend to those for  $\alpha = 1$  when the stability index approaches to that value. For no standard distributions, the PDF is calculated for the corresponding standard one ( $\sigma = 1$  and  $\mu_0 = 0$ ) and then properly scaled and displaced according to

$$\begin{aligned} f_{\mathbf{X}}(x; \alpha, \beta, \sigma, \mu_0) &= \frac{1}{\sigma} f_{\mathbf{X}}\left(\frac{x - \mu_0}{\sigma}; \alpha, \beta\right) \\ F_{\mathbf{X}}(x; \alpha, \beta, \sigma, \mu_0) &= F_{\mathbf{X}}\left(\frac{x - \mu_0}{\sigma}; \alpha, \beta\right) \end{aligned} \quad (9)$$

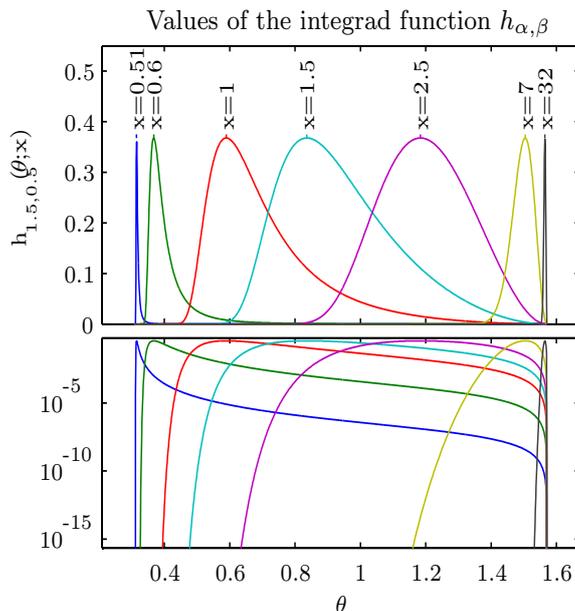


Figure 1: Integrand function  $h_{1.5,0.5}(\theta; x)$  for various values of  $x$  in linear (top) and logarithmic (bottom) scales. As  $x$  grows, most of the area under the curve gets concentrated in a narrow peak close to one extreme of the integration interval.

In figure 1 the function  $h_{\alpha,\beta}(\theta; x)$  to be integrated is represented both in linear and logarithmic scales. As the point of evaluation  $x$  of the PDF increases, the integrand becomes closer and closer to a singular peak that concentrates most of the area under it. The same behaviour occurs when  $x$  tends to  $\zeta$ . The numerical method employed to evaluate the integral can, on the one side, miss this peak and underestimate the integral and, on the other side, employ most of the evaluations of the integrand at regions of the integration interval where it takes very low values that contribute marginally to the final value of the integral increasing run time. In order to concentrate the evaluations there where they are more relevant, the integral is divided as represented in figure 2. Before integrating, the peak of  $h_{\alpha,\beta}(\theta; x)$  is located numerically. Then, a symmetric interval around the peak where the integrand holds above a determined threshold is defined. The value of the threshold depends on the accuracy required to evaluate the integral. This required accuracy can be easily adjusted by the user.

With the interval of integration properly divided, a first approximation to the final value of the integral is obtained in the interval around the peak by applying Gauss-Kronrod quadrature formulas (Press *et al.* 1994). Finally, the area under the rest of the integration interval is calculated. Since an initial estimation of the final value of the integral is available already, the area in these subintervals can be approximated only with necessary precision according to its contribution related to the first approximation obtained or even neglected, avoiding its calculation. As a result, the number of evaluations needed to approximate the integral decreases improving run time.

When calculating the CDF, a similar approach is followed. In this case, the maximum of the integrand is always in one extreme of the integration interval (see Nolan (1997) for details on the expressions involved), so there is no need to find it numerically. To calculate the quantile function, the CDF is numerically inverted. A initial guess of the value of the function is

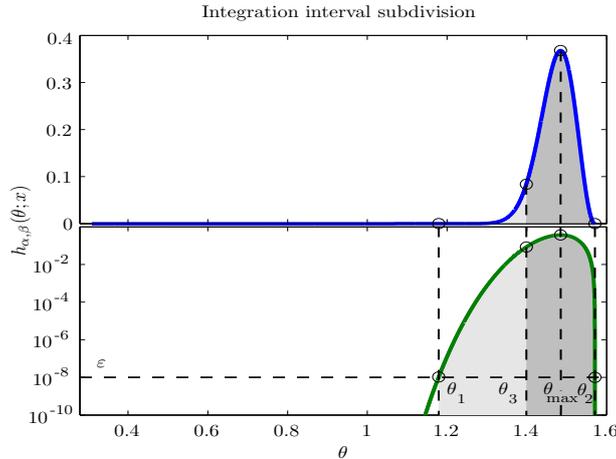


Figure 2: Subdivision of the integration interval in linear (top) and logarithmic scales (bottom). The maximum of the integrand ( $\theta_{max}$ ) and points for which  $h_{1.5,0.5}(\theta; x)$  crosses a threshold  $\epsilon$  ( $\theta_1, \theta_2$ ) are located. Then, a symmetric interval around  $\theta_{max}$  is determined ( $\theta_3$ ). In this example  $\alpha = 1.5$ ,  $\beta = 0.5$  and  $x = 6$ .

obtained by interpolation over tabulated CDF values and then a root finding algorithm is applied. Routines employed for numerical quadrature and root finding are supplied by the GNU Scientific Library (GSL) (Galassi *et al.* 2009).

### 3.2. Parallelization of the workload

The evaluation of the PDF, CDF or  $CDF^{-1}$  at one point is completely independent from the evaluation at a different point. Besides, in practical applications it will be required to evaluate the functions in several points, as when estimating the  $\alpha$ -stable parameters of given data with a method based on the PDF or CDF of distribution. Therefore, the evaluation at different points can be done in parallel. When called, the library functions distribute the points of evaluation between several threads of execution. The number of available threads of execution can be fixed manually or automatically determined. When computation finishes, the results are gathered together. Parallelism has been implemented using Pthreads (Barney 2011), that allows fine control on the threads creation and execution. The distributions can be calculated in the original parameterization with characteristic function given by equation (1) or in the modified one following equation (3).

### 3.3. Parameter estimation and random variable generation

Four different methods of parameter estimation of  $\alpha$ -stable distribution are available in the library. First, the McCulloch's method (McCulloch 1986) allows very fast parameter estimation, at the cost of lower accuracy. Second, the iterative Koutrouvelis method based on the sample characteristic function (Koutrouvelis 1981) produce better estimations with longer execution times. Third, maximum likelihood can be directly used. However, the elevated computation cost of both numerical evaluation of the PDF and maximization in the four-dimensional parameter space makes this method very slow when sample size is large. For these cases, a modified ML approach is implemented, in which the maximization search is

only performed in the 2-D  $\alpha$ - $\beta$  space. On each iteration of the maximization procedure,  $\sigma$  and  $\mu$  are estimated with McCulloch's algorithm according to current  $\alpha$ ,  $\beta$  estimations.

The CMS method modified by Weron is used to simulate  $\alpha$ -stable random variables. To this end, a high quality uniform random numbers generator provided by the GSL is employed.

## 4. Results and performance

In this section, the results obtained by the developed library are exposed and discussed. The analysis is focused on the precision and performance obtained when evaluating the PDF, CDF of  $\alpha$ -stable distribution.

### 4.1. Precision results

Since no tabulated values for these function are available with enough precision in the literature, errors are measured against the numerical results provided by the program STABLE, which has been used frequently in the literature as *ground truth* (Belov 2005; Weron 2004; Menn and Rachev 2006). Error is expressed in relative terms and measured for different values of the parameters  $\alpha$  and  $\beta$ . The parameters  $\sigma$  and  $\mu$  are fixed to 1 and 0. The abscissa axis is divided in several intervals and each interval is subdivided in an evenly distributed set of points. Errors committed inside each interval are averaged. This way, the behavior at the tails and in the central region of the distribution can be analyzed independently. The data obtained for a set of  $\alpha$  and  $\beta$  values is presented in tables 1 and 2 for the evaluation of the PDF and CDF, respectively. The precision obtained both at the tails and at the central part of the distribution is fairly high, being in many cases close to the hardware precision limit employed in the calculations (about  $10^{-16}$ ). The data obtained for a thinner sweep of the parameters is summarized in figure 3. The data has been smoothed for convenient visualization with a median filter of size three. When  $\alpha < 1$ , the results obtained are in practice equivalent to those obtained by the program STABLE. When  $\alpha$  gets close to 1, the error increases. In fact, Nolan's expressions become singular and hard to integrate when  $\alpha$  tends to one. However, given the small relative error measured and the lack of exact values of the distribution, we cannot determine witch software (the program STABLE used as reference or the proposed library) is in fact deviating from the true values of the distribution. The error increases also as  $\alpha$  approaches to two (the gaussian case). This can be due to the faster decaying of the tails of the distribution, therefore small absolute differences in the values obtained translate into higher relative errors measured. Despite this, relative error stays in the order of  $10^{-13}$ .

### 4.2. Performance results

The performance of the library is measured as the number of evaluations of the PDF or CDF it can execute per unit of time. To measure it, for a set of  $\alpha$  and  $\beta$  values, 100 calls to the library have been done, with 10000 evaluations of the PDF or CDF per call. The measurements are repeated setting the required precision to different values. Results are compared with those obtained with two available alternatives that achieve comparable precision: the program STABLE and Veillete's MATLAB functions. The tests have been performed in a machine with four Quad-Core AMD Opteron(tm) Processor 8350 (16 cores in total) with a CPU frequency of 2.0GHz.

| $\alpha$ | $\beta$ | $x$ -axis interval   |                      |                      |
|----------|---------|----------------------|----------------------|----------------------|
|          |         | $[-1000, -10]$       | $[-10, 10]$          | $(10, 1000]$         |
| 0.25     | 0       | $7.1 \cdot 10^{-16}$ | $6.5 \cdot 10^{-16}$ | $7.1 \cdot 10^{-16}$ |
|          | 0.5     | $1.5 \cdot 10^{-15}$ | $1.0 \cdot 10^{-15}$ | $6.9 \cdot 10^{-16}$ |
|          | 1       | --                   | $9.8 \cdot 10^{-16}$ | $6.5 \cdot 10^{-16}$ |
| 0.5      | 0       | $9.7 \cdot 10^{-16}$ | $7.2 \cdot 10^{-16}$ | $9.7 \cdot 10^{-16}$ |
|          | 0.5     | $1.5 \cdot 10^{-15}$ | $8.9 \cdot 10^{-16}$ | $8.6 \cdot 10^{-16}$ |
|          | 1       | --                   | $8.9 \cdot 10^{-16}$ | $7.2 \cdot 10^{-16}$ |
| 0.75     | 0       | $3.7 \cdot 10^{-15}$ | $8.1 \cdot 10^{-16}$ | $3.7 \cdot 10^{-15}$ |
|          | 0.5     | $7.5 \cdot 10^{-15}$ | $1.1 \cdot 10^{-15}$ | $2.6 \cdot 10^{-15}$ |
|          | 1       | --                   | $7.4 \cdot 10^{-14}$ | $2.2 \cdot 10^{-15}$ |
| 1        | 0       | $6.4 \cdot 10^{-16}$ | $7.1 \cdot 10^{-16}$ | $6.4 \cdot 10^{-16}$ |
|          | 0.5     | $6.7 \cdot 10^{-13}$ | $2.0 \cdot 10^{-15}$ | $8.1 \cdot 10^{-13}$ |
|          | 1       | --                   | $7.7 \cdot 10^{-13}$ | $3.5 \cdot 10^{-13}$ |
| 1.5      | 0       | $2.4 \cdot 10^{-13}$ | $1.3 \cdot 10^{-15}$ | $2.4 \cdot 10^{-13}$ |
|          | 0.5     | $2.3 \cdot 10^{-13}$ | $1.3 \cdot 10^{-15}$ | $1.8 \cdot 10^{-13}$ |
|          | 1       | --                   | $4.3 \cdot 10^{-13}$ | $7.9 \cdot 10^{-14}$ |

Table 1: Relative error committed in the calculation of the PDF.

| $\alpha$ | $\beta$ | $x$ -axis interval   |                      |                      |
|----------|---------|----------------------|----------------------|----------------------|
|          |         | $[-1000, -10]$       | $[-10, 10]$          | $(10, 1000]$         |
| 0.25     | 0       | $6.0 \cdot 10^{-15}$ | $1.4 \cdot 10^{-15}$ | $7.1 \cdot 10^{-16}$ |
|          | 0.5     | $1.2 \cdot 10^{-14}$ | $2.2 \cdot 10^{-15}$ | $7.7 \cdot 10^{-16}$ |
|          | 1       | --                   | $1.2 \cdot 10^{-15}$ | $8.4 \cdot 10^{-16}$ |
| 0.5      | 0       | $2.3 \cdot 10^{-14}$ | $1.6 \cdot 10^{-15}$ | $6.5 \cdot 10^{-16}$ |
|          | 0.5     | $4.0 \cdot 10^{-14}$ | $2.2 \cdot 10^{-15}$ | $6.6 \cdot 10^{-16}$ |
|          | 1       | --                   | $6.4 \cdot 10^{-16}$ | $2.7 \cdot 10^{-16}$ |
| 0.75     | 0       | $8.5 \cdot 10^{-14}$ | $1.8 \cdot 10^{-15}$ | $6.3 \cdot 10^{-16}$ |
|          | 0.5     | $1.6 \cdot 10^{-13}$ | $2.7 \cdot 10^{-15}$ | $6.3 \cdot 10^{-16}$ |
|          | 1       | --                   | $2.5 \cdot 10^{-13}$ | $6.8 \cdot 10^{-16}$ |
| 1        | 0       | $2.4 \cdot 10^{-15}$ | $4.7 \cdot 10^{-16}$ | $2.4 \cdot 10^{-16}$ |
|          | 0.5     | $6.0 \cdot 10^{-13}$ | $4.7 \cdot 10^{-16}$ | $2.4 \cdot 10^{-16}$ |
|          | 1       | --                   | $5.7 \cdot 10^{-14}$ | $1.8 \cdot 10^{-13}$ |
| 1.5      | 0       | $8.1 \cdot 10^{-13}$ | $2.2 \cdot 10^{-15}$ | $8.1 \cdot 10^{-16}$ |
|          | 0.5     | $6.2 \cdot 10^{-13}$ | $2.7 \cdot 10^{-15}$ | $8.1 \cdot 10^{-16}$ |
|          | 1       | --                   | $2.4 \cdot 10^{-14}$ | $5.9 \cdot 10^{-16}$ |

Table 2: Relative error committed in the calculation of the CDF.

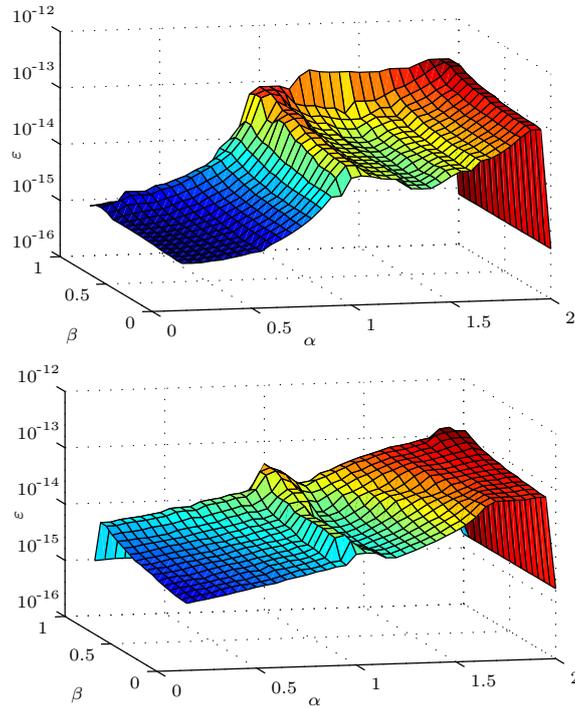


Figure 3: Averaged relative error committed in the calculation of the PDF (top) and CDF (bottom) of a standard  $\alpha$ -stable distribution.

Performance measurements obtained for  $\alpha = 0.75$  and  $\beta = 0.5$  are presented in figure 4. Results for other values of the parameters exhibit the same behavior, up to a scale factor. When moderate errors are tolerable, Veillete's MATLAB functions achieve a performance higher than the one of both program STABLE and the C/C++ library developed. This can be explained by the simpler quadrature method employed. However, when precision requirements increase, the performance of the MATLAB solution quickly decreases becoming much slower than the rest of methods, where performance is not so severely penalized. The more precise quadrature method and the strategy of integration followed can explain this behavior.

When using just one execution thread, the proposed library outperforms the program STABLE by a factor of  $\sim 1.5$  for the evaluation of the PDF, although similar results are obtained when evaluating the CDF. Finally, the parallel capability of the C/C++ library clearly outperforms the program STABLE when multiple processing cores are available (as is often the case in modern machines). For the CDF, the increase in performance with respect to the program STABLE is approximately equal to the number of threads used. In the case of the PDF, this increase is higher given the superior performance with one execution thread.

The results presented in figure 4 allows to estimate the to what extent the proposed library is able to parallelize the workload. According to the Amdahl's law ? the performance improvement obtained as a function of the number of execution threads used is given by

$$G_N = \frac{1}{1-p} + \frac{p}{N} \quad (10)$$

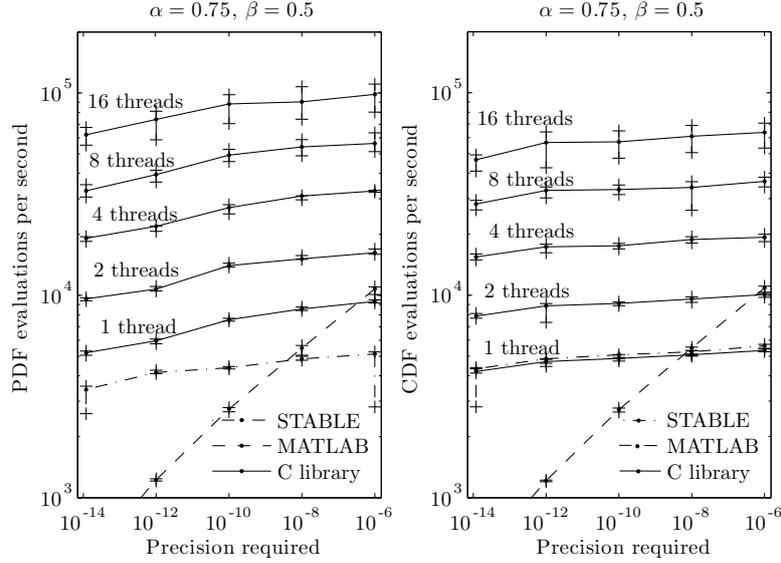


Figure 4: Performance of the calculation of PDF and CDF vs. required precision for various numbers of threads of execution compared with two available solutions. Median values and 95% confident intervals are represented.

where  $N$  is the number of threads and  $0 < p < 1$  represent the portion of the workload that has been parallelized.

By numerically fitting the curve given by equation (10) as a function of  $N$  to the data represented in figure 4, an estimation of the parallel fraction  $p$  is obtained. Results are shown in table 3, with  $p$  between 0.94 (94%) and 0.98 (98%) depending on the value of  $\alpha$  and  $\beta$ .

|                 | PDF  | CDF  |
|-----------------|------|------|
| $\alpha = 0.75$ | 0.98 | 0.98 |
| $\beta = 0.5$   |      |      |
| $\alpha = 1.5$  | 0.94 | 0.98 |
| $\beta = 0.5$   |      |      |

Table 3: Fracciones de paralelismo del método paralelo estimadas para los valores de  $\alpha$  y  $\beta$  indicados.

## 5. Usage of libstable

### 5.1. Compiling the library

The developed library can be easily compiled from the source code with the `make` command. **Libstable** depends on several numerical methods provided by the GSL, which must be installed in the system. After compilation, both shared (`libstable.so`) and static (`libstable.a`) versions of the library are produced. Several example programs to test the main functions of the library are also provided and compiled against the static version of the library by default.

Further documentation on the library functions can be found within the library distribution.

## 5.2. Usage in C/C++ developments

The next example program (`example.c`) illustrates how to use **libstable** to evaluate the PDF of an  $\alpha$ -stable distribution with given parameters and with 0 parameterization at a single point  $x$ :

```
#include <stdio.h>
#include <stable_api.h>

int main (void)
{
    double alpha = 1.25, beta = 0.5, sigma = 1.0, mu = 0.0;
    int parame = 0;
    double x = 10;
    StableDist *dist = stable_create(alpha,beta,sigma,mu,param);
    double pdf = stable_pdf_point(dist,x,NULL);
    printf("PDF(%g;%1.2f,%1.2f,%1.2f,%1.2f) = %1.15e\n",
           x,alpha,beta,sigma,mu,pdf);
    stable_free(dist);
    return 0;
}
```

The output of one execution of the example is:

```
PDF(10;1.25,0.50,1.00,0.00) = 3.225009046591384e-03
```

### *Compiling and linking*

If **libstable** header files and compiled library are not located on the standard search path of the compiler and linker respectively, their location must be provided as command line flag to compile and link the previous program. The program must also be linked to the GSL and system `math` libraries. Typical commands for compilation and static linking of a source file `example.c` with the GNU C compiler `gcc` is

```
$ gcc -I/path/to/headers -c example.c
$ gcc example.o /path/to/libstable/libstable.a -lgsl -lgslcblas -lm
```

When linking with the shared version of the library, path to `libstable.so` must be provided to the system's dynamic linker, typically by defining the shell variable `LD_LIBRARY_PATH`. The path to the shared library must also be provided when linking the program:

```
$ gcc -L/path/to/libstable example.o -lgsl -lgslcblas -lm -lstable
```

### *Setting general parameters*

Some general parameters can be adjusted on the library, such as precision required or available number of threads. These parameters are stored as global variables that can be read and modified with given functions described at continuation.

In multi-core systems, the number of threads of execution used by the library can be set and read by

```
stable_set_THREADS((unsigned int) THR);
unsigned int THREADS = stable_get_THREADS();
```

Example program illustrates the use of main features of `\pkg{libstable}`. In first place, a

```
\begin{Code}
  stable_setparams(dist,alpha,beta,sigma,mu,param);
```

When generating random samples, the function

```
  stable_rnd_seed(dist,seed);
```

initializes internal random generator can be initialized to a desired seed. This allows to reproduce results across different executions.

### 5.3. Usage in MATLAB environment

## 6. Conclusions

In this paper, a C/C++ library and MATLAB front-end to work with  $\alpha$ -stable distributions have been presented. The method of evaluation of the PDF, CDF and quantile function provided achieves a high precision, in most cases in the same order of magnitude than the widely acknowledged *ground truth* STABLE program. Based on the methods provided by the library, maximum likelihood and other estimation techniques based on the PDF can also be realized in reasonable times. If desired, less accurate estimates can also be obtained with much less time consumed.

The library developed implements parallelization techniques to carry out its computations. Hence, it can take full advantage of current multi-core systems. Besides, the use of appropriate quadrature techniques and strategies of integration allow to achieve an increment in performance with respect to current reference software when calculating the PDF with just one thread. As an example, when 16 threads of execution are available, a 25 fold increase in performance is achieved respect to current program STABLE (Nolan 2006) for PDF evaluation on the same machine.

Since the tools provided are in library form, they can be easily integrated in third party developments. The front-end provided allows to easily use the library from the user friendly graphical interface of the MATLAB (Inc. 2012) environment without appreciable loss of efficiency.

## References

- Arce GR (2005). *Nonlinear Signal Processing: A statistical Approach*. Wiley-Interscience, Hoboken, NJ, USA.

- Barney B (2011). *POSIX Threads Programming Tutorial*. Livermore, CA, Livermore, CA, USA. URL <https://computing.llnl.gov/tutorials/pthreads>.
- Belov I (2005). “On the computation of the probability density function of  $\alpha$ -stable distributions.” In *Proceedings of the 10th International Conference of Mathematical Modelling and Analysis*, pp. 333–341. Trakai, Lithuania.
- Borak S, Misiorek A, Weron R (2011). “Models for heavy-tailed asset returns.” In *Statistical tools for finance and insurance*, pp. 21–55. Springer, Berlin, Germany.
- Chambers JM, Mallows CL, Stuck BW (1976). “A method for simulating stable random variables.” *Journal of the American Statistical Association*, **71**(354), 340–344.
- Fama E (1965). “The behavior of stock-market prices.” *The journal of Business*, **38**(1), 34–105.
- Fama E, Roll R (1971). “Parameter estimates for symmetric stable distributions.” *Journal of the American Statistical Association*, **66**(334), 331–338.
- Fan Z (2006). “Parameter estimation of stable distributions.” *Communications in statistics-Theory and methods*, **35**(2), 245–255.
- Galassi M, Davies J, Theiler J, Gough B, Jungman G, Booth M, Rossi F (2009). *GNU Scientific Library Reference Manual*. Third edition. Network Theory Limited, Bristol, UK.
- Górska K, Penson KA (2011). “Lévy stable two-sided distributions: Exact and explicit densities for asymmetric case.” *Phys. Rev. E*, **83**, 061125. doi:10.1103/PhysRevE.83.061125. URL <http://link.aps.org/doi/10.1103/PhysRevE.83.061125>.
- Hill BM (1975). “A simple general approach to inference about the tail of a distribution.” *The annals of statistics*, pp. 1163–1174.
- Inc TM (2012). *MATLAB – The Language of Technical Computing, Release 2012a*. The MathWorks Inc., Natick, Massachusetts, USA. URL <http://www.mathworks.com/products/matlab/>.
- Koutrouvelis IA (1980). “Regression-type estimation of the parameters of stable laws.” *Journal of the American Statistical Association*, **75**(372), 918–928.
- Koutrouvelis IA (1981). “An iterative procedure for the estimation of the parameters of stable laws.” *Communications in Statistics - Simulation and Computation*, **10**(1), 17–28.
- Lévy P (1925). *Calcul des probabilités*. Gauthier-Villars, Paris, France.
- Liang Y, Chen W (2013). “A survey on computing Lévy stable distributions and a new MATLAB toolbox.” *Signal Processing*, **93**, 242–251.
- Mandelbrot BB, Wallis JR (1968). “Noah, Joseph, and operational hydrology.” *Water Resources Research*, **4**, 909–918.
- McCulloch JH (1986). “Simple consistent estimators of stable distribution parameters.” *Communications in Statistics – Simulation and Computation*, **15**(4), 1109–1136.

- Menn C, Rachev ST (2006). “Calibrated FFT-based density approximations for  $\alpha$ -stable distributions.” *Computational Statistics & Data Analysis*, **50**(8), 1891–1904.
- Mittnik S, Doganoglu T, Chenyao D (1999a). “Computing the probability density function of the stable Paretian distribution.” *Mathematical and Computer Modelling*, **29**(10-12), 235–240.
- Mittnik S, rachev S, Doganoglu T, Chenyao D (1999b). “Maximum likelihood estimation of stable Paretian models.” *Mathematical and Computer Modelling*, **29**(10–12), 275–293.
- Nolan JP (1997). “Numerical calculation of stable densities and distribution functions.” *Stochastic Models*, **13**(4), 759–774.
- Nolan JP (1999). “An algorithm for evaluating stable densities in Zolotarev’s (M) parameterization.” *Mathematical and Computer Modelling*, **29**(10-12), 229–233.
- Nolan JP (2001). “Maximum likelihood estimation and diagnostics for stable distributions.” In OE Barndorff-Nielsen, T Mikosch, SI Resnick (eds.), *Lévy Processes: Theory and Applications*, pp. 379–400. Birkhäuser, Boston, MA, USA.
- Nolan JP (2006). “Users guide for STABLE 4.0.xx.” URL <http://academic2.american.edu/~jpnolan/stable/stable.html>.
- Press W, Teukolsky S, Vetterling W, Flannery B (1994). *Numerical recipes in C: the art of scientific computing*. Second edition. Cambridge University Press, Cambridge, UK.
- Salas-Gonzalez D, Górriz J, Ramírez J, Schloegl M, Lang E, Ortiz A (2013). “Parameterization of the distribution of white and grey matter in MRI using the  $\alpha$ -stable distribution.” *Computers in Biology and Medicine*, **43**(5), 559–567. ISSN 0010-4825.
- Samorodnitsky G, Taqqu MS (1994). *Stable non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. Chapman and Hall/CRC, Boca Raton, CA, USA.
- Simmross-Wattenberg F, Asensio-Pérez JI, Casaseca-de-la-Higuera P, Martín-Fernández M, Dimitriadis IA, Alberola-López C (2011). “Anomaly detection in network traffic based on statistical inference and  $\alpha$ -stable modeling.” *IEEE Transactions on Dependable and Secure Computing*, **8**(4), 494–509. ISSN 1545-5971.
- Veillette MS (2010). “Alpha-stable distributions in MATLAB.” <http://math.bu.edu/people/mveillette/research.html>. Last accessed 1<sup>st</sup> July 2013.
- Weron R (1996). “On the Chambers-Mallows-Stuck method for simulating skewed stable random variables.” *Statistics & Probability Letters*, **28**(2), 165–171.
- Weron R (2004). “Computationally intensive Value at Risk calculations.” In J Gentle, W Härdle, Y Mori (eds.), *Handbook of Computational Statistics*, pp. 911–950. Springer, Berlin, Germany.
- Zolotarev VM (1986). *One-dimensional stable distributions*, volume 65 of *Translations of Mathematical Monographs*. American Mathematical Society, Ann Arbor, MI, USA.

**Affiliation:**

Javier Royuela-del-Val  
Laboratorio de Procesado de Imagen  
E.T.S. de Ingenieros de Telecomunicación  
Universidad de Valladolid  
Paseo de Belén, 15. 47011 Valladolid, España  
E-mail: [jroyval@lpi.tel.uva.es](mailto:jroyval@lpi.tel.uva.es)