Debora Chan*, Andrea Rey, Juliana Gambini and Alejandro C. Frery

Sampling from the $\mathcal{G}_I^0$ distribution

Abstract: Synthetic Aperture Radar (SAR) images are widely used in several environmental applications
because they provide information which cannot be obtained with other sensors. The $\mathcal{G}_I^0$ distribution is an important model for these images because of its flexibility (it provides a suitable way for modeling areas with different degrees of texture, reflectivity and signal-to-noise ratio) and tractability (it is closely related to the Snedekor-F, Pareto Type II, and Gamma distributions). Simulated data are important for devising tools for SAR image processing, analysis and interpretation, among other applications. We compare four ways for sampling data that follow the $\mathcal{G}_I^0$ distribution, using several criteria for assessing the quality of the generated data and the consumed processing time. The experiments are performed running codes in four different programming languages. The experimental results indicate that although there is no overall best method in all the considered programming languages, it is possible to make specific recommendations for each one.

Keywords: Random variable generation, SAR image modeling, programming languages, accuracy and goodness of fit

Introduction

Synthetic Aperture Radar (SAR) is a coherent radar of high resolution widely used in remote sensing. It works using an antenna in a movable platform. The antenna emits a signal, and the sensor measures the intensity and the delay between sent and returned signals. The image is then build with the energy backscattered by each point from the target. The independence of sunlight or other sources of illumination and that the signal is little affected by the presence of clouds and other adverse conditions that hamper the use of sensors which operate in the optical spectrum [32] are among the most important advantages of this kind of devices.

These images provide information which cannot be obtained with other remote sensors. They are widely used in many applications, including urban planning [13], building density estimation [44], urban-rural boundary identification [39], agricultural monitoring [3], crop discrimination [6], flooding mapping [26], oil releases detection [41], damage assessment in natural disasters such as Tsunamis [7, 37], earthquakes [9], volcanic activity, snow accumulation and landslides [45], and early detection of forest fires [5]. However, these imagesare contaminated by a deterministic interference pattern: speckle. This pattern can be conve-

***Corresponding author: Debora Chan,** Centro de Procesamiento de Señales e Imágenes (CPSI), Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, Medrano 951, Ciudad Autónoma de Buenos Aires, Argentina, e-mail: debiechan@gmail.com. http://orcid.org/0000-0003-0125-7345
Andrea Rey, Centro de Procesamiento de Señales e Imágenes (CPSI), Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, Medrano 951, Ciudad Autónoma de Buenos Aires, Argentina, e-mail: arey@utn.frba.edu.ar
Juliana Gambini, Departamento de Ingeniería Informática, ITBA & Departamento de Ingeniería en Computación, UNTreF, Instituto Tecnológico de Buenos Aires & Universidad Nacional de Tres de Febrero, Lavardén 315, Ciudad Autónoma de Buenos Aires, Argentina, e-mail: mgambini@itba.edu.ar. http://orcid.org/0000-0002-4534-1402
Alejandro C. Frery, Laboratório de Computação Científica e Análise Numérica – LaCCAN, Universidade Federal de Alagoas, Av. Lourival Melo Mota, s/n, Tabuleiro do Martins, Maceió, Brazil, e-mail: acfrery@laccan.ufal.br. http://orcid.org/0000-0002-8002-5341

niently modeled as stochastic, hence it has been called speckle noise. This noise also appears in laser and sonar imaging, medical ultrasound and optical coherence tomography. It is neither additive nor Gaussian and it is very difficult to eliminate, so statistical modeling is necessary.

In this work we use the multiplicative model for SAR image modeling, which appears to be an excellent choice [18]. It states that the observed data can be modeled by a random variable Z, which is the product of two independent random variables: X which describes the backscatter, and Y that models the speckle noise.

Following the multiplicative model, Frery, Müller, Yanasse and Sant'Anna [14] introduced the $\mathcal{G}_I^0$ distribution which has been widely used for SAR data analysis. It is referred to as a Universal Model because of its flexibility and tractability [25]. It provides a suitable way for modeling areas with different degrees of texture, reflectivity and signal-to-noise ratio. The $\mathcal{G}_I^0$ distribution is indexed by three parameters:

- α, related to the target texture,
- γ, related to the brightness and called scale parameter,
- L, which describes the signal-to-noise ratio and is termed "equivalent number of looks".

The two first may vary among positions, while the latter can be considered the same on the whole image.

The assessment of algorithm performance for automatic image interpretation and analysis is an important task. Simulation procedures typically in a Monte Carlo setup, allow inferring about performance in a wide variety of situations.

Several approaches based on the physical properties of the speckled data, have been developed to study SAR data simulation [22]. Among them we can mention simulation based on a fractal model of sea surface [21], ray tracing and diffraction geometrical theory [2, 12], and ab initio modeling [36].

Another scheme of simulating SAR data involves random number generation. References [19, 28, 30] use simulated SAR data for the assessment of edge detection techniques by means of non-parametric tests, stochastic distances and a geodesic distance, respectively. Gambini, Mejail, Jacobo-Berlles and Frery [17] examine the accuracy of contour detection procedures in SAR imagery using simulated data. Simulation is essential for the proposal of new speckle filters [8, 40], and for their quantitative evaluation [20, 27]. Simulation studies are used for assessing the performance of hypothesis test [29], classification techniques [25, 38] and parameter estimation [43].

In this work, we are interested in speckled data simulation under the $\mathcal{G}_I^0$ distribution. We discuss alternatives for this aim, and we propose a protocol for their assessment which includes error evaluation, execution times and a statistical significance study of differences.

In the existing literature, Devroye [10] discusses the assessment of non-uniform pseudorandom generators for parametric families in terms of portability, set-up and execution times. Code length and quality are mentioned. This approach assumes (i) that the underlying routines for obtaining such samples are error-free, and (ii) the availability of a perfect uniform pseudorandom deviates generator. The generation of random data from the $\mathcal{G}_I^0$ distribution has not been evaluated yet in this perspective.

We introduce four optional routines to generate $\mathcal{G}_I^0$ distributed data, based on Γ, χ^2, Fisher–Snedekor (F) and Pareto (P) distributions. The two first apply the multiplicative model, while the third and fourth require the inversion method. We assess these routines from the computational resources point of view, maximum value behavior, goodness-of-fit, moments and quantiles accuracy. We perform such assessment for Matlab, R, Ox, and Julia programming languages.

The paper unfolds as follows: in Section 1 we recall some properties of the $\mathcal{G}_I^0$ model and we refer to the single look case. In Section 2 we discuss techniques for sampling. Experiments and results are presented in Section 3. Finally, in Section 4 we expose our remarks and conclusions.

1 The $\mathcal{G}_I^0$ model

The statistical modeling of SAR data is strategic for the interpretation and understanding of the terrain electromagnetic scattering. Gao [18] details different models proposed for describing these kind of data and analyzes their properties and relationships.

Basic speckle properties attest that it follows a Gamma distribution with unitary mean, with density

$$f_Y(y; L) = \frac{L^L}{\Gamma(L)} y^{L-1} \exp\{-Ly\},$$

denoted by $Y \sim \Gamma(L, L)$. The physics of image formation imposes $L \geq 1$.

The model for X, the backscatter, may be any distribution with positive support. Frery, Müller, Yanasse and Sant'Anna [14] proposed using the Reciprocal Gamma law, a particular case of the Generalized Inverse Gaussian Distribution, which is characterized by the density

$$f_X(x; \alpha, \gamma) = \frac{\gamma^{-\alpha}}{\Gamma(-\alpha)} x^{\alpha-1} \exp\left\{-\frac{\gamma}{x}\right\},$$

where $\alpha < 0$ and $\gamma > 0$ are the texture and the scale parameters, respectively.

With this, the return Z follows a $\mathcal{G}_I^0(\alpha, \gamma, L)$ distribution, whose density is

$$f_Z(z) = \frac{L^L \Gamma(L-\alpha)}{\gamma^\alpha \Gamma(-\alpha)\Gamma(L)} \cdot \frac{z^{L-1}}{(\gamma + zL)^{L-\alpha}}, \tag{1.1}$$

where $-\alpha, \gamma, z > 0$ and $L \geq 1$.

The r-order moments of the $\mathcal{G}_I^0(\alpha, \gamma, L)$ distribution are

$$\mathrm{E}(Z^r) = \left(\frac{\gamma}{L}\right)^r \frac{\Gamma(-\alpha - r)}{\Gamma(-\alpha)} \frac{\Gamma(L + r)}{\Gamma(L)}, \tag{1.2}$$

provided $\alpha < -r$, and infinite otherwise. To simplify calculation and with the intention of obtaining comparable results, in most experiments we deal with a restricted case which assumes $E(Z) = 1$. With this, equation (1.2) implies the following relation between the parameters:

$$\gamma^* = -\alpha - 1.$$

We are interested in simulating the noisiest case which occurs when $L = 1$; it is called single-look and expression (1.1) becomes

$$f_Z(z) = \frac{-\alpha}{\gamma}\left(\frac{z}{\gamma} + 1\right)^{\alpha-1}. \tag{1.3}$$

In this case the cumulative distribution function is

$$F_Z(z) = 1 - \left(\frac{z}{\gamma} + 1\right)^{\alpha}.$$

Figure 1 illustrates three of these densities in linear and semilogarithmic scales, where the scale parameter has been set to have unitary mean. It can be observed the slow decay of the tail; this yields, as will be seen later, an outlier-prone distribution.

1.1 Properties

In this subsection, we detail some of the properties of the $\mathcal{G}_I^0$ law that are useful in our study. Consider the sample $Z_1, \ldots, Z_n$ of independent identically distributed $\mathcal{G}_I^0(\alpha, \gamma, 1)$ random variables, and sort it in non-decreasing order to obtain $Z_{1:n} \leq \cdots \leq Z_{n:n}$. Let

$$U_n = \min_{1 \leq i \leq n} \{Z_1, Z_2, \ldots, Z_n\} \quad \text{and} \quad V_n = \max_{1 \leq i \leq n} \{Z_1, Z_2, \ldots, Z_n\}$$

be the minimum and the maximum of the set of random variables, respectively. Then the probability density functions of these random variables are given by

$$f_{U_n}(u) = -n(1 - F_Z(u))^{n-1} f_Z(u) = -\frac{n\alpha}{\gamma}\left(1 + \frac{u}{\gamma}\right)^{n\alpha-1}$$

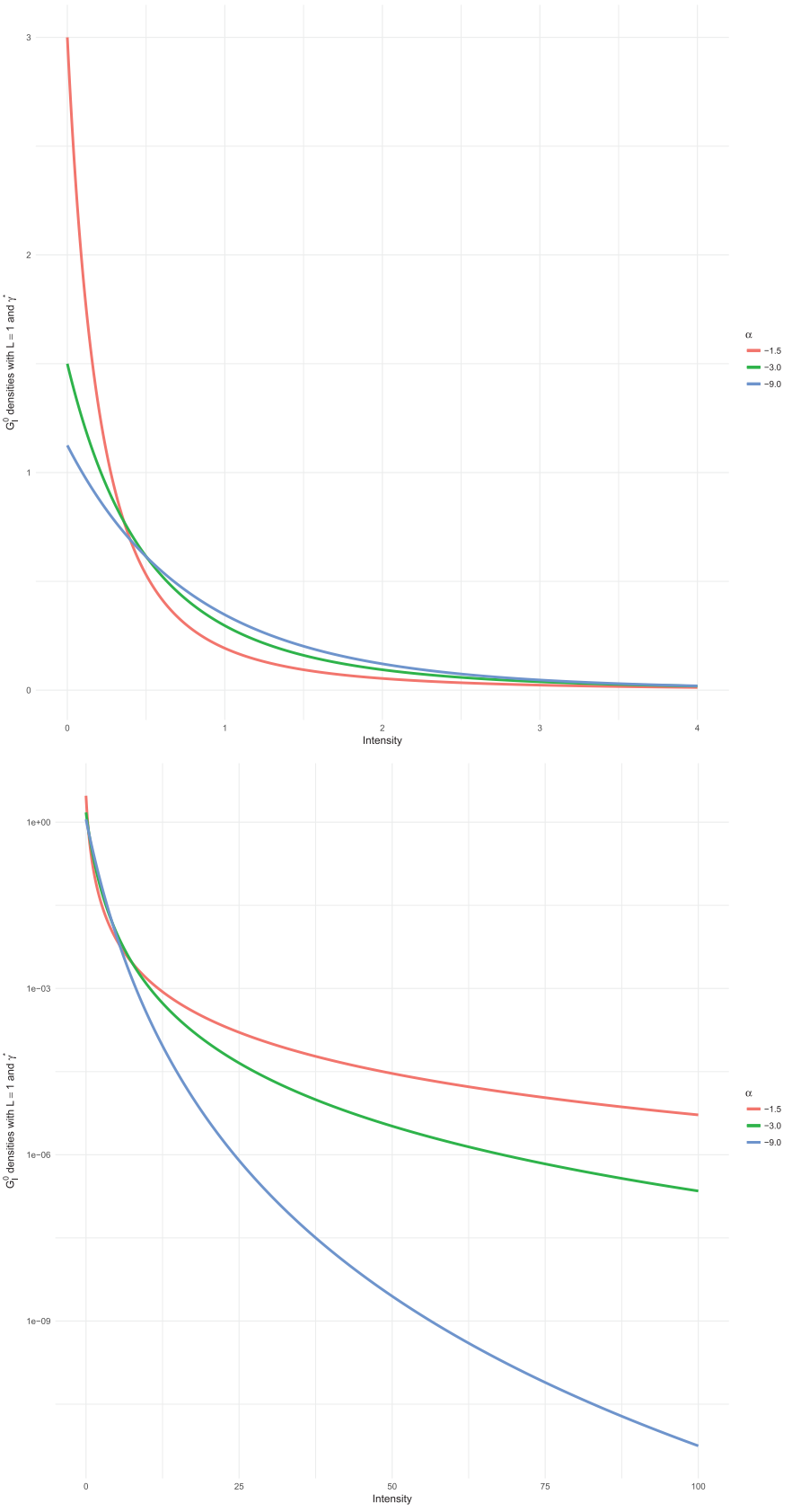

Figure 1: Linear and semilogarithmic scale densities of the $\mathcal{G}_I^0(\alpha, \gamma^*, 1)$ distribution compared with Exponential distribution (all with unitary mean).

and

$$f_{V_n}(v) = n(F_Z(v))^{n-1} f_Z(v) = -\frac{n\alpha}{\gamma}\left(1+\frac{v}{\gamma}\right)^{\alpha-1}\left[1-\left(1+\frac{v}{\gamma}\right)^{\alpha}\right]^{n-1},$$

where F_Z and f_Z are the cumulative distribution and the probability density function of the $\mathcal{G}_I^0$ law with $L = 1$, respectively.

The tails of a distribution are an important feature for the analysis of extreme events. In the article [30], several approaches for probability distributions classification are presented, according to the heaviness of their tails.

Gambini, Cassetti, Lucini and Frery [16] proved that the density function of the $\mathcal{G}_I^0(\alpha, \gamma, L)$ distribution is slow-varying at infinite and heavy tailed with tail index $1 - \alpha$, since its asymptotic behavior is comparable to the function $f(z) = z^{\alpha-1}$. In addition, it is prone to extreme observations.

1.2 Relationships

In this subsection we present the relationships that lead to alternative techniques for simulation. Mejail, Jacobo-Berlles, Frery and Bustos [25] proved that the family of $\mathcal{G}^0$ distributions is a reparametrization of the Fisher–Snedekor $\mathcal{F}$ law. Their cumulative distribution functions are related as

$$F_{\mathcal{G}_I^0(\alpha,\gamma,L)}(x) = \Upsilon_{2L,-2\alpha}\left(-\frac{\alpha x}{\gamma}\right),$$

where $\Upsilon_{s,t}$ is the cumulative distribution function of a Fisher–Snedekor's $\mathcal{F}_{s,t}$ distributed random variable with s and t degrees of freedom. Then this formula can be used to sample from the $\mathcal{G}_I^0(\alpha, \gamma, L)$ law, and to obtain quantiles.

The second kind Generalized Pareto distribution [31] is characterized by the following probability density function:

$$f_{\mathrm{P}}^{\mathrm{II}}(x) = \frac{\beta}{\sigma}\left(1+\frac{x-\mu}{\sigma}\right)^{-\beta-1}, \tag{1.4}$$

with $x > \mu$, $\mu \in \mathbb{R}$, $\sigma > 0$ and $\beta > 0$. Comparing (1.4) and (1.3), we notice that the single-look $\mathcal{G}_I^0$ distribution is a particular case of the second kind Generalized Pareto distribution: the one with $\mu = 0$, $\sigma = \gamma$, and $\beta = -\alpha$.

2 Data generation techniques

Four generation routines are implemented using the fact that the random variable $Z \sim \mathcal{G}_I^0(\alpha, \gamma, L)$ can be defined as the product of two independent random variables. In the following, we cite relations between distributions involved in the forthcoming techniques.

- If $X \sim \Gamma(\alpha, \gamma)$, then $\frac{1}{X} \sim \Gamma^{-1}(-\alpha, \gamma^{-1})$, where Γ^{-1} denotes the Inverse Gamma distribution.
- The χ^2 distribution is a particular case of a Γ distribution, if $X \sim \chi_n^2$, then $X \sim \Gamma(\frac{n}{2}, 2)$.
- $X \sim \Gamma(m, n)$ implies that $aX \sim \Gamma(m, \frac{n}{a})$ for $a > 0$.

Thus, we propose the following two ways of data generation:

- Γ *generation routine:*

$$Z = Y\frac{1}{X},$$

  where X and Y are independent random variables such that $X \sim \Gamma(-\alpha, \gamma)$ and $Y \sim \Gamma(1, 1)$.
- χ^2 *generation routine:*

$$Z = \gamma X\frac{1}{Y},$$

  where X and Y are independent random variables such that $X \sim \chi_2^2$ and $Y \sim \chi_{-2\alpha}^2$.

Multiplicative model	Inversion method
Γ	F
χ^2	P

Table 1: Classification of generation routines.

The following generation routine is based on the construction of Fisher–Snedekor $\mathcal{F}$ random variables as the quotient of independent χ^2 deviates scaled by their degrees of freedom:

- *F generation routine:*

$$Z = -\frac{\gamma}{\alpha}F,$$

where F is an $\mathcal{F}(2, -2\alpha)$ distributed random variable.

The fact that the $\mathcal{G}_I^0(\alpha, \gamma, 1)$ distribution is a particular case of Generalized Pareto Type II distribution leads to another manner of generation for the single look case:

- *P generation routine:*

$$Z \sim \mathrm{P(II)}(0, \gamma, -\alpha),$$

with P(II) denoting Pareto Type II distribution.

In all cases, $-\alpha, \gamma > 0$. Table 1 shows the classification of the methods of random number generation.

3 Experiments and results

The α parameter of the $\mathcal{G}_I^0$ distribution can be interpreted in terms of the texture of the image region, see [14] for details. A very textured area such as an urban spot is usually related to $\alpha \in (-3, 0)$. The level of texture declines as the value of this parameter decreases, for $\alpha \in (-6, -3]$ we may be in presence of a forest, while for $\alpha \in (-\infty, -6]$ we usually have pasture or crops. Since we are interested in studying all types of areas, we consider several representative values of the texture and brightness parameters (α, γ) in each experiment.

We report results in the following programming languages, which are used in image processing and remote sensing applications:

- **R** [47], version 3.3.1, is a software environment for statistical computing and image processing, among other applications. It runs on several platforms and provides statistical and graphical tools.
- **MatLab** [46], version R2017a, is a numerical computing framework. It has its own programming language and an image processing toolbox. MatLab allows a variety of matrix operations, and the creation of user interfaces.
- **Ox** [11], version 7.00, is an object-oriented programming language with an extensive statistical function library. It is available for several platforms.
- **Julia** [4], version 0.5.0, is a high-performance dynamic programming language for technical computing. It provides a compiler, distributed parallel execution with good numerical accuracy and a library of mathematical functions.

All these frameworks are open source except for the second one.

3.1 Execution times

An important feature to be analyzed is the computational time consumed in each routine. We examined cases involving the following parameters: $(\alpha, \gamma) \in \{-8, -7, -6, -5, -4, -3, -2\} \times \{0.1, 1, 5, 10, 25, 50, 100\}$, for each one of the considered languages using a computer with processor Intel© Core™, i7-6700K CPU 3.4 GHz, 16 GB RAM, System Type 64 bit operating system. Figure 2, exhibits the time consumed to generate samples of size 10^6 with these parameters, by each generation routine and for each programming language.

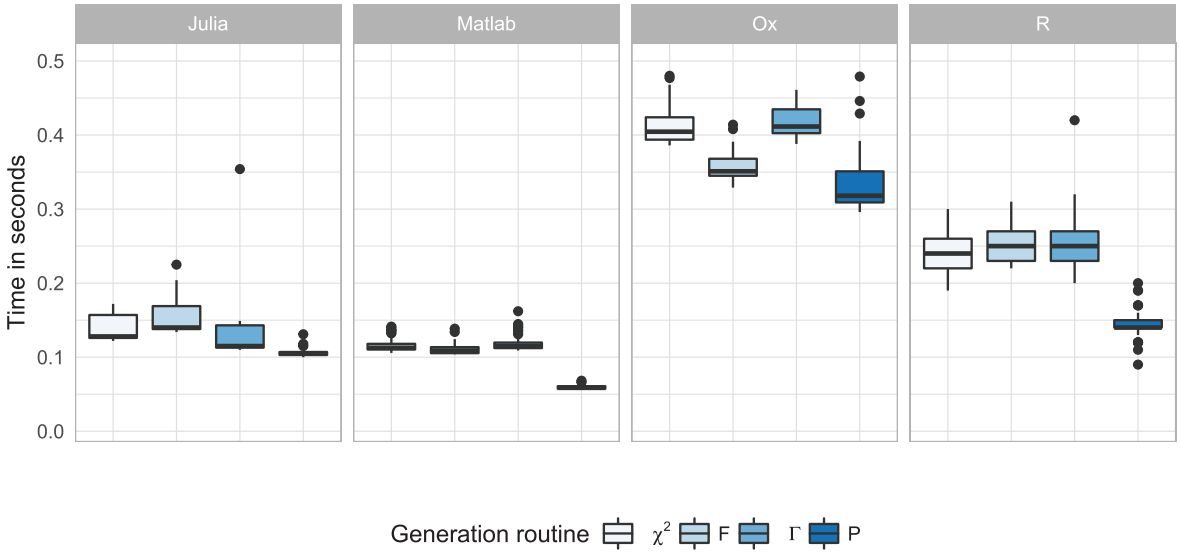

Figure 2: Processing time according to programming language.

In order to compare the computational times of generation routines and programming languages, we replicate this experience twice to apply Friedman test [15], first using generation routines as blocks and languages as factors and then viceversa. It can be observed that P generation routine consumes the shortest computational time in all the programming languages. Difference between the mean times consumed by P and Γ generation routines turn out significant in Friedman post-hoc pairwise comparisons (p-value = 0.065).

Difference among programming languages shows statistical significance (p-value = 0.007). After post-hoc pairwise comparisons we conclude that the principal difference among processing time according to languages is due to Matlab and Ox.

Listings 1, 2, 3 and 4 show the codes for data generation in R, MatLab, Ox and Julia programming languages, respectively, denoting `alpha` and `gam` the texture and the scale parameters, respectively, `n` the sample size, and `gen` the selected generation routine.

Listing 1: $\mathcal{G}_I^0$ generation function in R.

```
library(stats4)
library(rmutil)
rGI0= function(alpha,gam,n,L = 1,gen){
  switch(gen,
         "Pareto" = rpareto(n,-alpha,gam),
         "Gamma" = rgamma(n,L,L)/rgamma(n,-alpha,gam),
         "Chi2" = gam*rchisq(n,2)/rchisq(n,-2*alpha),
         "F-Snedecor" = (-1)*gam*rf(n,2,-2*alpha)/ alpha)
}
```

Listing 2: $\mathcal{G}_I^0$ generation function in Matlab.

```
function data = rGI0(alpha,gam,n,gen,varargin)
    L = 1;
    fprintf('Total␣number␣of␣inputs␣=␣%d\n',nargin);
    if nargin == 4
        gen = 'Pareto';
    end
        switch gen
```

```
            case 'Gamma'
                data=gamrnd(L,1/L,n,1)./gamrnd(-alpha,1/gam,n,1);
            case 'Chi2'
                data=gam*chi2rnd(2,n,1)./chi2rnd(-2*alpha,n,1);
            case 'F-Snedecor'
                data=(-1)*gam*frnd(2,-2*alpha,n,1)./alpha;
            case 'Pareto'
                data=gprnd(-1/alpha,-gam/alpha,0,n,1);
        end
```

Listing 3: $\mathcal{G}_I^0$ generation function in Ox.

```
#include <oxstd.h>
#include <oxprob.h>

rGI0(const alpha,const gam,const n,const gen)
        {decl data,i,XG,YG,XC,YC,XF,XU,DG,L = 1;
        switch (gen)
            {
                case "Gamma":
                                YG = rangamma(n,1,L,L);
                                XG = rangamma(n,1,-alpha,gam);
                                data = YG./XG;
                    break;
                case "Chi2":
                                XC=ranchi(n,1,2);
                                YC=ranchi(n,1,-2*alpha);
                                data = gam * XC./YC;
                                break;
                case "F-Snedecor":
                                XF=ranf(n,1,2,-2*alpha);
                                data = -(gam/ alpha) * XF;
                    break;
                case "Pareto":
                                XU=ranu(n,1);
                                data=gam*((1-XU).^(1/alpha)-1);
                    break;
            }
        return data; }
```

Listing 4: $\mathcal{G}_I^0$ generation function in Julia.

```
Pkg.add("Distributions")
Pkg.add("Switch")

using Distributions
using Switch
function rGI0(alpha,gam,n,gen)
  data=ones(n);
  L = 1;
    @switch gen begin
```

```
        @case "Gamma"
                        fg=Gamma(L,1/L)
                        dg=Gamma(-alpha,1/gam)
                        YG=rand(fg,n)
                        XG=rand(dg,n)
                        data=YG./XG
          break
        @case "Chi2"
                        fc=Chisq(2)
                        dc=Chisq(2*alpha)
                        XC=rand(fc,n)
                        YC=rand(dc,n)
                        data=gam.*XC./YC
          break
        @case "F-Snedecor"
                        ff=FDist(2,2*alpha)
                        XF=rand(ff,n)
                        data=gam.*XF./alpha
          break
        @case "Pareto"
                        XU=rand(n)
                        data=gam.*((1-XU).^(-1/alpha)-1)
    end
    return data
end
```

3.2 Moments accuracy

We generated 1000 random samples of size 500, with the following parameters:

$$(\alpha, \gamma) \in \{-8, -5, -3\} \times \{0.1, 1, 10, 100, 1000\}.$$

For each one we compute their mean and variance values. The second moment is not defined for $\alpha = -1.5$, so we do not take it into account in this section. The true values of mean and variance of the $\mathcal{G}_I^0$ distribution are calculated using equation (1.2). Then we computed the deviation between the obtained ($\hat{\theta}$) and the true values (θ), considering the following error measures, where n is the sample size:

- *Mean square error* (MSE)

$$\widehat{\mathrm{MSE}}(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{\theta}_i - \theta)^2.$$

- *Mean absolute error* (MAE)

$$\widehat{\mathrm{MAE}}(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{n}|\hat{\theta}_i - \theta|.$$

- *Maximum absolute error* (MxAE)

$$\widehat{\mathrm{MxAE}}(\hat{\theta}) = \max_{1\le i\le n}\{|\hat{\theta}_i - \theta|\}.$$

MAE is widely applied in image processing, since its memory requirements are noticeably smaller than the ones MSE needs [35].

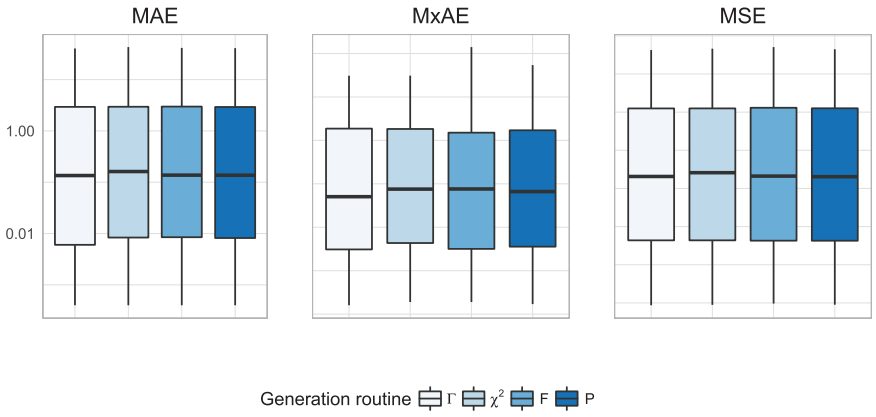

Figure 3: Mean errors according to the generation routine.

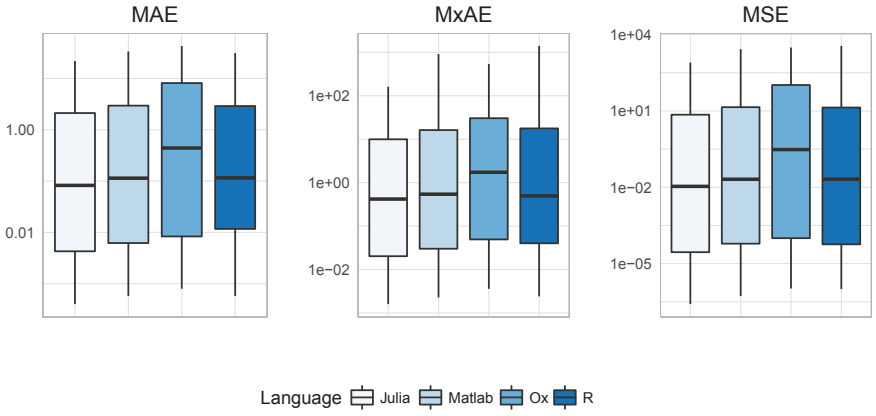

Figure 4: Mean errors according to the programming language.

3.2.1 Error comparison

Mean accuracy. We applied the Kruskal–Wallis (KW) test [23] to compare mean errors considering as factors: generation routine and programming language. Post hoc comparisons are conducted when statistical significance appeared. Clear differences are observed between Julia and the other languages and also between Ox and R, for the MAE measure. However, there is no statistical significant difference between Matlab and R. The KW test also shows significant differences for MxAE by programming language. Post hoc pairwise comparisons indicate that Ox has the lowest mean accuracy, Matlab and R match, and Julia presents the lowest errors. Similar results are found for MSE. Figures 3 and 4 show the distribution of error measures for the mean sample value, in logarithmic scale, according to the generation routines and to the programming language, respectively.

Variance accuracy. In this case MAE, MxAE and MSE differences are statistically significant according to programming language. Figures 5 and 6 show the distribution of error measures, in logarithmic scale, for the variance according to generation routine and to programming languages, respectively. In the case of MAE, we noticed the presence of two subgroups: Ox-R and Julia-Matlab.

3.3 Goodness of fit

In this subsection we assess the goodness of fit of the data provided by the proposed generation routines using several values of α and the corresponding γ^*.

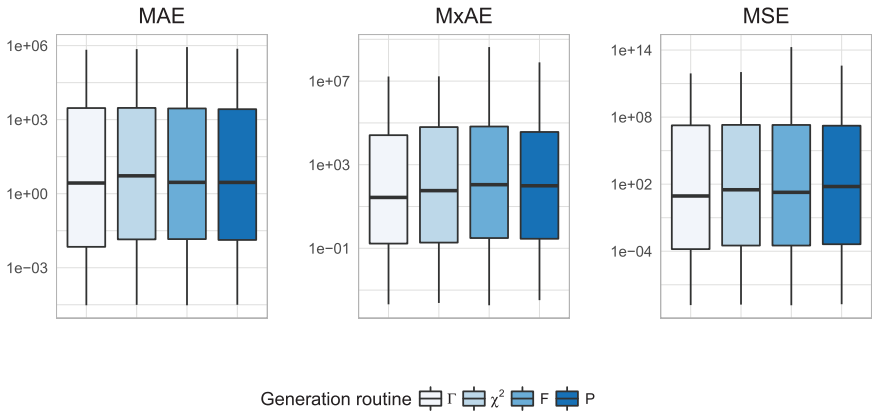

Figure 5: Variance errors according to generation routine.

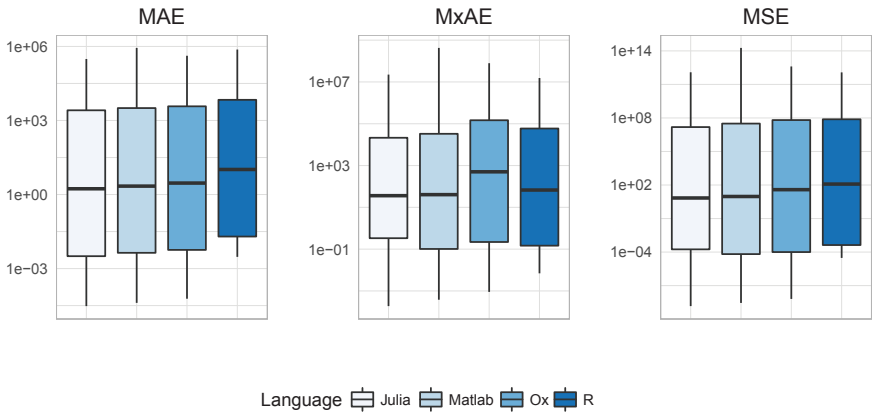

Figure 6: Variance errors according to language.

3.3.1 Raw data

For all the considered programming languages, we generated 1000 samples of size 1000 with each of the proposed generation routines using $\alpha \in \{-8, -5, -3, -1.5\}$. In order to assess the goodness of fit for each sample, we applied two classical criteria: Kolmogorov–Smirnov [24] and Anderson–Darling [33] tests. The first test is based on the maximum difference between the empirical and the cumulative distribution functions (equation (1.1)). One of its properties is that it is less sensitive in the tails of the distribution, but its statistic distribution does not depend on the underlying distribution when the true parameters are known. The second test gives more weight to the tails improving, thus, the KS test discriminatory ability. The tests are applied with significance level $\eta = 0.05$. Table 2 shows the average p-values obtained for each method, framework and test. Rejection level of the KS and the AD tests for the raw data has not significant differences from the theoretical fixed level, neither for programming language nor for generator.

3.3.2 Maximum distribution

By absolutely outlier-prone and slowly varying at infinite properties of the $\mathcal{G}_I^0$ distribution [16, 34], we analyze the extreme value data performance produced by each generation routine. For $n \in \{50, 100, 500, 1000\}$, we generated 1000 samples of size 1000 of n-size sample maxima, and we evaluate the goodness of fit of their distribution applying KS and AD tests. Tables 3 and 4 show the sample p-values at the $\eta = 0.05$ level. There are no significant deviations from the theoretical level of KS and AD tests rejection levels for the sample maximum , neither for programming language nor for generator.

Test	Language	Γ	χ^2	F	P
KS	R	0.0498	0.0495	0.0485	0.0470
	Matlab	0.0420	0.0420	0.0420	0.0420
	Ox	0.0410	0.0520	0.0440	0.0523
	Julia	0.0490	0.0409	0.0310	0.0530
AD	R	0.0493	0.0498	0.0490	0.0488
	Matlab	0.0430	0.0430	0.0430	0.0430
	Ox	0.0350	0.0523	0.0440	0.0578
	Julia	0.0510	0.0470	0.0510	0.0570

Table 2: KS and AD tests results.

KS test		Sample size				Average
Language	Routine	50	100	500	1000	
R	Γ	0.049	0.046	0.050	0.051	0.049
	χ^2	0.048	0.047	0.051	0.056	0.050
	F	0.053	0.075	0.042	0.046	0.054
	P	0.041	0.049	0.053	0.047	0.048
Matlab	Γ	0.053	0.050	0.043	0.050	0.049
	χ^2	0.048	0.049	0.046	0.044	0.047
	F	0.052	0.046	0.046	0.047	0.048
	P	0.052	0.057	0.053	0.048	0.053
Ox	Γ	0.054	0.045	0.051	0.049	0.050
	χ^2	0.052	0.041	0.052	0.052	0.051
	F	0.048	0.043	0.051	0.052	0.048
	P	0.049	0.050	0.047	0.056	0.051
Julia	Γ	0.048	0.043	0.051	0.052	0.048
	χ^2	0.053	0.047	0.047	0.047	0.048
	F	0.045	0.047	0.049	0.047	0.047
	P	0.054	0.058	0.070	0.052	0.059

Table 3: p-values of the KS test for the maxima.

AD test		Sample size				Average
Language	Routine	50	100	500	1000	
R	Γ	0.049	0.044	0.054	0.056	0.051
	χ^2	0.049	0.052	0.047	0.057	0.051
	F	0.053	0.065	0.047	0.047	0.053
	P	0.038	0.048	0.059	0.048	0.048
Matlab	Γ	0.051	0.048	0.047	0.050	0.049
	χ^2	0.047	0.051	0.051	0.047	0.049
	F	0.052	0.045	0.047	0.049	0.048
	P	0.051	0.051	0.049	0.055	0.052
Ox	Γ	0.053	0.045	0.047	0.048	0.048
	χ^2	0.054	0.044	0.055	0.054	0.052
	F	0.048	0.046	0.052	0.055	0.050
	P	0.053	0.052	0.051	0.053	0.052
Julia	Γ	0.053	0.054	0.054	0.043	0.051
	χ^2	0.050	0.048	0.045	0.038	0.045
	F	0.046	0.047	0.052	0.046	0.048
	P	0.056	0.060	0.057	0.045	0.055

Table 4: p-values of the AD test for the maxima.

3.3.3 Minimum distribution

Allende, Frery, Galbiati and Pizarro [1], while seeking for a robust estimator of the texture in a closely related model, noted that large outliers may cause problems in homogeneous areas, whereas very small ones are critical in extremely heterogeneous areas. The reason for this is that the ML estimator depends on the sum of the logarithm of the sample observations. This fact suggests that very small outliers may hamper the performance of the ML estimator. Therefore, we study here the sample minimum distribution behavior. We considered $n \in \{50, 100, 500, 1000\}$ and generated 1000 samples of size 1000 of n-size sample minima. Then, applying KS and AD tests, we evaluate the goodness of fit of their distribution. Tables 5 and 6 show the sample p-values

KS test		Sample size				Average
Language	Routine	50	100	500	1000	
R	Γ	0.050	0.049	0.046	0.047	0.048
	χ^2	0.050	0.049	0.046	0.047	0.048
	F	0.049	0.049	0.043	0.047	0.047
	P	0.055	0.044	0.050	0.050	0.050
Matlab	Γ	0.048	0.048	0.052	0.047	0.049
	χ^2	0.055	0.042	0.045	0.048	0.048
	F	0.044	0.043	0.051	0.041	0.045
	P	0.045	0.050	0.055	0.042	0.048
Ox	Γ	0.092	0.088	0.096	0.088	0.091
	χ^2	0.065	0.061	0.065	0.074	0.066
	F	0.071	0.066	0.076	0.068	0.071
	P	0.054	0.067	0.064	0.085	0.068
Julia	Γ	0.055	0.048	0.053	0.056	0.053
	χ^2	0.050	0.046	0.051	0.049	0.049
	F	0.046	0.048	0.049	0.051	0.048
	P	0.049	0.047	0.050	0.046	0.048

Table 5: p-values of the KS test for the minima.

AD test		Sample size				Average
Language	Routine	50	100	500	1000	
R	Γ	0.047	0.050	0.050	0.051	0.050
	χ^2	0.047	0.050	0.050	0.051	0.050
	F	0.049	0.051	0.047	0.049	0.049
	P	0.055	0.047	0.052	0.050	0.051
Matlab	Γ	0.049	0.046	0.052	0.048	0.049
	χ^2	0.053	0.043	0.050	0.055	0.050
	F	0.048	0.049	0.050	0.047	0.048
	P	0.047	0.050	0.055	0.044	0.049
Ox	Γ	0.086	0.077	0.086	0.077	0.081
	χ^2	0.059	0.057	0.058	0.070	0.061
	F	0.073	0.062	0.073	0.063	0.068
	P	0.051	0.064	0.067	0.082	0.066
Julia	Γ	0.051	0.048	0.053	0.054	0.051
	χ^2	0.053	0.049	0.049	0.047	0.049
	F	0.051	0.049	0.050	0.054	0.051
	P	0.049	0.049	0.055	0.046	0.050

Table 6: p-values of the AD test for the minima.

at the $\eta = 0.05$ level. As in the previous cases, we compared the rejection level with the p-values and we did not find any significant difference between them neither by language nor by generation routine.

3.3.4 Quantile confidence intervals

Considering each generated sample as mentioned in Sec. 3.3, we register the third quartile, the 90th percentile and the limits of 95 % level confidence interval for them based on a Bootstrap method [42]. We replicated this experiment using all the simulation techniques and all the selected programming languages. Tables 7 and 8 show the results.

Language	Routine	α				Average
		−8	−5	−3	−1.5	
R	Γ	0.960	0.957	0.946	0.955	0.9545
	χ^2	0.960	0.959	0.946	0.960	0.9470
	F	0.946	0.947	0.948	0.956	0.9493
	P	0.947	0.936	0.936	0.936	0.9388
Matlab	Γ	0.937	0.942	0.944	0.946	0.9423
	χ^2	0.937	0.942	0.944	0.946	0.9423
	F	0.937	0.942	0.944	0.946	0.9423
	P	0.942	0.942	0.942	0.942	0.9420
Ox	Γ	0.955	0.958	0.945	0.961	0.9548
	χ^2	0.961	0.948	0.959	0.963	0.9578
	F	0.953	0.951	0.954	0.950	0.9520
	P	0.942	0.958	0.944	0.943	0.9468
Julia	Γ	0.948	0.937	0.958	0.967	0.9525
	χ^2	0.969	0.956	0.949	0.945	0.9548
	F	0.950	0.948	0.953	0.937	0.9470
	P	0.948	0.944	0.951	0.953	0.9490

Table 7: q_3 confidence interval coverage.

Language	Routine	α				Average
		−8	−5	−3	−1.5	
R	Γ	0.969	0.951	0.952	0.946	0.955
	χ^2	0.969	0.942	0.952	0.951	0.954
	F	0.954	0.960	0.950	0.961	0.956
	P	0.952	0.961	0.961	0.961	0.959
Matlab	Γ	0.947	0.954	0.952	0.936	0.945
	χ^2	0.947	0.945	0.952	0.936	0.945
	F	0.947	0.945	0.952	0.936	0.945
	P	0.937	0.937	0.937	0.937	0.937
Ox	Γ	0.949	0.954	0.961	0.957	0.955
	χ^2	0.953	0.959	0.961	0.963	0.959
	F	0.945	0.952	0.946	0.946	0.947
	P	0.955	0.960	0.960	0.949	0.956
Julia	Γ	0.930	0.953	0.959	0.955	0.949
	χ^2	0.962	0.968	0.959	0.955	0.961
	F	0.960	0.943	0.959	0.959	0.955
	P	0.950	0.944	0.948	0.959	0.950

Table 8: p_{90} confidence interval coverage.

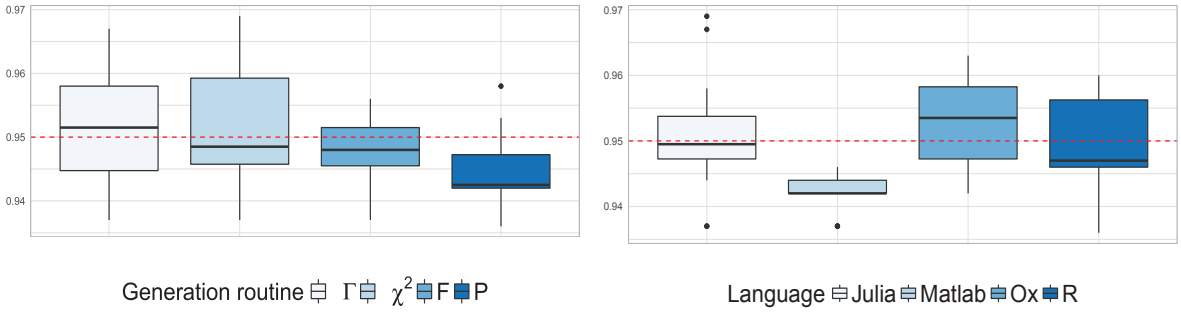

Figure 7: Percentage coverage for p_{75}.

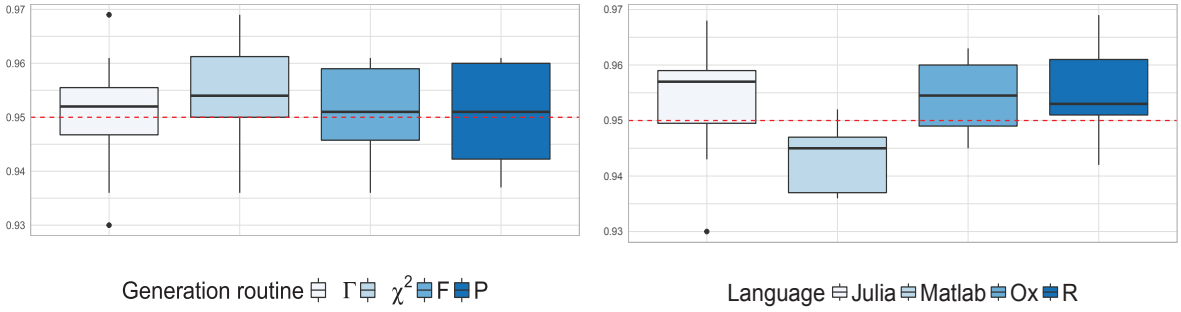

Figure 8: Percentage coverage for p_{90}.

Proportion differences for confidence interval coverage. We first observed that the coverage percentage does not depend on the texture parameter. We then wondered if the percentile was important and found significant differences between p_{75} and p_{90}. This fact led us to separate the study for each percentile and pay attention to the differences according to generation routine and language. The interaction between generation routine and language is not significant. The KW test pointed out significant differences among coverage percentage for p_{75} by generation routine and by language. Post hoc comparisons indicated that both χ^2 and P routines are different for p_{75} coverage. In the case of p_{90}, only the differences according language have statistical significance. This results can be appreciated in Figures 7 and 8. The red dotted line is the fixed confidence level.

4 Discussion and conclusions

In this section, we present conclusions and give recommendations for the selection of suitable generators according to the programming language.

(1) P and F routines are simpler than the others in terms of language programming code, length and complexity, since they are based on the inversion method for the single look case.
(2) If we prioritize the processing time, the P routine is the best since it consumes the shortest time in all the programming languages.
(3) According to the programming language, the difference between Matlab and Ox is significant.
(4) Focusing on data goodness of fit, minima and maxima samples, there was no difference neither by language nor by generation routine.
(5) We detected two groups: Ox & R and Julia & Matlab, with respect to the moment adjustment.
(6) Conforming to the confidence intervals coverage percentage, χ^2 and P routines are significantly different for the third quartile and there is no difference between generation routines for the 90th percentile. In both cases, there is difference between coverage by programming language. For this reason, Ox, R and Julia outperform Matlab.

References

[1] H. Allende, A. C. Frery, J. Galbiati and L. Pizarro, M-Estimators with asymmetric influence functions: The $\mathcal{G}_A^0$ distribution case, *J. Stat. Comput. Simul.* **76** (2006), no. 11, 941–956.

[2] S. Auer, M. Horning, I. Schmitt and P. Reinartz, Simulation-based interpretation and alignment of high-resolution optical and SAR images, *IEEE J. Appl. Earth Observ. Remote Sensing* **10** (2017), 4779–4793.

[3] M. Barber, F. Grings, P. Perna, M. Piscitelli, M. Maas, C. Bruscantini, J. Jacobo-Berlles and H. Karszenbaum, Speckle noise and soil heterogeneities as error sources in a Bayesian soil moisture retrieval scheme for SAR data, *IEEE J. Appl. Earth Observ. Remote Sensing* **5** (2012), no. 3, 942–951.

[4] J. Bezanson, A. Edelman, S. Karpinski and V. Shah, *Julia: A Fresh Approach to Numerical Computing*, Cornell University Library, New York, 2014.

[5] S. Canale, A. De Santis, D. Iacoviello, F. Pirri and S. Sagratella, Integrating X-SAR images and anthropic factors for fire susceptibility assessment, in: *IEEE International Geoscience and Remote Sensing Symposium*, IEEE Press, Piscataway (2011), 818–821.

[6] S. Chen, Y. Li and X. Wang, Crop discrimination based on polarimetric correlation coefficients optimization for PolSAR data, *Int. J. Remote Sensing* **36** (2015), no. 16, 4233–4249.

[7] S. W. Chen and M. Sato, Tsunami damage investigation of built-up areas using multitemporal spaceborne full polarimetric SAR images, *IEEE Trans. Geosci. Remote Sensing* **51** (2013), no. 4, 1985–1997.

[8] D. Cozzolino, S. Parrilli, G. Scarpa, G. Poggi and L. Verdoliva, Fast adaptive nonlocal SAR despeckling, *IEEE Trans. Geosci. Remote Sensing Lett.* **11** (2014), 524–528.

[9] F. Dell'Acqua and P. Gamba, Remote sensing and earthquake damage assessment: Experiences, limits, and perspectives, *Proc. IEEE* **100** (2012), no. 10, 2876–2890.

[10] L. Devroye, *Non-Uniform Random Variate Generation*, Springer, New York, 1986.

[11] J. A. Doornik and M. Ooms, *Introduction to Ox*, Timberlake Consultants Press, London, 2006.

[12] E. M. El-Desouki, K. F. A. Hussien and H. M. El-Hennawy, Electromagnetic simulation for land imaging using fully polarimetric SAR system, in: *34th National Radio Science Conference*, IEEE Press, Piscataway (2017), 132–141.

[13] T. Esch, M. Schmidt, M. Breunig, A. Felbier, H. Taubenböck, W. Heldens, C. Riegler, A. Roth and S. Dech, Identification and characterization of urban structures using VHR SAR data, in: *IEEE International on Geoscience and Remote Sensing Symposium* (IGARSS), IEEE Press, Piscataway (2011), 1413–1416.

[14] A. Frery, H. Müller, C. Yanasse and S. Sant'Anna, A model for extremely heterogeneous clutter, *IEEE Trans. Geosci. Remote Sensing* **35** (1997), no. 3, 648–659.

[15] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *J. Amer. Statist. Assoc.* **32** (1937), no. 200, 675–701.

[16] J. Gambini, J. Cassetti, M. Lucini and A. Frery, Parameter estimation in SAR imagery using stochastic distances and asymmetric kernels, *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sensing* **8** (2015), no. 1, 365–375.

[17] J. Gambini, M. Mejail, J. Jacobo-Berlles and A. Frery, Accuracy of local edge detection in speckled imagery, *Stat. Comput.* **18** (2008), no. 1, 15–26.

[18] G. Gao, Statistical modeling of SAR images: A survey, *Sensors* **10** (2010), no. 1, 775–795.

[19] E. Girón, A. C. Frery and F. Cribari-Neto, Nonparametric edge detection in speckled imagery, *Math. Comput. Simulation* **82** (2012), no. 11, 2182–2198.

[20] L. Gomez, R. Ospina and A. C. Frery, Unassisted quantitative evaluation of despeckling filters, *Remote Sensing* **9** (2017), DOI 10.3390/rs9040389.

[21] P. Guccione, L. Mascolo, G. Nico, A. Pelusi and M. Zonno, SAR image simulation of ocean environment and detection of oil slicks, in: *10th European Conference on Synthetic Aperture Radar*, IEEE Press, Piscataway (2014), 1–4.

[22] M. Kiemer and H. Breit, Eficient evaluation of multichanel SAR data recombination filters, *IEEE Trans. Geosci. Remote Sensing* **55** (2017), 6277–6286.

[23] W. H. Kruskal and W. A. Wallis, Use of ranks in one-criterion variance analysis, *J. Amer. Statist. Assoc.* **47** (1952), no. 260, 583–621.

[24] F. Massey, Jr., The Kolmogorov–Smirnov test for goodness of fit, *J. Amer. Statist. Assoc.* **46** (1951), no. 253, 68–78.

[25] M. Mejail, J. C. Jacobo-Berlles, A. C. Frery and O. H. Bustos, Classification of SAR images using a general and tractable multiplicative model, *Int. J. Remote Sensing* **24** (2003), no. 18, 3565–3582.

[26] R. T. Melrose, R. T. Kingsford and A. K. Milne, Using radar to detect flooding in arid wetlands and rivers, in: *IEEE International Geoscience and Remote Sensing Symposium* (IGARSS), IEEE Press, Piscataway (2012), 5242–524.

[27] E. Moschetti, M. G. Palacio, M. Picco, O. H. Bustos and A. C. Frery, On the use of Lee's protocol for speckle-reducing techniques, *Latin Amer. Appl. Res.* **36** (2006), no. 2, 115–121.

[28] J. Naranjo-Torres, J. Gambini and A. C. Frery, The geodesic distance between $\mathcal{G}_I^0$ models and its application to region discrimination, *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sensing* **10** (2017), no. 3, 987–997.

[29] A. D. C. Nascimento, R. J. Cintra and A. C. Frery, Hypothesis testing in speckled data with stochastic distances, *IEEE Trans. Geosci. Remote Sensing* **48** (2010), no. 1, 373–385.

[30] A. D. C. Nascimento, M. M. Horta, A. C. Frery and R. J. Cintra, Comparing edge detection methods based on stochastic entropies and distances for PolSAR imagery, *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sensing* **7** (2014), no. 2, 648–663.
[31] M. Newman, Power laws, Pareto distributions and Zipf's law, *Contemp. Phys.* **46** (2005), 323–351.
[32] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images*, Artech House, Boston, 1998.
[33] N. Razali and Y. Wah, Power comparisons of Shapiro–Wilk, Kolmogorov–Smirnov, Lilliefors and Anderson–Darling tests, *J. Stat. Model. Anal.* **2** (2011), no. 1, 21–33.
[34] J. Rojo, Heavy tailed densities, *Wiley Interdiscip. Rev. Comput. Statist.* **5** (2013), no. 1, 30–40.
[35] R. Roy, Comparison of different techniques to generate normal random variables, *J. East Central Europe* **545** (2002), 5–6.
[36] S. J. S. Sant'Anna, J. C. S. Lacava and D. Fernandes, From Maxwell's equations to polarimetric SAR images: A simulation approach, *Sensors* **8** (2008), 7380–7409.
[37] M. Sato, S. W. Chen and M. Satake, Polarimetric SAR analysis of tsunami damage following the March 11, 2011 East Japan earthquake, *Proc. IEEE* **100** (2012), no. 10, 2861–2875.
[38] W. B. Silva, C. C. Freitas, S. J. S. Sant'Anna and A. C. Frery, Classification of segments in PolSAR imagery by minimum stochastic distances between Wishart distributions, *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sensing* **6** (2013), no. 3, 1263–1273.
[39] C. D. Storie, J. Storie and G. Salinas de Salmuni, Urban boundary extraction using 2-component polarimetric SAR decomposition, in: *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE Press, Piscataway (2012), 5741–5744.
[40] L. Torres, S. J. S. Sant'Anna, C. C. Freitas and A. C. Frery, Speckle reduction in polarimetric SAR imagery with stochastic distances and nonlocal means, *Pattern Recognit.* **47** (2014), 141–157.
[41] D. Velotto, S. Lehner, A. Soloviev and C. Maingot, Analysis of oceanic features from dual-polarization high resolution X-band SAR imagery for oil spill detection purposes, in: *IEEE International Geoscience and Remote Sensing Symposium* (IGARSS), IEEE Press, Piscataway (2012), 2841–2844.
[42] J. A. Villaseñor-Alva and E. González-Estrada, A bootstrap goodness of fit test for the generalized Pareto distribution, *Comput. Statist. Data Anal.* **53** (2009), no. 11, 3835–3841.
[43] C. Wang, X. Wen and H. Xu, A robust estimator of parameters for $\mathcal{G}_I^0$-modeled SAR imagery based on random weighting method, *EURASIP J. Adv. Signal Process.* **22** (2017), 42–52.
[44] Q. Wu, R. Chen, H. Sun and Y. Cao, Urban building density detection using high resolution SAR imagery, in: *Joint Urban Remote Sensing Event*, IEEE Press, Piscataway (2011), 45–48.
[45] Y. Yamaguchi, Disaster monitoring by fully Polarimetric SAR data acquired with ALOS-PALSAR, *Proc. IEEE* **100** (2012), no. 10, 2851–2860.
[46] MathWorks, MatLab, The language of technical computing, The MathWorks, 2019.
[47] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2016.