Research

REVISTA Ingeniería

FRand: MATLAB Toolbox for Fuzzy Random Number Simulation

FRand: Toolbox de MATLAB para Simulación de Números Aleatorios Difusos

Juan Carlos Figueroa–García [*,1], ***Jhoan Sebastian Tenjo–García*** [2]

[1]Universidad Distrital Francisco José de Caldas, Bogotá - Colombia, [2]Universidad Nacional de Colombia, Sede Bogotá

[*]Correspondence: jcfigueroag@udistrital.edu.co

Received: 27-11-2019. Modified: 10-03-2020. Accepted: 12-03-2020

Abstract

Context: This paper presents a MATLAB code implementation and the GUI (General User Interface) for fuzzy random variable generation. Based on previous theoretical results and applications, a MATLAB toolbox has been developed and tested for selected membership functions.

Method: A two–step methodology was used: i) a MATLAB toolbox was implemented to be used as interface and ii) all `.m` functions are available to be used as normal code. The main goal is to provide graphical and code–efficient tools to users.

Results: The main obtained results are the MATLAB GUI and code. In addition, some experiments were ran to evaluate its capabilities and some randomness statistical tests were successfully performed.

Conclusions: Satisfactory results were obtained from the implementation of the MATLAB code/toolbox. All randomness tests were accepted and all performed experiments shown stability of the toolbox even for large samples ($>$10.000). Also, the code/toolbox are available online.

Keywords: Fuzzy random numbers, Random number generation, MATLAB.

Acknowledgements: The authors would like to thank to the Prof. M Sc. Miguel Melgarejo and Prof. José Jairo Soriano–Mendez sincerely for their interest and invaluable support, and a special gratefulness is given to all members of *LAMIC*.

Language: English.

Open access

CC BY NC ND

Cite this paper as: Figueroa–García J.C., Tenjo–García J.S, "FRand: MATLAB Toolbox for Fuzzy Random Number Simulation", Ingeniería, Vol. 25, Num. 1, pp. 38-49, Jan.-Apr. 2020.
© The authors; reproduction right holder Universidad Distrital Francisco José de Caldas.
DOI: https://doi.org/10.14483/23448393.15620

Resumen

Contexto: Este trabajo presenta una implementación de código de MATLAB y un GUI (interfaz de usuario) para la generación de variable aleatoria difusa. Basados en resultados teóricos y aplicación previos, un toolbox de MATLAB fué desarrollado y validado para diferentes funciones de pertenencia.

Método: Una metodología de dos pasos ha sido implementada: i) un toolbox de MATLAB es implementado para usarse como interfaz y ii) todas las funciones `.m` están disponibles para usarse como código normal. La meta principal es proveer herramientas gráficas y de código a los usuarios

Resultados: Los resultados principales de este trabajo son el MATLAB GUI y el código subyacente. Adicionalmente, algunos experimentos fueron realizados para evaluar las capacidades del toolbox, y algunas pruebas estadísticas de aleatoriedad fueron realizadas con éxito.

Conclusiones: Resultados satisfactorios de la implementación del código/toolbox de MATLAB fueron obtenidos. Todos los tests estadísticos fueron aceptados y todos los experimentos realizados mostraron que el toolbox es estable aún para tamaños de muestra grande ($>$10.000). Adicionalmente, el toolbox/código está disponible online.

Palabras clave: Números aleatorios difusos, generación de variable aleatoria, MATLAB.

Agradecimientos: Los autores agradecen sinceramente a los Prof. M Sc. Miguel Melgarejo y Prof. José Jairo Soriano–Mendez por su interés e invaluable apoyo, y agradecen de manera especial a todos los miembros del Grupo *LAMIC*.

Idioma: Inglés.

1. Introduction

Random variable generation from a probabilistic point of view is based on the ideas, postulates, theorems of A. N. Kolmogorov [8] which have been widely applied to different sciences, including the theory of random processes theory (see Grimmet & Stirzaker [4] and Ross [13]). Even more, commercial simulation software packages including ARENA, ProModel, FlexSIM, MATLAB-EventSIM etc. are successful commercial applications of random variable theory and methods.

Most of probabilistic methods are based on the cumulative probability function (which is monotonic non-decreasing and injective) and the idea behind is to use uniform random numbers to generate probabilistic variables. Further information about probabilistic random variable theory and methods can be found in Devroye [5], Law & Kelton [9], Monahan [10], Grimmet & Stirzaker [4], Wilks [15] and Mood et al [11].

Analogous concepts for fuzzy numbers have been proposed by Varón-Gaviria et al. [14] and Pulido-López et al. [12] who used the well known α-cuts representation of a fuzzy set and the cumulative membership function $\psi_A(x)$ of A (see Figueroa-García & López-Bello [2], [3] and Figueroa-García [1]). This way, it is possible to efficiently generate random variables using uniform random numbers and fuzzy sets easy to implement in general purpose code/software like MATLAB.

This paper focuses on the implementation of two fuzzy random variable generation methods into a toolbox over MATLAB general purpose software. Some graphic analysis and statistical test are performed to verify independence and randomness of the obtained samples.

This paper is divided into six sections: Section 1 introduces the main problem. Section 2 presents a theoretical background; two methods for fuzzy random variable generation are introduced in Section 3; Section 4 presents the developed MATLAB toolbox; some experiments are presented in Section 5, and finally Section 6 shows some concluding remarks.

2. Theoretical background

First, we refer to well known definitions for fuzzy sets/numbers (see Klir & Folger [6], Klir & Yuan [7]) and the two implemented methods: α-cuts and cumulative membership function.

2.1. Basics on fuzzy sets

A fuzzy set is denoted by emphasized capital letters A with a membership function $\mu_A(x)$ over a universal set $x \in X$. $\mu_A(x)$ measures the membership of a value $x \in X$ regarding the concept/word/label A. $\mathcal{P}(X)$ is the class of all crisp sets, $\mathcal{F}(X)$ is the class of all fuzzy sets, $\mathcal{F}(\mathbb{R})$ is the class of all real-valued fuzzy sets, and $\mathcal{F}_1(\mathbb{R})$ is the class of all fuzzy numbers. Thus, a fuzzy set A is a set of ordered pairs of an element x and its membership degree, $\mu_A(x)$, i.e.,

$$\tilde{A} = \{(x, \mu_A(x)) \,|\, x \in X\}. \tag{1}$$

A fuzzy number is defined as follows:

Definition 1 *Let $\tilde{A} : \mathbb{R} \to [0, 1]$ be a fuzzy subset of the reals. Then $\tilde{A} \in \mathcal{F}_1(\mathbb{R})$ is a Fuzzy Number (FN) iff there exists a closed interval $[x_l, x_r] \neq \emptyset$ with a membership function $\mu_A(x)$ such that:*

$$\mu_A(x) = \begin{cases} c(x) & \text{for } x \in [c_l, c_r], \\ l(x) & \text{for } x \in [-\infty, x_l], \\ r(x) & \text{for } x \in [x_r, \infty], \end{cases} \tag{2}$$

where $c(x) = 1$ for $x \in [c_l, c_r]$, $l : (-\infty, x_l) \to [0, 1]$ is monotonic non-decreasing, continuous from the right, i.e. $l(x) = 0$ for $x < x_l$; $l : (x_r, \infty) \to [0, 1]$ is monotonic non-increasing, continuous from the left, i.e. $r(x) = 0$ for $x > x_r$.

The set of elements with membership degree larger than α are called the α-*cut* i.e. ${}^{\alpha}A$

$${}^{\alpha}A = \{x \,|\, \mu_A(x) \geqslant \alpha\} \;\forall\; x \in X, \tag{3}$$

$${}^{\alpha}A = \left[\inf_x {}^{\alpha}\mu_A(x), \sup_x {}^{\alpha}\mu_A(x) \right] = [\,\check{a}_\alpha,\, \hat{a}_\alpha\,]\,. \tag{4}$$

and its α-*level* namely A_α is the set of elements with membership degree equal to α, i.e.

$$A_\alpha = \{x \,|\, \mu_A(x) = \alpha \text{ for some } x \in X\}, \tag{5}$$

$$A_\alpha = \left\{ \inf_x {}^{\alpha}\mu_A(x), \sup_x {}^{\alpha}\mu_A(x) \right\} = \{\,\check{a}_\alpha,\, \hat{a}_\alpha\,\}\,. \tag{6}$$

The *support* of $A \in \mathcal{F}_1(\mathbb{R})$, *supp*$(A)$ is the set of all elements $x \in X$ with nonzero membership:

$$\textit{supp}(A) = \{x \,|\, \mu_A(x) > 0\} \;\forall\; x \in X, \tag{7}$$

$$\textit{supp}(A) = [\check{a}, \bar{a}]. \tag{8}$$

and the *core* of $A \in \mathcal{F}_1(\mathbb{R})$ is the set of elements with maximum membership degree:

$$\textit{core}(A) = \{x \,|\, \mu_A(x) = 1\} \;\forall\; x \in X, \tag{9}$$

$$\textit{core}(A) = [c_l, c_l]. \tag{10}$$

A trapezoidal fuzzy number is shown in Figure 1.

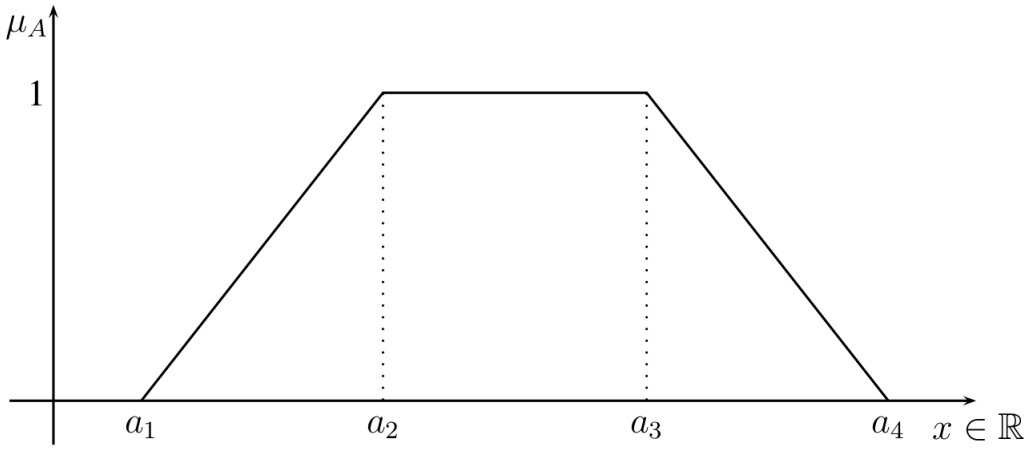

Figure 1. Fuzzy number $A \in \mathcal{F}_1(\mathbb{R})$

2.2. The Cumulative Membership Function (CMF)

The cumulative function $F(x)$ of a probability distribution $f(x)$ is defined as:

$$F(x) = \int_{-\infty}^{x} f(t)\, dt, \tag{11}$$

where $x \in \mathbb{R}$. Its fuzzy version is as follows (see Figueroa-García & López-Bello [2], [3], Figueroa-García [1] and Pulido-López et.al [12]).

Definition 2 (Cumulative Membership Function) *Let $\tilde{A} \in \mathcal{F}(\mathbb{R})$ be a fuzzy set and $X \subseteq \mathbb{R}$, then the cumulative membership function (CMF) of $\tilde{A}$, $\psi_A(x)$ is defined as:*

$$\psi_A(x) = Ps_A(X \leqslant x), \tag{12}$$

Eq. (12) is nothing else but the possibility that all elements of X are less or equal than a value x, i.e. $Ps(X \leqslant x)$. In the probabilistic theory $F(\infty) = 1$ so in the possibilistic theory $1 < \psi(\infty) < \Lambda$, where Λ is the cardinality of $\tilde{A}$:

$$\Lambda_A = \int_{-\infty}^{\infty} \mu_A(t)\, dt. \tag{13}$$

A normalized $\psi_A(x)$ is obtained as follows:

$$\psi_A(x) = \frac{1}{\Lambda_A} \int_{-\infty}^{x} \mu_A(t)\, dt. \tag{14}$$

Clearly, $\psi_A(x)$ is an surjection, so if we assume $\alpha \in [0, 1]$ then there exists one and only one value that satisfies $\psi_A(x) = \alpha$ i.e. $\psi_A^{-1} : \alpha \to X$.

3. Implementation of fuzzy random generation methods

There are essentially two kinds of fuzzy numbers: Singleton–core i.e. $core(A) = c \in \mathbb{R}$ and interval–core i.e. $core(A) = [c_l, c_l] \in \mathbb{I}$. The α-cuts method is adequate for single–core fuzzy numbers (triangular, Gaussian, exponential, etc), and the cumulative membership function approach is adequate for interval–core sets (trapezoidal, bell shaped, etc.). Both procedures are shown next:

3.1. α–cuts procedure

This method was proposed by Varón–Gaviria et al. [14] for single–core fuzzy sets. The main idea is to use the core c of $A \in \mathcal{F}_1(\mathbb{R})$ to compute the partial areas namely Λ_1, Λ_2 around c, as follows:

$$\Lambda_1 = \int_{x\in\mathbb{R}} l(x)dx;\ \ \Lambda_2 = \int_{x\in\mathbb{R}} r(x)dx, \tag{15}$$

$$\Lambda = \Lambda_1 + \Lambda_2 = \int_{x\in\mathbb{R}} l(x)dx + \int_{x\in\mathbb{R}} r(x)dx. \tag{16}$$

so the normalized areas λ_1, λ_2 of $\mu_A \in \mathcal{G}_1(\mathbb{R})$ are defined as follows:

$$\lambda_1 = \frac{\Lambda_1}{\Lambda};\ \ \lambda_2 = \frac{\Lambda_2}{\Lambda}, \tag{17}$$
$$\lambda_1 + \lambda_2 = 1.$$

Procedure 1 show the implemented routine.

Procedure 1 α-cuts method

Require: $n \in \mathbb{N}_+$ and $\mu_A \in \mathcal{F}_1(\mathbb{R})$
  Compute λ_1 and λ_2 using Eq. (17)
  for $n : 1 \to N$ **do**
    Set $\alpha_n = U_1 \in [0, 1]$ and $l_n = U_2 \in [0, 1]$
    Compute $A_{\alpha_n} = \{\check{a}_{\alpha_n}, \hat{a}_{\alpha_n}\}$
    If $l_n \leqslant \lambda_1$ then $x_n = \check{a}_{\alpha_n}$, otherwise set $x_n = \hat{a}_{\alpha_n}$
  end for
  return $[X, \alpha] = \{(x_1, \alpha_1), (x_2, \alpha_2), \cdots, (x_N, \alpha_N)\}$

3.2. Cumulative membership procedure

For interval-core fuzzy numbers, we implement the proposal of Pulido–López et al. [12] based on Eqs. (11) and (14), as shown in Procedure 2:

Procedure 2 ψ_A CMF method

Require: $n \in \mathbb{N}_+$ and $\mu_A \in \mathcal{F}_1(\mathbb{R})$
  Compute $\psi_A(x)$
  for $n : 1 \to N$ **do**
    Set $\alpha_n = U \in [0, 1]$
    Compute $x_n = \psi_A^{-1}(\alpha_n)$
  end for
  return $[X, \alpha] = \{(x_1, \alpha_1), (x_2, \alpha_2), \cdots, (x_N, \alpha_N)\}$

4. MATLAB Toolbox implementation

The implementation of both methods within MATLAB is divided into two parts: The code for all the implemented membership functions (useful for implementing fuzzy random variables as embedded code in other MATLAB routines/toolboxes) and the code for the GUI which are described as follows. Fig. 2 shows a screenshot of the MATLAB toolbox.

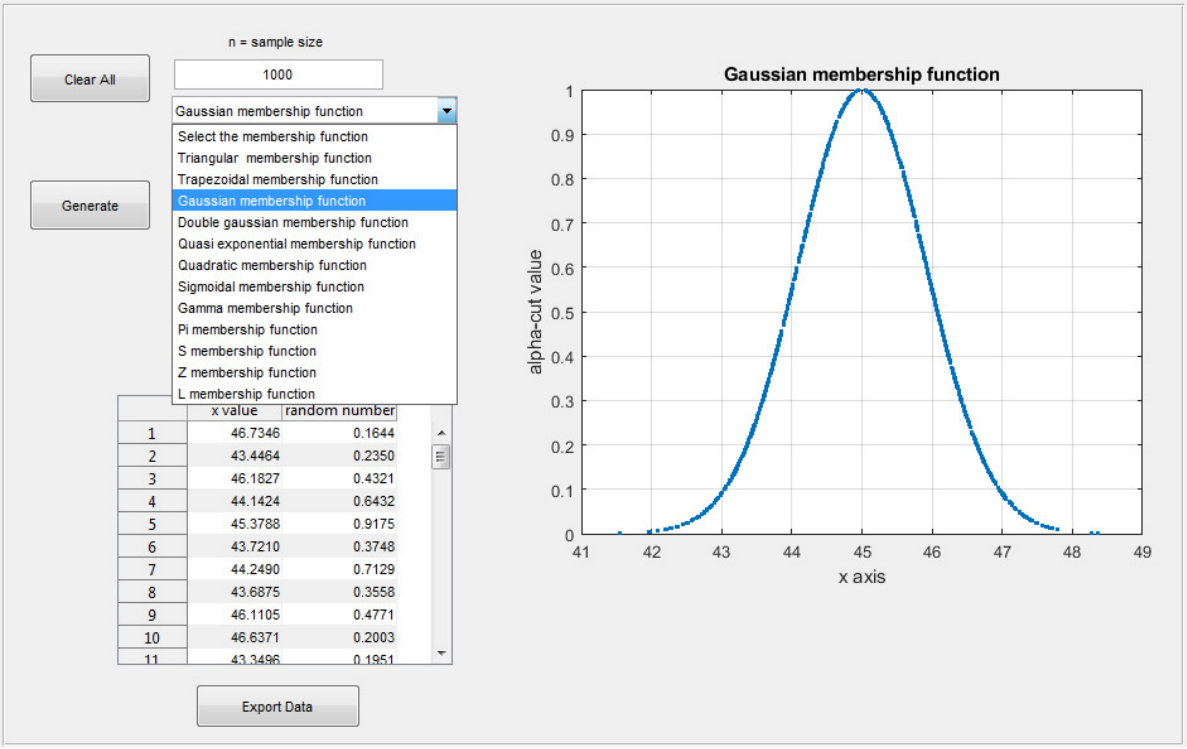

Figure 2. MATLAB Toolbox

4.1. MATLAB functions

MATLAB functions for five random generators are described below:

1. Triangular random generator

```
>> FT(a,m,b,n)
```

2. Gaussian random generator

```
>> FG(m,k,n)
```

3. Trapezoidal random generator

```
>> FTz(a,b,c,d,n)
```

4. Quasi-exponential random generator

```
>> FQe(m,k,n)
```

5. Double-gaussian random generator

```
>> FGT(m1,k1,l,r,m2,k2,n)
```

4.2. FRand toolbox

Basically, the toolbox contains all the previously described functions within a single file

`FRand.m`

that can be opened by simply writing

```
>> FRand
```

into the MATLAB command window.

4.3. Online availability

All functions have been compiled and compressed into a single file, available via the website of the Laboratory of Automation and Computational Intelligence (LAMIC) of the Universidad Distrital Francisco José de Caldas. To download the toolbox, please locate the file `FRcode.zip` at the URL:

https://comunidad.udistrital.edu.co/lamic/tools/

To install the functions, just copy the folder

`FRand`

into the path `\~\MATLAB\~~~~~\toolbox\` e.g. for the Release 2017b:

`\~\MATLAB\R2017b\toolbox\`

then add the toolbox path within MATLAB

```
File --> Set Path --> Add
```

The folder `\~\MATLAB\R2017b\toolbox\FRand\` contains the files:

```
FGT.m
FQe.m
FTz.m
FT.m
FG.m
```

After installing the folder and adding the path, the functions are ready to use and can be called from either the MATLAB command window, MATLAB SimuLink, MATLAB DiscreteEvents or any other toolbox.

Remark 1 *It is important to note that our approach uses MATLAB R2017a uniform random number routines, so it may not be compatible with future MATLAB versions.*

5. Experimentation and statistical analysis of generated data

To test the toolbox, we generated $n = 1000$ realizations of the following fuzzy random variables:

```
>> FT (44,60,85,1000)
>> FTz (10,15,20,22,1000)
>> FG (50,0.3,1000)
>> FQe (200,0.8,1000)
>> FGT (12,0.6,10,18,14,0.8,1000)
```

Then its mean, variance, runs test, turning points test, GAP (gap test) with the Kolmogorov Smirnov test, Chi–square test, and autocorrelation (min: $\downarrow$ and max: $\uparrow$) are computed to test if the obtained sequence is an independent. The results are shown in Table I.

Table I. Independence test on simulated variables(p-values)

Test	FT	FTz	FG	FQe	FGt
Mean	59,83	16,66	49,9	200,02	12,71
Variance	137,61	7,13	3,24	14,58	5,61
Runs	0,202	0,704	0,999	0,999	0,759
t. points	0,227	0,271	0,921	0,281	0,221
K-S	0,035	0,038	0,,031	0,027	0,018
χ^2	35,91	40,65	29,34	32,18	28,93
Autcor $\uparrow$	0,088	0,071	0,069	0,082	0,0936
Autcor $\downarrow$	-0,109	-0,092	-0,087	-0,096	-0,088

Runs and turning points tests show statistical significance of the hypothesis of independence on samples at a 95 % confidence level (see Table I). The autocorrelation test shows a Pearson test on autocorrelations for the 10 first lags where autocorrelation. $\downarrow$ is the lowest observed autocorrelation and autocorrelation. $\uparrow$ is the largest observed autocorrelation (at a 95 % confidence level).

The evidence shows that the generated samples are independent since the hypothesis of randomness is significant at a 95 % confidence level (p-value ¿0.05), and there is no evidence autocorrelated samples since the hypothesis of existence of autocorrelation is not significant at a 95 % confidence level (p-value ¡0.05). With the two gaps test, in the first case the Kolmogorov-Smirnov test is evaluated with a critical value $D_\alpha = 0{,}043$ for each membership function, all the values are less than D_α, therefore the 1000 numbers are randomly ordered, and finally the Chi-Square test, presented a value $\chi^2 = 44{,}98$ with 95 % confidence level and for $n = \sqrt{1000} - 1$. All the values obtained allow us to infer that the hypothesis that data are randomly distributed is significant for every membership function.

Figures 3–7 shows the exact location of the simulated values and their membership degree α.

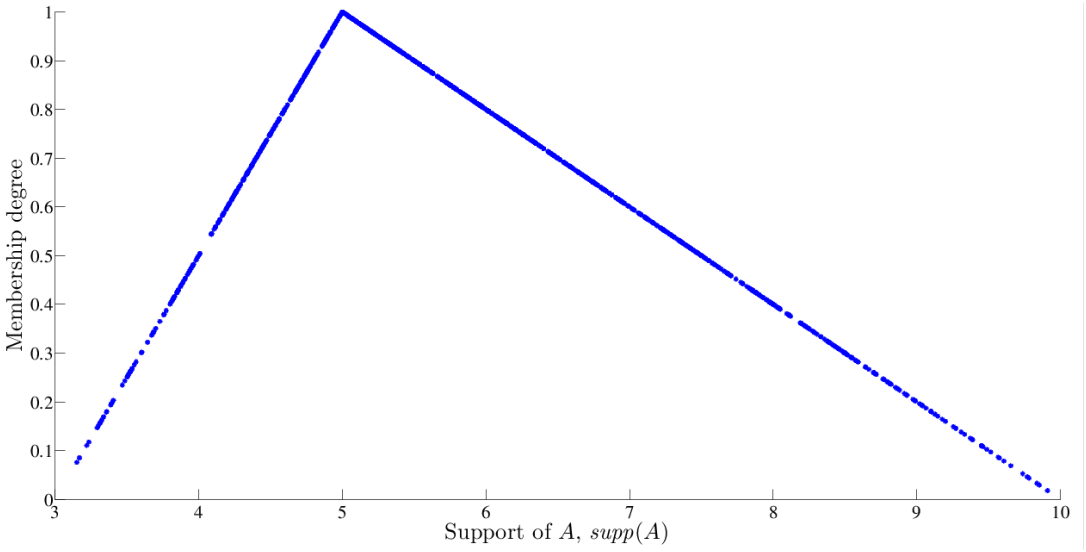

Figure 3. Simulated triangular random variables

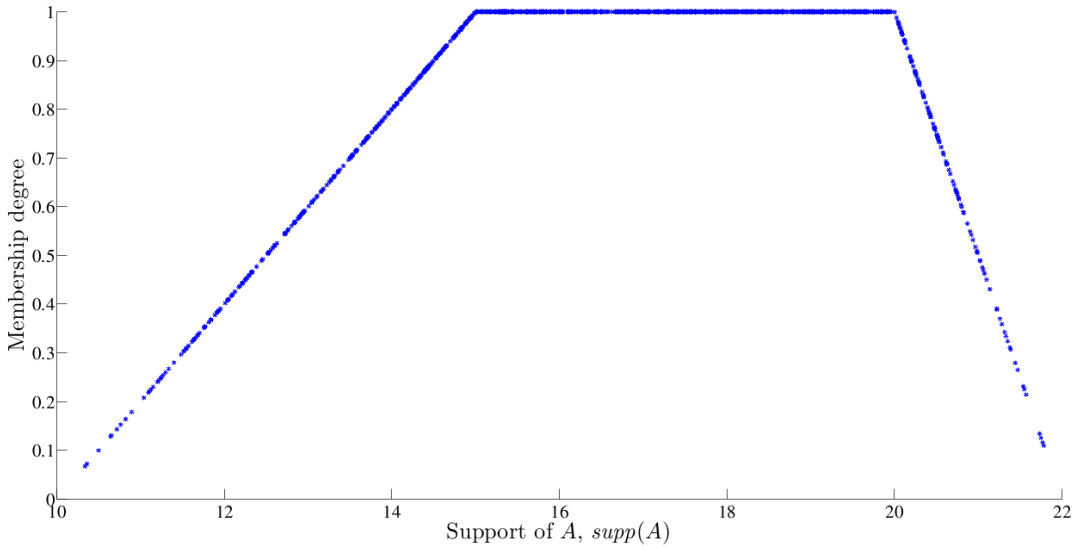

Figure 4. Simulated trapezoidal random variables

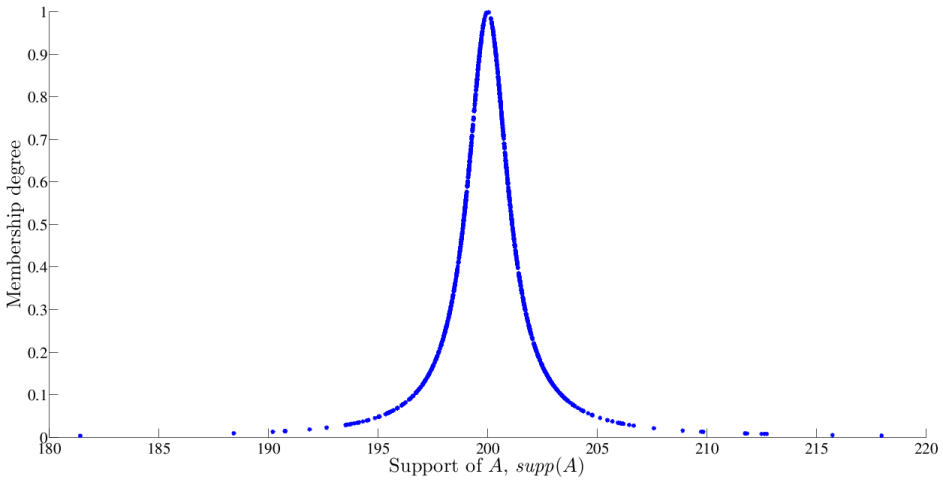

Figure 5. Simulated quasi–exponential random variables

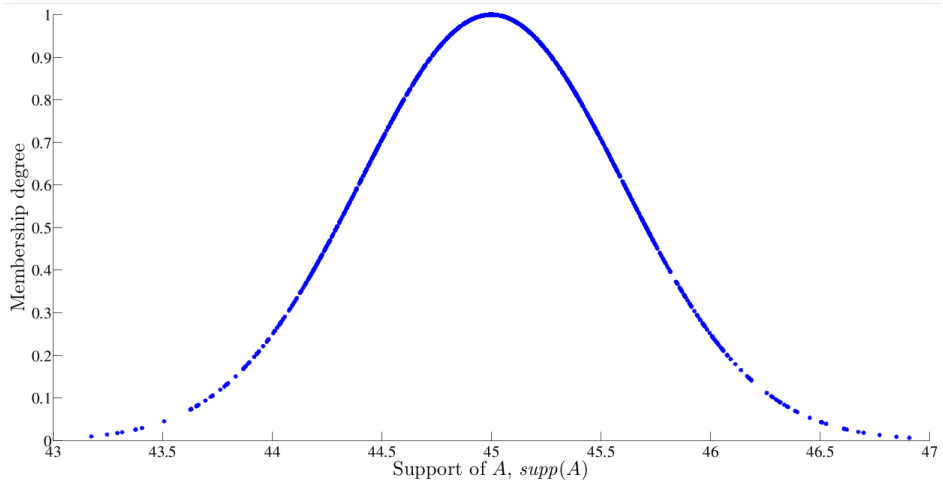

Figure 6. Simulated Gaussian random variables

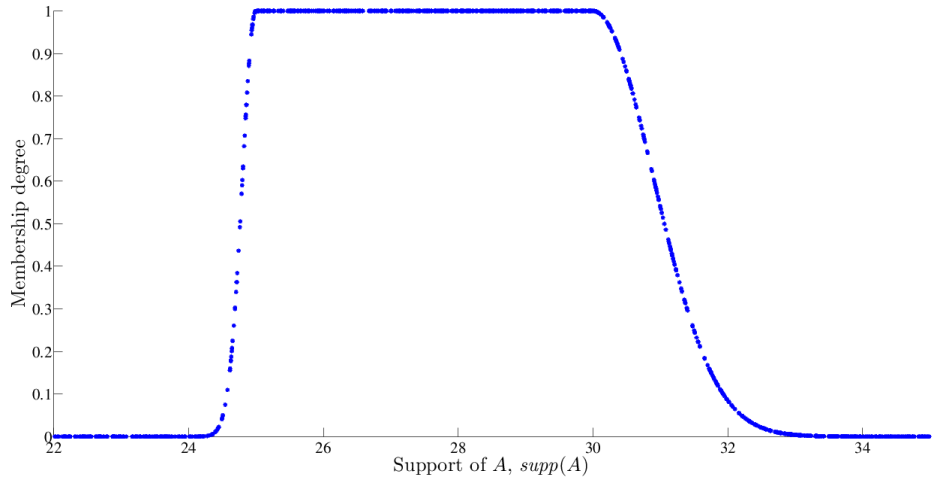

Figure 7. Simulated double–exponential random variables

Note that Figs. 3–7 are good representations of each fuzzy number since the obtained shapes correspond to each membership function.

6. Concluding Remarks

The theoretical results of Varón–Gaviria et al. [14] and Pulido–López et al. [12] were successfully implemented as MATLAB functions. All of them are available to download for free, and the are compatible with MATLAB 9.2 R2017a and later versions.

A theoretical background has been provided to explain how to use the implemented MATLAB's functions and toolbox. All functions are easy to understand, use, and are also complaint with MATLAB specifications for coding and compatibility.

Finally, some experiments are presented and tested using statistical tests on randomness and autocorrelation. The obtained results show generated samples to be independent and random, so the evidence supports the idea that the toolbox provides random samples.

References

[1] Figueroa-García, J.C.: Mixed production planning under fuzzy uncertainty: A cumulative membership function approach. In: IEEE (ed.) Workshop on Engineering Applications (WEA). vol. 1, pp. 1–6. IEEE (2012), https://doi.org/10.1109/WEA.2012.6220081 ↑39, 41

[2] Figueroa-García, J.C., López, C.A.: Linear programming with fuzzy joint parameters: A cumulative membership function approach. In: 2008 Annual Meeting of the IEEE North American Fuzzy Information Processing Society (NAFIPS). pp. 1–6. IEEE (2008), https://doi.org/10.1109/NAFIPS.2008.4531293 ↑39, 41

[3] Figueroa-García, J.C., López, C.A.: Pseudo-optimal solutions of FLP problems by using the cumulative membership function. In: Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS). vol. 28, pp. 1–6. IEEE (2009), https://doi.org/10.1109/NAFIPS.2009.5156464 ↑39, 41

[4] Grimmet, G., Stirzaker, D.: Probability and Random Processes. Oxford University Press (2001) ↑39

[5] Johnson, L.A., Montgomery, D.C.: Operations Research in Production Planning, Scheduling and Inventory Control. John Wiley & Sons (1974) ↑39

[6] Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty and Information. Prentice Hall (1992) ↑40

[7] Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall (1995) ↑40

[8] Kolmogorov, A.N.: Foundations of the Theory of Probability. Chelsea Publishing New York (1956) ↑39

[9] Law, A., Kelton, D.: Simulation Modeling and Analysis. Mc Graw Hill (2000) ↑39

[10] Monahan, J.F.: Numerical Methods of Statistics. Cambridge Press (2001) ↑39

[11] Mood, A.M., Graybill, F.A., Boes, D.C.: Introduction to the Theory of Statistics. Mc Graw Hill Book Company (1974) ↑39

[12] Pulido-López, D.G., García, M., Figueroa-García, J.C.: Fuzzy uncertainty in random variable generation: a cumulative membership function approach. Communications in Computer and Information Science 742(1), 398–407 (2017), https://doi.org/10.1007/978-3-319-66963-2_36 ↑39, 41, 42, 48

[13] Ross, S.M.: Stochastic Processes. John Wiley and Sons (1996) ↑39

[14] Varón-Gaviria, C.A., Barbosa-Fontecha, J.L., Figueroa-García, J.C.: Fuzzy uncertainty in random variable generation: an α-cut approach. Lecture Notes in Artificial Intelligence 10363(1), 1–10 (2017) ↑39, 42, 48

[15] Wilks, A.: Mathematical Statistics. John Wiley and Sons, New York (1962) ↑39

Juan Carlos Figueroa–García

He received the B.S. and M.S. degrees in Industrial Engineering from Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, in 2002/2009, and the Ph.D. degree in Industrial Engineering (meritorious degree) from National University of Colombia, Bogotá, Colombia, in 2014. Since 2006, he has been a Full Time teacher and researcher of the Engineering Faculty from Universidad Distrital Francisco José de Caldas, Bogotá, Colombia in the area of statistics and operations research. He is co-editor of different Springer LNCS/CCIS volumes and IEEE conference proceedings, and author of several book chapters and multiple scientific papers. He was awarded with the International Fuzzy Systems Association (IFSA) *"Young Scientist Award"* in 2012, the North American Fuzzy Information Processing Society (NAFIPS) *"Best Interval paper award"* in NAFIPS–IFSA conference in 2013 and the NAFIPS *"Outstanding Paper"* award in 2010. He is an IEEE Senior member, and currently serves as one of the board of directors of the North American Fuzzy Information Processing Society (NAFIPS). His main research interests include fuzzy measures, Type-2 fuzzy sets, fuzzy optimization and fuzzy statistics.
Correo electrónico: jcfigueroag@udistrital.edu.co

Jhoan Sebastian Tenjo–García

He obtained the B.S. degree in Industrial Engineering from Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, in 2017, and he is currently studying mathematics at National University of Colombia, Bogotá Campus. He is author of several scientific papers published by Springer–Nature LNCS/CCIS series. His main research interests include fuzzy sets, fuzzy mathematics, optimization and statistics.
Correo electrónico: jtenjo@unal.edu.co