Journal of Namibian Studies, 33 S2(2023): 2365–2376 ISSN: 2197-5523 (online)

Ordinary fractional differential equations to show results of oscillations, using the graphical and numerical approximation in the Julia programming language

Julio Cesar Morocho Orellana[1], Cristian Luis Inca Balseca[2], Franklin Marcelo Coronel Maji[3], Lizeth Fernanda Silva Godoy[4], Evelyn Geovanna Inca Balseca[5], José Luis Haro Mariño[6]

[1]Universidad Metropolitana Sede Machala, Ingeniería, Logística y Transporte, jmorocho@umet.edu.ec

[2]Escuela Superior Politécnica de Chimborazo (ESPOCH), Facultad de Informática y Electrónica. Tecnologías de la Información, cristianl.inca@espoch.edu.ec

[3]Escuela Superior Politécnica de Chimborazo (ESPOCH), fcoronel@espoch.edu.ec

[4]Departamento de Ciencias Exactas Universidad de las Fuerzas Armadas – ESPE, fercha.silva1@gmail.com

[5]independent researcher, eve.inca1999@gmail.com

[6]independent researcher, jose_pdagogiaymate@hotmail.com

Abstract

The present study developed at a documentary and explanatory level, which aims to assume through an illustrative approach in the Julia programming language, the representation of oscillating results in solutions of fractional ordinary differential equations, among these topics the solutions stand out in: Systems of differential equations equations, differential equations of partial fractions and the differential equations of fractional lags. In this sense, this scientific article assumes the recent foundations in this field and serves as foundations for future research contributions.

Keywords: Fractional differential equation, fractional derivatives, oscillation, Julia programming language Comentarios.

1. Introduction

Integer order differential equations have been used very frequently for a long time to describe phenomena or events of everyday life by implementing the theory of classical differential calculus. However, there are innumerable extremely complex situations in which this reality cannot be adjusted in terms of an entire order, but should be fractional to incorporate and take into account certain important characteristics. Addressing this gap by using fractional order in differential equations provides a better description and interpretation when constructing these models through the use of fractional calculus(Jarad, et al., 2017).

However, the growth of fractional calculus has stagnated due to insufficient interpretation and geometric results. In addition, inadequate physical interpretations of fractional derivatives are presented. (Jarad, et al., 2017)This is where the vertiginous advance in technology and high-speed computers comes in, which have encouragedresearchers and developers of programming languages to value the implicit importance of using thisderivative with absolute technical-computational precision to power. Create and apply a fractional differential operator specific to a real-life situation.

Evenfractional calculus has become a popular topic in virtually every branch of science and engineering. In fact, it has expanded rapidly due to the non-local character of fractional operators. As a result, fractional calculus and its multiple applications have aroused the interest of many researchers (Kilbas, et al., 2006; Podlubny, 1999).

For certain specific reasons, at the global level most real-life phenomena do not exhibit linear behavior. Therefore, it is possible to understand the nonlinear phenomenon of the real model through the resolution of this type of fractional differential equations both linear and non-linear. However,nonlinearity represents a qualitative property of differential equations that can be used to create or eliminate oscillation. Torsionoscillations, cardiac oscillations, sinusoidal oscillations and harmonic oscillations are allexamples of practical applications of the theory of oscillation of differential equations. (Grace & Tunç, 2018)

In view of the fact that there are considerable academic contributions in systematic developments of oscillation and non-oscillation in solutions of differential equations of integer order. It is of notable interest the approach in the theory of fractional calculus, concerning oscillation as a graphic solution for fractional differential equations, which has been investigated during the last twenty years. A characteristic example is represented by the study of oscillation in the graphical solution of nonlinear fractional differential equations developed by(Grace & Tunç, 2018) , these academics initiated and pioneered this topic. Therefore, in this line of research, various topics have been developed and supported with remarkable results.

This scientific article assumes the recent contributions in solving fractional differential equations and in the field of visualization in the graphic oscillations it represents; as well as, provide researchers with insight into the use ofthe programming language to facilitate such work.

The results of this research are based on the use of two contributions from the programming community around two libraries (FractionalDiffEq.jl and DifferentialEquations.jl) ofthe Julia programming language, which are an essential basis for solving differential equations with Different fractional operators.

2. Materials and Methods

This scientific article has been developed through the evolutionary methodology ofmathematical models related to fractional differential equations, their applicationin the field of science and the computational-technological environment. Therefore, the methodical bibliographic selection was implemented with the exhaustive review of articles that supportthe present line of research on the systems of fractional differential equations, differential equations of partial fractions and differential equations of fractional delays. Con fractional calculus theories related to fractional derivatives. By virtue of this, scientific databases were consulted through the use of meta-analyses from reliable sources; such as: Scopus, Springer, Elsevier, and MDPI (mathematics), online mathematical books, to then organize and synthesize the information obtained, in order to illustrate the reliability in the implementation of algorithms arranged in libraries or packages (FractionalDiffEq.jl and DifferentialEquations.jl) which have been the contributions of the programming community in the Julia language environment.

3. Implementation of EDF in the Julia programming language

The Julia programming language has received remarkable recognition for being a dynamic numerical software for mathematical and technical issues, which has been fundamentally optimal in the execution of algorithms, as performed by programs such as MATLAB, R and Python; which has recently achieved computational relevance within the scientific community and more in the mathematical field.(Lopez, 2019)

Thus, Julia is a homoiconic, cross-platform, multiparadigm programming language of high-level, high-performance dynamic typing for generic, technical, and scientific computing, with a syntax similar to that of other similar computing environments. (Lopez, 2018)

In addition, this language has an advanced compiler (JIT), mechanisms for parallel and distributed execution, as well as an extensive library of mathematical functions. Additionally, the community of contributors and developers between the Python and Julia communities contribute to the creation and distribution of external packages through Julia's integrated package manager at an accelerated pace. (Julia, 2022)

Regarding the handling of fractional differential equations, the developer community presents extensive libraries, such as: FractionalDiffEq.jl and DifferentialEquations.jl; which facilitates problem solvers associated with EDF. In addition, due to its dynamic nature, it presents a range of alternatives in problem solvers with absolute performance and ability to solve the different types of fractional differential equations.

For this scientific article, the installation in Julia of FractionalDiffEq.jl is developed, using the Julia package manager, as shown in the following figure:

Figure 1: The installation procedure for the FractionalDiffEq.jl package.

```
pkg> add FractionalDiffEq
```

Once the installation is executed, we proceed to the empirical illustration of situations related to the approach of ordinary differential equations (EDF).

3.1. Ordinary fractional differential equations (FDE)

The first situation in which an initial value ordinary EDF problem is addressed is considered:

$$D^{1.8}y(x) = 1 - y$$

$$y(0) = 0$$

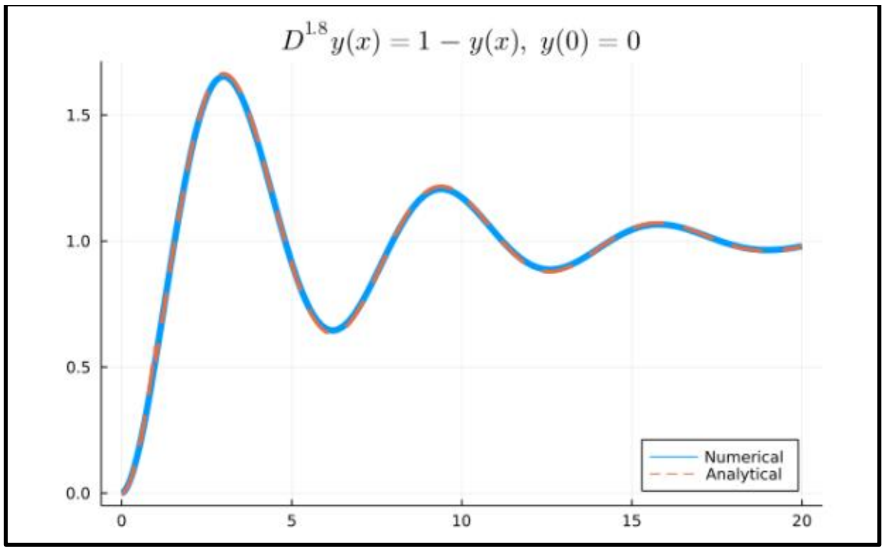
Where in this section to implement the FractionalDiffEq.jl library in Julia, to solve the previous problem:

Figure 2: Iterative procedure to resolve ordinary EDF of initial value.

```
using FractionalDiffEq, Plots
fun(u, p, t) = 1-u
u0 = 0; tspan = (0, 20); h = 0.001;
prob = SingleTermFODEProblem(fun, 1.8, u0, tspan)
sol = solve(prob, h, PECE())
plot(sol)
```

Where the numerical solution of the ordinary EDF with initial value can be represented graphically, plotting the numerical and analytical solution (See Figure 3). This solution uses a PECE lgorhythm, which refers to Predict-Evaluate-Correct-Evaluate.(Diethelm, et al., 2002)

Figure 3: Ordinary EDF oscillatory representation of initial value.



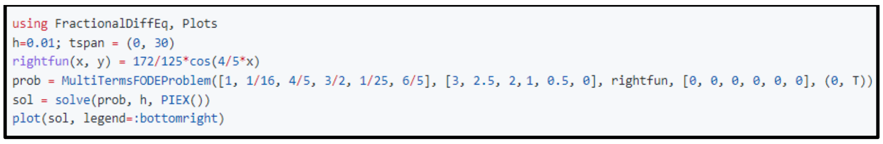
When considering a more complex problem of ordinary EDF with initial value, such as:

$$y'''(t) + \frac{1}{16}\,{}_0^C D_t^{2.5} y(t) + \frac{4}{5} y''(t) + \frac{3}{2} y'(t) + \frac{1}{25}\,{}_0^C D_t^{0.5} y(t) + \frac{6}{5} y(t) = \frac{172}{125} cos\left(\frac{4t}{5}\right)$$

$$y(0) = 0, y'(0) = 0, y''(0) = 0$$

In what systematizes the procedure in this section to implement from the FractionalDiffEq.jl library in Julia, to solve the previous problem:

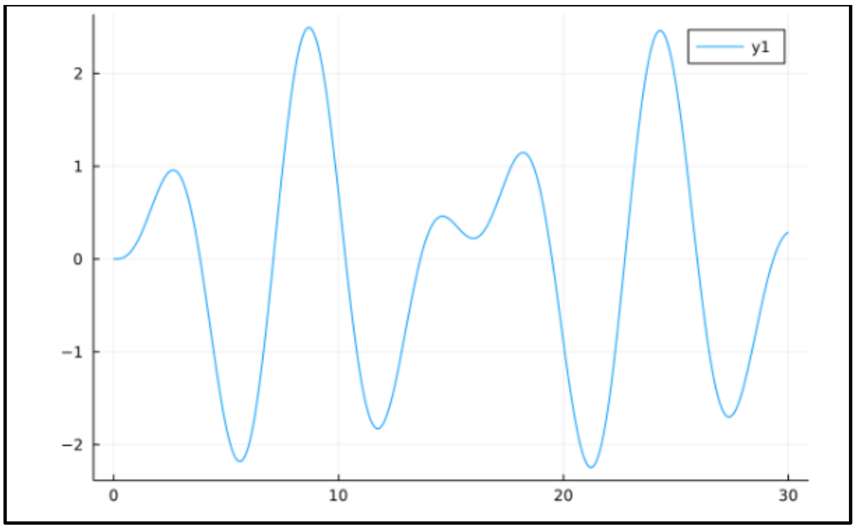
Figure 4: Iterative procedure for solving a complex ordinary EDF of initial value.



```
using FractionalDiffEq, Plots
h=0.01; tspan = (0, 30)
rightfun(x, y) = 172/125*cos(4/5*x)
prob = MultiTermsFODEProblem([1, 1/16, 4/5, 3/2, 1/25, 6/5], [3, 2.5, 2, 1, 0.5, 0], rightfun, [0, 0, 0, 0, 0, 0], (0, T))
sol = solve(prob, h, PIEX())
plot(sol, legend=:bottomright)
```

The numerical solution of the ordinary EDF complex with initial value can be represented graphically, plotting the numerical and analytical solution (See Figure 5). This solution uses the PIEX algorithm(Garrappa, 2018).

Figure 5: Oscillatory representation of an ordinary EDF complex with initial value.



3.2. System of Fractional Differential Equations

In this section, a mathematical equation is used that models dynamical systems such as the chaotic Chua system and that are to be used for the purpose of proof or illustration as an empirical solution based on the Julia programming language. This language presents facilities in the use of libraries in mathematical functions (FractionalDiffEq.jl and DifferentialEquations.jl); being these robust tool based on external packages included in the package manager integrated into Julia by developers and have to be used to solve systems of fractional differential equations.

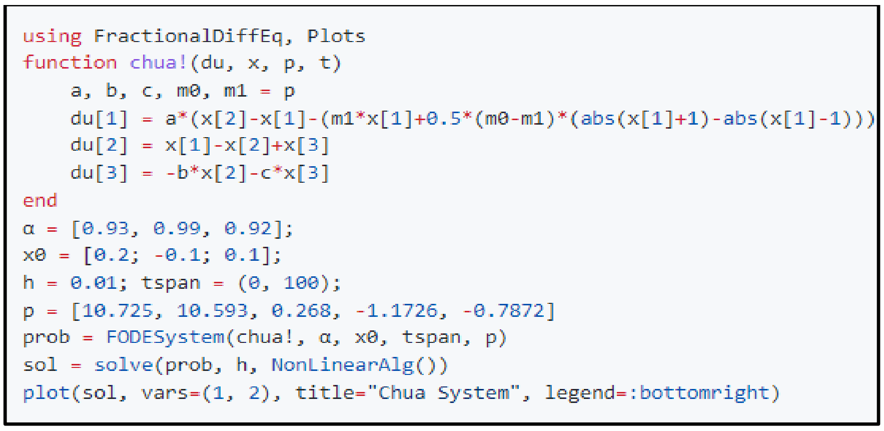
Therefore, the chaotic Chua system is considered:

$$\begin{cases} D^{\alpha_1}x = 10.725\big[y - 1.7802x - [0.1927(|x+1| - |x-1|)]\big] \\ D^{\alpha_2}y = x - y + z \\ D^{\alpha_3}z = -10.593y - 0.268z \end{cases}$$

Where are state variables andx,y,z numeric values are system parameters.(Kiseleva, et al., 2017)

The following nonlinear lgorhythm (NonLinearAlg) is used in the system of nonlinear fractional differential equations, of this forma (Dingyu, 2018)offers numerical and graphic results or d the chaotic system of Chua, by means of the following line of commands:

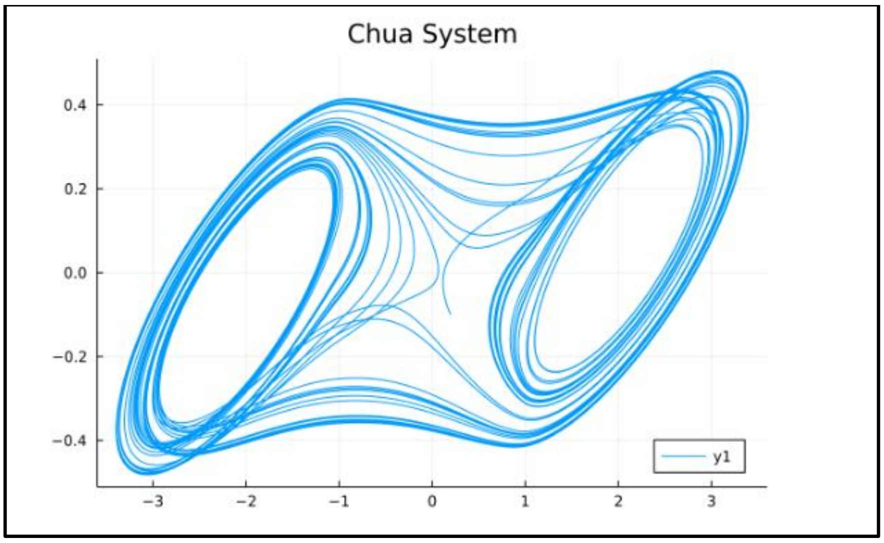
Figure 6: Iterative procedure for chaotic Chua EDF system.



```
using FractionalDiffEq, Plots
function chua!(du, x, p, t)
    a, b, c, m0, m1 = p
    du[1] = a*(x[2]-x[1]-(m1*x[1]+0.5*(m0-m1)*(abs(x[1]+1)-abs(x[1]-1)))
    du[2] = x[1]-x[2]+x[3]
    du[3] = -b*x[2]-c*x[3]
end
α = [0.93, 0.99, 0.92];
x0 = [0.2; -0.1; 0.1];
h = 0.01; tspan = (0, 100);
p = [10.725, 10.593, 0.268, -1.1726, -0.7872]
prob = FODESystem(chua!, α, x0, tspan, p)
sol = solve(prob, h, NonLinearAlg())
plot(sol, vars=(1, 2), title="Chua System", legend=:bottomright)
```

The graphic representation is reflected as follows:

Figure 7: Oscillatory representation of EDF's chaotic Chua system.



3.3. Partial fractional differential equations (PDEs)

The following empirical illustration using the FractionalDiffEq.jl library tends to providerobust algorithms for solving partial fractional differential equations. The following diffusion equation is taken into consideration:(Podlubny, et al., 2009)

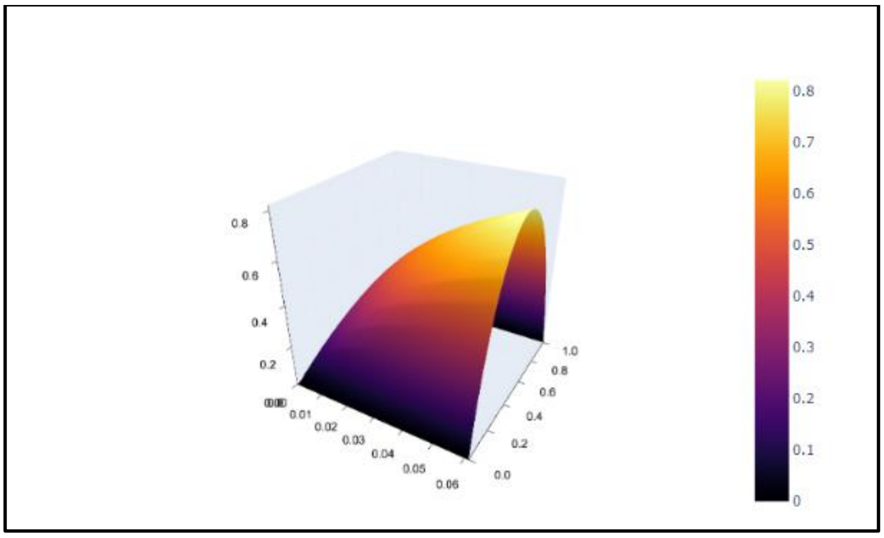
$$ {}_0^C! D_t^{\alpha} y - \frac{\partial^{\beta} y}{\partial |x|^{\beta}} = f(x,t) $$

Assuming the following initial and boundary conditions:

$$ y(0,t) = 0, \qquad y(1,t) = 0, \qquad y(x,0) = 0 $$

In this interest at this point, it should be to reflect the numerical approximation to the solution of the previous EDFP (See Figure 8). Using the explicit Caputo discretization method to solve the diffusion equations .(Murillo & Yuste, 2011)

Figure 8: Numerical representation of the EDFP.



3.4. Fractional delay differential equations (FREDs)

In practice, notable developments have emerged that have contributed to various and effective robust solvers to solve fractional delay differential equations. At this point, when considering the following logistic model of populations, introduced by Verhulst . Assuming in this model a variation introduced by , which proposes that the reproductive rate is reached in units with respect to the state defined by , that is, the structure of the fractional delay differential equation that describes this system is presented below, assuming values to the parameters that are characteristic:(Murray, 2002; Dreyer, 1993)(Hutchinson, 1958) $1 - N/K\tau > 0N$

$$D_t^{\alpha} y(t) = 3.5y(t)\left(1 - \frac{y(t-0.74)}{19}\right)$$

$$y(0) = 19.00001$$

With history function:

$$y(t) = 19, t < 0$$

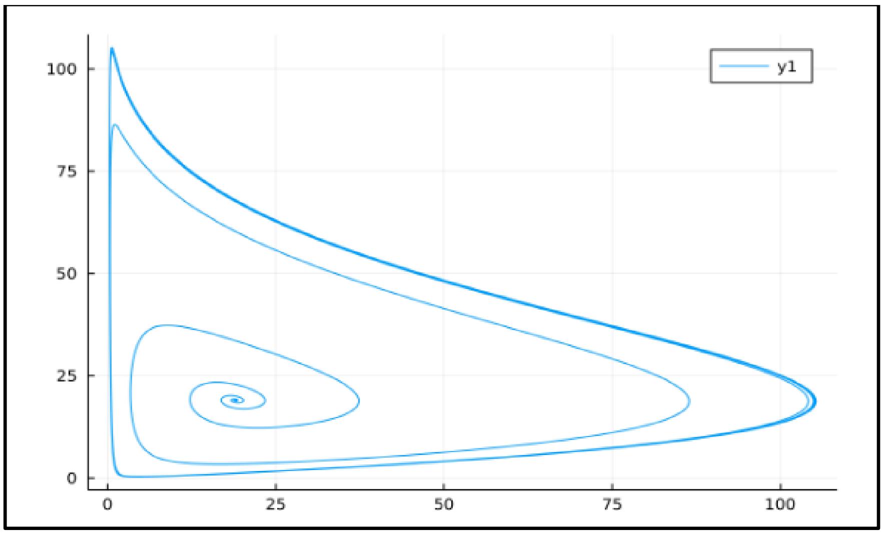
The DelayPECE() method, delayed predictor-corrector is used to solve the problem of the delayed fractional differential equation in the sense of Caputo . Capable of solving both single-term EDRF and multiple EDRFs, (Wang, 2013; Abdelmalek & Douaifia , 2019)so variable delays in time are supported. Ofcourse, the following command line is used to solve numerically and graphically the system corresponding to the EDRF.

Figure 9: Iterative procedure for EDRF.

```
using FractionalDiffEq, Plots
φ(x) = x == 0 ? (return 19.00001) : (return 19.0)
f(t, y, φ) = 3.5*y*(1-φ/19)
h = 0.05; α = 0.97; τ = 0.8; T = 56
fddeprob = FDDEProblem(f, φ, α, τ, T)
V, y = solve(fddeprob, h, DelayPECE())
plot(y, V, xlabel="y(t)", ylabel="y(t-τ)")
```

The graphic representation is reflected as follows:

Figure 10: Oscillatory representation of the EDRF system.



3.5. Lyapunov Characteristic Exponents in the System of Fractional Order Differential Equations

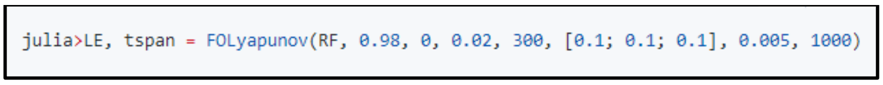
The so-called chaotic dynamical systems are often associated with systems that are very sensitive to variations in initial conditions. By virtue of the fact that the Lyapunov Characteristic Exponents (LCE) has been the tool that allows quantifying the speed at which two orbits separate with infinitely close initial conditions. Therefore, they are often used as indicators of the presence of chaos.

A dynamical system is considered to be stable (in the Lyapunov sense) if for every small one, there exists ϵ>0δ > 0 a , such that every solution with initial conditions at a distance less than , remains within the distance , for any . Note that this definition depends on the distance function used. $f(t)\delta\|\boldsymbol{f}(\boldsymbol{t_0})-\boldsymbol{f_e}(\boldsymbol{t_0})\| < \boldsymbol{\delta}\ \boldsymbol{\epsilon}\|\boldsymbol{f}(\boldsymbol{t})-\boldsymbol{f_e}(\boldsymbol{t})\| < \boldsymbol{\epsilon} t\ \geq\ t_0$

All this theory is immersed in the study of the stability of the solutions of differential equations and trajectories of dynamical systems under perturbations of the initial conditions. Consequently, the stability theory allows to see the convergence of the error, which translates into satisfactory synchronization. Stability can be studied under different approaches. Next, we consider the assumptions in linear stability theory and Lyapunov stability theory that are relevant to this scientific article under iterative procedures in the Julia programming language. Therefore, the FractionalDiffEq.jl library is used, which allows generating exponents of Lyapunov of a fractional order system, such consideration is made based on the Rabinovich-Fabrikant System, which is raised below:(Kumar, et al., 2019)

$$\begin{cases} D^{\alpha_1}x = y(z-1+z^2)+\gamma x \\ D^{\alpha_2}y = x(3z+1-x^2)+\gamma y \\ D^{\alpha_3}z = -2z(\alpha+xy) \end{cases}$$

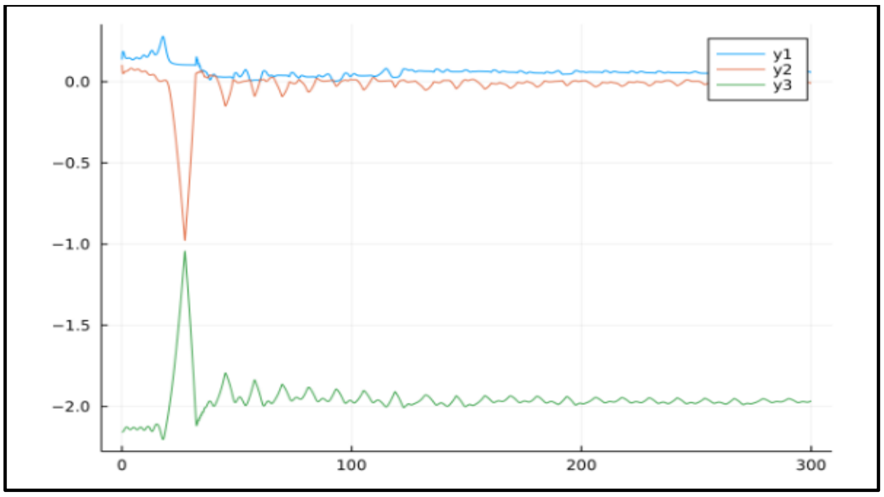
Figure 11: Iterative procedure for Lyapunov exponents of the fractional order system.



```
julia>LE, tspan = FOLyapunov(RF, 0.98, 0, 0.02, 300, [0.1; 0.1; 0.1], 0.005, 1000)
```

The graphic representation is reflected as follows:

Figure 12: Oscillatory representation of the fractional order LCE system.



4. Conclusions

Fractional calculus approached through a programming language such as Julia, provides an appropriate mathematical instrumentation to describe memory-dependent phenomena and the intermediate process, which has increasingly penetrated various areas of science and engineering. In particular the problems of diffusionassociated with contaminants, and those of viscoelastic materials are better approximated with a fractional model. From this perspective, numerical and graphical solutions are obtained through the packages FractionalDiffEq.jl and DifferentialEquations.jl. So, these solutions in terms of fractional ordinary differential equations, were obtained by implementing the PECE-Predict-Evaluate-Correct-Evaluate Algorithm and in more complex cases of numerical solution the PIEX algorithm tends to be used. As for the systems of fractional differential equations(Diethelm, et al., 2002)(Garrappa, 2018), it has evaluated the chaotic system of Chua by employing the nonlinear algorithm (NonLinearAlg) to solve numerically and graphically the system of chaotic nonlinear fractional differential equations. Then, the differential equations of (Dingyu, 2018) partial fractions are shown, in which the illustration has been taken into consideration by the diffusion equation, which was addressed by the method of discretization of Caputo explicit to solve the diffusion equations. And to conclude, the differential equations of fractional delays, the logistic model of population growth has been considered, using the delayed predictor-corrector method to solve the problem of the delayed fractional differential equation in the sense of Caputo (Wang, 2013; Abdelmalek & Douaifia, 2019). (Podlubny, et al., 2009)(Murillo & Yuste, 2011)UsingCaputo's fractional derivative operator tends to offer more advantages when working with problems of fractional differential equations with initial value and is actively used to model complex systems and processes with memory that have different time scales. In short, these methods can be used to obtain approximate numerical solutions (Luchko, 2022) and graphical oscillations of fractional differential equations in different typologies through simple programming in the Julia language.

Bibliography

Abdelmalek & Douaifia . (2019). A Predictor-Corrector Method for Fractional Delay-Differential System with Multiple Lags. . cna-journal Commun. Nonlinear Anal., 6(1), 78-88. Obtenido de http://www.cna-journal.com/article_91820_d0b302b5e7fb81cc1bb69b5a1013e558.pdf

Diethelm, et al. (2002). A predictor-corrector approach for the numerical solution of fractional differential equations. . Springer. doi:https://doi.org/10.1023/A:1016592219341

Dingyu. (2018). Fractional calculus and fractional control. (Vol. ISBN: 9787030543981). China: Northeastern University, China. Editorial Scientific Press.

Dreyer. (1993). Modelling with Ordinary Differential Equations (1.aed. ed.). Florida: Routledge.

Garrappa. (2018). Numerical Solution of Fractional Differential Equations: A Survey and a Software Tutorial. Mathematics, 6(2), 16. Obtenido de https://www.mdpi.com/256346

Grace & Tunç. (2018). On the oscillatory behavior of solutions of higher order nonlinear fractional differential equations. . Georgian Math. J. , 25, 363–369. doi:https://doi.org/10.1515/gmj-2017-0026

Hutchinson. (1958). Circular causal systems in ecology,. Annals of the New York Academy of Sciences, 50(4), 221–246.

Jarad, et al. (2017). On a new class of fractional operators. . Adv. Differ. Equ., 247(16). doi:https://doi.org/10.1186/s13662-017-1306-z

Julia. (2022). Retrieved from https://julialang.org/

Kilbas, et al. (2006). Theory and Applications of Fractional Differential Equations; North-Holland Mathematics Studies. Elsevier Science B.V. , 204. Obtenido de https://www.elsevier.com/books/theory-and-applications-of-fractional-differential-equations/kilbas/978-0-444-51832-3

Kiseleva, et al. (2017). Hidden and self-excited attractors in Chua circuit: synchronization and SPICE simulation. International Journal of Parallel, Emergent and Distributed Systems, 33(5), 513–523. doi:https://doi.org/10.1080/17445760.2017.1334776

Kumar, et al. (2019). Synchronization of fractional order Rabinovich-Fabrikant systems using sliding mode control techniques. Archives of Control Sciences, 29(2), 307–322.

Lopez . (August 13, 2018). The Julia 1.0 programming language is released. . Obtained from UNOCERO: https://www.unocero.com/software/se-libera-el-lenguaje-de-programacion-julia-1-0/

Lopez. (January 7, 2019). The creators of the Julia programming language receive an award. . Obtained from UNOCERO: https://www.unocero.com/noticias/los-creadores-del-lenguaje-de-programacion-julia-reciben-premio/

Luchko. (8 de Marzo de 2022). Fractional Differential Equations with the General Fractional Derivatives of Arbitrary Order in the Riemann–Liouville Sense. Mathematics, 10(849). doi:https:// doi.org/10.3390/math10060849

Murillo & Yuste. (2011). An Explicit Difference Method for Solving Fractional Diffusion and Diffusion-Wave Equations in the Caputo Form. Journal of Computational and Nonlinear Dynamics, 6, 1-6. doi:10.1115/1.4002687

Murray. (2002). Mathematical Biology I, An Introduction (3.aed. ed.). New York, E.E.U.U: Springer. doi:https://doi.org/10.1007/b98868

Podlubny. (1999). Fractional Differential Equations. An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications. Mathematics in Science and Engineering, 198. Obtenido de http://www.sciepub.com/reference/3051

Podlubny, et al. (2009). Matrix approach to discrete fractional calculus II: Partial fractional differential equations. . ELSEVIER, 222(8), 3137-3153. doi:https://doi.org/10.1016/j.jcp.2009.01.014

Wang. (2013). A Numerical Method for Delayed Fractional-Order Differential Equations. . J. Appl. Math., 2013, 1-7. doi:https://doi.org/10.1155/2013/256071