

One Method for the Numerical Resolution of One Type of Fractional Differential Equation of Euler-Lagrange

L. H. D. L. Raharimina¹, G. Rasolomampandry² and F. Randimbindraine³

¹Cognitive Sciences and Applications Research Laboratory (LR-SCA), Madagascar

²Doctoral School in Sciences and Techniques of Engineering and Innovation (ED-STII), Madagascar

³Higher Polytechnic School Antananarivo (ESPA), University of Antananarivo BP 1500 Ankatso-Antananarivo 101- Madagascar

Abstract

We propose in this article one method of numerical resolution using the programming under Matlab of one of the fractional differential equations of Euler-Lagrange containing a composition of the left and right fractional derivatives of order α , $0 < \alpha < 1$, of Riemann-Liouville and Caputo respectively.

Keywords: Numerical resolution • Fractional derivatives • Programming under Matlab

Introduction

Fractional differential equations are becoming more and more interesting in recent years for scientists. Many authors have already studied different types of problems and have formulated Euler-Lagrange equations with different boundary conditions [1-18]. An ordinary differential equation containing a composition of left and right fractional derivatives to order α , $0 < \alpha < 1$, with boundary conditions, is considered in our work. This type of equation is known as the fractional differential equation of Euler-Lagrange. Fractional differential equations appear in certain fields such as physics, mechanics, electrical engineering, etc... This type of equation can be solved with other methods such as the Laplace transformation or a method using the principle of least action [10,11]. An analytical method using the fixed-point theorem [6,7] also provides a solution, but this method is complicated. This is why we propose another numerical method to solve this type of fractional differential equation. We opted for the programming under Matlab language to find this numerical solution.

Research Methodology

This work will be presented in four sections: in section 2, we recall the definitions which are necessary tools for the resolution of the equation that we will present in section 3. In section 4, we propose a numerical solution, at first we search the numerical schemes for both fractional operators occurring in equation and then we determine a developed form of equation in order to establish a system of linear equations that we will transform into matrix form. In the last section, the programming under Matlab language allowing finding approximate numerical solutions graphically and the examples of numerical solution are given.

Preliminaries

In this section, we give some useful definitions for this numerical solution. The left and right fractional integrals in the sense of Riemann-Liouville of order α on an interval $[0, b]$ are defined respectively by:

***Address for Correspondence:** L.H.D.L. Raharimina. Department of Cognitive Sciences and Applications Research Laboratory (LR-SCA), Madagascar, E-mail: louisetteraharimina@gmail.com

Copyright: © 2022 Raharimina LHDL, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Date of Submission: 04 June, 2022, Manuscript No. jacm-22-66146; **Editor assigned:** 06 June, 2022, PreQC No. P-66146; **Reviewed:** 12 June, 2022, QC No. Q-66146; **Revised:** 18 June, 2022, Manuscript No. R-66146; **Published:** 24 June, 2022, DOI: 10.37421/2168-9679.2022.11.475

for $t > 0$ (1)

$$I_{b-}^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_t^b \frac{f(u)}{(u-t)^{1-\alpha}} du \text{ for } t < b \text{ (2)}$$

Where Γ is Euler's Gamma function and $b \in \mathbb{R}_+$.

The left and right fractional derivatives of Riemann-Liouville to order α , $0 < \alpha < 1$ are defined from these fractional integrals to order α by:

$$D_0^{\alpha} f(t) = \frac{d}{dt} I_0^{1-\alpha} f(t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_0^t \frac{f(u)}{(t-u)^{\alpha}} du \text{ for } t > 0 \text{ (3)}$$

$$D_{b-}^{\alpha} f(t) = -\frac{d}{dt} I_{b-}^{\alpha} f(t) = \frac{-1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_t^b \frac{f(u)}{(u-t)^{\alpha}} du \text{ for } t < b \text{ (4)}$$

The left and right fractional derivatives to order α in the sense of Caputo are also defined from the left and right fractional integrals of Riemann-Liouville of order α by:

$${}^c D_0^{\alpha} f(t) = I_0^{1-\alpha} \left(\frac{d}{dt} f(t) \right) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{f(u)}{(t-u)^{\alpha}} du \text{ for } t > 0 \text{ (5)}$$

$${}^c D_{b-}^{\alpha} f(t) = -I_{b-}^{1-\alpha} \left(\frac{d}{dt} f(t) \right) = \frac{-1}{\Gamma(1-\alpha)} \int_t^b \frac{f(u)}{(u-t)^{\alpha}} du \text{ for } t < b \text{ (6)}$$

The relationship between the left fractional derivatives of Riemann-Liouville and the left fractional derivative of Caputo, both of order α , is the following:

$$D_0^{\alpha} f(t) = {}^c D_0^{\alpha} f(t) + \frac{t^{-\alpha}}{\Gamma(1-\alpha)} f(0) \text{ for } t > 0 \text{ (7)}$$

$$D_{b-}^{\alpha} f(t) = {}^c D_{b-}^{\alpha} f(t) + \frac{(b-t)^{-\alpha}}{\Gamma(1-\alpha)} f(b) \text{ for } t < b \text{ (8)}$$

Presentation of the problem

We study in this work, one of type of the fractional differential equation of Euler-Lagrange of order α :

$${}^c D_{1-}^\alpha D_0^\alpha f(t) + rf(t) = mg(t) \quad (9)$$

Where: f is the unknown function, g is a given function, D_0^α and ${}^c D_{1-}^\alpha$ are respectively the left fractional derivatives of Riemann-Liouville and the right fractional derivatives of Caputo, both of order α with $0 < \alpha < 1$ and $t \in [0, 1]$, the boundary conditions are:

$$f(0) = 0 \text{ and } f(1) = 0 \quad (10)$$

Numerical resolution

To find the discrete form of equation (9), subdivide the interval $[0, 1]$ into N parts with constant step $h = \frac{1}{N}$, so there exist $(N + 1)$ points $t_i, i = 0, 1, \dots, N$ on this interval with $t_0 = 0$ and $t_N = 1, t_i = ih = \frac{i}{N}$ are arranged in ascending order.

The discrete value of the function f at each point t_i is noted $f_i = f(t_i)$.

And we note $f'_i = \left(\frac{\partial f}{\partial t} \right)_{t=t_i}$ the derivative of the function f at the point t_i .

The finite difference scheme of order 1 is written:

$$f'_i = \left(\frac{\partial f}{\partial t} \right)_{t=t_i} = \frac{f_{i+1} - f_i}{h} \quad (11)$$

This scheme allows us to find the approximate value of the derivative of the function f at each point t_i .

We determine the discrete form of this equation (9) and translate the equation obtained in matrix form.

Here are the different steps to follow to find this discrete form of the equation (9).

We will present the discrete form of each type of fractional derivative, firstly the left fractional derivative of Riemann-Liouville and secondly the right fractional derivative of Caputo, of order $\alpha, 0 < \alpha < 1$ [9,14].

Approximation of the fractional derivative of Riemann-Liouville of order α at point t_i .

For this, we will use the relation (7) between the left fractional derivative of Riemann-Liouville and the left fractional derivative of Caputo of order α

$$\begin{aligned} D_0^\alpha f(t) &= {}^c D_0^\alpha f(t) + \frac{t_i^{-\alpha}}{\Gamma(1-\alpha)} f(0) \\ &= \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{f'(u)}{(t_i-u)^\alpha} du + \frac{t_i^{-\alpha}}{\Gamma(1-\alpha)} f(0) \end{aligned}$$

By using the approximation of the derivative by the method of finite differences (11) and by decomposing the integral, we obtain the following relation:

$$D_0^\alpha f(t_i) \cong \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{i-1} \frac{f_{j+1} - f_j}{h} \int_{t_j}^{t_{j+1}} \frac{1}{(t_i-u)^\alpha} du + \frac{t_i^{-\alpha}}{\Gamma(1-\alpha)} f_0$$

But the calculation of the integral $\int_{t_j}^{t_{j+1}} \frac{1}{(t_i-u)^\alpha} du$ gives ,

$$\int_{t_j}^{t_{j+1}} \frac{1}{(t_i-u)^\alpha} du = \frac{(t_i-t_j)^{1-\alpha} - (t_i-t_{j+1})^{1-\alpha}}{1-\alpha} \quad (12)$$

Substitute the integral by the relation (12), we get:

$$D_0^\alpha f(t_i) \cong \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{i-1} \frac{f_{j+1} - f_j}{h} \left[\frac{(t_i-t_j)^{1-\alpha} - (t_i-t_{j+1})^{1-\alpha}}{1-\alpha} \right] + \frac{t_i^{-\alpha}}{\Gamma(1-\alpha)} f_0$$

$$D_0^\alpha f(t_i) \cong \frac{1}{(1-\alpha)\Gamma(1-\alpha)} \sum_{j=0}^{i-1} \frac{f_{j+1} - f_j}{h} \left[(t_i-t_j)^{1-\alpha} - (t_i-t_{j+1})^{1-\alpha} \right] + \frac{t_i^{-\alpha}}{\Gamma(1-\alpha)} f_0$$

$$D_0^\alpha f(t_i) \cong \frac{1}{\Gamma(2-\alpha)} \sum_{j=0}^{i-1} \frac{f_{j+1} - f_j}{h} \left[(t_i-t_j)^{1-\alpha} - (t_i-t_{j+1})^{1-\alpha} \right] + \frac{t_i^{-\alpha}}{\Gamma(1-\alpha)} f_0$$

Since $t_i = ih$ and $t_j = jh$ then we have,

$$D_0^\alpha f(t_i) \cong \frac{1}{\Gamma(2-\alpha)} \sum_{j=0}^{i-1} \frac{f_{j+1} - f_j}{h} \left[(ih-jh)^{1-\alpha} - (ih-(j+1)h)^{1-\alpha} \right] + \frac{(ih)^{-\alpha}}{\Gamma(1-\alpha)} f_0$$

By factoring $h^{1-\alpha}$ we get the following equality:

$$D_0^\alpha f(t_i) \cong \frac{h^{1-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{i-1} \frac{f_{j+1} - f_j}{h} \left[(i-j)^{1-\alpha} - (i-(j+1))^{1-\alpha} \right] + \frac{(ih)^{-\alpha}}{\Gamma(1-\alpha)} f_0$$

$$D_0^\alpha f(t_i) \cong \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{i-1} (f_{j+1} - f_j) \left[(i-j)^{1-\alpha} - (i-(j+1))^{1-\alpha} \right] + \frac{(1-\alpha)(ih)^{-\alpha}}{(1-\alpha)\Gamma(1-\alpha)} f_0$$

$$D_0^\alpha f(t_i) \cong \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{i-1} (f_{j+1} - f_j) \left[(i-j)^{1-\alpha} - (i-(j+1))^{1-\alpha} \right] + \frac{(1-\alpha)h^{-\alpha}}{\Gamma(2-\alpha)} f_0$$

$$D_0^\alpha f(t_i) \cong h^{-\alpha} \left\{ \frac{1}{\Gamma(2-\alpha)} \sum_{j=0}^{i-1} (f_{j+1} - f_j) \left[(i-j)^{1-\alpha} - (i-(j+1))^{1-\alpha} \right] + \frac{(1-\alpha)}{\Gamma(2-\alpha)} f_0 \right\}$$

$$D_0^\alpha f(t_i) \cong \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \left\{ \sum_{j=0}^{i-1} (f_{j+1} - f_j) \left[(i-j)^{1-\alpha} - (i-(j+1))^{1-\alpha} \right] + (1-\alpha) f_0 \right\}$$

For j from 0 to i , we obtain the coefficients of f_j , noted $h^{-\alpha} c_{i,j}$ with:

$$c_{i,j} = \frac{1}{\Gamma(2-\alpha)} [(1-\alpha)i^{-\alpha} - i^{1-\alpha} + (i-1)^{1-\alpha}] \text{ for } j = 0$$

$$c_{i,j} = \frac{1}{\Gamma(2-\alpha)} [(i-j+1)^{-\alpha} - 2(i-j)^{1-\alpha} + (i-j-1)^{1-\alpha}] \text{ for } 1 \leq j \leq i-1$$

$$c_{i,j} = \frac{1}{\Gamma(2-\alpha)} \text{ for } j = i$$

The approximation of the right fractional derivative of Caputo of order α for the function φ is,

$${}^c D_{1-}^\alpha \varphi(t) = -I_{1-}^{1-\alpha} D\varphi(t) = \frac{-1}{\Gamma(1-\alpha)} \int_t^1 \frac{\varphi'(u)}{(u-t)^\alpha} du$$

at point t_i , using the formulate of method of finite differences (11) and decomposition of the integral (12), we have:

$${}^c D_{1-}^\alpha \varphi(t_i) \cong \frac{-1}{\Gamma(1-\alpha)} \sum_{j=i}^{N-1} \frac{\varphi_{j+1} - \varphi_j}{h} \int_{t_j}^{t_{j+1}} \frac{1}{(u-t_i)^\alpha} du$$

$${}^c D_{1-}^\alpha \varphi(t_i) \cong \frac{-1}{\Gamma(1-\alpha)} \sum_{j=i}^{N-1} \frac{\varphi_{j+1} - \varphi_j}{h} \left[\frac{(t_{j+1} - t_i)^{1-\alpha} - (t_j - t_i)^{1-\alpha}}{1-\alpha} \right]$$

Substitute t_i and t_j by ih and jh , the approximation becomes:

$${}^c D_{1-}^\alpha \varphi(t_i) \cong \frac{-1}{\Gamma(1-\alpha)} \sum_{j=i}^{N-1} \frac{\varphi_{j+1} - \varphi_j}{h} \left[\frac{((j+1)h - ih)^{1-\alpha} - (jh - ih)^{1-\alpha}}{1-\alpha} \right]$$

$${}^c D_{1-}^\alpha \varphi(t_i) \cong \frac{-h^{1-\alpha}}{\Gamma(1-\alpha)} \sum_{j=i}^{N-1} \frac{\varphi_{j+1} - \varphi_j}{h} \left[\frac{(j+1-i)^{1-\alpha} - (j-i)^{1-\alpha}}{1-\alpha} \right]$$

$${}^c D_{1-}^\alpha \varphi(t_i) \cong \frac{-h^{1-\alpha}}{\Gamma(1-\alpha)} \sum_{j=i}^{N-1} \frac{\varphi_{j+1} - \varphi_j}{h} \left[\frac{(j+1-i)^{1-\alpha} - (j-i)^{1-\alpha}}{1-\alpha} \right]$$

$${}^c D_{1-}^\alpha \varphi(t_i) \cong \frac{-h^{-\alpha}}{(1-\alpha)\Gamma(1-\alpha)} \sum_{j=i}^{N-1} (\varphi_{j+1} - \varphi_j) [(j-i+1)^{1-\alpha} - (j-i)^{1-\alpha}]$$

$${}^c D_{1-}^\alpha \varphi(t_i) \cong \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=i}^{N-1} (\varphi_{j+1} - \varphi_j) [(j-i)^{1-\alpha} - (j-i+1)^{1-\alpha}]$$

For $i \leq j \leq N-1$, we obtain the coefficients noted $h^{-\alpha} d_{i,j}$ of the function φ_j , with :

$$d_{i,j} = \frac{1}{\Gamma(2-\alpha)} \text{ for } j = i$$

$$d_{i,j} = \frac{1}{\Gamma(2-\alpha)} [(j-i+1)^{1-\alpha} - 2(j-i)^{1-\alpha} + (j-i-1)^{1-\alpha}]$$

for $i+1 \leq j \leq N-1$

$$d_{i,j} = \frac{1}{\Gamma(2-\alpha)} [(N-i-1)^{1-\alpha} - (N-i)^{1-\alpha}] \text{ for } j = N$$

Thus, by composition of the two operators, we can write the initial equation (9) in the following discrete form:

$$\begin{cases} f_0 = 0 \\ \left[\frac{h^{-2\alpha}}{\Gamma(2-\alpha)} \sum_{j=i}^N d_{i,j} \sum_{k=0}^j c_{j,k} f_k \right] + r f_i = m g_i \text{ for } (13) \ 1 \leq i \leq N-1 \\ f_N = 0 \end{cases}$$

For i ranging from 1 to $N-1$, we get, for each value of i , the following expanded form:

$$a_{i,0} f_0 + a_{i,1} f_1 + \dots + a_{i,i} f_i + \dots + a_{i,N} f_N = m g_i \quad (14)$$

Where the coefficients $a_{i,k}$, for $0 \leq k \leq N$, are defined by:

$$a_{i,k} = \left(\frac{1}{N} \right)^{-2\alpha} \sum_{j=i}^N d_{i,j} c_{j,k} \text{ for } 0 \leq k < i$$

$$a_{i,k} = \left(\frac{1}{N} \right)^{-2\alpha} \sum_{j=i}^N d_{i,j} c_{j,k} + r \text{ for } k = i \quad (15)$$

$$a_{i,k} = \left(\frac{1}{N} \right)^{-2\alpha} \sum_{j=k}^N d_{i,j} c_{j,k} \text{ for } i+1 \leq k \leq N$$

Thus, we can write the following system of $N+1$ equations of unknowns f_i

$$\begin{cases} f_0 = 0 \\ a_{i,0} f_0 + a_{i,1} f_1 + \dots + a_{i,N} f_N = m g_i \\ \dots \\ a_{N-1,0} f_0 + a_{N-1,1} f_1 + \dots + a_{N-1,N} f_N = m g_{N-1} \\ f_N = 0 \end{cases} \quad (16)$$

This system can be translated by matrix form:

$$A_{i,k} \times F_i = G_i \text{ where: } (17)$$

$$A_{i,k} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ a_{i,0} & a_{i,1} & \dots & a_{i,N} \\ \dots & \dots & \dots & \dots \\ a_{N-1,0} & a_{N-1,1} & \dots & a_{N-1,N} \\ 0 & 0 & \dots & 1 \end{pmatrix}, \text{ the matrix of the coefficients of } f_i \text{ of the system}$$

$$F_i = \begin{pmatrix} f_0 \\ f_1 \\ \dots \\ f_{N-1} \\ f_N \end{pmatrix} \text{ the matrix of unknowns } f_i \text{ with } i \text{ rows, } 0 \leq i \leq N$$

$$G_i = \begin{pmatrix} 0 \\ m g_1 \\ \dots \\ m g_{N-1} \\ 0 \end{pmatrix} \text{ the matrix equation of the second member.}$$

These coefficients can be calculated with programming under Matlab language that we will present below.

Programming under Matlab language and numerical examples

In this programming, firstly, we presented the case where N fixed and α varies and secondly the case where α fixed and N varies to know the influences of the values of N and those of the values of α in the results obtained graphically.

The following programming allow us to calculate step by step these coefficients $c_{j,k}$, $d_{i,j}$, $a_{i,k}$ in order to represent the functions $f(N, \alpha)$ graphically.

In this section, the numerical results of calculations are presented. In presented solutions of equations, we assume the following parameters: for N fixed, we choose two values of N , $N=100$ and $N=200$, $\alpha \in \{0, 0.1, 0.2, 0.8, 0.9, 1\}$ but for α fixed, $\alpha = 0.1$, $\alpha = 0.8$, $\alpha = 0.9$, $N \in \{200, 300, 500, 1000\}$. Here, the function g , the constant r and m are given in the programming. These values of g , r and m can be changed in this programming.

In this example of programming, we take $g(t) = 2\pi^2 \sin(\pi t)$, $r = \pi^2$, $m = 2\pi^2$ (17).

Calculation function of $f(N, \alpha)$

```
function f=varalpha(N,alpha)
```

```
% N=input('value of N=')
```

```
% alpha=input('value of alpha=')
```

```
c = zeros(N,N+1);
```

```
% Calculation of c
```

```
% *****
```

```
for j=1:N
```

```
    for k=0:j
```

```
        if k ==0
```

```
            ta = 1/gamma(2-alpha);
```

```
            tb = (1 - alpha)*j^(-alpha);
```

```
            tc = (j)^(-1-alpha);
```

```
            td = (j - 1)^(-1-alpha);
```

```
            c(j,k+1) = ta * (tb - tc + td);
```

```
        elseif k < j
```

```
            ta = 1/gamma(2-alpha);
```

```
            tb = (j - k + 1)^(-1-alpha);
```

```
            tc = (j - k)^(-1-alpha);
```

```
            td = (j - k - 1)^(-1-alpha);
```

```
            c(j,k+1) = ta * (tb - 2*tc + td);
```

```
        else
```

```
            c(j,k+1) = 1/gamma(2-alpha);
```

```
        end
```

```
    end
```

```
end
```

```
% Calculation of d
```

```
% *****
```

```
d = zeros(N,N);
```

```
for i=1 : N-1
```

```
    for j= i : N
```

```
        if j == i && j ~= N
```

```
            d(i,j) = 1/gamma(2-alpha);
```

```
        elseif j < N
```

```
            tad = 1/gamma(2-alpha);
```

```
            tbd = (j - i + 1)^(-1-alpha);
```

```
            tcd = (j - i)^(-1-alpha);
```

```
            tdd = (j - i - 1)^(-1-alpha);
```

```
            d(i,j) = tad * (tbd - 2*tcd + tdd);
```

```
        else
```

```
            t1d = (N - i - 1)^(-1 - alpha);
```

```
            t2d = (N - i)^(-1 - alpha);
```

```
            d(i,j) = (t1d - t2d)/gamma(2-alpha);
```

```
        end
```

```
    end
```

```
end
```

```
% matrix coefficients
```

```
*****
```

```
b = 1;
```

```
a = zeros(N+1,N+1);
```

```
ce = (b / N)^(-2*alpha);
```

```
r = pi^2;
```

```
for i = 1 : N - 1
```

```
    for k = 0 : N
```

```
        S = 0;
```

```
        if k < i
```

```
            for j = i : N
```

```
                S = S + d(i,j)*c(j,k+1);
```

```
            end
```

```
            a(i+1,k+1) = ce * S;
```

```
        elseif k == i
```

```
            for j = i : N
```

```
                S = S + d(i,j)*c(j,k+1);
```

```
            end
```

```
            a(i+1,k+1) = ce * S + r;
```

```
        else
```

```
            for j = k : N
```

```
                S = S + d(i,j)*c(j,k+1);
```

```
            end
```

```
            a(i+1,k+1) = ce * S;
```

```
        end
```

```
    end
```

```
end
```

```
cc=1;
```

```
bb=zeros(N+1,1);
```

```

for i = 2: N
    bb(i,1)=2.*pi^2.*sin(pi.*(i/N));
end
a(1,1)=1;
a(N+1,N+1)=1;
bb(1,1)=0 ;
bb(N+1,1)=0 ;
f = a\bb;
for i=1:N+1
    t(i)=i.*(1/N)
end
plot(t,f)
grid minor

```

Analyzing the graphical representations seen previously, we note that when α is fixed and N varies then the shape of all the curves are the same and they are confused for all the values of N but the maximum value of the approximate solution f_i reached is the same at the point $t_i=0.5$, and decreases when the value of α increases. For fixed N and variable α , the shape of the curves does not change and the curves are different for different values of α but the maximum values of the functions f_i as a function of t_i remain the same for different values of N . We also observe that when α is close to 0 then the curves approach the curve corresponding to $\alpha=0$ and they approach the curve corresponding to $\alpha=1$ when α is close to 1 (Figures 1 and 2).

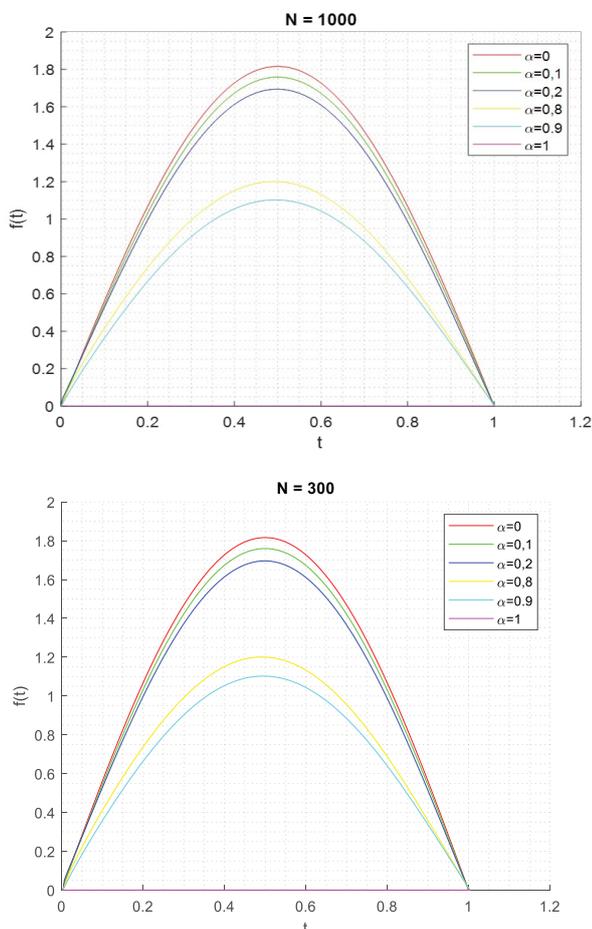


Figure 1. Approximate solutions for fixed N and variable α .

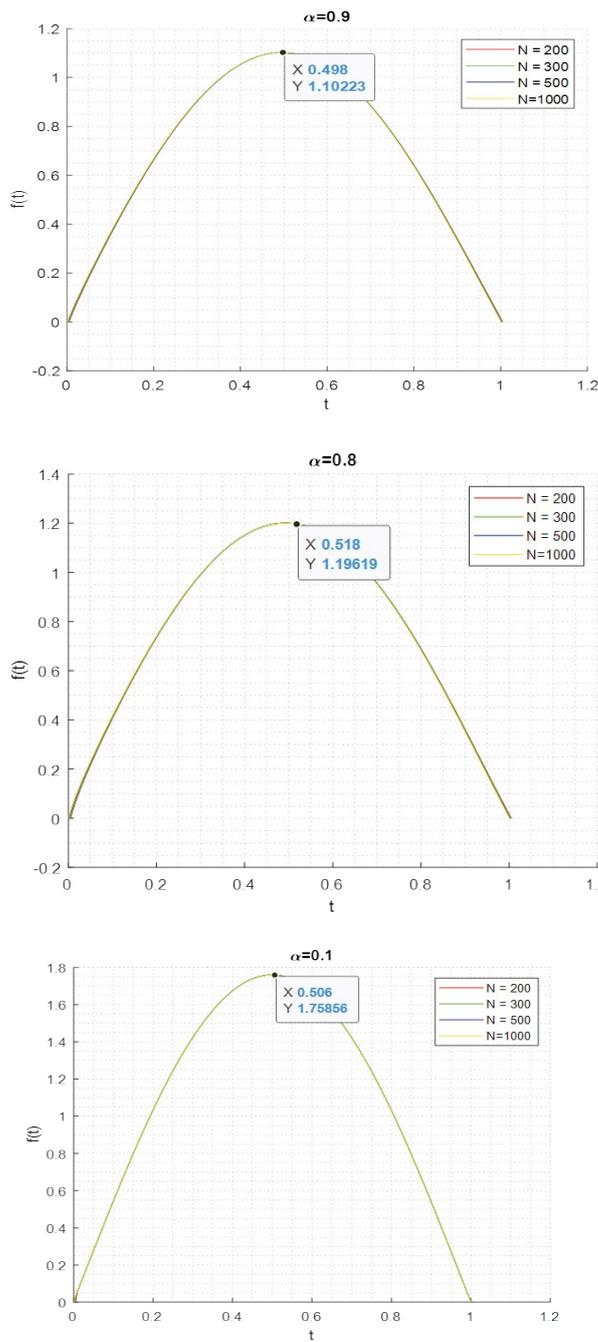


Figure 2. Approximate solutions for fixed α and variable N .

Conclusion

In our work, we opted for a numerical method using programming under Matlab one of the methods allowing to find the coefficients of the matrix in order to obtain approximate solutions of the function at the point for two different cases by fixing and by varying and by fixing and by varying. Our perspective is to apply this method for different problems in certain fields such as sciences, mechanics, etc.

References

1. Kilbas, Anatolii Aleksandrovich, Hari M. Srivastava and Juan J. Trujillo. "Theory and applications of fractional differential equations." *Elsevier* 204 (2006): 540.
2. Dubois, F., A.C. Galucio and N. Point. "Introduction to fractional derivation." *Theory Appl Eng Tech Series* (2009).

3. Samko, Stefan G., Anatoly A. Kilbas and Oleg I. Marichev. "Fractional integrals and derivatives, translated from the 1987 Russian original." (1993): 1749-024.
4. Baleanu, Dumitru and Juan J. Trujillo. "On exact solutions of a class of fractional Euler–Lagrange equations." *Nonlinear Dynamics* 52 (2008): 331-335.
5. Agrawal, Om P. "A numerical scheme and an error analysis for a class of fractional optimal control problems." *IDETC/CIE* 49019 (2009): 1253-1260.
6. Klimek, M. "Solutions of Euler-Lagrange equations in fractional mechanics." In AIP Conference Proceedings. *American Institute of Physics* 956 (2007): 73-78.
7. Agrawal, Om P. "Analytical schemes for a new class of fractional differential equations." *J Phys A Math Theor* 40 (2007): 5469.
8. Klimek, Małgorzata. "Fractional sequential mechanics-models with symmetric fractional derivative." *Czechoslov J Phys* 51 (2001): 1348-1354.
9. Goncalves, Eric. "Résolution numérique, discrétisation des EDP et EDO." *Institut National Polytechnique de Grenoble* (2005).
10. Riewe, Fred. "Nonconservative lagrangian and hamiltonian mechanics." *Phys Rev E* 53 (1996): 1890.
11. Riewe, Fred. "Mechanics with fractional derivatives." *Phys Rev E* 55 (1997): 3581.
12. Morel, J. M., Takens F. and Teissier B. "The analysis of fractional differential equations." *Lecture Notes in Mathematics*. Springer, Berlin (2004).
13. Hilfer, Rudolf, ed. "Applications of fractional calculus in physics." *World scientific* (2000).
14. Yang, Qianqian, Fawang Liu, and Ian Turner. "Numerical methods for fractional partial differential equations with Riesz space fractional derivatives." *Appl Math Model* 34 (2010): 200-218.
15. Robert, Janin. "Non-integer Derivatives and Integrals." (2010).
16. Oldham, Keith and Jerome Spanier. "The fractional calculus theory and applications of differentiation and integration to arbitrary order." *Elsevier* (1974).
17. Błaszczyk, Tomasz. "Application of the Rayleigh-Ritz method for solving fractional oscillator equation." *J Appl Math Comput Mech* 8 (2009): 29-36.
18. Agrawal, Om P. "Formulation of Euler–Lagrange equations for fractional variational problems." *J Math Anal Appl* 272 (2002): 368-379.

How to cite this article: Raharimina, L.H.D.L, G. Rasolomampandry and F. Randimbindrainibe. "One Method for the Numerical Resolution of One Type of Fractional Differential Equation of Euler-Lagrange" *J Appl Computat Math* 11 (2022): 475.