Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/072L22

A Numerical Method for Delayed Fractional-Order Differential Equations: Based on G-L Definition

Zhen Wang[1,2], *Xia Huang*[3] *and Jianping Zhou*[4]

[1]College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2] School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China
[3]Key Laboratory of Robotics and Intelligent Technology, College of Information and Electrical Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[4]School of Computer Science, Anhui University of Technology, Ma'anshan 243002, China

Received: 19 Oct. 2012, Revised: 4 Jan. 2013, Accepted: 6 Jan. 2013
Published online: 1 Jun. 2013

Abstract: In this paper, a numerical method for nonlinear fractional-order differential equations with constant or time-varying delay is devised. The order here is an arbitrary positive real number, and the differential operator is the Grünwald-Letnikov derivative. The detailed error analysis for this algorithm is given, meanwhile, the convergence of the iteration algorithm is proved. Compared with the exact analytical solution, a numerical example is provided to illustrate the effectiveness of the proposed method.

Keywords: Delay, fractional-order differential equations, Grünwald-Letnikov derivative

1 Introduction

Fractional calculus is an old mathematical problem, and mainly developed as a pure mathematics problem for nearly three centuries [1]-[3]. Though having a long history, it was not used in physics and engineering for a long period. However, in the last few decades, fractional calculus began to attract increasing attention of scientists from the viewpoint of application [3]-[6]. Fractional derivative has proven to be a very suitable tool for the description of memory and hereditary properties of various materials and processes. In the fields of continuous-time modeling, many researchers pointed out that fractional derivative is very useful in linear viscoelasticity, acoustics, rheology, polymeric chemistry, etc [7,12]. Nowadays, the mathematical theories and practical applications of these operators are well established, and their applicabilities to science and engineering are being considered as an attractive topic.

The development of effective and well-suited methods for numerically solving FDEs has drawn more and more attention over the last few years. Several numerical methods based on Caputo or Riemann-Liouville definition have been proposed and analyzed [13]-[32]. For instance, based on the predictor-corrector scheme, Diethelm et al introduced Adams-Bashforth-Moulton algorithm [17,18,19], and meanwhile some error analyses and an extension of Richardson extrapolation were also presented to improve the numerical accuracy. These techniques elaborated in [17,18,19] take advantage of the fact that the FDEs can be reduced to Volterra type integral equations. And therefore, one can apply the numerical schemes for Volterra type integral equations to find the solution of FDEs, the readers can refer to [20]-[24] and the literature cited therein for more details. The technique presented in [17,18,19] has been further analyzed and extended to multi-term [25] and multi-order systems [26]. Li et al. [32] studied the error analysis of the fractional Adams method for fractional-order ordinary differential equations in more general case. Deng [14] obtained a good numerical approximation by combining the short memory principle with the predictor-corrector approach.

However, in practice, delay is very often encountered in different technical systems, such as automatic control, biology and hydraulic networks, economics, long transmission lines, etc. Consequently, delayed differential equations are used to describe such kinds of dynamical systems. In recent years, delayed FDEs begin to arouse the attention of a number of researchers [33]-[35]. It is well known that finding robust and stable numerical

[*] Corresponding author e-mail: huangxia.qd@gmail.com

methods for solving the delayed FDEs is debatable and has become an open research question. To the best of our knowledge, there are very few works devoted in the literature to this problem so far. Since the Riemann-Liuville derivative and the Grünwald-Letnikov derivative are the same in effect [3], this gives an approximate method for fractional order derivatives. Therefore, this paper will study the numerical algorithm for the delayed FDEs based on the Grünwald-Letnikov definition.

The rest of this paper is organized as follows: In Section 2, basic definition and preliminaries in fractional calculus are presented. In Section 3, the numerical scheme is devised in the case of constant and time-varying delay, meanwhile, the convergence of the proposed algorithm is proved. The iteration algorithm is designed in Section 4. Numerical simulations are performed in Section 5 to illustrate the effectiveness of the proposed scheme. Finally, some concluding remarks are reported in Section 6.

2 Basic definition and preliminaries

There are several different definitions of fractional integration and differentiation till now [3]. The most frequently used are the Grünwald-Letnikov (G-L) definition and the Riemann-Liuville (R-L) definition. For a wide class of functions, the two definitions G-L and R-L are equivalent. In this paper, the Grünwald-Letnikov definition will be used.

Definition 1.*[3]. The Grünwald-Letnikov fractional derivative of order α is defined as follows*

$$ {}_aD_t^{\alpha}y(t) = \lim_{h\to 0}\sum_{j=0}^{[\frac{t-a}{h}]}(-1)^j\binom{\alpha}{j}y(t-jh), $$

where $[\cdot]$ denotes the integer part, $\alpha \in R$ is the order of the derivative, a is the initial time, and h is the sampling time.

In [36], Lubich proposed a second order numerical method for the fractional derivative of order α

$$ {}_aD_t^{\alpha}y(t) \approx \frac{1}{h^{\alpha}}\sum_{j=0}^{[\frac{t-a}{h}]}\omega_j^{(\alpha)}y(t-jh), \tag{1} $$

where

$$ \omega_j^{(\alpha)} = \frac{1}{2\pi i}\int\limits_{|x|\le 1}\frac{W_2(x)}{x^{j+1}}dx = \frac{1}{2\pi}\int\limits_0^{2\pi}W_2(e^{-i\phi})e^{ji\phi}d\phi, $$

and

$$ W_2(x) = (\frac{3}{2}-2x+\frac{x^2}{2})^{\alpha}. $$

3 Formulation of the numerical method for delayed FDEs

The main aim of this paper is to study a numerical scheme for the approximate solution of delayed FDEs. For this purpose, we consider delayed FDEs described as follows

$$ \begin{aligned} &{}_aD_t^{\alpha}y(t) = f(t,y(t),y(t-\tau)),\\ &\quad (a\le t\le b, m-1<\alpha\le m)\\ &\qquad y(t)=\varphi(t), \qquad t\le a, \end{aligned} \tag{2} $$

where α is the order of the differential equation, $\varphi(t)$ is the initial value, and m is an integer. By the Grünwald-Letnikov definition, the discrete form of system (3) can be written as

$$ h^{-\alpha}\sum_{j=0}^{n}\omega_j^{(\alpha)}y(t_{n-j}) + O(h^2) = f(t_n, y(t_n), y(t_n-\tau)). \tag{3} $$

Now, the key problem is to establish the approximation to the delayed term $y(t-\tau)$, which contains two cases, discussed as below.

Case I: when τ is constant

It can be seen that, for any positive constant τ, $t_j-\tau$ may not be a grid point t_n for any n. Suppose $(m+\delta)h=\tau$, and $0\le\delta<1$. When $\delta=0$, $y(t_n-\tau)$ can be approximated by

$$ y(t_n-\tau)\approx\begin{cases} y_{n-m}, & if\ n>m,\\ \varphi(t_n-\tau), & if\ n\le m,\end{cases} $$

and when $0<\delta<1$, $y(t_n-\tau)$ cannot be calculated directly.

By virtue of Taylor expansion and the numerical differentiation technique, we have

$$ \begin{aligned} & y(t_n-\tau)\\ &= y(t_n-(m+\delta)h) = y(t_{n-m-1}+(1-\delta)h)\\ &= y(t_{n-m-1}) + y'(t_{n-m-1})(1-\delta)h + O(h^2)\\ &\approx y(t_{n-m-1}) + [\frac{y(t_{n-m})-y(t_{n-m-1})}{h} + O(h^2)](1-\delta)h\\ &\quad +O(h^2)\\ &= (1-\delta)y(t_{n-m}) + \delta y(t_{n-m-1}) + O(h^2). \end{aligned} \tag{4} $$

Let v_n be the approximation to $y(t_n-\tau)$, and the numerical approximation for the computation of $y(t-\tau)$ are proposed as follows

$$ v_n = y(t_n-\tau)\approx(1-\delta)y_{n-m}+\delta y_{n-m-1}. \tag{5} $$

It can be seen from (5) that, if $m>0$, the numerical equation is explicit, and thus it can be computed directly. However, when $m=0$ and $\delta\ne 1$, i.e., $\tau<h$, the first term in the right-hand side of the above equation is $(1-\delta)y_n$. It still needs to predict, for this case, v_n is calculated as

$$ v_n = (1-\delta)y_n^{(l)} + \delta y_{n-1} $$

where $l=1,2,\cdots$ is the iteration number.

Case II: when τ is time varying

If τ is time varying, i.e., $\tau = \tau(t)$, then the approximation in this case is more complicated. Let v_n is the approximation to $y(t_n - \tau)$, and the linear interpolation of $\{y_j\}$ at point $t = t_n - \tau(t_n)$ is employed to approximate the delay term. Let $\tau(t_n) = (m_n + \delta_n)h$, where m_n is a positive integer and $\delta_n \in [0,1)$, then

$$v_n = (1-\delta_n)y_{n-m_n} + \delta_n y_{n-m_n-1}. \tag{6}$$

It should be noted that when τ is constant, for given h and τ, at the beginning of the programme, we can judge whether $m = 0$ or $m > 0$. However, when τ is time varying, m is also time varying, that is, at one moment it may equal to 0, but at another moment it may greater than 0. If $m_n = 0$, the first term in the right-hand side of (6) still needs to predict, otherwise, when $m_n > 0$, it does not need. So in each step of the computing, it needs to test weather $m_n = 0$ or not firstly, then computation of the first term in the right-hand side of Eq.(6) depends on whether it requires predicting or not in different steps.

By the above discussions, the numerical scheme for the FDEs (3) can be depicted as:

$$\sum_{j=0}^{n} \omega_j^{(\alpha)} y_{n-j} = h^\alpha f(t_n, y_n, v_n), \tag{7}$$

$$v_n = \begin{cases} (1-\delta)y_{n-m} + \delta y_{n-m-1}, & if\ n > m, m > 0, \\ (1-\delta)y_n^{(l)} + \delta y_{n-1}, & if\ n > m, m = 0, \\ \varphi(t_n - \tau), & if\ n \le m, \end{cases} \tag{8}$$

where $l = 1,2,\cdots$ is the iteration number and $n = 1,2,\cdots,N$.

Theorem 1. *Suppose the function $f(t,y,u)$ satisfies the following Lipschitz conditions*

$$|f(t,y_1,u) - f(t,y_2,u)| \le L_1|y_1 - y_2|, \tag{9}$$

$$|f(t,y,u_1) - f(t,y,u_2)| \le L_2|u_1 - u_2|, \tag{10}$$

then for the fractional order differential equation (3), the local truncation error of the numerical scheme (7), (8) is $O(h^{2+\alpha})$.

Proof. Since the numerical scheme (7) can be written as

$$\omega_0^{(\alpha)} y_n + \sum_{j=1}^{n} \omega_j^{(\alpha)} y_{n-j} = h^\alpha f(t_n, y_n, v_n), \tag{11}$$

we have

$$\omega_0^{(\alpha)}(y_n - y(t_n)) + \omega_0^{(\alpha)} y(t_n) + \sum_{j=1}^{n} \omega_j^{(\alpha)} y_{n-j} \tag{12}$$

$$\begin{aligned} &= h^\alpha[f(t_n,y_n,v_n) - f(t_n,y_n,y(t_n-\tau)) \\ &\quad + h^\alpha f(t_n,y(t_n),y(t_n-\tau)). \end{aligned} \tag{13}$$

Suppose

$$y_j = y(t_j), j = 0,1,\cdots,n-1,$$

then Eq.(13) can be rewritten as

$$\begin{aligned} &\omega_0^{(\alpha)}(y_n - y(t_n)) + \sum_{j=0}^{n} \omega_j^{(\alpha)} y(t_{n-j}) = \\ &f(t_n,y(t_n),y(t_n-\tau)) + [f(t_n,y_n,v_n) - f(t_n,y_n,y(t_n-\tau)). \end{aligned} \tag{14}$$

From Lubich numerical expression of fractional derivative (1), we have

$$\begin{aligned} &\frac{1}{h^\alpha}\omega_0^{(\alpha)}(y_n - y(t_n)) + {}_\alpha D_t^\alpha y(t) \\ &= h^\alpha[f(t_n,y_n,v_n) - f(t_n,y_n,y(t_n-\tau))] \\ &\quad + h^\alpha f(t_n,y(t_n),y(t_n-\tau)) + O(h^2), \end{aligned} \tag{15}$$

Eliminating the identical term in the above equation, we have

$$\begin{aligned} &\frac{1}{h^\alpha}\omega_0^{(\alpha)}(y_n - y(t_n)) \\ &= f(t_n,y_n,v_n) - f(t_n,y_n,y(t_n-\tau)) + O(h^2). \end{aligned} \tag{16}$$

From the assumptions (9) and (10), we get

$$(\omega_0^{(\alpha)} - Lh^\alpha)|y_n - y(t_n)| \le O(h^{2+\alpha}). \tag{17}$$

Thus, for sufficient small h, the local truncation error is $O(h^{2+\alpha})$, which completes the proof.

4 Design of the numerical algorithm

It is obvious that(7), (8) is an implicit nonlinear algebraic equation with respect to y_n. In order to solve y_n, we construct the iteration algorithm as follows:

$$y_n^{(l)} = \frac{1}{\omega_0^\alpha}[h^\alpha f(t_n, y_n^{(l-1)}, v_n^{(l-1)}) - \sum_{j=1}^{n} \omega_j^{(\alpha)} y_{n-j}], \tag{18}$$

$$v_n^{(l-1)} = \begin{cases} (1-\delta)y_{n-m} + \delta y_{n-m-1}, & if\ n > m, m > 0, \\ (1-\delta)y_{n-m}^{(l-1)} + \delta y_{m-1}, & if\ n > m, m = 0, \\ \varphi(t_n - \tau), & if\ n \le m, \end{cases} \tag{19}$$

$$y_0 = \varphi(a), y_n^{(0)} = 1, \tag{20}$$

$$l = 1,2,\cdots,n = 1,2,\cdots,N. \tag{21}$$

where l is the iteration number. If $|y_n^{(l)} - y_n^{(l-1)}| < \varepsilon$ (ε is the given error, e.g. $\varepsilon = 10^{-6}$), we would consider y_n as $y_n^{(l)}$.

Theorem 2. *Suppose the function $f(t,y,u)$ satisfies the Lipschitz conditions (9), (10), then the iteration algorithm (18)-(21) is convergent.*

Proof. Since

$$\begin{aligned}
&|y_n^{(l)} - y_n^{(l-1)}| \qquad (22)\\
&= \frac{h^\alpha}{\omega_0^{(\alpha)}} |f(t_n, y_n^{(l-1)}, v_n^{(l-1)}) - f(t_n, y_n^{(l-2)}, v_n^{(l-2)})|\\
&\le \frac{h^\alpha}{\omega_0^{(\alpha)}} \Big\{ |f(t_n, y_n^{(l-1)}, v_n^{(l-1)}) - f(t_n, y_n^{(l-2)}, v_n^{(l-1)})|\\
&\quad + |f(t_n, y_n^{(l-2)}, v_n^{(l-1)}) - f(t_n, y_n^{(l-2)}, v_n^{(l-2)})| \Big\}\\
&\le \frac{h^\alpha}{\omega_0^{(\alpha)}} \Big\{ L_1 |y_n^{(l-1)} - y_n^{(l-2)}| + L_2 |v_n^{(l-1)} - v_n^{(l-2)}| \Big\}\\
&\le \frac{h^\alpha}{\omega_0^{(\alpha)}} \Big\{ L_1 |y_n^{(l-1)} - y_n^{(l-2)}| + L_2(1-\delta) |y_n^{(l-1)} - y_n^{(l-2)}| \Big\}\\
&\le \frac{h^\alpha}{\omega_0^{(\alpha)}} (L_1 + L_2) |y_n^{(l-1)} - y_n^{(l-2)}|\\
&\le \cdots \le \left(\frac{h^\alpha}{\omega_0^{(\alpha)}} L \right)^{l-1} |y_n^{(1)} - y_n^{(0)}|
\end{aligned}$$

then for sufficient small h satisfying $\left(\frac{h^\alpha}{\omega_0^{(\alpha)}} L \right) < 1$, when $l \to \infty$, we have $|y_n^{(l)} - y_n^{(l-1)}| \to 0$, that is, the iteration algorithm (18)-(21) is convergent.

Now, we summarize the computing procedure for the iteration algorithm as follows

(1) For given initial value $y_0 = \varphi(a)$, time length h, and the tolerate error ε, $N = \frac{b-a}{h}$, $n = 1$, let $t_n = a + nh$, compute $\omega_0^{(\alpha)}$.

(2) If $m > n$, break. Otherwise, compute $\tau = \tau(t_n)$, $\omega_n^{(\alpha)}$, $sum = \sum\limits_{j=1}^{m} \omega_n^{(\alpha)} y_{n-j}$.

(3) Let $l = 1, y_n^{(l-1)} = c$, compute v_n.

(4) Compute $y_m^{(l)} = [sum - h^\alpha f(t_n, y_n^{(l-1)}, v_n)]/\omega_0^{(\alpha)}$.

(5) If $|y_n^{(l)} - y_n^{(l-1)}| < \varepsilon$, then $y_n = y_n^{(l)}, n = n+1$, and return to the step (2). Otherwise, turn into the next step.

(6) Let $l = l+1$, compute v_n, $y_m^{(l)} = [sum - h^\alpha f(t_n, y_n^{(l-1)}, v_n)]/\omega_0^{(\alpha)}$, and return to the step (5).

5 A numerical example

In this section, the following delayed FDE is considered:

$$\begin{aligned}
D_t^\alpha y(t) &= \tfrac{2}{\Gamma(3-\alpha)} t^{2-\alpha} - \tfrac{1}{\Gamma(2-\alpha)} t^{1-\alpha} + 2\tau t - \tau^2\\
&\quad -\tau - y(t) + y(t-\tau), \quad \alpha \in (0,1),\\
y(t) &= 0, \quad t \le 0.
\end{aligned}$$

Notice that the exact solution to this equation is

$$y(t) = t^2 - t.$$

In accordance with delay τ being constant or time-varying, the numerical results are displayed in Table 1 and Table 2 respectively, where E_A denotes the absolute numerical error and E_R denotes the relative numerical error. From the numerical results we can see that the computing errors are in general acceptable for engineering.

Table 1. Numerical solution, exact solution and the error estimate at time $t = T$ when $h = 1/20$

T	$y(t_n)$	y_n	E_A	E_R
2	2	2.0034	0.0034	0.0017
4	12	12.0032	0.0032	0.0003
6	30	30.0029	0.0029	0.0001
8	56	56.0029	0.0029	0.0001
10	90	90.0032	0.0032	0.0000

Table 2. Error behavior at time $t = T$ with analytical value with $\alpha = 0.9, \tau = 0.1e^{-10t}$

T	E	$h = 1/10$	$h = 1/20$	$h = 1/40$
2	E_A	3.4859e-2	1.4422e-2	5.2452e-3
	E_R	1.7429e-2	7.2110e-3	2.6226e-3
10	E_A	3.0599e-2	1.2563e-2	4.7581e-3
	E_R	3.3956e-4	1.3959e-4	5.2868e-5
20	E_A	2.9195e-2	1.2820e-2	8.0554e-3
	E_R	7.6829e-5	3.3737e-5	2.1198e-5

6 Conclusions

In this paper, a numerical algorithm is formulated based on Grünwald-Letnikov derivative for fractional-order differential equations with time delay. The error analysis of the numerical scheme is carried out, meanwhile, a numerical example with constant delay and time varying delay is proposed to testify the effectiveness of the proposed scheme. This algorithm can be used not only in the simulation of delayed fractional-order differential equations but also in the simulation of delayed fractional-order control systems.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under grants 61004078, 61174038, 61104064 and the Natural Science Foundation of Jiangsu Province under grant BK2010493.

References

[1] P. L. Butzer, U.Westphal, An Introduction to Fractional Calculus, World Scientific, Singapore, 2000.

[2] S. M. Kenneth, R. Bertram, An Introduction to the Fractional Calculus and Fractional Differential Equations, Wiley-Interscience Publication, US, 1993.

[3] I. Podlubny, Fractional Differential Equations, New York: Academic, 1999.

[4] Z. Wang, X. Huang, N. Li and X. Song, Chin. Phys. B, **5** 050506(2012).

[5] K. Oldham and J. Spanier, The Fractional Calculus: Theory and Application of Differentiation and Integration to Arbitrary Order, New York: Academic, 1974.

[6] R. Hilfer, Applications of Fractional Calculus in Physics, Hackensack, NJ: World Scientific, 2001.

[7] P. J. Torvik, and R. L. Bagley, J. Appl. Mech., **51**, 294-298(1984).

[8] Heaviside O 1971 Electromagnetic Theory, New York: Chelsea.

[9] H. G. Sun , W. Chen and Y. Q. Chen, Physica A, **388**, 4586-4592(2009).

[10] R. X. Zhang and S. P. Yang, Chin. Phys. B, **18** 090512(2011).

[11] Y. Q. Chen , H. S. Ahnand and D. Y. Xue, Signal Processing, **86** 2794-2802(2006).

[12] F. Mainardi, Chaos, Solitons and Fractals, **7**, 1461-1477(1996).

[13] V. Daftardar-Gejji, A. Babakhani, J. Math. Anal. Appl. **293**, 511-522(2004).

[14] W. Deng, Journal of Computational and Applied Mathematics, **206**, 174-188(2007).

[15] K. Diethelm, Numer. Algor., **47**, 361-390(2008).

[16] K. Diethelm, Journal of Vibration and Acoustics, **131**, article 014502(2009).

[17] K. Diethelm, N. J. Ford, J. Math. Anal. Appl., **265**, 229-248(2002).

[18] K. Diethelm, N. J. Ford, A. D. Freed, Nonlinear Dynamics, **29**, 3-22(2002).

[19] K. Diethelm, N. J. Ford, A. D. Freed, Numer. Algorithms, **36**, 31-52(2004).

[20] C. Li, G. Peng, **22**, 430-450(2004).

[21] R. Garrappa, Journal of Computational and Applied Mathematics, **229**, 392-399(2009).

[22] X. J. Wu, J. Li, and G. Chen, Journal of the Franklin Institute, **345**,392-401(2008).

[23] S. T. Mohammad, H. Mohammad, Physica D, **237**, 2628-2637(2008).

[24] S. Dadras, H. R. Momeni, Physica A, **389**, 2434-2442(2010).

[25] K. Diethelm, N. J. Ford, Appl. Math. Comput., **154**, 621-640(2004).

[26] J. T. Edwards, N. J. Ford, C. A. Simpson, J. Comput. Appl. Math., **148**, 401-418(2002).

[27] K. Diethelm, N. J. Ford, A. D. Freed, Y. Luchko, Comput. Methods Appl. Mech. Eng., **194**, 743-773(2005).

[28] N. J. Ford, A. C. Simpson, Numer. Algorithms, **26**, 336-346(2001).

[29] L. Galeone, R. Garrappa, Mathematics and Computers in Simulation, **79**, 1358-1367(2008).

[30] L. Galeone, R. Garrappa, Journal of Computational and Applied Mathematics, **228**, 548-560(2009).

[31] Z. Odibat, Appl. Math. Comput., **178**, 527-533(2006).

[32] C. Li, C. Tao, Computers and Mathematics with Applications, **58**, 1573-1588(2009).

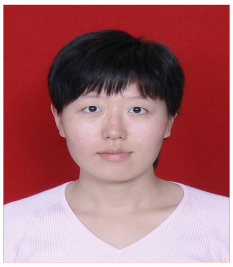
[33] Y. Chen, K. Moore, Nonlinear Dynamics, **29**, 191-200(2002).

[34] P. Lanusse, H. Benlaoukli, D. Nelson-Gruel and A. Oustaloup, IET Control Theory Appl., **1**, 16-23(2008).

[35] X. Zhang, Applied mathematics and computation, 197(2008), 407-411.

[36] CH. Lubich, Siam J. Math. Anal., **17**, 704-719(1986).

Zhen Wang is a lecturer at the College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China. His current research interest covers fractional-order systems, neural networks, and complex networks.



Xia Huang is an associate Professor of Shandong University of Science and Technology, Qingdao 266590, China. Her current research interest covers neural networks, chaos control and synchronization, and fractional-order nonlinear systems. Corresponding author of this paper. Email:huangxia.qd@gmail.com.

Jianping Zhou is a lecturer at the School of Computer, Anhui University of Technology, Ma'anshan 243002, China. His current research interest covers fractional order system, neural networks and applications.