Research on Numerical Simulation of Solving Fractional Calculus Based on one Step Forecast Method

Junwei Lei[1, a], Ruiqi Wang[1], Jing Li[2] and Hui Li[1]

[1]College of Coastal Defense, Naval Aviation University, Yantai, 264001, China

[2]College of Weaponry engineering, Naval Engineering University, Wuhan, 430000, China

[a]leijunwei@126.com

Keywords: Fractional order; Simulation; Stability; Control system; Volter equation

Abstract. By using the forecast method, we use an numerical method to solve the fractional order differential equation with the help of predictive-improved algorithm. We take a function as an example and translate five steps to show the complex algorithm. Especially the complex coefficient is calculate step by step and we found that the fractional order algorithm is a kind of iteration but it is more complex than the Eular method, so the integer order can also use fractional order method to do the simulation.

Introduction

The complex definition of fractional calculus makes the analytical solution of fractional differential equation more difficult and inconvenient to use, so the numerical solution with substitutability flourishes [1-7]. Fractional calculus operator approximation, as the basis of numerical calculation of fractional order systems, has developed a series of related methods, which greatly promotes the development of fractional order system control research [8-13]. Among these algorithms, "z" domain approximation algorithms mainly include A1-Alouni method and short memory principle, while "s" domain approximation algorithms mainly include Carlson method and continued fraction expansion method [14-21]. Because the initial value of fractional derivative defined by Caputo has clear physical meaning, the fractional derivative of Caputo is widely used in engineering. At present, the commonly used method for solving fractional differential equations is predictive-improved algorithm, which was proposed by Diethem in 2002. In this paper, simulation case study is carried out with the tool of MATLAB software M language.

Improved Fractional Order Differential Simulation Method with Forecast

According to the fractional definition, a fractional system can be described as

$$\frac{d^q x}{dt^q} = f(t,x) \tag{1}$$

And its Volter equation can be written as

$$x(t) = \sum_{k=0}^{m-1} x_0^{(k)} \frac{t^k}{k!} + \frac{1}{\Gamma(q)} \int_0^t (t-\tau)^{q-1} f(\tau, x(\tau)) d\tau \tag{2}$$

Where $x^{(k)}(0) = x_0^{(k)}$, $k = 0,1,\cdots,m-1$, $q \in (m-1,m)$.

According to the forecast method, we set simulation step as $h = T/N$, $t_n = nh$, $n = 0,1,2,\cdots,N$

Then

$$x_h(t_{n+1}) = \sum_{k=0}^{m-1} \frac{t_{n+1}^k}{k!} x_0^{(k)} + \frac{h^q}{\Gamma(q+2)} f(t_{n+1}, x_h^p(t_{n+1})) + \frac{h^q}{\Gamma(q+2)} \sum_{i=0}^{n} a_{i,n+1} f(t_i, x_h(t_i)) \tag{3}$$

Where

$$x_h^p(t_{n+1}) = \sum_{k=0}^{m-1} \frac{t_{n+1}^k}{k!} x_0^{(k)} + \frac{1}{\Gamma(q)} \sum_{i=0}^{n} b_{i,n+1} f(t_i, x_h(t_i)) \tag{4}$$

And

$$b_{i,n+1} = \frac{h^q}{q}((n-i+1)^q - (n-i)^q) \tag{5}$$

Its coefficient can be written as

$$a_{i,n+1} = \begin{Bmatrix} n^{q+1} - (n-q)(n+1)^q & i = 0 \\ (n-i+2)^{q+1} + (n-i)^{q+1} - 2(n-i+1)^{q+1} & 0 < i \le n \\ 1 & i = n+1 \end{Bmatrix} \tag{6}$$

Example Analysis

Choose $\frac{d^{1/2}x}{dt^{1/2}} = \sin t$ as an example and set initial value as $x^0(0) = 1$ and choose $m = 1$, then it can be written as

$$x(t) = \sum_{k=0}^{0} 1 + \frac{1}{\Gamma(1/2)} \int_0^t (t-\tau)^{1/2} f(\tau, x(\tau)) d\tau \tag{7}$$

Choose simulation time as T=5s and set $h = T/N = 0.001s$ Then

$$x_h(t_{n+1}) = \sum_{k=0}^{m-1} \frac{t_{n+1}^k}{k!} x_0^{(k)} + \frac{h^q}{\Gamma(q+2)} f(t_{n+1}, x_h^p(t_{n+1})) + \frac{h^q}{\Gamma(q+2)} \sum_{i=0}^{n} a_{i,n+1} f(t_i, x_h(t_i)) \tag{8}$$

Obviously $m = 1, \sum_{k=0}^{m-1} \frac{t_{n+1}^k}{k!} x_0^{(k)} = \frac{t_{n+1}^0}{0!} x_0^{(0)}$, where $x_0^{(0)}$ is the initial value of zero order differential of state x.

And we set $n = 0, 1, 2, \cdots, N$, then

$$q = 1/2, \quad \frac{h^q}{\Gamma(q+2)} \sum_{i=0}^{n} a_{i,n+1} f(t_i, x_h(t_i)) \tag{9}$$

If $n = 0$, then

$$\sum_{i=0}^{n} a_{i,n+1} f(t_i, x_h(t_i)) = \sum_{i=0}^{0} a_{0,1} f(t_i, x_h(t_i)) = a_{0,1} f(t_0, x_h(t_0)) \tag{10}$$

Where $t_0 = 0, x_h(t_0) = x_h(0) = x^0(0) = 1$

And according to

$$a_{i,n+1} = \begin{Bmatrix} n^{q+1} - (n-q)(n+1)^q & i = 0 \\ (n-i+2)^{q+1} + (n-i)^{q+1} - 2(n-i+1)^{q+1} & 0 < i \le n \\ 1 & i = n+1 \end{Bmatrix} \tag{11}$$

Then $a_{0,1} = 0 - (0 - 1/2) = 1/2$, and $t_0 = 0$, and

$$x_h^p(t_{n+1}) = x_h^p(t_{0+1}) = x_h^p(t_1) \tag{12}$$

Then

$$x_h^p(t_{0+1}) = x_h^p(t_1) = \sum_{k=0}^{1-1} \frac{t_{0+1}^0}{0!} x_0^{(0)} + \frac{1}{\Gamma(q)} \sum_{i=0}^{0} b_{i,0+1} f(t_0, x_h(t_0))$$
$$= \sum_{k=0}^{1-1} \frac{t_{0+1}^0}{0!} x_0^{(0)} + \frac{1}{\Gamma(q)} \sum_{i=0}^{0} b_{0,0+1} f(t_0, x_h(t_0)) \tag{13}$$

According to

$$b_{i,n+1} = \frac{h^q}{q}((n-i+1)^q - (n-i)^q) \tag{14}$$

$$b_{0,0+1} = \frac{h^q}{q}((n-0+1)^q - (0-0)^q) = \frac{h^q}{q}((0-0+1)^q - (0-0)^q) \tag{15}$$

Then if $n=1$,

$$x_h(t_{n+1}) = \sum_{k=0}^{m-1} \frac{t_{n+1}^k}{k!} x_0^{(k)} + \frac{h^q}{\Gamma(q+2)} f(t_{n+1}, x_h^p(t_{n+1})) + \frac{h^q}{\Gamma(q+2)} \sum_{i=0}^{n} a_{i,n+1} f(t_i, x_h(t_i)) \tag{16}$$

Where $\sum_{k=0}^{m-1} \frac{t_{n+1}^k}{k!} x_0^{(k)} = x_0^{(0)}$ and

$$\sum_{i=0}^{n} a_{i,n+1} f(t_i, x_h(t_i)) = a_{0,1+1} f(t_0, x_h(t_0)) + a_{1,1+1} f(t_1, x_h(t_1)) \tag{17}$$

Where $t_1 = 1*h$, and according to

$$a_{i,n+1} = \begin{cases} n^{q+1} - (n-q)(n+1)^q & i=0 \\ (n-i+2)^{q+1} + (n-i)^{q+1} - 2(n-i+1)^{q+1} & 0 < i \le n \\ 1 & i = n+1 \end{cases} \tag{18}$$

Then

$a_{0,1+1} = 1^{q+1} - (1-q)(1+1)^q$, $a_{1,1+1} = (1-1+2)^{q+1} + (1-1)^{q+1} - 2(1-1+1)^{q+1}$

And

$$x_h^p(t_{n+1}) = \sum_{k=0}^{m-1} \frac{t_{n+1}^k}{k!} x_0^{(k)} + \frac{1}{\Gamma(q)} \sum_{i=0}^{n} b_{i,n+1} f(t_i, x_h(t_i))$$
$$= x_h^p(t_{1+1}) = \sum_{k=0}^{m-1} \frac{t_{1+1}^k}{k!} x_0^{(0)} + \frac{1}{\Gamma(q)} \sum_{i=0}^{n} b_{i,1+1} f(t_i, x_h(t_i)) \tag{19}$$
$$= x_0^{(0)} + \frac{1}{\Gamma(q)} b_{0,1+1} f(t_0, x_h(t_0)) + \frac{1}{\Gamma(q)} b_{1,1+1} f(t_1, x_h(t_1))$$

Where

$$b_{0,1+1} = \frac{h^q}{q}((1-0+1)^q - (1-0)^q),\ b_{1,1+1} = \frac{h^q}{q}((1-1+1)^q - (1-1)^q) \tag{20}$$

And through above several step of discussion, we found that the predictive method is also a kind of iteration and we should always calculate the forecast value of system state first to solve the final value of fractional order system state.

Conclusion

The above research is mainly aimed at the algorithm of prediction revision, and the 0.5 order differential of sinusoidal function is used for case analysis. Especially, two kinds of coefficients of forecasting revision are analyzed in detail. Through theoretical analysis, it can be seen that the fractional order algorithm is essentially an iterative algorithm, and the above two kinds of coefficients are constant in 1 annex, and the results of Gama function are constant, which are similar to Euler method and Runge-Kutta method, but the simulation algorithm is more complex, so it is easier to lead to system divergence.

References

[1]L. O. Chua. Memristor-the missing circuit element, IEEE Trans. Circuit Theory 18 (5) (1971) 13.

[2]D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams. The missing memristor found, Nature 453 (7191) (2008) 4.

[3]T. He, J.M. Tour. Electronics: The fourth element, Nature 453 (7191) (2008) 2.

[4]M. Di. Ventra, Y.V. Pershin. Experimental demonstration of associative memory with memristive neural networks, Neural Netw. 23 (7) (2010) 6.

[5]G. A. Gibson, S. Musunuru, J. Zhang. An accurate locally active memristor model for s-type negative differential resistance in nbox, Appl. Phys. Lett. 108 (2) (2016) 023505.

[6]G. A. Gibson, S. Musunuru, J. Zhang. Nanocsale memristor device as synapse in neuromorphic systems, Nano Lett. 10 (4) (2010) 5.

[7]K. Norris, B. J. Choi, J. Zhang. Trilayer tunnel selectors for memristor memory cells, Adv. Mater. 28 (2) (2016) 7.

[8]H. Kim, S. P. Adhikari, C. Yang. Memristor bridge synapse-based neural network and its learning, Neural Netw. 23 (9) (2012) 10.

[9]Y. M. Banadaki, M. J. Sharifi. General spice models for memristor and application to circuit simulation of memristor-based synapses and memory cells, J. Circuits Syst. Comput. 19 (2) (2010) 18.

[10]Z. Zeng, A. Wu. Anti-synchronization control of a class of memristive recurrent neural networks, Commun. Nonlinear Sci. Numer. Simul. 18 (2) (2013)13

[11]Guo B. Z., & Zhao, Z. L. (2011). On the convergence of an extended state observer for nonlinear systems with uncertainty. Systems & Control Letters, 60, 420-430.

[12]Han J. Q. (2009). From PID to active disturbance rejection control. IEEE Transactions on Industrial Electronics, 56, 900-906.

[13]Jiang T. T., Huang C. D., & Guo L. (2015). Control of uncertain nonlinear systems based on observers and estimators. Automatic, 59, 35-47.

[14] Khalil H. K. (2002). Nonlinear systems. New Jersey: Prentice Hall.

[15]Levant A. (2003). Higher-order sliding modes, differentiation and output-feedback control. International Journal of Control, 76, 924-941.

[16]Li S. H., Yang, J. Chen, W. H. & Chen X. (2012). Generalized extended state observer based control for systems with mismatched uncertainties. IEEE Transactions on Industrial Electronics, 59, 4792–4802.

[17]Perruquetti W., Floquet T. & Moulay E. (2008). Finite-time observers: application to secure communication. IEEE Transactions on Automatic Control, 53, 356-360.

[18]Praly L. & Jiang Z. P. (2004). Linear output feedback with dynamic high gain for nonlinear systems. Systems & Control Letters, 53, 107-116.

[19]Rosier L. (1992). Homogeneous Lyapunov function for homogeneous continuous vector field. Systems & Control Letters, 19, 467-473.

[20]Sun B., & Gao Z. (2005). A DSP-based active disturbance rejection control design for a 1-kW H-bridge DC–DC power converter. IEEE Transactions on Industrial Electronics, 52, 1271-1277.

[21]Xia Y. Q., & Fu M. Y. (2013). Compound control methodology for flight vehicles. Berlin: Springer-Verlag.