Numerical Solutions of Fractional Differential Equations by Extrapolation

Kamal Pal[1], Fang Liu[2], and Yubin Yan[1](✉)

[1] Department of Mathematics, University of Chester, Thornton Science Park,
Chester CH2 4NU, UK
kamal_pal08@yahoo.co.uk, y.yan@chester.ac.uk
[2] Department of Mathematics, Lvliang University,
Lishi, People's Republic of China
13935870281@163.com

Abstract. An extrapolation algorithm is considered for solving linear fractional differential equations in this paper, which is based on the direct discretization of the fractional differential operator. Numerical results show that the approximate solutions of this numerical method has the expected asymptotic expansions.

1 Introduction

We consider the Richardson extrapolation algorithms for solving the following fractional order differential equation

$$ {}_0^C D_t^{\alpha} y(t) = \beta y(t) + f(t), \quad 0 \le t \le 1, \tag{1}$$

$$ y(0) = y_0, \tag{2}$$

where $\beta < 0$ and f is a given function on $[0, 1]$.

Extrapolation can be used to accelerate the convergence of a given sequence, [1,2,13]. Its applicability depends on the fact that a given sequence of the approximate solutions of the problem possesses an asymptotic expansion. Diethelm [3] introduced an algorithm for solving the above linear differential equation of fractional order, with $0 < \alpha < 1$, Diethelm and Walz [12] proved that the approximate solution of the numerical algorithm in [3] has an asymptotic expansion. See [5–11] for the numerical methods of the general nonlinear fractional differential equations.

Recently, Yan, Pal and Ford [14] extended the numerical method in [3] and obtained a high order numerical method for solving (1) and (2) and proved that the approximate solution has an asymptotic expansion. In this paper, we give some numerical results to show that the approximate solutions of the proposed numerical methods in this paper have the expected asymptotic expansions.

The paper is organized as follows: in Sect. 2, we introduce the numerical method for solving (1) and (2) and discuss how to approximate the starting values and the starting integrals appeared in the numerical method. In Sect. 3, we give some numerical examples to show that the approximate solutions of the proposed numerical methods in this paper have the expected asymptotic expansions.

I. Dimov et al. (Eds.): FDM 2014, LNCS 9045, pp. 299–306, 2015.
DOI: 10.1007/978-3-319-20239-6_32

2 Higher Order Numerical Method

In this section we will consider a higher order numerical method for solving (1) and (2). It is well-known that, when $0 < \alpha < 1$, (1) and (2) is equivalent to, with $0 < \alpha < 1$,

$$ {}_0^R D_t^\alpha [y(t) - y_0] = \beta y(t) + f(t), \quad 0 \le t \le 1, \tag{3} $$

where ${}_0^R D_t^\alpha y(t)$ denotes the Riemann-Liouville fractional derivative defined by, with $0 < \alpha < 1$,

$$ {}_0^R D_t^\alpha y(t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_0^t (t-u)^{-\alpha} y(u)\, d\tau. \tag{4} $$

By using Hadamard finite-part integral, ${}_0^R D_t^\alpha$ can be written into

$$ {}_0^R D_t^\alpha y(t) = \frac{1}{\Gamma(-\alpha)} \oint_0^t (t-u)^{-1-\alpha} y(u)\, du. \tag{5} $$

Here the integral $\oint$ denotes a Hadamard finite-part integral [3].

Yan, Pal and Ford [14] extended the numerical method in Diethelm and Walz [12] and obtained a high order numerical method for solving (1) and (2) for $0 < \alpha < 1$. Let M be a fixed positive integer and let $0 = t_0 < t_1 < t_2 < \cdots < t_{2j} < t_{2j+1} < \cdots < t_{2M} = 1$ be a partition of $[0,1]$ and h the step size. At the nodes $t_{2j} = \frac{2j}{2M}$, the Eqs. (1) and (2) satisfy

$$ {}_0^R D_t^\alpha [y(t_{2j}) - y_0] = \beta y(t_{2j}) + f(t_{2j}), \quad j = 1, 2, \ldots, M, $$

and at the nodes $t_{2j+1} = \frac{2j+1}{2M}$, the Eqs. (1) and (2) satisfy

$$ {}_0^R D_t^\alpha [y(t_{2j+1}) - y_0] = \beta y(t_{2j+1}) + f(t_{2j+1}), \quad j = 0, 1, 2, \ldots, M-1. \tag{6} $$

Note that

$$ {}_0^R D_t^\alpha y(t_{2j}) = \frac{1}{\Gamma(-\alpha)} \oint_0^{t_{2j}} (t_{2j} - \tau)^{-1-\alpha} y(\tau)\, d\tau = \frac{t_{2j}^{-\alpha}}{\Gamma(-\alpha)} \oint_0^1 w^{-1-\alpha} y(t_{2j} - t_{2j} w)\, dw. \tag{7} $$

For every j, we denote $g(w) = y(t_{2j} - t_{2j} w)$ and approximate the integral $\oint_0^1 w^{-1-\alpha} g(w)\, dw$ by $\oint_0^1 w^{-1-\alpha} g_2(w)\, dw$, where $g_2(w)$ is the piecewise quadratic interpolation polynomials on the nodes $w_l = l/2j$, $l = 0, 1, 2, \ldots, 2j$. More precisely, we have, for $k = 1, 2, \ldots, j$,

$$ \begin{aligned} g_2(w) = {} & \frac{(w - w_{2k-1})(w - w_{2k})}{(w_{2k-2} - w_{2k-1})(w_{2k-2} - w_{2k})} g(w_{2k-2}) \\ & + \frac{(w - w_{2k-2})(w - w_{2k})}{(w_{2k-1} - w_{2k-2})(w_{2k-1} - w_{2k})} g(w_{2k-1}) \\ & + \frac{(w - w_{2k-2})(w - w_{2k-1})}{(w_{2k} - w_{2k-2})(w_{2k} - w_{2k-1})} g(w_{2k}), \quad \text{for } w \in [w_{2k-2}, w_{2k}]. \end{aligned} $$

Thus

$$\begin{aligned}
{}_0^R D_t^{\alpha} y(t_{2j}) &= \frac{t_{2j}^{-\alpha}}{\Gamma(-\alpha)} \oint_0^1 w^{-1-\alpha} y(t_{2j} - t_{2j}w)\, dw \\
&= \frac{t_{2j}^{-\alpha}}{\Gamma(-\alpha)} \Big(\sum_{k=1}^{j} \oint_{w_{2k-2}}^{w_{2k}} w^{-1-\alpha} g_2(w)\, dw + R_{2j}(g) \Big) \\
&= \frac{t_{2j}^{-\alpha}}{\Gamma(-\alpha)} \Big(\sum_{k=0}^{2j} \alpha_{k,2j} y(t_{2j-k}) + R_{2j}(g) \Big)
\end{aligned}$$

where $R_{2j}(g)$ is the remainder term and $\alpha_{k,2j}, k = 0, 1, 2, \ldots, 2j$ are weights given by

$$\begin{aligned}
&(-\alpha)(-\alpha+1)(-\alpha+2)(2j)^{-\alpha}\alpha_{l,2j} \\
&= \begin{cases}
2^{-\alpha}(\alpha+2), & \text{for } l = 0, \\
(-\alpha)2^{2-\alpha}, & \text{for } l = 1, \\
(-\alpha)(-2^{-\alpha}\alpha) + \frac{1}{2}F_0(2), & \text{for } l = 2, \\
-F_1(k), & \text{for } l = 2k-1, \quad k = 2, 3, \ldots, j, \\
\frac{1}{2}(F_2(k) + F_0(k+1)), & \text{for } l = 2k, \quad k = 2, 3, \ldots, j-1, \\
\frac{1}{2}F_2(j), & \text{for } l = 2j.
\end{cases}
\end{aligned}$$

Here

$$\begin{aligned}
F_0(k) =&(2k-1)(2k)\Big((2k)^{-\alpha} - (2k-2)^{-\alpha}\Big)(-\alpha+1)(-\alpha+2) \\
&- \Big((2k-1) + 2k\Big)\Big((2k)^{-\alpha+1} - (2k-2)^{-\alpha+1}\Big)(-\alpha)(-\alpha+2) \\
&+ \Big((2k)^{-\alpha+2} - (2k-2)^{-\alpha+2}\Big)(-\alpha)(-\alpha+1),
\end{aligned}$$

$$\begin{aligned}
F_1(k) =&(2k-2)(2k)\Big((2k)^{-\alpha} - (2k-2)^{-\alpha}\Big)(-\alpha+1)(-\alpha+2) \\
&- \Big((2k-2) + 2k\Big)\Big((2k)^{-\alpha+1} - (2k-2)^{-\alpha+1}\Big)(-\alpha)(-\alpha+2) \\
&+ \Big((2k)^{-\alpha+2} - (2k-2)^{-\alpha+2}\Big)(-\alpha)(-\alpha+1),
\end{aligned}$$

and

$$\begin{aligned}
F_2(k) =&(2k-2)(2k-1)\Big((2k)^{-\alpha} - (2k-2)^{-\alpha}\Big)(-\alpha+1)(-\alpha+2) \\
&- \Big((2k-2) + (2k-1)\Big)\Big((2k)^{-\alpha+1} - (2k-2)^{-\alpha+1}\Big)(-\alpha)(-\alpha+2) \\
&+ \Big((2k)^{-\alpha+2} - (2k-2)^{-\alpha+2}\Big)(-\alpha)(-\alpha+1).
\end{aligned}$$

Hence (3) satisfies for $j = 1, 2, \ldots, M$,

$$y(t_{2j}) = \frac{1}{\alpha_{0,2j} - t_{2j}^{\alpha}\Gamma(-\alpha)\beta} \Big[t_{2j}\Gamma(-\alpha)f(t_{2j}) - \sum_{k=1}^{2j} \alpha_{k,2j} y(t_{2j-k}) + y_0 \sum_{k=0}^{2j} \alpha_{k,2j} - R_{2j}(g) \Big]. \quad (8)$$

At the nodes $t_{2j+1} = \frac{2j+1}{2M},\ j = 0, 1, 2, \ldots, M-1$, we have

$$\begin{aligned}
{}_0^R D_t^{\alpha} y(t_{2j+1}) &= \frac{1}{\Gamma(-\alpha)} \oint_0^{t_{2j+1}} (t_{2j+1} - \tau)^{-1-\alpha} y(\tau)\, d\tau \\
&= \frac{1}{\Gamma(-\alpha)} \oint_0^{t_1} (t_{2j+1} - \tau)^{-1-\alpha} y(\tau)\, d\tau \\
&\quad + \frac{t_{2j+1}^{-\alpha}}{\Gamma(-\alpha)} \oint_0^{\frac{2j}{2j+1}} w^{-1-\alpha} y(t_{2j+1} - t_{2j+1} w)\, dw.
\end{aligned}$$

For every j, we denote $g(w) = y(t_{2j+1} - t_{2j+1}w)$ and approximate the integral $\oint_0^{\frac{2j}{2j+1}} w^{-1-\alpha} g(w)\, dw$ by $\oint_0^{\frac{2j}{2j+1}} w^{-1-\alpha} g_2(w)\, dw$, where $g_2(w)$ is the piecewise quadratic interpolation polynomials on the nodes $w_l = \frac{l}{2j+1},\ l = 0, 1, 2, \ldots, 2j$. We then get

$$\begin{aligned}
{}_0^R D_t^{\alpha} y(t_{2j+1}) &= \frac{1}{\Gamma(-\alpha)} \int_0^{t_1} (t_{2j+1} - \tau)^{-1-\alpha} y(\tau)\, d\tau \\
&\quad + \frac{t_{2j+1}^{-\alpha}}{\Gamma(-\alpha)} \Big(\sum_{k=1}^{j} \oint_{w_{2k-2}}^{w_{2k}} w^{-1-\alpha} g_2(w)\, dw + R_{2j+1}(g) \Big) \\
&= \frac{1}{\Gamma(-\alpha)} \int_0^{t_1} (t_{2j+1} - \tau)^{-1-\alpha} y(\tau)\, d\tau \\
&\quad + \frac{t_{2j+1}^{-\alpha}}{\Gamma(-\alpha)} \Big(\sum_{k=0}^{2j} \alpha_{k,2j+1} y(t_{2j+1-k}) + R_{2j+1}(g) \Big)
\end{aligned}$$

where $R_{2j+1}(g)$ is the remainder term and $\alpha_{k,2j+1} = \alpha_{k,2j},\ k = 0, 1, 2, \ldots, 2j$. Hence

$$\begin{aligned}
y(t_{2j+1}) = \frac{1}{\alpha_{0,2j+1} - t_{2j+1}^{\alpha}\Gamma(-\alpha)\beta} \Big[& t_{2j+1}^{\alpha}\Gamma(-\alpha) f(t_{2j+1}) - \sum_{k=1}^{2j} \alpha_{k,2j+1} y(t_{2j+1-k}) \\
& + y_0 \sum_{k=0}^{2j} \alpha_{k,2j+1} - R_{2j+1}(g) - t_{2j+1}^{\alpha} \int_0^{t_1} (t_{2j+1} - \tau)^{-1-\alpha} y(\tau)\, d\tau \Big]. \qquad (9)
\end{aligned}$$

Here $\alpha_{0,l} - t_l^{\alpha}\Gamma(-\alpha)\beta < 0,\ l = 2j, 2j+1$, which follow from $\Gamma(-\alpha) < 0$, $\beta < 0$ and $\alpha_{0,2j+1} = \alpha_{0,2j} < 0$.

Let $y_{2j} \approx y(t_{2j})$ and $y_{2j+1} \approx y(t_{2j+1})$ denote the approximate solutions of $y(t_{2j})$ and $y(t_{2j+1})$, respectively. We define the following numerical methods for solving (1) and (2), with $j = 1, 2, \ldots, M$,

$$y_{2j} = \frac{1}{\alpha_{0,2j} - t_{2j}^{\alpha}\Gamma(-\alpha)\beta} \Big[t_{2j}\Gamma(-\alpha) f(t_{2j}) - \sum_{k=1}^{2j} \alpha_{k,2j} y_{2j-k} + y_0 \sum_{k=0}^{2j} \alpha_{k,2j} \Big], \quad (10)$$

and, with $j = 1, 2, \ldots, M-1$,

$$y_{2j+1} = \frac{1}{\alpha_{0,2j+1} - t_{2j+1}^{\alpha}\Gamma(-\alpha)\beta}\Big[t_{2j+1}^{\alpha}\Gamma(-\alpha)f(t_{2j+1}) - \sum_{k=1}^{2j}\alpha_{k,2j+1}y_{2j+1-k} \\ + y_0\sum_{k=0}^{2j}\alpha_{k,2j+1} - t_{2j+1}^{\alpha}\int_0^{t_1}(t_{2j+1}-\tau)^{-1-\alpha}y(\tau)\,d\tau\Big]. \tag{11}$$

Yan, Pal and Ford [14] proved the following Theorem.

Theorem 1 (Theorem 2.1 in [14]). *Let $0 < \alpha < 1$ and M be a positive integer. Let $0 = t_0 < t_1 < t_2 < \cdots < t_{2j} < t_{2j+1} < \cdots < t_{2M} = 1$ be a partition of $[0, 1]$ and h the step size. Let $y(t_{2j}), y(t_{2j+1}), y_{2j}$ and y_{2j+1} be the exact and the approximate solutions of (8)–(11), respectively. Assume that $y \in C^{m+2}[0, 1]$, $m \geq 3$. Further assume that we can approximate the starting value y_1 and the starting integral $\int_0^{t_1}(t_{2j+1}-\tau)^{-1-\alpha}y(\tau)\,d\tau$ in (11) by using some numerical methods and obtain the required accuracy. Then there exist coefficients $c_\mu = c_\mu(\alpha)$ and $c_\mu^* = c_\mu^*(\alpha)$ such that the sequence $\{y_l\}, l = 0, 1, 2, \ldots, 2M$ possesses an asymptotic expansion of the form*

$$y(t_{2M}) - y_{2M} = \sum_{\mu=3}^{m+1} c_\mu(2M)^{\alpha-\mu} + \sum_{\mu=2}^{\mu^*} c_\mu^*(2M)^{-2\mu} + o((2M)^{\alpha-m-1}), \quad \textit{for } M \to \infty,$$

that is,

$$y(t_{2M}) - y_{2M} = \sum_{\mu=3}^{m+1} c_\mu h^{\mu-\alpha} + \sum_{\mu=2}^{\mu^*} c_\mu^* h^{2\mu} + o(h^{m+1-\alpha}), \quad \textit{for } h \to 0,$$

where μ^* is the integer satisfying $2\mu^* < m + 1 - \alpha < 2(\mu^* + 1)$, and c_μ and c_μ^* are certain coefficients that depend on y.

3 Numerical Simulations

Example 1. Consider the following example in [9], with $0 < \alpha < 1$,

$$ {}_0^C D_t^{\alpha} y(t) + y(t) = 0, \quad t \in [0, 1], \tag{12}$$

$$ y(0) = 1. \tag{13}$$

It is well known that the exact solution is

$$y(t) = E_\alpha(-t^\alpha),$$

where

$$E_\alpha(z) = \sum_{k=0}^{\infty}\frac{z^k}{\Gamma(\alpha k + 1)}$$

is the Mittag-Leffler function of order α. Here the given function f is smooth and $f = 0$ (Table 1).

Table 1. Errors for Eqs. (12) and (13) with $\alpha = 0.3$, taken at $t = 1$.

Step size	Error of the method	1st extra. error	2nd extra. error
1/10	1.1296e-003		
1/20	2.0412e-004	5.3779e-006	
1/40	3.6454e-005	4.5025e-007	2.7527e-008
1/80	6.4759e-006	3.8539e-008	1.3800e-009
1/160	1.1475e-006	3.3431e-009	6.9354e-011
1/320	2.0310e-007	2.9225e-010	3.5604e-012

Table 2. Orders (“EOC ”) for Eqs. (12) and (13) with $\alpha = 0.3$, taken at $t = 1$.

Step size	The method	1st extrapolation	2nd extrapolation
1/10			
1/20	2.46		
1/40	2.49	3.58	
1/80	2.49	3.55	4.32
1/160	2.5	3.53	4.31
1/320	2.5	3.52	4.28

Choose the step size $h = 1/10$. In Table 2, we displayed the errors of the algorithms (10) and (11) at $t = 1$ and of the first two extrapolation steps in the Romberg tableau with $\alpha = 0.3$. We observe that the first column (the errors of the basic algorithm without extrapolation) converges as $h^{3-\alpha}$. The second column (errors using one extrapolation step)converges as $h^{4-\alpha}$, and the last column (two extrapolation steps) converges as h^4. We also consider other values of $\alpha \in (0, 1)$. We observe that when α is close to 1, the convergence seems to be even a bit faster. But when α is close to 0, the convergence is a bit slower than expected which is consistent with the numerical observation in [12] for the lower order method.

Example 2. Consider the following example in [4], with $0 < \alpha < 1$,

$$ {}_0^C D_t^\alpha y(t) + y(t) = t^4 - \frac{1}{2}t^3 - \frac{3}{\Gamma(4-\alpha)}t^{3-\alpha} + \frac{24}{\Gamma(5-\alpha)}t^{4-\alpha}, \quad t \in [0, 1], \tag{14} $$

$$ y(0) = 0, \tag{15} $$

whose exact solution is given by $y(t) = t^4 - \frac{1}{2}t^3$.

Choose the step size $h = 1/10$. In Table 3, we displayed the errors of the algorithms (10) and (11) at $t = 1$ and of the first two extrapolation steps in the Romberg tableau with $\alpha = 0.3$. We observe that the first column converges as $h^{3-\alpha}$. The second column converges as $h^{4-\alpha}$ and the last column converges as h^4. We also consider other values of $\alpha \in (0, 1)$. We observe that when α is close

Table 3. Errors for Eqs. (12)–(15) with $\alpha = 0.3$, taken at $t = 1$.

Step size	Error of the method	1st extra. error	2nd extra. error
1/10	1.4571e-004		
1/20	2.3118e-005	8.2097e-007	
1/40	3.6127e-006	6.5021e-008	2.0039e-009
1/80	5.6030e-007	5.1186e-009	1.2514e-010
1/160	8.6565e-008	4.0106e-010	7.8051e-012
1/320	1.3348e-008	3.1315e-011	4.9268e-013

Table 4. Orders ("EOC") for Eqs. (12)–(15) with $\alpha = 0.3$, taken at $t = 1$.

Step size	The method	1st extrapolation	2nd extrapolation
1/10			
1/20	2.66		
1/40	2.68	3.66	
1/80	2.69	3.67	4.00
1/160	2.70	3.67	4.00
1/320	2.70	3.68	3.98

to 1, the convergence seems to be even a bit faster. But when α is close to 0, the convergence is a bit slower than expected which is consistent with the numerical observation in [12] for the lower order method (Table 4).

Acknowledgements. We wish to express our sincere gratitude to Professor Neville. J. Ford for his encouragement, discussions and valuable criticism during the research of this work.

References

1. Brezinski, C.: A general extrapolation algorithm. Numer. Math. **35**, 175–187 (1980)
2. Brezinski, C., Redivo Zaglia, M.: Extrapolation Methods, Theory and Practice. North Holland, Amsterdam (1992)
3. Diethelm, K.: Generalized compound quadrature formulae for finite-part integrals. IMA J. Numer. Anal. **17**, 479–493 (1997)
4. Diethelm, K.: An algorithm for the numerical solution of differential equations of fractional order. Electron. Trans. Numer. Anal. **5**, 1–6 (1997)
5. Diethelm, K.: The Analysis of Fractional Differential Equations: An Application-Oriented Using Differential Operators of Caputo Type. Lecture Notes in Mathematics 2004. Springer, Heidelberg (2010)
6. Diethelm, K.: Monotonicity results for a compound quadrature method for finite-part integrals. J. Inequalities Pure Appl. Math. **5**(2) (2004). (Article 44)

7. Diethelm, K., Ford, N.J.: Analysis of fractional differential equations. J. Math. Anal. Appl. **265**, 229–248 (2002)
8. Diethelm, K., Ford, N.J., Freed, A.D.: Detailed error analysis for a fractional Adams method. Numer. Algorithms **36**, 31–52 (2004)
9. Diethelm, K., Ford, N.J., Freed, A.D.: A predictor-corrector approach for the numerical solution of fractional differential equations. Nonlinear Dynam. **29**, 3–22 (2002)
10. Diethelm, K., Luchko, Y.: Numerical solution of linear multi-term initial value problems of fractional order. J. Comput. Anal. Appl. **6**, 243–263 (2004)
11. Dimitrov, Y.: Numerical approximations for fractional differential equations. J. Frac. Calc. Appl. **5**(3S), 1–45 (2014)
12. Diethelm, K., Walz, G.: Numerical solution of fractional order differential equations by extrapolation. Numer. Algorithms **16**, 231–253 (1997)
13. Walz, G.: Asymptotics and Extrapolation. Akademie-Verlag, Berlin (1996)
14. Yan, Y., Pal, K., Ford, N.J.: Higher order numerical methods for solving fractional differential equations. BIT Numer. Math. **54**, 555–584 (2014). doi:10.1007/s10543-013-0443-3