Numerical Algorithm based on Fast Convolution for Fractional Calculus

by

An Chen, Peng Guo, Changpin Li*

Department of Mathematics, Shanghai University, Shanghai 200444, PR China

lcp@shu.edu.cn (C. P. Li)

In this paper, numerical algorithms based on fast convolution for the fractional integral and fractional derivative are proposed. Two examples are also included which show the efficiency of the derived method.

Key words: *numerical approach, fractional calculus, fast convolution, Riemann-Liouville derivative, Caputo derivative*

1. Introduction

Fast convolution for solving convolution quadrature was proposed several years ago [1–3]. It is known that the fractional integral and the fractional derivative are defined in a way of convolution [4]. Thus, it makes sense to design an algorithm for fractional calculus by using the fast convolution method. The derived algorithm in this paper requires $O(N\log N)$ operations and $O(\log N)$ active memory in place of $O(N^2)$ operations and $O(N)$ active memory given in [5], [8-9] where N denotes the total step number.

The definitions of fractional integral and fractional derivative are introduced below.

Definition1. *The fractional integral (or, Riemann-Liouville integral) of function* y *is defined by*

$$J_{0,t}^{\alpha} f(x) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} y(\tau) d\tau. \tag{1}$$

Definition2. *The Caputo derivative of* y, *for* $t>0$, $m-1<\alpha<m\in Z^{+}$, *is defined as*

$${}_{C}D_{0,t}^{\alpha} f(x) = \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-\tau)^{m-\alpha-1} y^{(m)}(\tau) d\tau. \tag{2}$$

The outline of the present paper is organized as follows. The numerical method of fractional calculus is introduced in Section 2. The numerical examples are given in Section 3 which show the efficiency of the constructed method in Section 2.

2. The fast convolution for fractional calculus

Firstly, we introduce the numerical inversion of the Laplace transform of a kernel $f(t)=\frac{t^{\alpha-1}}{\Gamma(\alpha)}$.

Simple calculation implies

$$F(s)=L\{f(t);s\}=\int_o^{\infty}e^{-st}f(t)dt=s^{-\alpha},\quad s\in\mathbb{C}. \tag{3}$$

So the inversion of $F(s)$ gives,

$$f(t)=\frac{1}{2\pi i}\int_{\Gamma}e^{st}F(s)ds, \tag{4}$$

where Γ is a counter-clockwise integral contour in the sector of analyticity, which goes to infinity with an acute angle to the negative real half-axis and is oriented with increasing imaginary part. Thus it ensures the absolute convergence of the integral appearing in (4).

For a suitable contour Γ and $t\in[t_0,\Lambda t_0]$, where $\Lambda>1$ is a positive scale parameter, the approximation algorithm of inverse Laplace transform by discretizing the contour integral can be used.

● In general, the contour Γ is a suitably chosen hyperbola which lies in the left branch, and is parameterized as follows:

$$\mathbb{R}\to\Gamma: x\mapsto s=T(x;\lambda,a):=\lambda(1-\sin(a+ix)), \tag{5}$$

where $\lambda>0$ is a scale parameter and $a>0$, [2], [7]. Fig. 1 is for $a=\pi/2-1/2$. We notice that the mapping T takes the negative x to the complex point where the real part is less than 0 and the imaginary part is larger than 0.

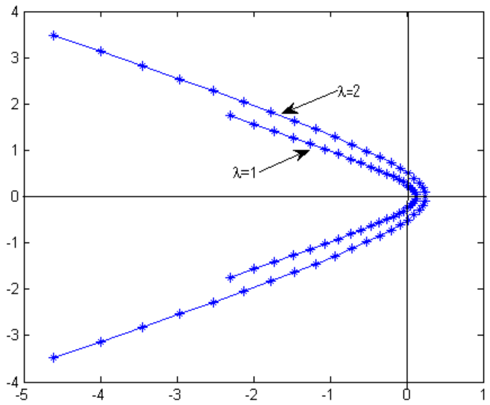

Figure 1: **Hyperbola** $(T([-2:0:0.1:2];\lambda,\pi/2-1/2)=\lambda(1-\sin(a+ix)))$

● Then the truncated trapezoidal rule can be applied to approximate the kernel function (4) according to [7], that is

$$
\begin{aligned}
f(t) &= \frac{1}{2\pi i}\int_{\Gamma} e^{st}s^{-\alpha}ds \\
&= \frac{1}{2\pi i}\int_{+\infty}^{-\infty} e^{T(x;\lambda,a)t}T(x;\lambda,a)^{-\alpha}T'(x;\lambda,a)dx \\
&\approx -\frac{\tau}{2\pi i}\sum_{k=-n}^{n} e^{T(k\tau;\lambda,a)t}T(k\tau;\lambda,a)^{-\alpha}T'(k\tau;\lambda,a) \\
&= \sum_{k=-n}^{n} w_k e^{s_k t}s_k{}^{-\alpha},
\end{aligned}
\tag{6}
$$

where weights w_k and quadrature nodes s_k are given by

$$
w_k = -\frac{\tau}{2\pi i}T'(k\tau;\lambda,a),\quad s_k = T(k\tau;\lambda,a), \tag{7}
$$

with step length $\tau > 0$.

In [6], the different choice of Γ and parameterizations were discussed in order to get a better convergence result. Due to this, we take

$$
t_0 = 1,\ \ a = 0.7,\ \ \Lambda = 20,\ \ h = \frac{1}{n}\operatorname{arccosh}\left(\frac{\Lambda}{0.5\sin(a)}\right),\ \ \lambda = \frac{0.6\pi n}{t_0\Lambda}\frac{1}{\operatorname{arccosh}\left(\frac{\Lambda}{0.5\sin(a)}\right)}.
$$

Figs. 2 and 3 are drawn for describing the convergence of numerical inversion for the Laplace transform of $f(t)$.

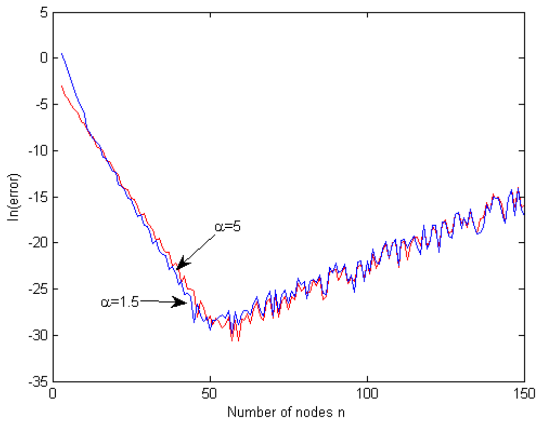

Figure 2: **The error of** $f(t)$ **for two different** α **'s (when** $n = 1, 2, \ldots, 150.$**)**

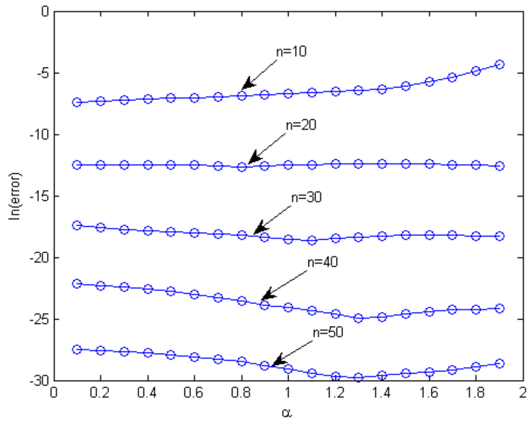

Figure 3: **The error of** $f(t)$ **for five different** n**'s (when** $\alpha \in (0,2)$**)**

From Fig. 2, it appears that the numerical error is very small which can be negligible when $0 < \alpha < 2$. And from Fig. 3, it shows the importance of the choice of n.

Next, we derive the numerical algorithm for fractional calculus.

For a given t to the fractional integral (1), the interval $[0,t]$ can be divided into $[0,t-\Delta t]$ and $[t-\Delta t,t]$ where $\Delta t = t/N$ with step number N. That is,

$$
\begin{aligned}
J_{0,t}^{\alpha} y(t) &= \frac{1}{\Gamma(\alpha)}\int_0^t (t-\tau)^{\alpha-1} y(\tau)d\tau \\
&= \int_0^{t-\Delta t} \frac{(t-\tau)^{\alpha-1}}{\Gamma(\alpha)} y(\tau)d\tau + \int_{t-\Delta t}^{t} \frac{(t-\tau)^{\alpha-1}}{\Gamma(\alpha)} y(\tau)d\tau \\
&= A_1 + A_2.
\end{aligned} \tag{8}
$$

- Firstly, we numerically seek A_1. Because the integral interval of I_1 is $[0,t-\Delta t]$, the range of τ in kernel function $(t-\tau)^{\alpha-1}/\Gamma(\alpha)$ is $[\Delta t,t]$. Combined with the fast convolution [1], a sequence of fast-growing time interval I_l can be applied to cover $[\Delta t,t]$:

$$
I_l = [B^{l-1}\Delta t, (2B^l - 1)\Delta t], \tag{9}
$$

where $B > 1$ is an integer. Let the total step number L be the smallest integer satisfying $t < 2B^L \Delta t$. For $l = 1,2,...,L-1$, determine the integer $p_l \geq 1$ such that $\tau_l = p_l B^l \Delta t$ satisfies $t-\tau_l \in [B^l\Delta t,(2B^l-1)\Delta t]$.

So $0 = \tau_L < \tau_{L-1} < ... < \tau_1 < \tau_0 = t-\Delta t$. When $B = 2$, the following of τ_l is obtained, also see Fig. 4.

1. $t = \Delta t : [0, \Delta t],$
2. $t = 2\Delta t : \underbrace{[0, \Delta t]}_{I_1} \bigcup [\Delta t, 2\Delta t],$
3. $t = 3\Delta t : \underbrace{[0, 2\Delta t]}_{I_1} \bigcup [2\Delta t, 3\Delta t],$
4. $t = 4\Delta t : \underbrace{[0, 2\Delta t]}_{I_2} \bigcup \underbrace{[2\Delta t, 3\Delta t]}_{I_1} \bigcup [3\Delta t, 4\Delta t],$
5. $t = 5\Delta t : \underbrace{[0, 2\Delta t]}_{I_2} \bigcup \underbrace{[2\Delta t, 4\Delta t]}_{I_1} \bigcup [4\Delta t, 5\Delta t],$
6. $t = 6\Delta t : \underbrace{[0, 4\Delta t]}_{I_2} \bigcup \underbrace{[4\Delta t, 5\Delta t]}_{I_1} \bigcup [5\Delta t, 6\Delta t],$
7. $t = 7\Delta t : \underbrace{[0, 4\Delta t]}_{I_2} \bigcup \underbrace{[4\Delta t, 6\Delta t]}_{I_1} \bigcup [6\Delta t, 7\Delta t],$
8. $t = 8\Delta t : \underbrace{[0, 4\Delta t]}_{I_3} \bigcup \underbrace{[4\Delta t, 6\Delta t]}_{I_2} \bigcup \underbrace{[6\Delta t, 7\Delta t]}_{I_1} \bigcup [7\Delta t, 8\Delta t],$
9. $t = 9\Delta t : \underbrace{[0, 4\Delta t]}_{I_3} \bigcup \underbrace{[4\Delta t, 6\Delta t]}_{I_2} \bigcup \underbrace{[6\Delta t, 8\Delta t]}_{I_1} \bigcup [8\Delta t, 9\Delta t],$
10. $t = 10\Delta t : \underbrace{[0, 4\Delta t]}_{I_3} \bigcup \underbrace{[4\Delta t, 8\Delta t]}_{I_2} \bigcup \underbrace{[8\Delta t, 9\Delta t]}_{I_1} \bigcup [9\Delta t, 10\Delta t],$
11. $t = 11\Delta t : \underbrace{[0, 4\Delta t]}_{I_3} \bigcup \underbrace{[4\Delta t, 8\Delta t]}_{I_2} \bigcup \underbrace{[8\Delta t, 10\Delta t]}_{I_1} \bigcup [10\Delta t, 11\Delta t],$
12. $t = 12\Delta t : \underbrace{[0, 8\Delta t]}_{I_3} \bigcup \underbrace{[8\Delta t, 10\Delta t]}_{I_2} \bigcup \underbrace{[10\Delta t, 11\Delta t]}_{I_1} \bigcup [11\Delta t, 12\Delta t],$
13. $t = 13\Delta t : \underbrace{[0, 8\Delta t]}_{I_3} \bigcup \underbrace{[8\Delta t, 10\Delta t]}_{I_2} \bigcup \underbrace{[10\Delta t, 12\Delta t]}_{I_1} \bigcup [12\Delta t, 13\Delta t],$
14. $t = 14\Delta t : \underbrace{[0, 8\Delta t]}_{I_3} \bigcup \underbrace{[8\Delta t, 12\Delta t]}_{I_2} \bigcup \underbrace{[12\Delta t, 13\Delta t]}_{I_1} \bigcup [13\Delta t, 14\Delta t],$
15. $t = 15\Delta t : \underbrace{[0, 8\Delta t]}_{I_3} \bigcup \underbrace{[8\Delta t, 12\Delta t]}_{I_2} \bigcup \underbrace{[12\Delta t, 14\Delta t]}_{I_1} \bigcup [14\Delta t, 15\Delta t],$
16. $t = 16\Delta t : \underbrace{[0, 8\Delta t]}_{I_4} \bigcup \underbrace{[8\Delta t, 12\Delta t]}_{I_3} \bigcup \underbrace{[12\Delta t, 14\Delta t]}_{I_2} \bigcup \underbrace{[14\Delta t, 15\Delta t]}_{I_1} [15\Delta t, 16\Delta t],$

$\vdots$

and so on.

For $I_l (l = 1, 2, ..., L-1)$, the approximation of $f(t)$ on it results from applying the trapezoidal rule to a parametrization of the contour integral for the inverse Laplace transform as discussed above (6). That is,

$$f(t) = \frac{1}{2\pi i} \int_{\Gamma_l} e^{st} s^{-\alpha} ds \approx \sum_{k=-n}^{n} w_k^{(l)} e^{s_k^{(l)} t} s_k^{(l)-\alpha}, t \in I_l, \tag{10}$$

with a suitably chosen complex contour Γ_l describing in [6].

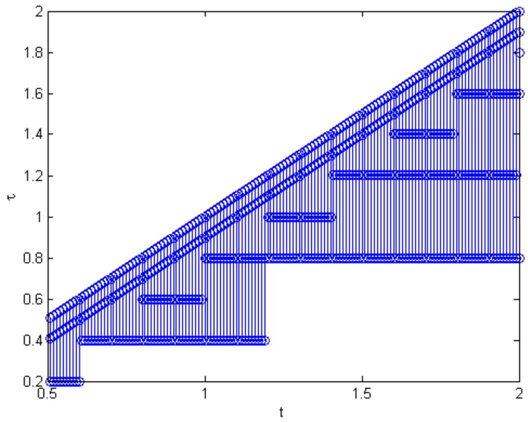

Figure 4: **Tessellation for** $B=2$

So,

$$\begin{aligned}
\int_0^{t-\Delta t} \frac{(t-\tau)^{\alpha-1}}{\Gamma(\alpha)} y(\tau)d\tau &= \sum_{l=1}^{L} \int_{\tau_l}^{\tau_{l-1}} \frac{(t-\tau)^{\alpha-1}}{\Gamma(\alpha)} y(\tau)d\tau \\
&= \sum_{l=1}^{L} \int_{\tau_l}^{\tau_{l-1}} w_k^{(l)} s_k^{(l)-\alpha} e^{(t-\tau)s_k^{(l)}} y(\tau)d\tau \\
&= \sum_{l=1}^{L} \sum_{k=-n}^{n} w_k^{(l)} s_k^{(l)-\alpha} e^{(t-\tau_{l-1})s_k^{(l)}} \int_{\tau_l}^{\tau_{l-1}} e^{(\tau_{l-1}-\tau)s_k^{(l)}} y(\tau)d\tau \\
&= \sum_{l=1}^{L} \sum_{k=-n}^{n} w_k^{(l)} s_k^{(l)-\alpha} e^{(t-\tau_{l-1})s_k^{(l)}} Y(\tau_l, \tau_{l-1}, s_k^{(l)}),
\end{aligned} \tag{11}$$

where $w_k^{(l)}$ and $s_k^{(l)}$ denote the weights and quadrature nodes in the corresponding contour Γ_l of $[\tau_l, \tau_{l-1}]$ and $Y(\tau_l, \tau_{l-1}, s_k^{(l)})$ denotes the solution at τ_{l-1} to the following linear inhomogeneous ordinary differential equation:

$$\begin{cases} Y'(t) = s_k^{(l)} Y(t) + y(t), \\ Y(\tau_l) = 0, \end{cases} \tag{12}$$

where $k=-n,-n+1,\ldots,n$, and $l=1,2,\ldots,L-1$.

- Then, the approximation of A_2 is discussed. Approximate $y(\tau)$ by linear interpolation:

$$S(\tau) = -\frac{\tau-\tau_{i+1}}{\Delta t} y(\tau_i) + \frac{\tau-\tau_i}{\Delta t} y(\tau_{i+1}), \tau_i < \tau < \tau_{i+1}. \tag{13}$$

Here $\tau_i = t-\Delta t$ and $\tau_{i+1} = t$. So,

$$
\begin{aligned}
A_2 &\approx \int_{t-\Delta t}^{t} \frac{(t-\tau)^{\alpha-1}}{\Gamma(\alpha)} S(\tau) d\tau \\
&= \frac{\Delta t^{\alpha}}{\Gamma(\alpha)} \left(\frac{y(t-\Delta t)}{1+\alpha} + \frac{y(t)}{\alpha+\alpha^2} \right).
\end{aligned}
\tag{14}
$$

Finally, by combining the approximation of (11) with (14), the algorithm for fractional integral is completed, that is,

$$
J_{0,t}^{\alpha} y(t) \approx \sum_{l=1}^{L} \sum_{k=-n}^{n} w_k^{(l)} s_k^{(l)-\alpha} e^{(t-\tau_{l-1}) s_k^{(l)}} Y(\tau_l, \tau_{l-1}, s_k^{(l)}) + \frac{\Delta t^{\alpha}}{\Gamma(\alpha)} \left(\frac{y(t-\Delta t)}{1+\alpha} + \frac{y(t)}{\alpha+\alpha^2} \right). \tag{15}
$$

Similarly, the approximation of (2) is given as follows,

$$
\begin{aligned}
D_{0,t}^{\alpha} y(t) \approx & \sum_{l=1}^{L} \sum_{k=-n}^{n} w_k^{(l)} s_k^{(l)-(m-\alpha)} e^{(t-\tau_{l-1}) s_k^{(l)}} Y_2(\tau_l, \tau_{l-1}, s_k^{(l)}) \\
& + \frac{\Delta t^{(m-\alpha)}}{\Gamma(m-\alpha)} \left(\frac{y^{(m)}(t-\Delta t)}{1+m-\alpha} + \frac{y^{(m)}(t)}{m-\alpha+(m-\alpha)^2} \right),
\end{aligned}
\tag{16}
$$

where $m = [\alpha] + 1$ and $Y_2(\tau_l, \tau_{l-1}, s_k^{(l)})$ is the solution at τ_{l-1} to the following equation:

$$
\begin{cases}
Y_2'(t) = s_k^{(l)} Y_2(t) + y^{(m)}(t), \\
Y_2(\tau_l) = 0.
\end{cases}
\tag{17}
$$

3. Numerical Examples

Two numerical examples for fractional calculus are given in this section. These numerical simulations show that our algorithms are efficient.

Example 1. Let $y(t) = t^4$, numerical solution and the error to fractional integral are given in Figs. 5-8. And the results are also compared with the numerical solution by using Diethelm's method in [5]. It shows that our method not only improves the algorithm complexity of fractional integral but also remains the precision to some extent.

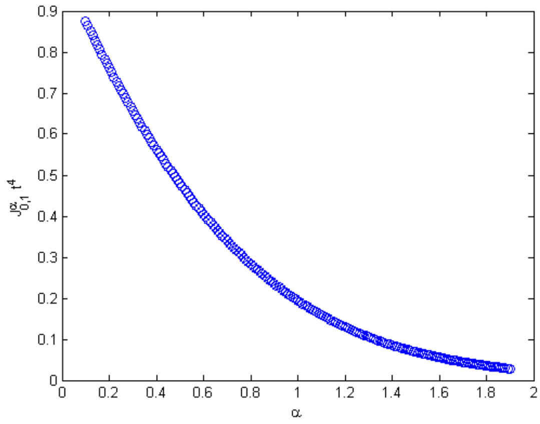

Figure 5: **Numerical solution by using fast convolution.**

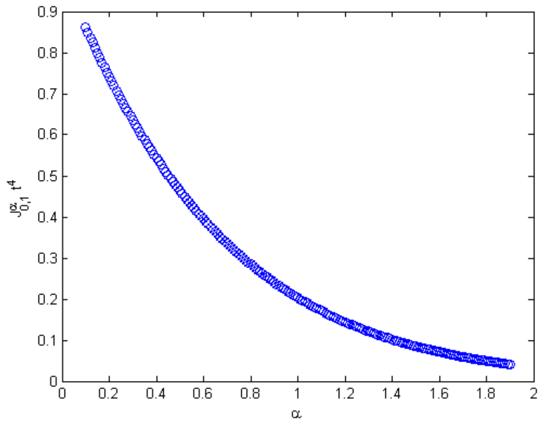

Figure 6: **Numerical solution by using Diethelm's method.**

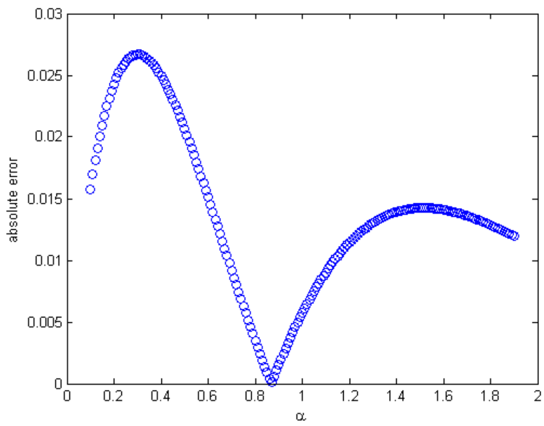

Figure 7: **Absolute error of** $J_{0,1}^{\alpha} t^4$ **by using the fast convolution.**

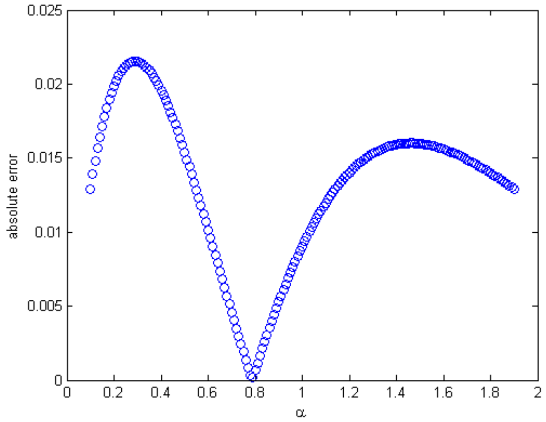

Figure 8: **Absolute error of $J_{0,1}^{\alpha} t^4$ by using the fast convolution and Diethelm's method.**

Example 2. Let $y(t) = t^4$, numerical solution and the error to fractional derivative in Caputo sense are given in Figs. 9-12. And the results are also compared with the numerical solution by using the method derived in [5]. It also shows that our method not only improves the algorithm complexity of fractional derivative but also remains the precision to some extent.

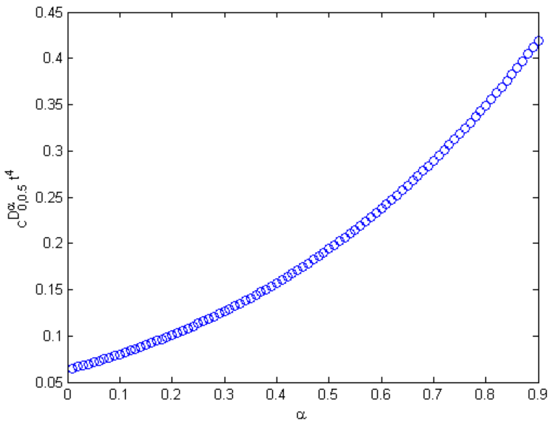

Figure 9: **Numerical solution by using fast convolution.**

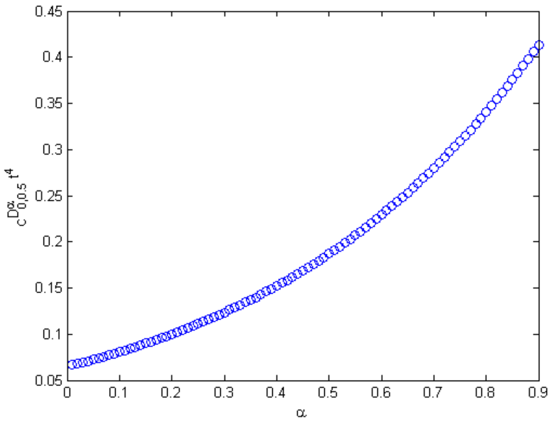

Figure 10: **Numerical solution by using Diethelm's method.**

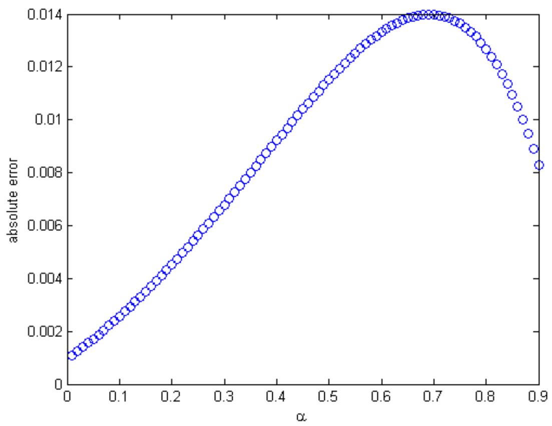

Figure 11: **Absolute error of** $_C D^{\alpha}_{0,0.5} t^4$ **by using the fast convolution.**

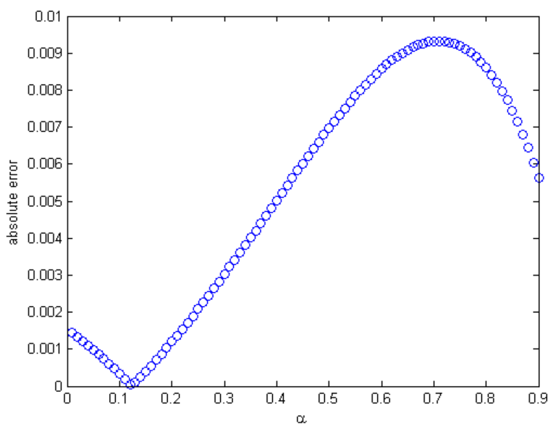

Figure 12: **Absolute error of** $_C D^{\alpha}_{0,0.5} t^4$ **by using fast convolution and Diethelm's method.**

Acknowledgement

This work was partially supported by the National Nature Science foundation of China (10872119) and the Shanghai Leading Academic Discipline Project (S30104).

References

[1] Lubich, C., Schadle, A., Fast convolution for nonreflecting boundary conditions, *SIAM J. Sci. Comput., 24* (2002), 1, pp. 161-182

[2] Schadle, A., López-Fernández, M., Lubich, C., Fast and oblivious convolution quadrature, *SIAM J. Sci. Comput., 28* (2006), 2, pp. 421-438

[3] López-Fernández, M., Lubich, C., Schädle, A., Adaptive, fast and oblivious convolution in evolution equations with memory, *SIAM J. Sci. Comput., 30* (2007), 2, pp. 1015-1037

[4] Li, C. P., Deng, W. H., Remarks on fractional derivatives, *Appl. Math. Comput., 187* (2007), 2, pp. 777-784

[5] Diethelm, K., Ford, N. J., Freed, A. D., Luchko, Yu., Algorithms for the fractional calculus: a selection of numerical methods, *Comput. Methods Appl. Mech. Eng., 194* (2005), 6-8, pp.743-773

[6] López-Fernández, M., Palencia C., Schädle, A., A spectral order method for inverting sectorial Laplace transforms, *Comput. SIAM J. NUMER. ANAL., 44* (2006), 3, pp. 1332-1350

[7] López-Fernández, M., Palencia C., On the numerical inversion of the Laplace transform of certain holomorphic mappings, *Appl. Numer. Math., 51* (2004), 2-3, pp. 289-303

[8] Li, C. P., Chen, A., Ye, J. J., Numerical approaches to fractional calculus and fractional ordinary differential equation, *J. Comput. Phys., 230* (2011), 9, pp. 3352-3368

[9] Li, C. P., Tao, C. X., On the fractional Adams method, *Comput. Math. Appl., 58* (2009), 8, pp. 1573-1588