NUMERICAL ALGORITHM BASED ON FAST CONVOLUTION FOR FRACTIONAL CALCULUS

by

An CHEN, Peng GUO, and Changpin LI*

Department of Mathematics, Shanghai University, Changhai, China

Original scientific paper
DOI: 10.2298/TSCI110407074C

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

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.

Definition 1. The fractional integral (or, Riemann-Liouville integral) of function \( y \) is defined by:

\[
J^\alpha_0 f(x) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-t)^{\alpha-1} y(t) \, dt
\]

Definition 2. The Caputo derivative of \( y \), for \( t > 0, m-1 < \alpha < m \), is defined as:

\[
D^\alpha_0 f(x) = \frac{1}{\Gamma(m-\alpha)} \int_0^x (t-r)^{m-\alpha-1} y^{(m)}(r) \, dr
\]

The fast convolution for fractional calculus

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

Simple calculation implies:

\[
F(s) = L[f(t), s] = \int_0^\infty e^{-st} f(t) \, dt = s^{-\alpha}, s \in \mathbb{C}
\]

* Corresponding author; lcp@shu.edu.cn
So the inversion of gives:

\[ f(t) = \frac{1}{2\pi i} \int_{\Gamma} e^{st} F(s) ds \]  

(4)

where \( \Gamma \) 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 eq. (4).

For a suitable contour \( \Gamma \) 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 \( \Gamma \) is a suitably chosen hyperbola which lies in the left branch, and is parameterized as follows:

\[ \mathbb{R} \rightarrow \Gamma, x \mapsto s = T(x, \lambda, a) = \lambda [1 - \sin(a + ix)] \]  

(5)

where \( \lambda > 0 \) is a scale parameter and \( a > 0 \), [2, 8]. Figure 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.

Then the truncated trapezoidal rule can be applied to approximate the kernel function (4) according to [8], that is:

\[
\begin{align*}
    f(t) &= \frac{1}{2\pi i} \int_{\Gamma} e^{st} F(s) ds \\
    &= \frac{1}{2\pi i} \int_{\Gamma} e^{x(T(x, \lambda, a))^\alpha} T'(x, \lambda, a) dx \\
    &\approx \sum_{k=-n}^{n} w_k e^{s_k T'(x, \lambda, a)} \\
    &= \sum_{k=-n}^{n} w_k e^{s_k T'(x, \lambda, a)} \\
    &\quad \text{with weights } w_k \text{ and quadrature nodes } s_k \text{ given by:} \\
    w_k &= -\frac{\tau}{2\pi i} T'(k\tau, \lambda, a), \quad s_k = T(k\tau, \lambda, a) \\
    \end{align*}
\]

(6)

where weights \( w_k \) and quadrature nodes \( s_k \) are given by:

\[
\begin{align*}
    w_k &= -\frac{\tau}{2\pi i} T'(k\tau, \lambda, a), \quad s_k = T(k\tau, \lambda, a) \\
    \end{align*}
\]

(7)

with step length \( \tau > 0 \).

In [9], the different choice of \( \Gamma \) and parameterizations were discussed in order to get a better convergence result. Due to this, we take:

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

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

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 $[0, t - D_t]$ and $[t - D_t, t]$ where $D_t = t/N$ with step number $N$. This:

\[
 J_t^\alpha y(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} y(\tau) d\tau = \int_0^{t-D_t} (t - \tau)^{\alpha-1} \Gamma(\alpha) y(\tau) d\tau + \int_{t-D_t}^t \frac{(t - \tau)^{\alpha-1}}{\Gamma(\alpha)} y(\tau) d\tau = A_1 + A_2 \quad (8)
\]

Firstly, we numerically seek $A_1$. Because the integral interval of $I_1$ is $[0, t - D_t]$, the range of $\tau$ 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 $[\Delta t, t]$: $I_l = [B^l \Delta t, (2B^l - 1)\Delta t]$ 

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 $I_l$ is obtained, also see fig. 4.

\[
 I_l = [B^l \Delta t, (2B^l - 1)\Delta t]
\]
1. \( t = \Delta r \cdot [0, \Delta r] \)
2. \( t = 2 \Delta r \cdot [0, \Delta r] \cup [\Delta r, 2 \Delta r] \)
3. \( t = 3 \Delta r \cdot [0, 2 \Delta r] \cup [2 \Delta r, 3 \Delta r] \)
4. \( t = 4 \Delta r \cdot [0, 2 \Delta r] \cup [2 \Delta r, 3 \Delta r] \cup [3 \Delta r, 4 \Delta r] \)
5. \( t = 5 \Delta r \cdot [0, 2 \Delta r] \cup [2 \Delta r, 4 \Delta r] \cup [4 \Delta r, 5 \Delta r] \)
6. \( t = 6 \Delta r \cdot [0, 4 \Delta r] \cup [4 \Delta r, 5 \Delta r] \cup [5 \Delta r, 6 \Delta r] \)
7. \( t = 7 \Delta r \cdot [0, 4 \Delta r] \cup [4 \Delta r, 6 \Delta r] \cup [6 \Delta r, 7 \Delta r] \)
8. \( t = 8 \Delta r \cdot [0, 4 \Delta r] \cup [4 \Delta r, 6 \Delta r] \cup [6 \Delta r, 7 \Delta r] \cup [7 \Delta r, 8 \Delta r] \)
9. \( t = 9 \Delta r \cdot [0, 4 \Delta r] \cup [4 \Delta r, 6 \Delta r] \cup [6 \Delta r, 8 \Delta r] \cup [8 \Delta r, 9 \Delta r] \)
10. \( t = 10 \Delta r \cdot [0, 4 \Delta r] \cup [4 \Delta r, 8 \Delta r] \cup [8 \Delta r, 9 \Delta r] \cup [9 \Delta r, 10 \Delta r] \)
11. \( t = 11 \Delta r \cdot [0, 4 \Delta r] \cup [4 \Delta r, 8 \Delta r] \cup [8 \Delta r, 10 \Delta r] \cup [10 \Delta r, 11 \Delta r] \)
12. \( t = 12 \Delta r \cdot [0, 8 \Delta r] \cup [8 \Delta r, 10 \Delta r] \cup [10 \Delta r, 11 \Delta r] \cup [11 \Delta r, 12 \Delta r] \)
13. \( t = 13 \Delta r \cdot [0, 8 \Delta r] \cup [8 \Delta r, 10 \Delta r] \cup [10 \Delta r, 12 \Delta r] \cup [12 \Delta r, 13 \Delta r] \)
14. \( t = 14 \Delta r \cdot [0, 8 \Delta r] \cup [8 \Delta r, 12 \Delta r] \cup [12 \Delta r, 13 \Delta r] \cup [13 \Delta r, 14 \Delta r] \)
15. \( t = 15 \Delta r \cdot [0, 8 \Delta r] \cup [8 \Delta r, 12 \Delta r] \cup [12 \Delta r, 14 \Delta r] \cup [14 \Delta r, 15 \Delta r] \)
16. \( t = 16 \Delta r \cdot [0, 8 \Delta r] \cup [8 \Delta r, 12 \Delta r] \cup [12 \Delta r, 14 \Delta r] \cup [14 \Delta r, 15 \Delta r] \cup [15 \Delta r, 16 \Delta r] \),

and so on.

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

\[
f(t) = \frac{1}{2 \pi i} \int_{\Gamma_{l}} e^{st} \, ds \approx \sum_{k} \left( \int_{I_{l}} e^{s_{k} t} \right)^{(l-\alpha)} \cdot t \in I_{l}
\]

with a suitably chosen complex contour \( \Gamma_{l} \), describing in [6].
Numerical simulations show that our algorithms are efficient.

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. 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 $\int_0^1 t^4$ by using the fast convolution

Figure 8. Absolute error of $\int_0^1 t^4$ by using the fast convolution and Diethelm’s method

Figure 9. Numerical solution by using fast convolution

Figure 10. Numerical solution by using 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. 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.

Acknowledgment

The present work was partially supported by the National Natural Science Foundation of China under Grant No. 10872119, Key Program of Shanghai Municipal Education Commission under grant no. 12ZZ084, and the Shanghai Leading Academic Discipline Project under Grant No. S30104.

References