

文章编号: 2095-4980(2022)06-0608-10

基于 IFFT 的 Lubich 数字分数微分器系数的快速算法

周 宇, 袁 晓*, 张月荣

(四川大学 电子信息学院, 四川 成都 610064)

摘 要: 从信号处理角度考察 Lubich 系数, 分析了 Lubich 系数的频域特性。设计了一种基于快速傅里叶逆变换(IFFT)的 Lubich 系数的快速算法。IFFT 算法直接求解的 Lubich 系数不准确, 在甚低阶运算时频域存在吉布斯效应, 新算法利用零频赋值可有效减弱该效应。数值仿真结果表明, 与 Lubich 准确系数相比, 在一定真分数运算阶范围内, 新算法求得的 Lubich 近似系数构建数字分数微分器有更好的效果, 且新算法计算复杂度低, 运算效率高。

关键词: 分数阶导数; 滤波函数; 高阶逼近; 频域特性; 计算复杂度

中图分类号: TN911.72

文献标志码: A

doi: 10.11805/TKYDA2021199

Fast algorithm based on IFFT for computing fractional Lubich coefficient of digital fractional differentiator

ZHOU Yu, YUAN Xiao*, ZHANG Yuerong

(College of Electronic and Information Engineering, Sichuan University, Chengdu Sichuan 610064, China)

Abstract: The Lubich coefficient is investigated from the point of view of signal processing, and the frequency characteristics of Lubich coefficient are analyzed. A fast algorithm based on Inverse Fast Fourier Transform (IFFT) for computing Lubich coefficient is designed. The Lubich coefficient directly solved by IFFT algorithm is not accurate. The Gibbs effect exists in the frequency domain with low order operations, and the new algorithm can reduce this effect effectively by zero-frequency assignment. The numerical simulation results show that, compared with Lubich accuracy coefficient, the Lubich approximation coefficients computed by the new algorithm have better performance in constructing the digital fractional differentiator with a certain proper fraction operation order range, and the new algorithm has low computational complexity and high efficiency.

Keywords: fractional derivative; filter function; high-order approximation; frequency characteristics; computational complexity

分数阶导数^[1]与积分理论已逐步发展为数学分析中的一个重要分支, 并引起人们在众多工程领域中的广泛研究使用, 例如信号分析与处理、非线性控制系统、分数阶混沌系统、分数阶图像处理^[2], 等等。

在众多分数阶微分定义中, 格林瓦尔莱特尼科夫(Grünwald-Letnikov, G-L)定义是最基本的定义, 它是连续函数整数阶导数的经典定义为出发点, 对整数阶微分的后向差分取极限, 将微积分的阶次由整数拓宽到分数推衍得到。其定义中, 式(1)为具有因果性的左导数, 式(2)为右导数:

$${}_a D_t^\alpha f(t) = \lim_{h \rightarrow 0} A_h^\alpha f(t), A_h^\alpha f(t) = h^{-\alpha} \sum_{k=0}^n g_k^{(\alpha)} f(t - kh) \quad (1)$$

$${}_a D_t^\alpha f(t) = \lim_{h \rightarrow 0} B_h^\alpha f(t), B_h^\alpha f(t) = h^{-\alpha} \sum_{k=0}^n g_k^{(\alpha)} f(t + kh) \quad (2)$$

式中: D 是分数阶微分算子; α 是运算阶, α 和 t 是与该操作相关联的限制端点, 也表示微分运算的上下界限; $f(t)$ 是自变量为 t 的函数, h 是采样步长, k 表示样本点的位置序号, $n=(t-a)/h$ 为当前时刻; $g_k^{(\alpha)}$ 称为 G-L 加权系

收稿日期: 2021-05-13; 修回日期: 2021-05-30

*通信作者: 袁 晓 email:yuanxiao@scu.edu.cn

数^[3], 具有递推关系:

$$g_0^{(\alpha)} = 1, g_k^{(\alpha)} = \left(1 - \frac{\alpha+1}{k}\right) g_{k-1}^{(\alpha)}, k = 1, 2, \dots \quad (3)$$

系数 $g_k^{(\alpha)}$ 可以为函数 $(1-z^{-1})^\alpha$ 的幂级数展开系数:

$$(1-z^{-1})^\alpha = \sum_{k=0}^{\infty} (-1)^k \binom{\alpha}{k} z^{-k} = \sum_{k=0}^{\infty} g_k^{(\alpha)} z^{-k} \quad (4)$$

式(1)~式(2)数值微分的逼近精确度阶为一阶, 即当式(1)、式(2)的 $f(t)$ 初值条件为0时, 式(1)、式(2)的计算精确度为 $O(h)$ 。为追求期望的高精确度, 高阶生成函数由Lubich^[4](卢比奇)通过拉格朗日插值多项式进行差分得到, Lubich生成函数表示为:

$$L_p^{(\alpha)}(z^{-1}) = \left(\sum_{i=0}^p \zeta_{p,i} z^{-i}\right)^\alpha = \left(\sum_{k=1}^p \frac{1}{k} (1-z^{-1})^k\right)^\alpha = \sum_{k=0}^{\infty} l_{p,k}^{(\alpha)} z^{-k} \quad (5)$$

式中: $\zeta_{p,i}$ 为生成函数系数; $l_{p,k}^{(\alpha)}$ 为Lubich加权系数。Lubich生成函数是一阶导数的 $(p+1)$ 点后向差分的 α 次幂, 具有 p 次逼近阶, 即式(1)、式(2)的计算精确度为 $O(h^p)$ 。

当运算阶 α 取非自然数的实数时, 式(5)中理想Lubich加权系数 $l_{p,k}^{(\alpha)}$ 是无限长的因果序列, 这在现实中无法实现。工程实践中, 可使用有限长Lubich加权系数 $l_{p,k}^{(\alpha)}(k=0\sim K-1, K \in \mathbf{N}^+)$ 构建分数阶数字微分器^[5-6]或求解分数阶微分方程^[7-8]。

本文仅在分数阶微分, 即 $\alpha > 0$ 情形下, 从两个方面开展研究:

- 1) 如何快速求解式(5)中的Lubich加权系数 $l_{p,k}^{(\alpha)}$ 。
- 2) 在频域考察Lubich滤波函数和有限长Lubich加权系数 $l_{p,k}^{(\alpha)}(k=0\sim K-1, K \in \mathbf{N}^+)$ 的逼近性能。

目前求解Lubich准确系数 $l_{p,k}^{(\alpha)}$ 的部分算法有: 幂级数展开算法^[9]、零点-卷积算法^[10]及递推算法^[11]。幂级数展开算法需对符号变量进行多次求导, 特别当系数序列长度 K 很大时, 计算量大。零点-卷积算法进一步减少了计算量, 但原算法步骤中存在多个复数序列连续卷积, 计算繁琐, 利用卷积的基本性质可在原算法基础上减少部分计算量。文献[11]给出的递推算法利用代数变换实现了更少的计算量, 但在高阶逼近时仍需一定迭代的乘加运算。

本文提出一种基于IFFT的近似求解Lubich系数的快速算法。对比上述3种算法求解的准确Lubich系数, 快速算法近似求解的系数在一定真分数运算阶范围内, 有更好的构建分数阶数字微分器效果。且快速算法的计算开销更小, 运算效率更高。

1 Lubich滤波函数

将 $z = e^{j\omega}$ 代入式(5)得Lubich生成函数的滤波函数:

$$L_p^{(\alpha)}(\omega) = L_p^{(\alpha)}(e^{-j\omega}) = \left(\sum_{i=0}^p \zeta_{p,i} e^{-j\omega}\right)^\alpha = \left(A_p(\omega) e^{j\theta_p(\omega)}\right)^\alpha = A_p^{(\alpha)}(\omega) e^{j\alpha\theta_p(\omega)}, |\omega| \leq \pi \quad (6)$$

理想的 α 阶数字微分滤波函数为:

$$D^{(\alpha)}(\omega) = (j\omega)^\alpha, |\omega| \leq \pi \quad (7)$$

数字信号的频谱周期为 2π 且具有共轭对称性, 仅需取 $\omega \in [0, \pi]$ 分析。为便于频域特征分析, 令

$$\omega = \pi \times 10^\sigma \Leftrightarrow \sigma = \lg(\omega/\pi), \sigma \in (-\infty, 0] \quad (8)$$

式中 σ 表示频率指数变量。将 σ 代入式(6)得到Lubich生成函数的幅频特征函数 $A_p^{(\alpha)}(\sigma)$ 、相频特征函数 $\Theta_p^{(\alpha)}(\sigma)$ 、阶频特征函数 $\mu_p^{(\alpha)}(\sigma)$ 。

$$A_p^{(\alpha)}(\sigma) = \lg |A_p^{(\alpha)}(10^\sigma \pi)|, \sigma \in (-\infty, 0] \quad (9)$$

$$\Theta_p^{(\alpha)}(\sigma) = \alpha \theta_p(10^\sigma \pi), \sigma \in (-\infty, 0] \quad (10)$$

$$\mu_p^{(\alpha)}(\sigma) = dA_p^{(\alpha)}(\sigma)/d\sigma, \sigma \in (-\infty, 0] \quad (11)$$

幅频特征函数和相频特征函数表征了Lubich滤波函数的频域特征。相频特征函数和阶频特征函数可完全表征Lubich滤波函数的运算特征。阶频特征是幅频特征的微分,不仅包含了幅频特征的信息,还直接表示了滤波函数的运算阶。同理,将 ϖ 代入式(7)得到理想的 α 阶数字微分的幅频特征函数 $D_A^{(\alpha)}(\varpi)$ 、相频特征函数 $D_\theta^{(\alpha)}(\varpi)$ 、阶频特征函数 $D_\mu^{(\alpha)}(\varpi)$ 。

$$D_A^{(\alpha)}(\varpi) = \lg |(j\varpi)^\alpha| = \lg |\varpi|^\alpha, \varpi \in (-\infty, 0] \quad (12)$$

$$D_\theta^{(\alpha)}(\varpi) = \arg \{(j\varpi)^\alpha\} = \pi\alpha/2, \varpi \in (-\infty, 0] \quad (13)$$

$$D_\mu^{(\alpha)}(\varpi) = dD_A^{(\alpha)}(\varpi)/d\varpi = \alpha, \varpi \in (-\infty, 0] \quad (14)$$

为了衡量Lubich滤波函数逼近理想的 α 阶数字微分滤波函数的性能,定义频域误差函数 $\varepsilon_p^{(\alpha)}(\varpi)$ 、相频逼近误差函数 $\varepsilon_{\theta,p}^{(\alpha)}(\varpi)$ 、阶频逼近误差函数 $\varepsilon_{\mu,p}^{(\alpha)}(\varpi)$ 为:

$$\varepsilon_p^{(\alpha)}(\varpi) = |L_p^{(\alpha)}(\varpi) - D^{(\alpha)}(\varpi)| \quad (15)$$

$$\varepsilon_{\theta,p}^{(\alpha)}(\varpi) = \theta_p^{(\alpha)}(\varpi) - \arg \{D^{(\alpha)}(\varpi)\} = \theta_p^{(\alpha)}(\varpi) - \pi\alpha/2 \quad (16)$$

$$\varepsilon_{\mu,p}^{(\alpha)}(\varpi) = \mu_p^{(\alpha)}(\varpi) - D_\mu^{(\alpha)}(\varpi) = \mu_p^{(\alpha)}(\varpi) - dD_A^{(\alpha)}(\varpi)/d\varpi \quad (17)$$

极大逼近误差为:

$$\varepsilon_{p,\max}^{(\alpha)} = \max_{\omega \in [0, \pi]} [\varepsilon_p^{(\alpha)}(\omega)] \quad (18)$$

式(18)可衡量逼近精确度,逼近程度的总体性能可用平均误差 $S_{1,p}^{(\alpha)}$ 和均方误差 $S_{2,p}^{(\alpha)}$ 评价,两者分别定义为:

$$S_{1,p}^{(\alpha)} = \frac{1}{\pi} \int_0^\pi \varepsilon_p^{(\alpha)}(\omega) d\omega, S_{2,p}^{(\alpha)} = \frac{1}{\pi} \int_0^\pi [\varepsilon_p^{(\alpha)}(\omega)]^2 d\omega \quad (19)$$

取 $\varpi \in [-3.0, 0]$,半阶运算($\alpha=0.5$)的Lubich滤波函数的频域特性及频域误差函数如图1所示。

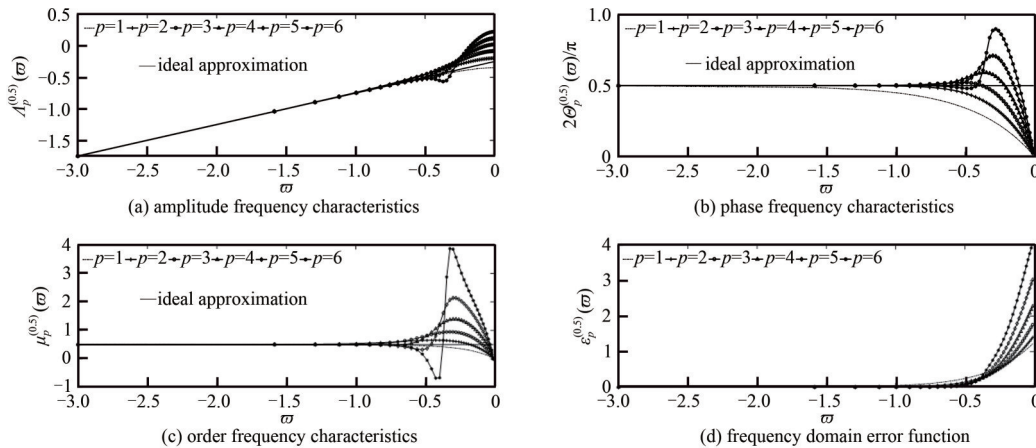


Fig.1 Frequency domain features and frequency error of generating function with different approximation orders

图1 不同逼近阶的频域特性和频域误差函数图

从图1可见,除一阶逼近($p=1$)外,在零频 $\omega=0(\varpi \rightarrow -\infty)$ 附近, $L_p^{(\alpha)}(\varpi)$ 可以很好地逼近 $D^{(\alpha)}(\varpi)$,从 $\varpi=-0.5(\omega \approx 0.32\pi)$ 附近开始,一直到高频 $\omega=\pi(\varpi=0)$ 处,其逼近的绝对误差函数 $\varepsilon_p^{(\alpha)}(\varpi)$ 值呈非线性增加,且逼近阶 p 越大,增加的趋势越大,极大逼近误差 $\varepsilon_{p,\max}^{(\alpha)}$ 、平均误差 $S_{1,p}^{(\alpha)}$ 、均方误差 $S_{2,p}^{(\alpha)}$ 也随之增大。通过数值分析得知,除一阶逼近外,频域的逼近程度还与运算阶 α 有关:运算阶 α 的变化趋势与极大逼近误差 $\varepsilon_{p,\max}^{(\alpha)}$ 、平均误差 $S_{1,p}^{(\alpha)}$ 、均方误差 $S_{2,p}^{(\alpha)}$ 呈同增同减关系。对于一阶逼近,虽然其整体($0 \leq \omega \leq \pi$)极大逼近误差 $\varepsilon_{p,\max}^{(\alpha)}$ 、平均误差 $S_{1,p}^{(\alpha)}$ 、均方误差 $S_{2,p}^{(\alpha)}$ 最小,但在零频 $\omega=0$ 附近, $L_p^{(\alpha)}(\varpi)$ 逼近 $L_p^{(\alpha)}(\varpi)$ 的性能并不如高逼近阶($2 \leq p \leq 6$),这是由一阶逼近的相频特征函数的逼近误差较大造成的。

为确切定量表示逼近的程度,通常考虑阶频相对误差函数 $r_{\mu,p}^{(\alpha)}(\varpi)$ 、相频相对误差函数 $r_{\theta,p}^{(\alpha)}(\varpi)$ 和4种逼近精确度取值^[12]。

$$r_{\theta,p}^{(\alpha)}(\varpi) = \varepsilon_{\theta,p}^{(\alpha)}(\varpi) / \arg \{D^{(\alpha)}(\varpi)\} \times 100\% = \varepsilon_{\theta,p}^{(\alpha)}(\varpi) / (\pi\alpha/2) \times 100\% \quad (20)$$

$$r_{\mu,p}^{(\alpha)}(\omega) = \varepsilon_{\mu,p}^{(\alpha)}(\omega) / D_{\mu}^{(\alpha)}(\omega) \times 100\% \tag{21}$$

逼近精确度 r_i :

$$r_0 = 1\%; r_1 = 5\%; r_2 = 10\%; r_3 = 20\% \tag{22}$$

依据式 (20)~式 (22) 定义逼近带宽:

$$\begin{cases} \beta_{P,p,i}^{(\alpha)} = \omega_{PU,p,i}^{(\alpha)} - \omega_{PL,p,i}^{(\alpha)} \\ \beta_{O,p,i}^{(\alpha)} = \omega_{OU,p,i}^{(\alpha)} - \omega_{OL,p,i}^{(\alpha)} \end{cases}, i = 0, 1, 2, 3 \tag{23}$$

式中: 下标变量 i 表示逼近精确度等级指标; $\omega_{PU,p,i}^{(\alpha)}$ 与 $\omega_{OU,p,i}^{(\alpha)}$ 称为上限频率指数, $\omega_{PL,p,i}^{(\alpha)}$ 与 $\omega_{OL,p,i}^{(\alpha)}$ 称为下限频率指数; $\beta_{P,p,i}^{(\alpha)}$ 与 $\beta_{O,p,i}^{(\alpha)}$ 称为(阶频与相频)逼近带宽指数, 指在 α 阶运算的 p 阶逼近时, 在 $\beta_{P,p,i}^{(\alpha)} = \omega_{PU,p,i}^{(\alpha)} - \omega_{PL,p,i}^{(\alpha)}$ ($\beta_{O,p,i}^{(\alpha)} = \omega_{OU,p,i}^{(\alpha)} - \omega_{OL,p,i}^{(\alpha)}$) 带宽数量级范围内, 满足 $|r_{\beta,p}^{(\alpha)}(\omega)| \leq r_i (|r_{\mu,p}^{(\alpha)}(\omega)| \leq r_i)$ 。

Lubich 滤波函数的低频有效逼近使得 $\beta_{P,p,i}^{(\alpha)}$ 与 $\beta_{O,p,i}^{(\alpha)}$ 为无穷大 ($\omega \in (-\infty, 0]$), 此时 $\omega_{PU,p,i}^{(\alpha)}$ 与 $\omega_{OU,p,i}^{(\alpha)}$ 上限频率指数可衡量逼近的性能。半阶运算时, Lubich 滤波函数的上限频率指数如图 2 所示。

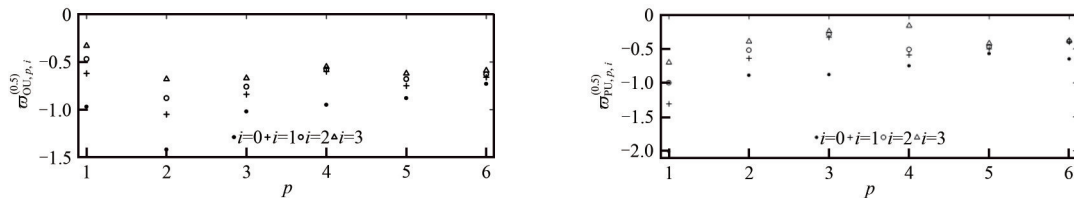


Fig.2 Upper-limit frequency exponent of Lubich filter function
图2 Lubich 滤波函数的上限频率指数

从图 2 可见, 当 $p=1$ 时, 相频的逼近效果较差, 尤其在超级逼近精确度 ($r_0=1\%$) 时。从整体上看, $p=5$ 和 $p=6$ 的逼近效果优于其他逼近阶, 且 $p=5$ 时体现在相频 ($\omega_{OL,p,i}^{(0.5)} \approx -0.57$), $p=6$ 时体现在阶频 ($\omega_{OU,p,i}^{(0.5)} \approx -0.73$)。通过数值分析得知, Lubich 滤波函数的上限频率指数与运算阶 α 无明显关系, 虽然 α 的减小会降低高频附近的绝对误差, 但绝对误差的降低仍不能使其相对误差函数值满足 4 种逼近精确度取值, 即不能明显影响上限频率指数。

综上可得: Lubich 滤波函数从低频延伸至零频的频带范围内可有效逼近理想的分数阶微分滤波函数, 高频附近的误差并不能通过改变逼近阶 p 来有效改善, 运算阶 α 的变化也几乎不影响上限频率指数。

2 Lubich 系数求解的算法

2.1 幂级数展开算法

根据 Lubich 加权系数与生成函数的对应关系, 对 $L_p^{(\alpha)}(z^{-1})$ 进行幂级数展开:

$$l_{p,k}^{(\alpha)} = \frac{1}{k!} \times \left. \frac{d^k L_p^{(\alpha)}(z^{-1})}{(dz^{-1})^k} \right|_{z^{-1}=0}, k = 0 \sim K-1 \tag{24}$$

式中 K 为展开项个数。对于运算阶 α 取自然数时, $l_{p,k}^{(\alpha)}$ 是有限长序列, 无需进行泰勒级数展开, 可直接对 $L_p^{(\alpha)}(z^{-1})$ 进行多项式展开获得 $l_{p,k}^{(\alpha)}$ 。显然, 幂级数展开算法计算量较大, 计算 $l_{p,k}^{(\alpha)}$ 的单个系数向量需对符号变量做 $K(K-1)/2$ 次求导。

2.2 零点-卷积算法

对 Lubich 生成函数因式分解:

$$L_p^{(\alpha)}(z^{-1}) = \left(\sum_{i=0}^p \varsigma_{p,i} z^{-i} \right)^\alpha = \varsigma_{p,0}^\alpha \prod_{j=1}^p (1 - z_j z^{-1})^\alpha = \varsigma_{p,0}^\alpha \prod_{j=1}^p \left(\sum_{k=0}^{\infty} g_k^{(\alpha)} z_j^k z^{-k} \right) = \varsigma_{p,0}^\alpha \prod_{j=1}^p \left(\sum_{k=0}^{\infty} h_{j,k} z^{-k} \right) = \varsigma_{p,0}^\alpha \sum_{k=0}^{\infty} h_{C,k} z^{-k} \tag{25}$$

$$h_{C,k} = h_{1,k} * h_{2,k} * \dots * h_{p,k} \cdot \varsigma_{p,0}^\alpha, h_{j,k} = g_k^{(\alpha)} \cdot z_j^k, k = 0 \sim K-1 \tag{26}$$

式中: z_j 为 $L_p(z^{-1})$ 的零点^[13]; $h_{C,k}$ 为连续卷积序列。高阶逼近 Lubich 加权系数 $l_{p,k}^{(\alpha)}$ 的求解变成了一阶逼近 G-L 加权系数 $g_k^{(\alpha)}$ 的连续卷积, 而 $g_k^{(\alpha)}$ 可由式 (3) 快速求得。

零点-卷积算法的计算量主要由三部分组成: 生成函数零点计算、被卷积序列计算及连续卷积运算, 计算量

重点集中在连续卷积部分。 p 阶逼近时,生成函数有 $p - [(-1)^p + 3]/2$ 个复零点和 $[(-1)^p + 3]/2$ 个实零点。对于共轭零点对,得到两个互为共轭的序列:

$$z_j \leftrightarrow h_{j,k} = g_k^{(\alpha)} \cdot z_j^k = x_{j,k} + jy_{j,k} \quad (27)$$

$$z_{j+1} = z_j^* \leftrightarrow h_{j+1,k} = g_k^{(\alpha)} \cdot (z_j^*)^k = h_{j,k}^* = x_{j,k} - jy_{j,k} \quad (28)$$

互为共轭的两个序列,卷积结果是一个实序列:

$$h_{j,k} * h_{j,k}^* = \sum_{m=-\infty}^{+\infty} h_{j,m} \cdot h_{j,k-m}^* = \sum_{m=-\infty}^{+\infty} (x_{j,m} + jy_{j,m})(x_{j,k-m} + jy_{j,k-m}) = x_{j,k} * x_{j,k} + y_{j,k} * y_{j,k} - jx_{j,k} * y_{j,k} + jy_{j,k} * x_{j,k} = x_{j,k} * x_{j,k} + y_{j,k} * y_{j,k} \quad (29)$$

由于复数零点共轭成对出现,加之卷积的可交换性,零点卷积算法的结果: $l_{p,k}^{(\alpha)} = \zeta_{p,0}^\alpha \cdot h_{C,k}$ 是一个实序列。对比无共轭性质的等长复序列卷积,两个等长共轭复序列卷积减少了两次实序列卷积运算和一次等长实序列加法运算。利用这一特点合理安排计算顺序——先计算连续复卷积序列中零点成共轭复数对应的序列,可有效减少计算量。

利用复实零点个数与卷积的关系,计算出不同逼近阶时,零点-卷积原算法的乘法复杂度 ${}_1C_{c \times}$ 和加法复杂度 ${}_1C_{c+}$ 及优化算法的乘法复杂度 ${}_2C_{c \times}$ 和加法复杂度 ${}_2C_{c+}$ 如下:

$$p=2: {}_1C_{c \times} = {}_2C_{c \times} = K^2 + 7K - 4, {}_1C_{c+} = {}_2C_{c+} = K^2 + 1 \quad (30)$$

$$p \geq 3: {}_1C_{c \times} = [4p - (-1)^p - 7]K^2 + [6p - 2(-1)^p - 3]K - 8p + 3(-1)^p + 9 \quad (31)$$

$${}_2C_{c \times} = \left[\frac{3}{2}p - \frac{(-1)^p + 7}{4} \right] K^2 + [6p - 2(-1)^p - 3]K - 8p + 3(-1)^p + 9 \quad (32)$$

$${}_1C_{c+} = [4p - (-1)^p - 7]K^2 + \left[-5p + \frac{(-1)^p + 21}{2} \right] K + (-1)^p - 1 \quad (33)$$

$${}_2C_{c+} = \left[\frac{3}{2}p - \frac{(-1)^p + 7}{4} \right] K^2 + \left[-\frac{p}{2} - \frac{7(-1)^p + 21}{4} \right] K - \frac{5}{2}p + \frac{7(-1)^p + 17}{4} \quad (34)$$

也可对连续卷积序列作 $pK - p + 1$ 点DFT,此时循环卷等于线性卷,并利用FFT计算DFT。零点卷积频域算法的实数乘法复杂度 ${}_3C_{c \times}$ 和实数加法复杂度 ${}_3C_{c+}$ 如下:

$${}_3C_{c \times} = [6p - 2(-1)^p - 3]K - 8p + 2(-1)^p + 6 + (pK - p + 1)[2(p + 1)\log_2(pK - p + 1) + 4(p - 1)] \quad (35)$$

$${}_3C_{c+} = [2p - (-1)^p - 3](K - 4) + (pK - p + 1)[3(p + 1)\log_2(pK - p + 1) + 2(p - 1)] \quad (36)$$

2.3 递推算法

文献[11]给出一种递推算法, Lubich系数 $l_{p,k}^{(\alpha)}$ 可以由下面的递推公式得到:

$$l_{p,k}^{(\alpha)} = \begin{cases} \zeta_{p,0}^\alpha, & \text{当 } k=0 \\ -\frac{1}{\zeta_{p,0}} \sum_{i=1}^p \zeta_{p,i} \left(1 - i \frac{1+\alpha}{k}\right) l_{p,k-i}^{(\alpha)}, & \text{当 } k > 0 \end{cases}, \quad k=0 \sim K-1 \quad (37)$$

递推算法的实数乘法复杂度 $C_{R \times}$ 和实数加法复杂度 C_{R+} 如下:

$$C_{R \times} = (K - p)(4p + 1) + (2p + 1)(p - 1) \quad (38)$$

$$C_{R+} = (p - 1)^2 + (K - p)p + 1 \quad (39)$$

幂级数算法、零点-卷积算法和递推算法均可计算准确的Lubich系数 $l_{p,k}^{(\alpha)}(k=0 \sim K-1)$ 。

3 IFFT快速算法

3.1 IFFT直接求解

对给定的Lubich滤波函数 $L_p^{(\alpha)}(\omega)$ 在 $\omega \in [0, 2\pi]$ 内进行 K 点等间距采样:

$$L_p^{(\alpha)}[n] = L_p^{(\alpha)}\left(\frac{2\pi n}{K}\right), n = 0 \sim K-1 \tag{40}$$

式中 n 表示采样点的位置序号。再对式(40)做快速傅里叶逆变换(IFFT)，有：

$${}_1^F I_{p,k}^{(\alpha)} = \frac{1}{K} \sum_{n=0}^{K-1} L_p^{(\alpha)}[n] e^{j\frac{2\pi nk}{K}}, k = 0 \sim K-1 \tag{41}$$

快速傅里叶逆变换^[14-16]是种近似算法，由式(40)和式(41)得到有限 K 个加权系数并有 ${}_1^F I_{p,k}^{(\alpha)} \approx I_{p,k}^{(\alpha)}$ ，且文献[11]分析了系数 ${}_1^F I_{p,k}^{(\alpha)}$ 的不准确性。理论上，可对误差分析和补偿使 ${}_1^F I_{p,k}^{(\alpha)}$ 逼近 $I_{p,k}^{(\alpha)}$ (${}_1^F I_{p,k}^{(\alpha)} \rightarrow I_{p,k}^{(\alpha)}$)。

3.2 零频赋值与误差建模

当运算阶 $\alpha > 0$ 时，频域的零频处 ($\omega = 0 \Leftrightarrow n = 0$) 理论值为： $L_p^{(\alpha)}(0) = 0$ 。根据加权系数的第一个值 $I_{p,0}^{(\alpha)}$ 等于生成函数的函数项 $\zeta_{p,0}^\alpha$ ，即首项已知，利用时域与频域的对应关系，对零频重新赋值，可使 Lubich 近似系数 ${}_1^F I_{p,k}^{(\alpha)}$ 与 Lubich 准确系数 $I_{p,k}^{(\alpha)}$ 的首项为零误差。此时，零频值为：

$$L_p^{(\alpha)}[0] = K\zeta_{p,0}^\alpha - \sum_{n=1}^{K-1} L_p^{(\alpha)}[n] \Leftrightarrow \zeta_{p,0}^\alpha = \sum_{n=0}^{K-1} L_p^{(\alpha)}[n] \tag{42}$$

零频重新赋值在时域等价于给近似系数 ${}_1^F I_{p,k}^{(\alpha)}$ 添加直流分量，即：

$${}_2^F I_{p,k}^{(\alpha)} = {}_1^F I_{p,k}^{(\alpha)} + L_p^{(\alpha)}[0]/K \tag{43}$$

并有 ${}_2^F I_{p,0}^{(\alpha)} - I_{p,0}^{(\alpha)} = 0$ ，但 ${}_2^F I_{p,k}^{(\alpha)}$ 与 $I_{p,k}^{(\alpha)}$ 除首项外的任意项仍存在误差。为此，定义 ${}_2^F I_{p,k}^{(\alpha)}$ 与 $I_{p,k}^{(\alpha)}$ 的相对误差：

$$r_p^{(\alpha)}[k] = ({}_2^F I_{p,k}^{(\alpha)} - I_{p,k}^{(\alpha)}) / I_{p,k}^{(\alpha)} \times 100\%, k = 1 \sim K-1 \tag{44}$$

并有 $r_p^{(\alpha)}[0] = 0$ 。取运算阶 $\alpha \in [0.1, 3]$ ，逼近阶 $p = 1 \sim 6$ ， $K = 200$ ，构建相对误差 $r_p^{(\alpha)}[k]$ 与自变量 α 和 k 的三维规律图，结果见图 3。

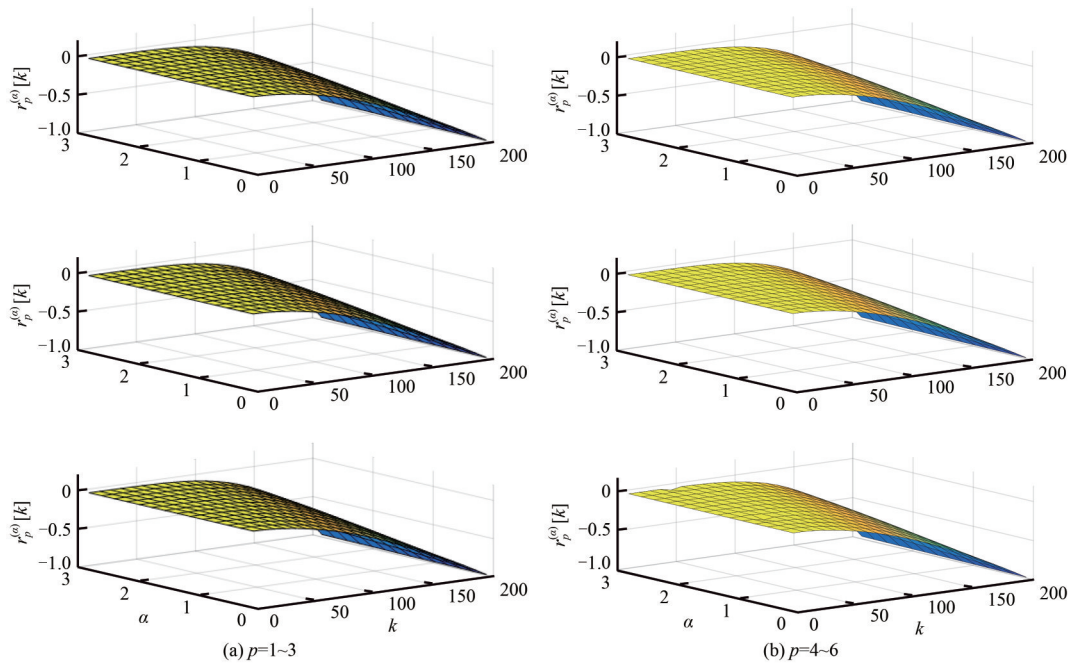


Fig.3 Relative error analysis
图3 相对误差分析图

由图 3 可见，相对误差 $r_p^{(\alpha)}[k]$ 随着 k 的增大呈现递增趋势，尾项几乎为 100% 的误差，其整体形状与逼近阶几乎无关，若 K 适当地大，则也与 K 几乎无关。从整体上看，几乎全都是递减的尖头钝尾形状的下近似。但 IFFT 算法求解系数中，在 $I_{p,k}^{(\alpha)}$ 未知时， $r_p^{(\alpha)}[k]$ 也未知。在理论上，算法相对误差 $r_p^{(\alpha)}[k]$ 是不变的，基于明显的规律特征，对相对误差建模。

$$r_{p,\alpha}[k] \approx r[k] = -\left(\frac{k}{K}\right)^{\nu(\alpha)} \times 100\%, k = 1 \sim K-1 \quad (45)$$

式中： $r[k]$ 为建立的相对误差模型； $\nu(\alpha)$ 为建模指数，一个仅与运算阶 α 有关的变量。利用Matlab中cftool工具箱拟合 $\nu(\alpha)$ 与 α 的对应数据点，并选择误差最小的二项式拟合法，即：

$$\nu(\alpha) = \nu_0 \alpha^2 + \nu_1 \alpha + \nu_2 \quad (46)$$

建模指数实验数据拟合见图4，计算得 $\nu_0 = 0.01859$ ， $\nu_1 = 0.7099$ ， $\nu_2 = 1.7$ 。

对比 ${}_2^F I_{p,k}^{(\alpha)}$ ，通过相对误差模型补偿得到的Lubich近似系数：

$${}_3^F I_{p,k}^{(\alpha)} = {}_2^F I_{p,k}^{(\alpha)} / (r[k] + 1) \quad (47)$$

除首项外($k = 1 \sim K-1$)更加接近准确系数 $l_{p,k}^{(\alpha)}$ 。

求解 ${}_2^F I_{p,k}^{(\alpha)}$ 的实数乘法复杂度 ${}_2 C_{F \times}$ 和实数加法复杂度 ${}_2 C_{F+}$ 为：

$${}_2 C_{F \times} = K \log_2 K + K + 1, \quad {}_2 C_{F+} = 2K \log_2 K + K + 1 \quad (48)$$

求解 ${}_3^F I_{p,k}^{(\alpha)}$ 的实数乘法复杂度 ${}_3 C_{F \times}$ 和实数加法复杂度 ${}_3 C_{F+}$ 为：

$${}_3 C_{F \times} = K \log_2 K + 3K + 4, \quad {}_3 C_{F+} = 2K \log_2 K + 2K + 3 \quad (49)$$

当 $K \rightarrow \infty$ 时，式(40)的结果显然等于 $L_p^{(\alpha)}(\omega)$ ，此时式(41)等价于：

$$\lim_{K \rightarrow \infty} {}_1^F I_{p,k}^{(\alpha)} = \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{n=0}^{K-1} L_p^{(\alpha)}[n] e^{j \frac{2\pi n k}{K}} = \int_0^{\infty} L_p^{(\alpha)}(\omega) e^{j k \omega} d\omega = l_{p,k}^{(\alpha)} \quad (50)$$

式(50)是Lubich系数 $l_{p,k}^{(\alpha)}$ 的傅里叶变换形式，并有 $\lim_{K \rightarrow \infty} {}_1^F I_{p,k}^{(\alpha)} - l_{p,k}^{(\alpha)} = 0$ 。故IFFT算法是收敛的。而零频赋值和误差建模可有效减少 K 为有限值时IFFT算法带来的截断误差。

4 算法运算效率比较

为便于图中的标注，将Taylor幂级数展开法标记为taylor；卷积的原算法、优化算法、频域算法分别标记为conv1,conv2,conv3；递推算法记为recursive；IFFT的直接求解，零频赋值，误差建模补偿算法分别标记为IFFT1,IFFT2,IFFT3，并统称IFFT算法。不同算法的乘法和加法复杂度见图5。可见，合理安排卷积次序的conv2计算复杂度比conv1更低，conv3是基于频域的算法，当展开项 K 较大时能逐渐体现优势。IFFT算法和递推算法的计算复杂度远低于卷积的3种算法。IFFT2和IFFT3算法的加法复杂度虽高于递推算法，但乘法复杂度更低，且乘法需要的运算时间远长于加法。IFFT算法的复杂度不受逼近阶 p 的影响，而递推算法的复杂度随逼近阶 p 的增大而增加。在高阶逼近时，IFFT算法在运算效率的优势更为显著。由式(30)~式(36)、式(38)~式(39)、式(48)~式(49)可知，在非整数阶情形下，各算法的计算复杂度与运算阶 α 无关。

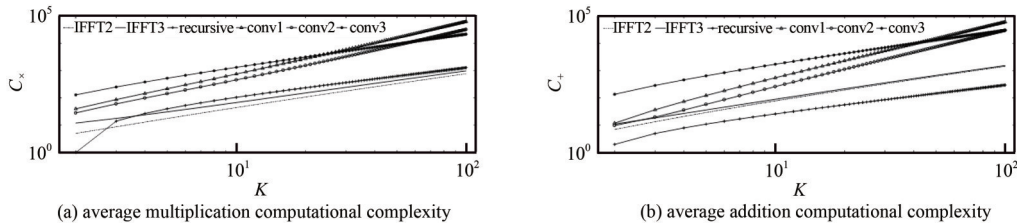


Fig.5 Average multiplication computational complexity and addition computational complexity for each $l_{3,k}^{(\alpha)}$
图5 求解单个 $l_{3,k}^{(\alpha)}$ 平均乘法计算复杂度及平均加法复杂度

表1 不同逼近阶下不同算法计算Lubich系数所用平均时间($\alpha = 0.5, K = 100$)

Table 1 Average time using different algorithms to compute Lubich coefficient with different order approximation ($\alpha = 0.5, K = 100$)

algorithm	running time/ms ($p=2$)	running time/ms ($p=3$)	running time/ms ($p=4$)	running time/ms ($p=5$)	running time/ms ($p=6$)
taylor	26.65	27.77	29.48	30.30	35.25
conv2	1.19	1.21	1.22	1.28	1.31
recursive	0.15	0.16	0.16	0.17	0.19
IFFT3	0.09	0.10	0.10	0.10	0.10
IFFT2	0.07	0.06	0.06	0.07	0.06

图 6 为在 CPU: Intel Core i5-9300H, 主频 2.40 GHz, 内存 8 GB 的计算机上运行不同算法求解 Lubich 系数的时间。可见, 零点-卷积算法运算与幂级数展开算法计算系数均需要较长的计算时间。随着 K 的增大, IFFT2 与 IFFT3 算法计算系数的时间曲线整体趋势较为稳定, 而递推算法所需时间在 $K \geq 10$ 后呈明显递增趋势。图 6 中曲线不连续处的“尖点”, 是计算机计算时间开销时无法避免的不准确数据点。

经统计, 不同算法计算 Lubich 系数 $l_{p,k}^{(\alpha)}$ 的平均运行时间见表 1, 结果保留小数点后 2 位。

计算机运算结果表明, 基于 IFFT 的快速算法比文献[11]的递推算法提高了至少一倍的运算速度, 若对比其他算法, 则运算效率高了 1~2 个数量级。

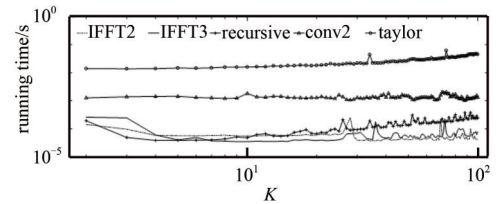


Fig.6 Running time comparison chart using different solution methods ($\alpha=0.5, K=100, p=3$)

图 6 不同算法求解 Lubich 系数运行时间对比图 ($\alpha=0.5, K=100, p=3$)

5 Lubich 数字分数微分器设计

理想的 Lubich 数字分数微分器 ($k=0 \sim \infty$) $l_{p,k}^{(\alpha)}$ 是无限冲激响应 (Infinite Impulse Response, IIR) 滤波器, 现实中实现困难。最简单的方法是截断系数序列长度, 即 K 取有限值, 来构建有限冲激响应 (Finite Impulse Response, FIR) 数字分数微分器。

图 7 给出通过 IFFT1, IFFT2, IFFT3 得到的 3 种 Lubich 近似系数 ${}_1^F l_{p,k}^{(\alpha)}, {}_2^F l_{p,k}^{(\alpha)}, {}_3^F l_{p,k}^{(\alpha)}$ 及 Lubich 准确系数 $l_{p,k}^{(\alpha)}$ 的运算特征曲线。可见, 在高频 ($\omega=0$) 附近, 4 种运算特征曲线逼近效果较差, 由图 1 知是 Lubich 滤波函数高频附近的固有属性, 但在零频 ($\omega \rightarrow -\infty$) 附近, 所有有限长系数的运算特征曲线与理想曲线误差较大。显然, 截断系数序列长度破坏了 Lubich 滤波函数低频有效逼近的性质。4 种系数 ${}_1^F l_{p,k}^{(\alpha)}, l_{p,k}^{(\alpha)}, {}_3^F l_{p,k}^{(\alpha)}$ 和 ${}_2^F l_{p,k}^{(\alpha)}$ 的运算特征曲线具有振荡现象—吉布斯效应, 除 ${}_3^F l_{p,k}^{(\alpha)}$ 外, 振荡幅度和振荡频率依次递减 (近似系数 ${}_3^F l_{p,k}^{(\alpha)}$ 的运算特征曲线可有效逼近 $l_{p,k}^{(\alpha)}$ 的运算特征曲线, 两者振荡程度类似)。振荡程度还具有不规则性 (振荡曲线非周期且幅度不同), 难以用解析函数描述, 但与运算阶 α 密切相关: 对于甚低阶 ($\alpha \rightarrow 0$) 振荡幅度和振荡频率高于其他情形 ($\alpha \rightarrow 1$), 这是由 Lubich 系数 $l_{p,k}^{(\alpha)}$ 的衰减性 [17] 引起的, 对于 α 趋近整数阶 ($\alpha \rightarrow 1$) 时 $l_{p,k}^{(\alpha)}$ 衰减更快, 此时截断的余项 ($l_{p,k}^{(\alpha)}, k \geq K$) 变换到频域的成分较小。

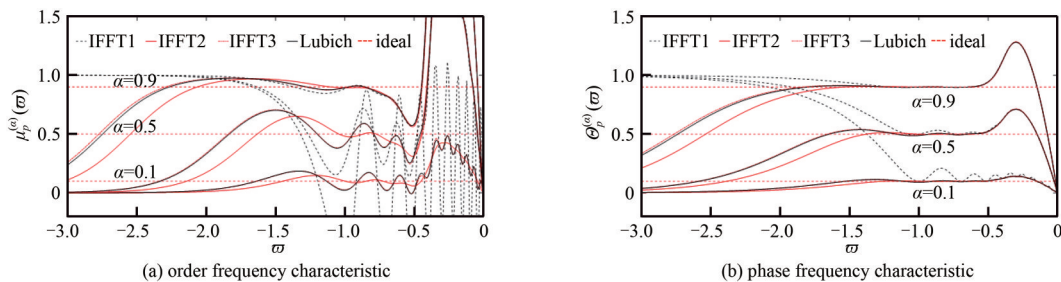


Fig.7 Operational characteristics curves of three Lubich approximate coefficients and Lubich coefficients ($p=5, K=20$)

图 7 3 种有限长 Lubich 近似系数与 Lubich 准确系数的运算特征曲线 ($p=5, K=20$)

为衡量近似系数 ${}_2^F l_{p,k}^{(\alpha)}$ 和准确系数 $l_{p,k}^{(\alpha)}$ 的频域逼近程度, 由图 8~图 9 给出系数 ${}_2^F l_{p,k}^{(\alpha)}$ 和 $l_{p,k}^{(\alpha)}$ 运算特征曲线的由式 (23) 定义的逼近带宽指数, 频率指数范围与图 7 保持一致 ($\omega \in [-3, 0]$)。通过数值分析可知, 对于甚低阶运算时 $l_{p,k}^{(\alpha)}$ 阶频存在的振荡现象, K 的增加不能降低其振荡幅度, 这是因为 K 的增加可减小幅频的振荡幅度, 但同时减小了振荡频率, 使得幅频振荡部分的斜率没有明显改变, 不能增加逼近带宽。大部分情形下 K 的增加可使阶频和相频有效逼近, 向低频处延伸, 若 K 足够大, 则逼近带宽指数曲线最终趋于平稳, 这是因为逼近带宽指数与频率指数范围有关, 此时仅在高频附近不能有效逼近。图 8 和图 9 曲线中的跳变点, 是由运算特征曲线振荡幅度的不规则性引起的。

从图 9 可见, 当运算阶 $\alpha \rightarrow 1$ 时, 系数 ${}_2^F l_{p,k}^{(\alpha)}$ 与 $l_{p,k}^{(\alpha)}$ 运算特征曲线的逼近带宽均高于运算阶 $\alpha \rightarrow 0$ 的情形。当 $K=20, p=5$ 时, 通过数值计算, 与准确系数 $l_{p,k}^{(\alpha)}$ 相比, 近似系数 ${}_2^F l_{p,k}^{(\alpha)}$ 在如下情形有更好的频域逼近性能:

- 1) 逼近精确度为 $r_0=1\%$, $\alpha \in (0, 0.95)$ 时, 阶频逼近性能更好, $\alpha \in (0, 0.72)$ 时相频逼近性能更好。
- 2) 逼近精确度为 $r_0=5\%$, $\alpha \in (0, 0.82)$ 时, 阶频逼近性能更好, $\alpha \in (0, 0.64)$ 时相频逼近性能更好。
- 3) 逼近精确度为 $r_0=10\%$, $\alpha \in (0, 0.71)$ 时, 阶频逼近性能更好, $\alpha \in (0, 0.36)$ 时相频逼近性能更好。
- 4) 逼近精确度为 $r_0=20\%$, $\alpha \in (0, 0.55)$ 或 $\alpha \in (0.69, 0.75)$ 时, 阶频逼近性能更好, 但 $\alpha \in (0, 1)$ 时相频逼近性能均无优势。

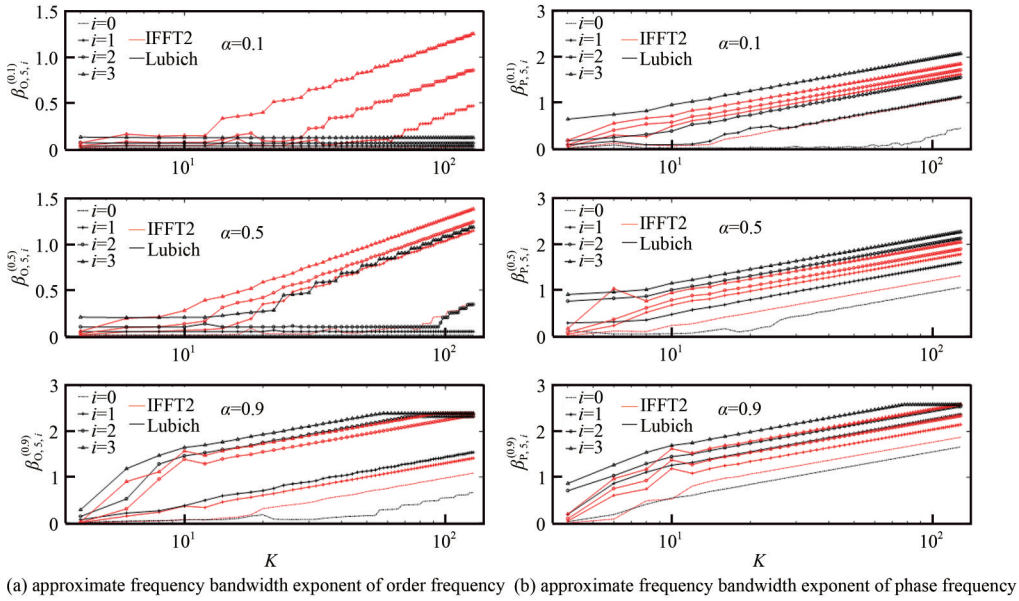


Fig.8 Approximate frequency bandwidth exponent of ${}^F_2 I_{p,k}^{(\alpha)}$ and $I_{p,k}^{(\alpha)}$'s operational characteristics curves with K as the independent variable($p=5$)
 图8 以 K 为自变量的系数 ${}^F_2 I_{p,k}^{(\alpha)}$ 与 $I_{p,k}^{(\alpha)}$ 运算特征曲线的逼近带宽指数($p=5$)

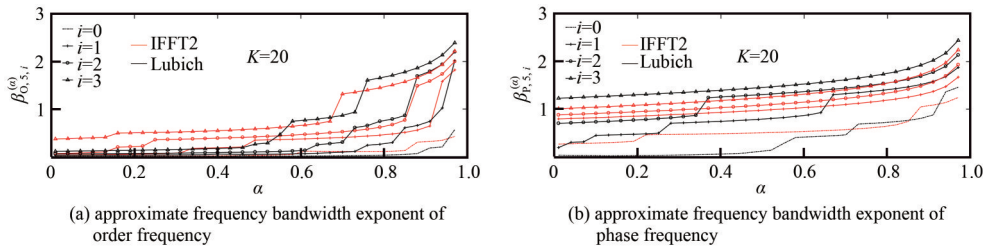


Fig.9 Approximate frequency bandwidth exponent of ${}^F_2 I_{p,k}^{(\alpha)}$ and $I_{p,k}^{(\alpha)}$'s operational characteristics with α as the independent variable ($p=5$)
 图9 以 α 为自变量的系数 ${}^F_2 I_{p,k}^{(\alpha)}$ 与 $I_{p,k}^{(\alpha)}$ 运算特征曲线的逼近带宽指数($p=5$)

实际设计中仅需选取效果最好的逼近阶构建Lubich数字微分器。由第1节已知Lubich滤波函数在逼近阶 $p=5$ 或 $p=6$ 时整体逼近性能好于其他逼近阶。通过数值分析可得，当 $p=6$ 时，近似系数 ${}^F_2 I_{p,k}^{(\alpha)}$ 相比于准确系数 $I_{p,k}^{(\alpha)}$ 仍有和 $p=5$ 时类似的优势。

综上所述可得：Lubich系数构建数字分数微分器更适用于运算阶 $\alpha \rightarrow 1$ 的情形；近似系数 ${}^F_3 I_{p,k}^{(\alpha)}$ 可有效逼近准确系数 $I_{p,k}^{(\alpha)}$ ，两者的运算特征曲线近似相同；在一定真分数运算阶范围内，近似系数 ${}^F_2 I_{p,k}^{(\alpha)}$ 的频域逼近性能好于准确系数 $I_{p,k}^{(\alpha)}$ 的逼近性能。

6 结论

本文从信号处理角度出发，首先分析Lubich滤波函数的频域特性，得出Lubich滤波函数低频有效逼近的结论。接着介绍3种常见的准确求解Lubich加权系数 $I_{p,k}^{(\alpha)}$ 的算法，利用卷积的基本性质减少零点-卷积算法的计算量，并提出一种基于IFFT的系数近似求解快速算法。快速算法具备两个优势：比文献[11]的算法提高至少一倍的运算效率；对比常见算法求解的准确Lubich系数 $I_{p,k}^{(\alpha)}$ ，快速算法求解的近似系数 ${}^F_2 I_{p,k}^{(\alpha)}$ 构建数字微分器可在一定真分数运算阶范围内有更好的效果。如何快速求解Lubich系数已有直观高效的算法，能否进一步突破具有一定挑战性。关于Lubich系数在频域的研究，还有如下值得深入的问题：能否进一步优化甚低阶 ($\alpha \rightarrow 0$) 时频域曲线出现的振荡现象；本文的方法及频域曲线优化是否适用于积分情形 ($\alpha < 0$)。

参考文献：

[1] 袁晓,张红雨,虞厥邦. 分数导数与数字微分器设计[J]. 电子学报, 2004,32(10):1658-1665. (YUAN Xiao,ZHANG Hongyu,YU Juebang. Fractional derivative and digital differentiator design[J]. Acta Electronica Sinica, 2004,32(10):1658-1665.)
 [2] 黄果,许黎,陈庆利,等. 非局部多尺度分数阶微分图像增强算法研究[J]. 电子与信息学报, 2019,41(12):2972-2979. (HUANG Guo,XU Li,CHEN Qingli,et al. Research on non-local multi-scale fractional differential image enhancement algorithm[J].

- Journal of Electronics & Information Technology, 2019,41(12):2972-2979.)
- [3] 张恒,袁晓,帅晓飞,等. 分数演算的G-L数值算法中加权系数求解[J]. 四川大学学报(自然科学版), 2007,44(4):831-834. (ZHANG Heng, YUAN Xiao, SHUAI Xiaofei, et al. Solving the weight coefficient of G-L numerical algorithm in the fractional calculus[J]. Journal of Sichuan University(Natural Science Edition), 2007,44(4):831-834.)
- [4] LUBICH Ch. Discretized fractional calculus[J]. Siam Journal on Mathematical Analysis, 1986,17(3):704-719.
- [5] 滕旭东,袁晓,赵元英,等. 数字分数微分器系数的快速算法[J]. 电子科技大学学报, 2004,33(4):457-460. (TENG Xudong, YUAN Xiao, ZHAO Yuanying, et al. Fast algorithm for coefficients of digital fractional differentiate[J]. Journal of University of Electronic Science and Technology of China, 2004,33(4):457-460.)
- [6] MITRA, KUMAR S. Digital signal processing-a computer-based approach[M]. 2nd ed. Singapore: McGraw-Hill Higher Education, 2002.
- [7] JIAN Huanyan, HUANG Tingzhu, GU Xianming. Fast second-order implicit difference schemes for time distributed-order and Riesz space fractional diffusion-wave equations[J]. Computers & Mathematics with Applications, 2021(94):136-154.
- [8] FANG Zhiwei, SUN Haiwei, WANG Hong. A fast method for variable-order Caputo fractional derivative with applications to time-fractional diffusion equations[J]. Computers & Mathematics with Applications, 2020,80(5):1443-1458.
- [9] 蒲亦非,袁晓,廖科,等. 现代信号分析与处理中分数阶微积分的五种数值实现算法[J]. 四川大学学报(工程科学版), 2005,37(5):118-124. (PU Yifei, YUAN Xiao, LIAO Ke, et al. Five numerical implementation algorithms for fractional calculus in modern signal analysis and processing[J]. Journal of Sichuan University(Engineering Science Edition), 2005,37(5):118-124.)
- [10] 王怡丹,袁晓. 高阶逼近Gr ü nwald-Letnikov分数阶加权系数的快速算法[J]. 信息技术, 2020,44(5):86-90,94. (WANG Yidan, YUAN Xiao. Fast algorithms for fractional Gr ü nwald-Letnikov weighting coefficients with high-order approximation[J]. Information Technology, 2020,44(5):86-90,94.)
- [11] 白鹭,薛定宇. 分数阶微积分的高精度递推算法[J]. 东北大学学报(自然科学版), 2018,39(4):604-608. (BAI Lu, XUE Dingyu. High precision recursive algorithm for computing fractional-order derivative and integral[J]. Journal of Northeastern University (Natural Science Edition), 2018,39(4):604-608.)
- [12] 袁晓. 分抗逼近电路之数学原理[M]. 北京:科学出版社, 2015. (YUAN Xiao. Mathematical principles of fractance approximation circuits[M]. Beijing: Science Press, 2015.)
- [13] 易舟,袁晓. 分形分抗逼近电路零极点的数值求解与验证[J]. 太赫兹科学与电子信息学报, 2017,15(1):98-103. (YI Zhou, YUAN Xiao. The numerical solution and verification of zero-pole for some fractal fractance approximation circuits[J]. Journal of Terahertz Science and Electronic Information Technology, 2017,15(1):98-103.)
- [14] PODLUBNY I. Fractional differential equations[C]// Mathematics in Science and Engineering. New York and London: Academic Press, 1999:161-176.
- [15] 薛定宇. 分数阶微积分学与分数阶控制[M]. 北京:科学出版社, 2018. (XUE Dingyu. Fractional calculus and fractional-order control[M]. Beijing: Science Press, 2018.)
- [16] PODLUBNY I. 分数微积分:理论基础与应用导论[M]. 袁晓注,译. 北京:电子工业出版社, 2021. (PODLUBNY I. Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications[M]. Translated by YUAN Xiao. Beijing: Publishing House of Electronics Industry, 2021.)
- [17] 袁晓,陈向东,李齐良,等. 微分算子与子波构造[J]. 电子学报, 2002,30(5):769-773. (YUAN Xiao, CHEN Xiangdong, LI Qiliang, et al. Differential operator and the constructon of wavelet[J]. Acta Electronica Sinica, 2002,30(5):769-773.)

作者简介:

周宇(1995-),男,浙江省江山市人,在读硕士研究生,主要研究方向为数字信号处理、分数阶微积分. email:18457077060@163.com.

袁晓(1964-),男,四川省中江县人,博士,副教授,主要研究方向为现代信息信号处理、分数微积分理论与应用、现代电路与系统理论与技术等.

张月荣(1994-),女,湖北省大悟市人,在读硕士研究生,主要研究方向为任意分数阶标度分形分抗的性能优化与运算恒定性研究.