

第 3 章



分数阶微积分: 定义与计算

正如以前指出的那样, 分数阶微积分学可以追溯到 Newton 和 Leibniz 刚创建传统微积分学的年代。由于没有统一并被广泛接受的定义, 分数阶微积分学在其早期发展过程中并没有很好的进展。直到 19 世纪中叶, 一些著名学者包括法国数学家 Liouville (1809–1882) 于 1834 年、德国数学家 Riemann (1826–1866) 于 1847 年、捷克数学家 Grünwald (1838–1920) 于 1867 年和俄国数学家 Letnikov (1837–1888) 于 1868 年分别提出了有意义的分数阶微积分的定义, 分数阶微积分学的领域才真正被建立起来。

后来, Liouville 和 Riemann 的定义统一成 Riemann–Liouville 定义, 而 Grünwald 和 Letnikov 的定义被整合成 Grünwald–Letnikov 定义。这两个定义一直在分数阶微积分学领域广泛使用, 后来被证明, 这两个定义在工程应用中是等效的。这两个定义都适合描述初始值为零函数的分数阶微积分问题。

对于非零初值的问题, 1967 年, 意大利数学家 Caputo 提出的定义是非常有用的, 特别是在分数阶微分方程的研究中, Riemann–Liouville 与 Grünwald–Letnikov 两个定义需要已知信号分数阶导数在初始时刻的值, 而 Caputo 定义需要已知信号及其整数阶导数在初始时刻的值, 更接近实际应用。

除了这几个广泛应用的定义之外, 文献中还有其他分数阶导数的定义, 如 Erdélyi–Kober 定义、Hadamard 定义、Marchaud 定义、Riesz 定义、Riesz–Miller 定义、Miller–Ross 定义、Weyl 定义等。本书不涉及这些分数阶微积分的定义, 但可以将本书介绍的数值算法拓展到其他定义中。

本章研究的主要问题是: 若原函数 $f(t)$ 或其样本点已知, 如何得出该函数的分数阶导数或积分。后面章节还将介绍如果原函数未知, 如何构造信号的分数阶导数与积分信号。

这里将引入一个统一的分数阶微分、积分的算子 $t_0 \mathcal{D}_t^\alpha$, 并广泛应用于本书的叙述, 其中, α 限于实数, t 为自变量, t_0 为该变量的下边界。

定义 3-1 ▶ 分数阶微积分的统一算子

统一的分数阶微积分算子 ${}_t \mathcal{D}_t^\alpha$ 的定义为

$${}_t \mathcal{D}_t^\alpha f(t) = \begin{cases} \frac{d^\alpha}{dt^\alpha} f(t), & \alpha > 0 \\ f(t), & \alpha = 0 \\ \int_t^\infty f(\tau) d\tau^{-\alpha}, & \alpha < 0 \end{cases} \quad (3-0-1)$$

说明 3-1 统一的微积分算子

(1) 若 $\alpha \geq 0$ 且 $t_0 = 0$, 则可以省略记号 t_0 , 如果自变量为 t 且没有其他变量, 则 t 也可以省略。

(2) 若 $\alpha > 0$, ${}_t \mathcal{D}_t^\alpha$ 算子表示函数对自变量 t 的 α 阶导数, $\alpha = 0$ 表示原信号, 若 $\alpha < 0$, 则表示 $-\alpha$ 阶积分。

(3) 如果 α 为复数, 则它的实部决定微分还是积分, 这超出本书的范围。

本章 3.1 节首先对整数阶 Cauchy 积分公式进行拓展, 给出分数阶的 Cauchy 积分公式, 但后续内容中不再考虑该定义。在 3.2 节和 3.3 节中, 分别对传统微积分定义进行直接扩展, 介绍分数阶微积分的 Grünwald-Letnikov 定义和 Riemann-Liouville 定义, 并给出这些定义的数值算法, 编写 MATLAB 的通用求解函数。3.4 节对非零初值函数的分数阶微积分概念进行拓展, 给出 Caputo 分数阶微积分定义。3.5 节将给出各种微积分定义之间的关系, 并介绍基于 Caputo 定义的分数阶微积分数值计算方法。3.6 节将给出分数阶微积分的性质, 并试图给出分数阶积分的物理解释方法。

必须指出的是:

(1) 本章给出的分数阶微积分计算方法只是常规的算法, 用于分数阶微积分的概略计算, 一般精度较低。第 4 章将全面介绍各种定义下函数分数阶微积分的高精度数值方法与 MATLAB 实现。

(2) 这里提到的数值解方法是在原函数或其样本点已知时提出的, 如果原函数未知, 其分数阶微积分信号的构造可以参见第 5 章。

3.1 分数阶 Cauchy 积分公式

在复分析领域有一个特别重要的 Cauchy 积分公式, 可以将已知复函数 $f(z)$ 的高阶导数问题转换为封闭曲线的积分问题。本节将探讨将 Cauchy 积分公式拓展到分数阶导数的分数阶 Cauchy 积分公式, 并给出相应分数阶微积分的性质。

3.1.1 Cauchy积分公式

首先观察 Cauchy 积分公式的整数阶导数表示

$$\frac{d^n}{dt^n} f(t) = \frac{n!}{2\pi j} \oint_C \frac{f(\tau)}{(\tau - t)^{n+1}} d\tau \quad (3-1-1)$$

式中, C 为光滑的封闭曲线, 在该区域内函数 $f(t)$ 是单值解析函数。

如果 n 被非整数 γ 取代, 在 $t = \tau$ 点处将出现孤立奇点。在计算封闭路径积分时应该移除该奇点, 并将阶乘替换成 Gamma 函数, 然后即可直接拓展式 (3-1-1) 处理分数阶导数的问题。

定义 3-2 ▶ 分数阶导数的 Cauchy 积分定义

函数 $f(t)$ 的分数阶导数的 Cauchy 积分定义为

$$\mathcal{D}_t^\gamma f(t) = \frac{\Gamma(\gamma + 1)}{2\pi j} \oint_C \frac{f(\tau)}{(\tau - t)^{\gamma+1}} d\tau \quad (3-1-2)$$

式中, 阶次 γ 可以是任意正实数。

3.1.2 常用函数的分数阶微分与积分公式

在传统微积分中, 众所周知, 三角函数的 n 阶导数可以如下计算:

$$\frac{d^n}{dt^n} \sin at = a^n \sin at + \frac{n\pi}{2} \quad (3-1-3)$$

$$\frac{d^n}{dt^n} \cos at = a^n \cos at + \frac{n\pi}{2} \quad (3-1-4)$$

如果 n 由任意实数 α 取代, 则上面的定义可以直接扩展, 得出三角函数的分数阶微积分公式

$$\mathcal{D}_t^\alpha \sin at = a^\alpha \sin at + \frac{\alpha\pi}{2} \quad (3-1-5)$$

$$\mathcal{D}_t^\alpha \cos at = a^\alpha \cos at + \frac{\alpha\pi}{2} \quad (3-1-6)$$

此外, 指数函数的分数阶导数可以由下式求出:

$$\mathcal{D}_t^\alpha e^{\lambda t} = \lambda^{-\alpha} E_{1,1-\alpha}(\lambda t) \quad (3-1-7)$$

幂函数 t^m 的 n 阶导数可以表示成

$$\frac{d^n}{dt^n} t^m = m(m-1) \cdots (m-n+1) t^{m-n} = \frac{m!}{(m-n)!} t^{m-n} \quad (3-1-8)$$

用 Gamma 函数取代阶乘, 则该式可以直接扩展到分数阶导数:

$$\mathcal{D}_t^\alpha t^m = \frac{\Gamma(m+1)}{\Gamma(m-\alpha+1)} t^{m-\alpha}, \quad m > -1 \quad (3-1-9)$$

这里, m 并不局限于整数, 它也可以是非整数。

文献 [1] 还给出了更多常用函数的分数阶导数公式, 感兴趣的读者可以参考该著作。不过, 该定义本身的条件也是比较苛刻的, 因为该定义要求 $f(t)$ 在封闭区域内为单值解析函数。本书将不再进一步探讨与使用基于 Cauchy 积分公式的分数阶微积分定义。

3.2 Grünwald–Letnikov 分数阶微积分定义与计算

Grünwald–Letnikov 的分数阶微积分定义是使用最广泛的分数阶微积分定义之一。本节先给出整数阶高阶导数的一般表示形式, 并对其拓展, 给出 Grünwald–Letnikov 的分数阶导数定义。本节还给出一些基于该定义的数值求解算法, 并通过例子比较各种算法的优劣。

3.2.1 高阶整数阶导数的推导

在给出 Grünwald–Letnikov 分数阶导数定义之前, 先观察一下整数阶导数的问题。先考虑函数 $f(t)$ 的一阶导数公式

$$\frac{d^1}{dt^1} f(t) = \lim_{h \rightarrow 0} \frac{1}{h} [f(t) - f(t-h)] \quad (3-2-1)$$

由上述的结果可以很容易地推导出二阶导数公式

$$\frac{d^2}{dt^2} f(t) = \lim_{h \rightarrow 0} \frac{1}{h^2} [f(t) - 2f(t-h) + f(t-2h)] \quad (3-2-2)$$

循环使用上述方法, 最终可以得出函数的 n 阶导数为

$$\frac{d^n}{dt^n} f(t) = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{j=1}^n \binom{n}{j} (-1)^j f(t-jh) \quad (3-2-3)$$

式中, 二项式展开式可以写成

$$(1-z)^n = \sum_{j=0}^n \binom{n}{j} (-1)^j z^j = \sum_{j=0}^n \frac{(-1)^j n!}{j! (n-j)!} z^j = \sum_{j=0}^n w_j z^j \quad (3-2-4)$$

且二项式系数可以由下式直接计算:

$$w_j = (-1)^j \binom{n}{j} = (-1)^j \frac{n!}{j! (n-j)!}, \quad j = 0, 1, 2, \dots, n \quad (3-2-5)$$

3.2.2 Grünwald–Letnikov 分数阶微分的定义

可以容易地将上面的 n 阶导数公式直接拓展到非整数 α 的情形。和整数阶不同的是, 二项式表达式不再是有限项的和, 而变成了无穷级数的形式, 即

$$(1-z)^\alpha = \sum_{j=0}^{\infty} (-1)^j \binom{\alpha}{j} z^j = \sum_{j=0}^{\infty} w_j z^j \quad (3-2-6)$$

这样,拓展的二项式系数变成

$$w_j = (-1)^j \binom{\alpha}{j} = \frac{(-1)^j \Gamma(\alpha + 1)}{\Gamma(j + 1) \Gamma(\alpha - j + 1)}, \quad j = 0, 1, 2, \dots \quad (3-2-7)$$

比较式(3-2-5)与式(3-2-7)中的二项式系数可见,后者只是将前者的阶乘替换成了更一般的Gamma函数,从而扩大了其适用范围。

假设 $t \leq t_0$ 时函数 $f(t)$ 的值为零,则无限项的和可以转换成有限项的和,这样可以引入 Grünwald–Letnikov 分数阶导数公式。

定义 3-3 ▶ Grünwald–Letnikov 定义

给定函数 $f(t)$ 的 α 阶导数的 Grünwald–Letnikov 定义为

$${}_{t_0}^{\text{GL}} \mathcal{D}_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{[(t-t_0)/h]} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (3-2-8)$$

式中, $[\cdot]$ 表示取最接近的整数。

说明 3-2 Grünwald–Letnikov 定义

- (1) 算子左上角的 GL 记号表示 Grünwald–Letnikov 定义,没有冲突时可略。
- (2) 可以看出整数阶微分只使用当前和前几个有限步长内的函数值,而分数阶微分涉及从 t_0 开始的所有函数值,因而可以认为分数阶导数是有记忆的。
- (3) 若 α 为整数,式(3-2-8)与式(3-2-3)完全一致。
- (4) 该定义同样适用于 $\alpha > 0$ 的微分和 $\alpha < 0$ 的积分。另外,若 $\alpha = 0$,由定义可知 ${}_{t_0}^{\text{GL}} \mathcal{D}_t^0 f(t) = f(t)$ 。
- (5) Grünwald–Letnikov 定义满足定义 3-1。

后面将介绍各种计算二项式系数的方法,并将介绍 Grünwald–Letnikov 分数阶微积分的数值计算方法。

3.2.3 Grünwald–Letnikov 分数阶微分与积分的数值计算

如果选择的计算步长 h 足够小,则式(3-2-8)中的求极限操作可以忽略,这样,Grünwald–Letnikov 定义下的分数阶导数与积分可以由下面的式子直接计算:

$${}_{t_0}^{\text{GL}} \mathcal{D}_t^\alpha f(t) \approx \frac{1}{h^\alpha} \sum_{j=0}^{[(t-t_0)/h]} w_j f(t-jh) \quad (3-2-9)$$

其中,二项式系数可以由式(3-2-7)直接计算。这样,似乎可以很自然地给出如下的计算方法,并写出 MATLAB 求解函数。

算法 3-1 ▶ 分数阶导数的直接计算方法

- (1) 计算出给定函数 $f(t)$ 在各个时刻的样本点并构造向量 $\mathbf{f}$ 。
- (2) 由式 (3-2-7) 计算二项式系数。
- (3) 通过式 (3-2-8) 直接计算分数阶导数。

根据算法 3-1 可以编写出如下的 MATLAB 函数 `glfdiff0()`：

```
function df=glfdiff0(f,t,gam)
    arguments, f(:,1), t(:,1) double, gam(1,1) double, end
    [f,h,n]=fdiffcom(f,t); J=0:(n-1); a0=f(1);
    if a0~=0 && gam>0, dy(1)=sign(a0)*Inf; end
    w=gamma(gam+1)*(-1).^J./gamma(J+1)./gamma(gam-J+1);
    for i=2:n, dy(i,1)=w(1:i)*f(i:-1:1)/h^gam; end
end
```

该函数的调用格式为 $\mathbf{f}_1 = \text{glfdiff0}(f, \mathbf{t}, \alpha)$ ，其中， $\mathbf{t}$ 为等间距的时间列向量， α 为阶次，这里的 f 既可以为函数 $f(t)$ 在 $\mathbf{t}$ 各点上函数值构成的列向量，也可以为 $f(t)$ 的函数句柄，换句话说，这时原始函数可以用匿名函数表示。分数阶导数的数值解在变元 $\mathbf{f}_1$ 列向量中返回。若 $\mathbf{t}$ 向量不是列向量，函数将自动转换 $\mathbf{t}$ 为列向量。

在该函数中使用了下一级公用函数 `fdiffcom()`，其内容为

```
function [y,h,n]=fdiffcom(f,t)
    y=f; if strcmp(class(f),'function_handle'), y=f(t); end
    h=t(2)-t(1); n=length(t);
end
```

遗憾的是，这样的算法有明显的缺陷：其中直接涉及大数的 Gamma 函数值，而 $\Gamma(172)$ 和以后各项在 MATLAB 双精度数据结构下已经成为无穷大项 `Inf`，所以后面各项的影响被完全忽略，导致最终计算结果出现大误差，所以分数阶导数的计算需要更可靠的方法。

上面算法的最大问题是在计算二项式系数时使用了 Gamma 函数，因此导致不可避免的误差。如果能避开 Gamma 函数，使用更有效的方法计算二项式系数，则可以得出更可靠的结果。下面将给出二项式系数的递推算法。

定理 3-1 ▶ 二项式系数的递推公式

二项式系数 w_j 可以通过下式递推求出：

$$w_0 = 1, \quad w_j = 1 - \frac{\alpha + 1}{j} w_{j-1}, \quad j = 1, 2, \dots \quad (3-2-10)$$

证明 由式 (3-2-7) 可见， $w_0 = 1$ 。对二项式系数当前项比前一项，并利用性质

$\Gamma(x+1) = x\Gamma(x)$, 就可以如下推导公式:

$$\begin{aligned}\frac{w_j}{w_{j-1}} &= \frac{(-1)^j \Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)} \bigg/ \frac{(-1)^{j-1} \Gamma(\alpha+1)}{\Gamma(j)\Gamma(\alpha-j+2)} \\ &= -\frac{\Gamma(j)\Gamma(\alpha-j+2)}{\Gamma(j+1)\Gamma(\alpha-j+1)} = -\frac{\Gamma(j)(\alpha-j+1)\Gamma(\alpha-j+1)}{j\Gamma(j)\Gamma(\alpha-j+1)} \\ &= -\frac{\alpha-j+1}{j} = 1 - \frac{\alpha+1}{j}\end{aligned}$$

等式两端同时乘以 w_{j-1} , 就可以证明式 (3-2-10) 中的递推公式。

这样, 利用式 (3-2-10) 递推求出二项式系数, 就可以由式 (3-2-9) 直接计算给定函数分数阶导数。由于递推算法成功地避免了 Gamma 函数的直接计算, 所以解决了算法 3-1 中的问题。可以证明, 下面算法的精度达到 $o(h)$ [2]。

算法 3-2 ▶ Grünwald–Letnikov 分数阶导数与积分

- (1) 计算给定信号在各个时刻的函数值, 构造向量 $\mathbf{f}$ 。
- (2) 由式 (3-2-10) 递推计算二项式系数 w_j 。
- (3) 由式 (3-2-9) 计算给定函数的分数阶微分或积分的值。

基于上述算法, 可以用 MATLAB 语言编写出 Grünwald–Letnikov 分数阶微积分数值计算的函数:

```
function dy=glfdiff(y,t,gam)
arguments, y(:,1), t(:,1) double, gam(1,1) double, end
[y,h,n]=fdiffcom(y,t); w=[1,zeros(1,n-1)]; a0=y(1); dy(1)=0;
if a0~=0 && gam>0, dy(1)=sign(a0)*Inf; end
for j=2:n, w(j)=w(j-1)*(1-(gam+1)/(j-1)); end
for i=1:length(t), dy(i,1)=w(1:i)*y(i:-1:1)/h^gam; end
end
```

该函数的调用格式为 $y_1 = \text{glfdiff}(y, t, \gamma)$, 与前面介绍的 $\text{glfdiff0}()$ 函数完全一致。和 $\text{glfdiff0}()$ 函数相比, 这里只修改了 w 向量的计算语句, 别的没有变化。如果 γ 为负值, 则该函数可以直接计算原函数的 $-\gamma$ 阶积分。

例 3-1 试求出常数的 0.75 阶 Cauchy 导数和 Grünwald–Letnikov 导数, 另外, 求出这两个定义下常数的 0.75 阶积分。

解 在整数阶微积分中, 常数的各阶导数均为零, 其一阶积分是一条斜线, 很自然地人们想知道, 常数的分数阶微积分到底是多少。

其实常数 1 可以表示成幂函数 $f(t) = 1 = t^0$, 这样由式 (3-1-9) 可见, 其分数阶导数与积分分别满足

$$\mathcal{D}_t^{0.75} y(t) = \frac{\Gamma(1) t^{-0.75}}{\Gamma(1-0.75)} = \frac{t^{-0.75}}{\Gamma(0.25)}, \quad \mathcal{D}_t^{-0.75} y(t) = \frac{\Gamma(1) t^{0.75}}{\Gamma(1+0.75)} = \frac{t^{0.75}}{\Gamma(1.75)}$$

由下面的语句可以得出常数的 Cauchy 定义与 Grünwald–Letnikov 定义的 0.75 阶导数与积分,如图 3-1 所示。可以看出,得出的两组曲线吻合度是很好的。

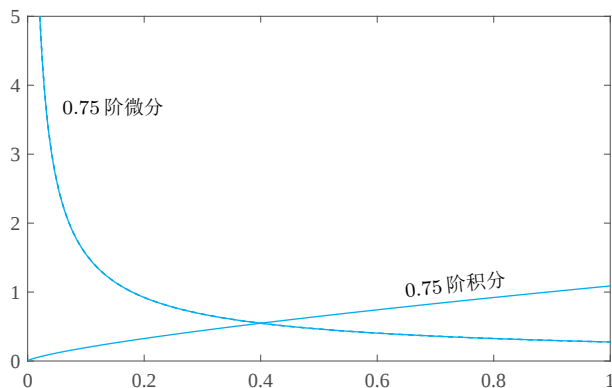


图 3-1 常数的 0.75 阶导数与积分

```
>> t=0:0.001:1; y=ones(size(t));
    y1=glfdiff(y,t,0.75); y1a=t.^(-0.75)/gamma(0.25);
    y2=glfdiff(y,t,-0.75); y2a=t.^(0.75)/gamma(1.75);
    y2a=gamma(1)*t.^(-0.75)/gamma(1-0.75);
    plot(t,y1,t,y1a,'--',t,y2,t,y2a,'--'), ylim([0 5])
```

选择不同的阶次 $\gamma = 0, 1/2, 3/4, 1$ 和 $5/4$, 函数的各阶导数如图 3-2 所示。可见, 当 γ 等于整数 0 和 1 时, 结果与原函数及其一阶导数完全一致。

```
>> t=0:0.001:0.5; y=ones(size(t)); gam=0:0.25:1.25;
    for g=gam, y1=glfdiff(y,t,g); plot(t,y1), hold on; end
    ylim([-3 5]), hold off
```

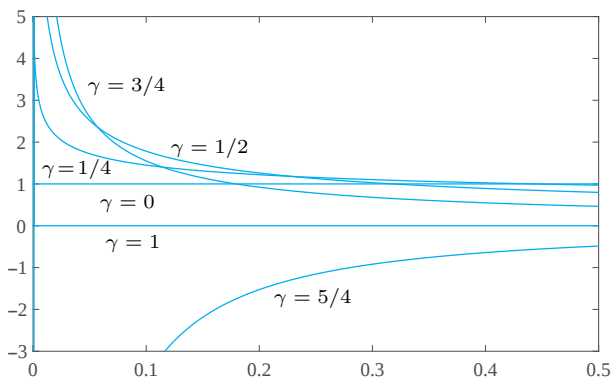


图 3-2 常数的各阶分数阶导数

函数 `glfdiff()` 还可以用于直接计算分数阶积分, 如果积分阶次分别设置为 0, $1/4, 1/2, 3/4, 1$ 和 $5/4$, 即设置阶次向量为 $\gamma = [0, -1/4, -1/2, -3/4, -1, -5/4]$, 则可

以由下面的语句绘制出分数阶积分,如图3-3所示。可见,当 $\gamma = 0$ 和 $\gamma = -1$ 时,得出的结果与原函数和整数阶积分完全一致。

```
>> t=0:0.01:1.5; y=ones(size(t)); gam=-1.25:0.25:0;
for g=gam, y1=glfdiff(y,t,g); plot(t,y1), hold on; end
```

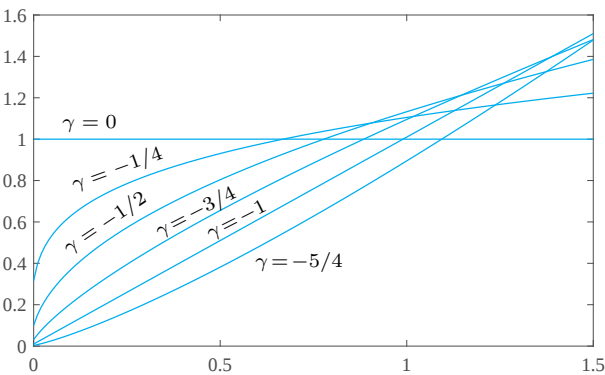


图3-3 不同阶次的分数阶积分曲线

例3-2 选择不同的计算步长求取阶跃函数的0.5阶导数并评价解的精度。

解 选择计算步长为 $h = 0.0001, 0.001, 0.01$ 和 0.1 ,就可以用下面的语句求取原函数的0.5阶导数,并和理论值对比得出其误差,结果在表3-1中列出。可见,理论上的 $o(h)$ 精度是正确的。

```
>> t0=0.2:0.2:1; y0=gamma(1)*t0(:).^(-0.5)/gamma(1-0.5);
t=0:0.0001:1; y=ones(size(t)); y1=glfdiff(y,t,0.5);
t=0:0.001:1; y=ones(size(t)); y2=glfdiff(y,t,0.5);
t=0:0.01:1; y=ones(size(t)); y3=glfdiff(y,t,0.5);
t=0:0.1:1; y=ones(size(t)); y4=glfdiff(y,t,0.5);
y1=y1(2001:2000:10001); y2=y2(201:200:1001);
y3=y3(21:20:101); y4=y4(3:2:11);
T=[t0, t0]; [y1 y2 y3 y4]', [y0-y1 y0-y2 y0-y3 y0-y4]']
```

表3-1 不同计算步长下的分数阶导数的误差

计算步长 h	t_k 时刻的导数				t_k 时刻的误差			
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
y_0	1.2616	0.8921	0.7284	0.6308				
0.0001	1.2615	0.8920	0.7284	0.6308	7.9×10^{-5}	2.8×10^{-5}	1.5×10^{-5}	7.1×10^{-6}
0.001	1.2608	0.8918	0.7282	0.6307	0.000786	0.000279	0.000152	7.1×10^{-5}
0.01	1.2537	0.8893	0.7269	0.6298	0.0079	0.00278	0.00152	0.00070
0.1	1.1859	0.8647	0.7134	0.6210	0.07571	0.02738	0.015	0.00701

在前面的例子中,因为分数阶导数的解析表达式已知,所以可以和理论值相比评价计算误差。而在一般应用中理论值是未知的,这时可以选择某计算步长得出结果,再选择一个更小的步长重新计算,看看二者是否一致。如果一致则说明结果是可靠的;若不一致则应该再减小计算步长重新计算,直到得出一致的结果。

例 3-3 现在考虑给定的函数 $f(t) = e^{-t} \sin(3t+1)$, $t \in (0, \pi)$, 可以看出,该函数的初始值不是零,试计算该信号的分数阶导数。

解 分别选择计算步长为 $h = 0.01$, $h = 0.001$ 与 $h = 0.0005$, 就可以由下面的语句得出已知信号的 0.5 阶 Grünwald-Letnikov 导数,如图 3-4 所示。可以看出,除了在 $t = 0$ 时刻附近的一个极小的区域外,几条曲线吻合度很高,说明 $h = 0.01$ 计算步长下的结果比较精确, $h = 0.001$ 与 $h = 0.0005$ 的结果几乎完全重合。

```
>> f=@(t)exp(-t).*sin(3*t+1);           %原函数还可以由匿名函数表示
t=0:0.001:pi; dy=glfdiff(f,t,0.5);      %在不同步长下求导
t1=0:0.01:pi; dy1=glfdiff(f,t1,0.5);
t2=0:0.0005:pi; dy2=glfdiff(f,t2,0.5);
plot(t,dy,t1,dy1,'--',t2,dy2,':'), axis([0,pi,-1.5 6])
```

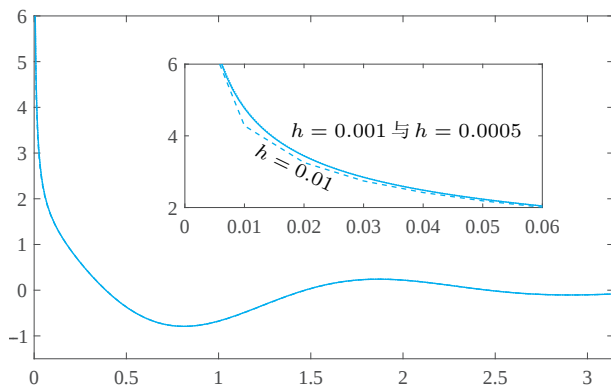


图 3-4 不同计算步长下结果的比较

选择不同的微分阶次 $\gamma = 0, 1/4, 1/2, 3/4, 1, 5/4$, 就可以由下面语句计算出该函数的分数阶微分,如图 3-5 所示。

```
>> t=0:0.01:pi; y=exp(-t).*sin(3*t+1); gam=0:0.25:1.25;
for g=gam, y1=glfdiff(y,t,g); plot(t,y1); hold on; end
axis([0 pi, -3 3]), hold off
```

下面的语句还可以直接得出不同阶次下的分数阶积分,如图 3-6 所示。可以看出,该函数的分数阶微分积分比整数阶微积分含有更丰富的信息。

```
>> for gam=-1:0.25:0
    y1=glfdiff(y,t,gam); plot(t,y1); hold on
end
```

```
xlim([0 pi]), hold off
```

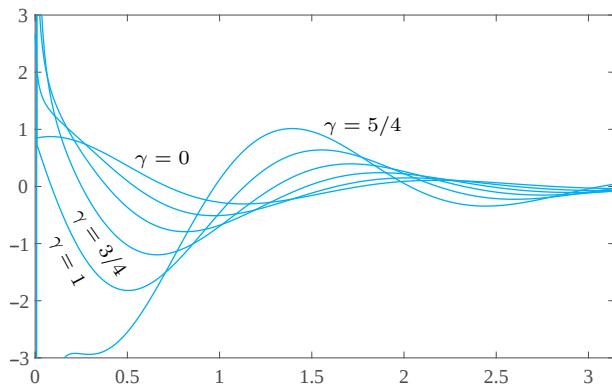


图 3-5 不同阶次的分数阶导数曲线

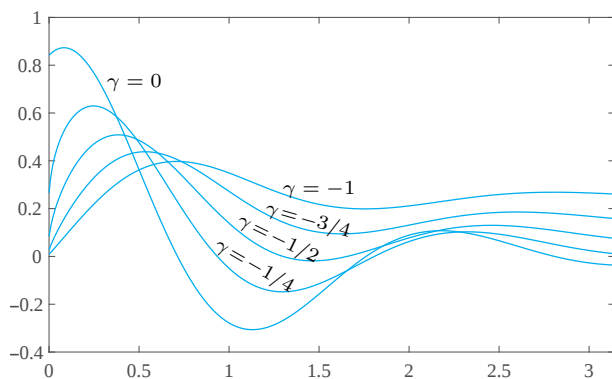


图 3-6 不同阶次的积分曲线

例 3-4 试用 Cauchy 定义和 Grünwald–Letnikov 定义分别求 $f(t) = \sin(3t + 1)$ 函数的 0.75 阶导数并比较其结果。

解 由式 (3-1-5) 给出的 Cauchy 积分公式可知, 函数的 0.75 阶导数为 ${}_0\mathcal{D}_t^{0.75}f(t) = 3^{0.75} \sin(3t + 1 + 0.75\pi/2)$, 而 Grünwald–Letnikov 定义的数值解可以由 `glfdiff()` 函数求出, 二者的曲线如图 3-7 所示。

```
>> t=0:0.01:pi;  
y=sin(3*t+1); y1=3^0.75*sin(3*t+1+0.75*pi/2);  
y2=glfdiff(y,t,0.75); plot(t,y1,t,y2,'--')  
axis([0 pi -4 8])
```

比较这两条导数曲线可见, 在 Cauchy 定义下, $t = 0$ 时并没有突然的跳变, 这是因为函数 $y(t)$ 在 $t \leq 0$ 时也被认为满足 $f(t) = \sin(3t + 1)$, 而在 Grünwald–Letnikov 定义中, 假设 $t < 0$ 时的初始值为零。

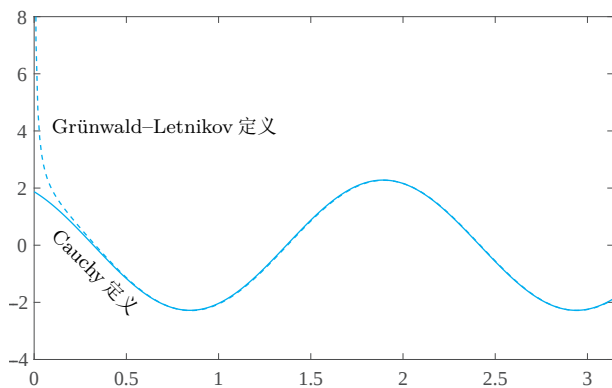


图 3-7 不同定义下的分数阶导数曲线

3.2.4 Podlubny 的矩阵算法

Podlubny 教授提出的矩阵算法也可以用来求给定函数的 Grünwald-Letnikov 分数阶导数与积分 [3], 其具体算法如下:

算法 3-3 ▶ Grünwald-Letnikov 分数阶导数的矩阵算法

- (1) 由给定函数先计算出原函数的样本点列向量 $\mathbf{f}$ 。
- (2) 由式 (3-2-10) 递推地求出二项式的系数 w_j 。
- (3) 式 (3-2-9) 还可以表示成下面的矩阵的形式, 由此求解分数阶导数:

$$\mathcal{D}^\alpha f = \frac{1}{h^\alpha} \begin{bmatrix} w_0 & & & & \\ w_1 & w_0 & & & \\ w_2 & w_1 & w_0 & & \\ \ddots & \ddots & \ddots & \ddots & \\ w_{N-1} & \ddots & w_2 & w_1 & w_0 \\ w_N & w_{N-1} & \ddots & w_2 & w_1 & w_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix} \quad (3-2-11)$$

注意, 式 (3-2-11) 中的矩阵其实是一个旋转的 Hankel 矩阵, 这样, 在 MATLAB 下可以实现矩阵算法, 求出给定函数的分数阶导数。

```
function dy=glfdiff_mat(y,t,gam)
arguments, y(:,1), t(:,1) double, gam(1,1) double, end
[y,h,n]=fdiffcom(y,t); w=[1,zeros(1,n-1)];
for j=2:n, w(j)=w(j-1)*(1-(gam+1)/(j-1)); end
dy=rot90(hankel(w(end:-1:1)))*y/h^gam;
end
```

可以看出, 这里给出的方法在数学上更简洁, 也更适合于信号的序贯求导, 如求解 $\mathcal{D}^\alpha \mathcal{D}^\beta \mathcal{D}^\gamma y(t)$ 的问题可以转换成矩阵乘积进行计算。该算法的局限性是, 如果总的计算点数 N 过大, 则可能需要连 MATLAB 都不能处理的大型矩阵运算。

3.2.5 短时记忆效应及其探讨

一般情况下,如果计算步长 h 选择得过小,或 $[(t-t_0)/h]$ 的值过大,则式 (3-2-9) 参与求和的点数会特别庞大,可能导致计算量显著增加,最终得不出计算结果,这样应该考虑减少计算点数。在实际应用中计算分数阶导数不一定非得使用以前 $[(t-t_0)/h]$ 所有的信息,用最近时间区间 $[t-L, t]$ 内的信息就能减少计算量^[2]:

$${}_{t_0}\mathcal{D}_t^\alpha f(t) \approx {}_{(t-L)}\mathcal{D}_t^\alpha f(t) \quad (3-2-12)$$

这种方法又称为短时记忆效应(short-memory effect)^[2,4],采用该方法,Grünwald-Letnikov 分数阶导数可以近似为

$$y(t) \approx \frac{1}{h^\alpha} \sum_{j=0}^{N(t)} w_j f(t-jh) \quad (3-2-13)$$

式中,

$$N(t) = \min \left\{ \frac{t-t_0}{h}, \frac{L}{h} \right\} \quad (3-2-14)$$

其中, L 称为记忆时长。这样,如何选择这个记忆时长 L 将成为最关键的问题。假设感兴趣时间区间 (t_0, T) 内函数 $f(t)$ 的值满足 $|f(t)| \leq M$, 则可见近似误差为

$$\begin{aligned} \Delta(t) &= {}_{t_0}\mathcal{D}_t^\alpha f(t) - {}_{(t-L)}\mathcal{D}_t^\alpha f(t) \\ &= \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^L \frac{f'(\tau)}{(t-\tau)^\alpha} d\tau \leq \frac{ML^{-\alpha}}{|\Gamma(1-\alpha)|} \end{aligned} \quad (3-2-15)$$

如果期望误差小于预先指定的正数 ε , 即 $\Delta(t) < \varepsilon$, 则记忆时长应该选为

$$L \geq \frac{M}{\varepsilon |\Gamma(1-\alpha)|}^{1/\alpha} \quad (3-2-16)$$

不过,该公式在选择记忆时长时可能过于苛刻。比如,若选择误差容限 $\varepsilon = 10^{-6}$, 记忆时长将是个非常大的数值,除非降低误差要求。例如,如果选择一个相对较大的误差容限 $\varepsilon = 10^{-3}$, 阶次 $\alpha = 0.5$, 函数的上界为 $M = 1$, 则需要的记忆时长为 $L = 318309$, 若选择 $\varepsilon = 10^{-4}$, 则记忆时长 $L = 31830988$, 如果阶次更小,如 $\alpha = 0.1$, 则记忆时长将是天文数字,所以在精度要求较高时,式 (3-2-16) 可能过于保守,有时短时记忆效应未必有实际意义。

基于短时记忆的思想,作者改写了 `glfdiff()` 函数,增加了短时记忆功能,编写了新的 `glfdiff_mem()` 函数,其调用格式为 `y1=glfdiff_mem(y,t,gam,L0)`, 其中, y, t, γ, y_1 与 `glfdiff()` 完全一致,增加的 L_0 为计算点数,即 $L_0 = L/h$ 。如果不给出 L_0 , 则得出的是 Grünwald-Letnikov 的分数阶微积分。

```
function dy=glfdiff_mem(y,t,gam,L0)
```

```
arguments, y(:,1), t(:,1), gam(1,1), L0=length(t); end
[y,h,n]=fdiffcom(y,t); w=[1,zeros(1,n-1)]; dy(1)=0;
for j=2:L0+1, w(j)=w(j-1)*(1-(gam+1)/(j-1)); end
for i=1:n
    L=min([i,L0]); dy(i,1)=w(1:L)*y(i:-1:i-L+1)/h^gam;
end, end
```

也可以考虑将矩阵算法3-3中的 w 向量后面的值设置为零,但这对提高矩阵运算的速度没有帮助。后面将给出例子探讨短时记忆效应的实际应用与效果。

例 3-5 在分数阶导数计算公式中,二项式系数 w_j 表明导数值对以往函数值的依赖程度,试绘制 $\alpha = 0.1$ 与 $\alpha = 0.8$ 时的 w_j 曲线。

解 利用式(3-2-10)中的递推公式可以直接计算出不同 α 值下的 w_j 系数,从而绘制出曲线,如图3-8所示,可以看出,分数阶导数对近处几个点的依赖程度很高,对远处的点依赖程度虽然减小,但也有些依赖。例如,当 $\alpha = 0.1$ 时, $w_{100} = -5.9076 \times 10^{-4}$ 。所以不能完全忽略以前点的信息。

```
>> t=0:0.1:3; a1=0.1; a2=0.8;
w1=1; for j=2:length(t), w1(j)=w1(j-1)*(1-(a1+1)/(j-1)); end
w2=1; for j=2:length(t), w2(j)=w2(j-1)*(1-(a2+1)/(j-1)); end
plot(t,w1,t,w2,'--')
```

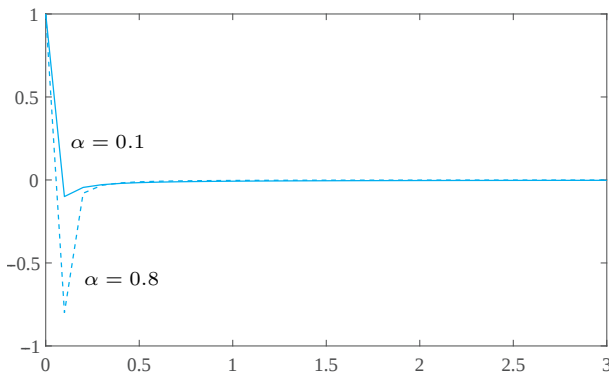


图3-8 不同阶次 α 下的二项式系数 w_j 曲线

例 3-6 再重新考虑例3-3中的函数 $f(t) = e^{-t} \sin(3t + 1)$ 。试利用短时记忆效应求取函数的0.5阶微分,并探讨短时记忆效应的近似效果。

解 选择计算步长 $h = 0.001$,并选择容许误差为 $\varepsilon = 10^{-3}$ 。因为函数的最大值为 $M = 1$,由式(3-2-16)计算出来的最小记忆时长为 $L = 1/(10^{-3}\Gamma(1 - 0.5))^{1/0.5} = 3.2 \times 10^5$ 。当然,这样的要求有些苛刻。

选择短一些的记忆时长,探讨近似的效果,对比结果在表3-2中给出。可以看出,如果记忆时长选择为 $L_0 = 1200$,基本上可以得出满意的近似结果。因为全时长总共3142

个点,选择记忆时长 $L_0 = 1200$ 将减少超过 $2/3$ 的计算时间,得出的曲线如图 3-9 所示。

表 3-2 当计算步长为 $h = 0.001$ 时记忆时长与误差的关系

记忆时长 L_0	误差范数	记忆时长 L_0	误差范数	记忆时长 L_0	误差范数
100	0.71736	1100	0.063559	2100	0.028305
200	0.38972	1200	0.057211	2200	0.026644
300	0.26417	1300	0.051879	2300	0.025141
400	0.19737	1400	0.047346	2400	0.023777
500	0.15597	1500	0.043449	2500	0.022532
600	0.12789	1600	0.04007	2600	0.021099
700	0.10767	1700	0.037115	2700	0.018628
800	0.092463	1800	0.034513	2800	0.015194
900	0.080647	1900	0.032206	2900	0.011026
1000	0.071227	2000	0.030149	3000	0.006443

```
>> t=0:0.001:pi; y=exp(-t).*sin(3*t+1);
y0=glfdiff(y,t,0.5); L0=[100:100:3000]; T=[];
for L=L0
    dy=glfdiff_mem(y,t,0.5,L); T=[T; L, norm(y0-dy,inf)];
end
dy=glfdiff_mem(y,t,0.5,1200);
plot(t,y0,t,dy,'--'); axis([0 pi -1 5])
```

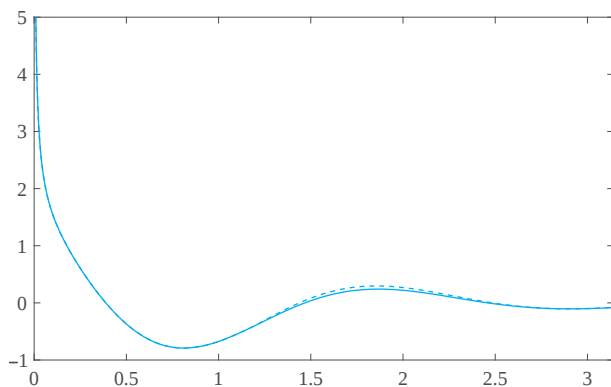


图 3-9 $L_0 = 1200$ 时的短时记忆效果

在同一个例子中,如果终止仿真时间变成 50 s,则用普通 Grünwald–Letnikov 算法将需要 2.74 s 才能得出分数阶导数,而使用记忆时长为 $L_0 = 1200$ 的短时记忆算法则只需 0.21 s。在实际应用中,在特别耗时的应用中可以考虑采用短时记忆算法,但得出的结果需要检验。

```
>> t=0:0.001:50; y=exp(-t).*sin(3*t+1);
tic, y0=glfdiff(y,t,0.5); toc
```

```
tic, dy=glfdiff_mem(y,t,0.5,1200); toc
```

下面给出一个短时记忆效应的反例。

例 3-7 试尝试用短时记忆方法求出阶跃函数的分数阶积分。

解 阶跃信号的 0.75 阶积分可以由 Cauchy 积分公式直接得出 $t^{0.75}/\Gamma(1+0.75)$ 。选择计算步长 $h = 0.001$, 则总共需要 5s, 产生 5000 个数据点。现在尝试一个较大的记忆时长, 如 $L_0 = 4000$, 则可以采用下面的语句直接计算分数阶积分, 如图 3-10 所示。可以看出, 即使采用这么大的记忆时长, 还会产生很大的计算误差, 说明在这个例子中短时记忆算法是不适合使用的, 必须使用全部信息。

```
>> t=0:0.001:10; y=ones(size(t));
y0=gamma(1)*t.^(0.75)/gamma(1+0.75); % Cauchy 积分公式
dy=glfdiff_mem(y,t,-0.75,4000); % 短时效应
dy1=glfdiff(y,t,-0.75); plot(t,dy,t,dy1,'--');
```

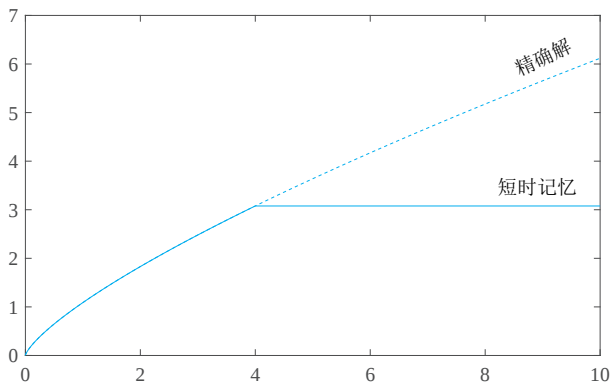


图 3-10 失效的短时记忆效应

从例 3-7 可以看出, 尽管很多人认为短时记忆效应是一种高效的近似方法, 事实上并不然。因为使用该方法还是有风险的, 有时甚至会得出错误或误导性的结果。由于式 (3-2-16) 过于苛刻, 要保证不太苛刻的精度要求, 也经常出现计算出的 L 远远大于实际计算点数的情况。所以除非特别必要, 请勿采用该算法, 而应该考虑更好的计算方法。

3.3 Riemann–Liouville 分数阶微积分定义与计算

前面给出的 Grünwald–Letnikov 定义虽然利于数值计算, 但在理论推导或公式证明方面并不容易处理, 所以也应该考虑其他数学定义。例如本节要介绍的由整数阶积分拓展而来的 Riemann–Liouville 定义等。

3.3.1 高阶整数阶积分公式

先考虑整数阶积分。很显然,给定函数 $f(t)$ 的一阶积分可以表示为

$$\frac{d^{-1}}{dt^{-1}} f(t) = \int_0^t f(\tau) d\tau \quad (3-3-1)$$

从该结果再求一次积分,就可以得出原函数的二阶积分:

$$\frac{d^{-2}}{dt^{-2}} f(t) = \int_0^t \int_0^\tau f(\tau) d\tau dt = \int_0^t f(\tau) (t - \tau) d\tau \quad (3-3-2)$$

类似地,可以推导出 n 阶积分公式为

$$\begin{aligned} \frac{d^{-n}}{dt^{-n}} f(t) &= \int_0^t \int_0^\tau \cdots \int_0^\tau f(\tau) d\tau_1 d\tau_2 \cdots d\tau \\ &\quad \text{\scriptsize n重} \quad \text{\scriptsize n重} \\ &= \frac{1}{(n-1)!} \int_0^t f(\tau) (t - \tau)^{n-1} d\tau = \frac{1}{(n-1)!} \int_0^t \frac{f(\tau)}{(t - \tau)^{1-n}} d\tau \end{aligned} \quad (3-3-3)$$

3.3.2 Riemann-Liouville 分数阶微积分定义

如果整数阶次 n 由实数 α 取代,则可以直接给出如下的 Riemann-Liouville 分数阶积分公式。

定义 3-4 ▶ Riemann-Liouville 积分定义

函数 $f(t)$ 的 α 阶 Riemann-Liouville 积分定义为

$${}^{\text{RL}}_{t_0} \mathcal{D}_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t \frac{f(\tau)}{(t - \tau)^{1-\alpha}} d\tau \quad (3-3-4)$$

式中,RL 标记表示 Riemann-Liouville 定义,如果没有冲突可以略去该标记。

Riemann-Liouville 定义是分数阶微积分领域应用较广的定义之一,算子 $\mathcal{D}$ 两端的下标为积分的上下限 [5]。

分数阶微分不能将积分中的 α 替换成 $-\alpha$ 直接定义,而是应该基于积分定义给出,现在考虑 β 阶分数阶微分问题。如果 $n - 1 < \beta \leq n$, 记 $n = [\beta]$, 则

$${}^{\text{RL}}_{t_0} \mathcal{D}_t^\beta f(t) = \frac{d^n}{dt^n} {}^{\text{RL}}_{t_0} \mathcal{D}_t^{-(n-\beta)} f(t) \quad (3-3-5)$$

基于积分定义,可以给出正式的 Riemann-Liouville 分数阶导数的定义。

定义 3-5 ▶ Riemann-Liouville 微分定义

给定函数 $f(t)$ 的 Riemann-Liouville 分数阶微分的定义为

$${}^{\text{RL}}_{t_0} \mathcal{D}_t^\beta f(t) = \frac{1}{\Gamma(n - \beta)} \frac{d^n}{dt^n} \int_{t_0}^t \frac{f(\tau)}{(t - \tau)^{1+\beta-n}} d\tau \quad (3-3-6)$$

式中, $n - 1 < \beta \leq n$, 且 $n = [\beta]$ 。

3.3.3 常用函数的 Riemann–Liouville 微积分公式

首先考虑下限 $t_0 = 0$ 时幂函数和指数函数的 Riemann–Liouville 分数阶导数的问题。

定理 3-2 ▶ 幂函数的分数阶导数

幂函数 $f(t) = t^\mu$, 且 $\mu > -1$, 则 ${}^{\text{RL}}_0 \mathcal{D}_t^\alpha f(t)$ 满足^[2]

$${}^{\text{RL}}_0 \mathcal{D}_t^\alpha t^\mu = \frac{\Gamma(\mu + 1)}{\Gamma(\mu + 1 - \alpha)} t^{\mu - \alpha} \quad (3-3-7)$$

定理 3-3 ▶ 指数函数的分数阶导数

指数函数 $f(t) = e^{\lambda t}$ 的 Riemann–Liouville 分数阶导数为

$${}^{\text{RL}}_0 \mathcal{D}_t^\alpha e^{\lambda t} = t^{-\alpha} E_{1,1-\alpha}(\lambda t) \quad (3-3-8)$$

证明 指数函数 $e^{\lambda t}$ 的 Taylor 级数展开为

$$e^{\lambda t} = \sum_{k=0}^{\infty} \frac{(\lambda t)^k}{k!} = \sum_{k=0}^{\infty} \frac{\lambda^k t^k}{\Gamma(k+1)} \quad (3-3-9)$$

利用式 (3-3-7) 的性质, 对式 (3-3-9) 右侧的项逐项计算 Riemann–Liouville 分数阶微积分, 即可以证明本定理:

$$\begin{aligned} {}^{\text{RL}}_0 \mathcal{D}_t^\alpha e^{\lambda t} &= \sum_{k=0}^{\infty} {}^{\text{RL}}_0 \mathcal{D}_t^\alpha \frac{\lambda^k t^k}{\Gamma(k+1)} = \sum_{k=0}^{\infty} \frac{\lambda^k \Gamma(k+1) t^{k-\alpha}}{\Gamma(k+1) \Gamma(k+1-\alpha)} \\ &= t^{-\alpha} \sum_{k=0}^{\infty} \frac{(\lambda t)^k}{\Gamma(k+1-\alpha)} = t^{-\alpha} E_{1,1-\alpha}(\lambda t) \end{aligned} \quad (3-3-10)$$

定理 3-4 ▶ 常用函数的 Riemann–Liouville 分数阶导数

一些常用函数的 Riemann–Liouville 分数阶导数公式^[2] 为

$${}^{\text{RL}}_t \mathcal{D}_t^\alpha H(t-a) = \frac{(t-a)^{-\alpha}}{\Gamma(1-\alpha)} H(t-a) \quad (3-3-11)$$

$${}^{\text{RL}}_0 \mathcal{D}_t^\alpha H(t-a) f(t) = H(t-a) {}^{\text{RL}}_a \mathcal{D}_t^\alpha f(t) \quad (3-3-12)$$

$${}^{\text{RL}}_0 \mathcal{D}_t^\alpha \delta(t) = \frac{t^{-\alpha-1}}{\Gamma(-\alpha)} \quad (3-3-13)$$

$${}^{\text{RL}}_0 \mathcal{D}_t^\alpha \delta^{(n)}(t) = \frac{t^{-\alpha-n-1}}{\Gamma(-\alpha-n)} \quad (3-3-14)$$

$${}^{\text{RL}}_0 \mathcal{D}_t^\alpha \cosh \sqrt{\lambda t} = t^{-\alpha} E_{2,1-\alpha}(\lambda t^2) \quad (3-3-15)$$

$${}^{\text{RL}}\mathcal{D}_t^\alpha \frac{\sinh\sqrt{\lambda t}}{\sqrt{\lambda t}} = t^{1-\alpha} E_{2,2-\alpha} \lambda t^2 \quad (3-3-16)$$

$${}^{\text{RL}}\mathcal{D}_t^\alpha e^{\lambda t} = \lambda^\alpha e^{\lambda t} \quad (3-3-17)$$

$${}^{\text{RL}}\mathcal{D}_t^\alpha e^{\lambda t + \mu} = \lambda^\alpha e^{\lambda t + \mu}, \lambda > 0 \quad (3-3-18)$$

$${}^{\text{RL}}\mathcal{D}_t^\alpha \sin \lambda t = \lambda^\alpha \sin(t + \alpha\pi/2) \quad (3-3-19)$$

$${}^{\text{RL}}\mathcal{D}_t^\alpha \cos \lambda t = \lambda^\alpha \cos(t + \alpha\pi/2) \quad (3-3-20)$$

在上面的定理中, $H(t-a)$ 为 Heaviside 函数, 即阶跃函数, 其含义为

$$H(t-a) = \begin{cases} 1, & t \geq a \\ 0, & \text{其他} \end{cases} \quad (3-3-21)$$

函数 $\delta(t)$ 为冲激 (impulsive) 函数, 又称 Dirac 函数, 其定义为

$$t \neq 0 \text{ 时 } \delta(t) = 0, \text{ 且 } \int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (3-3-22)$$

Dirac 函数是 Heaviside 函数的一阶导数。

此外, 引入中间变量 $r = \sqrt{\lambda^2 + \mu^2}$, $\tan \varphi = \mu/\lambda$, 还可以得出

$${}^{\text{RL}}\mathcal{D}_t^\alpha e^{\lambda t} \sin \mu t = r^\alpha e^{\lambda t} \sin(\mu t + \alpha\varphi) \quad (3-3-23)$$

$${}^{\text{RL}}\mathcal{D}_t^\alpha e^{\lambda t} \cos \mu t = r^\alpha e^{\lambda t} \cos(\mu t + \alpha\varphi) \quad (3-3-24)$$

必须指出的是, 目前只有很少几类函数的 Riemann–Liouville 分数阶导数的解析解存在。

从 Riemann–Liouville 定义来看, 因为变量 t 在积分边界与被积函数中同时出现, 所以一般函数 $f(t)$ 的分数阶导数的解析解很难求出, 只能使用数值方法求解 Riemann–Liouville 分数阶导数, 当然, 数值解的效果也不佳。

3.3.4 初始时刻平移的性质

在前面分数阶导数的叙述中, 一些公式涉及的初始时刻 $t_0 = 0$ 。如果想把这些公式的初始时刻都一般化为 t_0 , 则应该对感兴趣区间做出必要的变换。本节将引入一个定理, 允许读者做相应的变换。

重新回顾一下 Riemann–Liouville 分数阶导数的定义:

$${}^{\text{RL}}\mathcal{D}_{t_0}^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{1+\alpha-n}} d\tau \quad (3-3-25)$$

定理 3-5 ▶ 平移公式

如果已知 ${}^{\text{RL}}\mathcal{D}_t^\alpha f(t)$, 则 Riemann–Liouville 分数阶导数可以表示为

$${}^{\text{RL}}\mathcal{D}_t^\alpha f(t) = {}^{\text{RL}}\mathcal{D}_s^\alpha f(s+t_0) \quad (3-3-26)$$

其中, $s = t - t_0$ 。

证明 令 $\tau = \theta + t_0$, 则 $d\tau = d\theta$, $\theta \in [0, t - t_0]$, 代入式 (3-3-25) 的右侧, 则

$$\frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{1+\alpha-n}} d\tau = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_0^{t-t_0} \frac{f(\theta+t_0)}{(t-t_0-\theta)^{1+\alpha-n}} d\theta \quad (3-3-27)$$

再令 $s = \tau + t_0$, 则 $dt = ds$, 故 $t - t_0 = s$, 代入式 (3-3-27) 的右侧, 则

$$\frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_0^{t-t_0} \frac{f(\theta+t_0)}{(t-t_0-\theta)^{1+\alpha-n}} d\theta = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{ds^n} \int_0^s \frac{f(\theta+t_0)}{(s-\theta)^{1+\alpha-n}} d\theta \quad (3-3-28)$$

令 $F(\theta) = f(\theta + t_0)$, 对式 (3-3-28) 进行变换, 可以直接证明定理。

$$\frac{1}{\Gamma(n-\alpha)} \frac{d^n}{ds^n} \int_0^s \frac{F(\theta)}{(s-\theta)^{1+\alpha-n}} d\theta = {}^{\text{RL}}\mathcal{D}_s^\alpha F(s) = {}^{\text{RL}}\mathcal{D}_s^\alpha f(s+t_0) \quad (3-3-29)$$

例 3-8 已知指数函数的分数阶导数为 ${}^{\text{RL}}\mathcal{D}_t^\alpha e^{\lambda t} = t^{-\alpha} E_{1,1-\alpha}(\lambda t)$, 试推导下限为 t_0 的 Riemann–Liouville 分数阶导数。

解 根据定理 3-5, 记 $s = t - t_0$, 经过变量替换可以得出

$$\begin{aligned} {}^{\text{RL}}\mathcal{D}_t^\alpha e^{\lambda t} &= {}^{\text{RL}}\mathcal{D}_s^\alpha e^{\lambda(s+t_0)} = e^{\lambda t_0} {}^{\text{RL}}\mathcal{D}_s^\alpha e^{\lambda s} = e^{\lambda t_0} s^{-\alpha} E_{1,1-\alpha}(\lambda s) \\ &= e^{\lambda t_0} (t - t_0)^{-\alpha} E_{1,1-\alpha}(\lambda(t - t_0)) \end{aligned} \quad (3-3-30)$$

假设 $\lambda = -1$, $t_0 = 0.5$, $\alpha = 0.7$, 可以由下面的语句直接计算 Grünwald–Letnikov 导数, 并利用式 (3-3-30) 计算精确解, 如图 3-11 所示。可以看出, 这样绘制的两条曲线完全重合, 说明由平移公式导出的解析式是正确的。

```
>> lam=-1; t0=0.5; alfa=0.7;
    t=t0:0.01:5; u=exp(lam*t); y1=glfdiff(u,t,alfa);
    y2=exp(lam*t0)*(t-t0).^(-alfa).*ml_func([1,1-alfa],lam*(t-t0));
    plot(t,y1,t,y2,'--')
```

3.3.5 Riemann–Liouville 定义数值计算

这里给出一种 Riemann–Liouville 分数阶微积分的数值计算方法。如果计算步长为 h , 则可以由下面的步骤计算函数的 α 阶微积分。

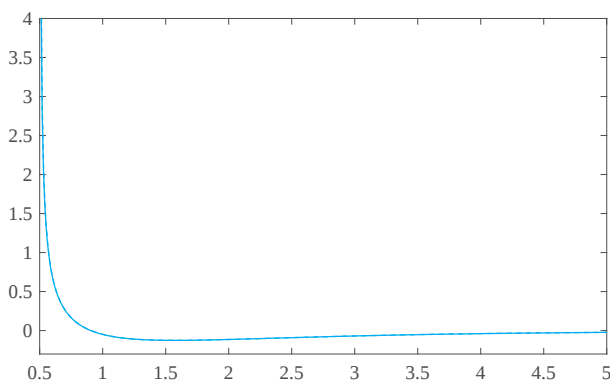


图 3-11 函数的 Grünwald-Letnikov 导数与平移公式比较

算法 3-4 ▶ Riemann-Liouville 分数阶微积分的数值算法

(1) 由已知的函数获得函数值向量 $\mathbf{f} = [f(0), f(h), f(2h), \dots]^T$ 。

(2) 令 $n = \lceil \alpha \rceil$, 计算出二项式系数 $w_i = ht_i^{n-\alpha} / \Gamma(n - \alpha + 1)$ 。

(3) 由下式计算向量 $\mathbf{s}$

$$s_j = \sum_{i=1}^j w_{j-i} f_i - \frac{1}{2} w_1 f_j - \frac{1}{2} w_j f_1 \quad (3-3-31)$$

(4) 在计算步长 h 下计算向量 $\mathbf{s}$ 的 $(n+1)$ 阶差分。

上述算法的 MATLAB 实现由下面给出。事实上在算法实现时作者加入了其他内容,使得分数阶与整数阶积分也可以直接计算出来。

```
function [dy,t]=rlfdiff(y,t,r)
arguments y(:,1), t(:,1) double, r(1,1) double, end
[y,h,n]=fdiffcom(y,t); dy=zeros(n,1); dy1=dy;
if r>-1, m=ceil(r)+1; p=m-r; y3=t.^(p-1);
elseif r== -1, yy=0.5*(y(1:n-1)+y(2:n)).*diff(t);
    for i=2:n, dy(i)=dy(i-1)+yy(i-1); end, return
else, m=-r; y3=t.^(m-1); end
for i=1:n, dy1(i)=y(i:-1:1).*(y3(1:i)); end
if r>-1, dy=diff(dy1,m)/(h^(m-1))/gamma(p); t=t(1:n-m);
else, dy=dy1*h/gamma(m); end
end
```

该函数的调用格式为 $[y_1, t] = \text{rlfdiff}(y, t, \gamma)$, 其格式类似于 $\text{glfdiff}()$ 函数, 不同的是该函数返回了两个向量 y_1 和 t 。因为该函数内部使用了 MATLAB 求差分的函数 $\text{diff}()$, 向量 y_1 的实际长度可能小于向量 y 的长度。

例 3-9 已知阶跃函数 0.75 阶导数的解析解为 $\mathcal{D}_t^{0.75} y(t) = t^{-0.75} / \Gamma(0.25)$, 试比较

Grünwald–Letnikov 与 Riemann–Liouville 分数阶导数数值解的精度。

解 选择计算步长 $h = 0.01$, 下面的语句可以计算出阶跃函数 0.75 阶导数的理论值和 Grünwald–Letnikov 与 Riemann–Liouville 分数阶导数的数值解, 如图 3-12 所示。可以看出, 直接计算的 Riemann–Liouville 分数阶导数有较大误差。在实际应用中不建议采用该函数计算 Riemann–Liouville 分数阶微积分。

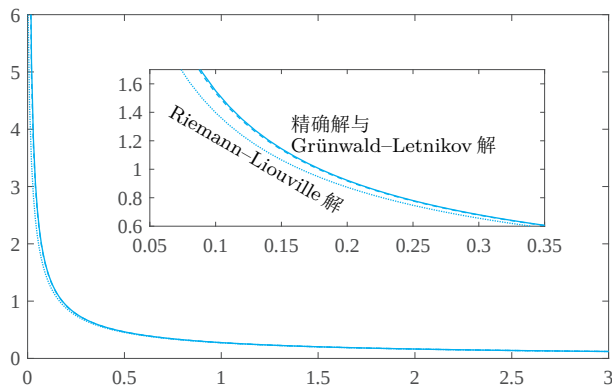


图 3-12 阶跃函数的 0.75 阶 Riemann–Liouville 导数

```
>> t0=0:0.01:3; y0=t0.^-0.75/gamma(0.25);
h=0.01; t1=0:h:3; y=ones(size(t1));
y1=glfdiff(y,t1,0.75); [y2,t2]=rlfdiff(y,t1,0.75);
plot(t0,y0,t1,y1,'--',t2,y2,':'), ylim([0 6])
```

3.3.6 Riemann–Liouville 微积分的符号计算

利用 MATLAB 的符号运算功能, 可以直接实现定义 3-4 和定义 3-5 的 Riemann–Liouville 分数阶积分与微分计算。可以编写如下的解析计算函数:

```
function dy=riemannsym(f,alpha,t0,t)
arguments f(1,1), alpha(1,1), t0(1,1)=0, t(1,1)=symvar(f), end
syms tau; alpha=sym(alpha);
switch class(f)
case 'sym', F=subs(f,t,tau);
case 'symfun', F=f(tau);
otherwise, error('f must be a symbolic expression')
end
if alpha<0, alpha=-alpha;
dy=1/gamma(alpha)*int(F/(t-tau)^(1-alpha),tau,t0,t);
elseif alpha==0, dy=f;
else, n=ceil(alpha);
f1=int(F/(t-tau)^(1+alpha-n),tau,t0,t);
```

```

dy=diff(f1,t,n)/gamma(n-alpha);
end, dy=simplify(dy);
end

```

其中, f 为符号表达式或符号函数, α 为微分的阶次, 若其值为负则求 Riemann–Liouville 积分。 t_0 为积分下限, 而 t 为符号变量。由此可以计算某些函数 Riemann–Liouville 微积分的解析解。

3.4 Caputo 分数阶微积分定义

在前面介绍的 Grünwald–Letnikov 和 Riemann–Liouville 分数阶微积分定义中, 如果初始条件非零, 有时微分方程描述可能出现困难, 所以在非零初值系统的研究与仿真中, 需要引入另一种分数阶微积分的定义——Caputo 定义^[6]。

3.4.1 Caputo 微积分定义

定义 3-6 ▶ Caputo 微分定义

Caputo 分数阶微分的定义为

$${}_t^C \mathcal{D}_t^\alpha y(t) = \frac{1}{\Gamma(m-\alpha)} \int_{t_0}^t \frac{y^{(m)}(\tau)}{(t-\tau)^{1+\alpha-m}} d\tau \quad (3-4-1)$$

式中, $m = [\alpha]$ 为整数, 且要求 $y(t)$ 及其 i 阶导数连续, $i = 1, 2, \dots, m$ ^[6]。

定义 3-7 ▶ Caputo 积分定义

Caputo 分数阶积分的定义为

$${}_t^C \mathcal{D}_t^{-\gamma} y(t) = \frac{1}{\Gamma(\gamma)} \int_{t_0}^t \frac{y(\tau)}{(t-\tau)^{1-\gamma}} d\tau \quad (3-4-2)$$

可以看出, Caputo 分数阶积分的定义与 Riemann–Liouville 定义完全一致。

3.4.2 常用的 Caputo 导数公式

作为例子, 先考虑一下指数函数 $f(t) = e^{\lambda t}$ 的 Caputo 分数阶导数问题。

定理 3-6 ▶ 指数函数的 Caputo 导数

记 $m = [\alpha]$, $\gamma = m - \alpha$, 指数函数 $f(t) = e^{\lambda t}$ 的 Caputo 分数阶导数的求取公式为

$${}_0^C \mathcal{D}_t^\alpha e^{\lambda t} = \lambda^m t^\gamma E_{1, \gamma+1}(\lambda t) \quad (3-4-3)$$

证明 因为 $\alpha > 0$, $m = [\alpha]$, $\gamma = m - \alpha$, 且指数函数的 Riemann–Liouville 分

数阶积分可以由定理 3-3 得出,该定理重新写成

$${}^{\text{RL}}_0 \mathcal{D}_t^{-\gamma} e^{\lambda t} = t^{\gamma} E_{1,1+\gamma}(\lambda t) \quad (3-4-4)$$

由于 Caputo 与 Riemann–Liouville 分数阶积分的定义是完全一致的,经过下面的推导可以发现,指数函数的 Caputo 分数阶导数满足定理的结论。

$${}_0^C \mathcal{D}_t^{\alpha} e^{\lambda t} = {}^{\text{RL}}_0 \mathcal{D}_t^{-\gamma} \frac{d^m}{dt^m} e^{\lambda t} = \lambda^m {}^{\text{RL}}_0 \mathcal{D}_t^{-\gamma} e^{\lambda t} = \lambda^m t^{\gamma} E_{1,1+\gamma}(\lambda t)$$

必须指出的是,上面第一步使用的性质是后面将介绍的定理 3-18。

定理 3-7 ▶ 常用函数的 Caputo 导数

一些常用函数的 Caputo 分数阶导数的公式^[2]为

$${}_0^C \mathcal{D}_t^{\alpha} t^{\lambda} = \begin{cases} \frac{d}{dt} t^{\lambda} & \lambda \text{ 为自然数且 } [\alpha] > \lambda \\ \frac{\Gamma(\lambda+1)}{\Gamma(\lambda+1-\alpha)} t^{\lambda-\alpha}, & \text{其他} \end{cases} \quad (3-4-5)$$

$${}_0^C \mathcal{D}_t^{\alpha} (t+c)^{\lambda} = \frac{\Gamma(\lambda+1)}{\Gamma(\lambda+1-m)} \frac{c^{\lambda-m-1} t^{\gamma}}{\Gamma(\gamma+1)} E_{1,\gamma+1}^{m-\lambda}(-t/c), \quad c > 0 \quad (3-4-6)$$

$${}_0^C \mathcal{D}_t^{\alpha} \sin \lambda t = \begin{cases} \frac{\lambda^m j (-1)^{m/2} t^{\gamma}}{\sqrt{2\Gamma(\gamma+1)}} - F(j\lambda t) + F(-j\lambda t), & m \text{ 为偶数} \\ \frac{\lambda^m (-1)^{(m-1)/2} t^{\gamma}}{\sqrt{2\Gamma(\gamma+1)}} F(j\lambda t) + F(-j\lambda t), & m \text{ 为奇数} \end{cases} \quad (3-4-7)$$

$${}_0^C \mathcal{D}_t^{\alpha} \cos \lambda t = \begin{cases} \frac{\lambda^m (-1)^{m/2} t^{\gamma}}{\sqrt{2\Gamma(\gamma+1)}} F(j\lambda t) + F(-j\lambda t), & m \text{ 为偶数} \\ \frac{\lambda^m j (-1)^{(m-1)/2} t^{\gamma}}{\sqrt{2\Gamma(\gamma+1)}} F(j\lambda t) - F(-j\lambda t), & m \text{ 为奇数} \end{cases} \quad (3-4-8)$$

式中, λ 为实数, $F(x) = {}_1F_1(1; \gamma+1; x)$, ${}_1F_1(\cdot)$ 为 2.5 节定义的超几何函数。

推论 3-1 三角函数函数的分数阶导数公式还可以写成

$${}_0^C \mathcal{D}_t^{\alpha} \sin \lambda t = \frac{(j\lambda)^m}{2j} t^{\gamma} [E_{1,\gamma+1}(j\lambda t) - (-1)^m E_{1,\gamma+1}(-j\lambda t)] \quad (3-4-9)$$

$${}_0^C \mathcal{D}_t^{\alpha} \cos \lambda t = \frac{(j\lambda)^m}{2} t^{\gamma} [E_{1,\gamma+1}(j\lambda t) + (-1)^m E_{1,\gamma+1}(-j\lambda t)] \quad (3-4-10)$$

证明 由著名的 Euler 公式就可以直接证明该推论。

$$\sin \lambda t = \frac{1}{2j} e^{j\lambda t} - e^{-j\lambda t}, \quad \cos \lambda t = \frac{1}{2} e^{j\lambda t} + e^{-j\lambda t} \quad (3-4-11)$$

令 $m=0, \gamma=-\alpha$, 可以直接用上述公式计算常用函数的 Caputo 分数阶积分。后面将介绍 Caputo 分数阶微积分的数值计算。

3.4.3 Caputo定义的符号运算

事实上,有些函数的Caputo导数是可以通过计算机解析计算的。由定义3-6,可以编写如下的MATLAB符号运算函数:

```
function dy=caputosym(f,alpha,t0,t)
    arguments f(1,1), alpha(1,1), t0(1,1)=0, t(1,1)=symvar(f), end
    if alpha<=0, dy=riemannsym(f,alpha,t0,t); return; end
    syms tau; alpha=sym(alpha); m=ceil(alpha);
    switch class(f)
        case 'sym', F=diff(subs(f,t,tau),m);
        case 'symfun', F=diff(f(tau),m);
        otherwise, error('f must be a symbolic expression')
    end
    dy=1/gamma(m-alpha)*int(F/(t-tau)^(1+alpha-m),tau,t0,t);
    dy=simplify(dy);
end
```

在函数内部先识别给定函数是符号变量还是符号函数,依据情况分别处理。

必须指出的是,由于这个函数底层使用了MuPAD提供的符号运算,很多涉及分数阶的特定问题无法通过解析积分得出;另外,由于MuPAD内核没有Mittag-Leffler函数等特殊函数,所以有时得出的结果与本书的不完全一致。这种情况下,可以通过绘制曲线对比,验证结果的正确性。

例3-10 试利用符号运算计算 ${}_0^C\mathcal{D}_t^{0.5}(2t+1)$ 和 ${}_0^C\mathcal{D}_t^{0.4}(2t+1)$ 。

解 当前版本的caputosym()函数在多项式函数求导方面只能处理整数阶多项式,不能处理伪多项式。

```
>> syms t positive, f(t)=2*t+1; %或由f=2*t+1命令定义函数
    R1=caputosym(f,sym(0.5),0,t), R2=caputosym(f,sym(0.4),0,t)
```

得出的结果为 $R_1 = 4\sqrt{t/\pi}$, $R_2 = 10t^{0.6}/(3\Gamma(0.6))$ 。

例3-11 已知指数函数 $f(t) = e^{3t}$, 试求 ${}_0^C\mathcal{D}_t^{4.3}f(t)$ 。

解 对这里给出的问题,由定理3-6可知, $m=5$, $\gamma=0.7$, $\lambda=-3$, 故 ${}_0^C\mathcal{D}_t^{4.3}f(t)$ 的解析解为 $-243t^{-0.7}E_{1,1.7}(-3t)$ 。现在尝试用caputosym()函数求其Caputo导数:

```
>> syms t positive, f=exp(-3*t); F(t)=caputosym(f,sym(4.3),0,t)
```

得出的结果极其烦琐,可以通过手工整理,得

$$F(t) = \frac{81\sqrt[10]{27}}{4\Gamma(0.7)}e^{-3t} \Gamma(0.7) - \Gamma(0.7, -3t) \quad (1 + \sqrt{5})i + \sqrt{10 - 2\sqrt{5}}$$

可见,这里的结果比前面推出的解析解复杂得多。不过,通过绘制 $t \in (0, 5)$ 区间的曲线,可见二者绘制的曲线完全重合,由此表明,符号运算得出的结果也是正确的。

```
>> t0=[0:0.01:5]'; y0=F(t0); y0=double(y0);
    y1=-243*t0.^(0.7).*ml_func([1,1.7],-3*t0);
    plot(t0,real(y0),t0,y1,'--')    %略去微小的虚部元素
```

3.5 各种不同分数阶微积分定义之间的关系

前面介绍了几种不同的分数阶微积分的定义,除了Cauchy积分公式之外,其他几个定义之间是有密切关系的。在某些条件下,有些定义是相互等效的,有些定义是可以在某些条件下相互转换的。本节将介绍不同定义之间的相互关系,并介绍Caputo定义的数值计算方法。

3.5.1 Grünwald–Letnikov与Riemann–Liouville定义的关系

文献[2]指出,对大量的函数而言,Grünwald–Letnikov定义与Riemann–Liouville定义是等效的,这里具体给出二者之间的关系。

定理 3-8 ▶ Grünwald–Letnikov的积分表示

如果函数 $f(t)$ 在 (t_0, T) 区间上是连续且 $(n-1)$ 阶可导的,且在该区间上 $f^{(n)}(t)$ 是可积的,则若 $0 \leq m-1 < \alpha < m \leq n$, Grünwald–Letnikov 分数阶导数可以写成下面的积分形式[2]:

$${}_{t_0}^{\text{GL}} \mathcal{D}_t^\alpha f(t) = \sum_{j=0}^{m-1} \frac{f^{(j)}(t_0)(t-t_0)^{j-\alpha}}{\Gamma(1+j-\alpha)} + \frac{1}{\Gamma(m-\alpha)} \int_{t_0}^t \frac{f^{(m)}(\tau) d\tau}{(t-\tau)^{1+\alpha-m}} \quad (3-5-1)$$

定理 3-9 ▶ 等效的定义

如果函数 $f(t)$ 在 (t_0, T) 上是连续且 $(n-1)$ 阶可导的,且在该区间上 $f^{(n)}(t)$ 是可积的,则若 $0 \leq m-1 < \alpha < m \leq n$, Grünwald–Letnikov 定义等效于 Riemann–Liouville 定义。

证明 定理的证明可以参见文献[2]。对式(3-5-1)右端求 m 阶导数,可以得出

$$\frac{d^m}{dt^m} {}_{t_0}^{\text{GL}} \mathcal{D}_t^\alpha f(t) = \sum_{j=0}^{m-1} \frac{f^{(j)}(t_0)(t-t_0)^{m+j-\alpha}}{\Gamma(1+m+j-\alpha)} + \frac{1}{\Gamma(2m-\alpha)} \int_{t_0}^t \frac{f^{(m)}(\tau) d\tau}{(t-\tau)^{\alpha-2m+1}} \quad (3-5-2)$$

若 $m \geq 1$, 则上式的第一项最高为 t^{m-1} , 故求 m 阶导数时该项将完全消失。使用分部积分法对上式求 m 次积分, 则

$${}_{t_0}^{\text{GL}} \mathcal{D}_t^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)} \int_{t_0}^t \frac{f^{(m)}(\tau) d\tau}{(t-t_0)^{1+\alpha-m}} = {}_{t_0}^{\text{RL}} \mathcal{D}_t^\alpha f(t) \quad (3-5-3)$$

可见,该结果与 Riemann–Liouville 定义完全一致。

3.5.2 Caputo与Riemann–Liouville定义的关系

当 $0 < \alpha < 1$, 可知 $m = 1$, 这样, 式 (3-4-1) 可以改写成

$${}^C_{t_0}\mathcal{D}_t^\alpha y(t) = \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^t \frac{y'(\tau)}{(t-\tau)^\alpha} d\tau \quad (3-5-4)$$

如果函数 $y(t)$ 的初值非零, 比较 Caputo 与 Riemann–Liouville 分数阶导数定义可得

$${}^C_{t_0}\mathcal{D}_t^\alpha y(t) = {}^{RL}_{t_0}\mathcal{D}_t^\alpha y(t) - y(t_0) \quad (3-5-5)$$

式中, 常数 $y(t_0)$ 的分数阶导数可以写成

$${}^{RL}_{t_0}\mathcal{D}_t^\alpha y(t_0) = \frac{y(t_0)}{\Gamma(1-\alpha)} (t-t_0)^{-\alpha} \quad (3-5-6)$$

可以看出, Caputo 与 Riemann–Liouville 分数阶微积分的区别为

$${}^C_{t_0}\mathcal{D}_t^\alpha y(t) = {}^{RL}_{t_0}\mathcal{D}_t^\alpha y(t) - \frac{y(t_0)}{\Gamma(1-\alpha)} (t-t_0)^{-\alpha} \quad (3-5-7)$$

更一般地, 如果阶次 α 满足 $\alpha > 0$, 则可以通过下面的定理描述 Caputo 与 Riemann–Liouville 定义之间的关系。

定理 3-10 ▶ Caputo 定义与 Riemann–Liouville 定义的关系

记 $m = [\alpha]$, 则 Caputo 与 Riemann–Liouville 定义之间的关系为

$${}^C_{t_0}\mathcal{D}_t^\alpha y(t) = {}^{RL}_{t_0}\mathcal{D}_t^\alpha y(t) - \sum_{k=0}^{m-1} \frac{y^{(k)}(t_0)}{\Gamma(k-\alpha+1)} (t-t_0)^{k-\alpha} \quad (3-5-8)$$

说明 3-3 Caputo 分数阶导数公式

(1) 可以看出, $0 \leq \alpha \leq 1$ 只是上面关系的一个特例。

(2) 正如前面介绍的, 如果 $\alpha < 0$, 则 Riemann–Liouville 分数阶积分与 Caputo 分数阶积分是完全等效的, 在实际应用中不区分二者。

3.5.3 Caputo 分数阶微分的数值计算

从 3.5.1 节的叙述可知, Caputo 与 Riemann–Liouville 或 Grünwald–Letnikov 定义的区别在于对初始值的处理, 二者之差可以由式 (3-5-8) 直接计算。这样, 以函数 `glfdiff()` 为基础就可以写出 Caputo 分数阶导数 ${}^C_{t_0}\mathcal{D}_t^\alpha y(t)$ 计算的代码。

```
function dy=caputo(y,t,gam,y0,L)
arguments, y(:,1), t(:,1), gam(1,1), y0, L(1,1)=10, end
dy=glfdiff(y,t,gam);
if gam>0, q=ceil(gam); if gam<=1, y0=y(1); end
for k=0:q-1, dy=dy-y0(k+1)*t.^(k-gam)./gamma(k+1-gam); end
```

```

yy1=interp1(t(L+1:end),dy(L+1:end),t(1:L),'spline');
dy(1:L)=yy1;
end, end

```

该函数的调用格式为 $y_1 = \text{caputo}(y, t, \alpha, y_0, L)$, 当 $\alpha \leq 0$, 则可以返回 Grünwald–Letnikov 分数阶积分, 即 Caputo 分数阶积分; 如果 $\alpha < 1$, 则 $y(t_0)$ 由向量 y 直接提取出来, 就可以计算 Caputo 分数阶导数; 如果 $\alpha > 1$, 则初值向量 y_0 实际上的内容为 $y_0 = y(t_0), y'(t_0), \dots, y^{(q-1)}(t_0)$, $q = [\alpha]$ 。

说明 3-4 Caputo 分数阶微积分

(1) 在实际计算中, 返回的 Caputo 导数前几个初始点会有较大的误差, 这样采用样条插值的方法重新计算前 L 个点的值, L 的默认值为 10, 且如果增大分数阶的阶次, 则 L 的值也应该增加。

(2) 因为在前几个点中使用样条插值计算, 所以有时得出的结果误差很大, 这个函数并不是很实用, 后面将给出更好的计算方法及其实现, 而不建议采用这里给出的算法与函数。

例 3-12 考虑例 3-4 中的正弦函数的分数阶微分问题, $f(t) = \sin(3t + 1)$ 。试求出其 0.3, 1.3 与 2.3 阶 Caputo 分数阶导数。

解 可以看出, 函数在 $t = 0$ 时刻的初值为 $\sin 1$, 这样, 两个定义之间的区别为 $d(t) = t^{-0.3} \sin 1 / \Gamma(0.7)$ 。通过下面的语句可以绘制出 Grünwald–Letnikov 与 Caputo 定义的曲线, 并绘制出 $d(t)$ 函数的曲线, 如图 3-13 所示。

```

>> t=0:0.01:pi; y=sin(3*t+1); d=t.^(-0.3)*sin(1)/gamma(0.7);
y1=glfdiff(y,t,0.3); y2=caputo(y,t,0.3,0);
plot(t,y1,t,y2,'--',t,d,':'), axis([0 pi -3 3])

```

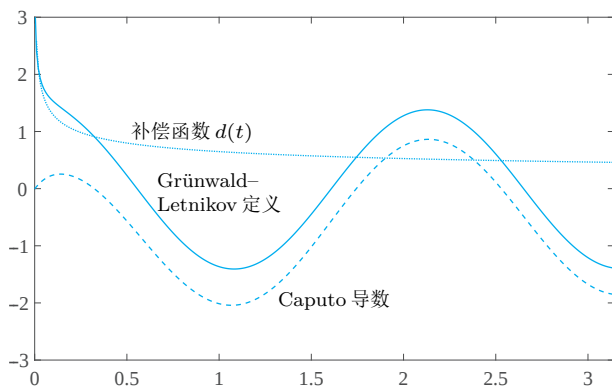


图 3-13 0.3 阶导数

因为需要 ${}_0^C \mathcal{D}_t^{2.3} y(t)$ 函数, 所以还需要的初始值是 $y'(0)$ 与 $y''(0)$, 这些值可以由原

函数通过符号运算的方式得出,结果可以变换回双精度的向量,就可以得出函数的1.3阶与2.3阶分数阶导数曲线。为了得到更好的近似效果,分数阶导数前几个点的值用样条插值方法重建,得出的曲线如图3-14所示。

```
>> syms t; y(t)=sin(3*t+1); y00=sin(1);
y1=diff(y); y10=double(y1(0)); y2=diff(y1); y20=double(y2(0));
t=0:0.01:pi; y=@(t)sin(3*t+1); y1=caputo(y,t,1.3,[y00 y10],10);
y2=caputo(y,t,2.3,[y00,y10,y20],30);
axis left, plot(t,y1), axis right, plot(t,y2,'--')
```

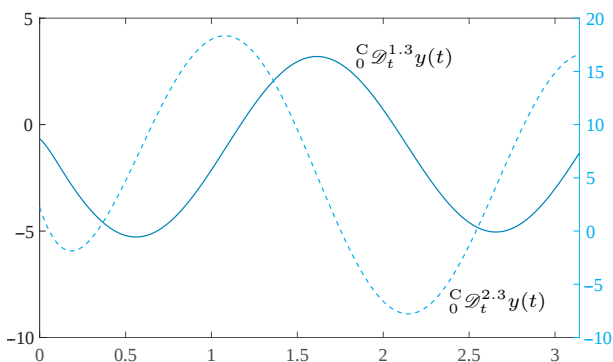


图3-14 1.3和2.3阶导数双纵轴曲线

事实上,这里给出的求解函数有两个隐患:

(1) 这里使用的插值结果引入了其他误差。

(2) 实际应用中只有 $y(t)$ 的采样点数据,不可能有函数各阶导数的初值。所以,这里给出的函数不适合 Caputo 导数计算,也不建议使用。第4章将回到这个问题重新介绍,并给出高精度的数值解。

3.6 分数阶微积分的性质与几何解释

和整数阶微积分类似,分数阶微积分也有很多自己的性质。本节首先介绍一些分数阶微积分的常用性质,然后,试图给出简单分数阶积分的物理解释与几何解释。不过,迄今为止,分数阶微积分尚没有被广泛接受的物理和几何解释。

3.6.1 分数阶微积分的性质

定理3-11 ▶ 线性性质(叠加原理)

分数阶微积分是线性的,即对任意常数 a, b , 存在

$$\mathcal{D}_t^\alpha af(t) + bg(t) = a\mathcal{D}_t^\alpha f(t) + b\mathcal{D}_t^\alpha g(t) \quad (3-6-1)$$

证明 这里先考虑证明 Grünwald–Letnikov 分数阶微积分是线性的, 记 $\tau = t - jh$, 可以通过下面的公式直接证明线性性质:

$$\begin{aligned} {}_{t_0}^{\text{GL}}\mathcal{D}_t^\alpha af(t) + bg(t) &= \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{[(t-t_0)/h]} (-1)^j \binom{\alpha}{j} af(\tau) + bg(\tau) \\ &= a \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{[(t-t_0)/h]} (-1)^j \binom{\alpha}{j} f(\tau) + b \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{[(t-t_0)/h]} (-1)^j \binom{\alpha}{j} g(\tau) \\ &= a {}_{t_0}^{\text{GL}}\mathcal{D}_t^\alpha f(t) + b {}_{t_0}^{\text{GL}}\mathcal{D}_t^\alpha g(t) \end{aligned}$$

对 Riemann–Liouville 分数阶微积分定义而言, 可以通过下面的推导证明它也是线性的:

$$\begin{aligned} {}_{t_0}^{\text{RL}}\mathcal{D}_t^\alpha af(t) + bg(t) &= \frac{1}{\Gamma(k-\alpha)} \frac{\text{d}^k}{\text{d}t^k} \int_{t_0}^t \frac{af(\tau) + bg(\tau)}{(t-\tau)^{1+\alpha-k}} \text{d}\tau \\ &= \frac{a}{\Gamma(k-\alpha)} \frac{\text{d}^k}{\text{d}t^k} \int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{1+\alpha-k}} \text{d}\tau + \frac{b}{\Gamma(k-\alpha)} \frac{\text{d}^k}{\text{d}t^k} \int_{t_0}^t \frac{g(\tau)}{(t-\tau)^{1+\alpha-k}} \text{d}\tau \\ &= a {}_{t_0}^{\text{RL}}\mathcal{D}_t^\alpha f(t) + b {}_{t_0}^{\text{RL}}\mathcal{D}_t^\alpha g(t) \end{aligned}$$

同样也可以证明 Caputo 分数阶微积分是线性的。

除了线性性质, 分数阶微积分还有其他性质, 这里将不加证明地列出一些常用的性质。关于这些性质的证明可以参见其他参考文献。

定理 3-12 ▶ 整数阶与分数阶的统一

当 $\alpha = n$ 为整数时, 分数阶微积分与整数阶微积分的结果完全一致, 且 ${}_0\mathcal{D}_t^0 f(t) = f(t)$ 。

定理 3-13 ▶ 分数阶算子的 Leibniz 性质

在传统微积分中, 两个函数 $f(t)$ 与 $g(t)$ 满足如下的性质:

$$\frac{\text{d}^n}{\text{d}t^n} f(t)g(t) = \sum_{k=0}^n \sqrt[n]{k} f^{(k)}(t)g^{(n-k)}(t) \quad (3-6-2)$$

如果 n 不是整数, 上述性质仍将满足, 证明可以参见文献 [2]。

定理 3-14 ▶ 分数阶微积分的解析性质

如果 $f(t)$ 是解析的, 则对 t 与 α 而言 ${}_{t_0}\mathcal{D}_t^\alpha f(t)$ 也是解析的。

定理 3-15 ▶ 序贯积分的交换律

分数阶的序贯积分满足交换律^[2], 且

$${}^{\text{RL}}_{t_0}\mathcal{D}_t^{-\alpha} {}^{\text{RL}}_{t_0}\mathcal{D}_t^{-\beta} f(t) = {}^{\text{RL}}_{t_0}\mathcal{D}_t^{-\beta} {}^{\text{RL}}_{t_0}\mathcal{D}_t^{-\alpha} f(t) = {}^{\text{RL}}_{t_0}\mathcal{D}_t^{-(\alpha+\beta)} f(t) \quad (3-6-3)$$

定理 3-16 ▶ 序贯导数不满足交换律

如果 $\alpha, \beta > 0$, 且函数 $f(t)$ 具有 r 阶连续导数, 其中 $r = \max([\alpha], [\beta])$, 则序贯分数阶微分可以如下计算^[2]:

$${}^{\text{RL}}_{t_0}\mathcal{D}_t^\alpha {}^{\text{RL}}_{t_0}\mathcal{D}_t^\beta f(t) = {}^{\text{RL}}_{t_0}\mathcal{D}_t^{\alpha+\beta} f(t) - \sum_{k=0}^{n-1} \frac{(t-t_0)^{k-\alpha-n}}{\sqrt[n]{(1+k-n-\alpha)}} {}^{\text{RL}}_{t_0}\mathcal{D}_t^{k+\beta-n} f(t) \Big|_{t_0} \quad (3-6-4)$$

式中, $n = [\beta]$, 且分数阶导数一般情况下不满足交换律。

推论 3-2 (序贯导数) 如果 n 为整数, 则下面的序贯分数阶微分满足:

$$\frac{d^n}{dt^n} {}^{\text{RL}}_{t_0}\mathcal{D}_t^\alpha f(t) = {}^{\text{RL}}_{t_0}\mathcal{D}_t^{n+\alpha} f(t) \quad (3-6-5)$$

$${}^{\text{RL}}_{t_0}\mathcal{D}_t^\alpha \frac{d^n}{dt^n} f(t) = {}^{\text{RL}}_{t_0}\mathcal{D}_t^{n+\alpha} f(t) - \sum_{k=0}^{n-1} \frac{(t-t_0)^{k-\alpha-n}}{\sqrt[n]{(1+k-n-\alpha)}} f^{(k)}(t_0) \quad (3-6-6)$$

定理 3-17 ▶ 零初值的序贯导数

如果函数 $f(t)$ 在 $t = t_0$ 时刻处于静止状态, 即该函数与其分数阶导数的初值均为零, 则该函数的序贯分数阶导数满足交换律, 且

$${}^{\text{RL}}_{t_0}\mathcal{D}_t^\alpha {}^{\text{RL}}_{t_0}\mathcal{D}_t^\beta f(t) = {}^{\text{RL}}_{t_0}\mathcal{D}_t^\beta {}^{\text{RL}}_{t_0}\mathcal{D}_t^\alpha f(t) = {}^{\text{RL}}_{t_0}\mathcal{D}_t^{\alpha+\beta} f(t) \quad (3-6-7)$$

定理 3-18 ▶ 建模定理(一)

如果 $n = [\gamma]$, 且 $y(t)$ 信号 n 阶可导, 则

$${}^{\text{C}}_{t_0}\mathcal{D}_t^\gamma y(t) = {}^{\text{RL}}_{t_0}\mathcal{D}_t^{\gamma-n} [y^{(n)}(t)] \quad (3-6-8)$$

证明 由定理 3-10 与推论 3-2 可以直接证明本定理。这是一个非常重要的定理, 将是后面将介绍的 Caputo 微分方程框图求解方法的理论基础之一。

3.6.2 分数阶积分的几何解释

整数阶微积分有很好的物理解释和几何解释。如果一个函数表示质点的位移, 则其一阶导数表示该质点的瞬时速度; 如果函数表示空间位置, 则其一阶导数表示

切线方向。

相比之下, 尽管分数阶微积分学也是一个有着 300 多年悠久历史的学科, 但分数阶微分与积分的定义至今尚没有被广泛接受的几何解释与物理解释。Podlubny 教授曾经给出了一个 Riemann–Liouville 分数阶积分的合理几何与物理解释^[7,8], 这里将介绍该解释。

先回顾一下 Riemann–Liouville 分数阶积分的定义

$${}^{\text{RL}}\mathcal{D}_t^{-\gamma}f(t) = \frac{1}{\Gamma(\gamma)} \int_0^t \frac{f(\tau)}{(t-\tau)^{1-\gamma}} d\tau \quad (3-6-9)$$

可以引入一个辅助函数

$$g(\tau) = \frac{1}{\Gamma(\gamma+1)} t^\gamma - (t-\tau)^\gamma \quad (3-6-10)$$

这样, 分数阶积分可以直接写成 Stieltjes 积分的形式:

$${}^{\text{RL}}\mathcal{D}_t^{-\gamma}f(t) = \int_0^t f(\tau) dg(\tau) \quad (3-6-11)$$

现在选择 3 个坐标轴 τ , $g_t(\tau)$ 和 $f(\tau)$ 建立起三维坐标系, 则函数 $f(t)$ 可以分别在 τ - $f(\tau)$ 坐标系、 $g(\tau)$ - $f(\tau)$ 坐标系和整个三维坐标系下表示出来。这样, 在 $g(\tau)$ - $f(\tau)$ 坐标系下围成的面积可以认为是 $f(t)$ 函数分数阶积分, 而 τ - $f(\tau)$ 坐标系下的面积则为该函数的一阶积分。作者编写了一个 MATLAB 函数 `fence_shadow()` 绘制三维坐标系下的图形:

```
function fence_shadow(t,f,gam,key)
arguments
    t(1,:), f, gam(1,1), key(1,1){mustBeMember(key,[1,2,3])};
end
x=t; z=f(x); tn=x(end);
switch key
case 1, y=(tn^gam-(tn-x).^gam)/gamma(1+gam);
    axis([minmax(x),minmax(y),minmax(z)]), hold on
    for i=1:length(x)-1, ii=[i,i,i+1,i+1]; x0=x(ii);
        z0=[0 z(i:i+1) 0]; y0=zeros(1,4); patch(x0,y0,z0,'c')
        y0=y(ii); patch(x0,y0,z0,'g')
        x0=zeros(1,4); patch(x0,y0,z0,'r')
    end, view(-37.5000,30)
case {2,3}, axis([minmax(x),minmax(z)]), hold on
    for i=1:length(x)-1, x1=0:0.01:x(i+1);
        tn=x1(end); y=(tn^gam-(tn-x1).^gam)/gamma(1+gam);
        if key==2, plot(x1,y);
```

```

else, z=f(x1); plot(y,z,[y(end),y(end)],[z(end),0]);
end, end, end, hold off
end

```

该函数的调用格式为 `fence_shadow(t,f,γ,key)`, 其中, t 为时间点向量, f 为函数句柄, γ 为积分阶次, key 为标示, 其取值对应文献 [7] 中的 3 种图形。

这里的函数 $g(\tau)$ 可以看作观测者自己的时间坐标系, 该坐标系是物理时间 t 扭曲的时间坐标系, 如果在观测者自己的时间坐标系 $g(t)$ 下观测到物体的运动速度为 $f(t)$, 则该观测者观测到的该物体位移就是 $f(t)$ 函数的 α 阶积分。

例 3-13 给定函数 $f(t) = t + 0.5 \sin t$, 且 $t \in (0, 10)$, 试绘制其 0.75 阶 Riemann–Liouville 积分的几何解释图。

解 根据上面叙述, 可以生成向量 t 和原函数 $f(t)$, 这样, 可以绘制出 Riemann–Liouville 分数阶积分的几何解释图, 如图 3-15 所示。可以看出, 事实上图形是由填充的颜色块构造出来的。

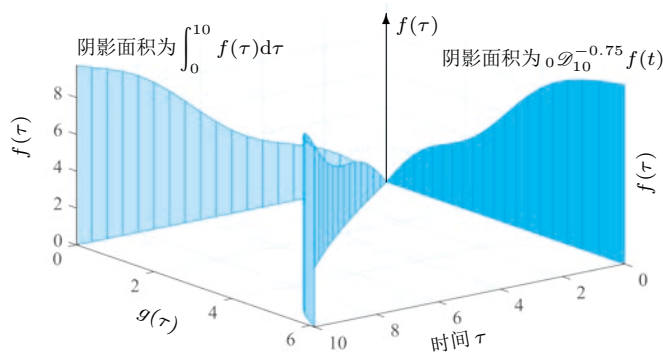


图 3-15 分数阶积分的几何解释示意图

```

>> t=0:0.5:10; f=@(t)t+sin(t)/2;
clf, fence_shadow(t,f,0.75,1), grid
set(gca,'xdir','reverse'), set(gca,'ydir','reverse')

```

可以看出, 在左侧墙上的阴影部分面积是函数 $f(t)$ 的一阶积分, 而右侧墙上的阴影部分则为所需的分数阶积分。如果选择 key 为 2 和 3, 则可以得出如图 3-16 和图 3-17 所示的其他几何解释图。

```

>> clf, fence_shadow(t,f,0.75,2)
set(gca,'xdir','reverse','ydir','reverse'), ylim([0 2*pi])
figure, fence_shadow(t,f,0.75,3), xlim([0 2*pi])

```

如果图 3-15 中的几何解释形象地将分数阶积分描述成栅栏上的影子, 则在 τ 增加时栅栏影子的长度也随之变化, 图 3-16 描述的是栅栏影子长度的变化过程, 即在底面 τ - $g_t(\tau)$ 上的投影, 而图 3-17 给出的则是在墙面 $g_t(\tau)$ - $f(\tau)$ 上投影的变化过程。

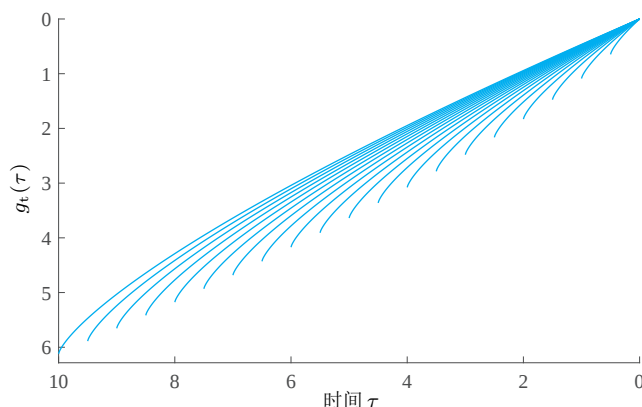


图 3-16 栅栏影子长度的变化示意图

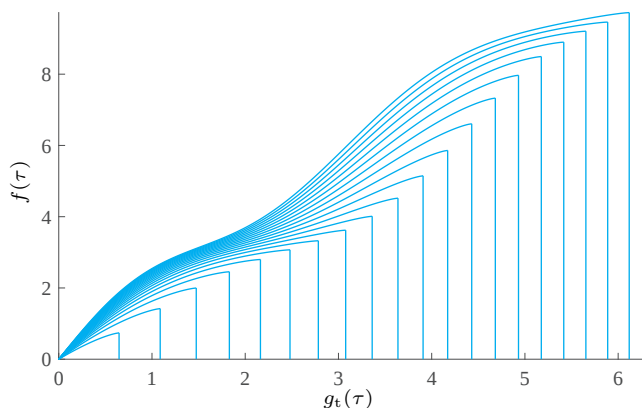


图 3-17 栅栏在墙上影子变化的示意图

尽管 Podlubny 教授的几何解释较好地描述了分数阶积分问题,但要想得出更易于理解的各种定义下分数阶微积分的几何与物理解释尚需更多努力。

本章习题

- (1) 设函数 $u(t) = C$, C 为任意常数。根据定义 3-4 和定义 3-5, 用 MATLAB 的符号运算功能计算 ${}^{\text{RL}}_0 \mathcal{D}_t^{0.5} u(t)$ 和 ${}^{\text{RL}}_0 \mathcal{D}_t^{-0.5} u(t)$ 的表达式。
- (2) 试应用数值方法验证公式

$${}_a^{\text{GL}} \mathcal{D}_t^\alpha (t-a)^v = \frac{\Gamma(v+1)}{\Gamma(v+1-\alpha)} (t-a)^{v-\alpha}$$

其中, a 为任意常数, $\alpha < 0$ 时, $v > -1$; $\alpha > 0$ 时, $v > \lfloor \alpha \rfloor$ 。

- (3) 列举一个公式 (3-3-2) 的实例, 并应用 MATLAB 的符号运算证明该实例。例如, 列举如下的实例:

$$\int_0^t \int_0^\tau \sin \tau d\tau dt = \int_0^t (t-\tau) \sin \tau d\tau$$

- (4) 试随机生成一组 λ, α 的值, 并用数值方法验证下面的等式:

$${}_0^{\text{RL}}\mathcal{D}_t^\alpha e^{\lambda t} = t^{-\alpha} E_{1,1-\alpha}(\lambda t)$$

- (5) 设函数 $u(t)$ 在定义域上连续, 则其 Riemann–Liouville 分数阶积分满足下面的关系式, 试给定一个符合条件的函数 $u(t)$, 用数值方法验证下面的等式:

$${}_0^{\text{RL}}\mathcal{D}_t^{-\alpha} {}_0^{\text{RL}}\mathcal{D}_t^{-\beta} u(t) = {}_0^{\text{RL}}\mathcal{D}_t^{-\beta} {}_0^{\text{RL}}\mathcal{D}_t^{-\alpha} u(t) = {}_0^{\text{RL}}\mathcal{D}_t^{-\alpha-\beta} u(t)$$

其中, $\alpha > 0, \beta > 0$ 。

- (6) 根据 Riemann–Liouville 分数阶导数的定义式, 函数 $u(t)$ 的 Riemann–Liouville 分数阶导数可以写成下面的形式:

$${}_0^{\text{RL}}\mathcal{D}_t^\alpha u(t) = \frac{d^n}{dt^n} {}_0^{\text{RL}}\mathcal{D}_t^{\alpha-n} u(t)$$

其中, $\alpha > 0, n = [\alpha]$ 。试列举一个函数 $u(t)$, 并用数值方法验证上面的等式。

- (7) 如果函数 $f(t)$ 具有任意阶的 Riemann–Liouville 分数阶导数和积分, 则有下面的等式^[9]:

$${}_0^{\text{RL}}\mathcal{D}_t^\alpha [tf(t)] = t {}_0^{\text{RL}}\mathcal{D}_t^\alpha f(t) + \alpha {}_0^{\text{RL}}\mathcal{D}_t^{\alpha-1} f(t)$$

其中, $\alpha > 0$ 。试列举一个满足上述条件的函数 $f(t)$, 并用数值方法验证该等式。

- (8) 试利用 MATLAB 的符号运算功能证明下面的等式^[9]:

$$\textcircled{1} \mathcal{L} {}_0^{\text{RL}}\mathcal{D}_t^{-\alpha} t^v = \frac{\Gamma(v+1)}{s^{\alpha+v+1}}, (\alpha > 0, v > -1)$$

$$\textcircled{2} \mathcal{L} {}_0^{\text{RL}}\mathcal{D}_t^{-\alpha} e^{at} = \frac{1}{s^\alpha(s-a)}, (\alpha > 0)$$

$$\textcircled{3} \mathcal{L} {}_0^{\text{RL}}\mathcal{D}_t^{-\alpha} \cos at = \frac{1}{s^\alpha(s^2+a^2)}, (\alpha > 0)$$

- (9) 根据 Caputo 分数阶导数的定义式, 函数 $u(t)$ 的 Caputo 分数阶导数可以写成下面的形式:

$${}_0^{\text{C}}\mathcal{D}_t^\alpha u(t) = {}_0^{\text{RL}}\mathcal{D}_t^{\alpha-n} u^{(n)}(t)$$

其中, $\alpha > 0, n = [\alpha]$ 。试列举一个函数 $u(t)$, 并用数值方法验证上面的等式。

- (10) 根据定理 3-6 可以得到下面的等式:

$${}_0^{\text{C}}\mathcal{D}_t^{1.3} e^{-0.5t} = (-0.5)^2 t^{0.7} E_{1,1.7}(-0.5t)$$

试用函数 `caputo()` 和第 2 章提供的函数 `ml_func()` 检验上面的等式。

- (11) 由例 3-1 可知常数的 Grünwald–Letnikov 导数非零, 其 Caputo 定义的 0.5 阶导数是什么?

- (12) 根据定理 3-9 可知, 如果函数 $f(t)$ 满足该定理的条件, 则函数 $f(t)$ 的 Grünwald–Letnikov 分数阶积分和导数与 Riemann–Liouville 分数阶积分和导数等效。试列举满足条件的函数, 并应用函数 `glfdiff()` 和 `rlfdiff()` 验证这一结论。

- (13) 根据定理 3-10 可以得到下面的等式:

$${}_0^{\text{C}}\mathcal{D}_t^{1.5} \sin t = {}_0^{\text{RL}}\mathcal{D}_t^{1.5} \sin t + \frac{1}{\Gamma(0.5)t^{0.5}}$$

应用数值算法检验上面的等式。

- (14) 应用函数 `rlfdiff()` 和 `caputo()` 计算 ${}^{\text{RL}}_0 \mathcal{D}_t^\alpha t^2$ 和 ${}_0^{\text{C}} \mathcal{D}_t^\alpha t^2$, 分别取 $\alpha = 0.5, 1, 1.5, 2, 2.5$, 比较相同阶数的 Riemann–Liouville 分数阶导数和 Caputo 分数阶导数, 得出二者之间的关系。
- (15) 根据定理 3-2 可以得到下面的等式:

$${}^{\text{RL}}_0 \mathcal{D}_t^{-0.5} t^{0.5} = \frac{\sqrt{\pi}}{2} t$$

试用函数 `rlfdiff()` 验证这一等式, 选择不同的步长计算上面的等式, 找出步长与计算误差的关系。

- (16) 证明下面的等式, 并用函数 `caputo()` 和 `rlfdiff()` 验证该等式。

$${}_0^{\text{C}} \mathcal{D}_t^{1.5} t^2 e^{-t} = {}^{\text{RL}}_0 \mathcal{D}_t^{-0.5} \sum_{k=0}^2 \frac{2!}{k!(2-k)!} \frac{d^k}{dt^k} t^2 \frac{d^{2-k}}{dt^{2-k}} e^{-t}$$

- (17) 证明下面的等式, 并用函数 `caputo()` 和 `rlfdiff()` 验证这一等式。

$${}_0^{\text{C}} \mathcal{D}_t^\alpha \sin t = {}^{\text{RL}}_0 \mathcal{D}_t^{\alpha - [\alpha]} \sin t + \frac{\pi}{2} [\alpha]$$

其中, $\alpha > 0$ 。

参考文献

- [1] Oldham K B, Spanier J. The fractional calculus: Theory and applications of differentiation and integration to arbitrary order[M]. San Diego: Academic Press, 1974.
- [2] Podlubny I. Fractional differential equations[M]. San Diego: Academic Press, 1999.
- [3] Podlubny I. Matrix approach to discrete fractional calculus[J]. Fractional Calculus and Applied Analysis, 2000, 3(4): 359–386.
- [4] Dorćák L. Numerical models for simulation the fractional-order control systems[R]. UEF-04-94. The Academy of Sciences Institute of Experimental Physics, Kosice, Slovak Republic, 1994. [2022-10-2]. <http://arxiv.org/pdf/math/0204108.pdf>.
- [5] Hilfer R. Applications of fractional calculus in physics[M]. Singapore: World Scientific, 2000.
- [6] Caputo M. Linear models of dissipation whose Q is almost frequency independent II[J]. Geophysical Journal International, 1967, 13(5): 529–539.
- [7] Podlubny I. Geometric and physical interpretations of fractional integration and differentiation[J]. Fractional Calculus and Applied Analysis, 2001, 5(4): 230–237.
- [8] Bai L, Xue D Y, Meng L. Geometric interpretation for Riemann–Liouville fractional-order integral[C]. Proceedings of the Chinese Control and Decision Conference. Hefei, China, 2020, 3225–3230.
- [9] 郭柏灵, 蒲学科, 黄凤辉. 分数阶偏微分方程及其数值解 [M]. 北京: 科学出版社, 2011.