Appendix

In this appendix, we use a specific MATLAB software, the package Chebfun, to obtain a few computational approximations for the main fractional operators in this book.

Chebfun is an open-source software package that "aims to provide numerical computing with functions" in MATLAB [2]. Chebfun overloads MATLAB's discrete operations for matrices to analogous continuous operations for functions and operators [3]. For the mathematical underpinnings of Chebfun, we refer the reader to Trefethen [3]. For the algorithmic backstory of Chebfun, we refer to Driscoll et al. [1].

In what follows, we study some computational approximations of Riemann–Liouville fractional integrals, of Caputo fractional derivatives and consequently of the combined Caputo fractional derivative, all of them with variable order. We provide, also, the necessary Chebfun code for the variable-order fractional calculus.

To implement these operators, we need two auxiliary functions: the gamma function Γ (Definition 1) and the beta function B (Definition 3). Both functions are available in MATLAB through the commands `gamma(t)` and `beta(t,u)`, respectively.

A.1 Higher-Order Riemann–Liouville Fractional Integrals

In this section, we discuss computational aspects to the higher-order Riemann–Liouville fractional integrals of variable-order ${}_aI_t^{\alpha_n(\cdot,\cdot)}x(t)$ and ${}_tI_b^{\alpha_n(\cdot,\cdot)}x(t)$.

Considering the Definition 34 of higher-order Riemann–Liouville fractional integrals, we implemented in Chebfun two functions `leftFi(x,alpha,a)` and `rightFI(x,alpha,b)` that approximate, respectively, the Riemann–Liouville fractional integrals ${}_aI_t^{\alpha_n(\cdot,\cdot)}x(t)$ and ${}_tI_b^{\alpha_n(\cdot,\cdot)}x(t)$, through the following Chebfun/MATLAB code.

R. Almeida et al., *The Variable-Order Fractional Calculus of Variations*, SpringerBriefs in Applied Sciences and Technology, https://doi.org/10.1007/978-3-319-94006-9

```
function r = leftFI(x,alpha,a)
g = @(t,tau) x(tau)./(gamma(alpha(t,tau)).*(t-tau).^(1-alpha(t,tau)));
r = @(t) sum(chebfun(@(tau) g(t,tau),[a t],'splitting','on'),[a t]);
end
```

and

```
function r = rightFI(x,alpha,b)
g = @(t,tau) x(tau)./(gamma(alpha(tau,t)).*(tau-t).^(1-alpha(tau,t)));
r = @(t) sum(chebfun(@(tau) g(t,tau),[t b],'splitting','on'),[t b]);
end
```

With these two functions, we illustrate their use in the following example, where we determine computacional approximations for Riemann–Liouville fractional integrals of a special power function.

Example 4.4 Let $\alpha(t,\tau)=\frac{t^2+\tau^2}{4}$ and $x(t)=t^2$ with $t\in[0,1]$. In this case, $a=0$, $b=1$ and $n=1$. We have ${}_aI_{0.6}^{\alpha(\cdot,\cdot)}x(0.6)\approx 0.2661$ and ${}_{0.6}I_b^{\alpha(\cdot,\cdot)}x(0.6)\approx 0.4619$, obtained in MATLAB with our Chebfun functions as follows:

```
a = 0; b = 1; n = 1;
alpha = @(t,tau) (t.^2+tau.^2)/4;
x = chebfun(@(t) t.^2, [0,1]);
LFI = leftFI(x,alpha,a);
RFI = rightFI(x,alpha,b);
LFI(0.6)
ans = 0.2661
RFI(0.6)
ans = 0.4619
```

Other values for ${}_aI_t^{\alpha(\cdot,\cdot)}x(t)$ and ${}_tI_b^{\alpha(\cdot,\cdot)}x(t)$ are plotted in Fig. A.1.

A.2 Higher-Order Caputo Fractional Derivatives

In this section, considering the Definition 36, we implement in Chebfun two new functions `leftCaputo(x,alpha,a,n)` and `rightCaputo(x,alpha, b,n)` that approximate, respectively, the higher-order Caputo fractional derivatives of variable-order ${}_a^CD_t^{\alpha_n(\cdot,\cdot)}x(t)$ and ${}_t^CD_b^{\alpha_n(\cdot,\cdot)}x(t)$.

The following code implements the left operator (4.2):

```
function r = leftCaputo(x,alpha,a,n)
dx = diff(x,n);
g = @(t,tau) dx(tau)./(gamma(n-alpha(t,tau)).
                     *(t-tau).^(1+alpha(t,tau)-n));
r = @(t) sum(chebfun(@(tau) g(t,tau),[a t],'splitting','on'),[a t]);
end
```

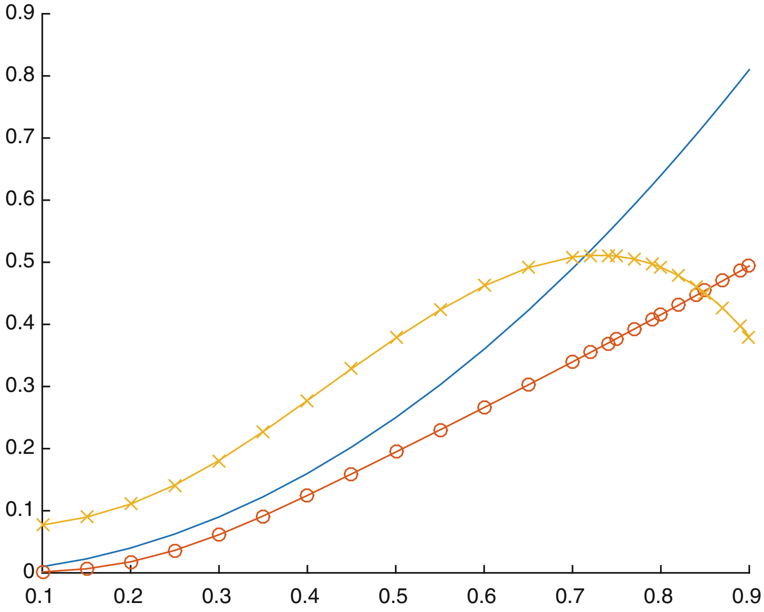

Fig. A.1 Riemann–Liouville fractional integrals of Example 4.4: $x(t) = t^2$ in continuous line, left integral ${}_aI_t^{\alpha(\cdot,\cdot)}x(t)$ with "∘−" style, and right integral ${}_tI_b^{\alpha(\cdot,\cdot)}x(t)$ with "×−" style

Similarly, we define the right operator (4.3) with Chebfun in MATLAB as follows:

```
function r = rightCaputo(x,alpha,b,n)
dx = diff(x,n);
g = @(t,tau) dx(tau)./(gamma(n-alpha(tau,t)).
                     *(tau-t).^(1+alpha(tau,t)-n));
r = @(t)(-1).^n.* sum(chebfun(@(tau) g(t,tau),[t b],
                     'splitting','on'),[t b]);
end
```

We use the two functions `leftCaputo` and `rightCaputo` to determine aproximations for the Caputo fractional derivatives of a power function of the form $x(t) = t^\gamma$.

Example 4.5 Let $\alpha(t,\tau) = \frac{t^2}{2}$ and $x(t) = t^4$ with $t \in [0, 1]$. In this case, $a = 0$, $b = 1$ and $n = 1$. We have ${}_a^CD_{0.6}^{\alpha(\cdot,\cdot)}x(0.6) \approx 0.1857$ and ${}_{0.6}^CD_b^{\alpha(\cdot,\cdot)}x(0.6) \approx -1.0385$, obtained in MATLAB with our Chebfun functions as follows:

```
a = 0; b = 1; n = 1;
alpha = @(t,tau) t.^2/2;
x = chebfun(@(t) t.^4, [a b]);
LC = leftCaputo(x,alpha,a,n);
RC = rightCaputo(x,alpha,b,n);
LC(0.6)
```

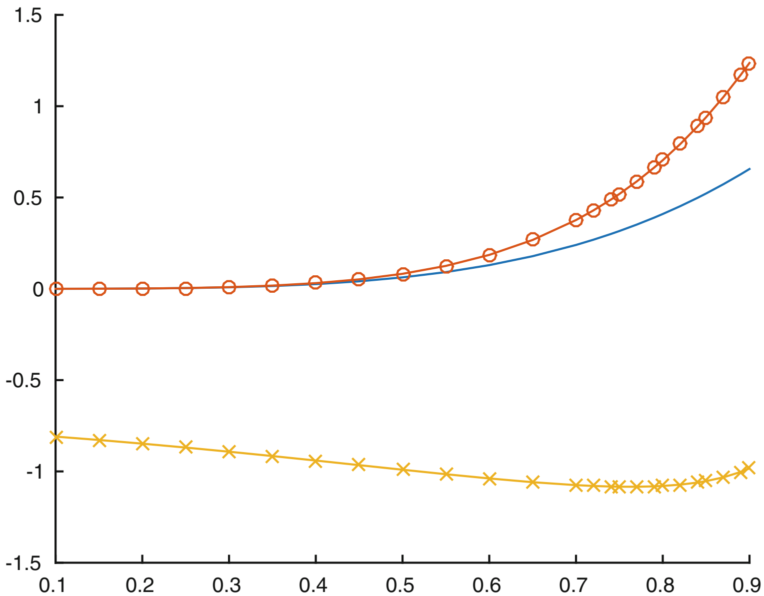

Fig. A.2 Caputo fractional derivatives of Example 4.5: $x(t) = t^4$ in continuous line, left derivative ${}^C_a D_t^{\alpha(\cdot,\cdot)} x(t)$ with "∘−" style, and right derivative ${}^C_t D_b^{\alpha(\cdot,\cdot)} x(t)$ with "×−" style

```
ans = 0.1857
RC(0.6)
ans = -1.0385
```

See Fig. A.2 for a plot with other values of ${}^C_a D_t^{\alpha(\cdot,\cdot)} x(t)$ and ${}^C_t D_b^{\alpha(\cdot,\cdot)} x(t)$.

Example 4.6 In Example 4.5, we have used the polynomial $x(t) = t^4$. It is worth mentioning that our Chebfun implementation works well for functions that are not a polynomial. For example, let $x(t) = e^t$. In this case, we just need to change

```
x = chebfun(@(t) t.^4, [a b]);
```

in Example 4.5 by

```
x = chebfun(@(t) exp(t), [a b]);
```

to obtain

```
LC(0.6)
ans = 0.9917
RC(0.6)
ans = -1.1398
```

See Fig. A.3 for a plot with other values of ${}^C_a D_t^{\alpha(\cdot,\cdot)} x(t)$ and ${}^C_t D_b^{\alpha(\cdot,\cdot)} x(t)$.

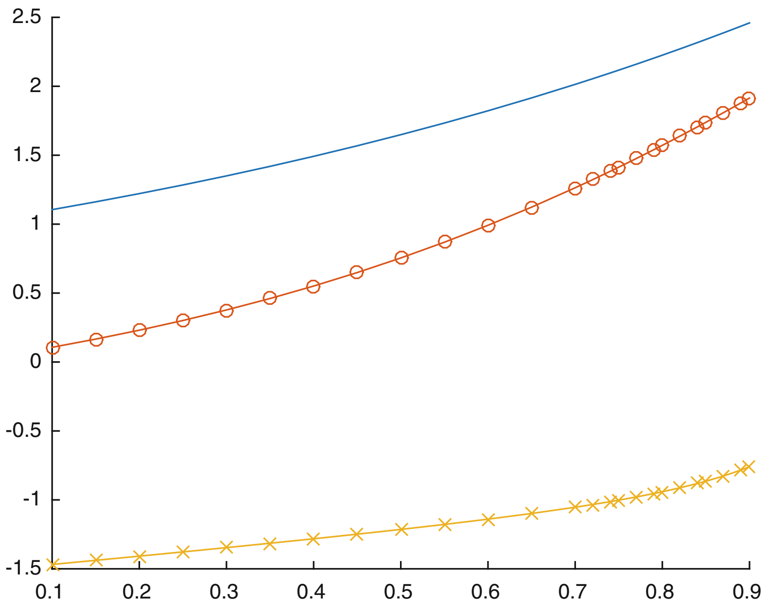

Fig. A.3 Caputo fractional derivatives of Example 4.6: $x(t) = e^t$ in continuous line, left derivative ${}_a^C D_t^{\alpha(\cdot,\cdot)} x(t)$ with "$\circ-$" style, and right derivative ${}_t^C D_b^{\alpha(\cdot,\cdot)} x(t)$ with "$\times-$" style

With Lemma 40 in Sect. 4.1, we can obtain, analytically, the higher-order left Caputo fractional derivative of a power function of the form $x(t) = (t-a)^\gamma$. This allows us to show the effectiveness of our computational approach, that is, the usefulness of polynomial interpolation in Chebyshev points in fractional calculus of variable order. In Lemma 40, we assume that the fractional order depends only on the first variable: $\alpha_n(t,\tau) := \overline{\alpha}_n(t)$, where $\overline{\alpha}_n : [a,b] \to (n-1,n)$ is a given function.

Example 4.7 Let us revisit Example 4.5 by choosing $\alpha(t,\tau) = \frac{t^2}{2}$ and $x(t) = t^4$ with $t \in [0,1]$. Table A.1 shows the approximated values obtained by our Chebfun function `leftCaputo(x,alpha,a,n)` and the exact values computed with the formula given by Lemma 40. Table A.1 was obtained using the following MATLAB code:

```
format long
a = 0; b = 1; n = 1;
alpha = @(t,tau) t.^2/2;
x = chebfun(@(t) t.^4, [a b]);
exact = @(t) (gamma(5)./gamma(5-alpha(t))).*t.^(4-alpha(t));
approximation = leftCaputo(x,alpha,a,n);
for i = 1:9
t = 0.1*i;
E = exact(t);
A = approximation(t);
```

Table A.1 Exact values obtained by Lemma 40 for functions of Example 4.7 versus computational approximations obtained using the Chebfun code

t	Exact Value	Approximation	Error
0.1	1.019223177296953e-04	1.019223177296974e-04	−2.046431600566390e-18
0.2	0.001702793965464	0.001702793965464	−2.168404344971009e-18
0.3	0.009148530806348	0.009148530806348	3.469446951953614e-18
0.4	0.031052290994593	0.031052290994592	9.089951014118469e-16
0.5	0.082132144921157	0.082132144921157	6.522560269672795e-16
0.6	0.185651036003120	0.185651036003112	7.938094626069869e-15
0.7	0.376408251363662	0.376408251357416	6.246059225389899e-12
0.8	0.704111480975332	0.704111480816562	1.587694420379648e-10
0.9	1.236753486749357	1.236753486514274	2.350835082154390e-10

```
error = E - A;
[t E A error]
end
```

Computational experiments similar to those of Example 4.7, obtained by substituting Lemma 40 by Lemma 41 and our `leftCaputo` routine by the `rightCaputo` one, reinforce the validity of our computational methods. In this case, we assume that the fractional order depends only on the second variable: $\alpha_n(\tau, t) := \overline{\alpha}_n(t)$, where $\overline{\alpha}_n : [a, b] \to (n-1, n)$ is a given function.

A.3 Higher-order Combined Fractional Caputo Derivative

The higher-order combined Caputo fractional derivative combines both left and right Caputo fractional derivatives; that is, we make use of functions `leftCaputo(x,alpha,a,n)` and `rightCaputo(x,alpha,b,n)` provided in Sect. A.1 to define Chebfun computational code for the higher-order combined fractional Caputo derivative of variable order:

```
function r = combinedCaputo(x,alpha,beta,gamma1,gamma2,a,b,n)
lc = leftCaputo(x,alpha,a,n);
rc = rightCaputo(x,beta,b,n);
r = @(t) gamma1 .* lc(t) + gamma2 .* rc(t);
end
```

Then, we illustrate the behavior of the combined Caputo fractional derivative of variable order for different values of $t \in (0, 1)$, using MATLAB.

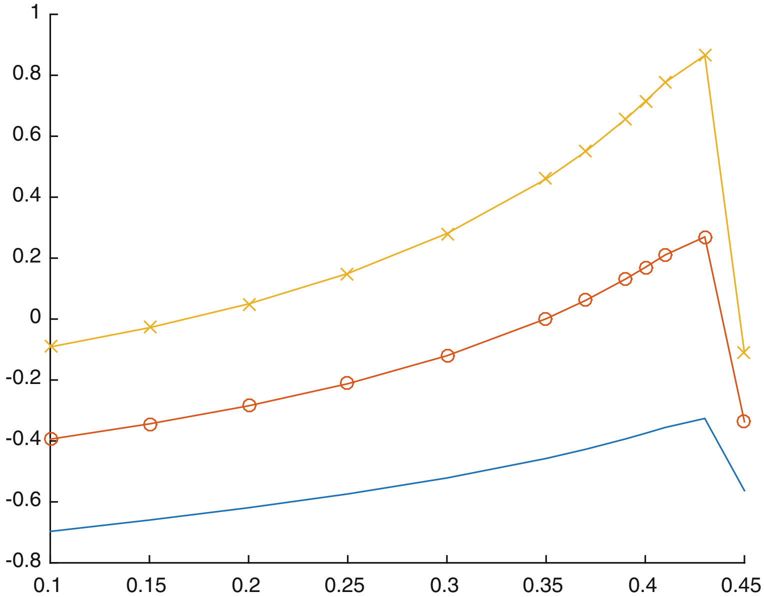

Fig. A.4 Combined Caputo fractional derivative ${}^C D_\gamma^{\alpha(\cdot,\cdot),\beta(\cdot,\cdot)} x(t)$ for $\alpha(\cdot,\cdot)$, $\beta(\cdot,\cdot)$ and $x(t)$ of Example 4.8: continuous line for $\gamma = (\gamma_1, \gamma_2) = (0.2, 0.8)$, "∘−" style for $\gamma = (\gamma_1, \gamma_2) = (0.5, 0.5)$, and "×−" style for $\gamma = (\gamma_1, \gamma_2) = (0.8, 0.2)$

Table A.2 Combined Caputo fractional derivative ${}^C D_\gamma^{\alpha(\cdot,\cdot),\beta(\cdot,\cdot)} x(t)$ for $\alpha(\cdot,\cdot)$, $\beta(\cdot,\cdot)$ and $x(t)$ of Example 4.8. Case 1: $\gamma = (\gamma_1, \gamma_2) = (0.2, 0.8)$; Case 2: $\gamma = (\gamma_1, \gamma_2) = (0.5, 0.5)$; Case 3: $\gamma = (\gamma_1, \gamma_2) = (0.8, 0.2)$

t	Case 1	Case 2	Case 3
0.4500	−0.5630	−0.3371	−0.1112
0.5000	−790.4972	−1.9752e+03	−3.1599e+03
0.5500	−3.5738e+06	−8.9345e+06	−1.4295e+07
0.6000	−2.0081e+10	−5.0201e+10	−8.0322e+10
0.6500	2.8464e+14	7.1160e+14	1.1386e+15
0.7000	4.8494e+19	1.2124e+20	1.9398e+20
0.7500	3.8006e+24	9.5015e+24	1.5202e+25
0.8000	−1.3648e+30	−3.4119e+30	−5.4591e+30
0.8500	−1.6912e+36	−4.2280e+36	−6.7648e+36
0.9000	5.5578e+41	1.3895e+42	2.2231e+42
0.9500	1.5258e+49	3.8145e+49	6.1033e+49
0.9900	1.8158e+54	4.5394e+54	7.2631e+54

Example 4.8 Let $\alpha(t,\tau) = \frac{t^2+\tau^2}{4}$, $\beta(t,\tau) = \frac{t+\tau}{3}$ and $x(t) = t$, $t \in [0,1]$. We have $a = 0$, $b = 1$ and $n = 1$. For $\gamma = (\gamma_1, \gamma_2) = (0.8, 0.2)$, we have ${}^C D_{\gamma}^{\alpha(\cdot,\cdot),\beta(\cdot,\cdot)} x(0.4)$ ≈ 0.7144:

```
a = 0; b = 1; n = 1;
alpha = @(t,tau) (t.^2 + tau.^2)/.4;
beta = @(t,tau) (t + tau)/3;
x = chebfun(@(t) t, [0 1]);
gamma1 = 0.8;
gamma2 = 0.2;
CC = combinedCaputo(x,alpha,beta,gamma1,gamma2,a,b,n);
CC(0.4)
ans = 0.7144
```

For other values of ${}^C D_{\gamma^n}^{\alpha(\cdot,\cdot),\beta(\cdot,\cdot)} x(t)$, for different values of $t \in (0,1)$ and $\gamma = (\gamma_1, \gamma_2)$, see Fig. A.4 and Table A.2.

References

1. Driscoll TA, Hale N, Trefethen LN (2014) Chebfun Guide. Pafnuty Publications, Oxford
2. Linge S, Langtangen HP (2016) Programming for computations-MATLAB/Octave. A gentle introduction to numerical simulations with MATLAB/Octave, Springer, Cham
3. Trefethen LN (2013) Approximation theory and approximation practice. Society for Industrial and Applied Mathematics

Index

R. Almeida et al., *The Variable-Order Fractional Calculus of Variations*, SpringerBriefs in Applied Sciences and Technology, https://doi.org/10.1007/978-3-319-94006-9