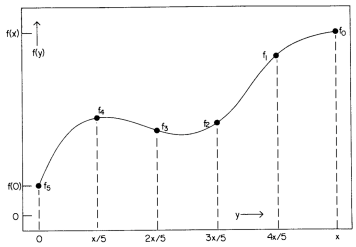
Numerical methods of the fractional calculus

Some notation

$$f_N \equiv f(0),$$
$$f_{N-1} \equiv f\left(\frac{x}{N}\right),$$
$$\vdots$$
$$f_j \equiv f\left(x - \frac{jx}{N}\right),$$
$$\vdots$$
$$f_0 \equiv f(x)$$



Notice: backwards numbering is convenient.

The G1 algorithm

Recall the Grunwald-Letnikov definition:

$$ {}_aD_t^\alpha f(t) = \lim_{h\to 0} \frac{{}_a\Delta_h^\alpha f(t)}{h^\alpha}, \qquad {}_a\Delta_h^\alpha f(t) = \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^j \binom{\alpha}{j} f(t-jh). $$

$[x]$ means the integer part of x

Omitting the limit gives the simplest approximation:

$$ {}_aD_t^\alpha f(t) \approx {}_a\Delta_h^\alpha f(t) $$

The G1 algorithm

Omitting the limit in the Grunwald-Letnikov definition gives the simplest formula for approximate calculations:

$$ {}_aD_t^\alpha f(x) \approx \left\{ \frac{\left[\frac{x-a}{N}\right]^{-q}}{\Gamma(-q)} \sum_{j=0}^{N-1} \frac{\Gamma(j-q)}{\Gamma(j+1)} f\left(x - j\left[\frac{x-a}{N}\right]\right) \right\} $$

For $a = 0$ we have:

$$ {}_0D_t^\alpha f(x) \approx \frac{x^{-q}N^q}{\Gamma(-q)} \sum_{j=0}^{N-1} \frac{\Gamma(j-q)}{\Gamma(j+1)} f_j $$

Computation of binomial coefficients

Using recurrence relationships

For the implementation we need to compute

$$ w_k^{(\alpha)} = (-1)^k \binom{\alpha}{k}, \qquad k = 0, 1, 2, \ldots $$

The recurrence relationships can be used:

$$ w_0^{(\alpha)} = 1; \quad w_k^{(\alpha)} = \left(1 - \frac{\alpha+1}{k}\right) w_{k-1}^{(\alpha)}, \quad k = 1, 2, 3, \ldots $$

Computation of binomial coefficients

Using FFT (fast Fourier transform)

The binomial coefficients can be considered as teh coefficients of the power series expansion

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

Taking $z = e^{-i\varphi}$ we obtain

$$ (1 - e^{-i\varphi})^\alpha = \sum_{k=0}^{\infty} w_k^{(\alpha)} e^{-ik\varphi} $$

Computation of binomial coefficients

Using FFT (fast Fourier transform)

We have (from the previous slide):

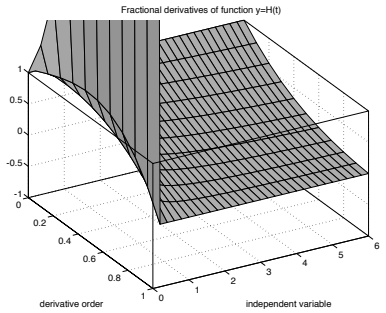
$$(1-e^{-i\varphi})^\alpha = \sum_{k=0}^{\infty} w_k^{(\alpha)} e^{-ik\varphi}$$

Therefore, the coefficients $w_k^{(\alpha)}$ can be considered as Fourier transforms:

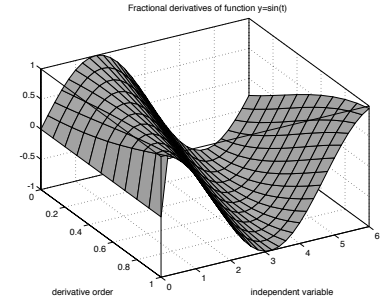
$$w_k^{(\alpha)} = \frac{1}{2\pi}\int_0^{2\pi} f_\alpha(\varphi) e^{ik\varphi} d\varphi, \qquad f_\alpha(\varphi) = (1-e^{-i\varphi})^\alpha.$$

The fast Fourier transform (FFT) can be used here!

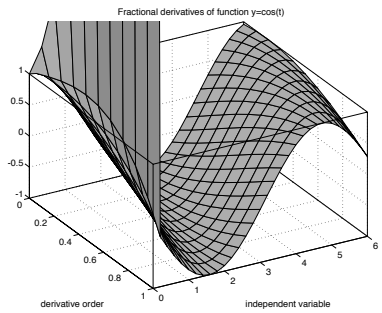
Using G1 for computations



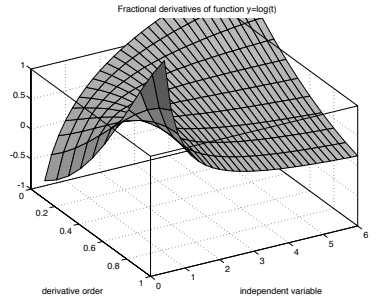
Using G1 for computations



Using G1 for computations



Using G1 for computations



Notice: infinities at t=0 are remove to make the rest of the picture visible.

Higher-order approximations

We have seen the first-order approximation:

$${}_0D_t^\alpha f(x) \approx h^{-\alpha} \sum_{k=0}^{[t/h]} w_k^{(\alpha)} f(t-kh)$$

The weights $w_k^{(\alpha)}$, $(k=0,1,2,\ldots,n,\ n=[\frac{t}{h}])$ assigned to the values $f(t\text{-}jh)$ are the first n+1 coefficients of the Taylor series expansion of the function

$$\omega_1^{(\alpha)}(z) = (1-z)^\alpha = \left(\omega_1(z)\right)^\alpha$$

Higher-order approximations

Christian Lubich's formulas:

$$\omega_2^{(\alpha)}(z) = (\tfrac{3}{2} - 2z + \tfrac{1}{2}z^2)^\alpha,$$

$$\omega_3^{(\alpha)}(z) = (\tfrac{11}{6} - 3z + \tfrac{3}{2}z^2 - \tfrac{1}{3}z^2)^\alpha,$$

$$\omega_4^{(\alpha)}(z) = (\tfrac{25}{12} - 4z + 3z^2 - \tfrac{4}{3}z^3 + \tfrac{1}{4}z^4)^\alpha,$$

$$\omega_5^{(\alpha)}(z) = (\tfrac{137}{60} - 5z + 5z^2 - \tfrac{10}{3}z^3 + \tfrac{5}{4}z^4 - \tfrac{1}{5}z^5)^\alpha,$$

$$\omega_6^{(\alpha)}(z) = (\tfrac{147}{60} - 6z + \tfrac{15}{2}z^2 - \tfrac{20}{3}z^3 + \tfrac{15}{4}z^4 - \tfrac{6}{5}z^5 + \tfrac{1}{6})^\alpha$$

Expand these functions in Taylor series and use the coefficients as weights in the formula

$${}_0D_t^\alpha f(t) \approx h^{-\alpha} \sum_{k=0}^{[t/h]} w_k^{()} f(t - kh)$$

The G2 algorithm

Oldham and Spanier (1974) observed that the approximations

$${}_aD_t^{-1} f(x) = \lim_{N\to\infty}\left\{\frac{x-a}{N}\sum_{j=0}^{N-1} f\left(x - [j+\tfrac{1}{2}]\left[\frac{x-a}{N}\right]\right)\right\}$$

$${}_aD_t^{1} f(x) = \lim_{N\to\infty}\left\{\left[\frac{x-a}{N}\right]^{-1}\sum_{j=0}^{1} [-]^j f\left(x - [j-\tfrac{1}{2}]\left[\frac{x-a}{N}\right]\right)\right\}$$

give faster convergence, and suggested "fractional central differences"

$${}_aD_t^{q} f(x) = \lim_{N\to\infty}\left\{\frac{\left[\frac{x-a}{N}\right]^{-q}}{\Gamma(-q)}\sum_{j=0}^{N-1}\frac{\Gamma(j-q)}{\Gamma(j+1)} f\left(x - [j-\tfrac{1}{2}q]\left[\frac{x-a}{N}\right]\right)\right\}$$

The G2 algorithm

Taking lower terminal $a = 0$ we have

$${}_0D_t^{q} f(x) = \lim_{N\to\infty}\left\{\frac{x^{-q}N^q}{\Gamma(-q)}\sum_{j=0}^{N-1}\frac{\Gamma(j-q)}{\Gamma(j+1)} f\left(x + \frac{qx}{2N} - \frac{jx}{N}\right)\right\}$$

This formula uses the function values other than at nodes, so we have to interpolate (Langrange three point interpolation):

$$f\left(x + \frac{qx}{2N} - \frac{jx}{N}\right) \approx \left[\frac{q}{4} + \frac{q^2}{8}\right] f\left(x + \frac{x}{N} - \frac{jx}{N}\right) + \left[1 - \frac{q^2}{4}\right] f\left(x - \frac{jx}{N}\right) + \left[\frac{q^2}{8} - \frac{q}{4}\right] f\left(x - \frac{x}{N} - \frac{jx}{N}\right)$$

This gives the G2 algorithm:

$${}_0D_t^{q} f(x) = \frac{x^{-q}N^q}{\Gamma(-q)}\sum_{j=0}^{N-1}\frac{\Gamma(j-q)}{\Gamma(j+1)}\{f_j + \tfrac{1}{4}q[f_{j-1} - f_{j+1}] + \tfrac{1}{8}q^2[f_{j-1} - 2f_j + f_{j+1}]\}$$

The R1 algorithm

G1 and G2 are based on fractional differences.
R1 and R2 are based on approximation of integration.

Take $q < 0$ (fractional integration):

$${}_0D_t^{q} f(x) = \frac{1}{\Gamma(-q)}\int_0^x \frac{f(y)\,dy}{[x-y]^{q+1}} = \frac{1}{\Gamma(-q)}\int_0^x \frac{f(x-y)\,dy}{y^{q+1}}$$

useful form

$$= \frac{1}{\Gamma(-q)}\sum_{j=0}^{N-1}\int_{jx/N}^{[jx+x]/N}\frac{f(x-y)\,dy}{y^{q+1}}$$

The R1 algorithm

Using the approximation

$$\int_{jx/N}^{[jx+x]/N}\frac{f(x-y)\,dy}{y^{q+1}} \approx \frac{f\left(x - \frac{jx}{N}\right) + f\left(x - \frac{x}{N} - \frac{jx}{N}\right)}{2}\int_{jx/N}^{[jx+x]/N}\frac{dy}{y^{q+1}}$$

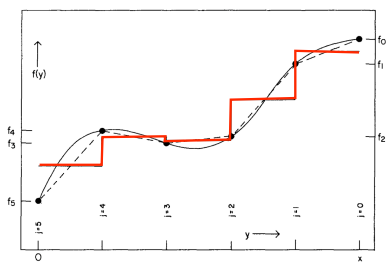
$$= \frac{f_j + f_{j+1}}{-2q}\left\{\left[\frac{jx+x}{N}\right]^{-q} - \left[\frac{jx}{N}\right]^{-q}\right\},$$

we obtain the R1 algorithm:

$${}_0D_t^{q} f(x) = \frac{x^{-q}N^q}{\Gamma(1-q)}\sum_{j=0}^{N-1}\frac{f_j + f_{j+1}}{2}\{[j+1]^{-q} - j^{-q}\} \qquad (\text{for } q < 0)$$

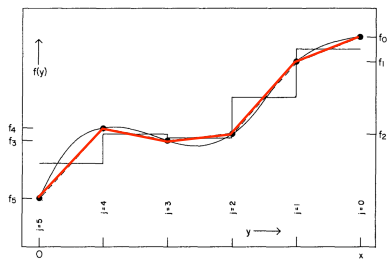
The R1 algorithm

In the previous slide we have used the piecewise constant approximation of the function using the function values in the middle of the subintervals:



The R2 algorithm

A better function approximation can be achieved using piecewise linear continuos function:



The R2 algorithm

Considering the piecewise linear approximation we have:

$$\int_{jx/N}^{[jx+x]/N} \frac{f(x-y)\,dy}{y^{q+1}} \approx \int_{jx/N}^{[jx+x]/N} \frac{\left\{\left[1+j-\frac{Ny}{x}\right]f\left(x-\frac{jx}{N}\right)+\left[\frac{Ny}{x}-j\right]f\left(x-\frac{x}{N}-\frac{jx}{N}\right)\right\}dy}{y^{q+1}}$$

and this gives the R2 algorithm:

$${}_0D_t^q f(x) = \frac{x^{-q}N^q}{\Gamma(-q)}\sum_{j=0}^{N-1}\frac{[j+1]f_j - jf_{j+1}}{-q}\{[j+1]^{-q}-j^{-q}\} + \frac{f_{j+1}-f_j}{1-q}\{[j+1]^{1-q}-j^{1-q}\}$$

(for $q<0$)

Notice: both R1 and R2 algorithms allow consideration of non-equidistant discretization nodes (derive the formulas)!

The L1 algorithm

Both R1 and R2 are restricted to negatives orders (integration only). How to extend beyond this "wall"?

Let us take $0 \leqq q < 1$. Then

Caputo

$${}_0D_t^q f(x) = \frac{x^{-q}f(0)}{\Gamma(1-q)} + \frac{1}{\Gamma(1-q)}\int_0^x \left[\frac{df}{dy}(y)\right]\frac{dy}{[x-y]^q}$$

$$= \frac{1}{\Gamma(1-q)}\left\{\frac{f(0)}{x^q} + \sum_{j=0}^{N-1}\int_{jx/N}^{[jx+x]/N}\left[\frac{df}{dy}(x-y)\right]\frac{dy}{y^q}\right\}$$

The L1 algorithm

Now let us approximate each term using

$$\int_{jx/N}^{[jx+x]/N}\left[\frac{df}{dy}(x-y)\right]\frac{dy}{y^q} \approx \frac{f\left(x-\frac{jx}{N}\right)-f\left(x-\frac{x}{N}-\frac{jx}{N}\right)}{x/N}\int_{jx/N}^{[jx+x]/N}\frac{dy}{y^q}$$

$$= \frac{x^{-q}N^q}{1-a}[f_j - f_{j+1}][(j+1)^{1-q}-j^{1-q}]$$

This leads to the LI algorithm:

$${}_0D_t^q f(x) = \frac{x^{-q}N^q}{\Gamma(2-q)}\left[\frac{[1-q]f_N}{N^q} + \sum_{j=0}^{N-1}[f_j - f_{j+1}][(j+1)^{1-q}-j^{1-q}]\right]$$

$0 \leqq q < 1$

The L2 algorithm

Similarly, taking $1 \leqq q < 2$ we can write

$${}_0D_t^q f(x) = \frac{x^{-q}f(0)}{\Gamma(1-q)} + \frac{x^{1-q}f^{(1)}(0)}{\Gamma(2-q)} + \frac{1}{\Gamma(2-q)}\int_0^x \frac{f^{(2)}(y)\,dy}{[x-y]^{q-1}}$$

$$= \frac{1}{\Gamma(2-q)}\left[\frac{[1-q]f(0)}{x^q} + \frac{f^{(1)}(0)}{x^{q-1}} + \sum_{j=0}^{N-1}\int_{jx/N}^{[jx+x]/N}\frac{f^{(2)}(x-y)\,dy}{y^{q-1}}\right]$$

We need some approximations for the first and second order derivatives here.

The L2 algorithm

Using the approximations

$$f^{(1)}(0) \approx \frac{f\left(\frac{x}{N}\right)-f(0)}{x/N} = \frac{N}{x}[f_{N-1}-f_N]$$

$$\int_{jx/N}^{[jx+x]/N} f^{(2)}(x-y)\frac{dy}{y^{q-1}} \approx \frac{f\left(x+\frac{x}{N}-\frac{jx}{N}\right)-2f\left(x-\frac{jx}{N}\right)+f\left(x-\frac{x}{N}-\frac{jx}{N}\right)}{[x/N]^2}\int_{jx/N}^{[jx+x]/N}\frac{dy}{y^{q-1}}$$

$$= \frac{x^{-q}N^q}{2-q}[f_{j-1}-2f_j+f_{j+1}][(j+1)^{2-q}-j^{2-q}]$$

we obtain the L2 algorithm

$${}_0D_t^q f(x) = \frac{x^{-q}N^q}{\Gamma(3-q)}\left[\frac{[1-q][2-q]f_N}{N^q} + \frac{[2-q][f_{N-1}-f_N]}{N^{q-1}} + \sum_{j=0}^{N-1}[f_{j-1}-2f_j+f_{j+1}][(j+1)^{2-q}-j^{2-q}]\right]$$

$1 \leqq q < 2$

The D algorithm

Kai Diethelm suggested (1997) numerical evaluation of fractional derivatives using quadrature formulas for finite-part integrals.
First, use change of variables to transform the interval [0, T] to [0,1] and introduce an equidistant grid with nodes $t_j = j/m$ .Then

$$ {}_0D^{\alpha}_{t_j} f(t) = \frac{1}{\Gamma(-\alpha)} \int_0^{t_j} \frac{f(\tau)d\tau}{(t_j-\tau)^{\alpha+1}} = \frac{t_j^{-\alpha}}{\Gamma(-\alpha)} \int_0^1 \frac{f(t_j - t_j\xi)\, d\xi}{\xi^{\alpha+1}}. $$

this singularity requires the use of finite part integrals

and the D algorithm is:

$$ {}_0D^{\alpha}_{t_j} f(t) \approx \frac{t_j^{-\alpha}}{\Gamma(-\alpha)} \sum_{k=0}^{j} \omega_{kj} f\left(\frac{k}{j}\right), \quad \omega_{kj} = \frac{j^{\alpha}}{\alpha(1-\alpha)} \begin{cases} -1, & \text{for } k=0, \\ 2k^{1-\alpha} - (k-1)^{1-\alpha} - (k+1)^{1-\alpha}, & \text{for } k=1,2,\ldots,j-1, \\ (\alpha-1)k^{-\alpha} - (k-1)^{1-\alpha} + k^{1-\alpha}, & \text{for } k=j. \end{cases} $$

$$ (j = 1, 2, \ldots, m) $$

A general framework

All considered algorithms (G1, G2, R1, R2, L1, L2, D) can be written in the same form as

$$ {}_0D^q_t f(x) = \frac{N^q}{x^q} \sum_{j=-1}^{N} w_j(q) f_j $$

G1 in brief:

Typical $w_j(q)$	$\frac{\Gamma(j-q)}{\Gamma(-q)\Gamma(j+1)}$
Range of typicality	$0 \leqq j \leqq N-1$
Values of j for which $w_j(q)=0$	$-1,\ N$
Atypical values	none

G2 in brief:

Typical $w_j(q)$	$\frac{\Gamma(j-1-q)}{\Gamma(-q)\Gamma(j+2)}\left\{j^2 - \frac{jq}{2}[q+3] - [q+1]\left[\frac{q^3}{8} + \frac{q^2}{2} - 1\right]\right\}$
Range of typicality	$0 \leqq j \leqq N-1$
Values of j for which $w_j(q)=0$	none
Atypical values	$w_{-1}(q) = \frac{q^2+2q}{8\Gamma(-q)}$ $\quad w_N(q) = \frac{\Gamma(N-1-q)}{\Gamma(-q)\Gamma(N)}\left[\frac{q^2}{8} - \frac{q}{4}\right]$

R1 in brief:

Typical $w_j(q)$	$\frac{[j+1]^{-q} - [j-1]^{-q}}{\Gamma(1-q)}$
Range of typicality	$1 \leqq j \leqq N-1$
Values of j for which $w_j(q)=0$	-1
Atypical values	$w_0(q) = \frac{1}{2\Gamma(1-q)}$ $\quad w_N(q) = \frac{N^{-q} - [N-1]^{-q}}{2\Gamma(1-q)}$

R2 in brief:

Typical $w_j(q)$	$\frac{[j+1]^{1-q} - 2j^{1-q} + [j-1]^{1-q}}{\Gamma(2-q)}$
Range of typicality	$1 \leqq j \leqq N-1$
Values of j for which $w_j(q)=0$	-1
Atypical values	$w_0(q) = 1/\Gamma(2-q)$ $w_N(q) = \frac{[N-1]^{1-q} - N^{1-q} + [1-q]N^{-q}}{\Gamma(2-q)}$

L1 in brief:

Typical $w_j(q)$	$\dfrac{[j+1]^{1-q} - 2j^{1-q} + [j-1]^{1-q}}{\Gamma(2-q)}$
Range of typicality	$1 \leqq j \leqq N-1$
Values of j for which $w_j(q) = 0$	-1
Atypical values	$w_0(q) = 1/\Gamma(2-q)$ $w_N(q) = \dfrac{[N-1]^{1-q} - N^{1-q} + [1-q]N^{-q}}{\Gamma(2-q)}$

L2 in brief:

Typical $w_j(q)$	$\dfrac{[j+2]^{2-q} - 3[j+1]^{2-q} + 3j^{2-q} - [j-1]^{2-q}}{\Gamma(3-q)}$
Range of typicality	$1 \leqq j \leqq N-2$
Values of j for which $w_j(q) = 0$	none
Atypical values	$w_{-1}(q) = 1/\Gamma(3-q)$ $w_0(q) = \dfrac{2^{2-q} - 3}{\Gamma(3-q)}$ $w_{N-1}(q) = [\Gamma(3-q)]^{-1}\{[2-q]N^{1-q} - 2N^{2-q} + 3[N-1]^{2-q} - [N-2]^{2-q}\}$ $w_N(q) = [\Gamma(3-q)]^{-1}\{[1-q][2-q]N^{-q} - [2-q]N^{1-q} + N^{2-q} - [N-1]^{2-q}\}$

Matrix approach to discretization
of
fractional integrals and derivatives

Triangular strip matrices (TSM)

Lower TSM:

$$L_N = \begin{bmatrix} \omega_0 & 0 & 0 & 0 & \cdots & 0 \\ \omega_1 & \omega_0 & 0 & 0 & \cdots & 0 \\ \omega_2 & \omega_1 & \omega_0 & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \cdots & \cdots \\ \omega_{N-1} & \ddots & \omega_2 & \omega_1 & \omega_0 & 0 \\ \omega_N & \omega_{N-1} & \ddots & \omega_2 & \omega_1 & \omega_0 \end{bmatrix},$$

Upper TSM:

$$U_N = \begin{bmatrix} \omega_0 & \omega_1 & \omega_2 & \ddots & \omega_{N-1} & \omega_N \\ 0 & \omega_0 & \omega_1 & \ddots & \ddots & \omega_{N-1} \\ 0 & 0 & \omega_0 & \ddots & \omega_2 & \ddots \\ 0 & 0 & 0 & \ddots & \omega_1 & \omega_2 \\ \cdots & \cdots & \cdots & \cdots & \omega_0 & \omega_1 \\ 0 & 0 & 0 & \cdots & 0 & \omega_0 \end{bmatrix},$$

If two TSMs are of the same type, then: $CD = DC$.

Truncation operation

$$\varrho(z) = \sum_{k=0}^{\infty} \omega_k z^k \quad \longrightarrow \quad \mathrm{trunc}_N\left(\varrho(z)\right) \stackrel{\mathrm{def}}{=} \sum_{k=0}^{N} \omega_k z^k = \varrho_N(z)$$

Function $\varrho(z)$ generates a sequence os lower TSMs:

$$L_N, \qquad N = 1, 2, \ldots$$

or upper TSMs

$$U_N, \qquad N = 1, 2, \ldots$$

Properties:

$$\mathrm{trunc}_N\left(\gamma\lambda(z)\right) = \gamma\ \mathrm{trunc}_N\left(\lambda(z)\right)$$

$$\mathrm{trunc}_N\left(\lambda(z) + \mu(z)\right) = \mathrm{trunc}_N\left(\lambda(z)\right) + \mathrm{trunc}_N\left(\mu(z)\right)$$

$$\mathrm{trunc}_N\left(\lambda(z)\mu(z)\right) = \mathrm{trunc}_N(\mathrm{trunc}_N\left(\lambda(z)\right)\ \mathrm{trunc}_N\left(\mu(z)\right))$$

Operations with TSMs

$$A_N = \sum_{k=0}^{N} a_k (E_1^-)^k = \lambda_N(E_1^-), \quad B_N = \sum_{k=0}^{N} b_k (E_1^-)^k = \mu_N(E_1^-),$$

$$\lambda_N(z) = \mathrm{trunc}_N\left(\lambda(z)\right), \qquad \mu_N = \mathrm{trunc}_N\left(\mu(z)\right)$$

Addition and subtraction:

$$A_N \pm B_N \longleftrightarrow \mathrm{trunc}_N\left(\lambda(z) \pm \mu(z)\right)$$

Multiplication by a constant:

$$\gamma A_N \longleftrightarrow \mathrm{trunc}_N\left(\gamma\lambda(z)\right)$$

Product of TSMs:

$$A_N B_N \longleftrightarrow \mathrm{trunc}_N\left(\lambda(z)\mu(z)\right)$$

Matrix inversion:

$$(A_N)^{-1} \longleftrightarrow \mathrm{trunc}_N\left(\lambda^{-1}(z)\right)$$

Left-sided R-L derivatives

$$ {}_aD_{t_k}^{\alpha} f(t) \approx \frac{\nabla^{\alpha} f(t_k)}{h^{\alpha}} = h^{-\alpha} \sum_{j=0}^{k} (-1)^j \binom{\alpha}{j} f_{k-j}, \qquad k = 0, 1, \ldots, N. $$

$$ \begin{bmatrix} h^{-\alpha}\nabla^{\alpha} f(t_0) \\ h^{-\alpha}\nabla^{\alpha} f(t_1) \\ h^{-\alpha}\nabla^{\alpha} f(t_2) \\ \vdots \\ h^{-\alpha}\nabla^{\alpha} f(t_{N-1}) \\ h^{-\alpha}\nabla^{\alpha} f(t_N) \end{bmatrix} = \frac{1}{h^{\alpha}} \begin{bmatrix} \omega_0^{(\alpha)} & 0 & 0 & 0 & \cdots & 0 \\ \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & 0 & 0 & \cdots & 0 \\ \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \cdots & \cdots \\ \omega_{N-1}^{(\alpha)} & \ddots & \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & 0 \\ \omega_N^{(\alpha)} & \omega_{N-1}^{(\alpha)} & \ddots & \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix} $$

$$ \omega_j^{(\alpha)} = (-1)^j \binom{\alpha}{j}, \qquad j = 0, 1, \ldots, N. $$

Example: Riesz kernel

$$ \frac{1}{\Gamma(1-\alpha)} \int_{-1}^{1} \frac{y(\tau)\, d\tau}{|t-\tau|^{\alpha}} = 1, \quad (-1 < t < 1), $$

Exact solution:

$$ y(t) = \pi^{-1}\Gamma(1-\alpha)\cos\left(\frac{\alpha\pi}{2}\right)(1-t^2)^{(\alpha-1)/2}. $$

Numerical solution:

$$ {}_{-1}D_t^{-(1-\alpha)} y(t) + {}_tD_1^{-(1-\alpha)} y(t) = 1, $$

$$ (B_N^{-(1-\alpha)} + F_N^{-(1-\alpha)}) Y_N = F_N $$

$$ B_N^{\alpha} = \frac{1}{h^{\alpha}} \begin{bmatrix} \omega_0^{(\alpha)} & 0 & 0 & 0 & \cdots & 0 \\ \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & 0 & 0 & \cdots & 0 \\ \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \cdots & \cdots \\ \omega_{N-1}^{(\alpha)} & \ddots & \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & 0 \\ \omega_N^{(\alpha)} & \omega_{N-1}^{(\alpha)} & \ddots & \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} \end{bmatrix} $$

$$ \beta_{\alpha}(z) = h^{-\alpha}(1-z)^{\alpha}. $$

$$ B_N^{\alpha} B_N^{\beta} = B_N^{\beta} B_N^{\alpha} = B_N^{\alpha+\beta}, $$

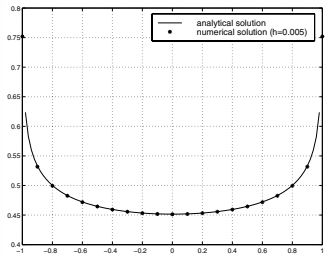
$$ {}_aD_t^{\alpha}({}_aD_t^{\beta} f(t)) = {}_aD_t^{\beta}({}_aD_t^{\alpha} f(t)) = {}_aD_t^{\alpha+\beta} f(t), $$

$$ f^{(k)}(a) = 0, \quad k = 1, 2, \ldots, r-1, $$

$$ r = \max\{n, m\} $$

Example: Riesz kernel

$\alpha = 0.8$



Left-sided R-L integration

$$ {}_aD_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau) d\tau, \qquad (a < t < b), $$

$$ I_N^{\alpha} = (B_N^{\alpha})^{-1}. $$

$$ I_N^{\alpha} \longleftrightarrow \varphi_N(z) = \mathrm{trunc}_N\left(\beta_{\alpha}^{-1}(z)\right) = \mathrm{trunc}_N\left(h^{\alpha}(1-z)^{-\alpha}\right). $$

$$ I_N^{\alpha} = h^{\alpha} \begin{bmatrix} \omega_0^{(-\alpha)} & 0 & 0 & 0 & \cdots & 0 \\ \omega_1^{(-\alpha)} & \omega_0^{(-\alpha)} & 0 & 0 & \cdots & 0 \\ \omega_2^{(-\alpha)} & \omega_1^{(-\alpha)} & \omega_0^{(-\alpha)} & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \cdots & \cdots \\ \omega_{N-1}^{(-\alpha)} & \ddots & \omega_2^{(-\alpha)} & \omega_1^{(-\alpha)} & \omega_0^{(-\alpha)} & 0 \\ \omega_N^{(-\alpha)} & \omega_{N-1}^{(-\alpha)} & \ddots & \omega_2^{(-\alpha)} & \omega_1^{(-\alpha)} & \omega_0^{(-\alpha)} \end{bmatrix} $$

Example (Caputo derivatives)

$$ y^{(\alpha)}(t) + y(t) = 1, $$
$$ y(0) = 0, \quad y'(0) = 0, $$

Exact solution:

$$ y(t) = t^{\alpha} E_{\alpha,\alpha+1}(-t^{\alpha}). $$

Numerical solution:

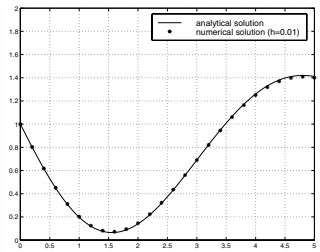
$$ \{B_{N-2}^{\alpha} + E_{N-2}\}\{S_{0,1} Y_N\} = S_{0,1} F_N. $$

and from the initial conditions we have:

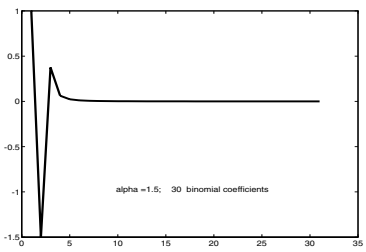
$$ y_0 = y_1 = 0 $$

Example (Caputo derivatives)

$\alpha = 1.8$



The "short memory" principle



Behavior of coefficients in GI

$${}_aD_t^\alpha f(t) \approx {}_{t-L}D_t^\alpha f(t), \qquad (t > a + L)$$

"Memory length" depends on required accuracy

The "short memory" principle

$${}_aD_t^\alpha f(t) \approx {}_{t-L}D_t^\alpha f(t), \qquad (t > a + L)$$

If $f(t) \le M$ for $a \le t \le b$, then it can be shown that

$$\Delta(t) = |\,{}_aD_t^\alpha f(t) - {}_{t-L}D_t^\alpha f(t)\,| \le \frac{ML^{-\alpha}}{|\Gamma(1-\alpha)|}, \qquad (a + L \le t \le b)$$

Therefore,

$$\Delta(t) \le \epsilon, \quad (a + L \le t \le b), \qquad \text{if} \quad L \ge \left(\frac{M}{\epsilon\,|\Gamma(1-\alpha)|}\right)^{1/\alpha}$$