The fact that time measurement as a process of counting of repeating discrete events does not really exclude inhomogeneity of time, has been nicely mentioned by L. Carroll in *Alice's Adventures in Wonderland*:

> "…I know I have to beat time when I learn music."
>
> "Ah! That accounts for it," said the Hatter. "He [Time] won't stand beating. Now, if you only kept on good terms with him, he'd do almost anything you liked to do with the clock…"

XXXVII SUMMER SCHOOL ON MATHEMATICAL PHYSICS, RAVELLO, ITALY, SEPTEMBER 17 -29, 2012

#4
Matrix approach extended:
distributed orders, non-equidistant grids, method of "large steps"

Igor Podlubny
Technical University of Kosice, Slovakia

http://www.tuke.sk/podlubny

Recall from lecture #2

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: $CD = DC$.

Left-sided fractional derivatives

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

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

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

Left-sided fractional integrals: inverse of fractional derivatives

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

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

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

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

What will change if we consider right-sided fractional-order derivatives and integrals ?

Symmetric Riesz fractional derivative

$$\frac{d^{\beta}\phi(x)}{d|x|^{\mu}} = D_R^{\beta}\phi(x) = \frac{1}{2}\Big({}_aD_x^{\beta}\phi(x) + {}_xD_b^{\beta}\phi(x)\Big)$$

Riemann-Liouville

RIESZ POTENTIAL OPERATORS AND INVERSES VIA FRACTIONAL CENTRED DERIVATIVES

MANUEL DUARTE ORTIGUEIRA

Received 2 January 2006; Revised 4 May 2006; Accepted 7 May 2006

Hindawi Publishing Corporation
International Journal of Mathematics and Mathematical Sciences
Volume 2006, Article ID 48391, Pages 1–12
DOI 10.1155/IJMMS/2006/48391

$$\frac{\partial^{\alpha}}{\partial|x|^{\alpha}}u(x,t) = -c_{\alpha}({}_0D_x^{\alpha} +_x D_L^{\alpha})u(x,t),$$

$$c_{\alpha} = \frac{1}{2\cos(\frac{\pi\alpha}{2})},$$

Discretization of symmetric Riesz fractional derivative

$$\left[v_m^{(\beta)}\ \ v_{m-1}^{(\beta)}\ \ \ldots\ v_1^{(\beta)}\ \ v_0^{(\beta)}\right]^T = R_m^{(\beta)}\left[v_m\ \ v_{m-1}\ \ \ldots\ v_1\ \ v_0\right]^T$$

(1) Using the left and right sided R-L derivatives:

$$R_m^{(\beta)} = \frac{h^{-\alpha}}{2}\Big[\,{}_{-1}U_m + {}_{+1}U_m\Big]$$

(2) Using Ortigueira's centred differences:

$$R_m^{(\beta)} = h^{-\beta}\begin{bmatrix} \omega_0^{(\beta)} & \omega_1^{(\beta)} & \omega_2^{(\beta)} & \omega_3^{(\beta)} & \cdots & \omega_m^{(\beta)} \\ \omega_1^{(\beta)} & \omega_0^{(\beta)} & \omega_1^{(\beta)} & \omega_2^{(\beta)} & \cdots & \omega_{m-1}^{(\beta)} \\ \omega_2^{(\beta)} & \omega_1^{(\beta)} & \omega_0^{(\beta)} & \omega_1^{(\beta)} & \cdots & \omega_{m-2}^{(\beta)} \\ \ddots & \ddots & \ddots & \ddots & \cdots & \cdots \\ \omega_{m-1}^{(\beta)} & \ddots & \omega_2^{(\beta)} & \omega_1^{(\beta)} & \omega_0^{(\beta)} & \omega_1^{(\beta)} \\ \omega_m^{(\beta)} & \omega_{m-1}^{(\beta)} & \ddots & \omega_2^{(\beta)} & \omega_1^{(\beta)} & \omega_0^{(\beta)} \end{bmatrix}$$

$$\omega_k^{(\beta)} = \frac{(-1)^k\,\Gamma(\beta+1)\,\cos(\beta\pi/2)}{\Gamma(\beta/2-k+1)\,\Gamma(\beta/2+k+1)}$$

Time- and Space- Fractional Diffusion Equation

$$\frac{\partial u}{\partial t} = a^2\frac{\partial^{\beta}u}{\partial|x|^{\beta}}$$

$${}_0^CD_t^{\alpha}y = a^2\frac{\partial^{\beta}y}{\partial|x|^{\beta}}$$

Symmetric Riesz fractional derivative:

$$\frac{d^{\beta}\phi(x)}{d|x|^{\beta}} = D_R^{\beta}\phi(x) = \frac{1}{2}\Big({}_aD_x^{\beta}\phi(x) + {}_xD_b^{\beta}\phi(x)\Big)$$

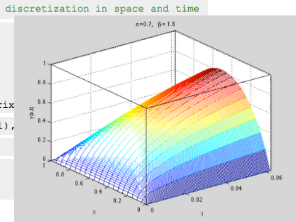
Time- and Space- Fractional Diffusion Equation

$${}_0^CD_t^{\alpha}y - \frac{\partial^{\beta}y}{\partial|x|^{\beta}} = f(x,t)$$

$$y(0,t)=0,\quad y(1,t)=0;$$

$$y(x,0)=0.$$

```
alpha = 0.7; beta = 1.8;

a2=1;        % coefficient from the diffusion equation
L = 1;       % length of spatial interval
m = 21;      % Number of spatial steps of discretization
n =148;      % Number of steps in time
h = L / (m-1);          % spatial step
tau = h^2 / (6*a2);     % time step

B1 = ban(alpha,n-1,tau)';     % alpha-th order derivative with respect to time
TD = kron(B1, eye(m));        % time derivative matrix

B2 = ransym(beta,m,h);        % beta-th order derivative with respect to X
SD = kron(eye(n-1), B2);      % spatial derivative matrix

SystemMatrix = TD - a2*SD;   % matrix corresponding to discretization in space and time

S = eliminator (m, [1 m]);
SK = kron(eye(n-1), S);
SystemMatrix_without_columns_1_m = SystemMatrix * SK';

S = eliminator (m, [1 m]);
SK = kron(eye(n-1), S);
SystemMatrix_without_rows_columns_1_m = SK * SystemMatrix

F = 8*ones(size(SystemMatrix_without_rows_columns_1_m,1),

Y = SystemMatrix_without_rows_columns_1_m\F;

YS = reshape(Y,m-2,n-1);
YS = fliplr(YS);
U = YS;
```

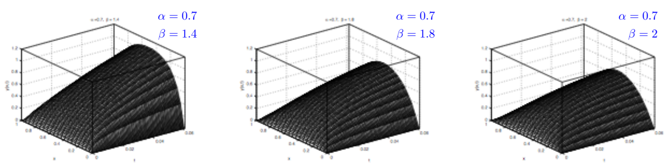


Example: Time-space fractional diffusion equation

$${}_0^CD_t^{\alpha}y - \frac{\partial^{\beta}y}{\partial|x|^{\beta}} = f(x,t), \qquad (\text{with } f(x,t)\equiv 8)$$

$$y(0,t)=0,\quad y(1,t)=0; \qquad y(x,0)=0.$$

$$\left\{B_n^{(\alpha)}\otimes E_m - E_n\otimes R_m^{(\beta)}\right\}y_{nm} = f_{nm}$$



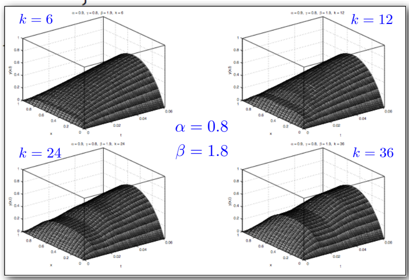
Example: Time-space fractional diffusion equation with delayed fractional derivative

$$\frac{1}{2}\left\{ {}_0^C D_t^{\alpha} y + {}_0^C D_{t-\delta}^{\gamma} y \right\} - \frac{\partial^{\beta} y}{\partial |x|^{\beta}} = f(x,t) \qquad (\text{with } f(x,t) \equiv 8)$$

$$y(0,t) = 0, \quad y(1,t) = 0 \quad y(x,0) = 0$$

$$\left\{ \frac{1}{2} \left(B_n^{(\alpha)} \otimes E_m + {}_{+k}B_n^{(\gamma)} \otimes E_m \right) - E_n \otimes R_m^{(\beta)} \right\} y_{nm} = f_{nm}$$

$${}_{+k}B_n^{(\gamma)} = S_{n+1,\dots,n+k} \, E_{n+k,k}^{+}$$

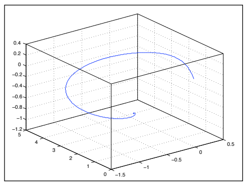


Historically the first example of numerical solution of fractional differential equations with delayed fractional derivatives

Systems of linear fractional ODEs? – YES!

```
B = [0.9 0.8 0.8]';

A = [-2.0000 -1.0000 -1.9000; ...
     -3.0000 -2.0000 -1.0000; ...
     -1.0000 -1.0000 -1.0000];

C = [1 5 2]';

timestep = 0.01; Steps = 2000;

Y = linsfdes(B, A, C, timestep, Steps);

plot3(Y(:,1), Y(:,2), Y(:,3))
```

$$x_1^{(\alpha)}(t) = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + c_1$$
$$x_2^{(\beta)}(t) = a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + c_2$$
$$x_3^{(\gamma)}(t) = a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + c_3$$
$$x_1(0) = x_2(0) = x_3(0) = 0.$$



Inside the function `linsfdes.m`:

```
for p = 1:k
 D(((p-1)*N+1):(p*N),((p-1)*N+1):(p*N)) ...
                                = ban(B(p),N,h);
end

AK = kron(A,eye(N));
```

Matrix approach and "short memory"

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

L

memory length

DO-fractional derivatives

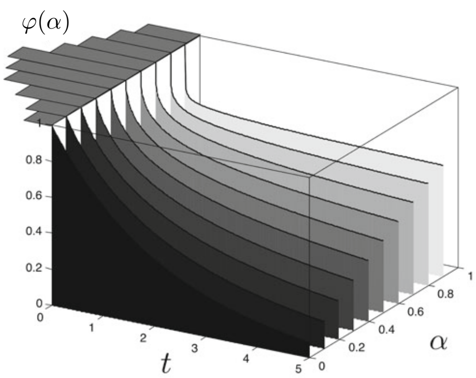
Left-sided $${}_aD_t^{\varphi(\alpha)} f(t) = \int_c^d \varphi(\alpha) \, {}_aD_t^{\alpha} f(t) d\alpha$$

Right-sided $${}_tD_b^{\varphi(\alpha)} f(t) = \int_c^d \varphi(\alpha) \, {}_tD_b^{\alpha} f(t) d\alpha$$

Symmetric $${}_aR_b^{\varphi(\alpha)} f(t) = \int_c^d \varphi(\alpha) \, {}_aR_b^{\alpha} f(t) d\alpha$$

Restriction: $$\int_c^d \varphi(\alpha) \, d\alpha = 1$$

Interpretation of DO operators

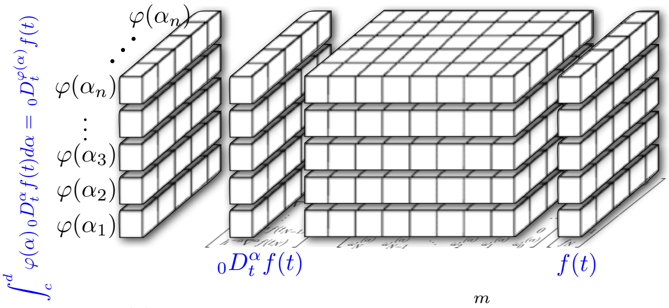


Discretization of DO-FDs:

a piece of cake!

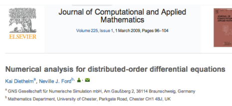
Discretization of DO-FDs:

a piece of cake!



$$ {}_0D_t^{\varphi(\alpha)} f(t) \approx B_{n,m}^{\varphi(\alpha)} f_n, \qquad B_{n,m}^{\varphi(\alpha)} = \sum_{k=1}^{m} B_n^{\alpha_k} \varphi(\alpha_k) \Delta\alpha_k $$

Example 0: DO-Diethelm-Ford equation(s)

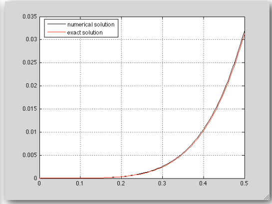


Example 5.1 *The equation*

$$ \int_0^2 \frac{\Gamma(6-r)}{120} D_*^r u(t)\,dr = \frac{t^5 - t^3}{\log t} \qquad (26) $$

with initial conditions $u(0) = u'(0) = 0$ *has the unique solution* $u(t) = t^5$.

```
h = 0.005;            % step of discretization
t = 0:h:0.5;          %
N = length(t) + 1;  % number of nodes
f = '(t.^5-t.^3)./log(t)';       % RHS, as in DF-paper, example 1
M = zeros(N,N);      % pre-allocate matrix M for the system
M = doban('gamma(6-alf)/120', [0 2], 0.01, N-1, h);

% Then we compute the right-hand side at discretization nodes:
F = eval (f, t)';

% Utilize zero initial condition:
M = eliminator(N-1,[1 2])*M*eliminator(N-1, [1 2])';
F = eliminator(N-1,[1 2])*F;

% And solve the system MY=F:
Y = M\F;

% Pre-pend the zero initial value
% (that one due to zero initial condition)
Y0 = [0; 0; Y];

% Plot the numerical solution:
plot(t, Y0, 'k')
```



Error at u(0.5) = -7.6896e-04

Example 1: DO-fractional relaxation equation

$$ {}_0D_t^{\varphi(\alpha)} x(t) + Bx(t) = f(t), \qquad \varphi(\alpha) = 6\,\alpha\,(1-\alpha), $$
$$ x(0) = 1 \qquad 0 \le \alpha \le 1 $$

$$ x(t) = u(t) + 1 $$

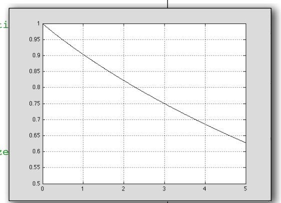
$$ {}_0D_t^{\varphi(\alpha)} u(t) + Bu(t) = f(t) - B, $$
$$ u(t) = 0. $$

Recalling lecture #3

Example #3-1: Fractional relaxation equation

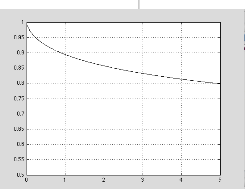
$$ {}_0D_t^{\alpha} x(t) + Bx(t) = 0, \quad (0 < \alpha \le 1), $$
$$ x(0) = 1, $$

```
h = 0.01;             % step of discretization
t = 0:h:5;            % as in [DOFDS-paper, caption to Fig.6]
N = length(t) + 1;    % number of nodes
B = 0.1;              % coefficient of the equation
                      % as in [DOFDS-paper, caption to Fig.6]
f = '0 + 0*t';        % RHS, as in [DOFDS-paper, caption to Fig.6]
M = zeros(N,N);       % pre-allocate matrix M for the system

alpha=exp(-0.01*5);  % beta = 0.9512,  order of equation

% First, we make the matrix for the entire equation -- this is really easy:
M = ban(alpha, N-1, h) + B*eye(N-1,N-1);

% Then we compute the right-hand side at discretizati
F = eval ([f '-B'], t)';

% Utilize zero initial condition:
M = eliminator(N-1,[1])*M*eliminator(N-1, [1])';
F = eliminator(N-1,[1])*F;

% And solve the system MY=F:
Y = M\F;

% Pre-pend the zero initial value (that one due to ze
Y0 = [0; Y];

% Plot the solution:
plot(t, Y0+1, 'g')
```



Example 1: DO-fractional relaxation equation

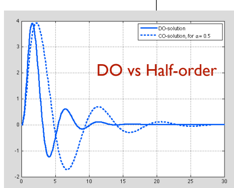
```
h = 0.01;             % step of discretization
t = 0:h:5;            % as in [DOFDS-paper, caption to Fig.6]
N = length(t) + 1;    % number of nodes
B = 0.1;              % coefficient of the equation
                      % as in [DOFDS-paper, caption to Fig.6]
f = '0 + 0*t';        % RHS, as in [DOFDS-paper, caption to Fig.6]
M = zeros(N,N);       % pre-allocate matrix M for the system

% First, we make the matrix for the entire equation -- this is really easy:
M = doban('6*alf.*(1-alf)', [0 1], 0.01, N-1, h) + B*eye(N-1,N-1);

% Then we compute the right-hand side at discretization nodes:
F = eval ([f '-B'], t)';

% Utilize zero initial condition:
M = eliminator(N-1,[1])*M*eliminator(N-1, [1])';
F = eliminator(N-1,[1])*F;

% And solve the system MY=F:
Y = M\F;

% Pre-pend the zero initial value
% (that one due to zero initial condition)
Y0 = [0; Y];

% Plot the solution:
U = Y0 + 1;
plot(t, U, 'k')
```



Example 2: Bagley-Torvik equation

$$ Ay''(t) + By^{(\varphi(\alpha))}(t) + Cy(t) = F(t), \qquad F(t) = \begin{cases} 8, & (0 \le t \le 1) \\ 0, & (t > 1) \end{cases} $$
$$ y(0) = y'(0) = 0 $$

```
% (1) Prepare constants and nodes (this
script):
alpha = 1.5;
A = 1; B = 1; C = 1;  % coefficients of
h = 0.075;            % step of discreti
T = 0:h:30;           % nodes
N = 30/h + 1;         % number of nodes
M = zeros(N,N);       % pre-allocate matrix M for the system

% (2) Make the matrix for the entire equation -- this is really easy:
M = A*ban(2,N,h) + B*doban('6*alf.*(1-alf)', [0 1], 0.01, N, h) + C*eye
(N,N);

% (3) Make right-hand side:
F = 8*(T<=1)';

% (4) Utilize zero initial conditions:
M = eliminator(N,[1 2])*M*eliminator(N, [1 2])';
F = eliminator(N,[1 2])*F;

% (5) Solve the system MY=F:
Y = M\F;

% (6) Pre-pend the zero values (those due
% to zero initial conditions)
Y0 = [0; 0; Y];

% Plot the solution:
plot(T,Y0)
```

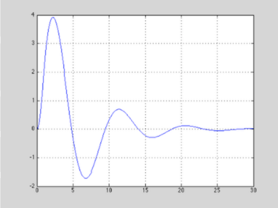


Recalling lecture #3

Example #3-5: Bagley-Torvik equation

$$Ay''(t) + By^{3/2}(t) + Cy(t) = F(t), \quad F(t) = \begin{cases} 8, & (0 \le t \le 1) \\ 0, & (t > 1) \end{cases} \qquad y(0) = y'(0) = 0$$

```
% (1) Prepare constants and nodes (this is the longest part of the script):
alpha = 1.5;
A = 1; B = 1; C = 1;  % coefficients of the Bagley-Torvik equation
h = 0.075;            % step of discretization
T = 0:h:30;           % nodes
N = 30/h + 1;         % number of nodes
M = zeros(N,N);       % pre-allocate matrix M for the system

% (2) Make the matrix for the entire equation -- this is really easy:
M = A*ban(2,N,h) + B*ban(alpha,N,h) + C*eye(N,N);

% (3) Make right-hand side:
F = 8*(T<=1)';

% (4) Utilize zero initial conditions:
M = eliminator(N,[1 2])*M*eliminator(N, [1 2])';
F = eliminator(N,[1 2])*F;

% (5) Solve the system MY=F:
Y = M\F;

% (6) Pre-pend the zero values (those due to zero
Y0 = [0; 0; Y];

% Plot the solution:
plot(T,Y0)
```



Example 3: DO-diffusion-wave equation

$$ {}_0^C D_t^{\varphi(\alpha)} y - \frac{\partial^\beta y}{\partial |x|^\beta} = f(x,t)$$

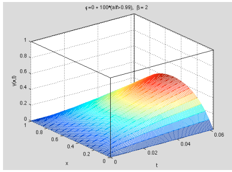
$$y(0,t) = 0, \quad y(1,t) = 0; \qquad y(x,0) = 0.$$

DO → CO

$\varphi(\alpha) = \delta(\alpha - \lambda)$ → ${}_0D_t^\lambda$

Q: How would you implement Dirac's delta function in MATLAB?

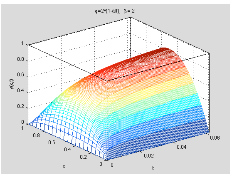
$\varphi(\alpha) = 1$



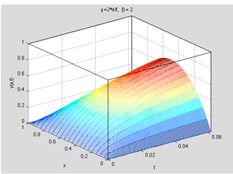
MATLAB: '0 + 100*(alf>0.99)'

Example 3: DO-diffusion-wave equation

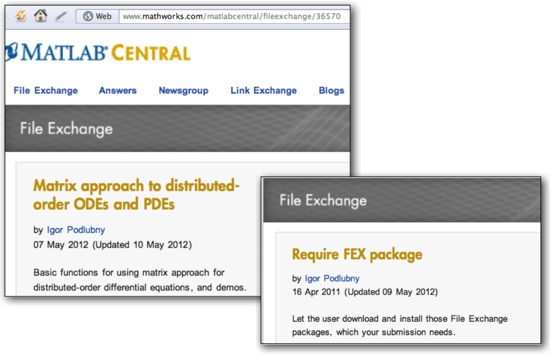
$\varphi(\alpha) = 2\,(1 - \alpha)$



$\varphi(\alpha) = 2\,\alpha$



Matrix approach toolbox for distributed orders



Non-equidistant grids

Non-equidistant grid: discretization is cumbersome even in simplest cases...

$$w''(x): \quad \frac{1}{\hat{h}_i}\left(\frac{w_{i+1} - w_i}{h_i} - \frac{w_i - w_{i-1}}{h_{i-1}}\right) \qquad h_i = x_{i+1} - x_i, \quad \hat{h}_i = \hat{x}_i - \hat{x}_{i-1},$$

The R1 and R2 algorithms

R1 and R2 are based on approximation of integration.

In such a case, the grid can be non-uniform.

Recall that $I_N^\alpha = \left(B_N^\alpha\right)^{-1}$

Change the viewpoint:

Left-sided fractional derivatives: inverse of fractional integrals; then

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

Any approximation of fractional integration after inversion gives an approximation for fractional differentiation on the same grid!

The simplest approach: approximation of a function under integration by a piecewise constant function

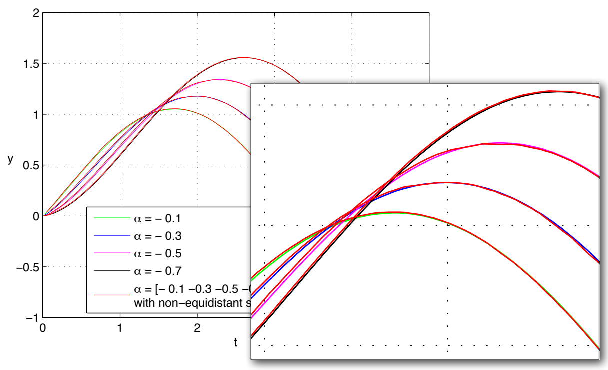
$$B_N^\alpha = (I_N^\alpha)^{-1}$$

Coefficients of I_N^α

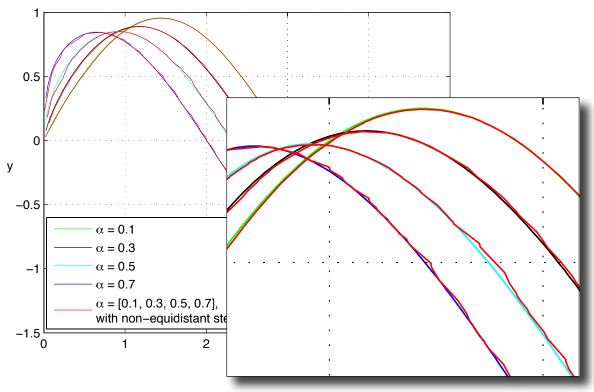
$$I_{k,j} = \frac{(t_k - t_{j-1})^\alpha - (t_k - t_j)^\alpha}{\Gamma(\alpha+1)},$$
$$j = 1, \ldots, k; \quad k = 1, \ldots, N.$$

For non-equidistant grids, the matrix is not a TSM .

Example 1: fractional integrals of $\sin(x)$



Example 2: fractional derivatives of $\sin(x)$



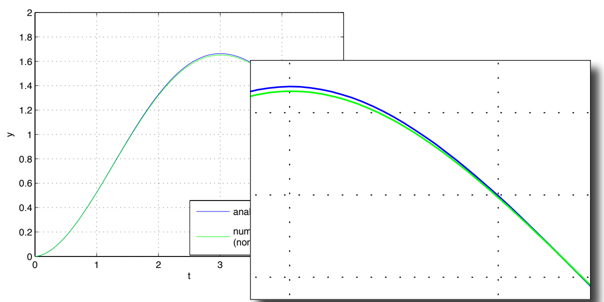
Example 3: two-term ordinary FDE

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

$y(0) = 0, \quad y'(0) = 0.$

Exact solution:

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

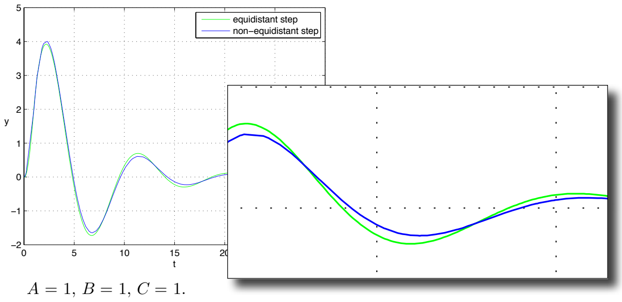


$\alpha = 1.8$, number of discretization nodes $N = 500$

Example 4(a): Bagley-Torvik equation

$$Ay''(t) + By^{3/2}(t) + Cy(t) = F(t),$$

$$F(t) = \begin{cases} 8, & (0 \le t \le 1) \\ 0, & (t > 1) \end{cases}, \quad y(0) = y'(0) = 0.$$

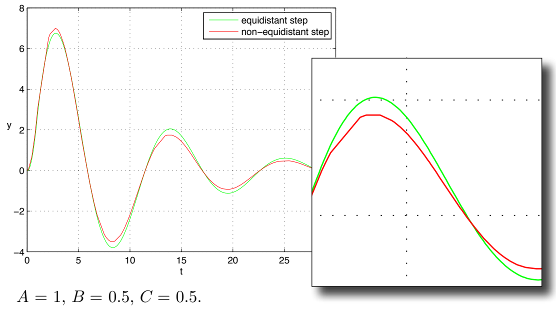


$A = 1,\ B = 1,\ C = 1.$

Example 4(b): Bagley-Torvik equation

$$Ay''(t) + By^{3/2}(t) + Cy(t) = F(t),$$

$$F(t) = \begin{cases} 8, & (0 \le t \le 1) \\ 0, & (t > 1) \end{cases}, \quad y(0) = y'(0) = 0.$$



$A = 1,\ B = 0.5,\ C = 0.5.$

Can we have variable step length?

As seen in MATLAB: **ode23.m** and **ode45.m** solvers

- optimized for a variable step
- using a variable step ensures that a large step size is used for low frequencies and a small step size is used for high frequencies
- make a step, estimate the error at this step, check if the value is greater than or less than the tolerance, and adapt the step size accordingly

There was nothing like this available for FDEs – so far...

Method of "large steps"

Method of "large steps"

$${}_0D_t^\alpha y(t) = f(y(t), t), \quad (t > 0),$$
$$y(0) = 0,$$

Suppose we obtained its solution in the interval (0,a) (and the final value y_a at $t = a$), then we can use this for transforming the above problem to

$${}_aD_t^\alpha y(t) = f(y(t), t) - {}_0R_a^\alpha y(t), \quad (t > a),$$
$$y(a) = y_a,$$

where

$${}_0R_a^\alpha y(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^a (t-\tau)^{\alpha-1} y(\tau)\, d\tau, \quad (t > a).$$

$${}_0R_a^\alpha y(t) = {}_0D_t^\alpha \big((1 - H(t-a))y(t)\big)$$

First "large step" in [0, a]

$${}_0D_t^\alpha y(t) = f(y(t), t), \quad (t > 0),$$
$$y(0) = 0,$$

$${}_aD_t^\alpha y(t) = f(y(t), t) - {}_0R_a^\alpha y(t), \quad (t > a),$$
$$y(a) = y_a,$$

Auxiliary function:

$$y(t) = u(t) + y_a,$$

Second "large step" in [a, b]

$${}_aD_t^\alpha u(t) = f(u(t) + y_a, t) - {}_0R_a^\alpha y(t) - y_a, \quad (t > a),$$
$$u(a) = 0.$$

Method of "large steps": example (1)

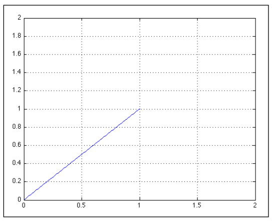
$$ {}_0D_t^{1/2}y(t) + y(t) = \frac{t^{0.5}}{\Gamma(1.5)} + t, \quad (t > 0), \qquad y(0) = 0. $$

Exact solution: $y(t) = t.$

First "large step": interval $[0, 1]$:

Using the matrix approach

```
clear all
h = 0.01;
t = 0:h:1;
N = 1/h + 1;
M = zeros(N,N);
M = ban(0.5, N, h) + eye(N,N);
F = (t.^(0.5)/gamma(1.5) + t)';
M = eliminator(N,[1])*M*eliminator(N, [1])';
F = eliminator(N,[1])*F;
Y = M\F;
Y0 = [0; Y];
plot (t,Y0,'b')
set(gca, 'xlim', [0 2], 'ylim', [0 2] )
grid on, hold on
```



Method of "large steps": example (2)

Second "large step": interval $[1, 2]$

$$ {}_0D_t^{1/2}y(t) = {}_1D_t^{1/2}y(t) + \frac{1}{\Gamma(0.5)} \int_0^1 \frac{y'(\tau)d\tau}{(t-\tau)^{1/2}}, \quad (t > 1) $$

$$ {}_1D_t^{1/2}y(t) + y(t) = \frac{t^{0.5}}{\Gamma(1.5)} + t - \frac{1}{\Gamma(0.5)} \int_0^1 \frac{d\tau}{(t-\tau)^{1/2}} \quad (t > 1). $$

$$ {}_1D_t^{\alpha}y(t) + y(t) = \frac{t^{0.5}}{\Gamma(1.5)} + t - \frac{2t^{0.5}}{\Gamma(0.5)} + \frac{2(t-1)^{0.5}}{\Gamma(0.5)}; \quad (t > 1) $$

$$ y(1) = 1. $$

$$ y(t) = u(t) + 1, $$

Method of "large steps": example (3)

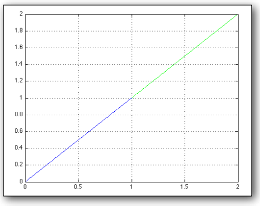
The problem to solve in $[1, 2]$:

$$ {}_1D_t^{\alpha}u(t) + u(t) = \frac{t^{0.5}}{\Gamma(1.5)} + t - \frac{2t^{0.5}}{\Gamma(0.5)} + \frac{2(t-1)^{0.5}}{\Gamma(0.5)} - 1; \quad (t > 1) $$

$$ u(1) = 0. $$

Using the matrix approach

```
clear all
h = 0.01;
t = 1:h:2;
N = 1/h + 1;
M = zeros(N,N);
M = ban(0.5, N, h) + eye(N,N);
F = (t.^(0.5)/gamma(1.5) + t - 2*t.^(0.5)/gamma(0.5) ...
        + 2*(t-1).^(0.5)/gamma(0.5) - 1)';
M = eliminator(N,[1])*M*eliminator(N, [1])';
F = eliminator(N,[1])*F;
U = M \F;
U0 = [0; U];
Y0 = U0 + 1;
plot(t, Y0, 'g')
```



Method of "large steps" and the problem of initialization

Lorenzo and Hartley raised the question about initialization of fractional derivatives. Their motivation was to use or recover the information about the process $y(t)$ in the interval $(0, a)$, if we consider fractional derivatives of $y(t)$ in (a, ∞).

NOTE: in the second "large step" in the considered sample problem we used, in fact, the proper initialization of the fractional derivative in the interval $(1, 2)$ based on the known behavior of $y(t)$ in $(0, 1)$.

Are you bored? Try my Matrix Approach to poker!