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

#3
Matrix approach to numerical fractional calculus

Igor Podlubny
Technical University of Kosice, Slovakia

http://www.tuke.sk/podlubny

Most used definitions of fractional differentiation

Riemann–Liouville, 1920s: (Letnikov, 1870s)

$$ {}_aD_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt}\right)^n \int_a^t \frac{f(\tau)\,d\tau}{(t-\tau)^{\alpha-n+1}}, \qquad (n-1 \le \alpha < n) $$

Caputo, 1967:

Miller–Ross, 1990s (Dzhrbashyan, 1960s)

For $f \in C^{(n)}[a,b], \quad f^{(k)}(a) = 0 \ (k = 0, \ldots, n-1)$
R-L, C, M-R and G-L definitions
are equivalent

Grünwald–Letnikov, 1860s (Liouville, 1830s)

$$ D^\alpha f(t) = \lim_{h\to 0} h^{-\alpha} \sum_{k=0}^{\left[\frac{t-a}{h}\right]} (-1)^k \binom{\alpha}{n} f(t-kh) $$

“Message in a Bottle” from lecture #2

All algorithms (G1, G2, R1, R2, L1, L2, D) discussed in lecture #2 can be written in the same form as

$$ {}_0D_t^\alpha f(t) = h^{-\alpha} \sum_{j=0}^{N} w_j(\alpha) f_j $$

Decoding the “message in a bottle”

- Triangular strip matrices, operations with TSMs
- Uniform approach to discretization of derivatives and integrals of arbitrary order
- Using TSMs for numerical solution of ordinary fractional differential equations
- Using TSMs for numerical solution of partial fractional differential equations

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

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

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:

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

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

Operations with TSMs

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

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

Addition and subtraction:

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

Multiplication by a constant:

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

Product of TSMs:

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

Matrix inversion:

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

Decoding the "message in a bottle"

- Triangular strip matrices, operations with TSMs
- Uniform approach to discretization of derivatives and integrals of arbitrary order
- Using TSMs for numerical solution of ordinary fractional differential equations
- Using TSMs for numerical solution of partial fractional differential equations

Integer-order differentiation
Backward differences

Approximation of the first order derivative:

$$f'(t_k) \approx \frac{1}{h}\,\nabla f(t_k) = \frac{1}{h}\,(f_k - f_{k-1}), \quad k = 1, \ldots, N.$$

All these formulas can be written simultaneously:

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

Integer-order differentiation
Backward differences

Approximation of the first order derivative:

$$B_N^1 = \frac{1}{h} \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & \cdots & 0 & -1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

Generating function:

$$\beta_1(z) = h^{-1}(1-z).$$

Integer-order differentiation
Backward differences

Approximation of the second order derivative:

$$f''(t_k) \approx \frac{1}{h^2}\,\nabla^2 f(t_k) = \frac{1}{h^2}\,(f_k - 2f_{k-1} + f_{k-2}), \quad k = 2, \ldots, N$$

All these formulas can be written simultaneously, too:

$$\begin{bmatrix} h^{-2}\,f_0 \\ h^{-2}\,(-2f_0 + f_1) \\ h^{-2}\,\nabla^2 f(t_2) \\ \vdots \\ h^{-2}\,\nabla^2 f(t_{N-1}) \\ h^{-2}\,\nabla^2 f(t_N) \end{bmatrix} = \frac{1}{h^2} \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix}$$

Integer-order differentiation
Backward differences

Approximation of the second order derivative:

$$B_N^2 = \frac{1}{h^2} \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix}$$

Generating function:

$$\beta_2(z) = h^{-2}(1 - 2z + z^2) = h^{-2}(1-z)^2$$

Integer-order differentiation
Backward differences

Approximation of the p-th order derivative:

$$B_N^p = \frac{1}{h^p}\begin{bmatrix} \omega_0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \omega_1 & \omega_0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \omega_2 & \omega_1 & \omega_0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \ddots & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \ddots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \ddots & 0 & \dots \\ 0 & \dots & 0 & \omega_p & \omega_{p-1} & \dots & \omega_0 & 0 \\ 0 & 0 & \dots & 0 & \omega_p & \omega_{p-1} & \dots & \omega_0 \end{bmatrix} \qquad \omega_j = (-1)^j \binom{p}{j}, \qquad j = 0,\, 1,\, 2,\, \dots,\, p$$

Generating function:

$$\beta_p(z) = h^{-p}(1-z)^p$$

Integer-order differentiation
Backward differences

For the generating functions we have:

$$\begin{aligned} \beta_2(z) &= \beta_1(z)\beta_1(z) \\ \beta_p(z) &= \underbrace{\beta_1(z)\dots\beta_1(z)}_{p} \\ \beta_{p+q}(z) &= \beta_p(z)\beta_q(z) = \beta_q(z)\beta_p(z) \end{aligned}$$

and therefore

$$\begin{aligned} B_N^2 &= B_N^1\, B_N^1, \\ B_N^p &= \underbrace{B_N^1\, B_N^1 \dots B_N^1}_{p}, \\ B_N^{p+q} &= B_N^p\, B_N^q = B_N^q\, B_N^p \end{aligned}$$

Left-sided fractional derivatives

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

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

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

Left-sided fractional derivatives

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

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

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

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

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

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

Integer-order integration
Moving upper limit

One-fold integral:

$$g_1(t) = \int_a^t f(t)dt.$$

Approximation:

$$g_1(t_k) \approx h\sum_{i=0}^{k-1} f_i, \qquad k = 1, \dots, N.$$

All these formulas can be written simultaneously:

$$\begin{bmatrix} g_1(t_1) \\ g_1(t_2) \\ g_1(t_3) \\ \vdots \\ g_1(t_N) \\ g_1(t_N+h) \end{bmatrix} = h\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ 1 & \cdots & 1 & 1 & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix}$$

Integer-order integration
Moving upper limit

Approximation of one-fold integration:

$$I_N^1 = h\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ 1 & \cdots & 1 & 1 & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 & 1 \end{bmatrix}$$

Generating function:

$$\varphi_1(z) = h(1-z)^{-1}$$

Integer-order integration
Moving upper limit

Notice that matrix I_N^1 is inverse to the matrix B_N^1 :

$$B_N^1 I_N^1 = I_N^1 B_N^1 \longleftrightarrow \mathrm{trunc}_N\left(\beta_1(z)\,\varphi_1(z)\right) = 1 \longleftrightarrow E.$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & \cdots & 0 & -1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ 1 & \cdots & 1 & 1 & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 & 1 \end{bmatrix} = ?$$

Integer-order integration
Moving upper limit

Two-fold integral:

$$g_2(t) = \int_a^t dt \int_a^t f(t)dt$$

Approximation:

$$\begin{aligned} g_2(t_k) &= h\sum_{i=0}^{k-1} g_1(t_i) = h\sum_{i=1}^{k-1} g_1(t_i) = h\sum_{i=1}^{k-1} h\sum_{j=0}^{i-1} f_j \\ &= h^2 \sum_{i=1}^{k-1}\sum_{j=0}^{i-1} f_j = h^2 \sum_{j=0}^{k-2} (k-j-1)\, f_j \\ &= h^2\Big((k-1)f_0 + (k-2)f_1 + \ldots + 2f_{k-3} + f_{k-2}\Big) \end{aligned}$$

$$k = 2, 3, \ldots, N.$$

Integer-order integration
Moving upper limit

Approximation of the two-fold integration:

$$g_2(t_k) = h^2\Big((k-1)f_0 + (k-2)f_1 + \ldots + 2f_{k-3} + f_{k-2}\Big), \qquad k = 2, 3, \ldots, N.$$

All these formulas can be written simultaneously, too:

$$\begin{bmatrix} g_2(t_2) \\ g_2(t_3) \\ \vdots \\ g_2(t_N) \\ g_2(t_N + h) \\ g_2(t_N + 2h) \end{bmatrix} = h^2 \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 2 & 1 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots \\ \cdots & 3 & 2 & 1 & 0 & 0 \\ N & \cdots & 3 & 2 & 1 & 0 \\ N+1 & N & \cdots & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-2} \\ f_{N-1} \\ f_N \end{bmatrix}$$

Integer-order integration
Moving upper limit

Approximation of the two-fold integration:

$$I_N^2 = h^2 \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 2 & 1 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots \\ \cdots & 3 & 2 & 1 & 0 & 0 \\ N & \cdots & 3 & 2 & 1 & 0 \\ N+1 & N & \cdots & 3 & 2 & 1 \end{bmatrix}$$

Generating function:

$$\varphi_2(z) = h^2(1-z)^{-2}$$

Integer-order integration
Moving upper limit

Notice that matrix I_N^2 is inverse to the matrix B_N^2 :

$$B_N^2 I_N^2 = I_N^2 B_N^2 \longleftrightarrow \mathrm{trunc}_N\left(\beta_2(z)\,\varphi_2(z)\right) = 1 \longleftrightarrow E.$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 2 & 1 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots \\ \cdots & 3 & 2 & 1 & 0 & 0 \\ N & \cdots & 3 & 2 & 1 & 0 \\ N+1 & N & \cdots & 3 & 2 & 1 \end{bmatrix} = ?$$

Integer-order integration
Moving upper limit

p-fold integration: $g_p(t) = \int_a^t d\tau_p \int_a^{\tau_p} d\tau_{p-1} \ldots \int_a^{\tau_2} f(\tau_1) d\tau_1$

Approximation:

$$I_N^p = h^p \begin{bmatrix} \gamma_0 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ \gamma_1 & \gamma_0 & 0 & 0 & \cdots & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \gamma_2 & \gamma_1 & \gamma_0 & 0 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ \gamma_{N-1} & \cdots & \cdots & \gamma_2 & \gamma_1 & \gamma_0 & 0 \\ \gamma_N & \gamma_{N-1} & \cdots & \cdots & \gamma_2 & \gamma_1 & \gamma_0 \end{bmatrix}$$

Generating function:

$$\varphi_p(z) = h^p(1-z)^{-p}$$

Integer-order integration

Moving upper limit

Notice that matrix I_N^p is inverse to the matrix B_N^p:

$$B_N^p I_N^p = I_N^p B_N^p \longleftrightarrow \mathrm{trunc}_N\left(\beta_p(z)\,\varphi_p(z)\right) = 1 \longleftrightarrow E$$

Properties:

$$\begin{aligned} I_N^2 &= I_N^1\, I_N^1, \\ I_N^p &= \underbrace{I_N^1\, I_N^1 \ldots I_N^1}_{p}, \\ I_N^{p+q} &= I_N^p\, I_N^q = I_N^q\, I_N^p \end{aligned}$$

Matrices I_N^p commute with matrices B_N^p .

Left-sided fractional integrals

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

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

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

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

Decoding the "message in a bottle"

- Triangular strip matrices, operations with TSMs
- Uniform approach to discretization of derivatives and integrals of arbitrary order
- Using TSMs for numerical solution of ordinary fractional differential equations
- Using TSMs for numerical solution of partial fractional differential equations

Useful matrices: Eliminators

Eliminator, $S_{r_1,r_2,\ldots,r_k}$, is obtained from the unit matrix by omitting rows with numbers $r_1, r_2, \ldots, r_k$.

How do they act:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}; \qquad S_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$$S_1\,A = \begin{bmatrix} a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}; \qquad A\,S_1^T = \begin{bmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}; \qquad S_1\,A\,S_1^T = \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}.$$

Useful matrices: Eliminators

In general,

$$S_0 \left\{ \begin{matrix} L_N \\ U_N \end{matrix} \right\} S_0^T = \left\{ \begin{matrix} L_{N-1} \\ U_{N-1} \end{matrix} \right\},$$

$$S_N \left\{ \begin{matrix} L_N \\ U_N \end{matrix} \right\} S_N^T = \left\{ \begin{matrix} L_{N-1} \\ U_{N-1} \end{matrix} \right\},$$

$$S_{0,1,\ldots,k} \left\{ \begin{matrix} L_N \\ U_N \end{matrix} \right\} S_{0,1,\ldots,k}^T = \left\{ \begin{matrix} L_{N-k-1} \\ U_{N-k-1} \end{matrix} \right\},$$

$$S_{N-k,N-k+1,\ldots,N} \left\{ \begin{matrix} L_N \\ U_N \end{matrix} \right\} S_{N-k,N-k+1,\ldots,N}^T = \left\{ \begin{matrix} L_{N-k-1} \\ U_{N-k-1} \end{matrix} \right\}.$$

Simultaneous multiplication of a triangular strip matrix by an eliminator $S_{0,1,\ldots,k}$ (or $S_{N-k,N-k+1,\ldots,N}$) on the left and its transpose on the right preserves the type and the structure of the triangular strip matrix, and only reduces its size by $k{+}1$ rows and $k{+}1$ columns.

Initial value problems for FDEs

Discretization of an equation

Consider linear FDE with non-constant coefficients:

(R-L or Caputo)

$$\sum_{k=1}^{m} p_k(t) D^{\alpha_k} y(t) = f(t), \qquad 0 \le \alpha_1 < \alpha_2 < \ldots < \alpha_m, \qquad n-1 < \alpha_m < n$$

Denote

$$P_N^{(k)} = \mathrm{diag}\big(p_k(t_0), p_k(t_1), \ldots, p_k(t_N)\big) = \begin{bmatrix} p_k(t_0) & 0 & \ldots & 0 \\ 0 & p_k(t_1) & 0 & \ldots \\ 0 & \ldots & \ddots & 0 \\ 0 & \ldots & 0 & p_k(t_N) \end{bmatrix},$$

$$Y_N = \big(y(t_0), y(t_1), \ldots, y(t_N)\big)^T, \qquad F_N = \big(f(t_0), f(t_1), \ldots, f(t_N)\big)^T.$$

Then the discrete form of the equation is simply:

$$\sum_{k=1}^{m} P_N^{(k)} B_N^{\alpha_k} Y_N = F_N$$

Initial value problems for FDEs

Handling zero initial conditions

If $n-1 < \alpha_m < n$ and $y^{(k)}(t_0) = 0, \qquad k = 0, 1, \ldots, n-1$,
then the Riemann-Liouville and Caputo derivatives coincide.

Approximating derivatives in the above conditions by backward differences we immediately have:

$$y(t_0) = y(t_1) = \ldots = y(t_{n-1}) = 0.$$

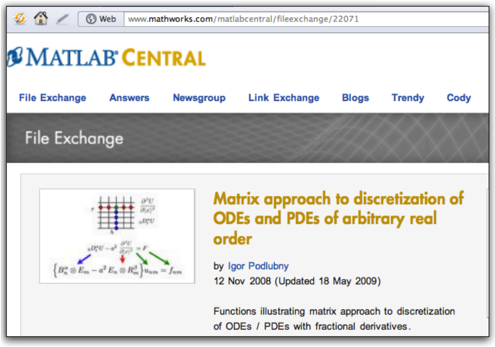
and the system for finding the rest is:

$$\left\{ S_{0,1,\ldots,n-1} \left\{ \sum_{k=1}^{m} P_N^{(k)} B_N^{\alpha_k} \right\} S_{0,1,\ldots,n-1}^T \right\} \{S_{0,1,\ldots,n-1} Y_N\} = S_{0,1,\ldots,n-1} F_N.$$

For constant coefficients it is even simpler:

$$\sum_{k=1}^{m} p_k B_{N-n}^{\alpha_k} \{S_{0,1,\ldots,n-1} Y_N\} = S_{0,1,\ldots,n-1} F_N.$$

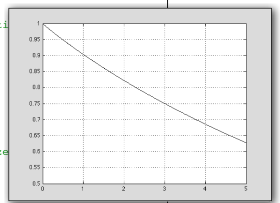
Matrix approach – toolbox for MATLAB



Example 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 2: Caputo derivatives

Zero initial conditions

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

Exact solution is: $y(t) = t^{\alpha} E_{\alpha,\alpha+1}(-t^{\alpha})$

From $$\sum_{k=1}^{m} p_k B_{N-n}^{\alpha_k} \{S_{0,1,\ldots,n-1} Y_N\} = S_{0,1,\ldots,n-1} F_N.$$

$$m = 2, \; \alpha_1 = \alpha, \; \alpha_2 = 0, \; n = 2, \; p_1 = p_2 = 1,$$
$$B_{N-n}^{\alpha_1} = B_{N-2}^{\alpha}, \; B_{N-n}^{\alpha_2} = E_{N-2}, \; F_N = (\underbrace{1, 1, \ldots, 1}_{N})^T$$

the system for determining $y_k, \; k = 2, 3, \ldots, N$ is:

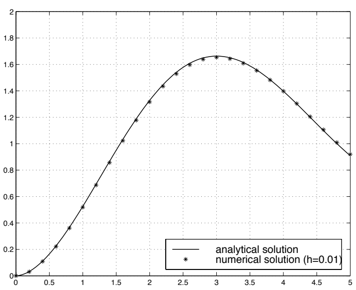
$$\{B_{N-2}^{\alpha} + E_{N-2}\} \{S_{0,1} Y_N\} = S_{0,1} F_N$$

... and don't forget to add $y_0 = y_1 = 0.$

Example 2: Caputo derivatives

Zero initial conditions

Solution of the problem $y^{(1.8)}(t) + y(t) = 1, y(0) = 0, y'(0) = 0$



Example 3: Caputo derivatives

Non-zero initial conditions: transform them to zeros.

The problem: $y^{(\alpha)}(t) + y(t) = 1,$
$$y(0) = c_0, \quad y'(0) = c_1$$

Exact solution is: $y(t) = c_0 E_{\alpha,1}(-t^{\alpha}) + c_1 t E_{\alpha,2}(-t^{\alpha}) + t^{\alpha} E_{\alpha,\alpha+1}(-t^{\alpha})$

Introduce an auxiliary function:

$$y(t) = c_0 + c_1 t + z(t)$$

Then the problem for $z(t)$ is:

$$z^{(\alpha)}(t) + z(t) = 1 - c_0 - c_1 t,$$

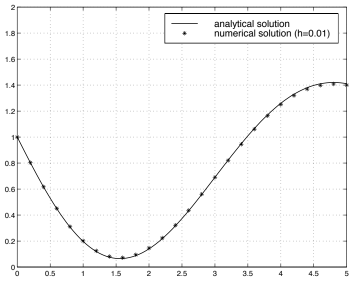
bounded RHS

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

Example 3: Caputo derivatives

Non-zero initial conditions: transform them to zeros.

Solution of the problem $y^{(1.8)}(t) + y(t) = 1, y(0) = 1, y'(0) = -1$



Example 4: Riemann-Liouville derivatives

Non-zero initial conditions: transform them to zeros.

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

$$y^{(\alpha-1)}(0) = c_0, \quad y^{(\alpha-2)}(0) = c_1.$$

Exact solution is:

$$y(t) = c_0 t^{\alpha-1} E_{\alpha,\alpha}(-t^\alpha) + c_1 t^{\alpha-2} E_{\alpha,\alpha-1}(-t^\alpha) + t^\alpha E_{\alpha,\alpha+1}(-t^\alpha)$$

Introduce an auxiliary function:

$$y(t) = c_0 t^{\alpha-1} + c_1 t^{\alpha-2} + z(t)$$

Then the problem for $z(t)$ is:

$$z^{(\alpha)}(t) + z(t) = 1 - c_0 t^{\alpha-1} - c_1 t^{\alpha-2}$$

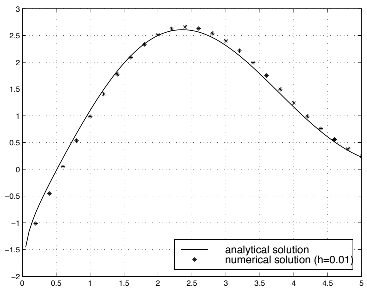
unbounded RHS

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

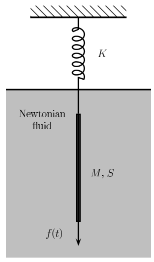
Example 4: Riemann-Liouville derivatives

Non-zero initial conditions: transform them to zeros.

Solution of the problem $y^{(1.8)}(t) + y(t) = 1; y^{(0.8)}(0) = 1; y^{(-0.2)}(0) = -1$



Motion of an immersed plate



Bagley-Torvik equation:

$$Ay''(t) + B\,{}_0D_t^{3/2} y(t) + Cy(t) = f(t) \qquad (t > 0),$$

$$A = M, \quad B = 2S\sqrt{\mu\rho}, \quad C = K,$$

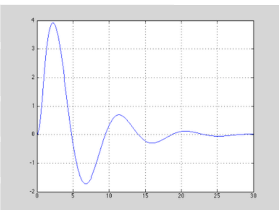
add initial conditions!

$$y(0) = 0, \qquad y'(0) = 0$$

Example 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)
```



Nonlinear FDEs? Not a problem

$y^{(\alpha_i)}(t) = {}_aD_t^{\alpha_i} y(t)$

$$y^{(\alpha_1)}(t) = f(t,\, y^{(\alpha_2)}(t),\, y^{(\alpha_3)}(t),\, \ldots,\, y^{(\alpha_k)}(t)),$$

$$(0 < \alpha_1 < \alpha_2 < \ldots < \alpha_k \le n.)$$

Suppose initial conditions are already transformed to zero initial conditions. Then replacement of derivatives with their discrete analogues gives:

$$B_N^{\alpha_1} Y_N = f(Et_N,\, B_N^{\alpha_2} Y_N,\, B_N^{\alpha_3} Y_N,\, \ldots,\, B_N^{\alpha_k} Y_N),$$

$$y_j = 0, \quad j = 1, 2, \ldots, n-1,$$

This is a nonlinear algebraic system.

Nonlinear FDEs? Not a problem

Diethelm K., Weilbeer M.: A numerical approach for Joulin's model of appoint source initiated flame.
Fractional Calculus and Applied Analysis, (7):2, 191-212, 2004

problem

$$R(t)D_*^{1/2}R(t) = R(t)\ln R(t) + Eq(t), \qquad R(0) = 0. \tag{1}$$

Here $D_*^{1/2}$ denotes the Caputo differential operator of order $1/2$, defined by (see, e.g., [6])

$$D_*^{1/2}y(t) = \frac{1}{\sqrt{\pi}}\int_0^t (t-s)^{-1/2}y'(s)\,\mathrm{d}s.$$

Moreover the function q describes a point source energy that depends on the time, and therefore it is assumed to be nonnegative, continuous and integrable

The nonlinear algebraic system is solved by Newton's method

Decoding the "message in a bottle"

- Triangular strip matrices, operations with TSMs
- Uniform approach to discretization of derivatives and integrals of arbitrary order
- Using TSMs for numerical solution of ordinary fractional differential equations
- Using TSMs for numerical solution of partial fractional differential equations

Kronecker matrix product

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}, \qquad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1q} \\ b_{21} & b_{22} & \dots & b_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \dots & b_{pq} \end{bmatrix}$$

Kronecker matrix product:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ a_{21}B & a_{22}B & \dots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \dots & a_{nm}B \end{bmatrix}$$

Kronecker matrix product

Example:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & -3 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$A \otimes B = \begin{bmatrix} 1 & 2 & 3 & 2 & 4 & 6 \\ 4 & 5 & 6 & 8 & 10 & 12 \\ 0 & 0 & 0 & -3 & -6 & -9 \\ 0 & 0 & 0 & -12 & -15 & -18 \end{bmatrix}$$

Kronecker matrix product

Important properties:

- if A and B are band matrices, then $A \otimes B$ is also a band matrix,
- if A and B are lower (upper) triangular, then $A \otimes B$ is also lower (upper) triangular.

Kronecker matrix product

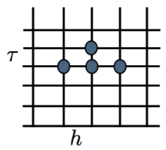
Kronecker products with identity matrices: example:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

$$E_2 \otimes A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{11} & a_{12} & a_{13} \\ 0 & 0 & 0 & a_{21} & a_{22} & a_{23} \end{bmatrix} \qquad A \otimes E_3 = \begin{bmatrix} a_{11} & 0 & 0 & a_{12} & 0 & 0 & a_{13} & 0 & 0 \\ 0 & a_{11} & 0 & 0 & a_{12} & 0 & 0 & a_{13} & 0 \\ 0 & 0 & a_{11} & 0 & 0 & a_{12} & 0 & 0 & a_{13} \\ a_{21} & 0 & 0 & a_{22} & 0 & 0 & a_{23} & 0 & 0 \\ 0 & a_{21} & 0 & 0 & a_{22} & 0 & 0 & a_{23} & 0 \\ 0 & 0 & a_{21} & 0 & 0 & a_{22} & 0 & 0 & a_{23} \end{bmatrix}$$

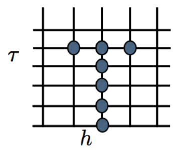
Discretization schemes

Integer orders:

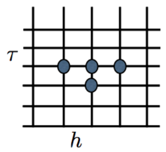


$$\frac{\partial U}{\partial t} \qquad \frac{\partial^2 U}{\partial x^2}$$

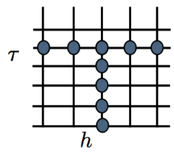
Fractional orders:



$$\frac{\partial^2 U}{\partial x^2} \qquad {}_0D_t^{\alpha}U$$



$$\frac{\partial^2 U}{\partial x^2} \qquad \frac{\partial U}{\partial t}$$

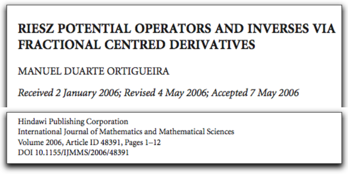


$$\frac{\partial^\beta U}{\partial |x|^\beta} \qquad {}_0D_t^{\alpha}U$$

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

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:

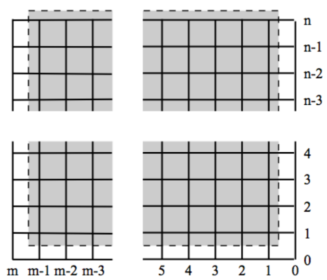
$$R_m^{(\beta)} = \frac{h^{-\alpha}}{2}\left[{}_{-1}U_m + {}_{+1}U_m \right]$$

(2) Using Ortigueira's centred differences:

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

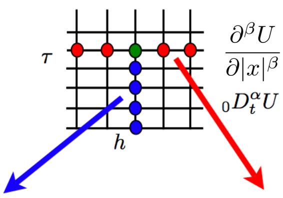
$$\omega_k^{(\beta)} = \frac{(-1)^k\,\Gamma(\beta+1)\cos(\beta\pi/2)}{\Gamma(\beta/2-k+1)\,\Gamma(\beta/2+k+1)}$$

Discretization net



Nodes and their right-to-left, and bottom-to-top numbering.

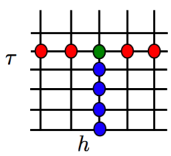
Discretization using TSMs



$$\frac{\partial^\beta U}{\partial |x|^\beta} \qquad {}_0D_t^{\alpha}U$$

$$B_n^\alpha \otimes E_m \qquad E_n \otimes R_m^\beta$$

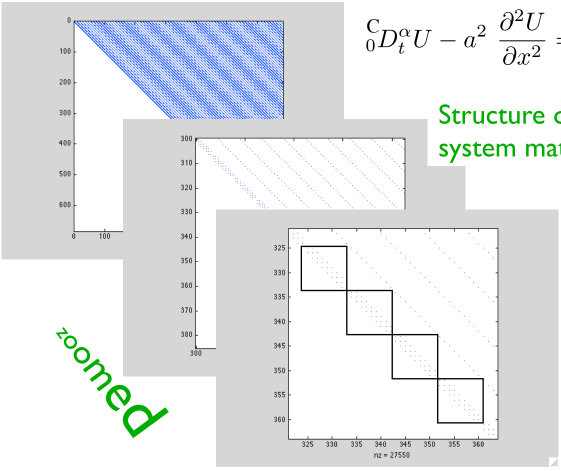
Discretization using TSMs



$$\frac{\partial^\beta U}{\partial |x|^\beta} \qquad {}_0D_t^{\alpha}U$$

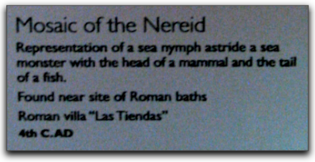
$${}_0D_t^\alpha U - a^2 \frac{\partial^\beta U}{\partial |x|^\beta} = F$$

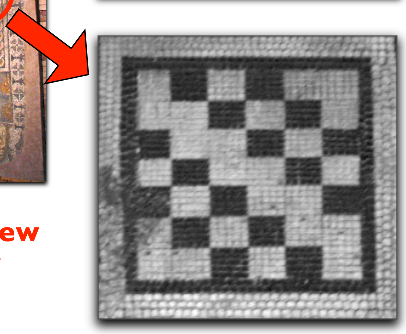
$$\left\{ B_n^\alpha \otimes E_m - a^2\, E_n \otimes R_m^\beta \right\} u_{nm} = f_{nm}$$



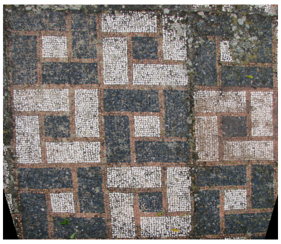
$${}^{\mathrm{C}}_{0}D^{\alpha}_{t}U - a^2\,\frac{\partial^2 U}{\partial x^2} = F$$

Structure of the system matrix



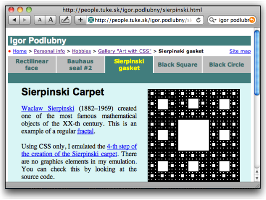


Did Romans knew about TSMs?



Sierpinski gasket:
found in ancient
Roman houses
in Merida,
(around 0 A.D.)

... and emulated
using HTML+CSS
on my web page...

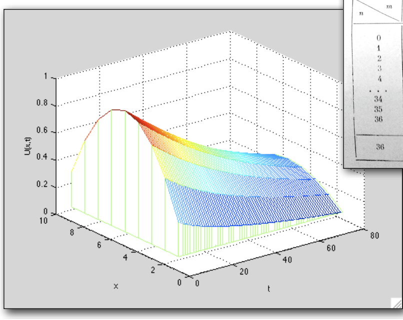


The HTML code is self-similar!

Test example

$$U_t = a^2 U_{xx}$$
$$U(0,t) = 0, \quad U(L,t) = 0$$
$$U(x,0) = 4x(L-x)/L^2$$

W. E. Milne, Numerical Solution of
Differential Equations, Wiley, NY, 1953
(the table below is from the Russian translation, 1955)



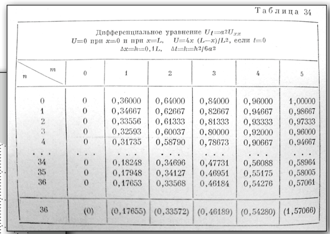


Таблица 34

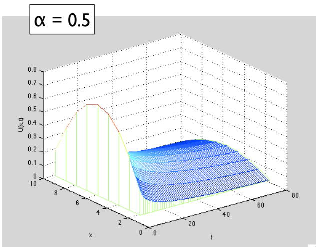
Дифференциальное уравнение $U_t = a^2 U_{xx}$
$U=0$ при $x=0$ и при $x=L$, $U=4x\,(L-x)/L^2$, если $t=0$
$\Delta x = h = 0,1L$, $\Delta t = k = h^2/6a^2$

n \ m	0	1	2	3	4	5
0	0	0,36000	0,64000	0,84000	0,96000	1,00000
1	0	0,34667	0,62667	0,82667	0,94667	0,98667
2	0	0,33556	0,61333	0,81333	0,93333	0,97333
3	0	0,32593	0,60037	0,80000	0,92000	0,96000
4	0	0,31735	0,58790	0,78673	0,90667	0,94667
...	...	...	...	...	...	...
34	0	0,18248	0,34696	0,47731	0,56088	0,58964
35	0	0,17948	0,34127	0,46951	0,55175	0,58005
36	0	0,17653	0,33568	0,46184	0,54276	0,57061
36	(0)	(0,17655)	(0,33572)	(0,46189)	(0,54280)	(1,57066)

0.3484	0.6269	0.8267	0.9467	0.9867
0.3380	0.6141	0.8135	0.9334	0.9733
0.3287	0.6017	0.8003	0.9201	0.9600
0.3203	0.5897	0.7873	0.9068	0.9467
...	...	...	...	...
0.1843	0.3504	0.4817	0.5659	0.5948
0.1813	0.3447	0.4740	0.5568	0.5853
0.1784	0.3391	0.4664	0.5479	0.5760

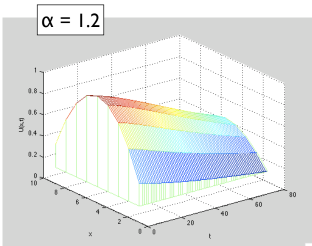
Extending the test example to fractional orders:

α = 0.5



$${}_{0}D^{\alpha}_{t}U = a^2\frac{\partial^2 U}{\partial x^2}$$
$$U(0,t) = 0, \quad U(L,t) = 0$$
$$U(x,0) = 4x(L-x)/L^2$$

α = 1.2



Conclusions

- Uniform approach to differentiation and integration of arbitrary (integer and non-integer) order.
- This approach is easy, algorithmic, modular.
- Allows solution of nonlinear problems.
- Allows solution of partial fractional differential equations with several fractional-order derivatives.
- Allows consideration and solution of fractional differential equations with delays.
- TSMs are a particular case of the Toeplitz matrices — theory and tools exist.