International Conference on Fractional Differentiation and Its Applications, ICFDA '14
Catania, June 23-25, 2014

Identification through approximation of fractional order models

P. NOVATI AND M. R. RUSSO
Department of Mathematics, University of Padua, Italy

Caputo's fractional derivative

We consider Fractional Differential Equations (FDEs) of the type

$$ {}_{t_0}D_t^{\alpha} y(t) = g(t, y(t)), \quad t_0 < t \leq T, \quad 0 < \alpha < 1, \qquad (1)$$

where ${}_{t_0}D_t^{\alpha}$ denotes the Caputo's fractional derivative operator defined as

$$ {}_{t_0}D_t^{\alpha} y(t) = \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t} \frac{y'(u)}{(t-u)^{\alpha}} du,$$

and Γ is the Gamma function.

Setting the initial condition $y(t_0) = y_0$, the solution of (1) exists and is unique under the hypothesis that g is continuous and fulfils a Lipschitz condition with respect to the second variable.

Fractional BDF formulas

Similarly to the integer order case $\alpha = 1$, a classical approach for solving (1) is based on the discretization of the fractional derivative, which generalizes the well known Grünwald-Letnikov discretization [2, sec.2.2]. This kind of discretization leads to the so-called Fractional Backward Differentiation Formulas (FBDFs) introduced in [2].

Taking a uniform mesh $t_0, t_1, \ldots, t_N = T$ of the time domain with stepsize $h = (T - t_0)/N$, FBDFs are full-term recursion formulas of the type

$$\sum_{j=0}^{n} \omega_{n-j}^{(p)} y_j = h^{\alpha} g(t_n, y_n), \quad p \leq n \leq N, \qquad (2)$$

where $y_j \approx y(t_j)$. The coefficients $\omega_{n-j}^{(p)}$ are the Taylor coefficients of the generating function

$$\begin{aligned} \omega_p^{(\alpha)}(\zeta) &= (a_0 + a_1\zeta + \ldots + a_p\zeta^p)^{\alpha} & (3)\\ &= \sum_{i=0}^{\infty} \omega_i^{(p)} \zeta^i, \quad \text{for} \quad 1 \leq p \leq 6, & (4)\end{aligned}$$

being $\{a_0, a_1, \ldots, a_p\}$ the coefficients of the underlying BDF. In [2] it is shown that the order p of the BDF is preserved.

Because of the typical lack of regularity of the solution in a neighborhood of the starting point, formula (3) is generally corrected as

$$\sum_{j=0}^{M} w_{n,j} y_j + \sum_{j=0}^{n} \omega_{n-j}^{(p)} y_j = h^{\alpha} g(t_n, y_n), \qquad (5)$$

where the sum $\sum_{j=0}^{M} w_{n,j} y_j$ is the so-called starting term, in which M and the weights $w_{n,j}$ depend on α and p.

Short Memory Approach

In order to reduce the computational cost of a method based on a full recursion like (2), one typically considers a truncated Taylor expansion of the generating function (4), which leads to

$$\sum_{j=n-m}^{n} \omega_{n-j}^{(p)} y_j = h^{\alpha} g(t_n, y_n), \quad n \geq m. \qquad (6)$$

While the above formula is very easy to implement and little expensive the simple truncation of the Taylor series may yield poor results if α is not closed to 1. For this reason here we consider a more accurate approach [1, 3].

Denoting by Π_m the set of polynomials of degree not exceeding m, the idea is to construct methods based on the rational approximations of the generating function, i.e.,

$$R_m(\zeta) \approx \omega_p^{(\alpha)}(\zeta), \quad R_m(\zeta) = \frac{p_m(\zeta)}{q_m(\zeta)}, \quad p_m, q_m \in \Pi_m. \qquad (7)$$

Writing $p_m(\zeta) = \sum_{j=0}^{m} \alpha_j \zeta^j$ and $q_m(\zeta) = \sum_{j=0}^{m} \beta_j \zeta^j$, the above approximation naturally leads to implicit short memory recursions of the type

$$\sum_{j=n-m}^{n} \alpha_{n-j} y_j = h^{\alpha} \sum_{j=n-m}^{n} \beta_{n-j} g(t_j, y_j), \quad n \geq m. \qquad (8)$$

Matrix Formulation

Starting from a BDF formula of order p, which discretizes the derivative operator, we consider lower triangular banded Toeplitz matrices of the type

$$A_p = \begin{pmatrix} a_0 & 0 & & & 0 \\ \vdots & a_0 & 0 & & \\ a_p & \vdots & \ddots & 0 & \\ 0 & \ddots & & \ddots & 0 \\ & 0 & a_p & \cdots & a_0 \end{pmatrix} \in \mathbb{R}^{(N+1)\times(N+1)}. \qquad (9)$$

In this setting, $A_p^{\alpha} e_1$, $e_1 = (1, 0, \ldots, 0)^T$, contains the whole set of coefficients of the corresponding FBDF for approximating the solution of (1) in $t_0, t_1, \ldots, t_N$, that is

$$e_{j+1}^T A_p^{\alpha} e_1 = \omega_j^{(p)}, \quad 0 \leq j \leq N, \qquad (10)$$

(cf. (4)). From the theory of matrix functions, we know that the fractional power of matrix can be written as a contour integral

$$A^{\alpha} = \frac{A}{2\pi i} \int_{\Gamma} z^{\alpha-1} (zI - A)^{-1} dz,$$

where Γ is a suitable closed contour enclosing the spectrum of A, $\sigma(A)$, in its interior. Since in our case $\sigma(A_p) = \{a_0\}$ and $a_0 > 0$ we can also write

$$A^{\alpha} = \frac{A \sin(\alpha\pi)}{\alpha\pi} \int_0^{\infty} (\rho^{1/\alpha} I + A)^{-1} d\rho. \qquad (11)$$

At this point, for each suitable change of variable for ρ, a k-point quadrature rule for the computation of the integral in (11) yields a rational approximation of the type

$$A_p^{\alpha} \approx A_p \widetilde{R}_k(A_p) := A_p \sum_{j=1}^{k} \gamma_j (\eta_j I + A_p)^{-1}, \qquad (12)$$

where the coefficients γ_j and η_j depend on the substitution and the quadrature formula.
In order to remove the dependence of α inside the integral we consider the change of variable [1]

$$\rho^{1/\alpha} = \tau \frac{1-t}{1+t}, \qquad \tau > 0, \qquad (13)$$

$$A_p^{\alpha} = \frac{2\sin(\alpha\pi)\tau^{\alpha}}{\pi} A_p \int_{-1}^{1} \frac{(1-t)^{\alpha-1}}{(1+t)^{\alpha}} \left(\tau (1-t) I + (1+t) A_p\right)^{-1} dt. \qquad (14)$$

The above formula naturally leads to the use of a k-point Gauss-Jacobi rule for the approximation of $A_p^{\alpha} e_1$ and hence to a rational approximation (12).
Using any algorithm which transforms partial fractions to polynomial quotient we finally arrive to an approximation of type (7), and therefore to a short-memory method (8).

Identification through approximation

Let $\bar{y}_i \in \mathbb{R}^M$, $n = 0, \ldots, N$ be an observed discrete solution at equally spaced points $t_0, \ldots, t_N$, which is assumed to satisfies an FDE model (1) in which α is unknown. Assuming that the function g depends on a certain number of unknown parameters $c_1, \ldots, c_k$, the identification problem can be formulated as

$$\min_{\alpha, c_1, \ldots, c_k} \sum_{n=0}^{N} \| r_n(\bar{y}) \|^2, \qquad (15)$$

where $r_n(\bar{y})$ is the residual at time t_n of a given integration scheme for FDEs applied to $\bar{y}$. This problem is generally solved by some optimization routine, such as particle swarm optimization (PSO) technique [5]. Working with the short memory recursion (8) the minimization problem (15) takes the form

$$\min_{\alpha, c_1, \ldots, c_k} \sum_{n=m}^{N} \left\| \sum_{j=n-m}^{n} \alpha_{n-j} \bar{y}_j - h^{\alpha} \sum_{j=n-m}^{n} \beta_{n-j} f(t_j, \bar{y}_j) \right\|^2. \qquad (16)$$

References

[1] L. Aceto, C. Magherini, P. Novati, On the costruction of M-step methods for FDEs, submitted 2014, Arxiv: 1405.5011v1.

[2] C. Lubich, Discretized fractional calculus, SIAM J. Math. Anal. 17 (1986), 704–719.

[3] P. Novati, Numerical approximation to the fractional derivative operator, to appear in Numerische Mathematik, 2013, DOI: 10.1007/s00211-013-0596-7.

[4] I. Podlubny, Fractional differential equations. Mathematics in Science and Engineering, 198. Academic Press, Inc., San Diego, CA, 1999.

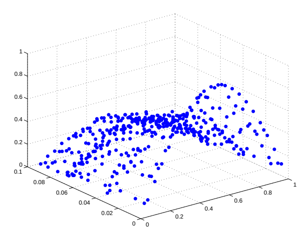
[5] J. Kennedy, R. C. Eberhart. Particle swarm optimization. Proc. of the IEEE International Conference on Neural Networks, pp.1942-1948, 1995.

A diffusion model

Given a polynomial $p(x) = c_1 + c_2 x + c_k x^{k-1}$, consider the time fractional diffusion equation

$$\begin{cases} \dfrac{\partial^{\alpha} u}{\partial t^{\alpha}} = \dfrac{\partial}{\partial x}\left(p(x)\dfrac{\partial u}{\partial x}\right), & 0 < x < 1, 0 < t < T \\ u(0,t) = u(1,t) = 0 & \\ u(x,0) = u_0(x) & \end{cases} \qquad (17)$$

which is assumed to be the model fulfilled by a discrete observed solution, representing the problem data. In order to test the minimization (16) we generate an observed solution the following way. After a semi-discretization with respect to the space variable, we take the exact solution of the semi discretized problem at randomly generated points $t'_1, \ldots, t'_q$. Then the discrete exact solution is perturbed with gaussian white noise. An example is given in the figure below.



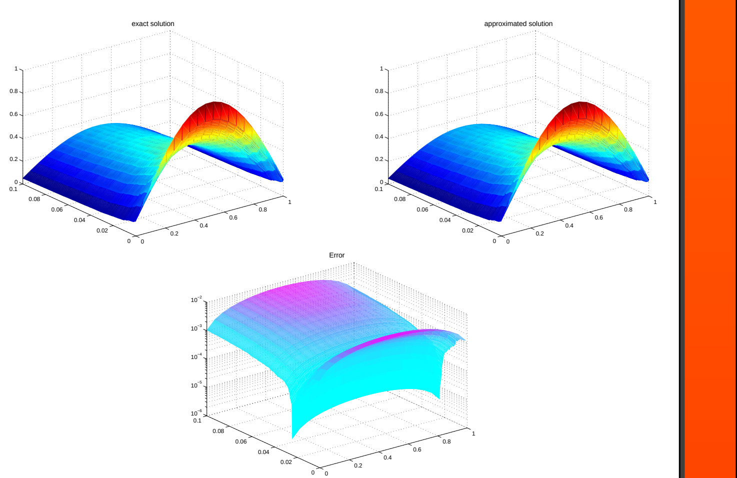
In order to deal with an uniform temporal mesh we interpolate the observed discrete solution to generate the data $\bar{y}_i$ of the formula (16).

Numerical results

We consider two experiments with $p(x) = c$, $q = 200$ random points, $N = 150$ equidistant points.

Example 1

The exact solution is generated by $\alpha = 0.5$, $c = 0.5$ and $u_0(x) = \sin(\pi x)$. The noise level is 1% of the exact data. The approximated solution is generated by the estimated parameters $\alpha_{est} = 0.517$, $c_{est} = 0.528$



Example 2

The exact solution is generated by $\alpha = 0.8$, $c = 1$ and $u_0(x) = x^3(1-x)$. The noise level is 1% of the exact data. The approximated solution is generated by the estimated parameters $\alpha_{est} = 0.801$, $c_{est} = 1.034$