Calculation of fractional derivatives of noisy data with genetic algorithms

J.A. Tenreiro Machado

Abstract This paper addresses the calculation of derivatives of fractional order for non-smooth data. The noise is avoided by adopting an optimization for- mulation using genetic algorithms (GA). Given the flexibility of the evolutionary schemes, a hierarchical GA composed by a series of two GAs, each one with a distinct fitness function, is established.

Keywords Fractional derivatives · Fractional calculus · Genetic algorithms · Numerical differentiation

1 Introduction

Fractional calculus (FC) deals with the generaliza- tion of integrals and derivatives to a non-integer, or even complex, order [1–4]. FC encompasses a wide range of potential fields of application by bringing into broader paradigm concepts of physics, chemistry and engineering [5–10]. Nevertheless, until recently, FC was an 'unknown' mathematical tool for the applied sciences, the present day interest being motivated by the developments in the areas of non-linear dynamics, chaos and modeling.

One of the reasons for this state of affairs is the lack of a simple interpretation for a fractional-order derivative. In fact, while for the integer-order case we have a common geometric concept, in the fractional- order case we have problems in finding a clear and comprehensive reasoning scheme. Several researchers proposed different approaches for the interpretation of fractional-order integrals and derivatives, but the fact is that a final paradigm is not yet well established [11–19].

A second reason for the difficulties in applying FC is due to the higher complexity of algorithms for the calculation of fractional derivatives and integrals. The generalization of the integrodifferential operator re- quires the adoption of approximations based on se- ries or rational fraction expansions [6, 8]. While the main volume of contributions is focused in getting the best approximation scheme, the problem of its calcu- lation for real data was not yet tackled. In fact, besides the quality of the approximation, two aspects must be considered in the calculation of fractional derivatives and integrals, namely, the computational load and the noise effect. The first aspect poses a small impact in today's computing systems, but the second remains to be investigated.

The problem of calculating integer-order deriva- tives for noisy data is well known. To avoid the emer- gence of high amplitude peaks the classical approach consists in adopting polynomials of increasing order, or a plethora of distinct types of low-pass filters, that somehow smooth the data [20, 28, 29]. However, it was verified that, in many cases, these measures are not successful. Bearing these facts in mind, it was recognized more recently that the problem was ill-posed and that an inverse formulation, incorporating an optimization scheme, was the best strategy [22].

In this line of thought, this paper addresses the cal-

culation of fractional derivatives of non-smooth data, and is organized as follows. Section 2 introduces the calculation of fractional derivatives for ideal data, the problem of noise and the formulation of the inverse problem, and the optimization scheme based on ge- netic algorithms. Section 3 presents a set of experi- ments that demonstrate the effectiveness of the pro- posed method. Finally, Sect. 4 outlines the main con- clusions.

2 Problem formulation and adopted tools

2.1 Fractional derivatives

Since the foundation of the differential calculus, the generalization of the concept of derivative and inte- gral to a non-integer order α has been the subject of several approaches such as the Riemann–Liouville, Grünwald–Letnikov, Caputo, and, based on trans- forms, the Fourier/Laplace definitions.

From the discrete-time point of view the Grünwald–Letnikov definition seems more attractive and, conse- quently, will be adopted in the sequel.

Based on the concept of fractional differential of order α, the Grünwald–Letnikov definition of a deriv- ative of fractional order α of the signal $x(t)$, $D^{\alpha}x(t)$, is:

$$D^{\alpha}x(t) = \lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{k=0}^{\infty}\frac{(-1)^k\Gamma(\alpha+1)x(t-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \quad (1)$$

where r is the gamma function and h is the time increment. This formulation inspires a discrete-time calculation algorithm, based on the approximation of the time increment h through the sampling period T, yielding the equation in the z domain:

$$D^{\alpha}X(z) \approx \left[\frac{1}{T^{\alpha}}\sum_{k=0}^{\infty}\frac{(-1)^k\Gamma(\alpha+1)}{k!\Gamma(\alpha-k+1)}z^{-k}\right]X(z). \quad (2)$$

The implementation of expression (2) corresponds to a r -term truncated series given by:

$$D^{\alpha}X(z) \approx \left[\frac{1}{T^{\alpha}}\sum_{k=0}^{r}\frac{(-1)^k\Gamma(\alpha+1)}{k!\Gamma(\alpha-k+1)}z^{-k}\right]X(z). \quad (3)$$

This series can be implemented by a rational frac- tion expansion which leads to a superior compromise in what concerns the number of terms versus the qual- ity of the approximation [8, 9]. Nevertheless, since the study focuses mainly on the problem of noise, the sim- ple series approximation will be adopted.

2.2 Calculation of derivatives of noisy data

In many scientific applications, it is necessary to calcu- late the derivative of numerical data. Classical finite- difference approximations amplify greatly any noise present in the data. Data denoising, before or after dif- ferentiating, does not generally give satisfactory re- sults. A method that leads to good results consists in the regularization of the differentiation process itself. This guarantees that the computed derivative will have some degree of regularity, to an extent that is under control by adjusting parameters. A common frame- work for the regularization [21–24] corresponds to the formulation of the inverse problem. In this perspective, $u(t)$, the derivative of a function $f(t)$ over the interval $t \in [0, L]$, is the minimizer of the functional:

$$F(u) = aR\{u\} + S\{A[u] - f\} \quad (4)$$

where $R\{u\}$ is a regularization term that penalizes ir- regularity in $u(t)$, $A[u(t)] = \int_0^t u\,dt$ is the operator of antidifferentiation, $S\{A[u] - f\}$ is a data similarity term that penalizes discrepancy between $A[u]$ and f, and $a \in H^+$ is a regularization parameter that controls the balance between the two terms.

The regularization and similarity terms, $R\{\}$ and $S\{\}$, adopt often the squared L^2 norm. Therefore, it is considered that the total-variation regularization and the computation of the derivative of f over the interval $[0, L]$ is the minimizer of the functional:

$$F(u) = a\int_0^L \dot{u}^2\,dt + \int_0^L \{A[u] - f\}^2\,dt \quad (5)$$

where for convenience it is assumed that $f(0) = 0$ which, in practice, consists in subtracting $f(0)$ from $f(t)$.

A simple approach to minimizing (5) is gradient descent as described in [22]. However, in the present study, due to its superior flexibility a different optimization technique will be adopted, based on genetic algorithms (GAs). In fact, the standard numerical optimization has difficulties in achieving an adequate compromise between the terms $R\{\}$ and $S\{\}$. The optimization requires several attempts, with distinct values of the regularization parameter α, and, often, we verify that there is no good tuning.

Bearing these ideas in mind, in the next sub-section we adopt the GA evolutionary scheme. We start by analyzing the performance of a standard GA in the optimization of expression (5) and, afterwards, we develop a novel technique consisting in two GAs in series, each one optimizing a separate term.

2.3 Optimization through genetic algorithms

A GA is a search technique used in computing to find exact or approximate solutions to optimization and search problems [25, 26]. GAs are simulated in a computing system, and consist in a population of representations of candidate solutions, of an optimization problem, that evolve toward better solutions.

Once the genetic representation and the fitness function are defined, the GA proceeds to initialize a population of solutions randomly, and then to improve it through the repetitive application of mutation, crossover, inversion and selection operators.

The evolution usually starts from a population of randomly generated individuals. In each generation, not only the fitness of every individual in the population is evaluated, but also several individuals are stochastically selected from the current population and modified to form a new population. The new population is then used in the next iteration of the algorithm. The GA terminates when either the maximum number of generations N has been produced, or a satisfactory fitness level has been reached.

During the successive generation, a part or the totality of the population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (measured by a fitness function) are usually more likely to be selected. The pseudo-code of the GA is:

1. Choose the initial population
2. 2. Evaluate the fitness of each individual in the population
3. Repeat
   3.1. Select best-ranking individuals to reproduce
   3.2. Breed new generation through crossover and mutation and give birth to offspring
   3.3. Evaluate the fitness of the offspring individuals
   3.4. Replace the worst ranked part of population with offspring
4. Until termination

The present article adopts also the common technique of elitism, which is the process of selecting the better individuals to form the parents in the offspring generation.

3 Fractional-order differentiation of non-smooth data

In this section, we evaluate the proposed technique in the numerical evaluation of a fractional derivative of a function corrupted by additive noise [27].

In Sect. 3.1, we start by analyzing the performance of a GA in the optimization of expression (5). We verify that the GA accomplishes the task but reveals problems similar to those encountered by standard gradient descent methods. In fact, as mentioned in the previous section, the minimization of $F(u)$ in expression (5) poses problems of establishing a compromise between the terms $R\{\}$ and $S\{\}$, leading to the necessity of several trials for the tuning of the regularization parameter α. Therefore, given the flexibility of the evolutionary schemes, in Sect. 3.2 a new hierarchical GA is being developed, composed by a series of two GAs, that is $GA_{12} = \{GA_1 + GA_2\}$, each one having a distinct fitness function corresponding to:

$$F_1(u) = R\{u\} = \int_0^L \dot{u}^2\,dt, \tag{6a}$$

$$F_2(u) = S\{A[u]-f\} = \int_0^L \{A[u]-f\}^2\,dt. \tag{6b}$$

For the calculation of a fractional derivative of order α, D^{α}, expression (6b) needs to be modified, namely, with the introduction of the fractional antidiferentiation operator $A[u(t)] = I^{\alpha}[u(t)]\big|_0^t$.

The GA population is constituted by a series of candidate values $\mathbf{U} = [u_i]$, established at the discrete sampling points $\mathbf{T} = [t_i]$, $i = 0, \ldots, n$, and the evolution consists in a loop of iterations of the GA according to the pseudo-code:

1. Choose the initial population
2. Repeat
   2.1. Execute N_1 iterations of GA_1 with fitness function F_1
   2.2. Execute N_2 iterations of GA_2 with fitness function F_2
3. Until termination

where termination occurs for a total number of iterations N_{12} proportional to $N_1 + N_2$.

3.1 Optimization with one genetic algorithm

In the experiments, the GA adopts a population of $P = 100$ individuals, mutation probability $p_m = 0.1$, single point crossover and reproduction within all population considering elitism. Moreover, a number of sampling points is considered to be $n = 30$, the function $f(t) = t$ is defined over the interval $t \in [0, 1]$, and an additive noise is given by a uniform probability density function in the interval $[-X, +X]$.

The GA performance is sensitive to the number of iterations N and the noise amplitude X. Therefore, in Fig. 1 we analyze the GA performance for $N = \{10^3, 10^5\}$ iterations, noise amplitude $X = \{0.0, 10^{-2}\}$ and regularization parameter $a = \{10^{-2}, 10^{-1}\}$.

The evaluation of a derivative of order $\alpha = 1/2$ through the series approximation (3) is considered, where $T = 1/n$ and $r = n = 30$ in order to avoid truncation errors. For initialization, it is considered that $f(t) = 0, t < 0$, and, consequently, that additive noise affects $f(t)$ only in the interval $t \in [0, 1]$. Moreover, due to the stochastic nature of the evolutionary schemes, the experiments are repeated for $N_T = 10$ cases with different initial random GA populations.

We verify that the GA requires a large number of iterations N to produce a good estimate of the derivative. Furthermore, it is visible that the value adopted for the regularization parameter a perturbs the results, similarly to what occurs in classical optimization methods, requiring several experiments to get the best tuning. In order to avoid the coupling between the two optimization criteria, in the next sub-section we analyze the performance of the series of two GAs, each with its own distinct optimization fitness.

3.2 Optimization with a series of two genetic algorithms

In the experiments, the hierarchical GA_{12} adopts $N_1 = N_2 = 1$ iterations, a population of $P = 100$ individuals, mutation probability $p_m = 0.1$, single point crossover and reproduction within all population considering elitism. For comparison purposes the operating conditions are maintained similar to those adopted previously. Identical operating conditions are adopted, namely, the number of sampling points $n = 30$, the function $f(t) = t$, $t \in [0, 1]$, and the additive noise $[-X, +X]$. We investigate the GA_{12} performance for $N_{12} = \{10^3, 10^4, 10^5\}$ iterations and $X = \{0.0, 10^{-2}\}$. Moreover, a derivative of order $\alpha = 1/2$ is evaluated through the series approximation (3), where $T = 1/n$ and $r = n = 30$ in order to avoid truncation errors. For initialization, it is considered that $f(t) = 0$, $t < 0$, and that noise affects $f(t)$ only in the interval $t \in [0, 1]$. The experiments are repeated for $N_T = 10$ cases with different initial random GA populations.

Figure 2 depicts the results of the new computational scheme for the case of $f(t)$ without any noise (i.e., $X = 0.0$) and $N_{12} = \{10^3, 10^5\}$ iterations. We verify again that the GA has a poor performance for a low number of iterations, but it captures adequately the derivative when a high number of iterations are executed leading to a chart very close to the theoretical value of $D^{1/2}t = \frac{2}{\sqrt{\pi}} t^{1/2}$. The slow GA convergence is, in fact, due to the requirement posed by the series of the two distinct fitness functions.

Figure 3 shows the corresponding result for an additive noise with amplitude $X = 0.01$ and $N_{12} = \{10^4, 10^5\}$ iterations. We observe that noise poses more stringent requirements; nevertheless, after a sufficient number of iterations, we get good results.

The GA scheme is, obviously, not restricted to the calculation of fractional derivatives. Nevertheless, for derivatives with a higher order α, that lead to discontinuities, the algorithm reveals convergence difficulties in the neighborhood of these points. This problem is known to occur for the classical optimization methods, and the substitution of $R\{u\} = \int_0^L \dot{u}^2 \, dt$ by $R\{u\} = \int_0^L |\dot{u}| \, dt$ was proposed [22]. Several experiments with the GA revealed that such assumption is not valid and that, in fact, it leads to inferior results. Therefore, we decided to evaluate the performance of fitness functions of the type $F_1(u) = R\{u\} = \int_0^L |\dot{u}|^{\beta} \, dt$, $\beta > 0$. The first three charts of Fig. 4 depict the first-order derivative, D^1, for noise amplitude $X = 0.01$ and $N_{12} = 10^5$ iterations, when $\beta = \{1, 2, 4\}$. We verify that the best result occurs for $\beta = 4$; therefore, we may consider the future formulation of an automatic

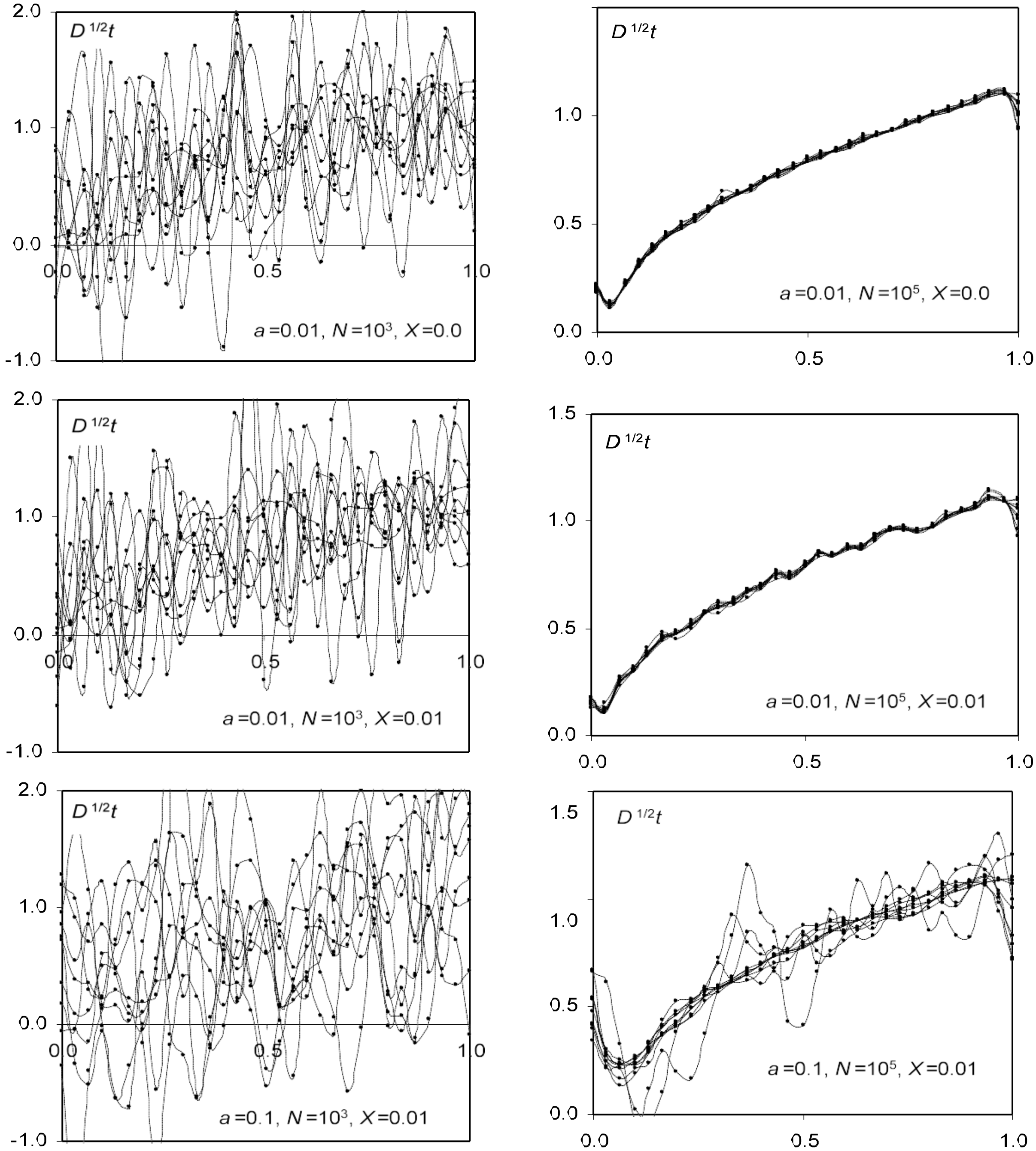

Fig. 1 Chart of $D^{1/2}t$, $t \in [0, 1]$, $X = 0.0$, for $N = \{10^3, 10^5\}$ iterations, $a = \{10^{-2}, 10^{-1}\}$ ($n = 30$, $N_T = 10$)

adjustment method that evaluates the best β for a given function and derivative. Yet another possible strategy is simply to have a large number of GA iterations. The fourth case in Fig. 4 shows the first-order derivative, D^1, for $X = 0.01$, $\beta = 2$ and $N_{12} = 10^6$ iterations.

The evolutionary optimization scheme is also not limited to the function $f(t) = t$. For example, using expression (6a), Fig. 5 depicts the results for $D^{1/2}$ in the case of $f(t) = t^2$, $t \in [0, 1]$, when $X = 0.01$ and $N_{12} = 10^5$ iterations.

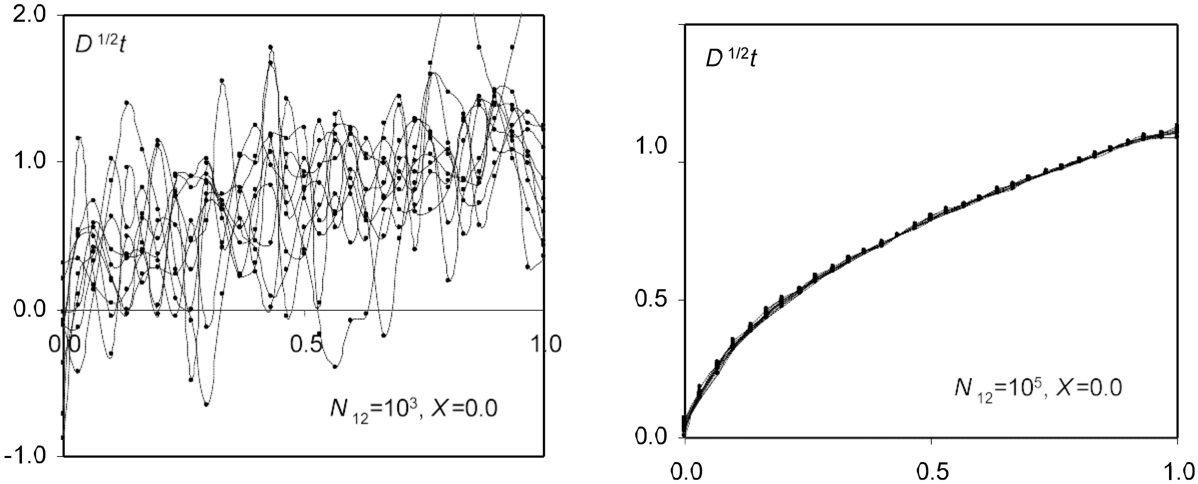

Fig. 2 Chart of $D^{1/2}t$, $t \in [0, 1]$, $X = 0.0$, for $N_{12} = \{10^3, 10^5\}$ iterations ($n = 30$, $N_T = 10$)

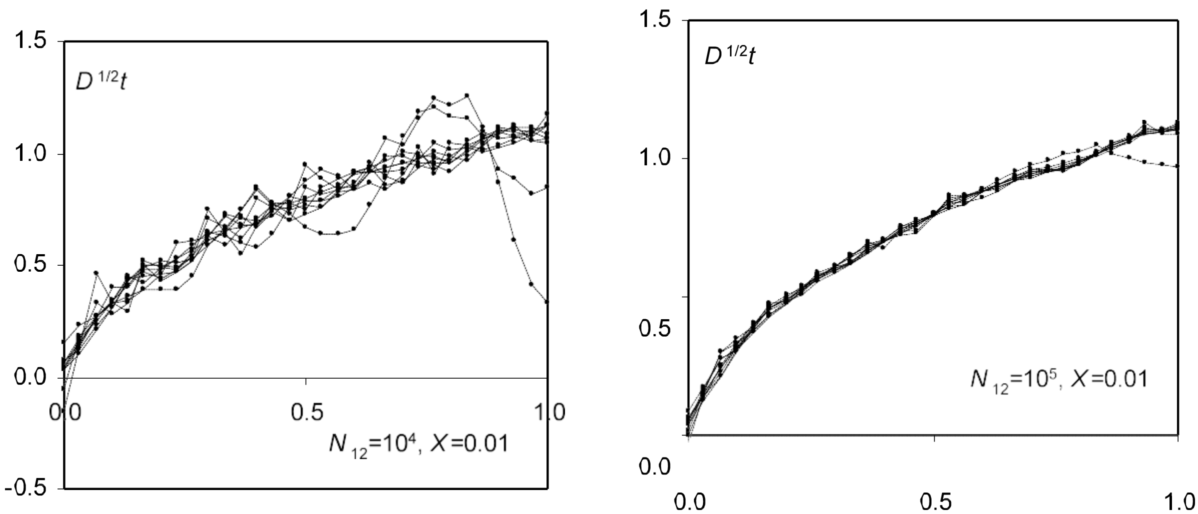

Fig. 3 Chart of $D^{1/2}t$, $t \in [0, 1]$, $X = 0.01$, for $N_{12} = \{10^4, 10^5\}$ iterations ($n = 30$, $N_T = 10$)

As expected, we verify that the chart follows the expression $D^{1/2}t^2 = \frac{8}{3\sqrt{\pi}} t^{3/2}$, $t \in [0, 1]$.

The calculation of derivatives through the minimization of the functionals (5) and (6) consists in finding its solution in the perspective of an inverse problem formulation. This strategy leads to the requirement of an optimization algorithm, either classical or evolutionary. Therefore, as often occurs in optimization problems, the calculation poses a considerable computational burden, making it not adapted to real- time applications. Moreover, the inverse problem formulation is suited to the cases where $\alpha > 0$ and, therefore, the case of $\alpha < 0$ is not addressed since it represents simply the numerical calculation of integrals.

4 Conclusions

The recent advances in fractional calculus point toward important developments in the application of this mathematical concept. During the last years, several algorithms for the approximate calculation of fractional derivatives and integrals were proposed. Nevertheless, the real case of data with noise was somewhat overlooked. In this paper, a new method, based on evolutionary concepts for the calculation of fractional derivatives, was proposed. In this line of thought, an optimization formulation and a hierarchical genetic algorithm were introduced, consisting in a series of two GAs capable of handling the distinct requirements

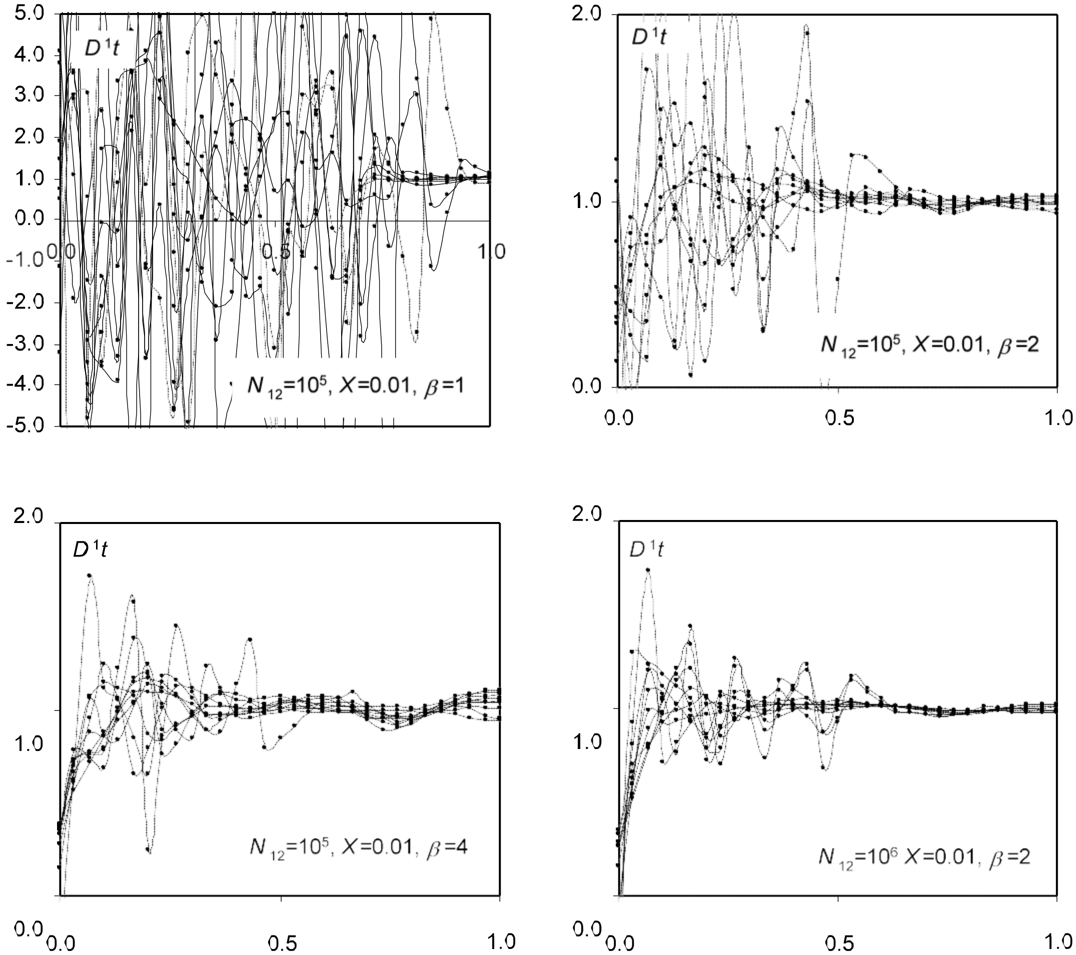

Fig. 4 Chart of $D^1 t$, $t \in [0, 1]$, $X = 0.01$ for $\beta = \{1, 2, 4\}$, $N_{12} = 10^5$ iterations, and $\beta = 2$, $N_{12} = 10^6$ ($n = 30$, $N_T = 10$)

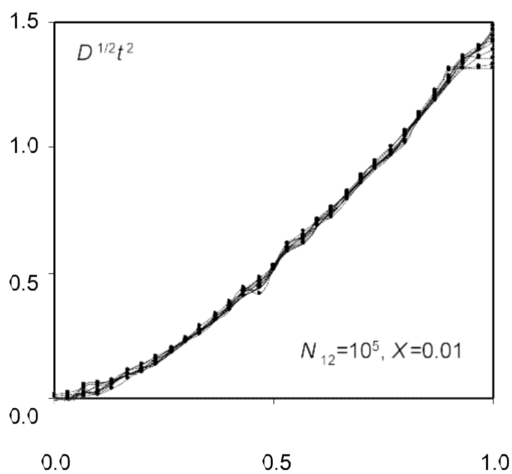

Fig. 5 Chart of $D^{1/2} t^2$ and $D^1 t^2$, $t \in [0, 1]$, $X =$ 0.01,
$N_{12} = 10^5$ iterations ($n = 30$, $N_T = 10$)

posed by the derivative calculation and the noise elimination. The results demonstrate the excellent performance, namely, the convergence and the robustness for high levels of noise.

References

1. Oldham, K.B., Spanier, J.: The Fractional Calculus: Theory and Application of Differentiation and Integration to Arbi- trary Order. Academic Press, San Diego (1974)
2. Ross, B.: Fractional calculus. Math. Mag. **50**, 15–122 (1977)
3. Samko, S.G., Kilbas, A.A., Marichev, O.I.: Fractional Integrals and Derivatives: Theory and Applications. Gordon and Breach, New York (1993)

4. Miller, K.S., Ross, B.: An Introduction to the Fractional Calculus and Fractional Differential Equations. Wiley, New York (1993)

5. Bagley, R.L., Torvik, P.J.: Fractional calculus—a different approach to the analysis of viscoelastically damped struc- tures. AIAA J. **21**, 741–748 (1983)
6. Oustaloup, A.: La commande CRONE: Commande Robuste d'Ordre Non Entier. Hermes (1991)
7. Mainardi, F.: Fractional relaxation-oscillation and fractional diffusion-wave phenomena. Chaos Solitons Fractals **7**, 1461–1477 (1996)
8. Machado, J.T.: Analysis and design of fractional-order dig- ital control systems. J. Syst. Anal. Model. Simul. **27**, 107– 122 (1997)
9. Podlubny, I.: Fractional Differential Equations. Academic Press, San Diego (1999)
10. Machado, J.T.: Discrete-time fractional-order controllers. J. Fract. Calc. Appl. Anal. **4**, 47–66 (2001)
11. Nigmatullin, R.R.: A fractional integral and its physical in- terpretation. Theor. Math. Phys. **90**, 242–251 (1992)
12. Rutman, R.S.: On the paper by R.R. Nigmatullin "A frac- tional integral and its physical interpretation". Theor. Math. Phys. **100**, 1154–1156 (1994)
13. Tatom, F.B.: The relationship between fractional calculus and fractals. Fractals **3**, 217–229 (1995)
14. Yu, Z., Ren, F., Zhou, J.: Fractional integral associated to generalized cookie-cutter set and its physical interpretation. J. Phys. A: Math. Gen. **30**, 5569–5577 (1997)
15. Adda, F.B.: Geometric interpretation of the fractional deriv- ative. J. Fract. Calc. **11**, 21–52 (1997)
16. Moshrefi-Torbati, M., Hammond, J.K.: Physical and geo- metrical interpretation of fractional operators. J. Franklin Inst. B **335**, 1077–1086 (1998)
17. Podlubny, I.: Geometrical and physical interpretation of fractional integration and fractional differentiation. J. Fract. Calc. Appl. Anal. **5**, 357–366 (2002)
18. Machado, J.T.: A probabilistic interpretation of the fractional-order differentiation. J. Fract. Calc. Appl. Anal. **6**, 73–80 (2003)
19. Stanislavsky, A.A.: Probabilistic interpretation of the inte- gral of fractional order. Theor. Math. Phys. **138**, 418–431 (2004)
20. Li, J.: General explicit difference formulas for numerical differentiation. J. Comput. Appl. Math. **183**, 29–52 (2005)
21. Knowles, I., Wallace, R.: A variational method for numeri- cal differentiation. Numer. Math. **70**, 91–110 (1995)
22. Chartrand, R.: Numerical differentiation of noisy, non-smooth data. Los Alamos National Laboratory, December 13 (2005)
23. Ahnert, K., Abel, M.: Numerical differentiation: local ver- sus global methods. Comput. Phys. (2006)
24. Le, T., Chartrand, R., Asaki, T.J.: A variational approach to reconstructing images corrupted by Poisson noise. J. Math. Imaging Vis. **27**, 257–263 (2007)
25. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
26. Goldenberg, D.E.: Genetic Algorithms in Search Optimiza- tion, and Machine Learning. Addison–Wesley, Reading (1989)
27. Machado, J.T., Galhano, A.: Numerical calculation of frac- tional derivatives of non-smooth data. In: ENOC 2008—6th EUROMECH Conference, Saint Petersburg, Russia (2008)
28. Hanke, M., Scherzer, O.: Inverse problems light: numerical differentiation. Am. Math. Mon. **108**, 512–521 (2001)
29. Wang, J.: Wavelet approach to numerical differentiation of noisy functions. Commun. Pure Appl. Anal. **6**, 873–897 (2007)