



ASME Accepted Manuscript Repository

Institutional Repository Cover Sheet

Lukas Burkhardt

ASME Paper Title: An Improvement of a Nonclassical Numerical Method for the Computation of

Fractional Derivatives

Authors: Kai Diethelm

ASME Journal Title: Journal of Vibrations and Acoustics

Volume/Issue 131, 1

Date of Publication (VOR* Online) January 8, 2009

ASME Digital Collection URL: <https://asmedigitalcollection.asme.org/vibrationacoustics/article-abstract/131/1/014502/471073/An-Improvement-of-a-Nonclassical-Numerical-Method>

DOI: <https://doi.org/10.1115/1.2981167>

*VOR (version of record)

An Improvement of a Nonclassical Numerical Method for the Computation of Fractional Derivatives

Kai Diethelm^{*†}

Abstract

Standard methods for the numerical calculation of fractional derivatives can be slow and memory-consuming due to the non-locality of the differential operators. Yuan and Agrawal have proposed a more efficient approach for operators whose order is between 0 and 1 that differs substantially from the traditional concepts. It seems however that the accuracy of the results can be poor. We modify the approach, adapting it better to the properties of the problem, and show that this leads to a significantly improved quality. Our idea also works for operators of order greater than 1.

Keywords: fractional derivative; numerical method; Yuan-Agrawal method

1 Introduction

Caputo-type fractional differential operators of order $\alpha > 0$ with $\alpha \notin \mathbb{N}$, denoted and defined by

$$D_*^\alpha y(x) := \frac{1}{\Gamma(\lceil \alpha \rceil - \alpha)} \int_0^x (x - \tau)^{\lceil \alpha \rceil - \alpha - 1} y^{(\lceil \alpha \rceil)}(\tau) d\tau, \quad (1)$$

have proven to be important tools in the mathematical modeling of many phenomena in physics and engineering, see, e.g., [1, 2, 3, 4, 5]. Here, $\lceil \cdot \rceil$ denotes the ceiling function that rounds up to the nearest integer whereas the floor function $\lfloor \cdot \rfloor$ rounds down.

The large number of potential applications of such operators imply a substantial demand for efficient schemes for their numerical handling. A number of methods for the solution of this problem have been proposed [6, 7, 8, 9]. All these methods essentially deal with the non-locality of the operator in the same way, namely by sampling the function y (or y') on a more or less regular grid in $[0, x]$. Typically one needs to find approximations for $D_*^\alpha y(x)$ for x in a certain interval $[0, X]$, i.e. one has to compute these values for many different choices of x . Since these computations are expensive in terms of run time and memory required for storing those function values that are re-used, the process is very costly. For a related problem, Liang et al. [10, 11] have proposed a substantially

^{*}Institut Computational Mathematics, Technische Universität Braunschweig, Pockelsstr. 14, 38106 Braunschweig, Germany

[†]GNS Gesellschaft für Numerische Simulation mbH, Am Gaußberg 2, 38114 Braunschweig, Germany

different approach that can be transferred to certain special cases of our problem here; however their method heavily uses Laplace transform techniques and is therefore essentially restricted to the treatment of linear problems.

A recent very general nonclassical approach to overcome these difficulties is due to Yuan and Agrawal [12]. The computational cost of the algorithm is evident [4, 13, 12], but reports on the quality of the results produced by this scheme give an unclear picture. Some authors apparently have used it successfully [12]; others criticize it [4, 13] and support their point of view by arguments based on a mechanical interpretation of the method [13]. We aim to collect information on the structure and the behavior of the algorithm and to explain how these properties lead to the unsatisfactory accuracy of the original method. Using these observations we will propose a relatively simple modification of the algorithm and demonstrate that this revised scheme behaves significantly better.

2 The original method

Let us first recall the precise definition of the original Yuan-Agrawal method from [12]. Whereas they have only discussed the case $0 < \alpha < 1$, our approach covers the general case of arbitrary positive $\alpha \notin \mathbb{N}$. For $0 < \alpha < 1$ we recover the original scheme.

We assume our function $y \in C^{[\alpha]}[0, X]$ in (1) and the order $\alpha > 0$, $\alpha \notin \mathbb{N}$, of the differential operator to be given. Then, the initial value problem

$$\begin{aligned} \frac{d}{dx}\phi(w, x) &= -w^2\phi(w, x) \\ &\quad + \frac{2 \sin \pi \alpha}{(-1)^{\lfloor \alpha \rfloor} \pi} w^{2\alpha - 2\lfloor \alpha \rfloor + 1} y^{([\alpha])}(x), \\ \phi(w, 0) &= 0, \end{aligned} \tag{2}$$

for a fixed $w > 0$ contains an ordinary differential equation (because w is fixed). Moreover it is a linear and inhomogeneous differential equation of order 1 with constant coefficients. Its solution ϕ is found to be

$$\phi(w, x) = \frac{2 \sin \pi \alpha}{(-1)^{\lfloor \alpha \rfloor} \pi} w^{2\alpha - 2\lfloor \alpha \rfloor + 1} \int_0^x \frac{y^{([\alpha])}(\tau)}{e^{-(x-\tau)w^2}} d\tau. \tag{3}$$

The relation between this function ϕ and the Caputo derivative of y is very simple; it has the form

$$D_*^\alpha y(x) = \int_0^\infty \phi(w, x) dw. \tag{4}$$

This can be shown using the method of [13, §2]. Based on eqs. (4) and (2), Yuan and Agrawal [12, §3] use two parameters $h > 0$ and $n \in \mathbb{N}$ to be given by the user and proceed in two steps to construct their approximation for the fractional derivative $D_*^\alpha y(x)$.

In the first step, let z_{kn} be the k th zero of the Laguerre polynomial of degree n , assuming the ordering $z_{1n} < z_{kn} < \dots < z_{nn}$. Then, Yuan and Agrawal compute an approximate solution $\phi_h(z_{kn}, x)$ of the differential equation (2) by a suitable numerical method for all k and for all x for which an approximate value of $D_*^\alpha y(x)$ is desired. This approach avoids to use the explicit representation (3) of ϕ . When choosing a numerical method for the solution of the differential equation, the user has to take into account one important point: Equation (2) needs to be solved for $w = z_{kn}$, $k = 1, 2, \dots, n$. The absolute value of the

coefficient of ϕ on the right-hand side of eq. (2), viz. the number w^2 , can be very large as $n \rightarrow \infty$ for some values of k ; e.g., $w = z_{nn} \approx 4n$ [14, eq. (6.32.4)], and thus $w^2 \approx 1600$ already for $n = 10$. Explicit methods are likely to fail under such conditions because of their insufficient stability unless the step sizes are chosen extremely small [15, Chap. IV]. To be on the safe side even for relatively large (and computationally cheaper) step sizes, we will only use A-stable implicit methods.

The second step performs the actual computation of the fractional derivative by means of eq. (4) but using the just computed approximation ϕ_h instead of the exact function ϕ in the integral on the right-hand side. The integration itself is also done numerically by writing

$$\int_0^\infty \phi_h(w, x) dw = \int_0^\infty e^{-w} [e^w \phi_h(w, x)] dw$$

and using an n -point Gauss (more precisely, Gauss-Laguerre) quadrature rule [14, §3.4] for the weight e^{-w} with respect to the factor in brackets, thus obtaining

$$D_{n,h}^\alpha y(x) := \sum_{k=1}^n w_{kn} e^{z_{kn}} \phi_h(z_{kn}, x), \quad (5)$$

with w_{kn} denoting the Laguerre weights, as the final approximation for $D_*^\alpha y(x)$.

It has been reported [4, 13] that the accuracy of the results of the original Yuan-Agrawal method is rather poor. Moreover, Schmidt and Gaul [13, §3] have applied this method to a fractional model of viscoelasticity and demonstrated that this is equivalent to using a classical viscoelasticity model with n Maxwell chains whose relaxation times are w_{kn}^2 , $k = 1, 2, \dots, n$. Thus the relaxation times cannot be chosen by the user, and they are distributed only over a relatively small subinterval of $(0, \infty)$. Schmidt and Gaul have solved the model problem

$$cD_*^\alpha x(t) + kx(t) = f(t), \quad x(0) = 0, \quad t \in [0, 1.35 \cdot 10^9], \quad (6)$$

(the equation of motion of a massless one-dimensional fractional Kelvin-Voigt model subject to the external force $f(t)$ [13, §2]) with $c = 100$, $k = 10$, $\alpha = 0.3$ and

$$f(t) = \begin{cases} 0 & \text{for } t = 0, \\ f_0 & \text{else,} \end{cases} \quad \text{with } f_0 = 1. \quad (7)$$

The equations were solved by the trapezoidal method with mesh points $t_j = 10^{-4} \sum_{k=0}^{j-1} 1.005^k$ ($j = 0, 1, \dots, 5000$). The results were rather poor [13, Figs. 3–5]. For example, the creep process of the exact fractional Kelvin-Voigt model, given by $x(t) = f_0 k^{-1} (1 - E_\alpha(-kt^\alpha/c))$ (E_α is the Mittag-Leffler function of order α), covers more than 14 decades while the Yuan-Agrawal approximation, even for $n = 15$, is limited to less than 8 decades. This reflects the unfavorable distribution of the relaxation times.

To understand and improve the behavior of the scheme it is useful to look at the asymptotics of the integrand function ϕ in eq. (4). In this context, Lu and Hanyga [4, eqs. (32) and (33)] have observed

$$\phi(w, x) \sim w^{2\alpha - 2[\alpha] + 1} \text{ as } w \rightarrow 0, \quad (8)$$

$$\phi(w, x) \sim w^{2\alpha - 2[\alpha] - 1} \text{ as } w \rightarrow \infty, \quad (9)$$

where $a(v) \sim b(v)$ means that $0 < A \leq |a(v)/b(v)| \leq B$ with constants A and B as v tends to the indicated limit.

We can now identify two reasons why the Yuan-Agrawal method cannot work well. Both reasons are related to the fact that the Gauss-Laguerre rule is not really suitable for integrals exhibiting such asymptotics.

Firstly the Laguerre method as originally proposed [12] assumes the integrand to be smooth at the origin, and by eq. (8) this is true if and only if the exponent of w in eq. (8) is an integer, i.e. if $\alpha = k + 1/2$ with some $k \in \mathbb{N}_0$.

There are ways to solve this problem. Lu and Hanyga [4, eq. (34)], e.g., suggest to split the integration interval into $[0, c]$ and $[c, \infty)$ with $c > 0$ and use a Gauss-Jacobi quadrature with weight $w^{2\alpha-2[\alpha]+1}$ for the integral on $[0, c]$. The integral over $[c, \infty)$ is handled by a shifted Laguerre formula, preceded by analytical manipulations accelerating the decay of the integrand as $w \rightarrow \infty$, thus speeding up the convergence of the quadrature formula.

An even simpler idea would be to replace the standard Laguerre formula on $[0, \infty)$, i.e. a Gauss rule with weight function e^{-w} , by its generalization with weight $w^{2\alpha-2[\alpha]+1}e^{-w}$. This reproduces the true asymptotics of the integrand near 0, thus removing the problem there.

The second, and much more serious, problem is the behavior of the integrand $\phi(w, x)$ as $w \rightarrow \infty$ for fixed x . It decays as $w^{2\alpha-2[\alpha]-1}$, see eq. (9). The exponent of w here is contained in $(-3, -1)$. This is sufficient to make sure that the improper integral exists, but far too slow for a good approximation by expressions of the form $w^\gamma e^{-w} p(w)$ with some polynomial p which is what a generalized Laguerre formula attempts. In contrast to the problem at $w = 0$ discussed above, this difficulty can only be handled by giving up the Gauss-Laguerre principle.

3 An improved method

We thus suggest to replace the Gauss-Laguerre rules by a method proposed by Gautschi [16, §3] and tailored for integrands with an algebraic decay. This approach, applied to our situation, amounts to using the formula

$$\int_0^\infty \phi_h(w, x) dw \approx 2 \sum_{k=1}^n W_{kn} \frac{(1+Z_{kn})^{\bar{\alpha}-2}}{(1-Z_{kn})^{\bar{\alpha}}} \phi_h\left(\frac{1-Z_{kn}}{1+Z_{kn}}, x\right) \quad (10)$$

with $\bar{\alpha} := 2\alpha - 2[\alpha] + 1 \in (-1, 1)$. Here W_{kn} and Z_{kn} are the weights and nodes of the Gauss-Jacobi quadrature rule for the weight function $(1-t)^{\bar{\alpha}}(1+t)^{-\bar{\alpha}}$ which can be computed easily with the standard algorithms of [17]. We may then expect a much faster convergence of the quadrature than in the original algorithm.

The smallest node of the Gauss-Jacobi method behaves as $-1 + cn^{-2}$ for $n \rightarrow \infty$ [14, Thm. 8.1.2]. Thus, because of the transformation of variables in our method, the largest point at which our quadrature evaluates ϕ is at $cn^2(1+o(1))$. This is much larger than the value cn of the corresponding point in the Gauss-Laguerre method from the original scheme. The difference between the two sets of nodes and weights is evident from Fig. 1.

We note three consequences of this observation: First, the value of $\phi(w, x)$ is taken into account for much larger values of w which, in view of the slow decay of $\phi(w, x)$ as $w \rightarrow \infty$ observed in eq. (9), is a main reason for the superiority of the Gauss-Jacobi method. Second, explicit methods or not A-stable methods for the solution of the ordinary differential equations are even less useful than

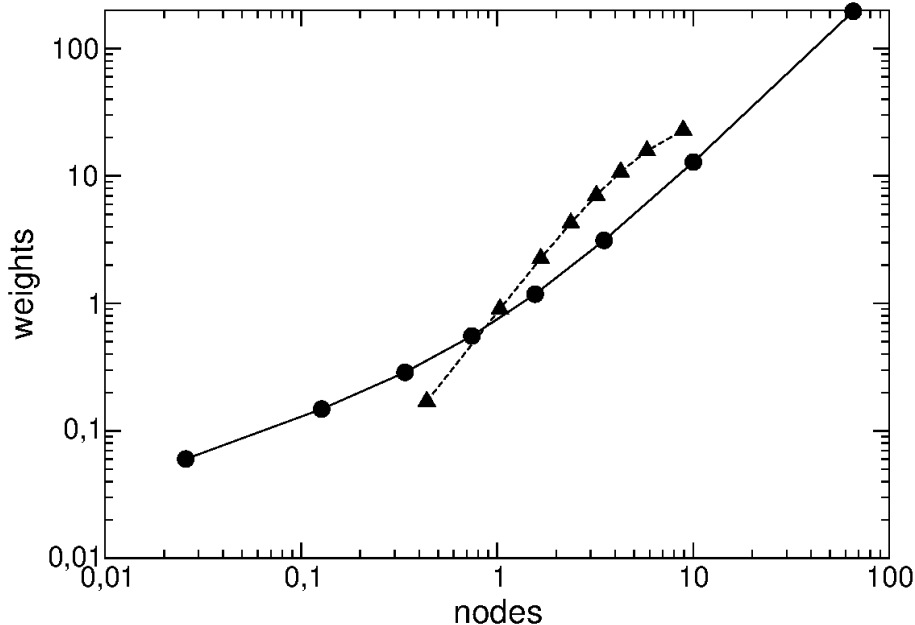


Figure 1: Log-log plot of transformed nodes and weights for 8-point Gauss-Laguerre rule (original method; triangles) and 8-point Gauss-Jacobi rule (modified method; circles).

in the original method. Finally, the fact that the distribution of the nodes now covers a larger interval is very attractive in view of the observations of Schmidt and Gaul concerning the interpretation of the squares of the nodes as relaxation times in a classical viscoelastic model.

Figure 2 shows the results of our modified approach for the problem given in eqs. (6) and (7) without changing any parameters, only replacing the n -point Gauss-Laguerre method proposed by Yuan and Agrawal by our n -point transformed Gauss-Jacobi method described in eq. (10). In particular, the ordinary differential equations generated by our method have been solved in the same way as those of the original method were solved in [13], namely by the trapezoidal algorithm with mesh points $t_j = 10^{-4} \sum_{k=0}^{j-1} 1.005^k$ ($j = 0, 1, \dots, 5000$). Thus the computational cost was unchanged. A comparison with the corresponding results for the original algorithm given in [13, Fig. 3] immediately shows the substantial advantage of the modified scheme.

Another experiment was to exchange the trapezoidal method for the solution of the differential equations in our approach for a backward Euler method, thus replacing a second-order by a first-order scheme. Nevertheless it was impossible to distinguish the resulting plots from those obtained by the modified method with the trapezoidal formula shown in Fig. 2. This emphasizes that the choice of the quadrature rule for the integral is much more important than the choice of the differential equation solver as long as the latter is A-stable and implicit.

Other numerical experiments have shown similar improvements. For the sake of brevity we only report the results of a few of them. The first one is the computation of $D_*^\alpha y(x)$ for $y(x) = x^3$ and $\alpha = 0.6$ on the interval $x \in [0, 1]$. The resulting absolute errors of the original Yuan-Agrawal method, computing the

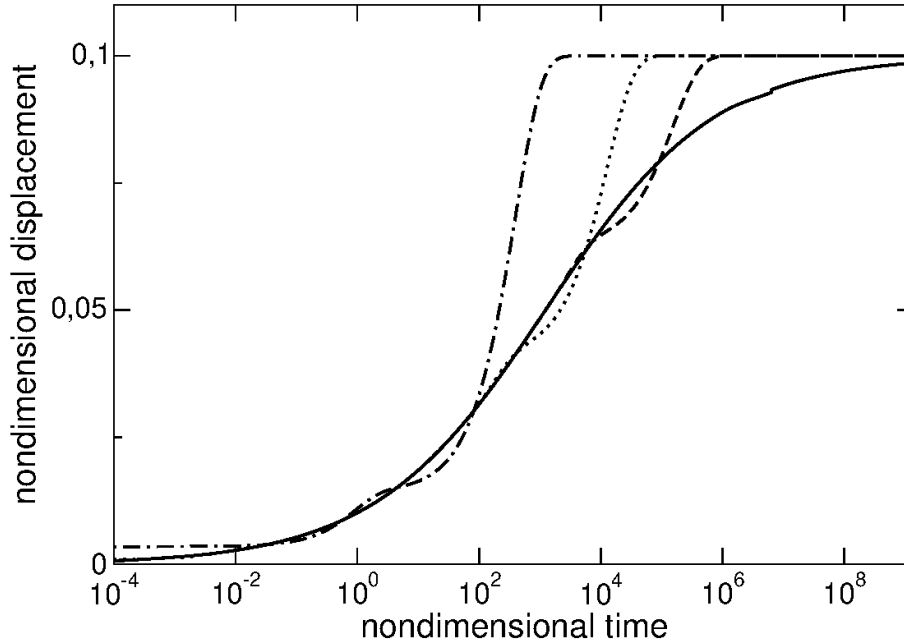


Figure 2: Exact solution (solid line) of eq. (6) with data given in eq. (7) and numerical solution with modified method, using trapezoidal ODE solver and 2 (dash-dotted), 7 (dotted), and 15 (dashed) quadrature points, respectively.

integral in eq. (4) with an n -point Gauss-Laguerre rule with $n \in \{8, 16, 32\}$, are shown in Fig. 3. In each case we have solved the first-order differential equations with trapezoidal methods with step sizes $h = 1/10$ and $h = 1/100$. The error decreases substantially when the number of quadrature nodes is doubled whereas a decrease of the step size of the trapezoidal method shows almost no effects.

We have repeated these examples with our modified algorithm and unchanged parameters. The results are given in Fig. 4. A comparison with Fig. 3 once again shows significant differences. First, by examining the scales on the vertical axes we note that the magnitude of the error is much smaller. Second, we could not distinguish the data for 32 quadrature points from those for 16 nodes graphically, so we only plotted the latter. Moreover now the quadrature error is so small that the error introduced by the trapezoidal differential equation solver dominates the overall error. Thus, in this sense, the roles of the two parts of the error have been reversed when compared to the original algorithm. All these points clearly reveal that the modification of the quadrature formula indeed results in the expected performance improvement.

Similar results can be found in our second example, the numerical computation of $D_*^\alpha y(x)$ for $y(x) = \exp(-x)$ and $\alpha = 1.3$ on the interval $x \in [0, 1]$. In contrast to the previous example, we now deal with a function y with the property $y(0) \neq 0$. Moreover we have chosen $\alpha > 1$ to illustrate that our method works well in this case too. The corresponding graphs are depicted in Fig. 5 for the original Yuan-Agrawal method and in Fig. 6 for our modified version, respectively.

As in our first example we observe that the error of the original method shows a quite strong dependence on the step size of the differential equation

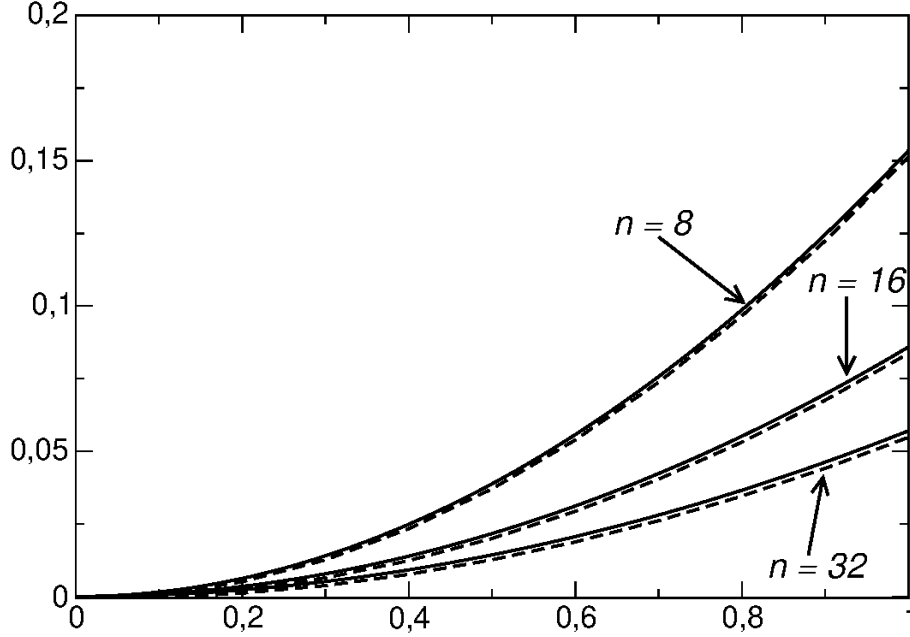


Figure 3: Absolute errors of the original Yuan-Agrawal method for the approximation of $D_*^\alpha y(x)$ with $y(x) = x^3$ and $\alpha = 0.6$ for $x \in [0, 1]$ with 8, 16 and 32 Gauss-Laguerre nodes and trapezoidal method with $h = 1/10$ (dashed) and $h = 1/100$ (solid lines).

solver. Moreover, we find an oscillatory behavior of the total error with an amplitude that decays rapidly as the step size of the differential equation solver decreases.

For the modified algorithm we can once again see errors with a much smaller order of magnitude. The oscillations of the error behave in a similar way as those of the original method.

Finally we aim to solve one more fractional differential equation by our new method, namely

$$D_*^\alpha y(t) = \frac{40320}{\Gamma(9-\alpha)} t^{8-\alpha} - 3 \frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)} t^{4-\alpha/2} + \frac{9}{4} \Gamma(\alpha+1) + \left(\frac{3}{2} t^{\alpha/2} - t^4 \right)^3 - [y(t)]^{3/2}, \quad (11)$$

$$y(0) = y'(0) = 0. \quad (12)$$

The equation is taken from [9, §4.1]; we shall specifically use $\alpha = 1.25$. The exact solution is $y(t) = t^8 - 3t^{4+\alpha/2} + 9t^\alpha/4$, see [9, p. 46]. We have solved it for t in the interval $[0, 1]$ using our modified Yuan-Agrawal method with 5 and 10 Gauss-Jacobi quadrature nodes. The resulting ordinary differential equations have, as usual, been solved by a trapezoidal integrator using the step sizes $h = 1/40$ and $h = 1/80$, respectively. The numerical results shown in Fig. 7 clearly demonstrate that our method can be used successfully to solve this nonlinear differential equation as well. Evidently the fact that the order of the

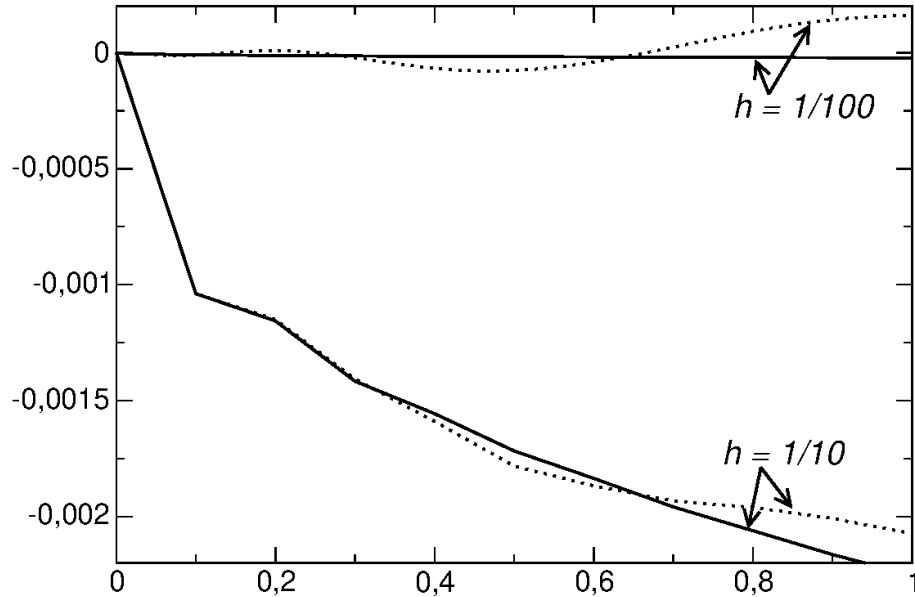


Figure 4: Absolute errors of the modified Yuan-Agrawal method for the approximation of $D_*^\alpha y(x)$ with $y(x) = x^3$ and $\alpha = 0.6$ for $x \in [0, 1]$ with 8 (dotted) and 16 (solid lines) Gauss-Jacobi nodes and trapezoidal method with $h = 1/10$ and $h = 1/100$.

differential operator is now greater than one does not produce any problems either.

4 Summary and Conclusions

The Yuan-Agrawal method for the numerical calculation of fractional derivatives consists of three steps:

- (a) express the fractional derivative as an integral,
- (b) compute the integrand of this integral by numerically solving a first-order initial value problem,
- (c) compute the integral itself numerically using a suitable quadrature formula and the function values of the integrand obtained in step (b).

We have extended all three steps of the idea from $0 < \alpha < 1$ to arbitrary positive non-integers α and identified various options for the user when it comes to selecting the solver for the ordinary differential equations in step (b). While explicit methods are of little practical use, A-stable implicit schemes can be used according to the user's preferences without having to worry about any potential negative influence on the final result. Really crucial is the choice of the quadrature formula in step (c). The Gauss-Laguerre rule suggested by Yuan and Agrawal leads to a rather poor quality of the algorithm because of the asymptotic behavior of the integrand ϕ . Knowing this, we have been able to propose an alternative method that leads to a much better overall performance.

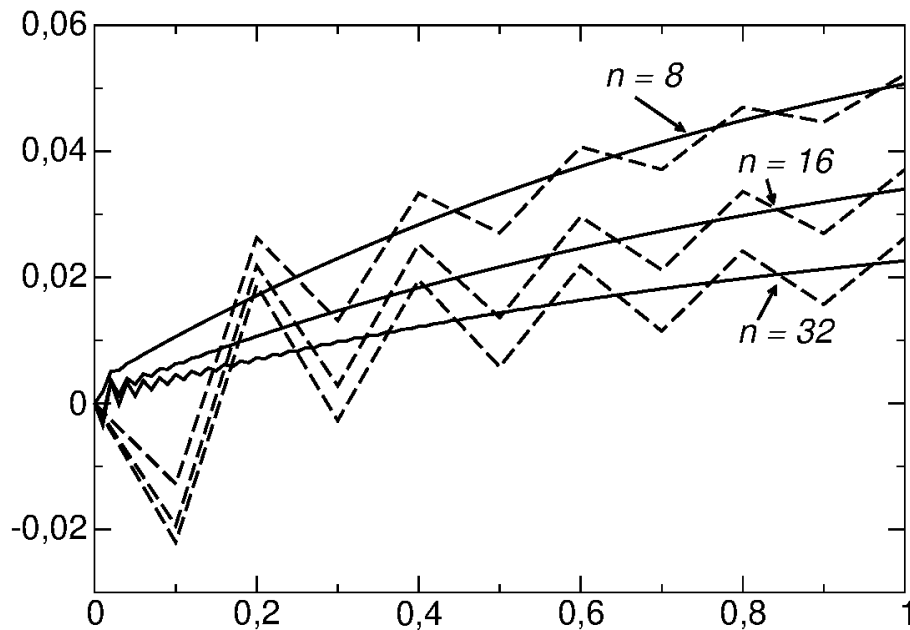


Figure 5: Absolute errors of the original Yuan-Agrawal method for the approximation of $D_*^\alpha y(x)$ with $y(x) = \exp(-x)$ and $\alpha = 1.3$ for $x \in [0, 1]$ with 8, 16 and 32 Gauss-Laguerre nodes and trapezoidal method with $h = 1/10$ (dashed) and $h = 1/100$ (solid lines).

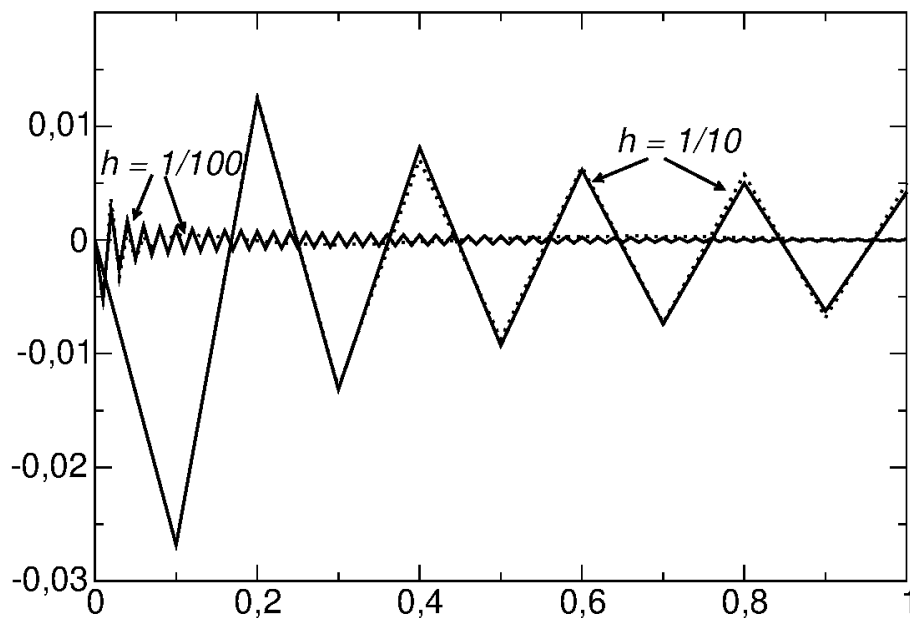


Figure 6: Absolute errors of the modified Yuan-Agrawal method for the approximation of $D_*^\alpha y(x)$ with $y(x) = \exp(-x)$ and $\alpha = 1.3$ for $x \in [0, 1]$ with 8 (dotted) and 16 (solid lines) Gauss-Jacobi nodes and trapezoidal method with $h = 1/10$ and $h = 1/100$.

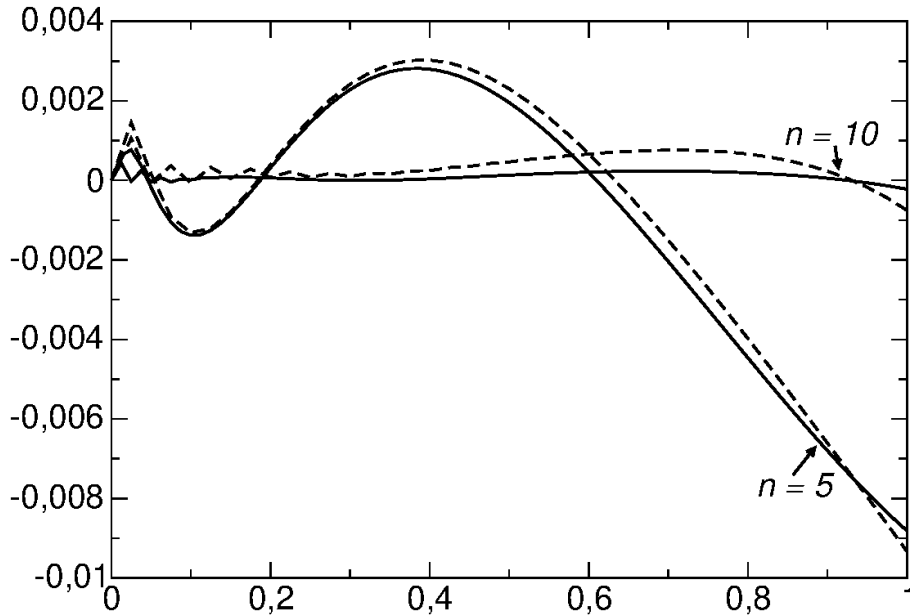


Figure 7: Absolute errors of the modified Yuan-Agrawal method for the solution of eq. (11) with initial condition (12) and $\alpha = 1.25$ for $x \in [0, 1]$ with 5 and 10 Gauss-Jacobi nodes and trapezoidal method with $h = 1/40$ (dashed) and $h = 1/80$ (solid lines).

Acknowledgments

The author thanks André Schmidt (Universität Stuttgart) for the useful discussion and for providing the exact details of the numerical simulations conducted in [13].

References

- [1] Bagley, R. L., and Calico, R. A., 1991, “Fractional order state equations for the control of viscoelastically damped structures,” *J. Guidance Control Dynam.* **14**, pp. 304–311.
- [2] Caputo, M., 1967, “Linear models of dissipation whose Q is almost frequency independent—II,” *Geophys. J. Royal Astronom. Soc.* **13**, pp. 529–539.
- [3] Gorenflo, R., and Mainardi, F., 1997, “Fractional calculus: Integral and differential equations of fractional order,” *Fractals and Fractional Calculus in Continuum Mechanics*, A. Carpinteri and F. Mainardi, eds., Springer, Wien, pp. 223–276.
- [4] Lu, J.-F., and Hanyga, A., 2005, “Wave field simulation for heterogeneous porous media with singular memory drag force,” *J. Comput. Phys.* **208**, pp. 651–674.
- [5] Podlubny, I., 1999, *Fractional Differential Equations*. Academic Press, San Diego.

- [6] Diethelm, K., 1997, “Generalized compound quadrature formulae for finite-part integrals,” *IMA J. Numer. Anal.* **17**, pp. 479–493.
- [7] Elliott, D., 1993, “An asymptotic analysis of two algorithms for certain Hadamard finite-part integrals,” *IMA J. Numer. Anal.* **13**, pp. 445–462.
- [8] Lubich, C., 1986, “Discretized fractional calculus,” *SIAM J. Math. Anal.* **17**, pp. 704–719.
- [9] Diethelm, K., Ford, N. J., and Freed, A. D., 2004, “Detailed error analysis for a fractional Adams method,” *Numer. Algorithms* **36**, pp. 31–52.
- [10] Liang, J., Chen, Y., and Guo, B.-Z., 2004, “A hybrid symbolic-numerical simulation method for some typical boundary control problems,” *Simulation* **80**, pp. 635–643.
- [11] Liang, J., and Chen, Y., 2006, “Hybrid symbolic and numerical simulation studies of time-fractional order wave-diffusion systems,” *Internat. J. of Control* **79**, pp. 1462–1470.
- [12] Yuan, L., and Agrawal, O. P., 2002, “A numerical scheme for dynamic systems containing fractional derivatives,” *J. Vibration Acoustics* **124**, pp. 321–324.
- [13] Schmidt, A., and Gaul, L., 2006, “On a critique of a numerical scheme for the calculation of fractionally damped dynamical systems,” *Mech. Res. Comm.* **33**, pp. 99–107.
- [14] Szegő, G., 1975, *Orthogonal Polynomials*. 4th ed., American Mathematical Society, Providence.
- [15] Hairer, E., and Wanner, G., 1996, *Solving Ordinary Differential Equations II*, 2nd revised ed., Springer, Berlin.
- [16] Gautschi, W., 1991, “Quadrature formulae on half-infinite intervals,” *BIT* **31**, pp. 438–446.
- [17] Gautschi, W., 1994, “Algorithm 726: ORTHPOL — A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules,” *ACM Transact. Math. Software* **20**, pp. 21–62.