International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:12, No:8, 2018

Application of a SubIval Numerical Solver for Fractional Circuits

Marcin Sowa

***Abstract*—**The paper discusses the subinterval-based numerical method for fractional derivative computations. It is now referred to by its acronym – SubIval. The basis of the method is briefly recalled. The ability of the method to be applied in time stepping solvers is discussed. The possibility of implementing a time step size adaptive solver is also mentioned. The solver is tested on a transient circuit example. In order to display the accuracy of the solver – the results have been compared with those obtained by means of a semi-analytical method called gcdAlpha. The time step size adaptive solver applying SubIval has been proven to be very accurate as the results are very close to the referential solution. The solver is currently able to solve FDE (fractional differential equations) with various derivative orders for each equation and any type of source time functions.

***Keywords*—**Numerical method, SubIval, fractional calculus, numerical solver, circuit analysis.

I. Introduction

FRACTIONAL calculus is a field dealing with the concept of fractional order derivatives and integrals or, generally – differintegrals [1].

It has been lately a popular topic due to its many applications that are tested in various fields e.g.

- control systems, where fractional PID controllers are studied [2], [3],
- electromagnetism – where materials with complex properties are modeled [4], [5],
- circuit theory (which this paper concerns), where fractional coils can be useful in modeling coils with ferromagnetic cores [6], [7] and fractional capacitors are useful in modeling supercapacitors [8], [9],
- viscoelasticity analyses [10], [11],
- biomedical signal analyses [12], [13].

Over the years many definitions of fractional derivatives have been proposed [14]. This study concerns one of the most popular ones – proposed by Caputo in 1967 [15]. Only fractional derivatives of order $\alpha \in (0, 1]$ are considered. They can be described by the equation:

$$ {}_{t_\mathrm{a}}\mathrm{D}^{\alpha}_{t_\mathrm{b}} x(t) = \frac{1}{\Gamma(1-\alpha)} \int_{t_\mathrm{a}}^{t_\mathrm{b}} \frac{x^{(1)}(\tau)}{(t-\tau)^{\alpha}} \mathrm{d}\tau. \tag{1} $$

In most cases when the above equation appears t_a is the initial time instance and t_b is equal to a "current" time instance t. The above generalized differintegral is introduced for further use. Also, the study uses an interval-based notation, where $\Xi = [t_\mathrm{a}, t_\mathrm{b}]$ and hence (1) can be written as:

$$ \mathrm{d}^{\alpha}_{\Xi} x(t) = {}_{t_\mathrm{a}}\mathrm{D}^{\alpha}_{t_\mathrm{b}} x(t) \tag{2} $$

M. Sowa is with the Faculty of Electrical Engineering, Silesian University of Technology, 44-100 Gliwice, Poland (e-mail: marcin.sowa@polsl.pl).

The possibilities, through which one can solve problems with appearing fractional derivatives could be divided into three main categories:

- evaluations of analytical solutions – if a problem is simple enough then it can be put in the form of fractional state equations, which have a general solution basing on the Mittag-Leffler function [16],
- semi-analytical methods [17], [18] – also aiming at deriving formulae for solutions rather than introducing approximations at an early stage,
- numerical methods [19]-[21] – which often consider approximations at an early stage; numerical methods are often able to solve the widest variety of problems.

This study concerns a numerical method, while analytical and semi-analytical methods are used only for verification when the method is modified and exemplary solutions are checked.

II. The Study of SubIval

This study revolves around the subinterval-based numerical method, which has been first proposed in the paper [22]. It is now more often referred to by its acronym – SubIval.

Over these couple of years improvements have been made to the method [23]-[25] leading to better efficiency and allowing faster computations.

The method relies on a partition of the interval $\Xi = [t_0, t_\mathrm{now}]$ to M subintervals:

$$ \mathrm{d}^{\alpha}_{\Xi} x(t) = \sum_{k=1}^{M} \mathrm{d}^{\alpha}_{\Xi_k} x(t), \tag{3} $$

where the intervals denoted by Ξ_k $(k = 1, 2, \ldots M)$ form a continuous set of closed subintervals.

The method uses approximations in each of these subintervals, which results in the general formula:

$$ \mathrm{d}^{\alpha}_{\Xi} x(t) \approx \sum_{k=1}^{M} \mathrm{d}^{\alpha}_{\Xi_k} \widetilde{x}_k(t). \tag{4} $$

For a more detailed description of SubIval – the reader is referred to the paper [26]. There it is also explained how the subintervals are established (this is done through an original algorithm).

SubIval has been designed with the thought of working well in a typical time-stepping solver, where the variables computed at each time instance $t = t_\mathrm{now}$ are treated as unknowns but values at previous time steps are treated as known values and are not modified anymore. Through this the implicit formula can be applied:

$$ \mathrm{d}^{\alpha}_{\Xi} x(t) \approx a x_\mathrm{now} + b, \tag{5} $$

where x_{now} is the value of $x(t)$ computed for $t = t_{\mathrm{now}}$. The values a and b are computed through polynomial differintegration, which can be done using analytical formulae as described e.g. in [27].

III. Application of SubIval in a Time-Stepping Solver

Currently SubIval is implemented in a DLL available at [29]. It can aid in solving FDAE (fractional differential algebraic equations) in the general form:

$$\begin{cases} \mathrm{d}^{\boldsymbol{\alpha}}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), t), \\ 0 = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), t), \end{cases} \tag{6}$$

where $\boldsymbol{x}(t)$ is the vector of state variables, $\boldsymbol{y}(t)$ contains any remaining computed variables, $\mathrm{d}^{\boldsymbol{\alpha}}\boldsymbol{x}(t)$ is a vector of the fractional derivatives (with $\boldsymbol{\alpha}$ representing a vector of the derivative orders).

The way it can be applied can be explained through the following steps of the solver:

(1) define the initial values for each state variable x_i; define the order α_i of the fractional derivative $\mathrm{d}^{\alpha_i}_{\Xi} x_i(t)$; define the initial time instance t_0; send the information on these values to the SubIval DLL,
(2) set the time step index $j = 1$
(3) set the initial time step Δt,
(4) define the time instance $t_1 = t_0 + \Delta t$,
(5) send t_j to the SubIval DLL,
(6) for each $\mathrm{d}^{\alpha_i}_{\Xi} x_i(t)$ obtain a and b and apply the approximation (5) – this step leads to a system of equations,
(7) solve the system of equations to obtain $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ for the current time instance,
(8) send the obtained solution for $\boldsymbol{x}(t)$ to the SubIval DLL,
(9) increment j,
(10) apply the choice for the next time instance $t_j = t_{j-1} + \Delta t$,
(11) if t_{j-1} is less than the selected ending time of the analysis then return to step (5), otherwise end the process.

IV. Step Size Adaptive Solver

The time stepping solver can be improved so that Δt is adjusted according to an error estimation. It has been proven in a parallel study [28] that the error of the SubIval computations can be approximated through the formula:

$$e \approx c\Delta t^{p-\alpha}, \tag{7}$$

where p is the polynomial order of the SubIval computations (more on the polynomial orders can be found e.g. in the paper [26]) and α is the derivative order of the variable for which the error is estimated. According to this error estimation the time step size is modified through its multiplication by the coefficient:

$$\eta = \sqrt[p-\alpha]{\frac{e}{e_{\mathrm{ctrl}}}}. \tag{8}$$

In the above equation e is the relative difference between differintegrals computed for two different polynomial orders (the differintegral is computed close to $t = t_{\mathrm{now}}$ – more on this in [28]) and e_{ctrl} is a measure of the desired accuracy.

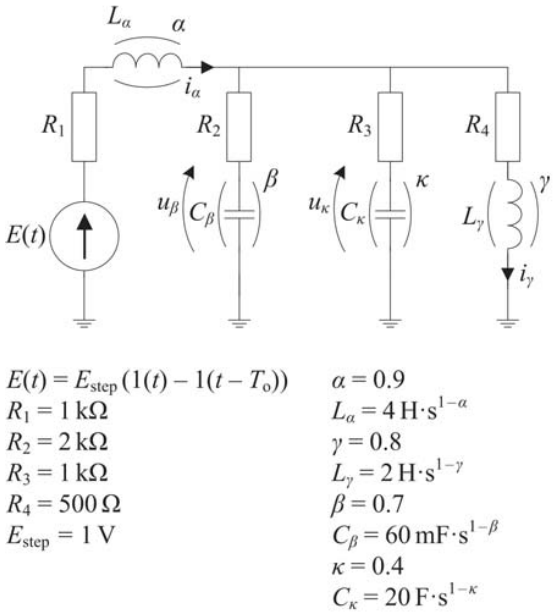

Fig. 1 Exemplary transient circuit example (the fractional elements are denoted by parentheses and the order of the element)

V. Solvers for Octave and Matlab

There are solvers available at [29] that base on SubIval. Both a simple constant time step size solver and an adaptive solver have been implemented. They have been implemented with the thought to work both in Matlab [30] and its freeware alternative – GNU Octave [31].

The solvers are so far implemented to solve the following system of FDE (fractional differential equations):

$$\mathrm{d}^{\boldsymbol{\alpha}}\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{v}(t), \tag{9}$$

where $\boldsymbol{x}(t)$ contains N state variables, $\boldsymbol{v}(t)$ is the vector of M source time functions, $\boldsymbol{A}$ is an $N \times N$ matrix and $\boldsymbol{B}$ is an $N \times M$ matrix.

VI. Computational Example

A simple circuit example has been brought forth to display the usefulness of the SubIval adaptive time step size solver. The problem is presented in Fig. 1.

For the demonstration a circuit has been deliberately chosen, for which one can obtain a solution by means of another method – one which is both easily applicable and has a completely different basis than that of SubIval. Such a method is the semi-analytical one called gcdAlpha – it has been discussed in [32] and is applied to obtain the referential solution.

The formulation of the circuit equations and how the terms in (9) are obtained is explained in the Appendix.

The gcdAlpha method itself sometimes requires a significant amount of time to solve a problem in comparison to the SubIval solver; however, it is a method only designed in order to produce referential solutions when designing the SubIval solver. Hence, its accuracy is important, but the computation speed – not as much.

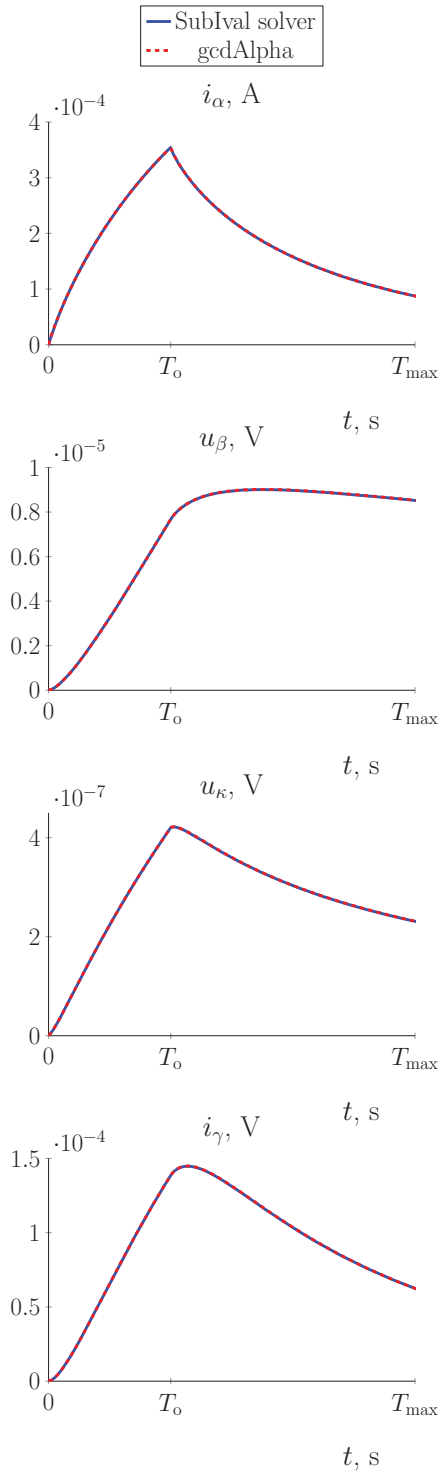

Fig. 2 Comparison of results (state variable time functions) obtained with the SubIval solver and the gcdAlpha method

A comparison of the results obtained with the SubIval solver and the gcdAlpha method is depicted in Fig. 2. $e_{\text{ctrl}} = 0.05\%$ has been selected for the computations. The final time instance of the analysis is $T_{\max} = 3 \cdot 10^{-3}$ s.

TABLE I
RELATIVE ERROR VALUES FOR EACH STATE VARIABLE (VALUES ROUNDED TO THE NEAREST THOUSANDTH

variable	average error, %
i_α	0.076
u_β	0.053
u_κ	0.062
i_γ	0.049

By looking at the figure itself – a difference between the results cannot be noticed. A relative difference (for each state variable x) between the numerical and semi-analytical results is computed for each node selected by the numerical solver:

$$e_j = 100 \cdot \frac{|x(t_j) - x_j|}{\max\limits_{j=1,2,\ldots n_t} |x(t_j)|} \ \%, \tag{10}$$

where j is the index of the node, t_j is the corresponding time instance, $x(t_j)$ is the result obtained through gcdAlpha and x_j is the result obtained for t_j through the numerical solver. The average of the computed error values, for each state variable, is given in Table I.

VII. CONCLUSION

A study concerning the application of SubIval (the subinterval-based numerical method) in time-stepping solvers has been discussed.

The solvers are designed to solve initial value problems with fractional derivatives. The basics of the method have been recalled from previous papers [22], [26].

The general form of the problems that the solvers are aimed at dealing with have been given (6), along with the ones the solvers currently support (9).

The possibility of applying time step size adaptivity has also been recalled [28].

A test example for the SubIval solver has been given in the form of a transient circuit with fractional coils and fractional capacitors.

The test example has also been solved by using the gcdAlpha method [32] (a semi-analytical method) to obtain a very accurate referential solution by means of a method operating on a completely different basis.

The numerical results show a good resemblance to the referential solution. Additionally, an error value has been computed – this allowed to verify the numerical solution.

APPENDIX

The circuit equations are first expressed in the following form [26]:

$$\begin{cases} \boldsymbol{M}_{\text{I}}\boldsymbol{y}(t) + \boldsymbol{M}_{\text{II}}\boldsymbol{x}(t) = \boldsymbol{T}\boldsymbol{v}(t), \\ \mathrm{d}^{\boldsymbol{\alpha}}\boldsymbol{x}(t) + \boldsymbol{M}_{\text{III}}\boldsymbol{y}(t) + \boldsymbol{M}_{\text{IV}}\boldsymbol{x}(t) = \boldsymbol{0}_N, \end{cases} \tag{11}$$

where $\boldsymbol{0}_N$ denotes a vector of N zeros, while $\mathrm{d}^{\boldsymbol{\alpha}}\boldsymbol{x}(t)$ is the vector of fractional derivatives of the state variables (as in (9)), $\boldsymbol{y}(t)$ contains any remaining useful variables (the vector length is denoted by N_y), $\boldsymbol{T}$ is an $N \times M$ matrix, $\boldsymbol{M}_{\text{I}}$ is an $N_y \times N_y$ matrix, $\boldsymbol{M}_{\text{II}}$ is an $N_y \times N$ matrix, $\boldsymbol{M}_{\text{III}}$ is an $N \times N_y$ matrix, and $\boldsymbol{M}_{\text{IV}}$ is an $N \times N$ matrix. There are many possibilities

on how to treat the circuit (e.g. a general formulation of node potential equations as in [26]); in this study – an approach has been chosen where initially not many equations need to be formulated. The state variable vector consists of the coil currents and the capacitor voltages:

$$\boldsymbol{x}(t) = [\ i_\alpha(t) \quad u_\beta(t) \quad u_\kappa(t) \quad i_\gamma(t)\ ]^{\mathrm{T}}. \tag{12}$$

The vector of fractional derivative orders is given by:

$$\boldsymbol{\alpha} = [\ \alpha \quad \beta \quad \kappa \quad \gamma\ ]^{\mathrm{T}}. \tag{13}$$

The auxiliary variable vector, on the other hand, consists of the coil voltages and capacitor currents:

$$\boldsymbol{y}(t) = [\ u_\alpha(t) \quad i_\beta(t) \quad i_\kappa(t) \quad u_\gamma(t)\ ]^{\mathrm{T}}. \tag{14}$$

The $\boldsymbol{v}(t)$ vector contains only the time function for the voltage source:

$$\boldsymbol{v}(t) = [\ E(t)\ ]. \tag{15}$$

With the above in mind: $N = 4$, $N_y = 4$ and $M = 1$. In the discussion below the time dependency notation of the variables is omitted (e.g. $i_\alpha(t)$ is now simply written as i_α).

From Kirchhoff's first law one can derive the equation:

$$i_\alpha - i_\beta - i_\kappa - i_\gamma = 0, \tag{16}$$

which gives the following entries for the $\boldsymbol{M}_{\mathrm{I}}$ and $\boldsymbol{M}_{\mathrm{II}}$ matrices:

$$\begin{aligned} \boldsymbol{M}_{\mathrm{I}\ 1,2} &= -1, \\ \boldsymbol{M}_{\mathrm{I}\ 1,4} &= -1, \\ \boldsymbol{M}_{\mathrm{II}\ 1,1} &= 1, \\ \boldsymbol{M}_{\mathrm{II}\ 1,4} &= 1. \end{aligned} \tag{17}$$

From Kirchhoff's second law the following equation can be obtained:

$$i_\alpha R_1 + u_\alpha + i_\beta R_2 + u_\beta = E, \tag{18}$$

along with:

$$u_\kappa + i_\kappa R_3 - i_\beta R_2 - u_\beta = 0, \tag{19}$$

and:

$$u_\gamma + i_\gamma R_4 - i_\beta R_2 - u_\beta = 0. \tag{20}$$

From (18) one can obtain the entries:

$$\begin{aligned} \boldsymbol{M}_{\mathrm{II}\ 2,1} &= R_1, \\ \boldsymbol{M}_{\mathrm{I}\ 2,1} &= 1, \\ \boldsymbol{M}_{\mathrm{I}\ 2,2} &= R_2, \\ \boldsymbol{M}_{\mathrm{II}\ 2,2} &= 1, \\ \boldsymbol{T}_{2,2} &= 1. \end{aligned} \tag{21}$$

From (19) one obtains:

$$\begin{aligned} \boldsymbol{M}_{\mathrm{II}\ 3,3} &= 1, \\ \boldsymbol{M}_{\mathrm{I}\ 3,3} &= R_3, \\ \boldsymbol{M}_{\mathrm{I}\ 3,2} &= -R_2, \\ \boldsymbol{M}_{\mathrm{II}\ 3,2} &= -1, \end{aligned} \tag{22}$$

while from (20):

$$\begin{aligned} \boldsymbol{M}_{\mathrm{II}\ 4,4} &= 1, \\ \boldsymbol{M}_{\mathrm{I}\ 4,4} &= R_4, \\ \boldsymbol{M}_{\mathrm{I}\ 4,2} &= -R_2, \\ \boldsymbol{M}_{\mathrm{II}\ 4,2} &= -1. \end{aligned} \tag{23}$$

In the case of this study $\boldsymbol{M}_{\mathrm{IV}}$ is a zero matrix; $\boldsymbol{M}_{\mathrm{III}}$ is filled with entries that result from the differential equations of the coils and capacitors. For the L_α coil the differential equation is:

$$L_\alpha\ {}_{t_0}\mathrm{D}_t^\alpha i_\alpha = u_\alpha, \tag{24}$$

for the C_β capacitor:

$$C_\beta\ {}_{t_0}\mathrm{D}_t^\beta u_\beta = i_\beta, \tag{25}$$

while for the C_κ capacitor:

$$C_\kappa\ {}_{t_0}\mathrm{D}_t^\kappa u_\kappa = i_\kappa. \tag{26}$$

Finally, for the L_γ coil:

$$L_\gamma\ {}_{t_0}\mathrm{D}_t^\gamma i_\gamma = u_\gamma. \tag{27}$$

The differential equations yield the following entries for $\boldsymbol{M}_{\mathrm{III}}$:

$$\begin{aligned} \boldsymbol{M}_{\mathrm{III}\ 1,1} &= -\tfrac{1}{L_\alpha}, \\ \boldsymbol{M}_{\mathrm{III}\ 2,2} &= -\tfrac{1}{C_\beta}, \\ \boldsymbol{M}_{\mathrm{III}\ 3,3} &= -\tfrac{1}{C_\kappa}, \\ \boldsymbol{M}_{\mathrm{III}\ 4,4} &= -\tfrac{1}{L_\gamma}. \end{aligned} \tag{28}$$

After the vectors are set up and the matrices are filled – to obtain the matrices of (9) one can apply the following formulae:

$$\boldsymbol{A} = \boldsymbol{M}_{\mathrm{III}}(\boldsymbol{M}_{\mathrm{I}}^{-1}\boldsymbol{M}_{\mathrm{II}}), \tag{29}$$

and:

$$\boldsymbol{B} = -\boldsymbol{M}_{\mathrm{III}}(\boldsymbol{M}_{\mathrm{I}}^{-1}\boldsymbol{T}). \tag{30}$$

References

[1] K.B. Oldham, J. Spanier, *The Fractional Calculus*, Academic Press, New York (1974).
[2] P.W. Ostalczyk, P. Duch, D.W. Brzeziński, D. Sankowski, "Order Functions Selection in the Variable-, Fractional-Order PID Controller", *Advances in Modelling and Control of Non-integer-Order Systems*, Springer, 159–170 (2015).
[3] D. Spałek, "Synchronous Generator Model with Fractional Order Voltage Regulator $\mathrm{PI}^b\mathrm{D}^a$", *Acta Energetica* 2/23, 78–84 (2015).
[4] L. Mescia, P. Bia, D. Caratelli, "Fractional Derivative Based FDTD Modeling of Transient Wave Propagation in Havriliak-Negami media", *IEEE Transactions on Microwave Theory and Techniques* 62 (9), 1920–1929 (2014).
[5] R. Garrappa, G. Maione, "Fractional Prabhakar Derivative and Applications in Anomalous Dielectrics: A Numerical Approach", *Theory and Applications of Non-Integer Order Systems*, Springer, 429–439 (2017).
[6] I. Schäfer, K. Krüger, "Modelling of lossy coils using fractional derivatives", *Phys. D: Appl. Phys.* 41, 1–8 (2008).
[7] M. Sowa, "DAQ-based measurements for ferromagnetic coil modeling using fractional derivatives", *International Interdisciplinary PhD Workshop IIPhDW 2018* (in print).
[8] A. Jakubowska, J. Walczak, "Analysis of the transient state in a circuit with supercapacitor", *Poznan University of Technology Academic Journals. Electrical Engineering* 81, 27–34 (2015).
[9] W. Mitkowski, P. Skruch, "Fractional-order models of the supercapacitors in the form of RC ladder networks", *Bull. Pol. Ac.: Tech.* 61 (3), 580–587 (2013).
[10] M.A. Ezzat, A.S. El-Karamany, A.A. El-Bary, "Thermo-viscoelastic materials with fractional relaxation operators", *Applied Mathematical Modelling* 39, 23–24, 7499–7512 (2015).
[11] M. Faraji Oskouie, R. Ansari, "Linear and nonlinear vibrations of fractional viscoelastic Timoshenko nanobeams considering surface energy effects", *Applied Mathematical Modelling* 43, 337–350 (2017).

[12] A. Kawala-Janik, M. Podpora, A. Gardecki, W. Czuczwara, J. Baranowski, W. Bauer: "Game controller based on biomedical signals", *Methods and Models in Automation and Robotics (MMAR) 2015 20th International Conference*, 934–939 (2015).
[13] W. Bauer, A. Kawala-Janik, "Implementation of Bi-fractional Filtering on the Arduino Uno Hardware Platform", *Theory and Applications of Non-Integer Order Systems*, Springer, 419–428 (2017).
[14] U.N. Katugampola, "Mellin transforms of generalized fractional integrals and derivatives", *Applied Mathematics and Computation*, 257, 566–580 (2015).
[15] M. Caputo, "Linear models of dissipation whose Q is almost frequency independent – II", *Geophysical Journal International* 13 (5), 529–539 (1967).
[16] T. Kaczorek, *Fractional linear systems and electrical circuits*, Springer (2015).
[17] S. Momani, M.A. Noor, "Numerical methods for fourth order fractional integro-differential equations", *Appl. Math. Comput.* 182, 754–760 (2006).
[18] N.S. Khodabakhshi, S.M. Vaezpour, D. Baleanu, "Numerical solutions of the initial value problem for fractional differential equations by modification of the Adomian decomposition method", *Fractional Calculus and Applied Analysis* 17 (2), 382–400 (2014).
[19] C. Lubich, "Fractional linear multistep methods for Abel-Volterra integral equations of the second kind", *Math. Comput.* 45, 463–469 (1985).
[20] W.-H. Luo, T.-Z. Huang, G.-C. Wu, X.-M. Gu, "Quadratic spline collocation method for the time fractional subdiffusion equation", *Applied Mathematics and Computation* 276, 252–265 (2016).
[21] R. Garrappa, "On accurate product integration rules for linear fractional differential equations", *Journal of Computational and Applied Mathematics* 235, 1085–1097 (2011).
[22] M. Sowa, "A subinterval-based method for circuits with fractional order elements", *Bull. Pol. Acad. Sci.: Tech.* 62 (3), 449–454 (2014).
[23] M. Sowa, "Solutions of Circuits with Fractional, Nonlinear Elements by Means of a SubIval Solver", In *Non-Integer Order Calculus and its Applications. RRNR 2017. Lecture Notes in Electrical Engineering*, vol. 496, Springer (2018).
[24] M. Sowa, "The subinterval-based method and its potential improvements", *XXXIX International Conference IC-SPETO 2016*, Gliwice-Ustron, 18-21.05.2016 (2016).
[25] M. Sowa, "SubIval computation time assessment", *Proceedings of International Interdisciplinary PhD Workshop 2017. IIPhDW 2017*, September 9-11, 2017, Lodz (2017).
[26] M. Sowa, "Application of SubIval in solving initial value problems with fractional derivatives", *Applied Mathematics and Computation* 319, 86–103 (2018).
[27] M. Sowa, "Application of SubIval, a method for fractional-order derivative computations in IVPs", In *Theory and applications of non-integer order systems. Lecture Notes in Electrical Engineering*, vol. 407, Springer (2017).
[28] M. Sowa, "A local truncation error estimation for a SubIval solver", *Bull. Pol. Acad. Sci.: Tech.* (in print) (2018).
[29] http://msowascience.com.
[30] http://www.mathworks.com/products/matlab.html.
[31] http://www.gnu.org/software/octave/.
[32] M. Sowa, "'gcdAlpha" – a semi-analytical method for solving fractional state equations", *Computer Applications in Electrical Engineering* 96, 231–242 (2018).

Marcin Sowa received his MSc degree in 2008 at the Silesian University of Technology in Gliwice, Poland. He received his PhD in the same university in 2013 and currently works there as an Assistant Professor at the Faculty of Electrical Engineering. He specializes in computations of circuits with fractional elements, electromagnetic field analyses, numerical methods and computer programming. His academic achievements include over 40 papers (published mostly in international journals and conference proceedings).