# Error computation strategies in an adaptive step size solver for time fractional problems

Marcin SOWA

**Abstract.** The paper concerns an adaptive time step size numerical solver for fractional order state equations. The core of the solver applies SubIval, i.e. the subinterval-based method for computations of the fractional derivative in Caputo and Riemann-Liouville definitions. In the solver the step size is modified basing on a formula deduced from an approximation of the truncation error. The error has been computed for either the state variables themselves or their respective fractional derivatives. This resulted in three different strategies for the adaptive step size solver (two of which have been proposed for the case when the error is estimated for the derivatives). Two circuit problems have served as examples for the numerical experiments.

**Keywords:** numerical method, SubIval, fractional derivative, error computation.

**2010 Mathematics Subject Classification:** 26A33, 34A08.

## 1. Introduction

Fractional calculus is a very popular topic of late, where many researchers are interested in its applications in modeling and, hence, the design of methods allowing to perform computations wherever the fractional derivative and fractional integral (or, generally, the fractional integroderivative) appear.

Fractional integroderivatives have been considered as operations with respect to time and space. This paper only concerns the first case for a derivative of order $\alpha \in (0, 1]$.

M. Sowa
Faculty of Electrical Engineering, Silesian University of Technology, Krzywoustego 2, 44-100 Gliwice, Poland, e-mail: marcin.sowa@polsl.pl

R. Wituła, B. Bajorska-Harapińska, E. Hetmaniok, D. Słota, T. Trawiński (eds.), *Selected Problems on Experimental Mathematics*. Wydawnictwo Politechniki Śląskiej, Gliwice 2017, pp. 89–102.

The most popular definitions of the fractional derivative are that of Caputo [6]:

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

and the Riemann-Liouville definition [25]:

$$ {}^{\mathrm{RL}}_{t_{\mathrm{a}}}\mathrm{D}^{\alpha}_{t_{\mathrm{b}}}x(t) = \frac{1}{\Gamma(1-\alpha)}\frac{\mathrm{d}}{\mathrm{d}t}\int_{t_{\mathrm{a}}}^{t_{\mathrm{b}}}\frac{x(\tau)}{(t-\tau)^{\alpha}}\mathrm{d}\tau, \tag{2}$$

where the author uses a special notation, where instead of the bounds, the integrodifferentiation interval $\mathit{\Xi} = [t_{\mathrm{a}}, t_{\mathrm{b}}]$ is used:

$$ \mathrm{d}^{\alpha}_{\mathit{\Xi}}x(t) = {}_{t_{\mathrm{a}}}\mathrm{D}^{\alpha}_{t_{\mathrm{b}}}x(t), \tag{3}$$

which allows to skip the left-hand indices if an equation holds true for both fractional derivative definitions.

Applications of the fractional derivative can be found i.a. in:

- modeling of human body bioimpedance [14],
- control theory and its applications, where fractional controllers are used [3, 34],
- diffusion kinetics in porous media [40]
- the design of fractional order filters [22, 29, 4],
- electromagnetic field analysis: when modeling dielectrics [7, 15] and when modeling soft ferromagnetics [11],
- temperature field analyses [5, 31],
- modeling viscoelastic materials [26],
- robot path control [2],
- what the author is interested in mostly – circuit theory, where the fractional capacitor, described by the equation:

  $$ C_{\alpha}\mathrm{d}^{\alpha}_{\mathit{\Xi}}u(t) = i(t), \tag{4}$$

  is useful when modeling supercapacitors [20, 13], while the fractional coil model, described by the equation:

  $$ L_{\alpha}\mathrm{d}^{\alpha}_{\mathit{\Xi}}i(t) = u(t), \tag{5}$$

  has found its uses in modeling of coils with ferromagnetic cores [30].

Solutions of problems with the fractional derivative can be obtained via analytical means (where the solutions base on the Mittag-Leffler function [21]) or with a properly selected numerical method. Some of the most commonly mentioned are:

- the simplest – methods that base on the fractional difference operator [27, 23],
- the Taylor expansion method [16],
- the Adomian decomposition method [9],
- collocation methods [28, 39],
- the differential transform method [37],
- the variational iteration method [38],
- fractional linear multistep methods [24].

The method used for the computations of the paper is the subinterval-based method, which is now referred to by its acronym – SubIval. It is a numerical method that can be used for the approximation of the time derivative in initial value problems. For a given time step it conveniently allows to obtain a system of equations from a system of fractional differential equations.

## 2. SubIval basis and resulting formula

This section gives a brief description on the basis of SubIval and the functionality of its implementation in the DLL available at [18], which has been described in [33]. Being called the subinterval-based method comes from its fundamental partition of the integrodifferentiation on the entire considered time interval $\mathit{\Xi}_{\mathrm{tot}} = [t_0, t_{\mathrm{now}}]$ into subsequent subinterval integrodifferentiations:

$$\mathrm{d}^{\alpha}_{\mathit{\Xi}_{\mathrm{tot}}} x(t) \approx \sum_{s=1}^{M} \mathrm{d}^{\alpha}_{\mathit{\Xi}_s} \widetilde{x}_s(t), \tag{6}$$

where in each subinterval $\mathit{\Xi}_s$, with a unique index $s$, an interpolation is performed in $\mathit{\Theta}_s \supseteq \mathit{\Xi}_s$. How the subintervals are established in each time step is determined by an original subinterval dynamics algorithm (explained in detail in [33]). After integrodifferentiations of obtained monomials had been performed – SubIval results in an implicit formula for each time step, where $t = t_{\mathrm{now}}$:

$$\mathrm{d}^{\alpha}_{\mathit{\Xi}_{\mathrm{tot}}} x(t) \approx a x(t_{\mathrm{now}}) + b, \tag{7}$$

which is the same as when backward differentiation formulae are applied for first order derivative computations. The generality of the formula allows it to be used in any equation where the fractional derivative appears. It can be even applied in nonlinear problems. Also, both the Caputo and Riemann-Liouville derivatives can be assumed. The initial computations actually take into account the Caputo derivative, however, the Riemann-Liouville fractional derivative can be computed easily from it as for $\alpha \in (0, 1]$ it only differs by one term dependent on the initial condition [1]:

$${}^{\mathrm{RL}}_{t_0}\mathrm{D}^{\alpha}_{t_{\mathrm{now}}} x(t) = {}^{\mathrm{C}}_{t_0}\mathrm{D}^{\alpha}_{t_{\mathrm{now}}} x(t) + \frac{x(t_0)}{\Gamma(1-\alpha)} (t_{\mathrm{now}} - t_0)^{-\alpha}. \tag{8}$$

## 3. SubIval in an adaptive step size scheme

A two stage methodology has been proposed [33], which follows what is commonly known as the predictor-corrector scheme. Computations of a problem are performed in two stages, which can be different in a lot of ways – in single options or even in the methods being used. E.g. for integer order derivatives one can use an explicit formula of order $n$ in the predictor stage, while an implicit formula of the same order is used in the corrector stage [12]. Another example is the use of two methods varying in their order [8].

The methodology proposed for SubIval (first presented in [32]) is one where in the last integrodifferentiation (i.e. in $\varXi_M$ on $\widetilde{x}_M$, which bases on the interpolation in $\varTheta_M$) for the first stage (called the *predictor*) the interpolation is of order $q_M - 1$, while the second stage (called the *corrector*) uses an interpolation of order $q_M$.

Before further considerations, it is worthwhile to point out that already for every variable $x$ under a fractional derivative one obtains two sets of $a$ and $b$ parameters of equation (7), where the ones for the predictor stage, in order to be distinguished, are being called $a_\mathrm{p}$ and $b_\mathrm{p}$ further on.

## 4. Time step adjustment

In a predictor-corrector scheme it is important to implement a reliable methodology for the adjustment of the step size. This is done by, first, an error estimation. Secondly, a formula for the adjustment is used, which usually bases on information concerning the truncation error.

In a separate analysis for the truncation error $e_\mathrm{trunc}$ (which will be discussed in a future paper), for the proposed methodology it has been found that the predictor error:

$$e_\mathrm{p\ trunc} \approx c\Delta t^{q_M - \alpha}. \tag{9}$$

Assuming that the corrector error is much smaller – the predictor error is used for error estimation. The coefficient, by which the time step is multiplied is, therefore, computed with the formula:

$$\eta = \sqrt[(q_M - \alpha)]{\frac{e_\mathrm{ctrl}}{e}}, \tag{10}$$

where $e$ is the error value computed for a considered variable $x$, while $e_\mathrm{ctrl}$ is called the *control error* (related to the desired accuracy).

For each variable an error $e$ and a proposed adjustment coefficient $\eta$ is computed. If any of the errors exceeds a previously given $e_\mathrm{max}$ value then the time step is repeated. In any case either the next or current step size is adjusted by taking the smallest $\eta$ value from those computed for all variables.

## 5. Error computation strategies

The error $e$ can be computed in a number of different ways. It is, after all, only an estimation of the true error between the SubIval computations and an exact solution.

The obvious choice is the computation of the solution for both sets of parameters (predictor, resulting in $x_\mathrm{p}$, and corrector, resulting in $x$), where the relative difference between these values, with respect to a selected $x_\mathrm{ref}$ value (being e.g. the maximum of previously computed absolute values):

$$e = \frac{|x - x_\mathrm{p}|}{x_\mathrm{ref}} \cdot 100\%. \tag{11}$$

The truncation error, however, has actually been obtained by judging the difference between the predictor and corrector fractional derivative computations (while it is also common for the truncation error to be actually obtained for the variable itself in such analyses [10]). Hence, the formula above assumes that errors for the variable can also be approximated through (9). An advantage is that it directly studies the end result, which is, in most cases, more or equally as important as the derivative.

If one wishes to estimate the error through the computed derivative values:

$$\mathrm{d}^{\alpha}_{\Xi_{\mathrm{tot}}} x(t) \approx d_{\mathrm{p}} = a_{\mathrm{p}} x_{\mathrm{p}} + b_{\mathrm{p}} \tag{12}$$

and

$$\mathrm{d}^{\alpha}_{\Xi_{\mathrm{tot}}} x(t) \approx d = ax + b \tag{13}$$

then the following formula is used:

$$e = \frac{|d - d_{\mathrm{p}}|}{d_{\mathrm{ref}}} \cdot 100\%. \tag{14}$$

$d_{\mathrm{ref}}$ is handled in the same way for the derivatives as $x_{\mathrm{ref}}$ for the variables themselves.

Both so far introduced error computation strategies assume implicit schemes in the predictor and corrector, hence, for both a solution must be computed after the derivative had been obtained. The derivative-based error computation approach, however, can be simplified. First – the coefficient sets $a_{\mathrm{p}}$, $b_{\mathrm{p}}$ and $a$, $b$ are obtained but only a solution is obtained for the second stage. The coefficients from the first stage are used only to obtain the derivative:

$$\mathrm{d}^{\alpha}_{\Xi_{\mathrm{tot}}} x(t) \approx d_{\mathrm{p}} = a_{\mathrm{p}} x + b_{\mathrm{p}}. \tag{15}$$

Notice that now the solution $x$ is used in both $d$ and $d_{\mathrm{p}}$ computations.

## 6. Computational examples

Three error computation strategies have been presented in the previous section:

- the variable based approach (further on called the $x$-based approach), where equation (11) is used,
- the derivative based approach (further on called the $d$-based approach), where equation (14) is applied; $d_{\mathrm{p}}$ is computed with formula (12), while $d$ is obtained through (13),
- the single solution derivative based approach (further on called the simplified $d$-based approach), where $d_{\mathrm{p}}$ is computed through (15), while $d$ is computed with (13) as in the standard $d$-based approach.

Two computational examples, with referential (analytical or steady state) solutions, have been brought forward to determine what are the actual error values in comparison to the values estimated in the time stepping process.

Each of these problems can be brought, firstly, to the system of equations [33]:

$$\begin{cases} \boldsymbol{M}_{\mathrm{I}}\boldsymbol{y}(t) + \boldsymbol{M}_{\mathrm{II}}\boldsymbol{x}(t) = \boldsymbol{T}\boldsymbol{v}(t), \\ \mathbf{d}_{\Xi}^{\boldsymbol{\alpha}}\boldsymbol{x}(t) + \boldsymbol{M}_{\mathrm{III}}\boldsymbol{y}(t) = 0, \end{cases} \tag{16}$$

where the vector of fractional derivatives (in either Riemann-Liouville or Caputo definitions):

$$\mathbf{d}_{\Xi}^{\boldsymbol{\alpha}}\boldsymbol{x}(t) = [\,\mathrm{d}_{\Xi}^{\alpha_1}x_1(t)\ \mathrm{d}_{\Xi}^{\alpha_2}x_2(t)\ \ldots\ \mathrm{d}_{\Xi}^{\alpha_{nx}}x_{n_x}(t)\,]^{\mathrm{T}}. \tag{17}$$

$\boldsymbol{x}(t)$ is the vector of $n_x$ state variables, $\boldsymbol{\alpha}$ is a vector of the fractional derivative orders, $\boldsymbol{y}(t)$ is a vector of $n_y$ variables that are not placed in the state vector $\boldsymbol{x}(t)$, $\boldsymbol{v}(t)$ is a vector of source time functions, $\boldsymbol{M}_{\mathrm{I}}$ is an $n_y \times n_y$ matrix, $\boldsymbol{M}_{\mathrm{II}}$ is an $n_y \times n_x$ matrix, $\boldsymbol{M}_{\mathrm{III}}$ is an $n_x \times n_y$ matrix and $\boldsymbol{T}$ is an $n_y \times n_v$ matrix. Secondly, the following system of fractional state equations can be obtained:

$$\mathbf{d}_{\Xi}^{\boldsymbol{\alpha}}\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{v}(t). \tag{18}$$

$\boldsymbol{A}$ is an $n_x \times n_x$ matrix, $\boldsymbol{B}$ is an $n_x \times n_v$ matrix. One can obtain (18) from (16) by applying the formulae:

$$\boldsymbol{A} = \boldsymbol{M}_{\mathrm{III}}(\boldsymbol{M}_{\mathrm{I}}^{-1}\boldsymbol{M}_{\mathrm{II}}) \tag{19}$$

and:

$$\boldsymbol{B} = -\boldsymbol{M}_{\mathrm{III}}(\boldsymbol{M}_{\mathrm{I}}^{-1}\boldsymbol{T}). \tag{20}$$

## *6.1. Series RLC circuit*

The first example concerns an *RLC* series circuit with a unit step voltage source (figure 1). Both the coil and capacitor are elements of order $\alpha$.

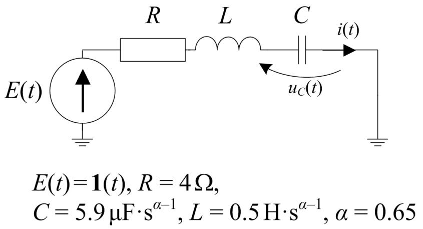

Fig. 1. Exemplary circuit problem: fractional *RLC* series circuit

As it is typical in such problems – the capacitor voltage and coil current are taken into account as the state variables. The problem has an analytical solution available in [36] assuming that the capacitor and coil are of the same order $\alpha$.

The $x$-based approach was the first one being tested. The estimated error has been compared with the actual error being the relative difference between the numerical and analytical solution:

$$e_{\mathrm{an}} = \frac{|x - x_{\mathrm{an}}|}{\max|x_{\mathrm{an}}|}. \tag{21}$$

The results are depicted in figure 2.

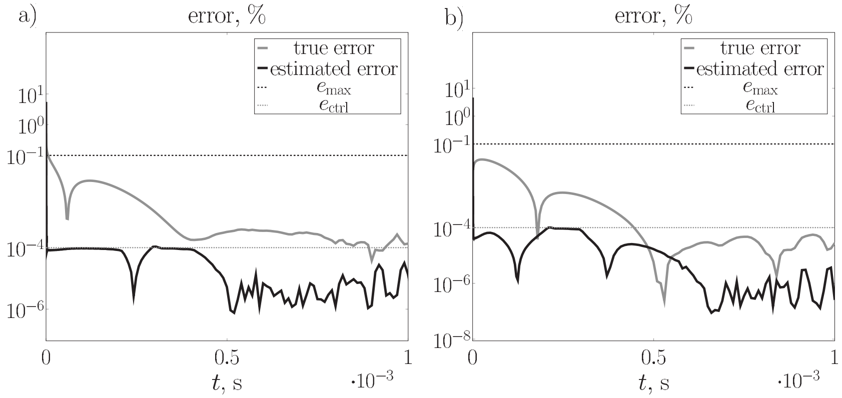

Fig. 2. Comparison of estimated and actual errors (for the *RLC* circuit, using the $x$-based approach) of: a) $i_L$, b) $u_C$

The step size adjustment is performed correctly as it keeps the maximum error close to $e_{\text{ctrl}}$. The estimated error and true error, however, are very different at the beginning. Obviously, the difference between the predictor and corrector results is an estimation of only one term as given in (9). The remaining terms cannot be estimated using this methodology, hence there can always be a difference in the numerical solution and an exact solution.

The values of the estimated error exceeding the $e_{\max}$ value, at the initial stages of the computations, are inevitable since initially the error estimation is computed as relative to $x_{\text{ref}}$, which is either an arbitrarily selected value $x_{\text{low}}$ or the maximum of absolute values so far obtained for the variable (whichever is bigger). Hence, for a small $x_{\text{low}}$ the time step is reduced to its minimum value of $\Delta t_{\min}$ and only increased when a couple of new nodes are added.

Next, the $d$-based approach is applied. The estimated error is compared with the actual relative difference between the derivative of the numerical and analytical solution:

$$e_{d\text{ an}} = \frac{|d - d_{\text{an}}|}{\max|d_{\text{an}}|}. \tag{22}$$

The results are presented in figure 3.

The comparison shows that for the selected example either strategy can be applied and are similarly accurate for the selected problem.

Finally, the simplified $d$-based approach is applied. The estimated error is compared with $e_{d\text{ an}}$ as in the $d$-based approach. The results are depicted in figure 4.

It comes as no surprise that the error estimation has not improved as the concept of the simplified $d$-based approach assumes an even lesser accuracy of the error estimation. What is surprising, however, is that the estimation presents results similar to those of the standard $d$-based approach. This is promising as the simplified approach only requires one solution, which (especially for problems with many equations) could reduce the computation time by around 50 %.

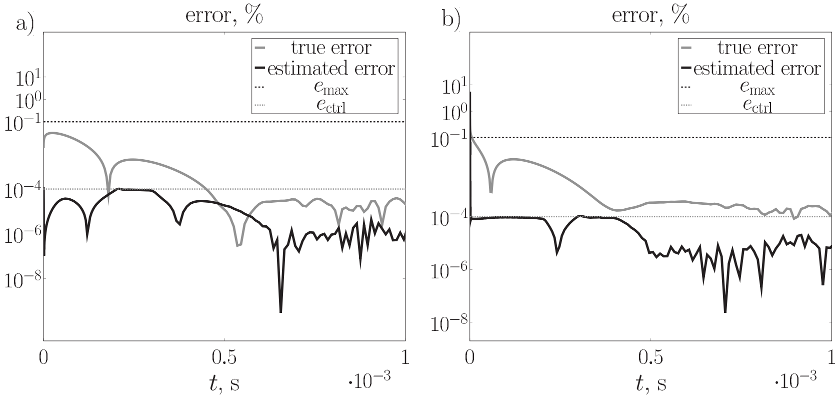

Fig. 3. Comparison of estimated and actual errors (for the *RLC* circuit, using the *d*-based approach) of: a) $\frac{\mathrm{d}i_L}{\mathrm{d}t}$, b) $\frac{\mathrm{d}u_C}{\mathrm{d}t}$

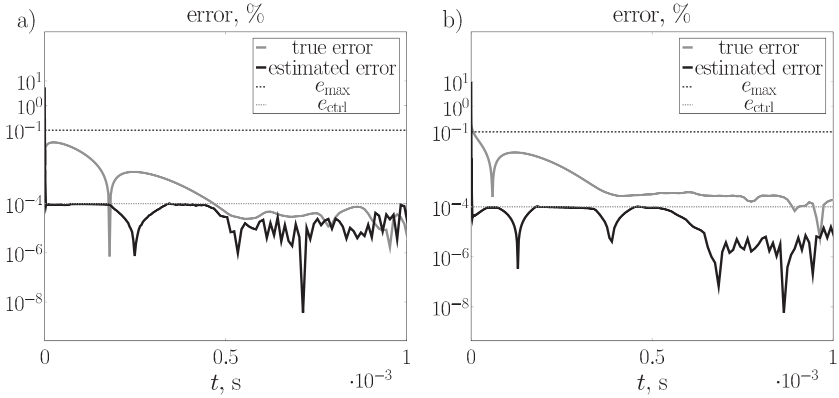

Fig. 4. Comparison of estimated and actual errors (for the *RLC* circuit, using the simplified *d*-based approach) of: a) $\frac{\mathrm{d}i_L}{\mathrm{d}t}$, b) $\frac{\mathrm{d}u_C}{\mathrm{d}t}$

## *6.2. AC circuit with fractional elements*

The second example concerns the steady state computation of a linear circuit with sinusoidal sources (figure 5). The steady state solution can be obtained through an analysis applying complex numbers like in [35].

The computed steady state solution and that of the adaptive step size solver (applying SubIval) have been compared. The capacitor voltages and coil current time functions constitute the state vector $\boldsymbol{x}$. Actually, $n_T = 10$ periods ($T = \frac{1}{f}$) have been computed, which assured a steady state of the time dependent solution in $t \in [(n_T - 1)T, n_T T]$. An error analysis, like the one in the previous subsection, has been performed for this interval. The results for the $x$-based approach are given in figure 6.

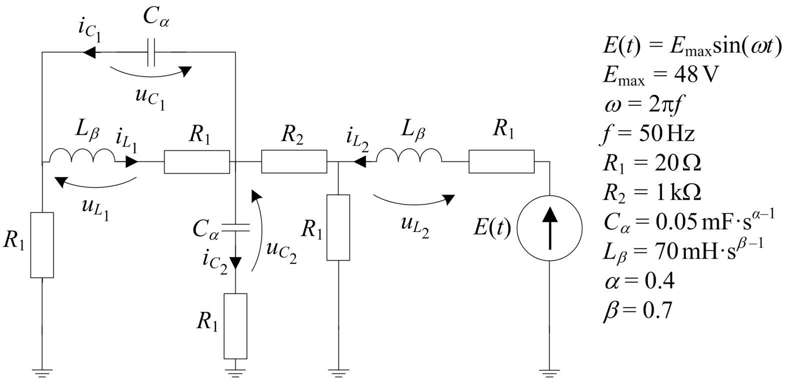

Fig. 5. Exemplary circuit problem: an AC circuit with fractional capacitors and coils

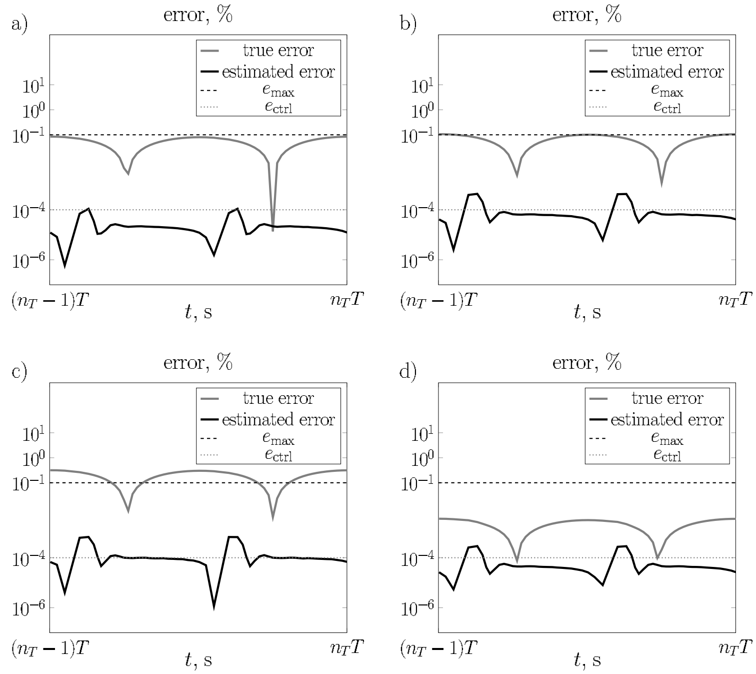

Fig. 6. Comparison of estimated and actual errors (for the AC circuit, using the $x$-based approach) of: a) $u_{C_1}$, b) $i_{L_1}$, c) $u_{C_2}$, d) $i_{L_2}$

There is a significant difference between the computed errors. The time adaptive strategy, however, is successful at keeping the maximum estimated error at values close to $e_{\text{ctrl}}$.

Next – the $d$-based approach has been applied. The errors obtained for the final period have been presented in figure 7.

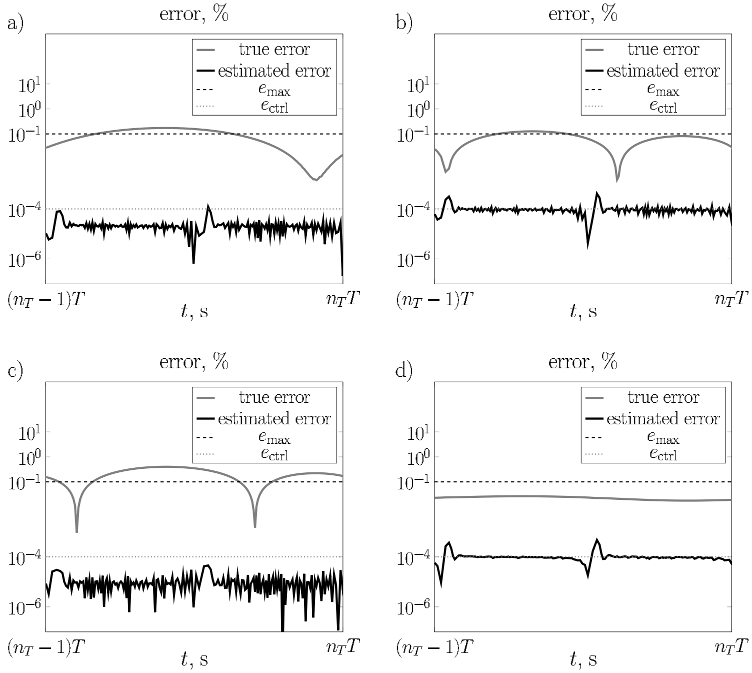

Fig. 7. Comparison of estimated and actual errors (for the AC circuit, using the $d$-based approach) of: a) $u_{C_1}$, b) $i_{L_1}$, c) $u_{C_2}$, d) $i_{L_2}$

Again the part of the error that could not be estimated has been significant as the difference between the estimated truncation error and the actual difference (between the numerical and analytical solutions) is apparent.

What is interesting is that this time there has been a need for more time steps to sustain the maximum of estimated errors at values close to $e_{\text{ctrl}}$. This reduction of the step size did, however, not improve the actual solution accuracy (it has not been shown in the figure but the actual error for the state variables has also not improved greatly). This inaccuracy will be further investigated in future studies.

The simplified $d$-based approach has been tested next. The error computation results for this strategy are presented in figure 8.

Again the simplified $d$-based approach has proven to be efficient enough as it resembles similar results as in the standard $d$-based approach. However, to keep the error of the computed derivatives at $e_{\text{ctrl}}$ the number of time steps had to be significantly bigger than for the $x$-based approach (and did not greatly improve the accuracy of the solution).

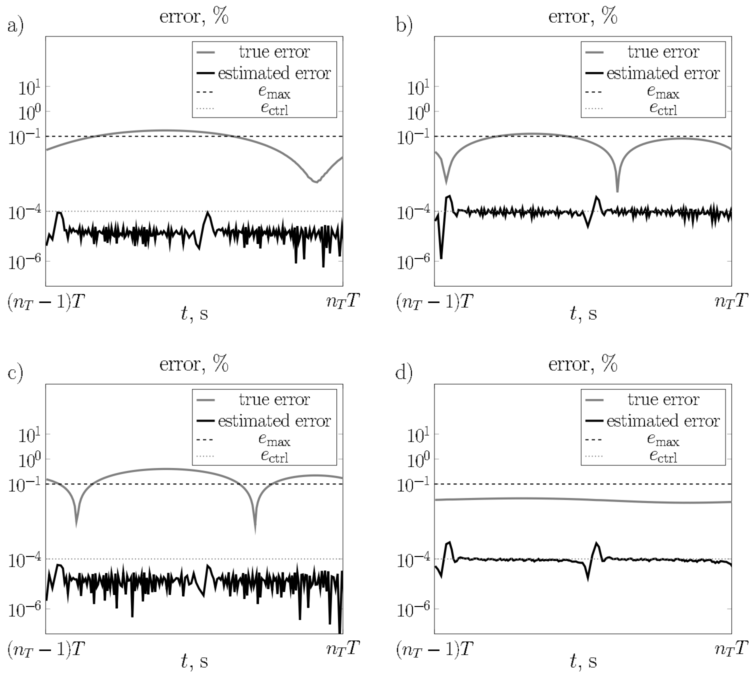

Fig. 8. Comparison of estimated and actual errors (for the AC circuit, using the simplified $d$-based approach) of: a) $u_{C_1}$, b) $i_{L_1}$, c) $u_{C_2}$, d) $i_{L_2}$

## 7. Summary and conclusions

Three error estimation options have been studied for a step size adaptive solver applying SubIval. The error has either been estimated taking into account the computed state variable values (this has been called the $x$-based approach), the computed fractional derivative values of the state variables (the $d$-based approach) and an approach taking into account two fractional derivative values, yet both obtained from the same (single) solution (the simplified $d$-based approach).

For all strategies the step adaptivity has been properly implemented as it managed to successfuly keep the maximum of estimated error values close to a desired value $e_{\mathrm{ctrl}}$.

The error control also has an effect on the actual error – though being different, it also seems to keep the error within range of values representing a reasonable accuracy.

A negative observation is that for the AC problem, after reducing the step size (resulting from the fact that the $d$-based approach encountered a bigger error estimation result) the actual error has not improved greatly. This fact will be investigated more in future studies. Perhaps purely mathematical examples, where the truncation error is actually close to the estimated value, should be studied to determine the source of inaccuracies. One also needs to take into account that the estimation only takes into account a local truncation error. A global truncation error is not taken into account but can definitely have an effect on the numerical solution.

As for the error estimation it would be interesting to apply another scheme, e.g. based on same order methods as is the suggested practice with backward differentiation formulae. Then at least the predictor subinterval strategy would have to be different.

The choice of the error strategy seems to depend mostly on whether the error of the derivative or the variable itself is important, though one must remember that, theoretically, formula (9) has been derived for a truncation error analysis of the derivative. Also, the simplified $d$-based approach has proven to be efficient as it gives similar results to the standard $d$-based approach but requires only one solution to be computed. This greatly reduces the computation time for problems resulting in many equations.

## 8. Remarks on computation tools

The computations have been performed with programs and libraries written by the author in C#. The numerical computations of the fractional derivative have been performed with the SubIval DLL available at [18]. The library uses a part of the code given in [17] for the computation of the gamma function. The step size adaptive solver required a system of equations to be solved at each iteration. The linear systems have been solved by applying methods from the MathNet Numerics library [19].

## Bibliography

1. Abdeljawad T.: *On Riemann and Caputo fractional differences.* Comput. Math. Applic. **62** (2011), 1602–1611.
2. Babiarz A., Łęgowski A., Niezabitowski M: *Robot path control with Al-Alaoui rule for fractional calculus discretization.* Theory and Applications of Non-integer Order Systems, Springer 2017, 405–418.
3. Baranowski J., Bauer W., Zagórowska M., Kawala-Janik A., Dziwiński T., Piątek P.: *Adaptive non-integer controller for water tank system.* Theoretical Developments and Applications of Non-Integer Order Systems, Springer 2016, 271–280.
4. Bauer W., Kawala-Janik A.: *Implementation of bi-fractional filtering on the arduino uno hardware platform.* In: Theoretical Developments and Applications of Non-Integer Order Systems, Springer 2016, 419–428.
5. Brociek R., Słota D., Wituła R.: *Reconstruction of the thermal conductivity coefficient in the time fractional diffusion equation.* In: Advances in Modelling and Control of Non-integer-Order Systems, Springer 2015, 239–247.
6. Caputo M.: *Linear models of dissipation whose Q is almost frequency independent – II.* Geophysical J. Int. **13**, no. 5 (1967), 529–539.
7. Caratelli D., Mescia L., Bia P., Stukach O.V.: *Fractional-calculus-based FDTD algorithm for ultrawideband electromagnetic characterization of arbitrary dispersive dielectric materials.* IEEE Trans. Antennas and Propagation **64**, no. 8 (2016), 3533–3544.
8. Cash J.R.: *The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae.* Comput. Math. Applic. **9**, no. 5 (1983), 645–657.
9. Daftardar-Gejji V., Jafari H.: *Adomian decomposition: a tool for solving a system of fractional differential equations.* J. Math. Anal. Appl. **301**, no. 2 (2005), 508–518.

10. Dolejši V., Kus P.: *Adaptive backward difference formula–discontinuous Galerkin finite element method for the solution of conservation laws.* Internat. J. Numer. Methods Engrg **73**, no. 12 (2008), 1739–1766.
11. Ducharne B. Sebald G., Guyomar D., Litak G.: *Dynamics of magnetic field penetration into soft ferromagnets.* J. Appl. Phys. **117**, no. 24 (2015), 10 pages.
12. Durran D.R., Blossey P.N.: *Implicit-explicit multistep methods for fast-wave-slow-wave problems.* Monthly Weather Review **140**, no. 4 (2012), 1307–1325.
13. Fouda M.E., Elwakil A.S., Radwan A.G., Allagui A.: *Power and energy analysis of fractional-order electrical energy storage devices.* Energy **111** (2016), 785–792.
14. Freeborn T.J., Elwakil A.S., Maundy B.: *Compact wide frequency range fractional-order models of human body impedance against contact currents.* Math. Problems Eng. **2016** (2016), 10 pages.
15. Garrappa R., Maione G.: *Fractional Prabhakar derivative and applications in anomalous dielectrics: a numerical approach.* In: Theory and Applications of Non-integer Order Systems, Springer 2017, 429–439.
16. Huang L., Li X.F., Zhao Y., Xiang-Yang D.: *Approximate solution of fractional integro-differential equations by Taylor expansion method.* Comput. Math. Applic. **62**, no. 3 (2011), 1127–1134.
17. http://www.johndcook.com/Gamma.cs
18. http://www.msowascience.com
19. http://numerics.mathdotnet.com/
20. Jakubowska A., Walczak J.: *Analysis of the transient state in a series circuit of the class $RL_\beta C_\alpha$.* Circuits, Systems, and Signal Processing **35**, no. 6 (2016), 1831–1853.
21. Kaczorek T.: *Cayley–Hamilton theorem for fractional linear systems.* In: Theory and Applications of Non-integer Order Systems, Springer 2017, 45–55.
22. Kawala-Janik A., Podpora M., Gardecki A., Czuczwara W., Baranowski J., Bauer W.: *Game controller based on biomedical signals.* In: Methods and Models in Automation and Robotics (MMAR), 20th International Conference, IEEE 2016, 934–939.
23. Li C., Zeng F.: *Finite difference methods for fractional differential equations.* Int. J. Bifurcation and Chaos **22**, no. 4 (2012), 28 pages.
24. Lubich C.: *Fractional linear multistep methods for Abel-Volterra integral equations of the second kind.* Math. Comput. **45** (1985), 463–469.
25. Munkhammar J.D.: *Riemann-Liouville fractional derivatives and the Taylor-Riemann series.* UUDM project report **7** (2004), 1–18.
26. Nadzharyan T.A., Sorokin V.V., Stepanov G.V., Bogolyubov A.N., Kramarenko E.Yu.: *A fractional calculus approach to modeling rheological behavior of soft.* Polymer **92** (2016), 179–188.
27. Pawłuszewicz E.: *Remarks on Mittag-Leffler discrete function and Putzer algorithm for fractional h-difference linear equations.* In: Theory and Applications of Non-integer Order Systems, Springer 2017, 89–99.
28. Pirkhedri A., Javadi H.H.S.: *Solving the time-fractional diffusion equation via Sinc–Haar collocation method.* Appl. Math. Comput. **257** (2015), 317–326.
29. Psychalinos C., Tsirimokou G., Elwakil A.S.: *Switched-capacitor fractional-step Butterworth filter design.* Circuits, Systems, and Signal Processing **35**, no. 4 (2016), 1377–1393.
30. Schäfer I., Krüger K.: *Modelling of lossy coils using fractional derivatives.* J. Physics D: Applied Physics **41**, no. 4 (2008), 1–8.
31. Sierociuk D., Skovranek T., Macias M., Podlubny I., Petráš I., Dzieliński A., Ziubiński P.: *Diffusion process modeling by using fractional-order models.* Appl. Math. Comput. **257** (2015), 2–11.
32. Sowa M.: *A subinterval-based method for circuits with fractional order elements.* Bull. Pol. Ac.: Tech. **62**, no. 3 (2014), 449–454.
33. Sowa M.: *Application of SubIval in solving initial value problems with fractional derivatives.* Appl. Math. Comput. (in review).
34. Tepljakov A., Gonzalez E.A., Petlenkov E., Belikov J. Monje C.A., Petráš I.: *Incorporation of fractional-order dynamics into an existing PI/PID DC motor control loop.* ISA Transactions **60** (2016), 262–273.
35. Walczak J., Jakubowska A.: *Analysis of parallel resonance $RLC_\alpha$ circuit with supercapacitor.* Computer Applications in Electrical Engineering **12** (2014), 30–37.

36. Włodarczyk M., Zawadzki A.: *Obwody RLC w aspekcie pochodnych niecałkowitych rzędów dodatnich*. Prace Nauk. Pol. i Śl. Elektryka **217**, no. 1 (2011), 75–88.
37. Yang X.-J., Tenreiro Machado J.A., Srivastava H.M.: *A new numerical technique for solving the local fractional diffusion equation: Two-dimensional extended differential transform approach*. Appl. Math. Comput. **274** (2016), 143–151.
38. Yang S., Xiao A., Su H.: *Convergence of the variational iteration method for solving multi-order fractional differential equations*. Comput. Math. Applic. **60**, no. 10 (2010), 2871–2879.
39. Zhao J., and Xiao J., Ford N.J. *Collocation methods for fractional integro-differential equations with weakly singular kernels*. Numerical Algorithms **65** no. 4 (2014), 723–743.
40. Zhokh A.A., Trypolskyi A.A., Strizhak P.E.: *Application of the time-fractional diffusion equation to methyl alcohol mass transfer in silica*. In: Theoretical Developments and Applications of Non-Integer Order Systems, Springer 2016, 501–510.