Marcin SOWA[1]

Silesian University of Technology (1)

doi:10.15199/48.2023.08.38

A study of issues in extreme cases in a numerical solver for nonlinear fractional circuits

Abstract. *The study deals with the computations of nonlinear fractional circuit analyses in transient states. A numerical solver for FDAE (Fractional Differential-Algebraic Equations) is recalled, which bases on SubIval (acronym for "the subinterval-based method"). In the study, a nonlinear fractional problem was deliberately selected and tuned so as to cause issues occuring only in extreme cases, but motivating the further development of the numerical method. A development idea arises after observations of the error estimation, the influence of the time step adaptivity and the polynomial order.*

Streszczenie. *Praca dotyczy obliczeń w analizach nieliniowych obwodów ułamkowych w stanach nieustalonych. Przywołano solwer numeryczny dla ułamkowych równań różniczkowo-algebraicznych, który opiera się na metodzie numerycznej SubIval (akronim od anglojęzycznej nazwy "the subinterval-based method", czyli "metody podprzedziałów"). W badaniu celowo wybrano i dostrojono nieliniowe zagadnienie ułamkowe tak, aby wywołać komplikacje występujące jedynie w skrajnych przypadkach, ale motywujące do dalszego rozwoju wspomnianej metody numerycznej. Pomysł rozwoju metody pojawia się po obserwacjach oszacowania błędu, wpływu automatycznego dostosowania kroku czasowego i rzędu wielomianów aproksymujących.* ***(Badanie problemów w skrajnych przypadkach solwera numerycznego dla nieliniowych obwodów ułamkowych)***

Keywords: numerical solver, fractional calculus, nonlinear problem, circuit analysis, polynomial order, time step adaptivity
Słowa kluczowe: solwer numeryczny, rachunek różniczkowy ułamkowego rzędu, zagadnienie nieliniowe, analiza obwodu, rząd wielomianu, automatyczny dobór kroku czasowego

Introduction

Fractional calculus is a branch of mathematics that deals with the introduction of derivatives and integrals of a fractional order [1, 2]. This study only deals with fractional derivatives of the order $\alpha \in [0, 1]$ in the Caputo definition [3], but with the initial time in $-\infty$:

$$(1) \qquad {}_{-\infty}\mathrm{D}_t^{\alpha} x(t) = \frac{1}{\Gamma(1-\alpha)} \int_{-\infty}^{t} \frac{x^{(1)}(\tau)}{(t-\tau)^{\alpha}} \mathrm{d}\tau,$$

In practice, for $x(t)$ a certain value is given for an initial time instance of the analysis ($t = t_0$), for simplicity further on called the initial condition (as in analyzes with first order derivatives), where it is then assumed that the variable under the fractional derivative was invariant in $t \in (-\infty, t_0]$. These so-called initial conditions are then given for all variables under fractional derivatives. In further parts of the paper the notation is simplified into:

$$(2) \qquad \mathrm{D}^{\alpha} x(t) = {}_{-\infty}\mathrm{D}_t^{\alpha} x(t)$$

In recent years, an increase in interest in the field of fractional calculus has been observed due to its use in various analyzes such as: viscoelasticity [4], control theory [5] (e.g., due to fractional PID controllers [6]), heat transfer [7] and, to which this study relates to, electrical engineering. In electrical engineering, fractional calculus is also used for several reasons, including the modeling of complex devices [8, 9] and basic elements (such as ferromagnetic core coils [10, 11] and supercapacitors [12, 13]). As for circuit theory, fractional order elements have been introduced, where a linear fractional coil and a linear fractional capacitor are described by the fractional differential equations:

$$(3) \qquad L_{(\alpha)} \mathrm{D}^{\alpha} i(t) = u(t),$$

and:

$$(4) \qquad C_{(\beta)} \mathrm{D}^{\beta} u(t) = i(t),$$

respectively. The introduction of such elements has caused some controversies and debates in the past about their relevance to classical circuit theory [14, 15, 16, 17]. In some recent papers the introduction of fractional elements as an extension of classical elements is still suggested, including the possibility of expanding electromagnetic field theory itself by including fractional derivatives [18]. In this paper, however, the fractional elements are treated as more complex ones, useful in phenomenological modeling and not derived from certain assumptions of electromagnetic field theory (as it can be done for classical elements). As mentioned above, these elements are treated as more complex creations, but still describing the relationship between the voltage and the current. The units themselves point out the artificiality of the fractional elements, where the parameter $L_{(\alpha)}$ has the unit $\mathrm{H} \cdot \mathrm{s}^{\alpha-1}$, while $C_{(\beta)}$ has the unit $\mathrm{F} \cdot \mathrm{s}^{\beta-1}$. Together with the linear elements, a nonlinear fractional coil and nonlinear fractional capacitor can also be introduced. Their symbols are shown in Fig. 1.

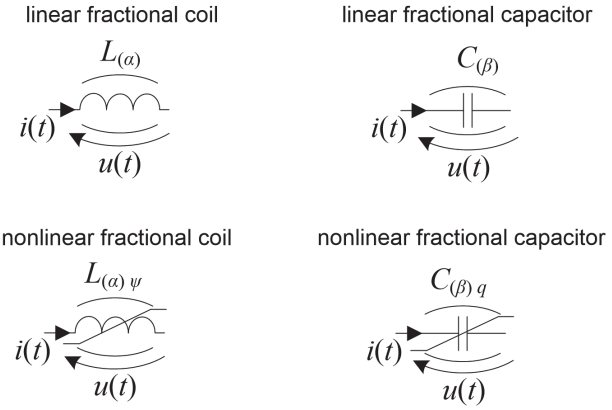


Fig. 1. Circuit elements described through fractional differential equations

The nonlinear fractional coil is described by the fractional differential equation:

$$(5) \qquad \mathrm{D}^{\alpha} \psi = u(t),$$

where the artificial variable ψ is used with the unit $\mathrm{Wb} \cdot \mathrm{s}^{\alpha-1}$. Meanwhile, the nonlinear fractional capacitor is described by:

$$(6) \qquad \mathrm{D}^{\beta} q = i(t).$$

where q has the unit $\mathrm{C} \cdot \mathrm{s}^{\beta-1}$. The description of these elements is completed by the addition of a ψ-i relationship for the fractional coil, while for the fractional capacitor the q-u relationship needs to be given.

Form of solved FDAE and available solvers

The introduction of fractional elements to circuit analyses can significantly increase the difficulty of solving problems. However, it depends on the other types of elements appearing in the circuit and the form of the sought solutions. The vastly different cases are such that when we are dealing with linear circuits with sinusoidal sources and we are seeking an AC steady-state – the increase in the difficulty is trivial (the impedances take slightly different forms), while for problems with nonlinear elements and sources described by various time functions where a transient solution is sought – we do not have access to analytical solutions and it is necessary to use advanced numerical methods. Publicly available solvers are a rarity: for many years the most popular are Garrappa's solvers [19, 20, 21], for many years built for Matlab and recently also implemented in other environments [22]. An alternative is the solver described in [26] built specifically on a form derived for nonlinear problems of circuit theory (although also proven to be useful in other analyses [8, 9, 23]) and based on the numerical method currently known under its acronym, SubIval (from the previous name: "subinterval-based method", where its first appearance was in [24]).

The solver basing on SubIval will be studied in this paper. It is available at [25]. It is mainly built to run under GNU Octave (meaning that, when slightly modified, it can also run under Matlab) running on Windows, where the SubIval ActiveX dynamic-link library needs to be registered (the automatic registration available on the page requires .NET Framework 2.0.50727).

The general form of the considered system of FDAE (Fractional Differential-Algebraic Equations) has the form [26]:

$$(7)\quad \begin{cases} \mathbf{M}_{\mathrm{I}}\mathbf{y}(t) + \mathbf{M}_{\mathrm{II}}\mathbf{x}(t) = \mathbf{T}\mathbf{v}(t) + \begin{bmatrix} \mathbf{0}_{n_y - n_{\mathrm{NL}}} \\ \mathbf{F}_{\mathrm{NL}}(\mathbf{w}(t)) \end{bmatrix} \\ \mathbf{D}^{\alpha}\mathbf{x}(t) + \mathbf{M}_{\mathrm{III}}\mathbf{y}(t) + \mathbf{M}_{\mathrm{IV}}\mathbf{x}(t) = \mathbf{0}_{n_x} \end{cases}$$

where the sought solution vector is:

$$(8)\qquad \mathbf{w}(t) = \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{x}(t) \end{bmatrix},$$

with $\mathbf{x}(t)$ being the vector (of length n_x) of unknowns appearing under fractional derivatives, while $\mathbf{y}(t)$ (of length n_y) is the vector of the remaining unknowns. The matrices: $\mathbf{M}_{\mathrm{I}}$ (of size $n_y \times n_y$), $\mathbf{M}_{\mathrm{II}}$ (of size $n_y \times n_x$), $\mathbf{M}_{\mathrm{III}}$ (of size $n_x \times n_y$), $\mathbf{M}_{\mathrm{IV}}$ (of size $n_x \times n_x$) and $\mathbf{T}$ (of size $n_y \times n_v$) represent the linear relationships between the variables. $\mathbf{D}^{\alpha}\mathbf{x}(t)$ is the vector of fractional derivatives of the subsequent variables in $\mathbf{x}(t)$, where the derivative orders are given in α. $\mathbf{F}_{\mathrm{NL}}(\mathbf{w}(t))$ represents the n_{NL} nonlinear dependencies, where each dependency is given by a nonlinear function of only one variable of $\mathbf{w}(t)$; the information on which variables are the arguments of these functions is stored as the indices of these variables in the auxiliary vector $\mathbf{i}_{\mathrm{arg}}$. Finally, the $\mathbf{0}_k$ notation is applied to express zero column vectors of size k. As mentioned before, the (7) form has been derived for problems of circuit theory where the following elements appear: autonomous sources described by arbitrary time functions, controlled sources described through strictly monotonically increasing or decreasing functions, linear or nonlinear resistors, linear or nonlinear coils (which can be nonlinear fractional coils) and linear or nonlinear capacitors (which can also be nonlinear fractional capacitors). The above mentioned conditions about increasing functions are there to ensure a unique solution. In further parts of the paper the variables (and vectors) are given with their time dependency only when the Author has felt that this should be emphasized (hence, e.g., $\mathbf{x}(t)$ is written as $\mathbf{x}$).

Information on the solver and its usage

The solver basing on SubIval has been extensively described in many publications, details on the development of the solver and the method itself can be found in [23, 27, 28, 29]. The current form of the numerical solver was established during the publication of [26]. The solver itself appears in both a "primitive" version (with a constant time step size) and as a solver with time step size adaptivity following a local truncation error estimation (this has been described with detail in [30]). The simpler variant of the solver is actually applied at one time in the more advanced solver in order to determine an approximate level of the typical values for the sought variables, where the result is then applied when solving nonlinear equations and during the estimation of the local truncation error (so that a relative error can be used). The solver accepts the following input data:

- the essentials for (7), i.e., the $\mathbf{M}_{\mathrm{I}}$, $\mathbf{M}_{\mathrm{II}}$, $\mathbf{M}_{\mathrm{III}}$, $\mathbf{M}_{\mathrm{IV}}$ and $\mathbf{T}$ matrices, the form of the nonlinear dependencies, the orders of fractional derivatives, the initial conditions (as they have been defined in the first section) of $\mathbf{x}(t)$ and the time functions representing the autonomous source waveforms,
- the e_{ctrl} parameter, which is the "control error", i.e., the error to which the maximum of local truncation error estimations (for the variables in $\mathbf{x}(t)$) should tend to and to which the time step size should be adjusted,
- $e_{\max}$, which is the maximum allowable value of e_{ctrl} (if this error has been exceeded in a time step for a time instance t_k, instead of adjusting the time step size for the next time instance (i.e., adjusting Δt to get $t_{k+1} = t_k + \Delta t$), the computations are repeated by modifying what is t_k (through $t_k = t_{k-1} + \Delta t$),
- $p_{\max}$, which is the maximum applied order of the approximating polynomials when obtaining the numerical solution; this is a very important parameter, whose influence will be studied in this paper,
- the $\Delta t_{\min}$ and $\Delta t_{\max}$ parameters being the minimum and maximum time step size, respectively (if $\Delta t = \Delta t_{\min}$ then the computations allow $e_{\max}$ to be exceeded).

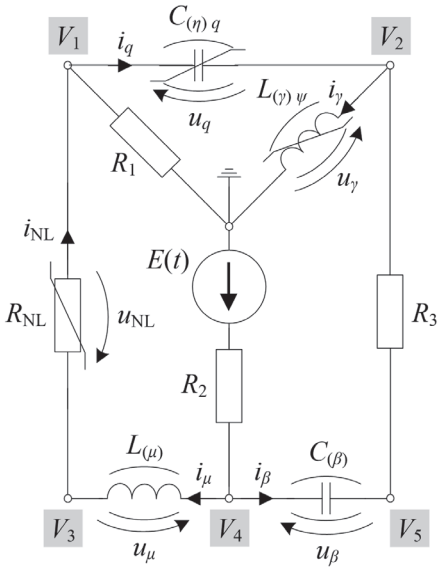
Additional, optional input data consists of:

- the test trust coefficient, which is a value in the range $(0, 1]$ and tells how the maximum values in the "primitive" solver test run are used as the initial referential values in the time step size adaptive solver (the default value of this parameter, which is also applied for the computations of this paper, is 0.5),
- Δt_{test}, which is the time step size applied in the mentioned test run (by default, the $\Delta t_{\max}$ value later applied in the adaptive solver is used).

When studying the solver during its development rarely there have been occurrences of numerical problems that could be called greatly troublesome. One could usually avoid the most typical issues (long computation time, high estimated error values for fast transients) through a careful selection of e_{ctrl}, $e_{\max}$, $\Delta t_{\min}$ and $\Delta t_{\max}$. This is why there haven't been many mentions of these problems in previous papers. In this paper, on the other hand, some extreme cases will be shown, where numerical complications may occur.

Computational example

For this study, a circuit (Fig. 2) will be used that includes the previously mentioned fractional elements.



$E(t) = E_1 \arctan((2/\pi)\, c_a \sin(\omega_1 t))\,(1 - \exp(-t/T_e))$
$E_1 = 300\,\mathrm{V}$
$c_a = 50$ or $c_a = 5000$
$\omega_1 = 2\pi f_1$
$f_1 = 50\,\mathrm{Hz}$
$R_1 = 100\,\Omega$
$R_2 = 20\,\Omega$
$R_3 = 300\,\Omega$
$L_{(\mu)} = 2.4\,\mathrm{H \cdot s}^{\mu-1}$
$\mu = 0.87$
$C_{(\beta)} = 0.03\,\mathrm{F \cdot s}^{\beta-1}$
$\beta = 0.55$

nonlinear and fractional nonlinear elements:

nonlinear resistor R_{NL}:
u_{NL}-i_{NL} dependency described by:
$i_{\mathrm{NL}}(u_{\mathrm{NL}}) = i_0 \operatorname{sgn}(u_{\mathrm{NL}})\,|u_{\mathrm{NL}}/i_0|^m$
with: $i_0 = 2 \cdot 10^{-3}\,\mathrm{A}$, $u_0 = 9\,\mathrm{V}$, $m = 7.3$

fractional, nonlinear coil $L_{\psi\,(\gamma)}$:
derivative order $\gamma = 0.91$
ψ-i_ψ dependency described by:
$\psi(i_\psi) = \psi_0 \arctan(i_\psi/i_{\psi 0})$,
with: $6 \cdot 10^{-3}\,\mathrm{Wb \cdot s}^{\gamma-1}$, $i_{\psi 0} = 10^{-3}\,\mathrm{A}$

fractional, nonlinear capacitor $C_{q\,(\eta)}$:
derivative order $\eta = 0.63$
q-u_q dependency described by:
$q(u_q) = c_1 u_q + c_3 u_q^{\,3}$
with: $c_1 = 2 \cdot 10^{-5}\,\mathrm{F \cdot s}^{\eta-1}$, $c_3 = 2 \cdot 10^{-6}\,\mathrm{V^{-2} \cdot F \cdot s}^{\eta-1}$

Fig. 2. Fractional nonlinear problem selected for the study of the SubIval solver

All the parameters of the problem are given on the figure. The remaining information that is required consists of the input parameters for the solver (which will vary) and the determination of the time interval $t \in [t_0, t_{\max}]$ for which the problem is solved, where in this study $t_0 = 0$ and $t_{\max} = 1.3T$. The vector of sought unknowns $\mathbf{w}$ is in this case divided into those appearing under the fractional derivatives of the orders:

$$\alpha = [\mu\ \beta\ \gamma\ \eta], \tag{9}$$

in the form of:

$$\mathbf{x} = [i_\mu\ u_\beta\ \psi\ q]^{\mathrm{T}} \tag{10}$$

and the remaining unknowns in the form of:

$$\mathbf{y} = [V_1\ V_2\ V_3\ V_4\ V_5\ u_{\mathrm{NL}}\ i_{\mathrm{NL}}\ u_\mu\ i_\beta\ u_\psi\ i_\psi\ u_q\ i_q\ ]^{\mathrm{T}}. \tag{11}$$

The vector of source time functions consists solely of the function for the voltage source:

$$\mathbf{v}(t) = [E(t)]. \tag{12}$$

Because we are dealing with such an advanced problem there are limitations to how a referential solution can be obtained. Hence, this solution will be obtained also from the studied solver, but with a careful selection of the input parameters so as to obtain a sufficiently accurate and reliable result. In order to obtain the referential solution, $p_{\max} = 4$, $\Delta t_{\max} = T_1/200$ and $\Delta t_{\max} = T_1/4 \cdot 10^4$ have been used, along with $e_{\max} = 10^{-2}\,\%$ and $e_{\mathrm{ctrl}} = 10^{-3}\,\%$. The solution in the form of waveforms of the variables of the $\mathbf{x}(t)$ vector have been presented in Fig. 3.

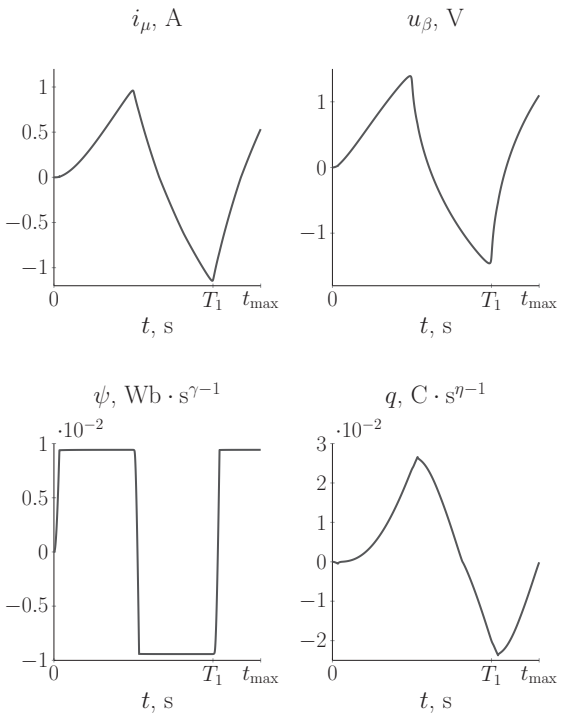


Fig. 3. Results for the referential solution of the selected nonlinear fractional circuit problem

How the maximum local truncation error estimation and the time step size have changed is shown in Fig. 4. In a few instances the error estimation has exceeded $e_{\max}$, however, these results have been allowed since the time step size was below $\Delta t_{\min}$. The selected error tolerance criteria have turned out to be very demanding – the computation time was 42.81 seconds. Normally, one could say that the computation time is too long and the error tolerance should be increased, but this result is intended as a referential solution, so the computation time has not been the most important criterion here so far.

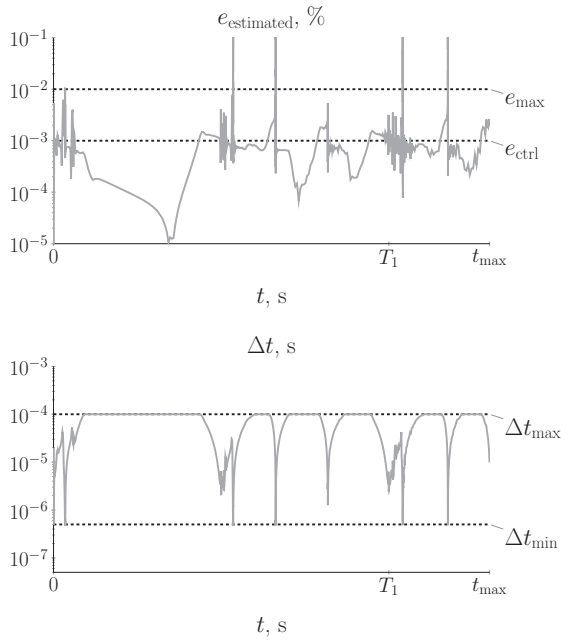


Fig. 4. Time step adaptivity results; top: maximum values of the local truncation error estimation; bottom: time step size

A more difficult case for the studied problem is when $c_{\text{a}} = 5000$. For this case the maximum time step size has been increased to $\Delta t_{\min} = T_1/5{\cdot}10^3$ (for reasons explained in the next section). In this case we will experience faster changes in the source waveform, which will also adjust the size of the time step to smaller values, which in turn will increase the computation time. That, however, is a typical feature of the solver. Untypical, rarely occurring features can also be observed for other selections of the solver options (this is discussed in the next section). The solution for this case is shown in Fig. 5. It took 46.2 seconds to obtain the solution. One can observe that $e_{\max}$ has been exceeded more times in this case. The error estimation results and time step sizes are shown for this more difficult case in Fig. 6. Naturally, a numerical solution is one where we allow some errors; however, the large value of the estimated local truncation error does not indicate significant errors in the entire solution. For example, some minor losses of information about the dynamics could occur here, where these dynamics could have occurred relatively quickly in comparison to the total studied time interval and the applied $\Delta t_{\min}$. The results obtained in this section will be used in further parts of the paper. A measure of the differences in computation results (between newly obtained solutions and the referential solution) will be defined in the form of the numerical evaluation of the integral:

$$\epsilon_i = \frac{1}{t_{\max} - t_0} \int_{t=t_0}^{t_{\max}} \frac{|x_i - x_{i\,\text{ref}}|}{x_{i\,\max}} \cdot 100\,\%, \tag{13}$$

where $x_{i\,\text{ref}}$ represents the waveform of the referential solution, while x_i is the one of the newly obtained solution; $x_{i\,\max}$ is the maximum value of the referential solution. These integrals are evaluated for every variable in $\mathbf{x}$. Of course, it is also important to remember that because of the acceptance of some inaccuracies, we will not necessarily refer to the referential solution as always being the most accurate. Hence, additionally, a critical approach is applied to the referential solution, causing the integral to be computed twice, where the nodes for the numerical evaluation are selected first by selecting those produced by the referential solution and then

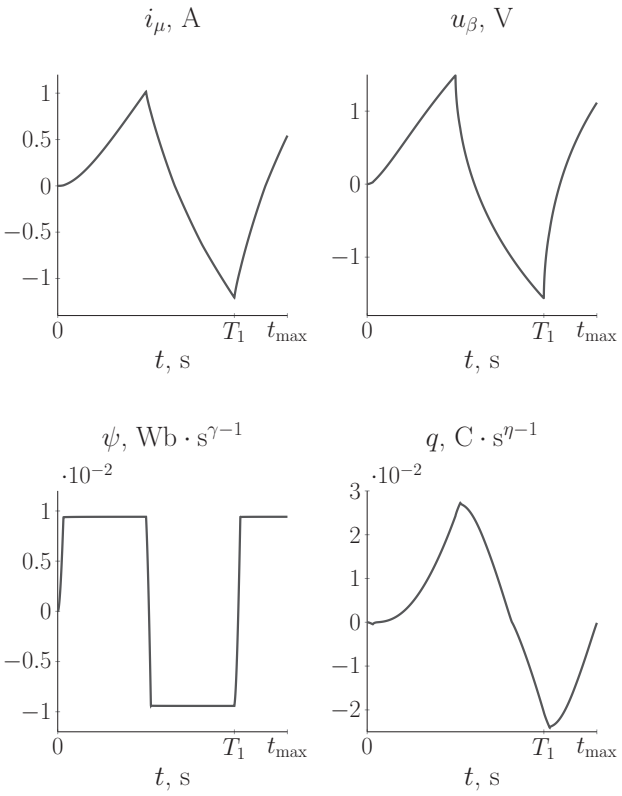


Fig. 5. Results for the referential solution of the selected nonlinear fractional circuit problem when $c_{\text{a}} = 5000$

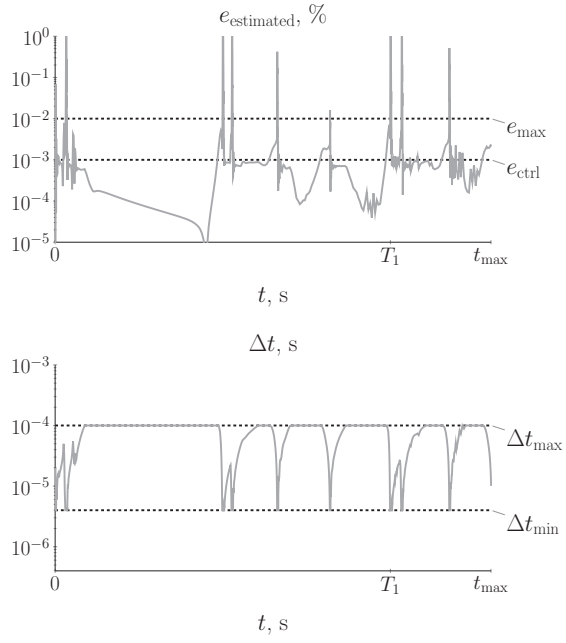


Fig. 6. Time step adaptivity results for the case when $c_{\text{a}} = 5000$: a) maximum values of the local truncation error estimation, b) time step size

those of the newly obtained solution. In both variants, spline interpolation is applied. The maximum of all the computed ϵ_i values is then referred to as $\epsilon_{\max}$.

Study of numerical complications

With a reasonable selection of e_{ctrl} and $e_{\max}$, and a careful selection of $\Delta t_{\min}$ and $\Delta t_{\max}$, the solver efficiently obtains a solution even for very complex problems. As will be shown, some problems are more demanding and not only require a significant computation time (because of strong nonlinearities) but can induce features in a solution of which it is difficult to identify the cause. In this section, we will study purposely induced extreme cases of unwanted features in the results when applying the FDAE solver basing on SubIval. Moreover, the parameters of the problem itself have been adjusted from the beginning in such a way that it was possible to observe these undesired phenomena. On the other hand, the solver parameters for the referential solution have been selected so that these will not contain the mentioned features.

Naturally, the first thing that comes to mind when the local truncation error estimation value has exceeded $e_{\max}$ is to decrease the value of $\Delta t_{\min}$. This has been done for the $c_{\mathrm{a}} = 50$ case first, where $\Delta t_{\min} = T_1/(6 \cdot 10^4)$ The results are visually indistinguishable from the referential solution waveforms and $\epsilon_{\max}$ computed between those results and the referential solution is $1.24 \cdot 10^{-3}\,\%$. They do, however, have some features pointing at numerical complications, which can best be described by observing the error estimation and Δt history of the simulation (Fig. 7). The error estimation values rapidly change, though in the vicinity of e_{ctrl}. On the other hand, the time step size seems to tend to rise and fall rapidly near $\Delta t_{\min}$. For $c_{\mathrm{a}} = 5000$ this behavior

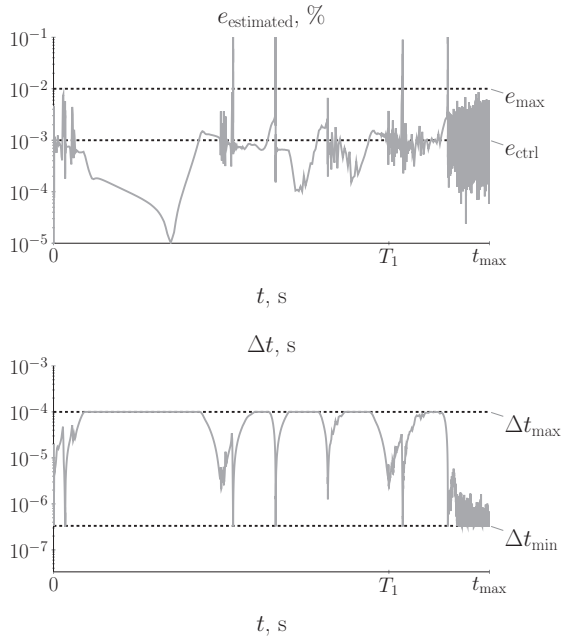


Fig. 7. Time step adaptivity results for $c_{\mathrm{a}} = 50$ and $\Delta t_{\min} = T_1/(6 \cdot 10^4)$; top: maximum values of the local truncation error estimation; bottom: time step size

is seen even for larger values of $\Delta t_{\min}$. It is enough to lower its value to $T_1/10^4$ to observe this type of phenomenon. Additionally, the problem appears earlier on the time axis when decreasing $\Delta t_{\min}$ even further as shown in Fig. 8 for the case when $\Delta t_{\min} = T_1/(4 \cdot 10^4)$.

Even with the obvious numerical issues the results are

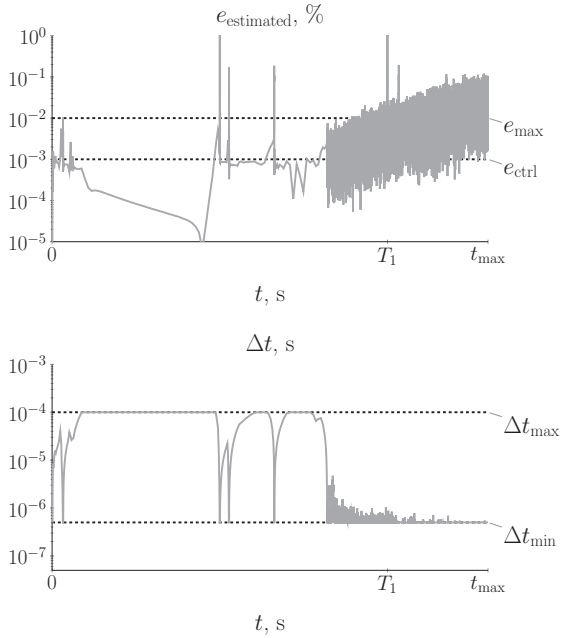


Fig. 8. Time step adaptivity results for $c_{\mathrm{a}} = 5000$ and $\Delta t_{\min} = T_1/(4 \cdot 10^4)$; top: maximum values of the local truncation error estimation; bottom: time step size

still visibly indistinguishable from the referential result with $\epsilon_{\max} = 5.466 \cdot 10^{-3}\,\%$. The first guess was that it is a problem caused by too many computation nodes. However, not only analyses with far more have been done in the past [26, 28], but also the phenomenon started at different node indices (differences were of the order of several hundred). The beginning of the phenomenon could rather be attributed to the vicinity of similar time instances until one includes the results for $\Delta t_{\min} = T_1/(4 \cdot 10^4)$ in the comparison (Fig. 9). The more difficult case of $c_{\mathrm{a}} = 5000$ is, hence, further studied more as here the extreme cases of numerical issues arise more significantly.

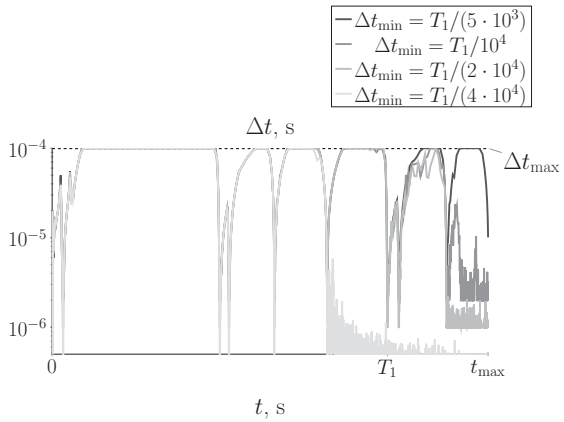


Fig. 9. Time step size adaptivity results for $c_{\mathrm{a}} = 5000$ and various values of $\Delta t_{\min}$

For subsequent experiments, the values of $e_{\max}$ and e_{ctrl} have been increased to $0.1\,\%$ and $10^{-2}\,\%$, respectively. In this way, it will be possible to adopt the referential solutions as the more reliable ones. Also, the maximum time step size will also be increased in this case so that $\Delta t_{\min} = T_1/50$. Selected results ($\epsilon_{\max}$, the computation time and the estimation of the beginning of where the numerical complications appear) are presented in Tab. 1. Additionally, a test with $\Delta t_{\min} = T_1/(8 \cdot 10^4)$ was also attempted but this trial was abandoned when the computations took longer than 15 min-

Table 1. Results for the simulations for $c_\mathrm{a} = 5000$ for various Δt_min (n.a. means that the numerical issues are either not appearing or they did not result in Δt_min falling and staying at a low level

Δt_min	ϵ_max	computation time, s	numerical issues	
			time instance index	time node, s
$T_1/(5 \cdot 10^3)$	$1.36 \cdot 10^{-2}$	17.5	n.a.	
$T_1/10^4$	$5.40 \cdot 10^{-3}$	16.8	n.a.	
$T_1/(2 \cdot 10^4)$	$5.49 \cdot 10^{-3}$	21.4	n.a.	
$T_1/(4 \cdot 10^4)$	$4.85 \cdot 10^{-3}$	185.7	800	0.021
$T_1/(6 \cdot 10^4)$	$6.61 \cdot 10^{-3}$	738.3	370	0.02

utes. In the results above, one can notice typical occurrences when the discussed FDAE solver is used. The first is that when Δt_min is reduced, when there are no numerical complications, the accuracy of the solution has been increased. Secondly, when Δt_min is limited, some dynamics are omitted, which turned out to be of little importance from the perspective of the entire solution. Finally, in case of numerical issues, depending on Δt_min, the solution may go through the final time interval with values close to Δt_min or get stuck earlier, which greatly increases the computation time because with more time nodes being included the computation time rises linearly (as it was studied in [31]).

Study of the influence of the polynomial order

At this point, it should be recalled that $p_\mathrm{max} = 4$ for all computations up to this point. This is a value that the Author often used for computations when using the discussed FDAE solver. After deliberately invoking the extreme cases and investigating possible imperfections, one can notice that the occurring numerical issues may be a result of a simple factor that lies at the core of the SubIval method. The method itself uses polynomial approximations and for each subinterval (as detailed in [27] for example) Lagrange interpolation is performed. By default, after exceeding the number of time nodes equal to $p_\mathrm{max} + 1$ the polynomial order of p_max is used and, to estimate the local truncation error, an additional polynomial of $p_\mathrm{max} - 1$ is applied temporarily. With a large change in the time step size for higher order approximating polynomials, there may be a loss of accuracy when comparing the results for the p_max order with those obtained using one order lower. This is an extreme case, which can be prevented simply by a reasonable choice od Δt_min and Δt_max, otherwise this undesired feature will only escalate and the time step size will always fall to Δt_min.

The logical next step was to investigate the solutions for the case where $p_\mathrm{max} = 3$. The case of $c_\mathrm{a} = 5000$ continues to be studied, but the case of $c_\mathrm{a} = 50$ will also be recalled at the end of this section. It turns out that reducing the order of the polynomials practically eliminated the existing numerical problems. For an extreme case with $\Delta t_\mathrm{min} = T_1/(2 \cdot 10^7)$ (error and time step size plots are given in Fig. 10), Δt decreased to the level of Δt_min only a few times, so there was no need to allow a local truncation error estimation greater than e_max. The results (computation time and ϵ_max) for exemplary cases with various Δt_min are given in Tab. 2.

Table 2. Results for the simulations for $c_\mathrm{a} = 5000$ for various Δt_min with $p_\mathrm{max} = 3$, $e_\mathrm{max} = 0.1\,\%$ and $e_\mathrm{ctrl} = 10^{-2}\,\%$

Δt_min	computation time	ϵ_max
$T_1/(4 \cdot 10^4)$	37.1	$5.18 \cdot 10^{-3}$
$T_1/(8 \cdot 10^4)$	43.6	$4.53 \cdot 10^{-3}$
$T_1/(2 \cdot 10^5)$	62.4	$5.16 \cdot 10^{-3}$
$T_1/(2 \cdot 10^6)$	56.1	$5.01 \cdot 10^{-3}$
$T_1/(2 \cdot 10^7)$	63.8	$5.01 \cdot 10^{-3}$

It can be noticed, that in perspective of ϵ_max the result did not change much. This was because the omission of certain transients (at larger Δt_min values) had no effect on the solution in a more general sense. Tests for $e_\mathrm{ctrl} = 10^{-2}\,\%$ oraz $e_\mathrm{max} = 10^{-3}\,\%$ (with a temporary decrease of $\Delta t_\mathrm{max} = T_1/200$) have also been performed – in this case the discussed numerical complications also did not occur; cases were simulated up to $\Delta t_\mathrm{min} = T_1/(2{\cdot}10^6)$, and further attempts had a too long computation time (Tab. 3).

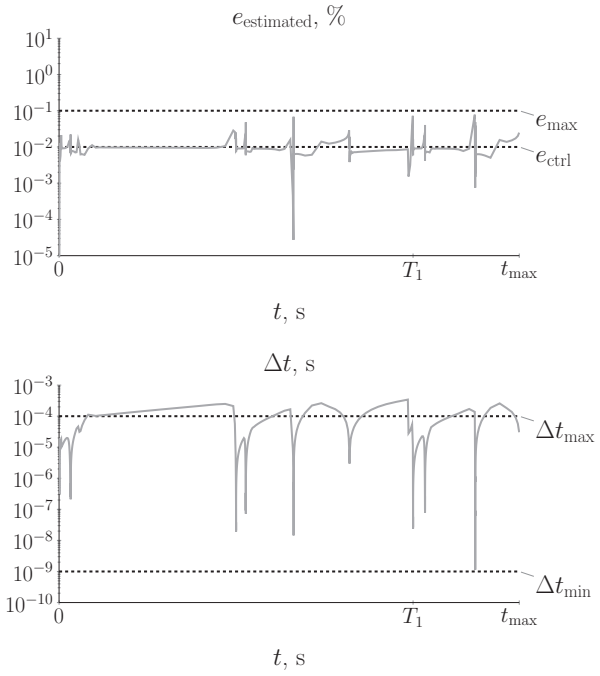


Fig. 10. Time step size adaptivity results for $c_\mathrm{a} = 5000$ and various values of Δt_min with $p_\mathrm{max} = 3$

Table 3. Results for the simulations for $c_\mathrm{a} = 5000$ for various Δt_min with $p_\mathrm{max} = 3$, $e_\mathrm{max} = 10^{-2}\,\%$ and $e_\mathrm{ctrl} = 10^{-3}\,\%$

Δt_min	computation time	ϵ_max
$T_1/(2 \cdot 10^4)$	63.5	$5.78 \cdot 10^{-3}$
$T_1/(4 \cdot 10^4)$	69.4	$5.68 \cdot 10^{-3}$
$T_1/(6 \cdot 10^4)$	78.2	$5.69 \cdot 10^{-3}$
$T_1/(2 \cdot 10^5)$	123.8	$5.68 \cdot 10^{-3}$
$T_1/(2 \cdot 10^6)$	173.9	$5.69 \cdot 10^{-3}$

The results indicate that the solutions did not vary much from the previously obtained (for larger error tolerances) in terms of their comparison with the referential solution. As for the lack of numerical complications, however, lowering the order of the polynomials is not a remedy to be used in every simulation; as mentioned earlier, orders of $p_\mathrm{max} = 4$ and higher were used frequently because they often required much less computation time to obtain a solution with a similar accuracy. To investigate how the order of the polynomials affects the computation time for the less troublesome case (where $c_\mathrm{a} = 50$), additional simulations were performed for

$\Delta t_{\min} = T_1/(8 \cdot 10^4)$, the results of which are presented in Tab. 4 (where the values of $\epsilon_{\max}$ are also shown).

Table 4. Results for the simulations for $c_{\mathrm{a}} = 50$ for various $p_{\max}$ ($e_{\max} = 0.1\,\%$ and $e_{\mathrm{ctrl}} = 10^{-2}\,\%$)

$p_{\max}$	computation time	$\epsilon_{\max}$
2	723.97	$3.38 \cdot 10^{-3}$
3	40.30	$1.51 \cdot 10^{-3}$
4	19.59	$5.41 \cdot 10^{-3}$

Conclusions

The paper explores the extreme cases of numerical complications that occur with the time step size adaptive FDAE solver basing on SubIval. When present, these complications only affect the computation time, as they cause the time step to drop to the minimum given value and barely increase until the end of the analysis (while not affecting the general accuracy of the solution). It has been identified that these complications are not caused by a large number of time nodes or the specificity of the selected problem itself (as the complications occurred in various time instances). Reducing the order of the approximating polynomials (which are used in the core of SubIval) to $p_{\max} = 3$ in the case of the examined problem practically eliminated the unwanted phenomenon, where it was possible to obtain satisfactory solutions. However, the choice of lower orders of polynomials in not always the best option, as demonstrated in the less extreme case of $c_{\mathrm{a}} = 50$, where higher order polynomials allowed to obtain the solution with a similar accuracy in a much shorter period of time (as shown in Tab. 4). These conclusions therefore motivate future research that aims at the implementation of polynomial order adaptivity. This requires a significant contribution to the further development of the discussed solver and the improvement of the subinterval dynamics algorithm itself (described in detail in [27]) at the core of SubIval.

***Authors**: Marcin Sowa, PhD, DSc, Department of Electrical Engineering and Computer Science, Faculty of Electrical Engineering, Silesian University of Technology, ul. Akademicka 2A, 44-100 Gliwice, Poland, email: marcin.sowa@polsl.pl*

REFERENCES

[1] Oldham K.B., Spanier J., *The Fractional Calculus Theory and Applications of Differentiation and Integration to Arbitrary Order*, Academic Press, New York (1974)
[2] Podlubny I., *Fractional Differential Equations. An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*, Academic Press, New York (1998)
[3] Caputo M., Linear models of dissipation whose Q is almost frequency independent – II, *Geophysical Journal International*, 13 (1967), nr 5, 529-539
[4] Qing J., Zhou S., Wu J., Shao M., Primary and secondary resonance responses of fractional viscoelastic PET membranes, *Communications in Nonlinear Science and Numerical Simulation*, 116 (2023), 106810
[5] Laski P.A., Fractional-order feedback control of a pneumatic servo-drive, *Bull. Pol. Ac. Tech.*, 67 (2019), nr 1, 53-59
[6] Matusiak M., Ostalczyk P., Problems in solving fractional differential equations a microcontroller implementation of an FOPID controller, *Arch. Electr. Eng.*, 68 (2019), nr 3, 565-577
[7] Brociek R., Słota D., Król M., Matula M., Kwaśny W., Comparison of mathematical models with fractional derivative for the heat conduction inverse problem based on the measurements of temperature in porous aluminum, *Int. J. Heat Mass Transf.*, 143 (2019), 118440
[8] Majka Ł., Sowa M., Exciter fractional model and its susceptibility on parameter changes, *Poznan University of Technology Academic Journals. Electrical Engineering*, 104 (2020), 87-98
[9] Sowa M., Majka Ł., Wajda K., Excitation system voltage regulator modeling with the use of fractional calculus, *AEU - International Journal of Electronics and Communications*, 159 (2023), 154471
[10] Schäfer I., Krüger K., Modelling of lossy coils using fractional derivatives, *J. Phys. D: Appl. Phys.*, 41 (2008), nr 4, 8 pages
[11] Sowa M., Majka Ł., Ferromagnetic core coil hysteresis modeling using fractional derivatives, *Nonlinear Dynamics*, 101 (2020), 775-793
[12] Mitkowski W., Skruch P., Fractional-order models of the supercapacitors in the form of RC ladder networks, *Bull. Pol. Ac.: Tech.*, 61 (2013), nr 3, 581-587
[13] Lewandowski M., Orzyłowski M., Fractional-order models: The case study of the supercapacitor capacitance measurement, *Bull. Pol. Ac.: Tech.*, 65 (2017), nr 4, 449-567
[14] Kaczorek T., Positive electrical circuits with zero transfer matrices and their discretization, *Computer Applications in Electrical Engineering*, 14 (2016), 1-13
[15] Sikora R., Fractional derivatives in electrical circuit theory – critical remarks, *Archives of Electrical Engineering*, 66 (2017), nr 1, 155-163
[16] Ostalczyk P., Jednak: "Rachunek różniczkowo-całkowy niecałkowitych rzędów". Riposta na artykuł Ryszarda Sikory pt.: "Pochodne ułamkowe w teorii obwodów elektrycznych. Uwagi krytyczne.", *Przegląd Elektrotechniczny*, 92 (2016), nr 10, 175-180
[17] Latawiec K., Stanisławski R., Łukaniszyn M., Czuczwara W., Rydel M., Fractional-order modeling of electric circuits: modem empiricism vs. classical science, *2017 Progress in Applied Electrical Engineering (PAEE)*, Kościelisko, Poland, (2017), 1-4
[18] Stefański T.P., Gulgowski J., Electromagnetic-based derivation of fractional-order circuit theory, *Commun Nonlinear Sci Numer Simulat*, 79 (2019), 104897
[19] Garrappa R., Numerical Solutions of Fractional Differential Equations: A Survey and a Software Tutorial, *Mathematics*, 6 (2018), nr 2, 1-16
[20] Garrappa R., Trapezoidal methods for fractional differential equations: Theoretical and computational aspects, *Mathematics and Computers in Simulation*, 110 (2015), 96-112
[21] Garrappa R., Predictor-corrector PECE method for fractional differential equations. https://www.mathworks.com/matlabcentral/fileexchange/32918-predictor-corrector-pece-method-for-fractional-differential-equations. Accessed: 2023-06-08
[22] https://juliahub.com/ui/Packages/FractionalDiffEq/K86Gm/0.3.1. Accessed: 2023-06-08
[23] Sowa M., Dziedzic K., Expansion of a solver for nonlinear fractional problems - the inclusion of time delays, *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Międzyzdroje, Poland, (2019), 249-254
[24] Sowa M., A subinterval-based method for circuits with fractional order elements, *Bull. Pol. Ac.: Tech.*, 62 (2014), nr 3, 449-454
[25] https://osf.io/download/x8pdt/. Accessed: 2023-06-08
[26] Sowa M., Numerical solver for fractional nonlinear circuit problems, *2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX)*, Guatemala City, Guatemala, (2019), 6 pages
[27] Sowa M., Application of SubIval in solving initial value problems with fractional derivatives, *Appl. Math. Comput.*, 319 (2018), 86-103
[28] Sowa M., A Harmonic Balance Methodology for Circuits with Fractional and Nonlinear Elements, *Circuits Syst Signal Process*, 37 (2018), nr 11, 4695-4727
[29] Sowa M., Solutions of circuits with fractional, nonlinear elements by means of a SubIval solver. in: *Conference on Noninteger Order Calculus and Its Applications. RRNR 2017: Non-Integer Order Calculus and its Applications*. Series: Lecture Notes in Electrical Engineering 496, (2019)217-228.
[30] Sowa M., A local truncation error estimation for a SubIval solver, *Bull. Pol. Ac.: Tech.*, 66 (2018), nr 4, 475-484
[31] Sowa M., SubIval computation time assessment, *Proceedings of International Interdisciplinary PhD Workshop 2017*, Łódź, Poland, (2017), 15-22