BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 62, No. 3, 2014
DOI: 10.2478/bpasts-2014-0047

A subinterval-based method for circuits with fractional order elements

M. SOWA[*]

Faculty of Electrical Engineering, Silesian University of Technology, Akademicka 10, 44-100 Gliwice, Poland

Abstract. The paper deals with the solution of problems that concern fractional time derivatives. Specifically the author's interest lies in solving circuit problems with so called fractional capacitors and fractional inductors. A numerical method is proposed that involves polynomial interpolation and the division of the entire time interval (for which computations are performed) into subintervals. Analytical formulae are derived for the integro-differentiation according to the Caputo fractional derivative. The rules that concern the subinterval dynamics throughout the computation are also presented in the paper. For exemplary linear circuit problems (AC and transient) involving fractional order elements the solutions have been obtained. These solutions are compared with ones obtained by means of traditional methods.

Key words: fractional derivative, circuit equations, numerical method, polynomial interpolation, subinterval.

1. Introduction

The recently growing interest in the application of fractional derivatives [1, 2] naturally motivates the design of analytical and numerical methods. The author's interest lies mostly in the aspect of fractional derivative models in electric circuits (i.e. for so called fractional capacitors and fractional inductors).

Fractional capacitor models have been lately known to reflect the properties of supercapacitors [3, 4], while fractional inductors are very useful as models of lossy coils [5].

In the case of AC analyses, the fractional models can be substituted by their equivalent complex impedances [6]. However, for transient simulations the matter is much more difficult.

An analytical solution to a system of linear differential equations with fractional orders is given e.g. in [7, 8]. An analytical form naturally can provide a more exact solution yet is sometimes difficult to obtain (especially if a greater number of elements is considered). An alternative approach is to use numerical methods, of which one of the main advantages is that they can be extended to solve nonlinear problems. The nowadays applied numerical methods for solving fractional differential equations mostly base on existing methods for solving systems of ordinary differential equations. Many of these were covered in [9] for the choice of a constant integration step. These are:

- Lubich's backward difference methods (which generally lead to methods of order from 1 to 6);
- generalized Taylor expansion method;
- Adomian's decomposition method.

These methods (which have been described for single equations) can be extended for the solution of sets of fractional order differential equations. Their biggest disadvantage is that they mainly are well described for the case of constant integration steps (i.e. equidistant nodes on the time axis).

This paper presents the application of a numerical method that could solve a general class of linear circuit problems with fractional order elements. The method bases on polynomial interpolations in designated time subintervals

2. Fractional time derivative approximation

Only the case for the derivative of order $0 < \alpha < 1$ is considered in the study because such are reflected by the modeled circuit elements. In this case the Caputo definition of the fractional derivative is given by the operator [10]:

$$ {}^{\mathrm{C}}_{t_\mathrm{c}}D^{\alpha}_{t_\mathrm{d}} x(t) = \frac{1}{\Gamma(1-\alpha)} \int\limits_{t_\mathrm{c}}^{t_\mathrm{d}} \frac{x^{(1)}(\tau)}{(t-\tau)^{\alpha}} \mathrm{d}\tau. \tag{1} $$

If an $m-1$ degree polynomial interpolation is performed on the variable x in some given time interval $t \in [t_\mathrm{a}, t_\mathrm{b}]$ then it can be expressed by means of a linear combination of Lagrange basis polynomials:

$$ \forall t \in [t_\mathrm{a}, t_\mathrm{b}] : x(t) \approx \widetilde{x}(t-t_\mathrm{a}) = \sum_{i=1}^{m} \ell_{m,i}(t-t_\mathrm{a}) \left. x\right|_{t=t_i}, \tag{2} $$

where $\ell_{m,i}$ is an $m-1$ degree polynomial being equal to 1 at t_i and 0 at all other interpolation points t_j:

$$ \ell_{m,i}(t-t_\mathrm{a}) = \prod_{\substack{1 \le j \le m+1 \\ j \ne i}} \frac{t-t_j}{t_i-t_j}. \tag{3} $$

This polynomial is further on used for the approximation of the variable x in selected time intervals. In a power series form the locally defined Lagrange polynomial is further on presented as:

[*] e-mail: marcin.sowa@polsl.pl

$$\ell_{m,i}(t_{\mathrm{loc}}) = \sum_{k=0}^{m-1} b_{i,k} t_{\mathrm{loc}}^k, \tag{4}$$

where

$$t_{\mathrm{loc}} = t - t_{\mathrm{a}}. \tag{5}$$

The local approximation of x is denoted by:

$$\widetilde{x}(t_{\mathrm{loc}}) = \sum_{k=0}^{m-1} a_k t_{\mathrm{loc}}^k. \tag{6}$$

The Caputo derivative of (6), with $t \in [t_{\mathrm{c}}, t_{\mathrm{d}}]$ is given by the general formula:

$$ {}^{\mathrm{C}}_{t_{\mathrm{c}}}D^{\alpha}_{t_{\mathrm{d}}}\widetilde{x}(t - t_{\mathrm{a}}) = \frac{1}{\Gamma(1-\alpha)} \int_{t_{\mathrm{c}}}^{t_{\mathrm{d}}} \frac{\widetilde{x}^{(1)}(\tau - t_{\mathrm{a}})}{(t-\tau)^{\alpha}} \mathrm{d}\tau, \tag{7}$$

which in terms of (5) assumes the form:

$$ {}^{\mathrm{C}}_{0}D^{\alpha}_{\Delta t_{\mathrm{loc}}}\widetilde{x}(\Delta T) = \frac{1}{\Gamma(1-\alpha)} \int_{0}^{\Delta t_{\mathrm{loc}}} \frac{\widetilde{x}^{(1)}(\tau_{\mathrm{loc}})}{(\Delta T-\tau_{\mathrm{loc}})^{\alpha}} \mathrm{d}\tau_{\mathrm{loc}}, \tag{8}$$

where the following auxiliary notations are used:

$$\tau_{\mathrm{loc}} = \tau - t_{\mathrm{c}}, \tag{9}$$

$$\Delta t_{\mathrm{loc}} = t_{\mathrm{d}} - t_{\mathrm{c}}, \tag{10}$$

$$\Delta T = t - t_{\mathrm{c}}. \tag{11}$$

Suppose that computations are performed throughout $t \in [0, t_{\mathrm{end}}]$ concerning a fractional time derivative of the variable x. At each time instance the integral (1) is dependent on all the previous states. This dependency varies subject to t being included in the denominator. If one wants to apply the polynomial (2) to approximate the fractional derivative then the definition (8) is not enough as it only applies for the local subinterval $[t_{\mathrm{c}}, t_{\mathrm{d}}]$, which because of the polynomial boundaries – should be contained inside $[t_{\mathrm{a}}, t_{\mathrm{b}}]$.

Let the interval $[0, t_{\mathrm{end}}]$ be divided into S subintervals $[t_{s,1}, t_{s,m_s}]$ where $s = 1, 2, \ldots, S$ and for $s > 1 : t_{s-1,m_s} = t_{s,1}$; m_s defines the number of nodes on the time axis in an interval with the unique index s. Subject to integral properties the fractional time derivative can be then expressed by means of the sum:

$$\begin{aligned} {}^{\mathrm{C}}_{0}D^{\alpha}_{t}x(t) &= \sum_{s=1}^{S} {}^{\mathrm{C}}_{t_{s,1}}D^{\alpha}_{t_{s,m_s}}x(t) \\ &\approx \sum_{s=1}^{S} {}^{\mathrm{C}}_{t_{s,1}}D^{\alpha}_{t_{s,m_s}}\widetilde{x}_s(t - t_{s,1}), \end{aligned} \tag{12}$$

where $\widetilde{x}_s$ is the local polynomial approximation in the subinterval $\Theta_s \subseteq [t_{s,1}, t_{s,m_s}]$. In fact, in most cases it will be true that $\Theta_s = [t_{s,1}, t_{s,m_s}]$, however Sec. 5 covers the cases where it is not. To simplify the description, the following notation is used:

$$\mathrm{d}^{\alpha}_{s}x(t) = {}^{\mathrm{C}}_{t_{s,1}}D^{\alpha}_{t_{s,m_s}}\widetilde{x}_s(t - t_{s,1}). \tag{13}$$

Equation (12) clearly shows that the fractional time derivative can be computed by means of a sum of fractional derivatives defined along the contiguous subintervals, hence at this stage it is worthwhile to derive a simpler form of (8) that is more suitable for further computations. For a single term of the polynomial (6) the fractional derivative along a subinterval $[t_{\mathrm{c}}, t_{\mathrm{d}}]$ is:

$$\begin{aligned} &{}^{\mathrm{C}}_{0}D^{\alpha}_{\Delta t_{\mathrm{loc}}}\left(a_k t^k_{\mathrm{loc}}\right) \\ &\quad = \frac{a_k}{\Gamma(1-\alpha)} \int_{0}^{\Delta t_{\mathrm{loc}}} \frac{k\tau^{k-1}_{\mathrm{loc}}}{(\Delta T-\tau_{\mathrm{loc}})^{\alpha}} \mathrm{d}\tau_{\mathrm{loc}}, \end{aligned} \tag{14}$$

which when defining the auxiliary value $\theta = \frac{\tau_{\mathrm{loc}}}{\Delta T}$ takes the form:

$$ {}^{\mathrm{C}}_{0}D^{\alpha}_{\Delta t_{\mathrm{loc}}}\left(a_k t^k_{\mathrm{loc}}\right) = \frac{a_k k \Delta T^{k-\alpha}}{\Gamma(1-\alpha)} \int_{0}^{\frac{\Delta t_{\mathrm{loc}}}{\Delta T}} \frac{\theta^{k-1}}{(1-\theta)^{\alpha}} \mathrm{d}\theta. \tag{15}$$

The integral above is actually the incomplete beta function:

$$B_{\frac{\Delta t_{\mathrm{loc}}}{\Delta T}}(k, \alpha - 1) = \int_{0}^{\frac{\Delta t_{\mathrm{loc}}}{\Delta T}} \frac{\theta^{k-1}}{(1-\theta)^{\alpha}} \mathrm{d}\theta. \tag{16}$$

Because k are positive integers – the above can be brought to the analytical expression [11]:

$$\begin{aligned} B_{\frac{\Delta t_{\mathrm{loc}}}{\Delta T}}(k, \alpha - 1) &= \frac{\Gamma(k)\Gamma(\alpha-1)}{\Gamma(k+\alpha-1)} \\ &\cdot \left(1 - \left(1 - \frac{\Delta t_{\mathrm{loc}}}{\Delta T}\right)^{\alpha-1} \sum_{j=0}^{k-1} \frac{\left(\frac{\Delta t_{\mathrm{loc}}}{\Delta T}\right)^{j} (\alpha-1)_j}{j!} \right), \end{aligned} \tag{17}$$

where the Pochhammer function is used:

$$(\alpha - 1)_j = \begin{cases} (\alpha-1)(\alpha-2)\ldots(\alpha-j) & j \neq 0, \\ 1 & j = 0. \end{cases} \tag{18}$$

If $\dfrac{\Delta t_{\mathrm{loc}}}{\Delta T} = 1$, then the integration in (15) is simplified and leads to [12]:

$$ {}^{\mathrm{C}}_{0}D^{\alpha}_{\Delta T}\left(a_k t^k_{\mathrm{loc}}\right) = a_k \frac{k\Delta T^{k-\alpha}\Gamma(k)}{\Gamma(1+k-\alpha)}. \tag{19}$$

This simplification can be applied for the interval expressed by the unique index $s = S$.

3. Fractional state equations

In this subsection formula (12) is applied so that a general system of linear fractional order state equations can be solved by means of a matrix equation for each selected time instance.

Suppose that one wants to obtain an implicit scheme where only the variable values at the computed time instance $t = t_{\mathrm{now}}$ are not known. In such a case in all subintervals the coefficients of the polynomial (6) are known except in the one defined by the index $s = S$. There it is only possible to obtain the coefficients of the Lagrange basis polynomials (4). Hence $\mathrm{d}_s^\alpha x(t)|_{t=t_\text{now}}$ are known values for $s = 1, 2, \ldots, S-1$ but $\mathrm{d}_S^\alpha x(t)|_{t=t_\text{now}}$ needs to be considered separately at first. However, the components of (2) can be divided into those dependent on $x|_{t=t_\text{now}}$ and those that are dependent on the variable values at previously selected time instances. So the polynomial in subinterval Θ_S is expressed by:

$$\begin{aligned}\widetilde{x}_S(t_{\text{loc }S}) &= \sum_{k=0}^{m_S-1} a'_k t_{\text{loc }S}^k \\ &+ \left(\sum_{k=0}^{m_S-1} b_{m_S,k} t_{\text{loc }S}^k\right) x|_{t=t_\text{now}},\end{aligned} \tag{20}$$

where

$$t_{\text{loc }S} = t - t_{S,1}. \tag{21}$$

For a circuit with fractional capacitors defined by the relation:

$$C_\alpha \frac{\mathrm{d}^\alpha u_C}{\mathrm{d}t^\alpha} = i_C \tag{22}$$

and fractional inductors given by:

$$L_\beta \frac{\mathrm{d}^\beta i_L}{\mathrm{d}t^\beta} = u_L, \tag{23}$$

a system of state equations is formulated in the following general form:

$$\begin{aligned}&\left[\frac{\mathrm{d}^{\alpha_1}}{\mathrm{d}t^{\alpha_1}} x_1, \quad \frac{\mathrm{d}^{\alpha_2}}{\mathrm{d}t^{\alpha_2}} x_2, \quad \ldots, \quad \frac{\mathrm{d}^{\alpha_n}}{\mathrm{d}t^{\alpha_n}} x_n\right]^\mathrm{T} \\ &= \boldsymbol{Ax} + \boldsymbol{Bv},\end{aligned} \tag{24}$$

where $\boldsymbol{x}$ denotes the vector of state variables $x_1, x_2 \ldots, x_n$, $\boldsymbol{v} = \boldsymbol{v}(t)$ is the vector of autonomous source values, $\boldsymbol{A}$ is the state matrix and $\boldsymbol{B}$ denotes a dependency on the respective sources. Subject to matrix multiplication the following formula is obtained for a single equation of (24):

$$\frac{\mathrm{d}^{\alpha_i}}{\mathrm{d}t^{\alpha_i}} x_i = \sum_{k=1}^{n} A_{i,k} x_k + \sum_{k=1}^{n_v} B_{i,k} v_k, \tag{25}$$

where n_v denotes the number of sources. Subject to the approximation (12), the above equation yields:

$$-\mathrm{d}_S^{\alpha_i} x_i + \sum_{k=1}^{n} A_{i,k} x_k = -\sum_{k=1}^{n_B} B_{i,k} v_k + \sum_{s=1}^{S-1} \mathrm{d}_s^{\alpha_i} x_i, \tag{26}$$

where the right-hand side values can be obtained at the beginning of each time-step as they are dependent on either the source values (which are known) or values of the state variables at $t < t_\text{now}$. Equations of the form (26) together build the matrix equation:

$$\boldsymbol{Gx} = \boldsymbol{g} - \boldsymbol{Bv}, \tag{27}$$

which can be solved by a chosen numerical method.

4. Time-step adaptation

In order to apply the method one needs to define the nodes on the time variable axis, which are going to be considered in the polynomial interpolation. It is easier to obtain the coefficients $b_{i,k}$ if these are equidistant nodes i.e. the timestep is constant In fact it is possible to give exact formulae for such a case (e.g. this has been done for other methods in [9]). However, time-step adaptation techniques are known to improve the quality of numerical methods for systems of differential equations in terms of both accuracy and computation time [13].

In this subsection a simple idea of a time-adaptive predictor-corrector scheme is presented:

a) first, for a selected time instance the polynomial order $p \geq 2$ is chosen;
b) in the predictor step the equation (27) is formulated and solved with the polynomial at Θ_S of order $p_o = p - 1$;
c) in the corrector step the equation (27) is solved for a polynomial of order $p_o = p$ (the polynomials in the rest of the subintervals do not change their order);
d) an error is computed;
e) if the error is greater than a given maximum value e_max then the time-step value Δt is modified and the computations are repeated for a changed t; else the time-step value is added and computations are performed for the next time instance.

Taking into account a temporary assumption that the predictor and corrector give respectively the approximated and exact solution, one can write the estimate of the local error for the state variable x as:

$$e = \frac{|x_\text{p} - x_\text{c}|}{w}. \tag{28}$$

x_p is the value of the state variable computed in the predictor step, while x_c is the value of x obtained in the corrector step; w is an arbitrarily chosen weight value, which could be given e.g. by a typical positive value for the considered state variable.

The following coefficient is used as a multiplier for the modification of the time-step:

$$\eta = \sqrt[p+1]{\frac{e_\text{ctrl}}{e_{\boldsymbol{x}}}}, \tag{29}$$

where $e_{\boldsymbol{x}}$ is the maximum taken from all error values computed by (28) for a current time instance and e_ctrl is an assigned error tolerance. Note that the above formula is used in adaptive time-stepping of methods used in integer order differential equations [13]. It has been chosen empirically for fractional order equations. A computation of the coefficient η basing on a local discretization error analysis [14] is not discussed in this paper.

5. Algorithm of subinterval dynamics

In this section it is explained how the subintervals Θ_s are being defined and modified in each time instance. In order to distinguish the subintervals from the integral boundaries, the latter are denoted by $\Xi_s = [t_{s,1}, t_{s,m_s}]$.

Firstly, in order to minimize the amount of numerical computation being performed at each time-step – it is established that only two of the polynomial subintervals are modified at most – specifically Θ_{S-1} and Θ_S. Secondly, only Θ_S will be able to change both its boundaries at every time-step.

There are four types of intervals included in the algorithm:

- sealed intervals – in these intervals it is always true that $\Theta_s = \Xi_s$; sealed intervals are not changed and do not change their type,
- closed intervals – these intervals do not change, however Ξ_s can take into account a smaller interval for integration; only Θ_{S-1} can be this type of interval,
- built intervals – also only reserved for Θ_{S-1}; they change their size by taking into account time nodes after their respective state vector had been obtained,
- the moving interval – this is always Θ_S, it changes its size to $N-1$ for the predictor step and N for the corrector step, where $N = \min(p+1, i)$ with i being the number of the current time step (where the first time step is filled with initial values); in the case of this interval it is always true that $\Theta_S = \Xi_S$.

The algorithm of how the subintervals are arranged in a single time-step can be presented in a couple of basic actions as depicted in Fig. 1.

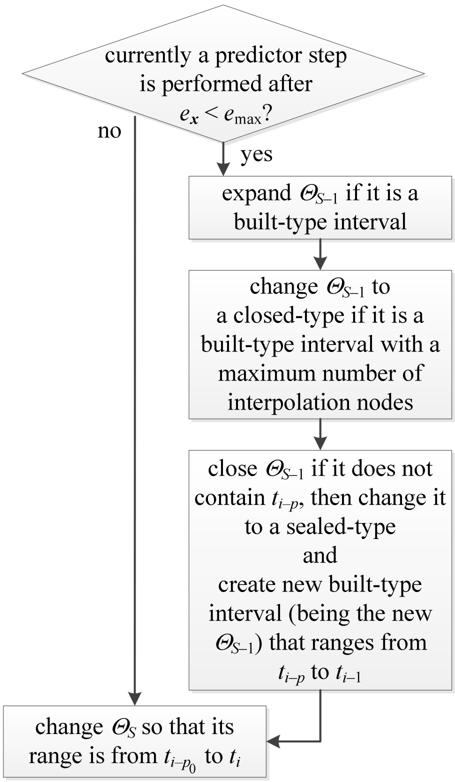

currently a predictor step is performed after $e_x < e_{\max}$?
no
yes
expand Θ_{S-1} if it is a built-type interval
change Θ_{S-1} to a closed-type if it is a built-type interval with a maximum number of interpolation nodes
close Θ_{S-1} if it does not contain t_{i-p}, then change it to a sealed-type and create new built-type interval (being the new Θ_{S-1}) that ranges from t_{i-p} to t_{i-1}
change Θ_S so that its range is from t_{i-p_0} to t_i

Fig. 1. General scheme of the subinterval dynamics algorithm

An exemplary outline of how the intervals change in the subsequent time-steps is displayed in Fig. 2. The exemplary case shown in Fig. 2 is for subsequent successful predictor-corrector steps (i.e. when $e_{\boldsymbol{x}} < e_{\max}$).

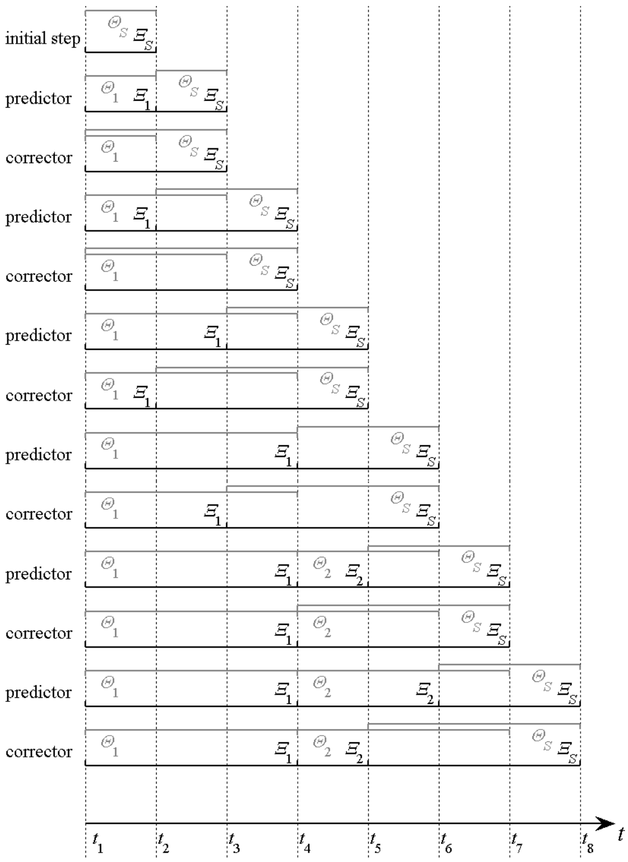


Fig. 2. Subinterval dynamics in the case of $p = 3$

6. General steps of the method

Generally the proposed method can be summarized by the following list of consecutive procedures executed in each time-step:

a) the Lagrange polynomial interpolation (3) is performed, which results in the (4) form and the coefficients $b_{i,k}$ are obtained,

b) for the polynomials at Θ_s $(s = 1, 2, \ldots, S-1)$ the a_k coefficients of (6) are obtained,

c) for Θ_S the coefficients a'_k and $b_{m,Sk}$ of Eq. (20) are computed,

d) for each interval Ξ_s the integrations are performed subject to the analytical formulae given in Sec. 2 (i.e. (15)–(19)),

e) the subinterval dynamics are performed subject to the algorithm given in Sec. 5,

f) the fractional state Eqs. (24) are computed by means of a reduction to the matrix Eq. (27) through the formula (26),

g) if currently the corrector step is computed then an error analysis is performed that determines the next step, where either:

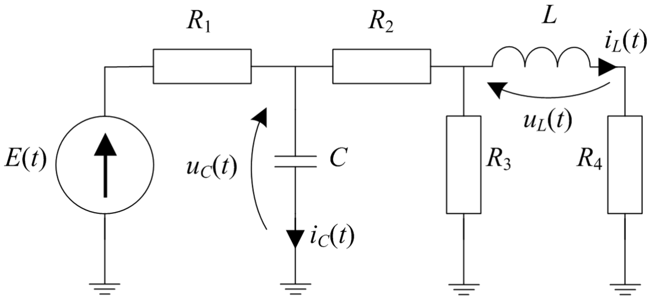
  - the time-step is repeated for a changed Δt (multiplied by η),
  - the next time step is computed with a Δt changed according to η).

7. Examples

This section presents exemplary problems and their numerical solutions obtained by means of the discussed method. In Subsec. 7.2 the letter s, unlike in previous chapters, denotes the complex argument resulting from the Laplace transform.

7.1. AC analysis for circuits with fractional order elements. For an AC analysis the circuit can be solved by a traditional approach using complex numbers. The subinterval-based method will treat the problem in the same way as it would a transient one and (if working properly) after a sufficient amount of periods – should assume the same result as the AC solution The above fact is used in this subsection to check the correctness of the proposed numerical method.

An exemplary circuit that has been considered is presented in Fig. 3.



$E(t) = \sin(\omega t)$, $\omega = 2\pi f$, $f = 50\,\text{Hz}$
$R_1 = 2\,\Omega$, $R_2 = 3\,\Omega$, $R_3 = 3\,\Omega$, $R_4 = 4\,\Omega$
$C = 2{\cdot}10^{-2}\ \text{F}{\cdot}\text{s}^{\alpha-1}$, $\alpha = 0.6$, $L = 0.1\ \text{H}{\cdot}\text{s}^{\beta-1}$, $\beta = 0.7$

Fig. 3. Exemplary electric circuit with fractional order elements and an AC voltage source

The fractional capacitor voltage and inductor current are computed. The state equations for the studied circuit are:

$$\begin{bmatrix}\dfrac{\mathrm{d}^\alpha u_C}{\mathrm{d}t^\alpha} & \dfrac{\mathrm{d}^\beta i_L}{\mathrm{d}t^\beta}\end{bmatrix}^T = \boldsymbol{A}\begin{bmatrix}u_C\\ i_L\end{bmatrix} + \begin{bmatrix}\dfrac{1}{CR_1}\\ 0\end{bmatrix}E(t), \quad (30)$$

where

$$\boldsymbol{A} = \begin{bmatrix}\dfrac{-(R_1+R_2+R_3)}{CR_1(R_2+R_3)} & \dfrac{-R_3}{C(R_2+R_3)}\\[2ex] \dfrac{R_3}{L(R_2+R_3)} & \dfrac{-\left(R_3R_4+R_2(R_3+R_4)\right)}{L(R_2+R_3)}\end{bmatrix}, \quad (31)$$

is the system matrix.

For an AC approach Eq. (30) yields:

$$\begin{bmatrix}\underline{u}_C\\ \underline{i}_L\end{bmatrix} = \left(\mathrm{diag}\left(\omega^\alpha e^{j\frac{\pi}{2}\alpha}, \omega^\beta e^{j\frac{\pi}{2}\beta}\right) - \mathbf{A}\right)^{-1}\begin{bmatrix}\dfrac{1}{CR_1}\\ 0\end{bmatrix}. \quad (32)$$

The comparison between the transient solution (obtained with the subinterval-based method) and the AC solution are presented as time dependent plots in Fig. 4.

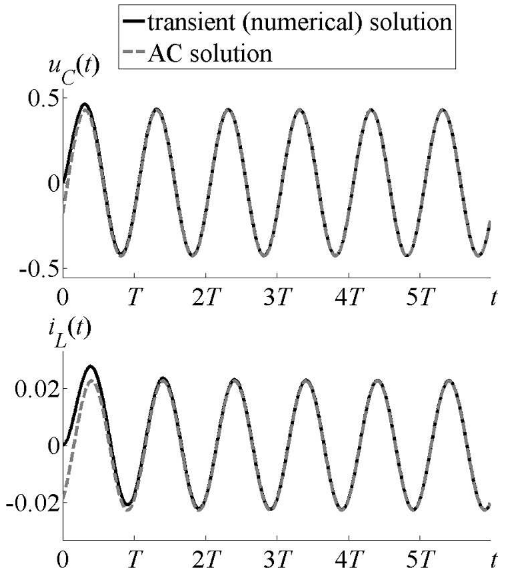
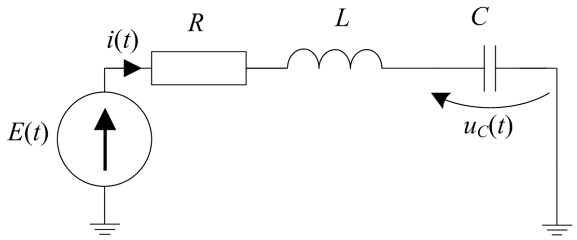


Fig. 4. Comparison of the AC analysis result and the transient numerical solution for six periods of the sinusoidal voltage source $E(t)$

A very good resemblance of the steady-state can be observed in the result. For the subinterval method the polynomial order $p = 4$ has been chosen and the designated maximum error value $e_{\max} = 0.1\%$.

7.2. Unit step response of a fractional RLC circuit. In order to observe a comparison with a transient analytical solution, a simple problem concerning a unit-step voltage source supplying a series RLC circuit is considered (Fig. 5).



$E(t) = \mathbf{1}(t)$, $R = 2\,\Omega$,
$C = 10^{-2}\ \text{F}{\cdot}\text{s}^{\alpha-1}$, $L = 5{\cdot}10^{-2}\ \text{H}{\cdot}\text{s}^{\alpha-1}$, $\alpha = 0.7$

Fig. 5. Exemplary transient problem of an electric circuit supplied by a unit-step source

When applying the Laplace transform, the current is determined by the equation:

$$I(s) = \frac{1}{s\left(R + s^\alpha L + s^{-\alpha}\frac{1}{C}\right)}, \quad (33)$$

while the capacitor voltage in the s-domain is:

$$U_C(s) = \frac{1}{s(CRs^\alpha + s^{2\alpha}CL + 1)}. \quad (34)$$

The above s-domain formulae yield respectively the following analytical results (according to equations given in [15]):

$$\mathcal{L}^{-1}\left(I(s)\right) = i(t) =$$
$$= \frac{1}{L(\lambda_1 - \lambda_2)} t^\alpha \left(\lambda_1 E_{\alpha,\alpha+1}\left(\lambda_1 t^\alpha\right) - \lambda_2 E_{\alpha,\alpha+1}\left(\lambda_2 t^\alpha\right)\right) \quad (35)$$

and

$$L^{-1}\left(U_C(s)\right) = u_C(t) = \\ = \frac{1}{2LC\sqrt{\Delta}} t^{\alpha}\left(E_{\alpha,\alpha+1}\left(\lambda_1 t^{\alpha}\right) - E_{\alpha,\alpha+1}\left(\lambda_2 t^{\alpha}\right)\right), \tag{36}$$

where E is the Mittag-Leffler function. Δ, λ_1 and λ_2 are the following auxiliary values:

$$\Delta = -\frac{R^2}{4L^2} - \frac{1}{LC}, \tag{37}$$

$$\lambda_1 = -\frac{R^2}{4L^2} + \sqrt{\Delta}, \qquad \lambda_2 = -\frac{R^2}{4L^2} - \sqrt{\Delta}. \tag{38}$$

A comparison of the results obtained by the Mittag Leffler function in (35), (36) and by means of the subinterval-based method is presented in Fig. 6.

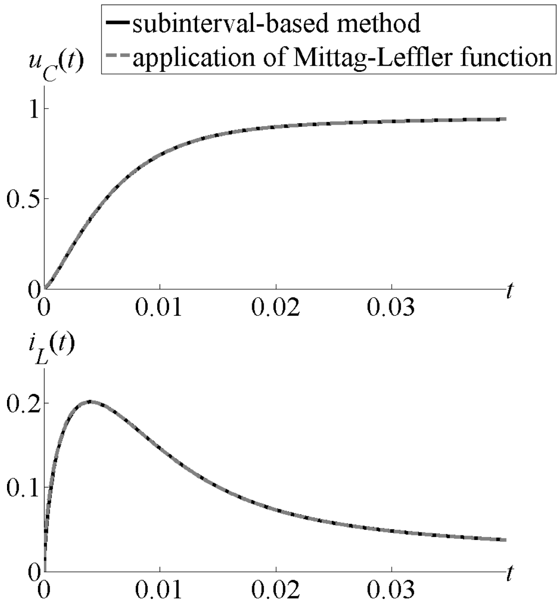


Fig. 6. Comparison of the numerical results for the transient problem

The same parameters for the subinterval-based method have been used as for the example in Subsec. 6.1. Once more the result obtained with the proposed method exhibits a very close similarity to the analytical solution.

8. Summary

A method based on a polynomial approximation and a subinterval dynamics algorithm has been presented.

A general advantage of the presented method is the subinterval dynamics algorithm, which allows to separate the fractional derivative computations. Additionally, as has been shown in Sec. 2 – it is possible to derive analytical formulae for the Caputo fractional derivative for polynomials of arbitrary order defined on the subintervals. The adaptive time-stepping procedure additionally can influence the accuracy of the result especially in cases when it is difficult to initially ascertain the variable changes in time.

In order to ascertain the applicability of the subinterval-based method – a comparison has been performed for the solutions obtained by this method with those obtained by traditional methods. This has been done for an AC problem and one that deals with a transient fractional order circuit response.

In both cases the solution obtained by the subinterval-based method closely reflects the referential method, which proves its proper functioning and its usefulness.

REFERENCES

[1] J.T. Machado, V. Kiryakova, and F. Mainardi, "Recent history of fractional calculus" *Commun. Nonlinear Sci. Numer. Simul.* 16, 1140–1153 (2011).

[2] C.P. Li, Y.Q. Chen, and J. Kurths, "Fractional calculus and its applications", *Phil. Trans. R. Soc.* A 2013, 371 (2013).

[3] W. Mitkowski and P. Skruch, "Fractional-order models of the supercapacitors in the form of RC ladder networks", *Bull. Pol. Ac.: Tech.* 61 (3), 580–587 (2013).

[4] T.J. Freeborn, B. Maundy, and A.S. Elwakil, "Measurement of supercapacitor fractional-order model parameters from voltage-excited step response", *IEEE J. on Emerging and Selected Topics in Circuits and Systems* 3 (3), 367–376 (2013).

[5] I. Schäfer and K. Krüger, "Modelling of lossy coils using fractional derivatives", *J. Phys. D: Appl. Phys.* 41, 1–8 (2008).

[6] J. Walczak and A. Jakubowska, "The analysis of resonance phenomena in a series RLC circuit with a supercapacitor", *PAK* 59 (10), 1105–1108 (2013), (in Polish).

[7] T. Kaczorek, "Positive linear systems with different fractional orders", *Bull. Pol. Ac.: Tech.* 58 (3), 453–458 (2010).

[8] T. Kaczorek "Stability of positive fractional switched continuous-time linear systems", *Bull. Pol. Ac.: Tech.* 61 (2), 349–352 (2013).

[9] M. Weilbeer, "Efficient numerical methods for fractional differential equations and their analytical background", *PHD Dissertation* pp. 105–159, The Carl-Friedrich-Gauß School Faculty for Mathematics and Computer Science, Braunschweig University of Technology, Braunschweig, 2005.

[10] M. Caputo, "Linear model of dissipation whose Q is almost frequency independent – II", *The Geophysical J. Royal Astronomical Society* 13, 529–539 (1967).

[11] http://functions.wolfram.com/GammaBetaErf/Beta3/03/01/02/0003/.

[12] M. Ishteva, L. Boyadjiev, and R. Scherer, "On the Caputo operator of fractional calculus and C-Laguerre functions", *Mathematical Sciences Research J.* 9, 161–170 (2005).

[13] V. Dolejší and P. Kůs, "Adaptive backward difference formula-Discontinuous Galerkin finite element method for the solution of conservation laws", *Int. J. Numerical Methods in Engineering* 73, 1739–1766 (2008).

[14] E. Hairer, S.P. Nørsett, and G. Wanner, "Solving ordinary differential equations I: Nonstiff problems", *Springer Series in Computational Mathematics* 8, 368–374 (1993).

[15] M. Włodarczyk and A. Zawadzki, "RLC circuits in aspect of positive fractional derivatives", *Scientific Works of the Silesian University of Technology Q. Electrical Engineering* 1 (217), 75–88, (2011), (in Polish).