Krzysztof OPRZĘDKIEWICZ[1]

AGH University

doi:10.15199/48.2023.07.22

Numerical properties of discrete approximations of an elementary fractional order transfer function

Abstract. *The paper deals with the analysis of basic numerical properties of discrete approximations of the elementary fractional order, inertial transfer function. The considered transfer function is approximated with the use of two most typical approaches. The first one uses Continuous Fraction Expansion (CFE) approximation, the next one employes the Fractional Order Backward Difference (FOBD) approximation, based on the Grünwald-Letnikov (GL) definition of fractional operator. Elementary properties of both approximants: accuracy and duration of calculations are numerically analysed using PC and MATLAB. Publications in this field are not known to the author.*
Results of numerical tests point that at the considered software-hardware platform the FOBD approximation assures better accuracy than CFE approximation with practically the same duration of computation. Next, the speed of computing is determined by the form of source code. Additionally, the computing of step response with the use of both tested approximations is much faster than the use of analytical solution employing the MATLAB implementation of Mittag-Leffler function.

Streszczenie. *W artykule omówiono podstawowe własności numeryczne dyskretnych aproksymacji elementarnej transmitancji obiektu inercyjnego niecałkowitego rzędu. Do aproksymacji zastosowano dwie najbardziej typowe metody. Pierwsza z nich bazuje na aproksymacji CFE niskiego rzędu, druga aproksymacja (FOBD) wykorzystuje definicję operatora ułamkowego podaną przez Grüunwalda i Letnikova. Podstawowe własności obu aproksymacji: dokładność i złożoność (w sensie czasu obliczeń) zostały poddane analizie numerycznej z użyciem środowiska MATLAB na typowej platformie PC. Wcześniejsze publikacje w literaturze z tego zakresu nie są znane autorowi.*
Na podstawie wyników testów numerycznych można stwierdzić, że w rozważanym wypadku zastosowanie aproksymacji FOBD zapewnia lepszą dokładność przy praktycznie tej samej szybkości obliczeń. Stwierdzono też zależność pomiędzy szybkością obliczeń i postacią kodu źródłowego programu. Dodatkowo zauważono, że zastosowanie każdej z omawianych aproksymacji pozwala na wykonanie obliczeń znacznie szybciej, niż wykorzystanie w tym celu analitycznej formuły na odpowiedź skokową rozważanej transmitancji. Wynika to prawdopodobnie z długiego czasu wyznaczania funkcji Mittag-Lefflera na platformie MATLAB. ***(Właściwości numeryczne dyskretnych aproksymacji elementarnej transmitancji ułamkowego rzędu)***

Keywords: fractional order transfer function, CFE approximation, FOBD approximation, numerical complexity
Słowa kluczowe: transmitancja ułamkowa, aproksymacja CFE, aproksymacja FOBD, złożoność numeryczna

Introduction

The fractional order calculus is a convenient tool to describe many complex physical phenomena. Non-integer models have been presented by many Authors, e.g. by [3], [5], [7], [8],[20], [21].

FO transfer function is one of most typical models applied to modeling of different processes and phenomena. A transfer function can have simple or complex form (see e.g. [6], [11], [13]). In most situations computations done with the use of FO transfer functions require to use approximations of basic FO element s^{α} due to an analytical formulae can be obtained only for single, particular situations.

The "fractional" analogue of the 1'st order inertial transfer function $\frac{1}{Ts^{\alpha}+1}$ is a typical tool used in FO modeling. It is applied both in stand-alone form and as a component of more complex models. Its at each digital platform (PLC, microcontroller) requires to use of its discrete, finite dimensional approximation. To do it two alternative approaches are employed.

The first one directly derives from definition of fractional operator given by Grünwald and Letnikov. It is called Fractional Order Backward Difference (FOBD) see(e.g. [16]). Its use requires to apply of high order FIR filter, containing only zeros. The order of this approximation should be relatively high to assure a good accuracy of approximation.

An alternative way is the use of the Continuous Fraction Expansion (CFE) approximator (see e.g. [4], [22]). It has the form of the IIR filter containing both poles and zeros. It is faster covergent and easier to implement due to its relatively low order, typically not higher than 5.

Important problem during use of approximated form of each model there is its accuracy and numerical complexity, estimated for example by duration of computations. It is worth to note that these properties are relatively rare discussed for FO models. The basic element s^{α} implemented at PLC platform is presented in [12], the accuracy of the discrete state-space model of heat transfer process is analysed in [14].

This paper is devoted to analyze of main numerical properties of the elementary FO transfer function $\frac{1}{Ts^{\alpha}+1}$ expressed with the use of FOBD and CFE approximators. The accuracy analysis uses the analytical formula of the step response, the duration of computations is done for the MATLAB platform. The most time consuming parts of source code are detected and modified to accelerate of computations.

The paper is organized as follows. Preliminaries recall some elementary ideas from fractional calculus as well as both considered approximation methods. Next the considered transfer function, its step response and discrete approximations are presented. Finally numerical tests covering the accuracy and numerical complexity are presented and discussed.

Preliminaries

Basic ideas

Elementary ideas from fractional calculus can be found in many books, e.g. [5], [9], [16] or [20]. Here only some definitions necessary to present of results will be given.

Firstly recall the fractional-order, integro-differential operator (see for example [5], [10], [20]):

Definition 1 *(The elementary fractional order operator) The fractional-order integro-differential operator is defined as follows:*

$$ {}_{t_s}D^{\alpha}_{t_f} f(t) = \begin{cases} \frac{d^{\alpha} f(t)}{dt^{\alpha}} & \alpha > 0 \\ f(t) & \alpha = 0 \\ \int_{t_s}^{t_f} f(\tau)(d\tau)^{\alpha} & \alpha < 0 \end{cases} . \tag{1} $$

where t_s and t_f denote time limits for operator calculation, $\alpha \in \mathbb{R}$ denotes the non integer order of the operation.

Next recall an idea of Mittag-Leffler functions. The two parameter Mittag-Leffler function is defined as follows:

Definition 2 *(The two parameter Mittag-Leffler function)*

$$ E_{\alpha,\beta}(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(k\alpha + \beta)}. \tag{2} $$

For $\beta = 1$ we obtain the one parameter Mittag-Leffler function:

Definition 3 *The one parameter Mittag-Leffler function*

$$E_\alpha(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(k\alpha + 1)}. \tag{3}$$

The fractional-order, integro-differential operator (1) can be described by different definitions, given by Grünwald and Letnikov (GL Definition), Riemann and Liouville (RL Definition) and Caputo (C Definition). In this paper C and GL definitions will be employed. The C defintion takes the following form [10]:

Definition 4 *(The Caputo definition of the FO operator)*

$${}_0^C D_t^\alpha f(t) = \frac{1}{\Gamma(V-\alpha)} \int_0^\infty \frac{f^{(V)}(\tau)}{(t-\tau)^{\alpha+1-V}} d\tau. \tag{4}$$

In (4) V is an integer limiter of the non integer order: $V-1 \leq \alpha < V \in \mathbb{N}$. If $V = 1$ then consequently $0 \leq \alpha < 1$ is considered and the definition (4) takes the form:

$${}_0^C D_t^\alpha f(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^\infty \frac{\dot{f}(\tau)}{(t-\tau)^{\alpha}} d\tau. \tag{5}$$

For the Caputo operator the Laplace transform can be defined [9]:

Definition 5 *(The Laplace transform for the Caputo operator)*

$$\begin{aligned} &\mathcal{L}({}_0^C D_t^\alpha f(t)) = s^\alpha F(s), \quad \alpha < 0 \\ &\mathcal{L}({}_0^C D_t^\alpha f(t)) = s^\alpha F(s) - \sum_{k=0}^{v-1} s^{\alpha-k-1} {}_0 D_t^k f(0), \\ &\alpha > 0, \quad v-1 < \alpha \leq v \in \mathbb{N}. \end{aligned} \tag{6}$$

The GL definition is given e.g. in ([3] or [15]). It is as beneath:

Definition 6 *(The Grünwald-Letnikov definition of the FO operator)*

$${}_0^{GL} D_t^\alpha f(t) = \lim_{h \to 0} h^{-\alpha} \sum_{l=0}^{[\frac{t}{h}]} (-1)^l \binom{\alpha}{j} f(t - lh). \tag{7}$$

In (7) $\binom{\alpha}{l}$ is the binomial coefficient:

$$\binom{\alpha}{l} = \left\{ \begin{array}{ll} 1, & l = 0 \\ \frac{\alpha(\alpha-1)\ldots(\alpha-l+1)}{l!}, & l > 0 \end{array} \right\}. \tag{8}$$

Finally the fractional order transfer function can be defined. In general it can be expressed as follows:

$$G(s) = \frac{b_m s^{\beta_m} + \ldots + b_1 s^{\beta_1} + b_0}{a_n s^{\alpha_n} + \ldots + a_1 s^{\alpha_1} + a_0}. \tag{9}$$

where α and β are fractional orders (commensurate or not commensurate), a and b are coefficients.

In modeling elementary forms of the general transfer function (9) are applied. In this paper the following one is considered:

$$G(s) = \frac{1}{T_\alpha s^\alpha + 1}. \tag{10}$$

Table 1. Coefficients of CFE polynomials $CFE_{N,D}(z^{-1}, \alpha)$ for Tustin approximation [4]

Order M	w_m	v_m
M=1	$w_1 = -\alpha$	$v_1 = \alpha$
	$w_0 = 1$	$v_0 = 1$
M=3	$w_3 = -\frac{\alpha}{3}$	$v_3 = \frac{\alpha}{3}$
	$w_2 = \frac{\alpha^2}{3}$	$v_2 = \frac{\alpha^2}{3}$
	$w_1 = -\alpha$	$v_1 = \alpha$
	$w_0 = 1$	$v_0 = 1$
M=5	$w_5 = -\frac{\alpha}{5}$	$v_5 = \frac{\alpha}{5}$
	$w_4 = \frac{\alpha^2}{5}$	$v_4 = \frac{\alpha^2}{5}$
	$w_3 = -\left(\frac{\alpha}{5} + \frac{2\alpha^3}{35}\right)$	$v_3 = -\left(\frac{-\alpha}{5} + \frac{-2\alpha^3}{35}\right)$
	$w_2 = \frac{2\alpha^2}{5}$	$v_2 = \frac{2\alpha^2}{5}$
	$w_1 = -\alpha$	$v_1 = \alpha$
	$w_0 = 1$	$v_0 = 1$

The useful advantage of the elementary transfer function (10) is that the analytical formula of step response is known. It is as follows (see e.g. [3], p. 11):

$$y_{an}(t) = \left(1(t) - E_\alpha\left(-\frac{t}{T_\alpha}\right)\right). \tag{11}$$

In (11) $E_\alpha(..)$ is the one parameter Mittag-Leffler function (3). The formula (11) will be used as the reference to estimate of an accuracy of tested discrere approximations.

An implementation of operator (1) at each digital platform (PLC, microcontroller) requires to apply integer-order finite-length discrete-time approximators mentioned in the introduction. Both approximators are presented in the next subsections.

The CFE approximation

The basic FO element s^α can be approximated using the generating function $s \approx \omega(z^{-1})$. The new operator raised to the power α has the form of discrete transfer function of z^{-1} (see for example [4], [17] p.119):

$$\begin{aligned} &\left(\omega(z^{-1})\right)^\alpha = \left(\tfrac{1+a}{h}\right)^\alpha CFE\{\left(\tfrac{1-z^{-1}}{1+az^{-1}}\right)^\alpha\}_{M,M} = \\ &= \tfrac{P_{\alpha M}(z^{-1})}{Q_{\alpha M}(z^{-1})} = \left(\tfrac{1+a}{h}\right)^\alpha \tfrac{CFE_N(z^{-1},\alpha)}{CFE_D(z^{-1},\alpha)} = \\ &= \tfrac{\sum_{m=0}^{M} w_m z^{-m}}{\sum_{m=0}^{M} v_m z_{-m}} \quad \tfrac{}{D_{CFE}(z^{-1})} = N_{CFE(z-1)}. \end{aligned} \tag{12}$$

where h denotes the sampling time and M is the order of approximation. Numerical values of coefficients w_m and v_m and various values of the parameter a can be computed using e.g. the MATLAB function given in [19].

In Eqn. (12) a is the coefficient depending on an approximation type. For $a = 0$ and $a = 1$ we obtain the Euler and Tustin approximations, respectively. For $a \in (0, 1)$ we arrive at the Al-Alaoui-based approximation, which is a linear combination of the Euler and Tustin approaches. Note that in this case the parameter a in Eqn. (12) is equal to $a = \frac{1-\beta}{1+\beta}$, with β being the Al-Alaoui weighting coefficient (see [1, 22]). This parameter can be also employed in particular situations to improve an accuracy of an approximation.

If the Tustin approximation is considered (a=1) then $CFE_D(z^{-1}, \alpha) = CFE_N(z^{-1}, -\alpha)$ and the polynomial $CFE_D(z^{-1}, \alpha)$ can be given in the direct form (see [4]). Examples for the polynomial $CFE_D(z^{-1}, \alpha)$ for $M = 1, 3, 5$ are given in Table 1. The detailed analysis of various forms of the CFE approximators has been given in [22].

The Fractional Order Backward Difference

The GL definition (7) is the limit case for $h \to 0$ of the Fractional Order Backward Difference (FOBD), commonly em-

ployed in discrete FO calculations (see for example [16], p. 68).

Definition 7 *(The Fractional Order Backward Difference-FOBD)*

$$(\Delta^{\alpha}x)(t) = \frac{1}{h^{\alpha}}\sum_{l=0}^{L} d_l x(t-lh). \tag{13}$$

where:

$$d_l = (-1)^l \binom{\alpha}{l}. \tag{14}$$

Numerical values of coefficients d_l and various values of the parameters α and memory length L can be computed using e.g. the MATLAB function given in [18]. They can be also computed recursively using the following formula (see e.g. [3], p. 12):

$$d_0 = 1$$
$$d_l = \left(1 - \frac{1+\alpha}{l}\right) d_{l-1}, \quad l = 1, ..., L. \tag{15}$$

It is proven in [2] that:

$$\sum_{l=1}^{\infty} d_l = 1 - \alpha. \tag{16}$$

From (15) and (16) we obtain at once that:

$$\sum_{l=2}^{\infty} d_l = 1. \tag{17}$$

In (13) L denotes a memory length necessary to correct approximation of a non integer order operator. Unfortunately good accuacy of PSE approximation requires to use a long memory L what can make difficulties in implementation.
The z^{-1} transform of FOBD takes the following form:

$$\mathrm{Z}\{(\Delta^{\alpha}x)(kh)\} = \frac{1}{h^{\alpha}}\sum_{l=0}^{L} d_l z^{-l} X(k), \ k = 0, 1, 2, ... \tag{18}$$

In (18) k denotes discrete time instants. Consequently the discrete transfer function of basic element s^{α} using FOBD is as follows:

$$N_{FOBD}(z^{-1}) = \frac{1}{h^{\alpha}}\sum_{l=0}^{L} d_l z^{-l}. \tag{19}$$

The approximations of the considered transfer function

The use of (12) or (19) to transfer function (10) allows to obtain its finite dimensional, integer order approximations. The use of CFE approximation yields:

$$G_{CFE}(z^{-1}) = \frac{\sum_{m=0}^{M} v_m z^{-m}}{\sum_{m=0}^{M} (T_{\alpha} g_h w_m + v_m) z^{-m}}. \tag{20}$$

where v_m, w_m and g_h are coefficients of the CFE approximation (12) and M is the order of approximation. The step response of the approximated transfer function (20) is as follows:

$$y_{CFE}(k) = Z^{-1}\{\frac{1}{1-z^{-1}} G_{CFE}(z^{-1})\}. \tag{21}$$

Analogically, using (19) gives:

$$G_{FOBD}(z^{-1}) = \frac{1}{\frac{T_{\alpha}}{h}\sum_{l=0}^{L} d_l z^{-l} + 1}. \tag{22}$$

In (22) d_l are coefficients (14), (15), L is the memory length. Analogically the step response of the transfer function is:

$$y_{FOBD}(k) = Z^{-1}\{\frac{1}{1-z^{-1}} G_{FOBD}(z^{-1})\}. \tag{23}$$

Both step responses (21) and (23) can be computed using MATLAB function *step*.

Numerical tests

Accuracy

The accuracy of both considered approximations was tested using known Integral Absolute Error (IAE) cost function:

$$IAE = h\sum_{k=0}^{K} \left| y_{CFE,FOBD}(k) - y_{an}(k) \right|. \tag{24}$$

where $k = 0, ..., K$ is the index of time instant, h is the sample time, $y_{CFE,FOBD}$ and y_{an} are step responses approximated and analytical respectively. Tests of accuracy for both approximations are described by tables 2, 3 and illustrated by exepmplary figures 1-3.

Table 2. The cost function IAE for CFE approximation, order $M = 5$ and different fractional orders α and parameter a.

$a\downarrow$ $\alpha\rightarrow$	0.25	0.50	0.75
0.00	0.5466	1.1009	1.1702
0.78	0.9349	1.9843	2.3270
1.00	1.0188	2.1794	2.6100

Table 3. The cost function IAE for FOBD approximation, different orders α and memory lengths L.

$L\downarrow$ $\alpha\rightarrow$	0.25	0.50	0.75
100	0.1217	0.1512	0.1175
150	0.0353	0.0444	0.0520
200	0.0227	0.0355	0.0486
250	0.0227	0.0355	0.0486
300	0.0227	0.0355	0.0486

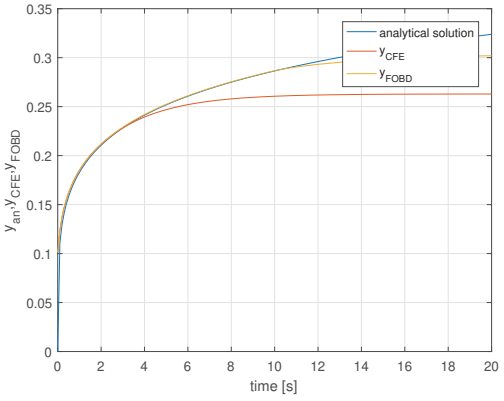

Fig. 1. Analytical step response vs CFE and FOBD approximations for $\alpha = 0.25$, $T = 5[s]$, $a = 0$, $L = 100$.

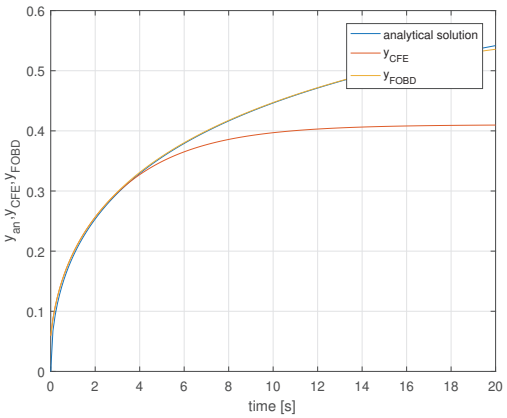

Fig. 2. Analytical step response vs CFE and FOBD approximations for $\alpha = 0.50$, $T = 5[s]$, $a = 0$, $L = 150$.

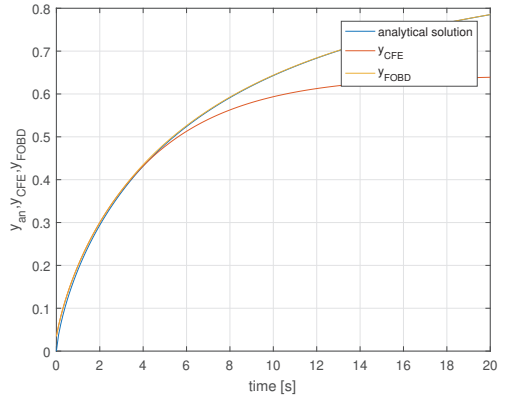

Fig. 3. Analytical step response vs CFE and FOBD approximations for $\alpha = 0.75$, $T = 5[s]$, $a = 0$, $L = 200$.

From tables 2, 3 and figures 1-3 it can be concluded that the use of the FOBD approximation in each situation assures much better accuracy than the use of CFE method.
However an application of FOBD method requires to apply a high order transfer function. This makes the duration of calculations more longer than use of CFE.

Duration of computations

The parameters of the PC computer employed to experiments are collected in the table 4. The duration of computations with the use of both approximations and analytical solution at MATLAB was investigated using MATLAB *tic* and *toc* functions. The duration of computing of approximated step responses (21), (23) and analytical solution (11) were done for: $\alpha = 0.5$, $T_\alpha = 5[s]$, $M = 5$, and different values of coefficient a (CFE) and order L (FOBD). The sigle step response (21), (23) and (11) was tested using the following sequences of instructions:

```
%Approximation CFE:
start_CFE=tic;
M=5; %order CFE
a=1; %0.0 - Euler, 0.78-Al Aloui, 1.0 - Tustin
sysCFE=dfod1(M,h,a,alfa); %_KOp
sysFO1=1/(Talfa*sysCFE+1);
yCFE=step(sysFO1,t);
```

Table 4. The parameters of the PC computer used to experiments.

Parameter	
Processor	Intel(R) Core(TM) i5-8600K CPU @ 3.60GHz
RAM	16 GB
OS	Windows 10 Pro
MATLAB	R2016b

Table 5. The mean of execution time for CFE approximation, order $M = 5$, $\alpha = 0.5$ and different values of parameter a.

a	0.00	0.78	1.00
Duration [s]	0.0130	0.0128	0.0129

Table 6. The mean of execution time for FOBD approximation, $\alpha = 0.5$ and different values of order L.

L	100	150	200	250	300
Duration [s]	0.0124	0.0122	0.0123	0.0122	0.0122

```
t_CFE=toc(start_CFE);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Approximation FOBD:
start_PSE=tic;
L=200;
sysPSE=dfod2(L,h,alfa);
sysFO2=1/(Talfa*sysPSE+1);
yPSE=step(sysFO2,t);
t_PSE=toc(start_PSE);
%%%%%%%%%%%%%%%
%Analytical formula of the step response:
start_an=tic;
y_an=(1-ml(((-t.^alfa)/Talfa),alfa));
t_an=toc(start_an);
```

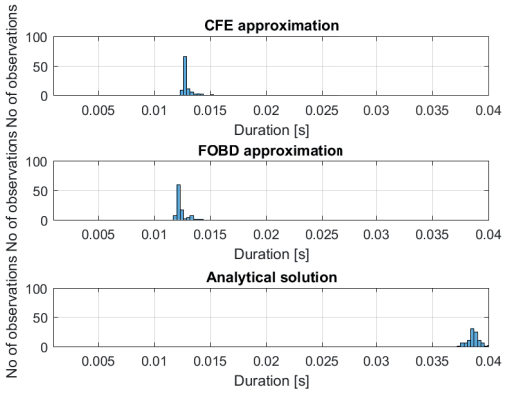

Fig. 4. Histograms of duration of computing of step response using CFE, FOBD and analytical formula for: $\alpha = 0.5$, $T_\alpha = 5[s]$, $a = 1.0$ and $L = 200$.

As the experiment 100 tests were executed. Results are illustrated by histograms 4, mean values of computations for different values of a coefficient and order L are described by tables 5 and 6.
They allow to formulate little bit surprising conclusions. The duration of calculations using both approximations is practically the same, but more longer is computing of the analytical response (11). Next, in the case of FOBD the duration practically does not depend on memory length L. This requires to make more detailed tests of complexity. Such tests as well as suitable modifications of the source code are presented in the next subsection.

Improving of the execution times

Detailed tests of numerical complexity can be executed using MATLAB function *profile*. Its use makes computations much more longer, but it detects most time consuming lines of a program code. In our case the use of *profile* is as beneath:

```
profile clear;
profile on;
M=5; %order CFE
a=.0; %0 - Euler, .78-Al Aloui,
% 1 - Tustin
sysCFE=dfod1(M,h,a,alfa); %_KOp
sysFO1=1/(Talfa*sysCFE+1);
yCFE=step(sysFO1,t);
```

```
profile viewer;
  profile clear;
profile clear;
 profile on;
 L=200;
sysPSE=dfod2(L,h,alfa);
sysFO2=1/(Talfa*sysPSE+1);
yPSE=step(sysFO2,t);
profile viewer;
 profile clear;
profile clear;
profile on;
 y_an=(1-ml(((-t.^alfa)/Talfa),alfa));
profile viewer;
 profile clear;
```

Results of execution of the *profile* function are illustrated by screenshots 5, 6 and 7.

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
34	sysFO2=1/(Talfa*sysPSE+1);	1	0.114 s	60.0%	
33	sysPSE=dfod2(L,h,alfa);	1	0.045 s	23.8%	
35	yPSE=step(sysFO2,t);	1	0.028 s	14.6%	
39	profile viewer;	1	0.003 s	1.4%	
31	L=200;	1	0.000 s	0.0%	
All other lines			0.000 s	0.2%	
Totals			0.190 s	100%	

Fig. 5. Analysis of numerical complexity for FOBD approximation and $L = 200$.

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time
19	sysFO1=1/(Talfa*sysCFE+1);	1	0.102 s	53.1%
18	sysCFE=dfod1(M,h,a,alfa); %_KO...	1	0.064 s	33.3%
20	yCFE=step(sysFO1,t);	1	0.023 s	12.0%
24	profile viewer;	1	0.003 s	1.3%
16	M=5; %rząd CFE	1	0.000 s	0.0%
All other lines			0.000 s	0.2%
Totals			0.191 s	100%

Fig. 6. Analysis of numerical complexity for CFE approximation, $M = 5$, $a = 1$ (Tustin).

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time
46	y_an=(1-ml(((-t.^alfa)/Talfa),...	1	0.077 s	95.6%
47	profile viewer;	1	0.003 s	3.8%
All other lines			0.000 s	0.5%
Totals			0.080 s	100%

Fig. 7. Analysis of numerical complexity of the analytical solution.

From screenshots 5 and 6 it can be concluded that the most time consuming operation is the direct assignment of the tested transfer function using direct form of the transfer function obtained from functions *dfod1* and *dfod2*. There are lines no 19 in 6 and no 34 in 5.

In order to shorten the execution time of this operation, the source code was modified as follows. Firstly the functions *dfod1* and *dfod2* have been modified in such a way that they do not return the entire transfer function (12) or (18), but vectors containing parameters of this transfer function. Next, the approximated transfer function (20) or (22) has been expressed using these vectors. This is written as the following sequence of instructions:

```
%Approximation CFE:
start_CFE=tic;
M=5; %Order CFE
%a=0; %0.0 - Euler, 0.78-Al Aloui,
%1.0 - Tustin
[nCFE,dCFE]=dfod1_KOp(M,h,a,alfa); %_KOp
sysFO1=tf (dCFE,[Talfa*nCFE+dCFE],h,
'variable','z^-1');
yCFE=step(sysFO1,t);
t_CFE=toc(start_CFE);
%Approximation FOBD:
start_PSE=tic;
nPSE=dfod2_KOp(L,h,alfa);
sysPSE=tf([1],[Talfa*nPSE+1],h,
'variable','z^-1');
yPSE=step(sysPSE,t);
t_PSE=toc(start_PSE);
```

Next the above instructions has been tested using *tic, toc* and*profile* instructions, analogically as previously. Results are illustrated by tables 7 and 8, histograms (8) and screenshots 10 and 9.

Table 7. The mean of execution time for modified CFE approximation, order $M = 5$, $\alpha = 0.5$ and different values of parameter a.

a	0.00	0.78	1.00
Duration [s]	0.0017	0.0016	0.0016

Table 8. The mean of execution time for modified FOBD approximation, $\alpha = 0.5$ and different values of order L.

L	100	150	200	250	300
Duration [s]	0.0021	0.0021	0.0021	0.0021	0.0021

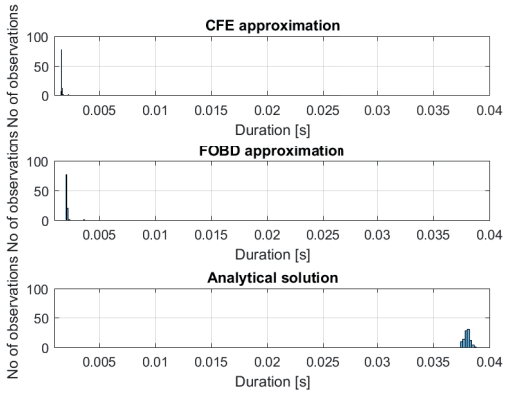

Fig. 8. Histograms of duration of computing of step response using modified CFE and FOBD approximations for: $\alpha = 0.5$, $T_\alpha = 5[s]$, $a = 1.0$ and $L = 200$.

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time
36	nPSE=dfod2_KOp(L,h,alfa);	1	0.044 s	53.3%
38	yPSE=step(sysPSE,t);	1	0.029 s	35.2%
37	sysPSE=tf([1],[Talfa*nPSE+1],h...	1	0.007 s	8.0%
39	profile viewer;	1	0.003 s	3.1%
31	L=200;	1	0.000 s	0.0%
All other lines			0.000 s	0.5%
Totals			0.082 s	100%

Fig. 9. Analysis of numerical complexity for modified FOBD approximation and $L = 200$.

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time
22	sysFO1=tf (dCFE,[Talfa*nCFE+dC...	1	0.041 s	40.6%
23	yCFE=step(sysFO1,t);	1	0.029 s	28.9%
21	[nCFE,dCFE]=dfod1_KOp(M,h,a,al...	1	0.028 s	27.6%
24	profile viewer;	1	0.003 s	2.6%
16	M=5; %rząd CFE	1	0.000 s	0.0%
All other lines			0.000 s	0.4%
Totals			0.100 s	100%

Fig. 10. Analysis of numerical complexity for modified CFE approximation, $M = 5$, $a = 1$ (Tustin).

Analysis of tables 5 vs 7 and 6 vs 8 allows to conclude that the proposed modification of the source code allows to shorten the duration of computations 6 to 8 times in comparing to the first version of source code, although the defining of transfer function is still being the longest operation.
Next, the modified CFE approximation is slightly faster than modified FOBD approximation. This is as expected, however the difference is small.
Finally it can be noted that the longest operation is the computing of the step response using the analytical formula (11). This is caused probably by the long calculation of the Mittag-Leffler function.

Conclusions

The main final conclusion is that the use of FOBD approximation is the best idea during implementation of the considered, elementary FO transfer function at MATLAB platform. Its use assures better accuracy than CFE with comparable duration of computations. Next, calculations can be accelerated by appropriate preparation of the source code.
The area of further investigations associated to the presented issue covers e.g. tests of accuracy and numerical complexity at industrial platform, e.g. PLC.

Author: *Ph D., D. Sc. Krzysztof Oprzędkiewicz, Dept of Automatic Control and Robotics, Faculty of Electrical Engineering, Automatic Control, Informatics and Biomedical Engineering, AGH University, al. A. Mickiewicza 30, 30-059 Kraków, Poland, e-mail:kop@agh.edu.pl*

Acknowledgments

This paper was sponsored by AGH UST project no 16.16.120.773.

REFERENCES

[1] M.A. Al-Alaoui. Novel digital integrator and differentiator. *Electronics Letters*, 29(4):376–378, 1993.
[2] M. Buslowicz and T. Kaczorek. Simple conditions for practical stability of positive fractional discrete-time linear systems. *International Journal of Applied Mathematics and Computer Science*, 19(2):263–269, 2009.
[3] R. Caponetto, G. Dongola, L. Fortuna, and I. Petras. Fractional order systems: Modeling and Control Applications. In Leon O. Chua, editor, *World Scientific Series on Nonlinear Science*, pages 1–178. University of California, Berkeley, 2010.
[4] Y. Q. Chen and K. L. Moore. Discretization schemes for fractional-order differentiators and integrators. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 49(3):263–269, 2002.
[5] S. Das. *Functional Fractional Calculus for System Identification and Controls*. Springer, Berlin, 2010.
[6] M. Dlugosz and P. Skruch. The application of fractional-order models for thermal process modelling inside buildings. *Journal of Building Physics*, 1(1):1–13, 2015.
[7] A. Dzieliński, D. Sierociuk, and G. Sarwas. Some applications of fractional order calculus. *Bulletin of the Polish Academy of Sciences, Technical Sciences*, 58(4):583–592, 2010.
[8] C.G. Gal and M. Warma. Elliptic and parabolic equations with fractional diffusion and dynamic boundary conditions. *Evolution Equations and Control Theory*, 5(1):61–103, 2016.
[9] T. Kaczorek. *Selected Problems of Fractional Systems Theory*. Springer, Berlin, 2011.
[10] T. Kaczorek and K. Rogowski. *Fractional Linear Systems and Electrical Circuits*. Bialystok University of Technology, Bialystok, 2014.
[11] A. Obrączka. *Control of heat processes with the use of non-integer models*. PhD thesis, AGH University, Krakow, Poland, 2014.
[12] K. Oprzędkiewicz, E. Gawin, and T. Gawin. Real-time plc implementations of fractional order operator. In *Automation 2018 : innovations in automation, robotics and measurment techniques : 21-23 March, Warsaw, Poland*, pages 36–51, 2018.
[13] K. Oprzędkiewicz, E. Gawin, and W. Mitkowski. Application of fractional order transfer functions to modeling of high - order systems. In *MMAR 2015 : 20th international conference on Methods and Models in Automation and Robotics : 24-27 August 2015, Międzyzdroje, Poland*, page 1169–1174, 2015.
[14] K. Oprzędkiewicz, W. Mitkowski, and E. Gawin. An accuracy estimation for a non integer order, discrete, state space model of heat transfer process. In *Automation 2017 : innovations in automation, robotics and measurment techniques : 15-17 March, Warsaw, Poland*, pages 86–98, 2017.
[15] P. Ostalczyk. Equivalent descriptions of a discrete-time fractional-order linear system and its stability domains. *International Journal of Applied Mathematics and Computer Science*, 22(3):533–538, 2012.
[16] P. Ostalczyk. *Discrete Fractional Calculus. Applications in Control and Image Processing*. World Scientific, New Jersey, London, Singapore, 2016.
[17] I. Petras. Fractional order feedback control of a dc motor. *Journal of Electrical Engineering*, 60(3):117–128, 2009.
[18] I. Petras. Digital fractional order differentiator/integrator - iir type (https://www.mathworks.com/matlabcentral/fileexchange/3672-digital-fractional-order-differentiator-integrator-iir-type), matlab central file exchange. retrieved january 30, 2023.http://people.tuke.sk/igor.podlubny/usu/matlab/petras/dfod1.m, 2023.
[19] I. Petras. Ivo petras (2023). digital fractional order differentiator/integrator - fir type. (https://www.mathworks.com/matlabcentral/fileexchange/3673-digital-fractional-order-differentiator-integrator-fir-type), matlab central file exchange. retrieved january 30, 2023., 2023.

[20] I. Podlubny. *Fractional Differential Equations*. Academic Press, San Diego, 1999.
[21] D. Sierociuk, T. Skovranek, M. Macias, I. Podlubny, I. Petras, A. Dzielinski, and P. Ziubinski. Diffusion process modeling by using fractional-order models. *Applied Mathematics and Computation*, 257(1):2–11, 2015.
[22] R. Stanisławski, K. Latawiec, and M. Łukaniszyn. A comparative analysis of Laguerre-based approximators to the Grünwald-Letnikov fractional-order difference. *Hindawi Publishing Corporation Mathematical Problems in Engineering*, 2015(1):1–10, 2015.