fractal and fractional

MDPI

Article

Numerical Algorithm for Calculating the Time Domain Response of Fractional Order Transfer Function

Lu Bai [1,*] and Dingyü Xue [2]

[1] School of Information Engineering, Shenyang University, Shenyang 110044, China
[2] School of Information Science and Engineering, Northeastern University, Shenyang 110819, China; xuedingyu@mail.neu.edu.cn
* Correspondence: bailu820601@163.com

Abstract: This paper proposes new numerical algorithms for calculating the time domain responses of fractional order transfer functions (FOTFs). FOTFs are divided into two categories, explicit fractional order transfer functions (EFOTFs) and implicit fractional order transfer functions (IFOTFs). Transforming an EFOTF into an equivalent fractional order differential equation, its time domain response can be obtained by solving the equation by the difference method. IFOTF cannot be transformed into an equivalent equation, so its time domain response cannot be calculated by existing difference methods. A new numerical algorithm is designed for calculating a convolution and its inverse operation, the time domain response of IFOTF can be calculated based on the algorithm. Error analysis shows that the proposed numerical algorithms are of first-order accuracy. Four calculation examples are presented, and the results are consistent with the theoretical analysis.

Keywords: fractional order transfer function; time domain response; numerical algorithm; benchmark problem

check for updates

Citation: Bai, L.; Xue, D. Numerical Algorithm for Calculating the Time Domain Response of Fractional Order Transfer Function. *Fractal Fract.* **2022**, *6*, 122. https://doi.org/10.3390/fractalfract6020122

Academic Editor: David Kubanek

Received: 20 January 2022
Accepted: 17 February 2022
Published: 19 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Fractional calculus is a theory about the integration and differentiation of non-integer order. It is a powerful tool to describe systems which have long-term memory and long-range spatial interaction. Fractional calculus has been applied in control theory in recent years, and there have been many theoretical achievements in controllability [1,2], observability [3,4], stability [5–7], robustness [8,9] and control methods [10–12]. At the same time, fractional order theory has also achieved great success in control engineering, such as the fractional order PID controller [13,14], air-based precision positioning system [15], fractional order positive position feedback compensator [16], and fractional order gray model [17,18]. More achievements can be found in [19].

As many novel fractional order transfer functions (FOTFs) have emerged in the above applications, an urgent problem is how to calculate their time domain responses. It is usually difficult to find the analytical expression of the time domain response; therefore, numerical algorithms are used to solve this problem. If a FOTF can be transformed into an equivalent fractional order differential equation, its time domain response can be calculated by solving the equation with a numerical algorithm; such a FOTF is called explicit fractional order transfer function (EFOTF) in this paper. There are many numerical algorithms for solving fractional order differential equations, such as the linear multi-step algorithm [20–22], separation of variables algorithm [23], predictor–corrector algorithm [24–26], matrix method [27] and some recently proposed algorithms [28–30]. In addition, the time domain response of EFOTF can also be calculated by MATLAB toolbox, FOTF [31].

If a FOTF cannot be transformed into an equivalent fractional order differential equation, its time domain response cannot be calculated by the difference algorithm; such a FOTF is called implicit fractional order transfer function (IFOTF) in this paper. Some

frequency methods are effective for some simple IFOTFs, for example, the time domain response of the following IFOTF can be calculated by the method proposed in [32].

$$F(s) = \frac{1}{\left(1 + \frac{s}{P_T}\right)^{\alpha}}, \tag{1}$$

where P_T is the transitional frequency, and α is a real number between 0 and 1. However, it is difficult to estimate the calculation error, and frequency methods are ineffective for more complex IFOTFs, such as

$$F(s) = \frac{340}{s^{0.756}(s^2 + 3.85s + 5880)^{1.15}}. \tag{2}$$

The IFOTF is a model of ionic polymer metal composite (IPMC) proposed in [33]; more information can be found in [34,35]. Therefore, a time domain algorithm is needed, which can calculate the time response of the IFOTF directly.

The main contribution of this work is to design a numerical algorithm for the IFOTF. The time domain response can be calculated by the proposed algorithm without considering the specific form of the IFOTF. The error analysis is presented, and it is proved that the proposed algorithm is of first-order accuracy. For the IFOTFs which can only be analyzed by frequency methods before, the proposed algorithm can give their time domain characteristics directly. The IFOTF is analyzed in both the frequency domain and time domain, so the result is more reliable.

Section 1 is devoted to expound the background and the significance of this work. The definition and the notation are introduced in Section 2. The first algorithm is proposed for EFOTF in Section 3. The second algorithm is designed for IFOTF based on the first algorithm in Section 4. The error analysis is presented in Section 5. Section 6 consists of four calculation examples. Section 7 contains the conclusion and the future work.

2. Definition

There are different fractional order definitions; Riemann–Liouville (RL) and Caputo definitions are used in this paper.

Definition 1 ([36]). *The RL fractional order derivative is defined as*

$${}^{\mathrm{RL}}_{t_0}\mathscr{D}^{\alpha}_{t} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \int_{t_0}^{t} \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} \mathrm{d}\tau, \tag{3}$$

where t_0 is an arbitrary real constant, α is a real constant bigger than zero, $n = \lceil \alpha \rceil$, variable $t \in [t_0, +\infty)$, Γ is the Gamma function.

Definition 2 ([37]). *The Caputo fractional order derivative is defined as*

$${}^{\mathrm{C}}_{t_0}\mathscr{D}^{\alpha}_{t} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_{t_0}^{t} \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}} \mathrm{d}\tau. \tag{4}$$

The parameters in (3) and (4) have the same meaning.

In (3) and (4), t_0 denotes the initial time. If $t_0 \neq 0$, it can always be converted to zero by variable substitution; therefore, t_0 is always equal to zero.

The transfer function refers to the ratio of the Laplace transform of the response (output) to the Laplace transform of the excitation (input) under the zero initial condition. Therefore, it is always assumed that the function has a zero initial condition when discussing

the transfer function. Under the zero initial condition, Definitions 1 and 2 are equivalent [36]; they are written as ${}_0\mathscr{D}_t^{\alpha} f(t)$, i.e.,

$$ {}_0\mathscr{D}_t^{\alpha} f(t) = {}_0^{\mathrm{RL}}\mathscr{D}_t^{\alpha} f(t) = {}_0^{\mathrm{C}}\mathscr{D}_t^{\alpha} f(t). \tag{5}$$

FOTFs are divided into two categories, EFOTF and IFOTF.

Definition 3. *The explicit fractional order transfer function (EFOTF) is defined as*

$$ F(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_1 s^{\beta_1}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_1 s^{\alpha_1}}, \tag{6}$$

where s is a complex variable, coefficients $a_1, \cdots, a_n$, $b_1, \cdots, b_m$ are all real constants, orders $\alpha_n > \cdots > \alpha_1 \geqslant 0$, $\beta_m > \cdots > \beta_1 \geqslant 0$ are all nonnegative real constants, and $\alpha_n \geqslant \beta_m$.

Definition 4. *The implicit fractional order transfer function (IFOTF) is defined as*

$$ F(s) = \frac{\left(b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_1 s^{\beta_1}\right)^{\mu_2/\nu_2}}{\left(a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_1 s^{\alpha_1}\right)^{\mu_1/\nu_1}}, \tag{7}$$

where μ_1, ν_1, μ_2, ν_2 are all positive integers, other parameters have the same meaning as the parameters in (6).

The topic of this paper is how to calculate the time domain response of (6) and (7). The details of the problem are described below. The time interval $[0, T]$ is divided into N identical subintervals and the step size $h = T/N$. The value of the excitation $u(t)$ at the grid point is written as $u_k (0 \leqslant k \leqslant N)$. The same notation is applied to the response $y(t)$, written as $y_k (0 \leqslant k \leqslant N)$. The problem is how to calculate the response y_k of (6) and (7), under some given excitation u_k.

3. The Numerical Algorithm for EFOTF

The numerator and denominator of (6) are denoted as $V(s)$ and $W(s)$, and the Laplace transforms of excitation $u(t)$ and response $y(t)$ are written as $U(s)$ and $Y(s)$. EFOTF (6) is written as

$$ W(s)Y(s) = V(s)U(s). \tag{8}$$

According to the convolution theorem, the inverse Laplace transform of (8) is

$$ \int_0^t w(\tau) y(t-\tau) \mathrm{d}\tau = \int_0^t v(\tau) u(t-\tau) \mathrm{d}\tau, \tag{9}$$

where $v(t)$ and $w(t)$ are the inverse Laplace transforms of $W(s)$ and $V(s)$, respectively. The trapezoidal rule is employed to calculate the two integrals in (9),

$$ h\left(\sum_{j=1}^{k-1} w_j\, y_{k-j} + \frac{w_0\, y_k + w_k\, y_0}{2}\right) = h\left(\sum_{j=1}^{k-1} v_j\, u_{k-j} + \frac{v_0\, u_k + v_k\, u_0}{2}\right) + O(h^2). \tag{10}$$

The recurrence formula is obtained from (10),

$$ y_k = \frac{2}{w_0} \sum_{j=1}^{k-1} (v_j\, u_{k-j} - w_j\, y_{k-j}) + \frac{v_0\, u_k + v_k\, u_0 - w_k\, y_0}{w_0} + O(h). \tag{11}$$

Formula (11) can be used to calculate y_k, as long as w_j and v_j are known. The following method is proposed for calculating w_j and v_j.

EFOTF (6) is equivalent to the following fractional order differential equation,

$$\sum_{i=1}^{n} a_i \, {}_0\mathscr{D}_t^{\alpha_i} y(t) = \sum_{i=1}^{m} b_i \, {}_0\mathscr{D}_t^{\beta_i} u(t), \tag{12}$$

with zero initial condition, its solution is the response of (6).

The fractional linear multistep method [21,22] is employed to calculate the fractional order derivative in (12),

$${}_0\mathscr{D}_t^{\alpha_i} y(t)|_{t=kh} = \frac{1}{h^{\alpha_i}} \sum_{j=0}^{k} \omega_j^{(\alpha_i,p)} y_{k-j} + O(h^p), \tag{13}$$

where the coefficient $\omega_j^{(\alpha_i,p)}$ is calculated by the following recursive formula,

$$\omega_j^{(\alpha_i,p)} = \begin{cases} 0, & j < 0, \\ \theta_0^{\alpha_i}, & j = 0, \\ -\dfrac{1}{\theta_0} \displaystyle\sum_{k=1}^{p} \theta_k \left(1 - k\,\frac{1+\alpha_i}{j}\right) \omega_{j-k}^{(\alpha_i,p)}, & j > 0. \end{cases} \tag{14}$$

The calculation error of (13) is $O(h^p)$, and the proof can be found in [21].

Plug (13) into (12). It yields

$$\sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}} \sum_{j=0}^{k} \omega_j^{(\alpha_i,p)} y_{k-j} = \sum_{i=0}^{m} \frac{b_i}{h^{\beta_i}} \sum_{j=0}^{k} \omega_j^{(\beta_i,p)} u_{k-j} + O(h^p). \tag{15}$$

Multiply both sides of (15) by step size h, transpose the order of summation,

$$h \sum_{j=0}^{k} \left(\sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}} \omega_j^{(\alpha_i,p)} \right) \cdot y_{k-j} = h \sum_{j=0}^{k} \left(\sum_{i=0}^{m} \frac{b_i}{h^{\beta_i}} \omega_j^{(\beta_i,p)} \right) \cdot u_{k-j} + O(h^{p+1}). \tag{16}$$

Set $p = 1$ in (16) and compare (10) and (16). It yields the recurrence formula for calculating w_j and v_j,

$$\begin{cases} w_j = \displaystyle\sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}} \omega_j^{(\alpha_i,1)} + O(h),\ v_j = \sum_{i=0}^{m} \frac{b_i}{h^{\beta_i}} \omega_j^{(\beta_i,1)} + O(h), & \text{when } 0 < j < k, \\ w_j = 2\displaystyle\sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}} \omega_j^{(\alpha_i,1)} + O(h),\ v_j = 2\sum_{i=0}^{m} \frac{b_i}{h^{\beta_i}} \omega_j^{(\beta_i,1)} + O(h), & \text{when } j = 0 \text{ or } k. \end{cases} \tag{17}$$

The above calculation process is summarized as the numerical algorithm.

Algorithm 1. *Calculating the time domain response of EFOTF (6).*

1. *Calculate w_j, v_j by (17).*
2. *Use (11) to calculate y_k, the time domain response at the grid point.*

4. The Numerical Algorithm for IFOTF

There is no equation equivalent to IFOTF (7), so its time domain response cannot be calculated by Algorithm 1. A new numerical algorithm is proposed for IFOTF in this section.

The numerator and denominator of (7) are denoted as $\hat{V}(s)$ and $\hat{W}(s)$, and the Laplace transforms of excitation $u(t)$ and response $y(t)$ are written as $U(s)$ and $Y(s)$. IFOTF (7) is written as

$$\hat{W}(s)Y(s) = \hat{V}(s)U(s). \tag{18}$$

Take the $\nu_1\nu_2$th power of both sides of (18),

$$\hat{W}^{\nu_1\nu_2}(s)Y^{\nu_1\nu_2}(s) = \hat{V}^{\nu_1\nu_2}(s)U^{\nu_1\nu_2}(s), \tag{19}$$

where $\hat{W}^{\nu_1\nu_2}(s)$ and $\hat{V}^{\nu_1\nu_2}(s)$ are polynomials. Consider $U^{\nu_1\nu_2}(s)$ as the excitation and $Y^{\nu_1\nu_2}(s)$ as the response. An equivalent EFOTF is obtained,

$$\frac{Y^{\nu_1\nu_2}(s)}{U^{\nu_1\nu_2}(s)} = \frac{\left(\sum_{i=1}^{n} a_i\, s^{\alpha_i}\right)^{\mu_1}}{\left(\sum_{i=1}^{m} b_i\, s^{\beta_i}\right)^{\mu_2}}. \tag{20}$$

4.1. Numerical Convolution

According to the convolution theorem, the inverse Laplace transform of $U^{\nu_1\nu_2}(s)$ is

$$\mathscr{L}^{-1}\{U^{\nu_1\nu_2}(s)\} = \underbrace{u(t) * u(t) * \cdots * u(t)}_{\nu_1\nu_2}. \tag{21}$$

The notation $*$ denotes a convolution,

$$u(t) * u(t) = \int_0^t u(t-\tau)u(\tau)\mathrm{d}\tau. \tag{22}$$

The value of $u(jh)$ is written as u_j, and h is the step size. Assume that the first-order error is contained in u_j, i.e.,

$$u_j = u(jh) + O(h),\ (0 \leqslant j \leqslant N). \tag{23}$$

Apply the trapezoidal rule to calculate the value of (22),

$$\begin{aligned}\int_0^{kh} u(kh-\tau)u(\tau)\mathrm{d}\tau &= h\sum_{j=0}^{k} u_{k-j}\,u_j - \frac{h}{2}(u_0u_k + u_ku_0) + O(h^2)\sum_{j=0}^{k} u_j + O(h^2)\\ &= h\sum_{j=0}^{k} u_{k-j}\,u_j + O(h),\ (0 \leqslant k \leqslant N).\end{aligned} \tag{24}$$

Although u_j contains $O(h)$, (24) is still of first-order accuracy. Therefore, the right side of (21) can be calculated by applying (24) repeatedly. The result remains of first-order accuracy.

The generating polynomial is introduced to simplify the expression. The generating polynomial of u_j is defined as

$$\bar{u}(x) = \sum_{j=0}^{N} u_j\, x^j, \tag{25}$$

where u_j is defined as (23). The square of $\bar{u}(x)$ is

$$\bar{u}^2(x) = \sum_{k=0}^{2N}\Big(\sum_{j=0}^{k} u_{k-j}\,u_j\Big)x^k. \tag{26}$$

Comparing (24) and (26), it yields that the result of (24) is the coefficient of (26) multiplied by step size h. The right side of (21) can be calculated by this method. It is expressed as,

$$\underbrace{u(t) * u(t) * \cdots * u(t)}_{\nu_1\nu_2}|_{t=kh} = h^{\nu_1\nu_2-1}\mathscr{S}_k(\bar{u}^{\nu_1\nu_2}(x)) + O(h),\ (0 \leqslant k \leqslant N), \tag{27}$$

where $\mathscr{S}_k$ represents selecting the coefficient of x^k.

The result of (27) is the value of $\mathscr{L}^{-1}\{U^{\nu_1\nu_2}(s)\}$ at grid point. It is the excitation of (20). Because (20) is an EFOTF, its response can be calculated by Algorithm 1. The result is written as R_k $(0 \leqslant k \leqslant N)$. The result is also the value of $\mathscr{L}^{-1}\{Y^{\nu_1\nu_2}(s)\}$ at grid point, so it yields the equation,

$$\underbrace{y(t) * y(t) * \cdots * y(t)}_{\nu_1\nu_2} |_{t=kh} = R_k + O(h),\ (0 \leqslant k \leqslant N). \tag{28}$$

4.2. Numerical Inversion of Convolution

The solution of (28) is written as $y_j (0 \leqslant j \leqslant N)$, and its generating polynomial is

$$\bar{y}(x) = \sum_{j=0}^{N} y_j\, x^j. \tag{29}$$

Calculate the left side of (28) by the same method as (27) and ignore the small quantity term $O(h)$. The following equation is obtained:

$$h^{\nu_1\nu_2-1}\mathscr{S}_k(\bar{y}^{\nu_1\nu_2}(x)) = R_k,\ (0 \leqslant k \leqslant N). \tag{30}$$

Let $x = 0$ and $k = 0$ in (30). It yields

$$y_0 = \left(R_0 h^{1-\nu_1\nu_2}\right)^{\frac{1}{\nu_1\nu_2}}. \tag{31}$$

It is supposed that $y_0, \cdots, y_{k-1}$ are calculated, and thus, y_k can be calculated by the following method. The polynomial $\bar{y}(x)$ is written as the sum of three parts,

$$\bar{y}(x) = \bar{y}_l(x) + \bar{y}_e(x) + \bar{y}_h(x) = \sum_{j=0}^{k-1} y_j\, x^j + y_k\, x^k + \sum_{j=k+1}^{N} y_j\, x^j. \tag{32}$$

According to the binomial theorem,

$$\begin{aligned}
\bar{y}^{\nu_1\nu_2}(x) &= \left(\bar{y}_l(x) + \bar{y}_e(x)\right)^{\nu_1\nu_2} + \sum_{j=1}^{\nu_1\nu_2}\binom{\nu_1\nu_2}{j}\bar{y}_h^j(x)\left(\bar{y}_l(x) + \bar{y}_e(x)\right)^{\nu_1\nu_2-j} \\
&= \sum_{j=0}^{\nu_1\nu_2}\binom{\nu_1\nu_2}{j}\bar{y}_e^j(x)\bar{y}_l^{\nu_1\nu_2-j}(x) + \sum_{j=1}^{\nu_1\nu_2}\binom{\nu_1\nu_2}{j}\bar{y}_h^j(x)\left(\bar{y}_l(x) + \bar{y}_e(x)\right)^{\nu_1\nu_2-j} \\
&= \bar{y}_l^{\nu_1\nu_2}(x) + \nu_1\nu_2\bar{y}_l^{\nu_1\nu_2-1}(x)\bar{y}_e(x) + \sum_{j=2}^{\nu_1\nu_2}\binom{\nu_1\nu_2}{j}\bar{y}_e^j(x)\bar{y}_l^{\nu_1\nu_2-j}(x) \\
&+ \sum_{j=1}^{\nu_1\nu_2}\binom{\nu_1\nu_2}{j}\bar{y}_h^j(x)\left(\bar{y}_l(x) + \bar{y}_e(x)\right)^{\nu_1\nu_2-j}.
\end{aligned} \tag{33}$$

In (33), only the first and the second terms contain x^k, B_{k1} denotes the coefficient of x^k contained in the first term, and B_{k2} denotes the coefficient of x^k contained in the second term. Plug B_{k1} and B_{k2} into (30). It yields,

$$B_{k1} + B_{k2} = R_k h^{1-\nu_1\nu_2}. \tag{34}$$

Expand the second term of (33),

$$\begin{aligned} \nu_1\nu_2\bar{y}_l^{\nu_1\nu_2-1}(x)\bar{y}_e(x) &= \nu_1\nu_2 y_k x^k \Big(\sum_{j=0}^{k-1} y_j\, x^j\Big)^{\nu_1\nu_2-1} \\ &= \nu_1\nu_2 y_k y_0^{\nu_1\nu_2-1} x^k + \nu_1\nu_2 y_k x^k \Big(\sum_{j=1}^{k-1} y_j\, x^j\Big)^{\nu_1\nu_2-1}. \end{aligned} \tag{35}$$

The second term of (35) only contains the term whose order is higher than k, so B_{k2} is equal to the coefficient of the fist term in (35),

$$B_{k2} = \nu_1\nu_2 y_0^{\nu_1\nu_2-1} y_k. \tag{36}$$

Lemma 1 is cited from [38] for expanding the first term of (33).

Lemma 1 ([38]). *Let c be a positive integer. For all $x_1, x_2, \cdots, x_d$,*

$$(x_1 + x_2 + \cdots + x_d)^c = \sum \binom{c}{c_1\ c_2\ \cdots\ c_d} x_1^{c_1} x_2^{c_2} \cdots x_d^{c_d}, \tag{37}$$

where the summation extends over all nonnegative integer solutions $c_1, c_2, \cdots, c_d$ of $c_1 + c_2 + \cdots + c_d = c$, and the multinomial coefficients are defined by

$$\binom{c}{c_1\ c_2\ \cdots\ c_d} = \frac{n!}{c_1! c_2! \cdots c_d!}. \tag{38}$$

Expand the first term of (33) by Lemma 1. The coefficient of x^k is

$$B_{k1} = \sum \binom{\nu_1\nu_2}{c_0\ c_1\ \cdots\ c_{k-1}} y_0^{c_0} y_1^{c_1} \cdots y_{k-1}^{c_{k-1}}. \tag{39}$$

The summation extends over all nonnegative integer solutions $c_1, c_2, \cdots, c_{k-1}$, which satisfy the following equations,

$$\begin{cases} \sum_{j=0}^{k-1} c_j = \nu_1\nu_2, \\ \sum_{j=0}^{k-1} j c_j = k. \end{cases} \tag{40}$$

According to (39), B_{k1} can be calculated directly with $y_0, y_1, \cdots, y_{k-1}$, which are calculated. Plug (36) and (39) into (34). It yields the formula for calculating y_k,

$$y_k = \frac{R_k h^{1-\nu_1\nu_2} - B_{k1}}{\nu_1\nu_2 y_0^{\nu_1\nu_2-1}}. \tag{41}$$

The above calculation process is summarized as the numerical algorithm.

Algorithm 2. *Calculating the time domain response of IFOTF (7).*

1. *Transform IFOTF (7) into EFOTF (20).*
2. *Calculate the time domain excitation of (20) by (27).*
3. *Solve the time domain response of (20) by Algorithm 1; obtain equation (30).*
4. *Calculate y_0 by (31), and calculate y_k $(0 \leqslant k \leqslant N)$ by (41).*

5. Error Analysis

Two formulas, (11) and (17), are employed in Algorithm 1; the parameters in (11) are calculated by (17). It yields that Algorithm 1 is of first-order accuracy by plugging (17) into (11).

Algorithm 2 consists of three parts. In the first part, IFOTF (7) is transformed into EFOTF (20), the excitation of (20) is calculated by (27), which is of first-order accuracy. The generating polynomial is introduced solely for the sake of convenience in expression; it does not produce calculation error. Therefore, the result of this part is of first-order accuracy.

In the second part, the response of EFOTF (20) is calculated by Algorithm 1; it is of first-order accuracy. It is worth noting that the excitation of EFOTF (20) is used in this part; it contains the first-order error. This error does not affect the calculation accuracy of (11); therefore, the result of the second part is of first-order accuracy.

In the last part, (28) is solved by (41), and solution y_k is the time domain response of IFOTF (7). Use p to denote the order of error, i.e.,

$$y(kh) = y_k + O(h^p). \tag{42}$$

Plug y_k into (30). The calculation is of pth-order accuracy if $p < 1$; otherwise, the calculation is of first-order accuracy.

Assume $p < 1$ and plug (42) into the left side of (30). The result is of pth-order accuracy, i.e.,

$$h^{\nu_1\nu_2-1}\mathscr{S}_k(\bar{y}^{\nu_1\nu_2}(x)) = R_k + O(h^p). \tag{43}$$

It is contradictory to (28) because (28) and (43) have the same solution. It yields $p \geqslant 1$.

The solution y_k is calculated by (41). The numerator of (41) contains R_k, which is calculated by Algorithm 1. Algorithm 1 is of first-order accuracy, so the calculation accuracy of y_k cannot be higher than the first order, i.e., $p \leqslant 1$. Combined with the preceding analysis, it yields $p = 1$; Algorithm 2 is of first-order accuracy.

6. Calculation Examples

Four examples are presented in this section, the calculation results are used to validate the proposed algorithms.

6.1. Example 1

An EFOTF is expressed as

$$Y(s) = \frac{1}{s^{0.7} + s^{0.5}}U(s), \tag{44}$$

where $Y(s)$ and $U(s)$ are the Laplace transforms of response and excitation, respectively. The inverse Laplace transform of $U(s)$ is

$$u(t) = \frac{\Gamma(1.8)}{\Gamma(1.1)}t^{0.1} + \frac{\Gamma(1.8)}{\Gamma(1.3)}t^{0.3}. \tag{45}$$

It can be verified that the time domain response is

$$y(t) = t^{0.8}. \tag{46}$$

Use Algorithm 1 to calculate the time domain response in $[0, 10]$. Select step size $h = 0.01$; the analytical and numerical solutions are shown in Figure 1. There are two curves in Figure 1, and because the calculation error is very small, the two curves coincide.

Use (46) to test the accuracy of the numerical solution. The computation error is defined as $|y(jh) - y_j|$, and the computation errors under different step sizes are reported in Table 1. The error decreases along with decreasing the step size h, and they are proportional

to each other. This result is consistent with the theoretical analysis, Algorithm 1 is of first-order accuracy.

Table 1. Computation error of the time domain response of EFOTF (44).

Step Size	$t = 2$	$t = 4$	$t = 6$	$t = 8$	$t = 10$
$h = 0.1$	8.2728×10^{-3}	8.4603×10^{-3}	8.4762×10^{-3}	8.3949×10^{-3}	8.3039×10^{-3}
$h = 0.05$	4.7671×10^{-3}	4.7630×10^{-3}	4.6350×10^{-3}	4.5479×10^{-3}	4.4700×10^{-3}
$h = 0.01$	1.1865×10^{-3}	1.1491×10^{-3}	1.0730×10^{-3}	1.0384×10^{-3}	1.0109×10^{-3}
$h = 0.005$	6.3320×10^{-4}	6.0817×10^{-4}	5.6145×10^{-4}	5.4124×10^{-4}	5.2541×10^{-4}
$h = 0.001$	1.4169×10^{-4}	1.3431×10^{-4}	1.2169×10^{-4}	1.1655×10^{-4}	1.1261×10^{-4}

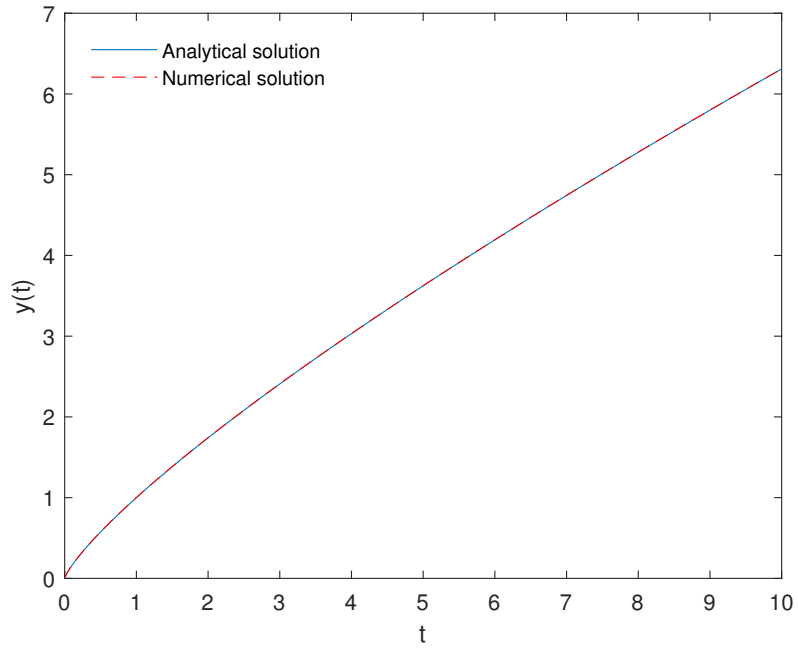

Figure 1. The time domain response of EFOTF (44).

6.2. Example 2

Algorithm 1 can also be used to solve fractional order differential equations. This example was previously completed by the author of this paper; only the problem and result are cited, and the calculation process can be found in [39].

The equation is proposed in [26],

$$ {}_0^{\mathrm{C}}\mathscr{D}_t^{1.455} y(t) = -t^{0.1} \frac{\mathscr{E}_{1,1.545}(-t)}{\mathscr{E}_{1,1.445}(-t)} \mathrm{e}^t y(t) {}_0^{\mathrm{C}}\mathscr{D}_t^{0.555} y(t) + \mathrm{e}^{-2t} - \left[{}_0^{\mathrm{C}}\mathscr{D}_t^{1} y(t) \right]^2, \tag{47} $$

where $\mathscr{E}_{1,1.545}(-t)$ and $\mathscr{E}_{1,1.445}(-t)$ are two-parameter Mittag–Leffler functions. The analytical solution is $y(t) = \mathrm{e}^{-t}$ under the initial condition $y(0) = 1, y'(0) = -1$.

Solve (47) by Algorithm 1 and the PECE-type Adams algorithm proposed in [26]. The maximal error and the execution time are reported in Table 2; the result of the PECE-type Adams algorithm is cited from [26], and the result of Algorithm 1 is cited from [39]. Considering the difference of computer hardware, it is of little significance to compare the execution time. The maximal error is an important parameter for testing an algorithm. It is obvious that Algorithm 1 is more accurate; its maximal error is proportional to the step size.

Table 2. Comparison of the errors and the execution times.

Step Size	Standard PECE-Type Adams Algorithm		Algorithm 1	
	Maximal Error	Execution Time	Maximal Error	Execution Time
$h = 1/200$	0.3904	101.2 s	0.01250	0.35667 s
$h = 1/400$	0.2193	368.4 s	0.00681	0.43396 s
$h = 1/800$	0.1164	1358.0 s	0.00358	0.99232 s
$h = 1/1600$	0.0600	5017.4 s	0.00185	2.54765 s

6.3. Example 3

An IFOTF is expressed as

$$Y(s) = \frac{1}{(4s+1)^{0.5}} U(s), \tag{48}$$

where $Y(s)$ and $U(s)$ are the Laplace transforms of response and excitation, respectively. The inverse Laplace transform of $U(s)$ is

$$u(t) = t^2. \tag{49}$$

The time domain response is

$$y(t) = 0.5 \mathrm{e}^{-\frac{t}{4}} {}_0\mathscr{D}_t^{-0.5} \left(t^2 \mathrm{e}^{\frac{t}{4}} \right). \tag{50}$$

The analytical expression of (50) can be verified by the formula introduced in [40]. An IFOTF is expressed as

$$Y(s) = \frac{1}{(\tau s+1)^{\nu}} U(s), \tag{51}$$

and the analytical expression of the time domain response is

$$y(t) = \tau^{-\nu} \mathrm{e}^{-\frac{t}{\tau}} {}_0\mathscr{D}_t^{-\nu} \left(u(t) \mathrm{e}^{\frac{t}{\tau}} \right). \tag{52}$$

Use Algorithm 2 to calculate the time domain response in $[0, 10]$. Select step size $h = 0.01$; the analytical and numerical solutions are shown in Figure 2. There are two curves in Figure 2; because the computation error is very small, the two curves coincide.

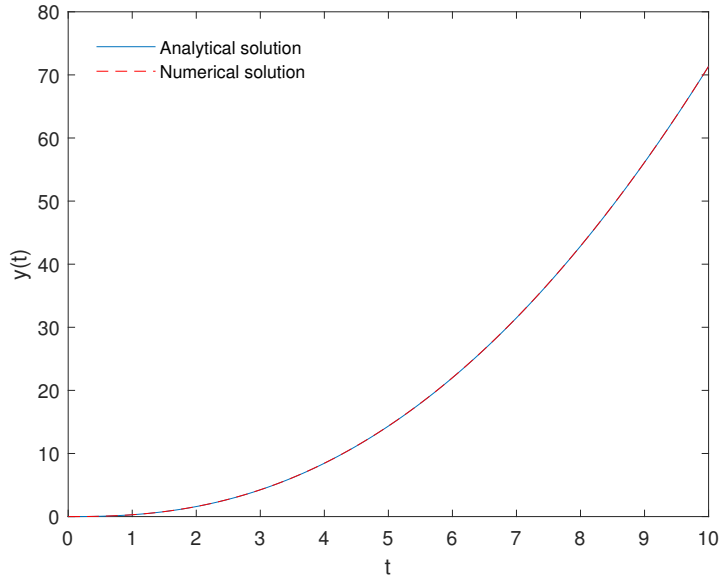

Figure 2. The time domain response of IFOTF (48).

Use (50) to test the accuracy of the numerical solution. The computation error is defined as $|y(jh) - y_j|$, computation errors under different step sizes are reported in Table 3. The error is proportional to step size, the result is consistent with the theoretical analysis, Algorithm 2 is of first-order accuracy.

Table 3. Computation error of the time domain response of IFOTF (48).

Step Size	$t = 2$	$t = 4$	$t = 6$	$t = 8$	$t = 10$
$h = 0.1$	3.9420×10^{-2}	6.3127×10^{-2}	1.2113×10^{-1}	1.4719×10^{-1}	1.6519×10^{-1}
$h = 0.05$	1.9809×10^{-2}	3.1679×10^{-2}	6.0705×10^{-2}	7.3727×10^{-2}	8.2712×10^{-2}
$h = 0.01$	3.9728×10^{-3}	6.3512×10^{-3}	1.2162×10^{-2}	1.4766×10^{-2}	1.6560×10^{-2}
$h = 0.005$	1.9870×10^{-3}	3.1764×10^{-3}	6.0825×10^{-3}	7.3841×10^{-3}	8.2810×10^{-3}
$h = 0.001$	3.9748×10^{-4}	6.3543×10^{-4}	1.2167×10^{-3}	1.4770×10^{-3}	1.6564×10^{-3}

6.4. Example 4

IFOTF (2) is a model for the ionic polymer metal composite (IPMC) actuator [33], which is mentioned at the beginning of the paper. Its time domain response is calculated in this example. Rewrite its expression

$$Y(s) = \frac{340}{s^{0.756}(s^2 + 3.85s + 5880)^{1.15}} U(s), \tag{53}$$

where $Y(s)$ and $U(s)$ are the Laplace transforms of response and excitation, respectively. The expression of $U(s)$ is

$$U(s) = \frac{\Gamma(8)}{(s+1)^8}, \tag{54}$$

the inverse Laplace transform of $U(s)$ is

$$u(t) = t^7 \mathrm{e}^{-t}. \tag{55}$$

The expression of $Y(s)$ is

$$Y(s) = \frac{340\Gamma(8)}{(s+1)^8 s^{0.756}(s^2 + 3.85s + 5880)^{1.15}}. \tag{56}$$

Use Algorithm 2 to calculate the time domain response in $[0, 20]$. The time domain response is calculated by another method to verify the result of Algorithm 2. The inverse Laplace transform of (56) is calculated by the numerical algorithm proposed in [41]. Select step size $h = 0.01$; the results of the two algorithms are plotted in Figure 3. The two curves almost coincide, and it verifies that the two results are credible. It is worthy noting that (56) is used in the numerical inversion of the Laplace transform, but not used in Algorithm 2. That means that the expression of the excitation is not necessary for Algorithm 2; it is a more effective method in practical application.

The difference is defined as the result of Algorithm 2 minus the result of the numerical inversion of the Laplace transform. The difference is reported in Table 4; the difference decreases along with decreasing the step size. It shows that the two results are credible, although neither of them is an exact solution.

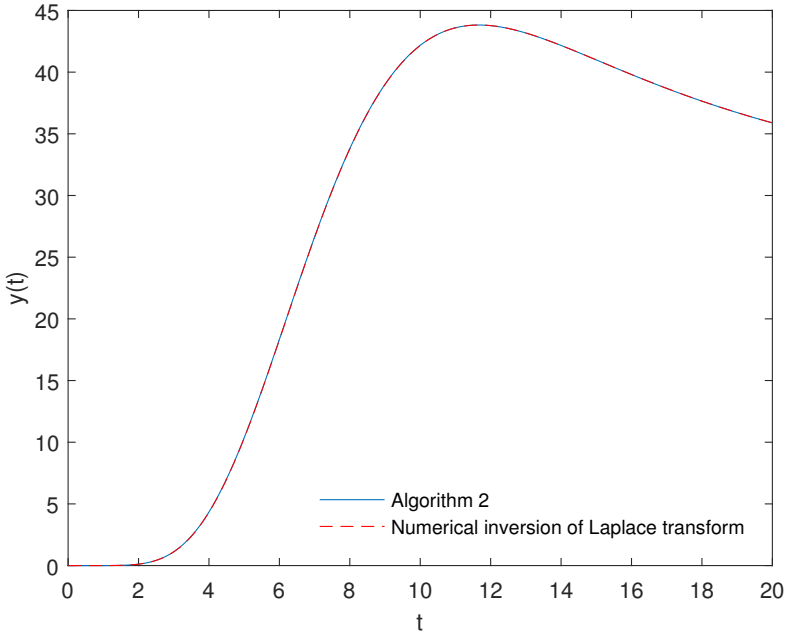

Figure 3. The time domain response of IFOTF (2).

Table 4. Difference between the results of the two algorithms.

Step Size	$t = 4$	$t = 8$	$t = 12$	$t = 16$	$t = 20$
$h = 0.2$	3.55×10^{-1}	4.70×10^{-1}	-2.26×10^{-2}	-8.80×10^{-2}	-5.97×10^{-2}
$h = 0.1$	1.76×10^{-1}	2.36×10^{-1}	-1.11×10^{-2}	-4.41×10^{-2}	-3.00×10^{-2}
$h = 0.05$	8.76×10^{-2}	1.18×10^{-1}	-5.55×10^{-3}	-2.21×10^{-2}	-1.51×10^{-2}
$h = 0.02$	3.48×10^{-2}	4.71×10^{-2}	-2.31×10^{-3}	-8.96×10^{-3}	-6.12×10^{-3}
$h = 0.01$	1.73×10^{-2}	2.35×10^{-2}	-1.24×10^{-3}	-4.56×10^{-3}	-3.14×10^{-3}

7. Discussion

Two numerical algorithms are proposed in this paper. The novelty is that the time domain response can be calculated without considering the form of FOTF. Although frequency methods are also effective for some simple IFOTFs, they are ineffective for more complex IFOTFs. The proposed algorithms are useful tools for analyzing FOTFs, and also for verifying the results of the frequency methods. The FOTF is analyzed in both the frequency domain and time domain; thus, the result is more reliable. Four examples are presented in the paper; the calculation results are consistent with the theoretical analysis. Examples 1 and 3 are simple problems for which the analytical expressions of the time domain responses are provided. The computation error can be obtained with the analytical expression, so the two examples can be used as benchmark problems for testing numerical algorithms; they are helpful for researchers to test their own numerical algorithms. In Example 3, Algorithm 1 is compared with the PECE-type Adams algorithm, and the result shows that Algorithm 1 is more accurate. In Example 4, the time domain response of (2) is calculated by Algorithm 2. The IFOTF (2) is the model of IPMC; the authors of this paper have not found other numerical algorithms to calculate its time domain response. If the parameters of Algorithm 1 are changed, a more accurate solution can be obtained; however, Algorithm 2 is of first-order accuracy. In future work, further studies will be carried out to improve the calculation accuracy of Algorithm 2.

Author Contributions: Conceptualization, L.B. and D.X.; methodology, L.B.; writing—original draft preparation, L.B.; writing—review and editing, L.B. and D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, B.; Su, H.S.; Wu, L.C.; Li, X.L.; Lu, X. Fractional-order controllability of multi-agent systems with time-delay. *Neurocomputing* **2021**, *424*, 268–277. [CrossRef]
2. Ahmad, I.; Rahman, G.U.; Ahmad, S.; Alshehri, N.A.; Elagan, S.K. Controllability of a damped nonlinear fractional order integrodifferential system with input delay. *Alex. Eng. J.* **2022**, *61*, 1956–1966. [CrossRef]
3. Shamardan, A.B.; Moubarak, M.R.A. Controllability and observability for fractional control systems. *J. Fract. Calc.* **1999**, *15*, 25–34.
4. Hassanzadeh, I.; Tabatabaei, M. Calculation of controllability and observability matrices for special case of continuous-time multi-order fractional systems. *ISA Trans.* **2018**, *82*, 62–72. [CrossRef]
5. Zhang, X.F.; Zhao, Z.L. Normalization and stabilization for rectangular singular fractional order T-S fuzzy systems. *Fuzzy Sets Syst.* **2020**, *381*, 140–153. [CrossRef]
6. Zhang, X.F. Relationship between integer order systems and fractional order systems and its two applications. *IEEE-CAA J. Autom. Sin.* **2018**, *5*, 639–643. [CrossRef]
7. Zhang, X.F.; Chen, Y.Q. Admissibility and robust stabilization of continuous linear singular fractional order systems with the fractional order α: The $0 < \alpha < 1$ case. *ISA Trans.* **2018**, *82*, 42–50.
8. Chen, Y.Q.; Ahn, H.S.; Xue, D.Y. Robust controllability of interval fractional order linear time invariant systems. *Signal Process.* **2006**, *86*, 2794–2802. [CrossRef]
9. Zhang, J.X.; Yang, G.H. Robust Adaptive Fault-Tolerant Control for a Class of Unknown Nonlinear Systems. *IEEE Trans. Ind. Electron.* **2017**, *64*, 585–594. [CrossRef]
10. Zhang, X.F.; Huang, W.K. Adaptive neural network sliding mode control for nonlinear singular fractional order systems with mismatched uncertainties. *Fractal Fract.* **2020**, *4*, 50. [CrossRef]
11. Zhang, J.X.; Yang, G.H. Low-complexity tracking control of strict-feedback systems with unknown control directions. *IEEE Trans. Autom. Control* **2019**, *64*, 5175–5182. [CrossRef]
12. Zhang, J.X.; Yang, G.H. Fuzzy adaptive output feedback control of uncertain nonlinear systems with prescribed performance. *IEEE Trans. Cybern.* **2018**, *48*, 1342–1354. [CrossRef] [PubMed]
13. Mandić, P.D.; Šekara, T.B.; Lazarević, M.P.; Bošković, M. Dominant pole placement with fractional order PID controllers: D-decomposition approach. *ISA Trans.* **2017**, *67*, 76–86. [CrossRef] [PubMed]
14. Mandić, P.D.; Lazarević, M.P.; Šekara, T.B. D-Decomposition technique for stabilization of furuta pendulum: Fractional approach. *Bull. Pol. Acad. Sci. Tech. Sci.* **2016**, *64*, 189–196. [CrossRef]
15. Krijnen, M.E.; van Ostayen, R.; HosseinNia, S.H. The application of fractional order control for an air-based contactless actuation system. *ISA Trans.* **2018**, *82*, 172–183. [CrossRef]
16. Marinangeli, L.; Alijani, F.; HosseinNia, S.H. Fractional-order positive position feedback compensator for active vibration control of a smart composite plate. *J. Sound Vib.* **2018**, *412*, 1–16. [CrossRef]
17. Yang, Y.; Xue, D.Y. A novel actual load forecasting by interval grey modeling methodology based on the fractional calculus. *ISA Trans.* **2018**, *82*, 200–209. [CrossRef]
18. Yang, Y.; Xue, D.Y. Continuous fractional-order grey model and electricity prediction research based on the observation error feedback. *Energy* **2016**, *115*, 722–733. [CrossRef]
19. Sun, H.G.; Yong, Z.; Baleanu, D.; Chen, W.; Chen, Y.Q. A new collection of real world applications of fractional calculus in science and engineering. *Commun. Nonlinear Sci. Numer. Simul.* **2018**, *64*, 213–231. [CrossRef]
20. Lubich, C. On the stability of linear multistep methods for Volterra convolution equations. *IMA J. Numer. Anal.* **1983**, *3*, 439–465. [CrossRef]
21. Lubich, C. Discretized fractional calculus. *SIAM J. Math. Anal.* **1986**, *17*, 704–719. [CrossRef]
22. Lubich, C. Fractional linear multistep methods for Abel-Volterra integral equations of the first kind. *IMA J. Numer. Anal.* **1987**, *7*, 97–106. [CrossRef]
23. Daftardar-Gejji, V.; Bhalekara, S. Boundary value problems for multi-term fractional differential equations. *J. Math. Anal. Appl.* **2008**, *345*, 754–765. [CrossRef]
24. Diethelm, K.; Ford, N.J.; Freed, A.D. A predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dyn.* **2002**, *29*, 3–22. [CrossRef]
25. Diethelm, K. An investigation of some nonclassical methods for the numerical approximation of Caputo-type fractional derivatives. *Numer. Algorithms* **2008**, *47*, 361–390. [CrossRef]
26. Diethelm, K. *The Analysis of Fractional Differential Equations*; Springer: Berlin, Germany, 2010; pp. 229–248.
27. Podlubny, I. Matrix approach to discrete fractional calculus. *Fract. Calc. Appl. Anal.* **2000**, *3*, 359–386.
28. Wei, Y.Q.; Liu, D.Y.; Driss, B.; Chen, Y.M. An improved pseudo-state estimator for a class of commensurate fractional order linear systems based on fractional order modulating functions. *Syst. Control. Lett.* **2018**, *118*, 29–34. [CrossRef]
29. Chen, Y.M.; Liu, L.Q.; Liu, D.Y.; Boutat, D. Numerical study of a class of variable order nonlinear fractional differential equation in terms of Bernstein polynomials. *Ain Shams Eng. J.* **2018**, *9*, 1235–1241. [CrossRef]

30. Liu, D.Y.; Tian, Y.; Boutat, D.; Laleg-Kirati, T. An algebraic fractional order differentiator for a class of signals satisfying a linear differential equation. *Signal Process.* **2015**, *116*, 78–90. [CrossRef]
31. Xue, D.Y. *Fractional-Order Control Systems-Fundamentals and Numerical Implementations*; De Gruyter: Berlin, Germany, 2017; pp. 1–30.
32. Mansouri, R.; Bettayeb, M.; Djennoune, S. Approximating of high order integer systems by fractional order reduced parameter models. *Math. Comput. Model.* **2010**, *51*, 53–62. [CrossRef]
33. Caponetto, R.; Dongola, G.; Fortuna, L.; Graziani, S.; Strazzeri, S. A fractional model for IPMC actuators. In Proceedings of the 2008 IEEE Instrumentation and Measurement Technology Conference, Victoria, BC, Canada, 20 June 2008.
34. Caponetto, R.; Graziani, S.; Pappalardo, F.; Umana, E.; Xibilia, M.G.; Giamberardino, P. A scalable fractional order model for IPMC actuators. *IFAC Proc. Vol.* **2012**, *45*, 593–596. [CrossRef]
35. Caponetto, R.; Graziani, S.; Pappalardo, F.; Xibilia, M.G. Parametric Control of IPMC Actuator Modeled as Fractional Order System. *Adv. Sci. Technol.* **2013**, *79*, 63–68.
36. 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: Cambridge, MA, USA, 1998; pp. 41–119.
37. Caputo, M. Linear Models of dissipation whose Q is almost frequency independent. *Geophys. J. R. Astr. Soc.* **1967**, *13*, 529–539. [CrossRef]
38. Brualdi, R.A. *Introductory Combinatorics*, 5th ed.; Pearson Education: New York, NY, USA, 2009; pp. 127–160.
39. Xue, D.Y.; Bai, L. Numerical algorithms for Caputo fractional-order differential equations. *Int. J. Control* **2017**, *90*, 1201–1211. [CrossRef]
40. Sabatier, J.; Farges, C. Analysis of fractional models physical consistency. *J. Vib. Control* **2017**, *23*, 895–908. [CrossRef]
41. Valsa, J.; Brancik, L. Approximate formulae for numerical inversion of Laplace transforms. *Int. J. Numer. Model.* **1998**, *11*, 153–166. [CrossRef]