Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2011
August 28-31, 2011, Washington, DC, USA

DETC2011-47408

HIGH SPEED ALGORITHM FOR COMPUTATION OF FRACTIONAL DIFFERENTIATION AND INTEGRATION

Masataka Fukunaga
College of Engineering
Nihon University (P.T. Lecturer)
(home)1-2-35-405, Katahira, Aoba-ku,
Sendai, 980-0812, Japan
Email: fukunaga@apple.ifnet.or.jp

Nobuyuki Shimizu*
Department of Mechanical Systems
and Design Engineering
Iwaki Meisei University
Iwaki, 970-8551
Japan
e-mail: nshim@iwakimu.ac.jp

ABSTRACT

A high speed algorithm for computing fractional differentiations and fractional integrations in fractional differential equations is proposed. In this algorithm the stored data is not the history of the function to be differentiated or integrated but the history of the weighted integrals of the function. It is shown that, by the computational method based on the new algorithm, the integration time only increases in proportion to $n \log n$*, different from* n^2 *by a standard method, for n steps of integrations of a differential integration.*

INTRODUCTION

Fractional differentiations and integrations are now being widely applied to various fields such as physics, industry, and other areas [1, 2]. Many systems are described by differential equations that include fractional differentiations and fractional integrations. The $q(>0)$th order Riemann-Liouville definition for fractional time differentiation of a function $f(t)$ is given by [1, 3, 4]

$$D_a^q f(t) = \frac{d^{n_q}}{dt^{n_q}} \int_a^t \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau) d\tau, \tag{1}$$

where the integer n_q satisfies the inequality, $n_q - 1 \leq q < n_q$, and $\Gamma(z)$ is the gamma function.

Some of fractional differential equations (FEs) can be solved analytically with the use of integration transformation methods including Laplace transformations and the Fourier transformation. However, in many cases numerical solutions by difference methods are necessary, especially for nonlinear FEs. At this point we come to the difficulty inherent to fractional differentiation. We need all the history of $f(\tau)$ between the time a to the current time t to calculate the fractional derivative of $f(t)$, since the fractional derivative of $f(t)$ includes the integration of $f(\tau)$ from a to t. Owing to this property, the computing time increases in proportion to n^2 in order to solve an FE for n time steps by a difference method. This property limits researchers from performing large size the FEs such as the dynamics of continuous media that include fractional derivatives.

In order to avoid this difficulty which comes from the definition equation, Yuan and Agrawal ([5, 6]) proposed a method. It is well known that the fractional differentiation is derived as a special limit of the generalized Maxwell model. Thus, they approximate the fractional derivative by a set of simultaneous first order differential equations with adequately distributed decay constants. Singh and Chatterjee ([7]) adopted this line with a somewhat different concept. However, it is necessary to have a large number of first order equations with a wide range of decay constants in order to solve FEs for large t.

*Address all correspondence to this author.

Copyright © 2011 by ASME

There is another method to calculate fractional derivatives. Ford and Simpson [8] proposed exponentially increasing time steps based on the scaling property of the kernel of fractional derivatives. However, their method is not applicable to oscillating problems, since the function $f(t)$ is also scaled with extending time steps. For oscillating problems the interval for $f(\tau)$ stored as memory in eq. (1) must be smaller than the period of oscillation. This is one of the reasons why $f(\tau)$ must be stored as memory with a small interval.

In this paper we propose a calculation method of eq. (1) in FEs with a small number of data points. It will be shown that if weighted integrals of $f(\tau)$ by τ^k, $k = 0,1,2\ldots$, is used as memory, the time interval to store the memory can be increased even greater than the period of oscillation. As a consequence the data points in the calculation of eq. (1) can be drastically reduced.

In chapter 2 we explain how eq. (1) is expressed by the weighted integrals of $f(t)$. The crucial point is to separate the kernel $(t-\tau)^{n_q-q-1}$ in eq. (1) into the product of functions of t and functions of τ. An example of calculating the weighted integral of $f(\tau)$ is given in chapter 3. It will be shown that the computing time for solving FEs for n steps increases in proportion to $n\log n$, if the new method for calculating eq. (1) is used. Two FEs are solved in Chapter 4 to demonstrate the validity of the new method.

POSSIBILITY OF $n\log n$ ALGORITHM FOR FES

In this chapter we will explain the idea of decreasing the memory in calculating fractional derivatives. This is accomplished by decreasing the divisions in calculating the integral in eq. (1). We show one possibility of enlarging the interval of integration depending on the distance from the current time t. This idea is similar to that of Ref. [8]. However, the present model is different from their model in the sense that the scaling property of the kernel is not employed. The basic idea is that weighted integrals of $f(\tau)$ are stored as the memory in eq. (1) instead of $f(\tau)$ itself.

For this purpose the variable τ in the kernel $(t-\tau)^{n_q-q-1}$ has to be separated from t. Such separation is possible as will be shown below. Equation (1) is divided into two parts by a point c in the interval (a,t) as

$$D_a^q f(t) = \frac{d^{n_q}}{dt^{n_q}} \int_a^c \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau + \frac{d^{n_q}}{dt^{n_q}} \int_c^t \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau. \tag{2}$$

The point c varies with t. Therefore, the derivative of the integral of the first term on the RHS in eq. (2) with respect to t consists of the derivative of integrand and the derivative of the upper limit

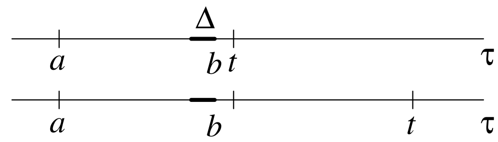

FIGURE 1. τ IN THE INTEGRAL AT TWO CURRENT TIMES t. The thick lines show the same time interval counted from the time a.

of integration. However, the latter is precisely cancelled out with the derivative of the lower limit of integration of the second term on the RHS of eq. (2) (see Appendix A). Therefore, we may treat c as a constant in calculating eq. (2).

Based on the discussion given above, the first term on the RHS of eq. (2) denoted by $I_c f(t)$ can be replaced by the following integral.

$$I_c f(t) = \int_a^c \frac{d^{n_q}}{dt^{n_q}} \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau = \int_a^c \frac{(t-\tau)^{-1-q}}{\Gamma(-q)} f(\tau)d\tau. \tag{3}$$

In the classical method of numerical integration the interval $[a,c]$ is divided into sufficiently small intervals. One such small interval is given by, $b \in (a,c]$ and $\Delta > 0$,

$$I_\Delta f(b) = \int_{b-\Delta}^b \frac{(t-\tau)^{-1-q}}{\Gamma(-q)} f(\tau)d\tau, \tag{4}$$

where $b-\Delta \geq 0$. Then the integrand is replaced by an approximated form with the use of, e.g. the trapezoidal rule. However, this method needs an increasing amount of data points in proportion to the computing time.

There is another method of calculating eq. (4) in FEs. The kernel, $(t-\tau)^{-1-q}$, in the integral is expanded around b as

$$\frac{(t-\tau)^{-1-q}}{\Gamma(-q)} = \frac{(t-b+b-\tau)^{-1-q}}{\Gamma(-q)} = \sum_{k=0}^{\infty} \frac{(t-b)^{-1-q}}{k!\Gamma(-q-k)} \left(\frac{b-\tau}{t-b}\right)^k. \tag{5}$$

This expression converges, if $|\tau-b| < t-b$. Substituting eq. (5) into eq. (4) and writing $\tau = b-u$, we have

$$I_\Delta f(b) = \int_0^\Delta \sum_{k=0}^{\infty} \frac{(t-b)^{-q-k-1}}{k!\Gamma(-q-k)} u^k f(b-u)du. \tag{6}$$

Copyright © 2011 by ASME

Since the sum with respect to k in eq. (6) converges absolutely, the order of sum and integration is changeable. It should be noted that the integration of eq. (6) with respect to u has been separated from t. Therefore the factor $(t-b)^{-q-k-1}$ can be moved out of the integration as

$$I_\Delta f(b) = \sum_{k=0}^{\infty} \frac{(t-b)^{-q-k-1}}{k!\Gamma(-q-k)} \int_0^\Delta u^k f(b-u)du. \qquad (7)$$

Equation (7) shows that the integral $I_c f(t)$ in eq. (3) is calculated through the weighted integrals $I_{\Delta,k} f(b)$ defined by

$$I_{\Delta,k} f(b) = \int_0^\Delta u^k f(b-u)du, \qquad k = 0,1,2,\ldots. \qquad (8)$$

It is also noted that $I_{\Delta,k} f(b)$ does not include t. Therefore, $I_{\Delta,k} f(b)$ can be calculated whenever t exceeds b (see Fig. 1). The width of interval Δ is limited by the convergency condition of the series in eq. (7) as

$$\Delta < t-b. \qquad (9)$$

Equation (7) is not practical for numerical calculation of eq. (3), since it includes infinite sum. Therefore the sum in eq. (7) is truncated at $k = K$ as

$$I_\Delta^K f(b) = \sum_{k=0}^{K} \frac{(t-b)^{-q-k-1}}{k!\Gamma(-q-k)} I_{\Delta,k} f(b). \qquad (10)$$

Thus, $I_\Delta^K f(b)$ gives an approximation of $I_\Delta f(b)$. The truncation error is estimated to be

$$|I_\Delta f(b) - I_\Delta^K f(b)|/\|f(\tau)\| = O([\Delta/(t-b)]^{K+1}), \qquad (11)$$

where $\|f(\tau)\|$ is the upper limit of $|f(\tau)|$, and $O(\cdot)$ is the Landau's symbol.

The width Δ of integration in eq. (8) can be chosen arbitrarily, if condition (9) is met. The width Δ can be larger than the period of oscillation for oscillatory problems without loss of accuracy. Now the number of divisions for integration is estimated. For concreteness of this discussion the value $\Delta/(t-b) = \varepsilon < 1$ is fixed. Thus Δ increases with $\varepsilon(t-b)$. The number of divisions per unit length of τ is thus $\simeq 1/\varepsilon(t-b)$. The total number of divisions is derived by the integration of the number per unit length of τ as $\sim (1/\varepsilon)\log(t-a)$ for large t. Thus the number of integrations $I_{\Delta,k} f(b)$ necessary to calculate $I_c f(t)$ increases logarithmically with $\log(t-a)$. In the next chapter we will give an algorithm for calculating $I_{\Delta,k} f(b)$ in eqs. (8) and (10) for solving FEs.

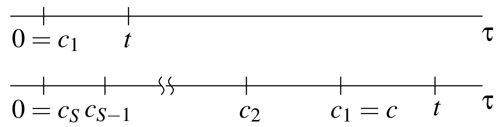

FIGURE 2. DIVISION TO c_s AT TWO CURRENT TIMES t's. The upper line shows when $2h/(t-2h) > \varepsilon$, whereas the lower line shows when $2h/(t-2h) < \varepsilon$. The width of the interval $(c_{s+1}, c_s]$ increases as 2^s with s.

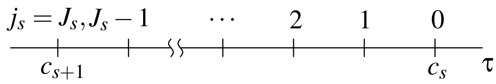

FIGURE 3. DIVISION OF THE sTH INTERVAL $(c_{s+1}, c_s]$ INTO SUBINTERVALS OF WIFTH Δ_s.

ALGORITHM OF DIFFERENTIATION

In this chapter we demonstrate a method of calculating eq. (8) and hence eq. (7) and eq. (3). We also show that the memory necessary to calculate fractional differentiation at the nth step is reduced to $\log n$. The form of eq. (3) shows that the method is directly applicable to fractional integrations by allowing negative q.

Algorithm of Storing Memory

The interval $[0,t]$ in eq. (1) with $a = 0$ is divided into intervals $(c_{s+1}, c_s]$, $s = 1,2,\ldots,S-1$, and $(c_1, t]$ for $s = 0$, where $c_0 = t$ (see Fig. 2). These intervals are labeled by the right end point of the intervals as the sth intervals. The point c_1 is identified with c. The label c_S is fixed to $c_S = 0$. These intervals are further divided into smaller subintervals of a width $\Delta_s = 2^s h$. The width of the subinterval is $\Delta_0 = h$ in the 0th interval $(c_1, t]$. The number of subintervals in the sth interval is denoted by J_s (see Fig. 3).

In the 0th interval the values of $f(\tau)$ are stored in memory at the points $j = 0,1,2,\ldots,J_0-1$, and $j = J_0$, where $t - J_0 h = c_1$. As for the sth interval, $s = 1,2,\ldots,S-1$, the integrations on the subintervals given by eq. (8) are stored as memory, where Δ is replaced by Δ_s. Let the width of the subinterval satisfy the following condition

$$\frac{\Delta_s}{t-c_s} < \varepsilon, \qquad 0 < \varepsilon < 1, \quad s = 1,2,\ldots,S-1. \qquad (12)$$

Copyright © 2011 by ASME

Then the integration (3) with $a=0$ is given by

$$I_c f(t) = \sum_{s=1}^{S-1} \left\{ \sum_{j=0}^{J_s-1} \sum_{k=0}^{\infty} \frac{(t-c_s+j\Delta_s)^{-q-k-1}}{k!\Gamma(-q-k)} I_{s,k} f(c_s - j\Delta_s) \right\}, \tag{13}$$

where $I_{s,k}f(c_s - j\Delta_s)$ is defined by eq. (8). However, the notation is changed for the present purpose:

$$I_{s,k} f(b) = \int_0^{\Delta_s} u^k f(b-u)du, \qquad k = 0,1,2,\ldots. \tag{14}$$

In eq. (13) the sum in the brace gives the integral over the interval $(c_{s+1}, c_s]$.

If the summation for k is truncated at $k=K$, the sum

$$I_c^K f(t) = \sum_{s=1}^{S-1} \left\{ \sum_{j=0}^{J_s-1} \sum_{k=0}^{K} \frac{(t-c_s+j\Delta_s)^{-q-k-1}}{k!\Gamma(-q-k)} I_{s,k} f(c_s - j\Delta_s) \right\} \tag{15}$$

gives an approximation of $I_c f(t)$. The error of $I_c^K f(t)$ is estimated to be

$$|I_c f(t) - I_c^K f(t)|/L = O(\varepsilon^{K+1}), \tag{16}$$

where L is the upper limit of $|f(t)|$, and ε is defined by eq. (12).

In the rest of this section we show a method of calculating the integrals $I_{s,k}f(b)$ given by eq. (14), where the intervals of integration are located far from the current time t. The procedure consists of four steps. We also utilize the fact that the time $\tau = b$ occurs once in the neighborhood of t (see Fig. 1). We define a reference number $M>0$ as the smallest even integer that satisfies the inequality, $2/M < \varepsilon$.

(i) For the early steps of solving an FE, in which the current time t satisfies the condition, $2h/(t-2h) \geq \varepsilon$, (or $J_0 < M+2$), any conventional scheme of fractional differentiation with a constant time step h may be adopted. At this stage $c_S = c_1 = 0$.

(ii) Let t be firstly reached to satisfy the condition $2h/(t-2h) < \varepsilon$, (or $J_0 = M+2$) (see Fig. 4(a)). At this point the following integral is calculated with use of $f(\tau)$ at $\tau = 0, h$, and $2h$ (or $j_0 = J_0, J_0-1$, and J_0-2):

$$I_{1,k} f(2h) = \int_0^{2h} u^k f(2h-u)du, \quad k = 0,1,2,\ldots. \tag{17}$$

Then we replace the labels as

$$\begin{aligned} &c_1 := 2h, \qquad S := S+1, \quad c_S = 0, \\ &I_{1,k}f(c_1) := I_{1,k}f(2h). \end{aligned} \tag{18}$$

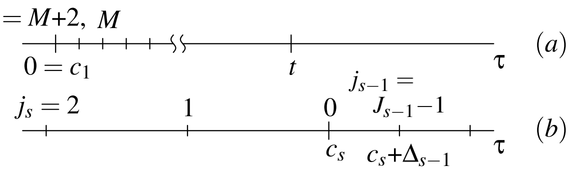

FIGURE 4. DIVISION OF τ INTO SUBINTERVALS. The upper line shows at $t = (M+2)h$ when t firstly satisfies the condition, $2h/(t-2h) > \varepsilon$, whereas the lower line shows around $\tau = c_s$ at arbitrary time $t(> c_1)$.

In the above expressions the symbol $(:=)$ means substitution of the RHS into the LHS. The number J_0 decreases by 2, i.e. $J_0 := M$ and $J_1 := 1$.

After t exceeds $(M+2)h$, the values of $f(\tau) = f(t-jh)$, $j = 0,1,\ldots,J_0$, in the 0th interval are stored in memory. These data are used to calculate the second term on the RHS of eq. (2) in a usual method. They are also used to calculate the integrals $I_{s,k}f(c_s - j\Delta_s)$, $s = 1,2,\cdots,S-1$, $j = 0,1,\ldots,J_s-1$. The integral $I_{s,k}f(c_s - j\Delta_s)$ is calculated in two ways. For the 1st interval the method is similar to that given in (ii) as will be shown in (iii). The integral $I_{1,k}f(c_1 - j\Delta_1)$ is calculated at $\tau = c_1$ with use of $f(\tau)$ in the 0th interval. As for the intervals of $s \geq 2$, the integrals $I_{s,k}f(c_s - j\Delta_s)$ are calculated with use of the integrals on the $(s-1)$-th intervals (see (iv) below).

(iii) Let t be reached to satisfy the condition, $2h/(t-c_1-2h) < \varepsilon$, (or $J_0 = M+2$) (see Fig. 4(b) with $s=1$). At this stage the following integral is calculated with use of $f(\tau)$ at $\tau = c_1, c_1+h$, and c_1+2h, (or at $j_0 = M+2, M+1$, and M, or at $j_0 = J_0, J_0-1$, and J_0-2).

$$I_{1,k} f(c_1+2h) = \int_0^{2h} u^k f(c_1+2h-u)du, \quad k = 0,1,2,\ldots. \tag{19}$$

The labels in the 0th and the first intervals are replaced by

$$\begin{aligned} &c_1 := c_1 + 2h, \\ &I_{1,k}f(c_1 - j\Delta_1) := I_{1,k}f(c_1 - (j-1)\Delta_1), \quad j = 1,2,\ldots,J_1, \\ &I_{1,k}f(c_1) := I_{1,k}f(c_1+2h), \\ &J_1 := J_1+1, \qquad J_0 := J_0 - 2. \end{aligned} \tag{20}$$

The last two substitutions for J_s, $s = 0,1$, reflect the change of the number of subintervals by the conversion of the data in the 0th interval to the data in the first interval.

(iv) As for the sth interval $(c_{s+1}, c_s]$, $s = 2,3,\ldots,S-1$, and the point $\tau = 0$ for $s = S$, the integrals $I_{s,k}f(c_s - j\Delta_s)$, $j =$

Copyright © 2011 by ASME

$0, 1, \ldots, J_s - 1$ can be calculated with use of the integrals in the $(s-1)$th interval. Let the current time t satisfy the condition $\Delta_s/(t - c_s - \Delta_s) < \varepsilon$, (or $J_{s-1} = M/2 + 2$) [see Fig. 4(b)]. At this point the following integral is calculated.

$$\begin{aligned} I_{s,k} f(c_s + \Delta_s) &= I_{s,k} f(c_s + 2\Delta_{s-1}) \\ &= \int_0^{2\Delta_{s-1}} u^k f(c_s + 2\Delta_{s-1} - u) du \end{aligned} \tag{21}$$

The RHS of this expression is calculated with use of the following two integrals in the $(s-1)$th interval (see Fig. 4(b)).

$$I_{s-1,k} f(c_s + \Delta_{s-1}) = I_{s-1,k} f(c_{s-1} - (J_{s-1} - 1)\Delta_{s-1}) \tag{22}$$

$$I_{s-1,k} f(c_s + 2\Delta_{s-1}) = I_{s-1,k} f(c_{s-1} - (J_{s-1} - 2)\Delta_{s-1}) \tag{23}$$

as

$$\begin{aligned} & I_{s,k} f(c_s + 2\Delta_{s-1}) \\ & = I_{s-1,k} f(c_{s-1} - (J_{s-1} - 2)\Delta_{s-1}) \\ & \quad + \sum_{r=0}^{k} \binom{k}{r} (\Delta_{s-1})^{k-r} I_{s-1,r} f(c_{s-1} - (J_{s-1} - 1)\Delta_{s-1}). \end{aligned} \tag{24}$$

The proof is given in Appendix B. When eq. (21) is calculated, the labels in the sth interval are replaced by:

$$\begin{aligned} & c_s := c_s + \Delta_s, \\ & I_{s,k} f(c_s - j\Delta_s) := I_{s,k} f(c_s - (j-1)\Delta_s), \quad j = 1, 2, \ldots, J_s. \\ & I_{s,k} f(c_s) := I_{s,k} f(c_s + \Delta_s) \\ & J_s := J_s + 1, \quad J_{s-1} := J_{s-1} - 2 = M/2, \end{aligned} \tag{25}$$

where the relation $\Delta_s = 2\Delta_{s-1}$ is used. The label J_s is increased by 1, while J_{s-1} is decreased by 2 so that $J_{s-1} = M/2$. If $s = S$, the subscript for label $\tau = 0$ is replaced with $S+1$, i.e. $c_{S+1} = 0$.

The steps (i) - (iv) have completed the calculation of $I_{s,k} f(c_s - j\Delta_s)$ in eqs. (13) and (15) for $j = 0, 1, 2, \ldots, J_s - 1$ and $s = 1, 2, \ldots, S-1$ with use of the data $f(\tau) = f(t - jh)$, $j = 0, 1, \ldots, J_0$, that are originally in the 0th interval.

Precision

In the method of the previous section, the width Δ_s of the subintervals satisfies the condition $\Delta_s/(t - c_s) < \varepsilon$ for $s = 1, 2 \ldots, S-1$. The proof is as follows. As can be seen in the steps (ii) and (iii), the width Δ_1 in the first interval satisfies inequality $\Delta_1/(Mh) = 2h/(Mh) < \varepsilon$. Therefore, in the first interval Δ_1 satisfies the condition $\Delta_1/(t - \tau) < \varepsilon$, since $t - \tau \geq Mh$.

The integrals $I_{s,k} f(c_s)$ for $s > 1$ is calculated with use of old $I_{s-1,k}(c_s)$ and $I_{s-1,k} f(c_s - \Delta_{s-1})$ in step (iv), where c_s is shifted by $+\Delta_s$ from that in eqs. (21), (22), and (23) [see eq. (25)]. The value of c_s changes with the computing time $t = nh$, where n is the time step. However, the minimum number of subintervals in the sth interval is $M/2$ by the assumption, whereas the width of subinterval is $\Delta_s = 2^s h$. Thus the minimum of $t - c_s$ is estimated to be

$$Mh + \sum_{k=1}^{s-1} (M/2) \cdot 2^k h = Mh \cdot 2^{s-1}. \tag{26}$$

Therefore, the ratio $\Delta_s/(t - c_s)$ is estimated to be less or equal to $2/M$ which is less than ε.

Memory for Calculation of Fractional Derivatives

We have demonstrated in the previous sections an algorithm that reduces the data points and the total number of data necessary to calculate fractional derivatives. For simplicity of explanation the data were combined every two time steps. However, this may not give the most effective timing to combine the data. In this section we show that the number of data can be further reduced. In the step (iv), the integral data at two subintervals in the $(s-1)$th interval are combined to one $I_{s,k} f(c_s)$ in the sth interval. It will be shown that this is the most memory saving method compared with the method in which integrals of more than two subintervals are combined to $I_{s,k} f(c_s)$. In steps (ii) and (iii), every two steps of $f(\tau)$ are combined to $I_{1,k} f(c_1)$ at c_1 with $t - c_1 = Mh$. However, for $K > 0$, the amount of data is reduced, if c is identified with a point that gives a larger $t - c_1$ than that given above. In other words, if $f(\tau)$ of m steps are combined to the integral $I_{1,k} f(c_1)$ at c_1 preserving the condition $m/M < \varepsilon$, then, the minimum of total data occurs at m larger than 2.

In order to prove the statement given above, we replace the conditions adopted in steps (ii) and (iii) with the followings. Let $m \geq 2$ be an integer, and let the integer M be a multiple of m. When the current time t (or J_0) satisfies the condition, $J_0 = M + m$, $m \geq 2$, we calculate $I_{1,k}(c_1 + mh)$ with use of $f(\tau)$ at $c_1 + (j-1)h$, $j = 0, 1, \ldots, m$. We also replace the condition in step (iv) with the following: If J_{s-1} at the $(s-1)$th interval satisfies $J_{s-1} = m_1 + M/m$, $m_1 \geq 2$, then the integrals at m_1 subintervals, $I_{s-1,k}(c_s + j\Delta_{s-1})$, $j = 1, 2, \ldots, m_1$, $k = 0, 1, \ldots, K$, are combined to the integral $I_{s,k}(c_s + \Delta_s)$ for the new subinterval in the sth interval. The method of combining the integrals is similar to that given in section 3.1 taking into account the number of combined data.

In these procedures we impose a condition in order to fix the precision of integral (3). The ratio of the width Δ_s of subintervals to the minimum of $(t - c_s)$ has to satisfy the following condition

$$\Delta_s/(t - c_s)_{min} \leq m/M, \quad s \geq 1. \tag{27}$$

Copyright © 2011 by ASME

This condition is satisfied if the minimum of $t-c_s$ is given by

$$(t-c_s)_{min} = m_1^{s-1} Mh, \qquad s \geq 1. \tag{28}$$

The proof is as follows. The width of the subinterval Δ_s in the sth interval is $\Delta_s = m m_1^{s-1} h$ for $s \geq 1$, since $f(\tau)$'s at the end of m intervals are combined to one integrals at c_1, and integrals of m_1 intervals are combined to one integrals at c_i, $i = 2, \ldots, s$. Thus eq. (27) is derived.

The necessary data points for computation at $t = c_s$ is given by the following considerations. The minimum width of the rth interval is $(m_1^{r+1} - m_1^r)Mh$ for $r = 1, 2, \ldots, s-1$, while the width of the subinterval is $\Delta_r = m m_1^r h$. Thus the minimum number of subintervals of the rth interval are $(m_1-1)M/m$ independent of r. The necessary number of subintervals is $(m_1-1)M/m + m_1$ for the rth interval of $r > 0$, while that in the 0th is $M+m+1$ from the consideration of combining the data. The total amount of data, N_{mem}, necessary for the Kth approximation is given by

$$N_{\mathrm{mem}} = (M+m+1) + (K+1)(s-1)\left[\frac{(m_1-1)M}{m} + m_1\right]. \tag{29}$$

The total number of steps n to calculate a fractional derivative to the time $t = c_s$ is given by $n = m_1^{s-1} M$ from eq. (28), from which we have

$$s-1 = \frac{\log_2(n/M)}{\log_2 m_1}. \tag{30}$$

Introducing the last expression into eq. (29), we have

$$\begin{aligned} N_{\mathrm{mem}} &= (M+m+1) + (K+1)\frac{\log_2(n/M)}{\log_2 m_1} \cdot \left[\frac{(m_1-1)M}{m} + m_1\right] \\ &= (m+1) + \frac{M}{m}\left[m + (K+1)\frac{m_1-1}{\log_2 m_1} \cdot \log_2\left(\frac{n\varepsilon_0}{m}\right)\right] \\ &\quad + (K+1)\frac{m_1}{\log_2 m_1} \cdot \log_2\left(\frac{n\varepsilon_0}{m}\right), \end{aligned} \tag{31}$$

where $\varepsilon_0 = m/M$ is a fixed value. First we derive the value of m_1 that minimizes N_{mem} for fixed m. The ratios $(m_1)/\log_2 m_1$ and $(m_1-1)/\log_2 m_1$ increase monotonically for $m_1 \geq 2$. Therefore, the total number of data is the minimum at $m_1 = 2$ for fixed m.

Thus, m_1 in eq. (31) is substituted with $m_1 = 2$.

$$\begin{aligned} N_{\mathrm{mem}} &= m + 1 + \frac{1}{\varepsilon_0}\left[m - (K+1)\log_2\left(\frac{m}{n\varepsilon_0}\right)\right] \\ &\quad -2(K+1)\log_2\left(\frac{m}{n\varepsilon_0}\right) \end{aligned} \tag{32}$$

In order to find m for minimum of N_{mem}, this expression is differentiated with respect to m under the assumption that m is real:

$$\frac{d(N_{\mathrm{mem}})}{dm} = \frac{1}{\varepsilon_0}\left[1 + \varepsilon_0 - (1+2\varepsilon_0)\frac{K+1}{\log_e 2}\frac{1}{m}\right], \tag{33}$$

that vanishes at

$$m_{min} = \frac{1+2\varepsilon_0}{1+\varepsilon_0}\frac{K+1}{\log_e 2} = 1.4427 \cdot \frac{1+2\varepsilon_0}{1+\varepsilon_0}(K+1). \tag{34}$$

Thus, N_{mem} is minimum when the integer m is nearest to m_{min}.

The amount of saved data is estimated as follows. We write $m = 2^{\nu+1}$ and $M = M_2$ for $m = 2$, where ν has the meaning of the reduced number of intervals by adopting m subintervals for integration of $f(\tau)$ at c_1 instead of $m = 2$. It is natural to assume that the equality $M/m = M_2/2$ is satisfied. This equality ensures the same accuracy of integration among different m. Thus, we have $M = M_2 \cdot 2^\nu$. For $m = 2$, the total number of data is given by

$$(N_{\mathrm{mem}})_2 = (M_2+3) + (K+1)(s-1)\left(\frac{M_2}{2} + 2\right), \tag{35}$$

from eq. (29) with $m_1 = 2$. The total number of data for arbitrary m necessary to calculate fractional derivatives is given by

$$(N_{\mathrm{mem}})_m = (M+m+1) + (K+1)(s-\nu-1)\left(\frac{M}{m} + 2\right). \tag{36}$$

The difference between the two expressions using of $M/m = M_2/2$ is given by

$$\begin{aligned} &(N_{\mathrm{mem}})_2 - (N_{\mathrm{mem}})_m \\ &= \nu(K+1)\left(\frac{M_2}{2} + 2\right) - \left(\frac{m}{2} - 1\right)M_2 - (m-2) \\ &= \nu(K+1) + [\nu(K+1) - (m-2)](\frac{M_2}{2} + 1). \end{aligned} \tag{37}$$

The difference depends on K, m, and $M_2/2$, while it is independent of the time steps.

Finally we estimate the total number of data for fixed precision. Let the precision ε_T be fixed as

$$\left(\frac{m}{M}\right)^{K+1} < \varepsilon_T. \tag{38}$$

Copyright © 2011 by ASME

Let K_{ε_T} be the smallest integer that satisfies this condition, i.e.

$$K_{\varepsilon_T}+1>\frac{\log(1/\varepsilon_T)}{\log(M/m)}. \tag{39}$$

When the RHS of this expression is substituted into eq. (36), the least number of data for fixed precision is estimated to be

$$\begin{aligned}\text{Least Number of Data} = {} & (K_{\varepsilon_T}+1)(s-\nu-1)(M+4)/2 \\ & +(M+m+1)\end{aligned} \tag{40}$$

at $t=c_s$.

Equation (31) with $m_1=2$ is rearranged to show that the total number of data increases logarithmically with $n=2^{s-1}M$ for $n/M\gg 1$:

$$N\text{mem}=(M+m+1)+(K+1)\left(\frac{M}{m}+2\right)\log_2\left(\frac{n}{M}\right). \tag{41}$$

The total data necessary to calculate the derivative is reduced to $\simeq 1/3.4$ at the 1280th time step ($s=5$), $\simeq 1/18$ at the $10,240$th time step ($s=8$), and be reduced to $\simeq 1/200$ at the $160,000$th time step ($s=12$) for $K=2$, $m=2$, and $M=40$.

NUMERICAL EXAMPLES BY NEW ALGORITHM

In this chapter two examples of FEs are solved by means of the new algorithm of fractional differentiation. The approximated eq. (15) is adopted for the first term in eq. (2). The second term in eq. (2) is written as

$$D_c^q f(t)=\frac{d^{n_q}}{dt^{n_q}}I_c^{n_q-q}f(t), \tag{42}$$

where $I_c^{n_q-q}f(t)$ is the (n_q-q)th order fractional integration of $f(t)$ defined by

$$I_c^{n_q-q}f(t)=\int_c^t\frac{(t-\tau)^{n_q-q-1}}{\Gamma(n-q)}f(\tau)d\tau. \tag{43}$$

For $0<q<1$, the fractional differentiation by the central difference method is given by

$$D_c^q f(t-h/2)=\frac{1}{h}\left[I_c^{1-q}f(t)-I_c^{1-q}f(t-h)\right]. \tag{44}$$

The numerical integration of $I_c^{1-q}f(t)$ is calculated by the trapezoidal rule. Equation (44) gives the derivative at $t-h/2$. Thus, interpolation or extrapolation is necessary to find the derivatives at $t-h$ or at t. We adopt $K=2$ for eq. (15), and $M=40$ and $m=2$ for the division of the second term of eq. (2).

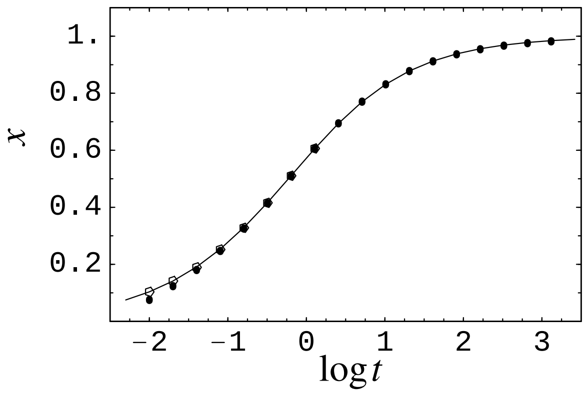

FIGURE 5. SOLUTION OF EQ. (45) WITH $q=1/2$ AND $f=k=1$. The time step is $h=0.01$. The dotted line shows the solution by the difference method. The open circles show the solution by the difference method in which the solutions at first two steps are obtained analytically. The solid line show the analytic solution.

The fractional Voigt equation

First we solve the fractional Voigt equation defined by

$$D_0^q x(t)+kx(t)=f(t) \tag{45}$$

for $0<q<1$ with the initial condition

$$x(t)=0,\qquad t\le 0. \tag{46}$$

The lower terminal $a<0$ of fractional differential equation is replaced by $a=0$ due to the initial condition. The Riemann-Liouville definition (1) can be replaced by the Caputo definition due also to the initial condition given by eq. (46), $D_0^q x(t)=I_0^{1-q}Dx(t)$. Thus eq. (45) is rewritten as

$$I_0^{1-q}Dx(t)=f(t)-kx(t). \tag{47}$$

Differentiating $(1-q)$ times on both sideS of equation, we have

$$Dx(t)=D_0^{1-q}[f(t)-kx(t)], \tag{48}$$

where the RHS is the $(1-q)$th order Riemann-Liouville derivative of $(f-kx)$. Equation (48) will be solved for $q=1/2$ and $k=f(t)=1$.

In Fig. 5 the solution by the difference method is compared with the analytic solution that is given by

$$x_{ana}(t)=1-\sum_{j=0}^{\infty}\frac{(-kt^q)^j}{\Gamma(qj+1)}. \tag{49}$$

Copyright © 2011 by ASME

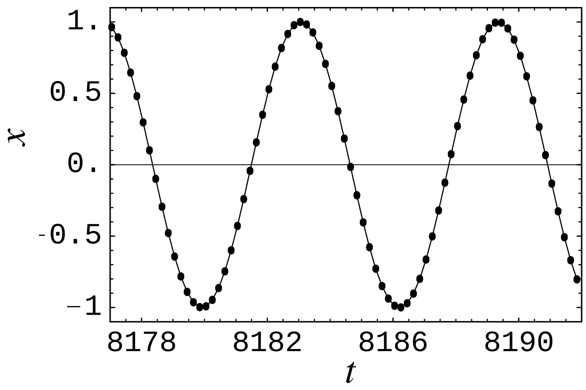

FIGURE 6. SOLUTION OF EQ. (50) WITH $q = 1/2$ AND $\mu = k = f_0 = \omega = 1$ AT AROUND 1300 PERIOD OF OSCILLATIONS. The time step is 0.05. The dotted line show the solution by the difference method, while the solid line shows the asymptotic analytic solution.

over $2^{17} \sim 130000$ time steps with $h = 0.01$. The numerical solution (the dotted line) does not reproduce well the analytic solution (the solid curve) for the initial several time steps. The discrepancy comes from the discontinuity of the derivative of $x(t)$ at $t = 0$. The difference method assumes continuity of derivatives. Therefore, the solution does not coincide with the analytic solution at discontinuous points. After several time steps the numerical solution follows the analytic solution. The numerical error is less than 0.5 % excluding the initial several time steps.

In order to reduce the numerical error in the early stage, a method is tried in which the analytic solution is used for the initial two steps. After these steps the equation is solved by the difference method. The open circles in Fig. 5 show the solution by this method. It is shown that the solution coincides well the analytic solutions including the early stage.

The fractional Voigt model with inertia

We give another example,

$$D^2x(t) + \mu D_0^q x(t) + kx(t) = f_0 \sin \omega t \tag{50}$$

with the initial conditions

$$\begin{aligned} &x(t) = 0, \qquad t < 0, \\ &x(0) = 0, \qquad x^{(1)}(0) = 0. \end{aligned} \tag{51}$$

In Fig. 6 the solution by the difference method is plotted by the dotted line for $q = 1/2$ and $\mu = k = f_0 = \omega = 1$ together with the analytic solution by the solid line. The time step is $h = 0.05$, and the solution is given for two periods of oscillation around $t =$ 8190 after 1300 periods of oscillation. The asymptotic form of the analytic solution is derived from the Fourier transformation. Assuming that $f(t) = f_0 e^{i\omega t}$, the solution of the type $x = x_0 e^{i\omega t}$ is obtained:

$$\begin{aligned} &-\omega^2 x + \mu (i\omega)^q x + kx = f, \\ &x = \frac{f}{k - \omega^2 + \mu (i\omega)^q}. \end{aligned} \tag{52}$$

The solution of eq. (50) is obtained from the imaginary part of eq. (52) as the asymptotic solution. The numerical solution coincides well with the analytic solution. The numerical error does not change appreciably throughout the computational time steps.

Since the numerical algorithm of the fractional differentiation is first order, the error changes as h^2. However, the error is at its minimum $\simeq 0.0002$ at $h = 0.05$. The minimum of numerical error comes from the truncation at $K = 2$ of expansion in eq. (15). The numerical error of truncation is estimated from eq. (16) with eq. (38) to be $O(\varepsilon^{K+1}) = O((2/M)^3) \sim 1/8000$ for $M = 40$. Thus the minimum error is consistent with the truncation error.

TABLE 1. COMPARISON OF COMPUTING TIME (sec)

Steps	New Algo.	Trapez.	G.L.
2500	0.062	1.034	0.061
5000	0.137	4.137	0.231
10,000	0.298	16.665	0.906
20,000	0.644	66.260	3.607
40,000	1.386		14.403

In Table 1 the computing times of calculating eq. (50) are compared among algorithms. In the table "New Algo." means the new algorithm, "Trapez." means the use of trapezoidal rule in calculating the integral in eq. (1), and "G.L" means the difference scheme of the Grünwald-Letnikov derivative. The use of trapezoidal rule is time consuming compared with the Grünwald-Letonikov method, since it needs long computing time in calculating power functions in eq. (1). The new algorithm also includes the calculation of power functions (see eq. (14)). Nevertheless, the speed of computation by the new algorithm is faster than that by Grüwald-Letnikov method for larger steps than 2500. The former is ten times faster than the latter at the 20,000th time step. The speed of computation is increased by $\simeq 1.4$ times, if a set of parameters of $M = 20$ and $K = 3$ is adopted preserving the accuracy instead of $M = 40$ and $K = 2$. The speed of computation can be further increased by several times, if the values of power functions in eq. (14) are tabulated. This method is effective for repeating computation in a program, simultane-

Copyright © 2011 by ASME

ous calculations of fractional derivatives in many elements such as in FEM, etc.

REFERENCES

[1] Podlubny, I., 1999. *Fractional differential equations*. Academic Press, New York.

[2] Kilbas, A. A., Srivastava, H. M., and Trujillo, J. J., 2006. *Theory and Applications of Fractional Differential Equations*. Elsevior, Amsterdam, Boston.

[3] Mirror, K. S., and Ross, B., 1993. *An introduction to the fractional calculus and fractional differential equations*. Elsevior, Amsterdam, Boston.

[4] Oldham, K. H., and Spanier, J., 1974. *The Fractional Calculus*. Wiley & Sons, New York.

[5] Yuan, L., and Agrawal, O. P., 2009. "A numerical scheme for initial compliance and creep response of a system". *Journal of Vibration and Acoustics,* ***124***, pp. 321–324.

[6] Agrawal, O. P., 2009. "A numerical scheme for dynamic systems containing fractional derivatives". *Mechanical Research Communications,* ***36***, pp. 444–451.

[7] Singh, S. J., and Chatterjee, A., 2006. "Galerkin projections and finite elements for fractional order derivatives". *Nonlinear Dynamics,* ***45***, pp. 183–226.

[8] Ford, N. J., and Simpson, A. C., 2001. "The numerical solution of fractional differential equations: Speed versus accuracy". *Numerical Algorithms,* ***26***, pp. 333–346.

Appendix A: Treatment of the Upper and Lower Limits of Eq. (2)

The time derivative of the first term on the RHS of eq. (2) is given as follows. The one time derivative is given by

$$\begin{aligned}&\frac{d}{dt}\int_a^c \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau \\ &= \int_a^c \frac{d}{dt}\frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau + \frac{dc}{dt}\frac{(t-c)^{n_q-q-1}}{\Gamma(n_q-q)} f(c). \end{aligned} \tag{53}$$

In order to calculate the contribution of the lower limit of integration of the second term on the RHS eq. (2), the integration is separated into two parts by a fixed point b.

$$\begin{aligned}&\frac{d^{n_q}}{dt^{n_q}}\int_c^t \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau \\ &= \frac{d^{n_q}}{dt^{n_q}}\int_c^b \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau \\ &\quad + \frac{d^{n_q}}{dt^{n_q}}\int_b^t \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau. \end{aligned} \tag{54}$$

The derivative of the first term on the RHS of this expression is given by

$$\begin{aligned}&\frac{d}{dt}\int_c^b \frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau \\ &= \int_c^b \frac{d}{dt}\frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau - \frac{dc}{dt}\frac{(t-c)^{n_q-q-1}}{\Gamma(n_q-q)} f(c). \end{aligned} \tag{55}$$

The second term on the RHS of eq. (53) is cancelled out by the first term on the RHS of eq. (55).

The higher order derivatives of eqs. (53) and (55) are also composed of the derivatives of the integrands and the derivatives of the upper limits and the lower limits of the integrals. However, all the derivatives of the upper and lower limits of the integrals are cancelled out. Thus the remaining contributions of the n_qth derivatives are

$$\int_a^c \frac{d^{n_q}}{dt^{n_q}}\frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau, \tag{56}$$

and

$$\int_c^b \frac{d^{n_q}}{dt^{n_q}}\frac{(t-\tau)^{n_q-q-1}}{\Gamma(n_q-q)} f(\tau)d\tau, \tag{57}$$

respectively. Equation (56) is reduced to $I_c f(t)$ given by eq. (3). If c is constant, the derivative in eq. (57) can be moved out of the integral. Thus, $D_a^q f(t)$ can be divided into two terms by t-dependent on c as if c were a constant.

Appendix B: Derivation of Eq. (24)

Equation (24) is calculated with use of eqs. (22) and (23). For this purpose eq. (21) is divided into two by the fact $\Delta_s = 2\Delta_{s-1}$.

$$\begin{aligned} I_{s,k} f(c_s+\Delta_s) &= \int_0^{\Delta_{s-1}} u^k f(c_s+\Delta_s-u)du \\ &\quad + \int_{\Delta_{s-1}}^{2\Delta_{s-1}} u^k f(c_s+\Delta_s-u)du. \end{aligned} \tag{58}$$

The first term on the RHS is given by eq. (23) as

$$\begin{aligned}&\int_0^{\Delta_{s-1}} u^k f(c_s+\Delta_s-u)du \\ &= \int_0^{\Delta_{s-1}} u^k f(c_s+2\Delta_{s-1}-u)du \\ &= I_{s-1,k} f(c_{s-1}-(J_{s-1}-2)\Delta_{s-1}) \end{aligned} \tag{59}$$

Copyright © 2011 by ASME

As for the second term on the RHS of eq. (21), the variable u is replaced by $u+\Delta_{s-1}$:

$$\int_{\Delta_{s-1}}^{2\Delta_{s-1}} u^k f(c_s+\Delta_s-u)du = \int_0^{\Delta_{s-1}} (u+\Delta_{s-1})^k f(c_s+\Delta_{s-1}-u)du \tag{60}$$

Using the equality $f(c_s+\Delta_{s-1}-u) = f(c_{s-1}-(J_{s-1}-1)\Delta_{s-1}-u)$ and the binomial theorem

$$(u+\Delta_{s-1})^k = \sum_{r=0}^{k} \binom{k}{r} u^r (\Delta_{s-1})^{k-r},$$

we have

$$\begin{aligned}
&\int_{\Delta_{s-1}}^{2\Delta_{s-1}} u^k f(c_s+\Delta_s-u)du \\
&= \sum_{r=0}^{k} \binom{k}{r} (\Delta_{s-1})^{k-r} \int_0^{\Delta_{s-1}} u^r f(c_s+\Delta_{s-1}-u)du \\
&= \sum_{r=0}^{k} \binom{k}{r} (\Delta_{s-1})^{k-r} I_{s-1,r} f(c_{s-1}-(J_{s-1}-1)\Delta_{s-1}),
\end{aligned} \tag{61}$$

where eq. (22) is used in the last equality. Thus, we obtain the following expression for $I_{s,k}f(c_s+\Delta_s)$ with use of eq. (59) and (61) as

$$\begin{aligned}
&I_{s,k} f(c_s+\Delta_s) \\
&= I_{s-1,k} f(c_{s-1}-(J_{s-1}-2)\Delta_{s-1}) \\
&\quad + \sum_{r=0}^{k} \binom{k}{r} (\Delta_{s-1})^{k-r} I_{s-1,r} f(c_{s-1}-(J_{s-1}-1)\Delta_{s-1})
\end{aligned} \tag{62}$$

Copyright © 2011 by ASME