

RESEARCH ARTICLE

An efficient numerical method for fractional ordinary differential equations - based on exponentially decreasing random memory on uniform meshes

LW Somathilake*

Department of Mathematics, Faculty of Science, University of Ruhuna, Matara.


Submitted: 11 April 2019; Revised: 13 January 2020; Accepted: 03 February 2020

Abstract: This paper proposes a new numerical method to solve non-linear fractional ordinary differential equations (FODEs) of the form $D^\gamma u(t) = f(t, u(t))$, with initial conditions $u(0) = u_0$. Here, $f(t, u)$ is a continuous function, $0 < \gamma \leq 1$ is an arbitrary positive real number and the fractional differential operator, D^γ , is in the sense of Caputo derivative. Fixed (short) memory method (SMM) and full memory method (FMM) are two established numerical methods for fractional differential equations. In fixed memory method, tail of the memory at each time step is cut off and hence an uncontrollable error occurs. Further, full memory method is not suitable for long time integration of fractional differential equations because of high computational cost. In the proposed method [hereinafter referred to as decreasing random memory method (DRMM)], number of memory points in the past are chosen randomly and they are decreasing along the tail of the memory. Numerical experiments showed that the error occurs in the proposed DRMM is less than that of SMM. The solutions obtained by FMM and DRMM were also very close to the actual solutions of the considered fractional differential equations. The proposed method-DRMM is more accurate than the SMM and the estimated order of convergence (EOC) of DRMM is almost same as that of FMM. The proposed method DRMM is more efficient than the established methods SMM and FMM.

Keywords: Fractional derivatives, fractional differential equation, full memory method, short memory principle.

INTRODUCTION

It is accepted that fractional differential equations are more suitable than standard differential equations to describe some real world problems. Applications of fractional differential equations are found in many areas such as physics and engineering (Schuessel *et al.*, 1995; Diethelm & Freed, 1999; Sabatier *et al.*, 2007), chemistry and biochemistry (Oldham, 2010), finance (Scalas *et al.*, 2000) mechanics and system biology, etc. Fractional-order models are more adequate than the standard order (with integer order derivatives and integrals) models, because fractional derivatives and integrals bring forward the temporal and spatial memory heredity of the materials and processes. This is the main advantage of fractional order models when compared with integer order models. It is very difficult if not impossible to obtain analytical solutions for most fractional differential equations. However, some analytical (Podlubny, 1999; Diethelm, 2010; Gafiychuk & Datsko, 2010; Kirane *et al.*, 2014) and numerical (Yuste & Acedo, 2005; Zhuang & Liu, 2006; Yu *et al.*, 2008; Rida *et al.*, 2010; Scherer *et al.*, 2011; Xu & He, 2011; Deng & Li, 2012; Deng *et al.*, 2016) techniques play an important role in identifying the solution behaviour and obtaining approximate solutions of such fractional differential equations.

* Corresponding author (sthilake@maths.ruh.ac.lk;  <https://orcid.org/0000-0002-7276-1357>)



However, numerical integration of fractional ordinary differential equations (FODEs) on finer meshes and on long time periods is a computationally expensive process and reducing computational cost by controlling numerical errors is the main challenge. The aim of this paper is to face this challenge by introducing an efficient numerical scheme to integrate FODEs.

Fractional derivatives

There are different definitions for fractional derivatives. Riemann-Liouville definition and Caputo definition are the popular definitions. These definitions are reported in previous studies (Podlubny, 1999; Daftardar-Gejji & Jafari, 2005; Diethelm & Kai, 2010; Pindza & Owolabi, 2016; Garrappa, 2018).

Definition 1.1 (Daftardar-Gejji & Jafari, 2005; Pindza & Owolabi, 2016) A real valued function $f(x); x > 0$, is said to be in the space C_ν ; $\nu \in \mathbb{R}$ if there exists a real number $p > \nu$ such that $f(x) = x^p f_1(x)$, where $f_1(x) \in [0, \infty)$. This implies $C_\nu \subset C_\nu$ if $\nu \leq \nu$.

Definition 1.2 (Daftardar-Gejji & Jafari, 2005; Pindza & Owolabi, 2016) A function $f(x)$, $x > 0$, is said to be in the space C_ν^m , $m \in \mathbb{N} \cup \{0\}$, if $f^{(m)} \in C_\nu$.

Definition 1.3 [Riemann-Liouville derivative (Pindza & Owolabi, 2016)]. Let $u \in C_q^m$, $m \in \mathbb{N}$. The Riemann-Liouville derivative of order q , ${}^R D_t^q u(t)$, is defined as

$${}^R D_t^q u(t) = \begin{cases} \frac{1}{\Gamma(m-q)} \frac{d^m}{dt^m} \int_a^t (t-\tau)^{m-q-1} u(\tau) d\tau, & m-1 \leq q < m, \\ \frac{d^m u(t)}{dt^m}, & q = m \in \mathbb{N}, \end{cases} \dots(01)$$

where $\Gamma(z)$ ($z \in \mathbb{C}$) denotes the Euler gamma function defined by $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$, q is the fractional order of the derivative, $m (\equiv \lceil q \rceil)$ is the smallest integer greater than or equal to q and a is the initial time.

Definition 1.4 [Caputo fractional derivative (Pindza & Owolabi, 2016)]. Let $u \in C_q^m$, $m \in \mathbb{N}$. The Caputo fractional derivative operator of order $q > 0$ is defined as:

$${}^C D_t^q u(t) = \begin{cases} \frac{1}{\Gamma(m-q)} \int_a^t (t-\tau)^{m-q-1} \frac{d^m u(\tau)}{d\tau^m} d\tau, & m-1 < q < m, \\ \frac{d^m u(t)}{dt^m}, & q = m \in \mathbb{N}. \end{cases} \dots(02)$$

METHODOLOGY

In this paper numerical simulations of fractional differential equations are performed using numerical schemes based on two established discrete fractional order derivatives and newly introduced discrete fractional order derivative. These discrete fractional order derivatives are constructed based on finite difference formula of Caputo derivative proposed in Karatay *et al.* (2011).

Discrete fractional order derivatives (DFOD)

Fractional-order derivative with full memory

Consider the fractional differential equation $D^\gamma u(t) = f(t, u(t))$, $u(0) = u_0$, $0 < \gamma \leq 1$, $0 \leq t \leq T$. f is continuous function on a suitable set. According to Diethelm and Ford (2002) the above fractional differential equation has a unique solution u on $0 \leq t \leq T$. Discretising the time interval $0 \leq t \leq T$ into N number of partitions with equal step size Δt , one gets a uniform mesh.

Finite difference formula of first order on a uniform mesh for Caputo derivative of order γ ($0 < \gamma < 1$) is reported in (Karatay *et al.*, 2011) in the form:

$$D^\gamma u(t_m) = \frac{1}{(\Delta t)^\gamma} \sum_{k=0}^m g_k^\gamma (U^{m-k} - U^0) + O(\Delta t),$$

where $t_m = m\Delta t$ and U^m denote the numerical approximation to the exact value $u(t_m)$. Here,

$$g_k^\gamma = (-1)^k \binom{\gamma}{k} \text{ where } \binom{\gamma}{k} = \frac{\Gamma(\gamma+1)}{k! \Gamma(\gamma-k+1)}$$

represents the fractional binomial coefficients. Also, g_k^γ ($k = 1, 2, \dots$) has the recursive relation

$$g_0^\gamma = 1, g_k^\gamma = \left(1 - \frac{\gamma+1}{k}\right) g_{k-1}^\gamma$$

for $k \geq 1$ and $\sum_{k=0}^\infty g_k^\gamma = 0$. The approximation

$${}_{FM} D_{0,t}^\gamma u(t_m) \approx \frac{1}{(\Delta t)^\gamma} \sum_{k=0}^m g_k^\gamma (U^{m-k} - U^0)$$

is called the discrete fractional order derivative with full memory.

In full memory method, all the memory points up to the considered time step are taken into account in the calculation of the fractional order derivative at that point (Figure 1).

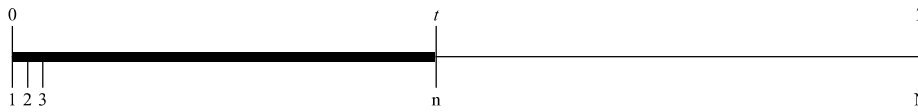


Figure 1: Sketch of the selection of full memory length

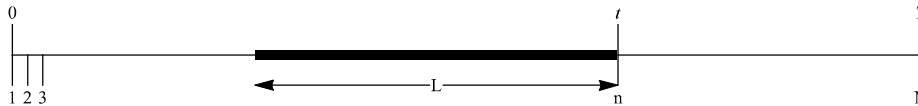


Figure 2: Sketch of the short memory

Fractional order derivative with fixed (short) memory length

In full memory method, the unknown values depend on all the past time steps. Applying full memory method in solving fractional differential equations is computationally inefficient. However, in fixed (short) memory principle (Podlubny, 1999), it is assumed that the unknown values at a time step depend only on the values at recent past time steps corresponding to time length L (Figure 2). Based on this assumption, Grünwald-Letnikov fractional-order derivative with fixed memory length L , is defined in Abdelouahab and Hamri (2016). This method is computationally efficient but computational error is higher due to ignoring the tail of the memory.

Based on the short memory principle, the fractional order derivative with short memory L , ${}_{SM}D_{0,t}^\gamma u(t_m)$, corresponding to ${}_{FM}D_{0,t}^\gamma u(t_m)$ can be written in the form:

$${}_{SM}D_{0,t}^\gamma u(t_m) = \begin{cases} \frac{1}{(\Delta t)^\gamma} \sum_{k=0}^{\lceil L/\Delta t \rceil} g_k^\gamma (U^{m-k} - U^0), & \text{if } t_m > L, \\ \frac{1}{(\Delta t)^\gamma} \sum_{k=0}^m g_k^\gamma (U^{m-k} - U^0), & \text{if } t_m \leq L, \end{cases} \dots(03)$$

where $[x]$ denotes the smallest integer greater than or equal to x for $x \in \mathbb{R}$. Although, this scheme

is computationally cheap, the error propagation due to neglecting the tail of the memory in each time step is uncontrollable without increasing the fixed memory length. When memory length is increased the computation becomes difficult again.

A new discrete fractional order derivative with exponentially decreasing Random memory

In this section, a new discrete fractional order derivative on uniform meshes is introduced by choosing memory points randomly and they are decreasing along the tail of the memory. Suppose that the time interval $[0, T]$ is discretised into N number of partitions with equal step size Δt . Let M and N_1 be integers such that $N_1 = N/M$. That is the interval $[0, T]$ is divided into N_1 number of partitions P_1, P_2, \dots, P_{N_1} with equal length $M\Delta t$ (Figure 3). Now consider the case of integration up to time $t = m\Delta t$. Let $n_1 = \lceil \frac{m}{N} \rceil - 1$ Generate $R_i = \lceil M \text{Exp}(-\alpha i/N_1) \rceil$ number of random integers within the partition P_i for $i = 1, 2, \dots, n_1 - 1$. Here α is a real number which determines the decreasing speed of the number of random points through the partitions P_i as i increases from 1 to $n_1 - 1$. Let A_i be the set of R_i number of random integers chosen on partition P_i sorted in ascending order. Denote the set A_i as $A_i = \{a_1^i, a_2^i, \dots, a_{R_i}^i\}$ ($i = 1, 2, 3, \dots, n_1 - 1$). Let $n_2 = m - M(n_1 - 1)$ if $n_1 \geq 2$ (Figure 4). Choose memory points on $[0, m\Delta t]$ as follows:

$$\left\{ \begin{array}{l} \text{No of} \\ \text{memory} \\ \text{points} \end{array} \right\} = \left\{ \begin{array}{l} m, \\ \left\{ \begin{array}{l} \text{All, from } M(n_1 - 1) \text{ to } m \\ R_i, \text{ on the partition } P_i \text{ (} i = 1, 2, \dots, n_1 - 1 \text{)} \end{array} \right\}, \end{array} \right. \begin{array}{l} \text{if } n_1 = 0, 1 \text{ (or } m < 2M) \\ \text{if } n_1 \geq 2 \text{ (or } m \geq 2M). \end{array}$$

Fractional-order derivative with random memory is defined as follows.

$${}_{RM}D_{0,t}^\gamma u(t_m) = \begin{cases} \frac{1}{(\Delta t)^\gamma} \sum_{k=0}^m g_k^\gamma (U^{(m-k)} - U^0), & m < 2M, \\ \frac{1}{(\Delta t)^\gamma} \left(\sum_{k=0}^{n_2} g_k^\gamma (U^{(m-k)} - U^0) + \sum_{j=1}^{n_1-1} w_j \sum_{i=1}^{R_j} g_{l_{i,j}}^\gamma (U^{(m-l_{i,j})} - U^0) \right), & m \geq 2M, \end{cases} \dots(04)$$

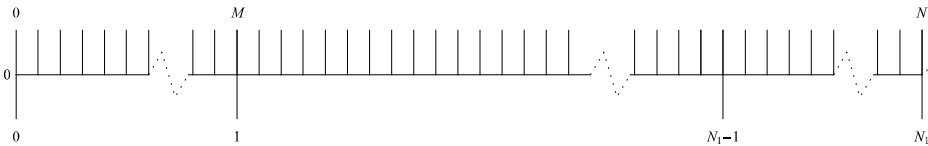


Figure 3: Sketch for the discretisation of time interval $[0, T]$

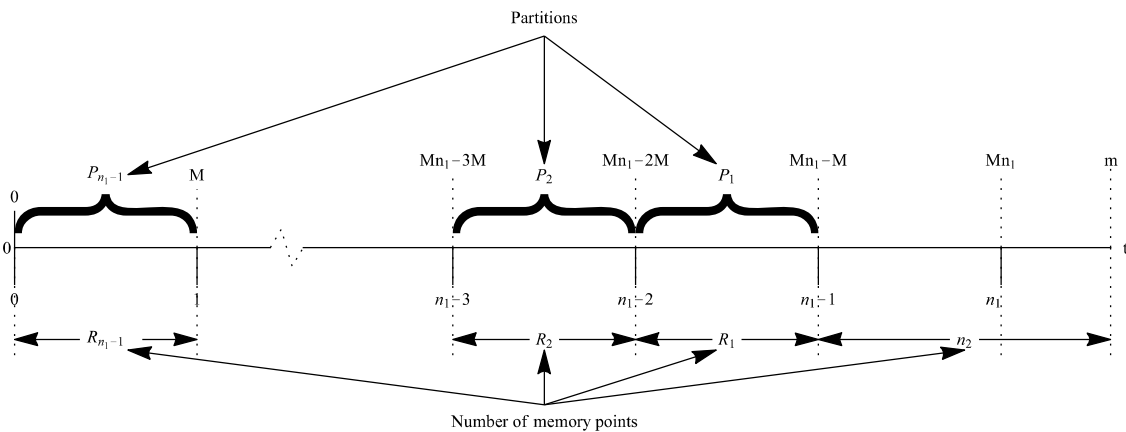


Figure 4: Sketch for the distribution of number of memory points on the time interval $[0, t], t < T$

where $l_{i,j} = n_2 + (j - 1)M + a_i^j$. The weights $w_i = M/R_i$ ($i = 1, 2, \dots, n_1 - 1$) are introduced to reduce the error, which occurs at the partition P_i due to ignoring $(M - R_i)$ number of memory points.

Figure 5 shows the variation of the distribution of random memory points over the time interval $[0, 1]$ at different levels of α . These distributions are not fixed as memory points are chosen randomly. It can be observed that as α increases the sparsity of the distribution of random memory points increases. Figure 6 depicts how the random memory points distribute over the space upto different time levels.

Numerical schemes

This section introduces three numerical schemes based on the above introduced fractional order derivatives to simulate fractional differential equations of the form:

$$D^\gamma u(t) = f(t, u(t)), \quad u(0) = u_0, \quad 0 < \gamma \leq 1, \quad 0 \leq t \leq T, \dots(05)$$

where $f(t, u(t))$ is a continuous function.

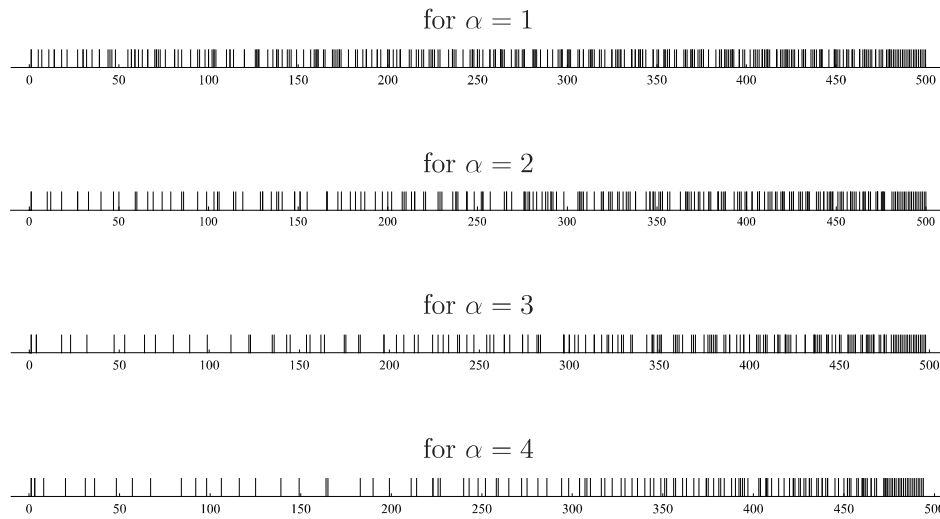


Figure 5: Distribution of random memory points for different levels of α when $T = 1, dt = 0.002, M = 50$

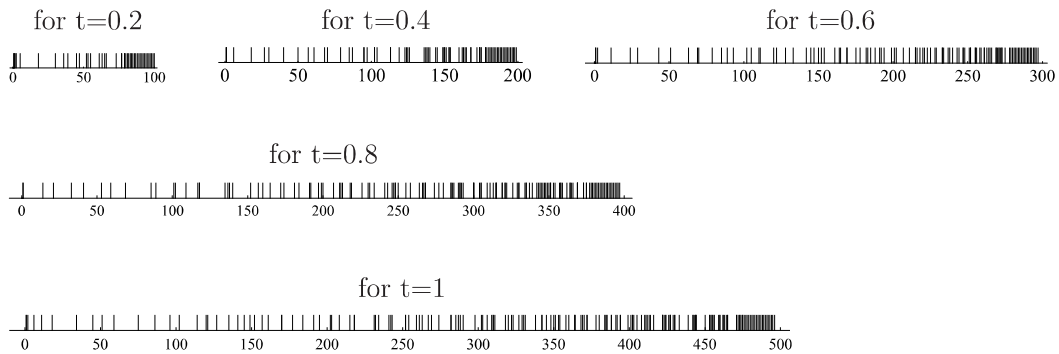


Figure 6: Distribution of random memory points over different time intervals for $dt = 0.002, M = 50$ and $\alpha = 3$

A fully explicit scheme with full memory

Applying fractional-order derivative with full memory for the fractional derivative of equation (05), the following fully explicit scheme can be obtained.

$$\frac{1}{(\Delta t)^\gamma} \sum_{k=0}^{m=\lceil t/\Delta t \rceil} g_k^\gamma (U^{m-k} - U^0) = f(U^{m-1}, t_{m-1}). \quad \dots(06)$$

$$\Rightarrow U^m = U^0 - \sum_{k=1}^{m=\lceil t/\Delta t \rceil} g_k^\gamma (U^{m-k} - U^0) + (\Delta t)^\gamma f(U^{m-1}, t_{m-1}), \quad \dots(\text{FMM})$$

for $m = 1, 2, \dots, N$. When integrating a FODE from 0 to T using this scheme, $N(N + 1)/2$ total number of summations (TNOS) have to be done in the process.

A fully explicit scheme with fixed (short) memory

A fully explicit scheme with short memory length L can be obtained in the following form by replacing fractional order derivative of (05) by (03) and rearranging the terms.

$$U^m = \begin{cases} U^0 - \sum_{k=1}^{N_L} g_k^\gamma (U^{m-k} - U^0) + (\Delta t)^\gamma f(U^{m-1}, t_{m-1}), & \text{if } m > N_L, \\ U^0 - \sum_{k=1}^m g_k^\gamma (U^{m-k} - U^0) + (\Delta t)^\gamma f(U^{m-1}, t_{m-1}), & \text{if } m \leq M_L, \end{cases} \dots(\text{SMM})$$

for $m = 1, 2, \dots, N$, where $N_L = \lceil L/\Delta t \rceil$. TNOS of this process is $N_L(2N - N_L + 1)/2$.

A fully explicit scheme with decreasing random memory

Replacing fractional order derivative of (05) by (04) and rearranging the terms, the following fully explicit scheme with decreasing random memory can be obtained.

$$U^m = \begin{cases} U^0 - \sum_{k=1}^m g_k^\gamma (U^{(m-k)} - U^0) & \text{if } m < 2M, \\ U^0 - \sum_{k=1}^{n_2} g_k^\gamma (U^{(m-k)} - U^0) - \sum_{j=1}^{n_1-1} w_j \sum_{i=1}^{R_j} g_{l_{i,j}}^\gamma (U^{(m-l_{i,j})} - U^0) & \\ \quad + (\Delta t)^\gamma f(U^{m-1}, t_{m-1}), & \text{if } m \geq 2M, \end{cases} \dots(\text{DRMM})$$

for $m = 1, 2, \dots, N$. The TNOS of a process of integration up to given time T of this scheme depends on the values M and α introduced previously. Table 1 shows the TNOS of FMM, SMM and DRMM. The dependence of TNOS

on M and α of DRMM is also shown in the same table. These numbers for DRMM are not fixed as memory points and are chosen randomly. PML means percentage memory length of SMM.

Table 1: TNOS in FMM, SMM and variation of TNOS in DRMM with α and M in the process of integrating on the time interval $[0, 5]$ with $\Delta t = 0.001$

Time	Δt	FMM	TNOS			
			SMM (PML=20)	M	DRMM α	
5	0.001	12502500	4500500	50	1	6687389
					2	3855436
					3	2388359
					4	1604685
					5	1141832
5	0.001	12502500	4500500	100	1	6497607
					2	3735798
					3	2294762
					4	1504851
					5	1055341

Computational cost of the above schemes mainly depend on the corresponding TNOS. According to Table 1 TNOS of DRMM (for $\alpha = 2, 3, 4, 5$) is less than that of SMM (for PML = 20) and FMM. Therefore, the computational cost of DRMM (for $\alpha = 2, 3, 4, 5$) should be less than that of SMM (for PML = 20) and FMM.

RESULTS AND DISCUSSION

Numerical examples

Accuracies of the above three numerical schemes are compared with the exact solutions of the following fractional differential equations with $0 < \gamma \leq 1$ and the initial conditions $u(0)=0$:

$$D^\gamma u(t) = \frac{2}{\Gamma(3-\gamma)}t^{2-\gamma} - \frac{1}{\Gamma(1-\gamma)}t^{1-\gamma} - u(t) + t^2 - t. \quad \dots(07)$$

$$D^\gamma u(t) = \frac{\Gamma(9)}{\Gamma(9-\gamma)}t^{8-\gamma} - 3\frac{\Gamma(5+\gamma/2)}{\Gamma(5-\gamma/2)}t^{4-\gamma/2} + \frac{9}{4}\Gamma(\gamma+1) + \left(\frac{3}{2}t^{\gamma/2} - t^4\right)^3 - (u(t))^{3/2}. \quad \dots(08)$$

$$D^\gamma u(t) = \frac{\Gamma(6)}{\Gamma(6-\gamma)}t^{5-\gamma} - 3\frac{\Gamma(5)}{\Gamma(5-\gamma)}t^{4-\gamma} + \frac{\Gamma(4)}{\Gamma(4-\gamma)}t^{3-\gamma}. \quad \dots(09)$$

The exact solutions of the FODEs (07), (08) and (09) are $u(t) = t^2 - t$, $u(t) = \left(t^8 - 3t^{4+\gamma/2} + \frac{9}{4}t^\gamma\right)$ and $u(t) = (t^5 - 3t^4 + 2t^3)$, respectively. The numerical solutions obtained by FMM, SMM, DRMM and exact solutions of above three differential equations are shown in Figure 7. According to this Figure, numerical solutions obtained by DRMM is closer to the exact solutions than that of obtained by SMM. Numerical solutions obtained by FMM are the closest solutions to the exact solutions of the considered FODEs. However, according to Table 5 the computational cost of FMM is higher than that of DRMM.

Now compare the errors between numerical solutions obtained by the numerical schemes FMM, SMM, DRMM and exact solutions. Table 2 shows the relative mean square error (Re-MSE) between exact solutions and numerical solutions at the end point of the considered time ranges of the FODEs (07), (08) and (09) obtained by the three numerical schemes. In these simulations $\alpha = 3$, $M = 100$ for DRMM, PML = 20 for SMM and

Table 2: Re-MSE between exact solutions and numerical solutions of FODEs (07), (08) and (09) for $\gamma = 0.8$

FODE	T	FMM	DRMM	SMM
(07)	20	1.1216E-07	2.8698E-04	0.0011
(08)	1	1.5480E-09	1.3628E-04	4.3231E-04
(09)	2	1.0610E-09	1.0592E-09	2.5609E-05

$\Delta t = 0.0001$. Re-MSE of DRMM is less than that of SMM. This implies that DRMM is more accurate than SMM.

Estimated (or experimental) order of convergence (EOC)

The estimated (or experimental) order of convergence (Diethelm *et al.*, 2004; Li & Zeng, 2013; Garrappa, 2015; 2018; Liu *et al.*, 2018) an estimator for convergence order of a numerical scheme, is measured by $EOC = \log_2 \left(\frac{E_{\max}(\Delta t)}{E_{\max}(\Delta t/2)} \right)$,

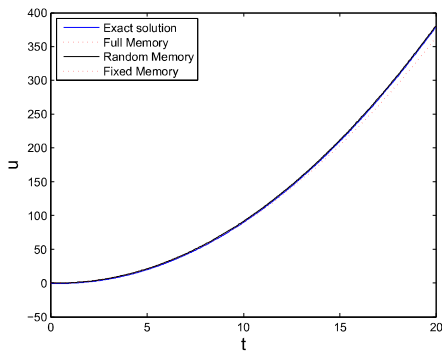
$E_{\max}(\Delta t) = \max_{1 \leq i \leq N} |u(t_i) - U_{\Delta t}^i|$. Here, $U_{\Delta t}^i$ is the numerical value of $u(t)$ at $t = t_i (= i\Delta t)$ and $u(t_i)$ is the exact solution at $t = t_i$. EOCs of FMM and DRMM corresponding to the fractional differential equations (07), (08) and (09) are shown in Tables 3 and 4, respectively. EOC of SMM is not taken into account as it is very low. Convergence orders of FMM and DRMM are approximately equal as EOCs of DRMM and FMM are approximately equal.

Computational cost

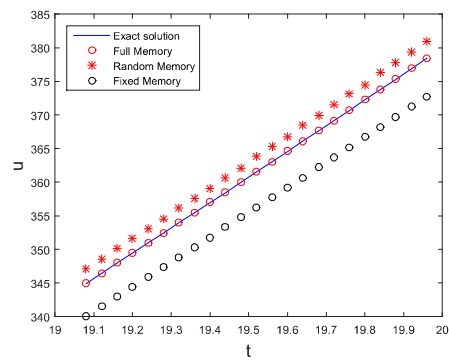
In this section, the computational time (CPU time) of the three numerical schemes FMM, DRMM and SMM are compared. Algorithms for the numerical schemes FMM, SMM and DRMM were developed and solved using Matlab on a 2.3GHz, Intel core i5 laptop computer that had 8GB of RAM and Microsoft Windows 10. Now define the two terms, computational time (CT) reduction percentage (CTRP) and E_{\max} increment percentage ($E_{\max}IP$) between two methods FMM and DRMM as follows:

$$CTRP = \frac{CT \text{ of FMM} - CT \text{ of DRMM}}{CT \text{ of FMM}} \times 100 \text{ and}$$

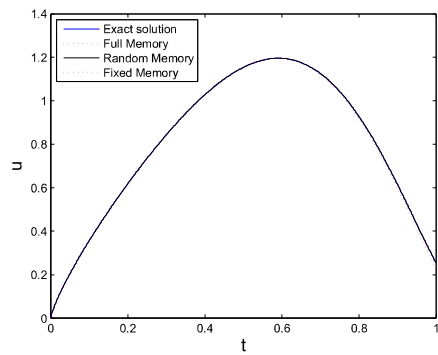
$$E_{\max}IP = \frac{E_{\max} \text{ of DRMM} - E_{\max} \text{ of FMM}}{E_{\max} \text{ of FMM}} \times 100.$$



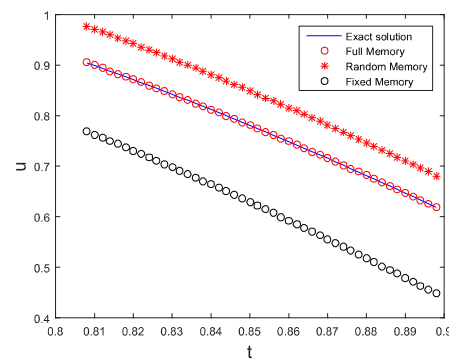
(a) Solutions of (07) for $0 \leq t \leq 20$



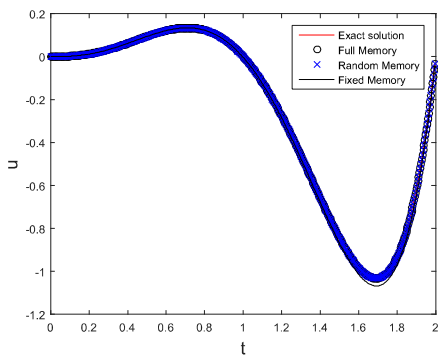
(b) Solutions of (07) for $19 \leq t \leq 20$



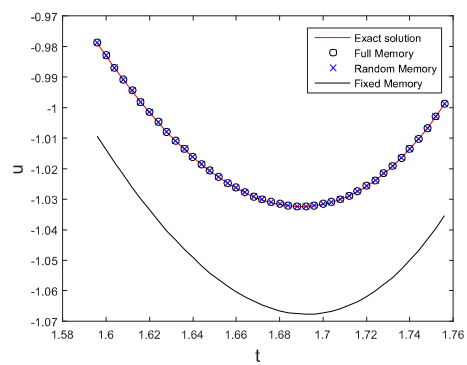
(c) Solutions of (08) for $0 \leq t \leq 1.0$



(d) Solutions of (08) for $0.8 \leq t \leq 0.9$



(e) Solutions of (09) for $0 \leq t \leq 2.0$



(f) Solutions of (09) for $1.58 \leq t \leq 1.78$

Figure 7: Numerical solutions and exact solutions of the FODEs (07), (08) and (09). (In these simulations $\alpha = 3$, $M = 100$ in DRMM, $PML = 20$ in SMM and $\Delta t = 0.0001$).

Table 3: EOC of FMM corresponding to FODEs (07), (08) and (09) for $\gamma = 0.8$

Step Size ($\Delta t = 1/2^p$)	$E_{\max}(\Delta t)$ of FODE			EOC of FODE		
	(07)	(08)	(09)	(07)	(08)	(09)
1/2 ⁹	1.4362	0.0042	0.0175	-	-	-
1/2 ¹⁰	0.7177	0.0021	0.0088	1.0007	1.0004	0.9966
1/2 ¹¹	0.3588	0.001	0.0044	1.0004	1.0002	0.9982
1/2 ¹²	0.1794	5.2490×10^{-4}	0.0022	1.0002	1.0001	0.9991
1/2 ¹³	0.0897	2.6244×10^{-4}	0.0011	1.0001	1.000	0.9996
1/2 ¹⁴	0.0448	1.3122×10^{-4}	5.5042×10^{-4}	1.000	1.000	0.9998
1/2 ¹⁵	0.0224	6.5608×10^{-5}	2.7523×10^{-4}	1.000	1.000	0.9999

Table 4: EOC of DRMM (for $M = 2^{(p-6)}, \alpha = 3$) corresponding to FODEs (07), (08) and (09) for $\gamma = 0.8$

Step Size ($\Delta t = 1/2^p$)	$E_{\max}(\Delta t)$ of FODE			EOC of FODE		
	(07)	(08)	(09)	(07)	(08)	(09)
1/2 ⁹	15.1765	0.0252	0.0129	-	-	-
1/2 ¹⁰	6.1683	0.0097	0.0038	1.2989	1.3804	1.7633
1/2 ¹¹	3.1805	0.0056	0.0011	0.9556	0.7848	1.7885
1/2 ¹²	1.7715	0.0033	3.8387×10^{-4}	0.8443	0.7851	1.5188
1/2 ¹³	0.8401	0.0015	2.3576×10^{-4}	1.0763	1.0989	0.7033
1/2 ¹⁴	0.4412	4.7399×10^{-4}	1.1435×10^{-4}	0.9292	1.6827	1.0439
1/2 ¹⁵	0.2076	2.2416×10^{-4}	5.1306×10^{-5}	1.0912	1.0803	1.1563

Table 5: Variation of CT, E_{\max} , CTRP and E_{\max} IP with respect to time levels, when integrate FODE (07) using FMM and DRMM (for $\gamma = 0.8$ and mesh size $\Delta t = 10^{-3}$)

T	FMM		DRMM ($M = N/10, \alpha = 3$)		CT reduction and E_{\max} increment	
	CT (in secnds)	E_{\max}	CT (in seconds)	E_{\max}	CTRP	E_{\max} IP
25	8.2281	0.0466	6.0066	0.0596	26.9989	27.8970
50	31.8865	0.0962	23.0700	0.1175	27.6496	22.1414
75	71.4645	0.1459	50.7977	0.1699	28.9189	16.4496
100	148.9797	0.1958	91.4200	0.238	38.6359	21.5526
125	197.7228	0.2456	152.7657	0.2862	22.7374	16.5309
150	334.7025	0.2955	218.9782	0.3372	34.5753	14.1117
175	387.3641	0.3454	294.6666	0.3929	23.9303	13.7522
200	596.3558	0.3953	487.7036	0.4296	18.2194	8.6769
225	639.4727	0.4452	487.7036	0.4741	23.7335	6.4915
250	932.6186	0.4957	616.2594	0.5330	33.9216	7.5247
275	972.7584	0.545	724.3764	0.5793	25.5338	6.2936
300	1146.7000	0.5949	863.8433	0.6211	24.6670	4.4041
325	1408.2000	0.6449	1036.0000	0.6828	26.4309	5.8769
350	1520.6000	0.6948	1201.3000	0.7294	20.9983	4.9798

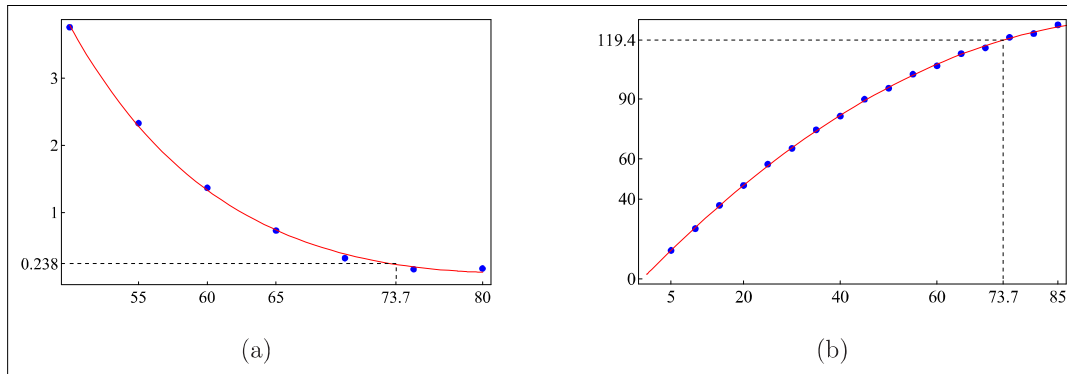


Figure 8: (a): Plot of E_{\max} and fitted values against PML of SMM correspond to FODE (07); (b): Plot of CT and fitted values against PML of SMM correspond to FODE (07)

Table 6: Variation of CT and E_{\max} with respect to PML when integrate FODE (07) up to $T = 100$ using SMM for $\gamma = 0.8$ and mesh size $\Delta t = 10^{-3}$

PML	20	30	40	50	60	70	80
CT	46.7706	65.2946	81.4340	95.3357	106.5582	115.5391	122.6595
E _{max}	49.8436	20.6525	8.9790	3.7580	1.3673	0.3211	0.1663

Table 5 shows the CT, E_{\max} , CTRP and E_{\max}^{IP} of FMM and DRMM ($\alpha = 3, M = 100$) when FODE (07) is integrated up to different time levels. Computational cost of DRMM has been reduced approximately by 20 % when compared with FMM. It can be seen that CTRP increases and E_{\max}^{IP} decreases as time increases. Therefore, the DRMM is computationally more efficient than the FMM in integration of FODEs on more finer meshes or on larger time ranges.

Table 6 shows the CT and E_{\max} of SMM corresponding to FODE (07) at different PML levels.

E_{\max} and CT of SMM of FODE (07) (for $\gamma = 0.8, T = 100$ and $\Delta t = 0.0001$) are fitted against PML and these are best fitted respectively as follows:

$$E_{\max} = 8.74551 - \frac{1429.17}{\text{PML}} - \frac{59065.1}{(\text{PML})^2} \text{ and}$$

$$\text{CT} = 2.29745 - 2.46271\text{PML} - 0.0118535(\text{PML})^2.$$

Figures 8(a) and 8(b) show the plots of E_{\max} , CT and their fittings against PML of the SMM corresponding to the FODE (07). E_{\max} of DRMM for $\gamma = 0.8, T = 100, \Delta t = 0.0001, M = N/10$ and $\alpha = 3$ is 0.238 (Table 5). According to Figure 8, it can be seen that the computational cost of SMM corresponding to $E_{\max} = 0.238$ is approximately 119.4. This computational cost is higher than that of DRMM (91.4200). Therefore, DRMM is computationally efficient than SMM when computational error is fixed to a same value of both methods.

CONCLUSION

The relative mean square error between exact solution and numerical solutions of the considered FODEs obtained by DRMM is less than that of SMM. Therefore, the proposed method, DRMM, is more accurate than SMM. The values of EOC of FMM corresponding to the considered three FODEs are approximately equal to one. These results agree with the fact that the order of convergence of FMM is one. The values of EOC

of DRMM is approximately equal to that of FMM. Therefore, EOC of the proposed method, DRMM, is good enough for numerical simulations of fractional differential equations. Computational cost of DRMM is less than that of FMM and SMM when the accuracy of SMM and DRMM are in the same level. Based on these facts, it can be concluded that the proposed method, DRMM, is more efficient than the established methods, SMM and FMM, in numerical computations of FODEs. Future research would consider the applicability of newly proposed DRMM to higher order numerical schemes for fractional differential equations. Further, it is expected to investigate the suitability of DRMM to solve time fractional reaction diffusion systems.

REFERENCES

- Abdelouahab M.-S. & Hamri N.-E. (2016). The Grünwald-Letnikov fractional-order derivative with fixed memory length. *Mediterranean Journal of Mathematics* **13**(2): 557–572.
DOI: <https://doi.org/10.1007/s00009-015-0525-3>
- Daftardar-Gejji V. & Jafari H. (2005). Adomian decomposition: a tool for solving a system of fractional differential equations. *Journal of Mathematical Analysis and Applications* **301**(2): 508–518.
DOI: <https://doi.org/10.1016/j.jmaa.2004.07.039>
- Deng J., Zhao L. & Wu Y. (2017). Fast predictor-corrector approach for the tempered fractional differential equations. *Numerical Algorithms* **74**: 717–754.
DOI: <https://doi.org/10.1007/s11075-016-0169-9>
- Deng W. & Li C. (2012). Numerical schemes for fractional ordinary differential equations. *Numerical Modelling* **16**: 355–374.
DOI: <https://doi.org/10.5772/34965>
- Diethelm K. & Ford N.J. (2002). Analysis of fractional differential equations. *Journal of Mathematical Analysis and Applications* **265**(2): 229–248.
DOI: <https://doi.org/10.1006/jmaa.2000.7194>
- Diethelm K., Ford N.J. & Freed A.D. (2004). Detailed error analysis for a fractional Adams method. *Numerical Algorithms* **36**(1): 31–52.
DOI: <https://doi.org/10.1023/B:NUMA.0000027736.85078.be>
- Diethelm K. & Freed A. D. (1999). On the solution of nonlinear fractional-order differential equations used in the modeling of viscoplasticity. In: *Scientific Computing in Chemical Engineering II*, pp. 217–224. Springer, Berlin, Germany.
DOI: https://doi.org/10.1007/978-3-642-60185-9_24
- Diethelm K. (2010). *The Analysis of Fractional Differential Equations: An Application-Oriented Exposition Using Differential Operators of Caputo Type*. Springer, New York, USA.
DOI: https://doi.org/10.1007/978-3-642-14574-2_8
- Gafiychuk V. & Datsko B. (2010). Mathematical modeling of different types of instabilities in time fractional reaction-diffusion systems. *Computers and Mathematics with Applications* **59**(3): 1101–1107.
DOI: <https://doi.org/10.1016/j.camwa.2009.05.013>
- Garrappa R. (2015). Trapezoidal methods for fractional differential equations: theoretical and computational aspects. *Mathematics and Computers in Simulation* **110**: 96–112.
DOI: <https://doi.org/10.1016/j.matcom.2013.09.012>
- Garrappa R. (2018). Numerical solution of fractional differential equations: a survey and a software tutorial. *Mathematics* **6**(2): 16.
DOI: <https://doi.org/10.1007/s11075-016-0169-910.3390/math6020016>
- Karatay I., Bayramoglu S.R. & Sahin A. (2011). Implicit difference approximation for the time fractional heat equation with the nonlocal condition. *Applied Numerical Mathematics* **61**(12): 1281–1288.
DOI: <https://doi.org/10.1016/j.apnum.2011.08.007>
- Kirane M., Ahmad B., Alsaedi A. & Al-Yami M. (2014). Non-existence of global solutions to a system of fractional diffusion equations. *Acta Applicandae Mathematicae* **133**(1): 235–248.
DOI: <https://doi.org/10.1007/s10440-014-9865-4>
- Li C. & Zeng F. (2013). The finite difference methods for fractional ordinary differential equations. *Numerical Functional Analysis and Optimization* **34**(2): 149–179.
DOI: <https://doi.org/10.1007/s11075-017-0419-5>
- Liu Y., Roberts J. & Yan Y. (2018). Detailed error analysis for a fractional Adams method with graded meshes. *Numerical Algorithms* **78**(4): 1195–1216.
DOI: <https://doi.org/10.1007/s11075-017-0419-5>
- Oldham K.B. (2010). Fractional differential equations in electrochemistry. *Advances in Engineering Software* **41**(1): 9–12.
DOI: <https://doi.org/10.1016/j.advengsoft.2008.12.012>
- Pindza E. & Owolabi K.M. (2016). Fourier spectral method for higher order space fractional reaction-diffusion equations. *Communications in Nonlinear Science and Numerical Simulation* **40**: 112–128.
DOI: <https://doi.org/10.1016/j.cnsns.2016.04.020>
- Podlubny I. (1999). Fractional differential equations, an introduction to fractional derivatives, fractional differential equations to methods of their solution and some of their applications. In: *Mathematics in Science and Engineering*, pp. 198. Academic Press, San Diego, USA.
- Rida S., El-Sayed A. & Arafa A. (2010). On the solutions of time-fractional reaction-diffusion equations. *Communications in Nonlinear Science and Numerical Simulation* **15**(12): 3847–3854.
DOI: <https://doi.org/doi:10.1016/j.cnsns.2010.02.007>
- Sabatier J., Agrawal O.P. & Machado J.T. (2007). *Advances in Fractional Calculus*, volume 4. Springer, Berlin, Germany.
- Scalas E., Gorenflo R. & Mainardi F. (2000). Fractional calculus and continuous-time finance. *Physica A: Statistical Mechanics and its Applications* **284**(1): 376–384.
DOI: [https://doi.org/10.1016/S0378-4371\(00\)00255-7](https://doi.org/10.1016/S0378-4371(00)00255-7)

- Scherer R., Kalla S.L., Tang Y. & Huang J. (2011). The Grünwald-Letnikov method for fractional differential equations. *Computers and Mathematics with Applications* **62**(3): 902–917.
DOI: <https://doi.org/10.1016/j.camwa.2011.03.054>
- Schiessel H., Metzler R., Blumen A. & Nonnenmacher T. (1995). Generalized viscoelastic models: their fractional equations with solutions. *Journal of Physics A: Mathematical and General* **28**(23): 6567–6584.
DOI: <https://doi.org/10.1088/0305-4470/28/23/012>
- Xu Y. & He Z. (2011). The short memory principle for solving Abel differential equation of fractional order. *Computers and Mathematics with Applications* **62**(12): 4796–4805.
DOI: <https://doi.org/10.1016/j.camwa.2011.10.071>
- Yu Q., Liu F., Anh V. & Turner I. (2008). Solving linear and non-linear space-time fractional reaction-diffusion equations by the Adomian decomposition method. *International Journal for Numerical Methods in Engineering* **74**(1): 138–158.
DOI: <https://doi.org/10.1002/nme.2165>
- Yuste S.B. & Acedo L. (2005). An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations. *SIAM Journal on Numerical Analysis* **42**(5): 1862–1874.
DOI: <https://doi.org/10.1137/030602666>
- Zhuang P. & Liu F. (2006). Implicit difference approximation for the time fractional diffusion equation. *Journal of Applied Mathematics and Computing* **22**(3): 87–99.
DOI: <https://doi.org/10.1007/BF02832039>