



An iterative approach for numerical solution to the time-fractional Richards equation with implicit Neumann boundary conditions

Mustafa Zeki^{1,*}, Salih Tatar²

¹ *College of Engineering and Technology, American University of the Middle East, Kuwait*

² *Department of Mathematics and Computer Science, College of Science and General Studies, Alfaisal University, Riyadh, KSA*

Abstract. In this paper, we develop an iterative method of lines scheme for the numerical solution to the time fractional Richards equation with implicit Neumann boundary conditions, which is an effective tool for describing a process of flow through unsaturated media. A numerical example is provided to show the effectiveness of the presented method for different model parameters and inputs. The method illustrated here can be applied to other types Richards equation with various input functions and Dirichlet boundary conditions.

2020 Mathematics Subject Classifications: 35R11, 65M20, 65N40

Key Words and Phrases: Fractional Richards equation, Caputo fractional derivative, Method of Lines

1. Introduction

Richards equation is the fundamental model for describing flow through unsaturated media. Richards equation takes the following form when water flows through one-dimensional horizontal soils:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(d(u) \frac{\partial u}{\partial x} \right), \quad (1)$$

where $u = u(t, x)$ is the volumetric water content, $d(u)$ is the water diffusivity, t is the time and x is the distance from the inlet of the horizontal medium column. The equation (1) takes the following form after the Boltzmann scaling $x = \lambda t^{1/2}$:

$$-\frac{\lambda}{2} \frac{\partial u}{\partial \lambda} = \frac{\partial}{\partial \lambda} \left(d(u) \frac{\partial u}{\partial \lambda} \right).$$

*Corresponding author.

DOI: <https://doi.org/10.29020/nybg.ejpam.v16i1.4679>

Email addresses: mustafa.zeki@aum.edu.kw (M. Zeki), statar@alfaisal.edu (S. Tatar)

However many experiments show that the evolution of a horizontal wetting front deviates significantly from Boltzmann scaling. These experiments reflect anomalous Boltzmann scaling $x = \lambda(u)t^{\beta/2}$, where β is the dimensionless exponent and $0 < \beta < 2$. Different models are proposed to capture non-Boltzmann scaling, for example see [4] - [13]. In [4] and [9], the authors allow the water diffusivity d varies as a function of both water content and time, i.e., $d = d(t, u)$. In [13], the authors propose the following time-fractional Richards equation:

$$\frac{\partial^\beta u}{\partial t^\beta} = \frac{\partial}{\partial x} \left(d(u) \frac{\partial u}{\partial x} \right), \quad (2)$$

where $0 < \beta < 1$ is the order of the time-fractional derivative, $0 < \beta < 1$, $\frac{\partial^\beta u}{\partial t^\beta}$ is the time-fractional Caputo fractional derivative and defined as follows:

$$\frac{\partial^\beta u}{\partial t^\beta} = \int_0^t \frac{(t - \xi)^{-\beta}}{\Gamma(1 - \beta)} \frac{\partial u(x, \xi)}{\partial \xi} d\xi,$$

where $\Gamma(\cdot)$ is the Gamma function. We note that there is another kind of fractional derivative used frequently called Riemann-Liouville fractional derivative defined by

$${}^R \partial_t^\beta u(x, t) = \frac{\partial}{\partial t} \int_0^t \frac{(t - \xi)^{-\beta}}{\Gamma(1 - \beta)} u(x, \xi) d\xi.$$

These two fractional derivatives agree when the initial condition is zero. Kilbas et al [11] and Podlubny [17] can be referred for further properties of the Caputo and Riemann-Liouville fractional derivatives. Since the initial condition is zero in our problem, any result found in the literature for one of these hold for the other one.

For the convenience of the reader, we present the main steps of the derivation of the governing equation by following [16]. If the fluid particles are trapped in some regions for several periods of time s_1, \dots, s_n then the continuity equation becomes

$$u_t = - \sum_{i=1}^n w_i \nabla \cdot q(x, t - s_i), \quad (3)$$

where w_i are some weights. If we take $w_i = w(s_i) \Delta s_i$ with $\Delta s_i = s_i - s_{i-1}$ for some weight density $w = w(s)$, and take limit as n goes to ∞ , (3) becomes

$$u_t = - \int_0^t w(t - s) \nabla \cdot q(x, s) ds. \quad (4)$$

Equation (4) accounts for the fluid particles that can be trapped for any period of time. The amount of flux of the particles that wait for the time s equals $w(s)$. Using the choice of [16] for w we arrive at:

$$u_t = -\frac{1}{\Gamma(\beta)} \frac{\partial}{\partial t} \int_0^t (t-s)^{\beta-1} \nabla \cdot q(x, s) ds = -{}^R\partial_t^{1-\beta} \nabla \cdot q, \quad (5)$$

where ${}^R\partial_t^{1-\beta}$ denotes the Riemann-Liouville fractional derivative of order $1 - \beta$. If we apply the Riemann-Liouville fractional operator $I_t^{1-\beta}$, defined by

$$I_t^{1-\beta} u(x, t) := \frac{1}{\Gamma(1-\beta)} \int_0^t (t-s)^{-\beta} u(x, s) ds$$

to both sides of (5) and take into account the composition formula for the functions with vanishing initial conditions we have:

$$\frac{\partial^\beta}{\partial t^\beta} u = \nabla \cdot (d(u) \nabla u). \quad (6)$$

Equation (6) is also called generalized Richards equation can be found in many papers. Note that in the one-dimensional case, the equation (6) is the same as (2). To the best of authors knowledge, there are only a few studies for the numerical solution of the nonlinear nonlocal partial differential equations. In [14], the fractional Richards equation is solved numerically by using an implicit finite difference approximation. They convert the resulting system of equations into a tridiagonal system of non-linear equations. Then they solve the nonlinear system by Gauss elimination. In [8], after the authors state the fractional Richards equation as an integral equation, they adopt Adams-Bashforth-Moulton algorithm for the nonlinear case. They test their numerical code by comparing with analytical results for both $\beta = 1$ and $\beta < 1$ of the linear classical and fractional equations. In [7], the authors show that the fractional Richards equation has a form that is suitable for a self-similar solution using the variable $\xi = x/t^{\beta/2}$ and represent in a form that is suitable for finding the self-similar solution. The self-similar solution is obtained only in cases of certain initial and boundary conditions: $u(t, 0) = 1$, $u(t, \infty) = 0$, $u(0, x) = 0$. The authors develop an implicit numerical approach for numerical simulation of nonlinear variable order time-fractional diffusion/wave-diffusion equations in [20]. These equations are useful to describe liquid infiltration for both subdiffusion and superdiffusion in porous media. As a special case, a time-fractional Boussinesq equation is considered. In the mentioned papers above, either a numerical method based on finite difference approximation or a self-similar solution method is used. While the self-similar solution method has a restriction on the initial and boundary conditions, numerical methods based on finite difference approximation require solving a system of nonlinear algebraic equations which is very difficult to solve because solving such systems requires a good initial guess for the solution. The method that we introduce in this paper is memory efficient compared to the methods that use solving system of nonlinear algebraic equations. In addition, usage of nonlinear algebraic equations involves matrices of dimension $M \times N \times K$, where M, N, K are number of mesh points in x, y and t directions respectively. Whereas, our method uses matrices only as big as $M \times N$ in solution steps and finally piles them up which makes our method memory efficient and fast, see section 3 for further details and discussions.

This paper is organized as follows: In the next section, we formulate and linearize the problem and provide its analysis. In section 3, we present the numerical method. In particular, by introducing an iterative approach, we convert the fractional PDE with implicit boundary conditions into a linear one with explicit boundary conditions. We showed that, the resulting equations can be reduced to system fractional ODEs. We tested the numerical method with an example for the critical parameters of the method. In particular, we obtained the dependency of error, cpu time and other important numeric outcomes on the method parameters.

2. Formulation and analysis of the problem

In this section, we formulate and linearize the problem under consideration and provide its analysis. We consider the following problem:

$$\begin{cases} \frac{\partial^\beta}{\partial t^\beta} u = (d(u)u_x)_x + (d(u)u_y)_y + f(t, x, y), (t, x, y) \in \Omega_T, \\ u(0, x, y) = 0, (x, y) \in \Omega, \\ d(u)u_x(t, x, y) = g_2(t, y), (t, x, y) \in \Gamma_{2T}, \\ d(u)u_y(t, x, y) = g_1(t, x), (t, x, y) \in \Gamma_{1T}, \\ u(t, x, y) = 0, (t, x, y) \in \Gamma_{3T} \cup \Gamma_{4T}, \end{cases} \quad (7)$$

where $\beta \in (0, 1)$ is the order of the Caputo fractional time derivative, $\Omega := (0, 1) \times (0, 1)$, $\Omega_T := (0, T) \times \Omega$, $\Gamma_{iT} := (0, T) \times \Gamma_i, i = 1, 2, 3, 4$, $\Gamma_1 := (0, 1) \times \{1\}$, $\Gamma_2 := \{1\} \times (0, 1)$, $\Gamma_3 := (0, 1) \times \{0\}$, $\Gamma_4 := \{0\} \times (0, 1)$ and $T > 0$ is a final time. We assume that Ω is a bounded simply connected domain with a piece-wise smooth boundary $\partial\Omega = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$, $\Gamma_i \cap \Gamma_j = \emptyset, i \neq j$.

Definition 1. Let l be a closed interval. A set \mathbb{D} satisfying the following conditions is called the class of admissible coefficients for the problem (7) :

$$d \in C(l), \quad c_0 \leq d(s) \leq c_1, \quad \forall s \in l, \quad (8)$$

$$(d(u_1)\nabla u_1 - d(u_2)\nabla u_2) \cdot \nabla(u_1 - u_2) \geq c_2 \|\nabla(u_1 - u_2)\|^2, \quad \forall u_1, u_2 \in H_0^1(\Omega), \quad (9)$$

where c_0, c_1, c_2 are positive constants.

Definition 2. A weak solution of the problem (7) is a function $u \in S^\beta(\Omega_T) := L^2(0, T; H_0^1(\Omega)) \cap W_2^\beta(0, T; L^2(\Omega))$ such that the following integral identity holds for a.e. $t \in [0, T]$:

$$\begin{aligned} \int_{\Omega} \frac{\partial^\beta u}{\partial t^\beta} v \, dx \, dy + \int_{\Omega} d(u) \nabla u \cdot \nabla v \, dx \, dy \\ = \int_{\Omega} f v \, dx \, dy + \int_{\Gamma_{1T}} g_1 v \, dx \, dy + \int_{\Gamma_{2T}} g_2 v \, dx \, dy, \end{aligned} \quad (10)$$

for each $v \in S^\beta(\Omega_T)$, where

$$W_2^\beta(0, T) := \left\{ u \in L^2[0, T] : \frac{\partial^\beta u}{\partial t^\beta} \in L^2[0, T] \text{ and } u(0, \cdot) = 0 \right\},$$

is the fractional Sobolev space of order β . We note that $S^\beta(\Omega_T)$ is a Banach space with the norm :

$$\|u\|_{S^\beta(\Omega_T)} = \left(\|u\|_{W_2^\beta(0, T; L^2(\Omega))}^2 + \|u\|_{L^2(0, T; H_0^1(\Omega))}^2 \right)^{\frac{1}{2}}.$$

Theorem 1. [22] Let $d \in \mathbb{D}$. Then the direct problem (7) has a unique weak solution $u \in S^\beta(\Omega_T)$. Moreover, for a.e $t \in [0, T]$ there exist some constants $c, C > 0$ such that

$$\frac{\partial^\beta \|u\|^2}{\partial t^\beta} + c \|u\|_{H_0^1(\Omega)}^2 \leq C \left[\|f\|^2 + \|g_1\|_{L^2(\Gamma_1)}^2 + \|g_2\|_{L^2(\Gamma_2)}^2 \right].$$

We note that the conditions (8) and (9) arise in the solvability of the direct problem (7) and can be found in some papers, for example see the condition H3 in [12]. The main difficulty in solving the problem (7) numerically comes from the fact that the boundary conditions are given implicitly. That is, boundary conditions depends on the solution itself. Without knowing the values of the solution at the boundary, the explicit numeric approximation at the interior points is not possible. To overcome this difficulty, we approximate the solution using the following recurrent approximation scheme:

$$\begin{cases} \frac{\partial^\beta}{\partial t^\beta} u^n = (d(u^{n-1})u_x^n)_x + (d(u^{n-1})u_y^n)_y + f(t, x, y), (t, x, y) \in \Omega_T, \\ u^n(0, x, y) = 0, (x, y) \in \Omega, \\ d(u^{n-1})u_x^n(t, x, y) = g_2(y, t), (t, x, y) \in \Gamma_{2T}, \\ d(u^{n-1})u_y^n(t, x, y) = g_1(x, t), (t, x, y) \in \Gamma_{1T}, \\ u^n(t, x, y) = 0, (t, x, y) \in \Gamma_{3T} \cup \Gamma_{4T}. \end{cases} \tag{11}$$

Here, the first approximation $u^{(0)}(t, x, y)$ is given and $u^{(n)}(t, x, y)$ are the n^{th} iteration for $n = 1, 2, \dots$. In this way, the problem becomes linear. But, more importantly, the boundary conditions for the solution are now known. In [15], the authors linearized a problem for the following nonlinear equation:

$$\frac{\partial}{\partial t} u(t, x, y) - \nabla \cdot (f(T^2) \nabla u) = f(t, x, y), \tag{12}$$

where $T^2 = |\nabla u|^2$. They proved that the solution of the linearized problem converges to the solution of the associated nonlinear problem. By modifying the method used in [15] for the problem (7), it can be proved that the solution to the linearized problem (11) converges to the solution to the problem (7) under the conditions $d \in \mathbb{D}$, $F(t, x, y) \in L_2(\Omega_T)$, $g := (g_1(t, x), g_2(t, y)) \in L_2(\Gamma_{1T}) \times L_2(\Gamma_{2T})$.

3. Numerical solution to the fractional Richards equation

This section is devoted to numerical solution to the problem (7). We first illustrate the methodology on a simple equation and then apply the idea for the problem (7) [e.g. see 21]. The main idea was to convert the equation to a system of ordinary differential equations (ODEs) using method of lines and vectorization, and solve the resulting system of fractional ODEs. In addition to using the classical method of lines, we adopt operator approach to approximate derivatives which reduces computational and memory demand of the algorithm. For this purpose, we consider the discretization of the following equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(t, x, y), (t, x, y) \in \Omega_T. \tag{13}$$

Let $x_i = 0 + i\Delta x$, $i = 0, \dots, M$ and $y_j = 0 + j\Delta y$, $j = 0, \dots, N$ with $\Delta x = 1/M$ and $\Delta y = 1/N$. Also, let $u_{ij}(t)$ represent the solution at point (x_i, y_j) at a time $t > 0$. Then, the centered difference approximation of the time derivative at point (x_i, y_j) is

$$\frac{u_{ij}}{dt} = \frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{\Delta x^2} + \frac{u_{ij+1} - 2u_{ij} + u_{ij-1}}{\Delta y^2} + f(t, x_i, y_j),$$

with $i = 1, \dots, N - 1$ and $j = 1, \dots, M - 1$. This approximation can be vectorized by first defining the solution matrix at the interior points $[u_{ij}] = u(x_i, y_j)$ with $i = 1, \dots, N - 1$ and $j = 1, \dots, M - 1$. We define the left and right shift operators on the matrix $[u_{ij}]$ of solution approximations at the interior points as follows:

$$LS([u_{ij}]) = [u_{i+1j}] \text{ and } RS([u_{ij}]) = [u_{i-1j}], \tag{14}$$

with $i = 1, \dots, N - 1$ and $j = 1, \dots, M - 1$. Then, by using (14), (13) can be expressed in matrix form using the left and right shift operators in the following way:

$$\left[\frac{du_{ij}}{dt} \right] = \frac{[LS([u_{ij}]) - 2[u_{ij}] + RS([u_{ij}])]}{\Delta x^2} + \frac{[LS([u_{ij}]') - 2[u_{ij}] - RS([u_{ij}]')]}{\Delta y^2} + f(t, [X], [Y]),$$

where $[a_{ij}]'$ denotes the transpose of the matrix $[a_{ij}]$ and $[X]$ and $[Y]$ represent the matrices with $(i, j)_{th}$ entries given as $X(i, j) = x_i$ and $Y(i, j) = y_j$ for $i = 1, \dots, M - 1$ and $j = 1, \dots, N - 1$.

Next we consider the vectorization of the problem (11). For $n = 1$, and $u^{(0)}(t, x, y) = 0$, we have to solve the following linear problem to get the solution $u^{(1)}(t, x, y)$:

$$\begin{cases} \frac{\partial^\beta}{\partial t^\beta} u^{(1)} = (d(0)u_x^{(1)})_x + (d(0)u_y^{(1)})_y + f(t, x, y), (t, x, y) \in \Omega_T, \\ u^{(1)}(0, x, y) = 0, (x, y) \in \Omega, \\ -d(0)u_x^{(1)}(t, 1, y) = g_1(t, y), y \in (0, 1), t \in (0, T), \\ -d(0)u_y^{(1)}(t, x, 1) = g_2(t, x), y \in (0, 1), t \in (0, T), \\ u^{(1)}(t, 0, y) = 0, x \in (0, 1), t \in (0, T), \\ u^{(1)}(t, x, 0) = 0, x \in (0, 1), t \in (0, T). \end{cases} \tag{15}$$

The problem (15) is a fractional order and a linear problem on the given domain. Then, vectorized method of line approach described in (15) results in the following difference approximation:

$$\begin{cases} \left[\frac{d^\beta u_{ij}^{(1)}}{dt} \right] = d(0) \left[u_{xxij}^{(1)} \right] + d(0) \left[u_{yyij}^{(1)} \right] + f(t, X, Y), \\ \left[u_{ij}^{(1)}(0) \right] = [h(X, Y)]. \end{cases}$$

This is a system of linear fractional ODEs which we solve using a Matlab implementation of the Adam-Bashfort-Moulton (ABM) type predictor-corrector PECE method given in [1]. The ABM is a PECE (Predict-Evaluate-Correct-Evaluate) type method; that is, for the approximation of a first order ode of the form

$$\begin{cases} y' = f(t, y(t)), \\ y(0) = y_0, \end{cases} \quad (16)$$

with time approximation nodes t_j , and corresponding approximations, $y_j \cong y(t_j)$ at each j_{th} step, there are two approximations computed for the next node, namely, predictor, $yp(t_{j+1})$, and using the predictor, the corrector approximation $yc(t_{j+1})$ is obtained and used in the calculation. The error is obtained by finding the difference of predictor and corrector approximations. There are two main advantages of using PECE type compared to the classical equivalent-order Range-Kutta methods. The first of is the increased accuracy and stability, see [10], [5, Ch. 6]. For the ODEs with fractional derivatives, it was shown that the stability and accuracy remains high compared to equivalent-order numerical methods [2, 3, 6]. On the other hand, it is proven in [2] that, under the assumption that the right-hand side in the equation (17) are from $C^2[0, T]$ for some $T > 0$, then the error is of order $O(h^{1+\beta})$ for $\beta < 1$, and $O(h^2)$ for $\beta > 1$, respectively. The second advantage of using the PECE type numerical approximation is the fact that, this method can assume variable time steps that reduces the computational cost of the approximation. The method can control the time steps by using the difference between the corrector and the predictor approximations. When the difference is smaller than the desired level of accuracy with the current time step, this is used as an indication that the solver is in a non-stiff area, and time steps are increased in an adaptive manner.

The idea of combining the Method of Lines approach to reduce the given PDE to a system of ODEs and using shift operators in the evaluation of the RHS of the PDE has a two major advantages compared to similar operator approaches such as that of Podlubny's matrix operator approach [18, 19]. The first major advantage comes from the amount of the memory required to solve the system. Matrices required to compute the RHS function are of dimension $M - 1 \times N - 1$. Whereas, in matrix operator approach of the Podlubny, the three-diagonal matrices used for difference approximations are of dimension $M \times N \times K$. The second major advantage of the method applied here is that, the ease of usage in solving linear fractional PDE. This is because, even though the matrix operator approach of the Podlubny's method is very natural in dealing with the linear

PDEs, solving multi-variable ones requires one to solve algebraic equation of very high dimensions. This is another challenge that may require different approach depending on the memory demand involved in solving the given PDE. In addition, the introduced shift approach provides gains in calculations in terms of speed and memory compared to other difference calculations with loops and diagonal matrices.

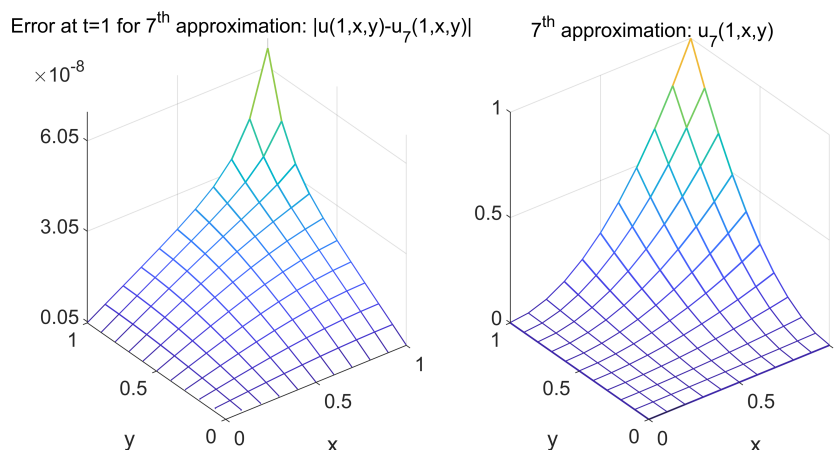


Figure 1: Approximate solution: The numerical solution and the error distribution is plotted at $t=1$ for $\Delta t = 10^{-4}$, $\alpha = 0.8$ and the relative error parameter $\epsilon = 10^{-6}$. The relative error level, $\epsilon = 10^{-6}$ could be achieved after 7 iterations.

Example. In this example, we solve the problem (7) for the water diffusivity $d(u) = 1/\sqrt{1+u}$, where $u(t, x, y) = tx^2y^2$. The function $f(t, x, y)$ is obtained by the substitution of the analytical solution to the equation. The Dirichlet type boundary conditions at $x = 0$ and $y = 0$ are obtained by the substitution of the corresponding values to the analytic solution as $u(t, 0, y) = 0$ and $u(t, x, 0) = 0$. The Neumann type boundary conditions at $x = 1$ and $y = 1$ are obtained from the equations:

$$\begin{cases} -d(u)u_x(t, 1, y) = -\frac{2ty^2}{\sqrt{1+tx^2y^2}}, \\ -d(u)u_y(t, x, 1) = -\frac{2tx^2}{\sqrt{1+tx^2y^2}}. \end{cases} \tag{17}$$

Unless otherwise stated, the parameters used for the numerical simulations are taken as $\Delta x = 0.1$ ($M = 10$), $\Delta y = 0.1$ ($N = 10$), $\Delta t = 10^{-6}$ and $\beta = 0.6$. The initial approximation function, $u^0(t, x, y)$, is taken to be zero. The absolute error estimations at n^{th} iteration are defined as the maximum of the difference between the approximating numerical solution and the analytic solution at time $t = 1$ on the approximation points on Ω . Namely, $\text{Error} = \max_{i,j} |u_{i,j}^k(1) - u(1, x_i, y_j)|$ for $i = 0, 1, \dots, M$ and $j = 0, 1, \dots, N$ with $u_{i,j}^k(t)$ representing the numerical approximation to the solution at the k^{th} iteration at time t . The relative error, RelError , is defined as the maximum of the difference between the consecutive iterations of the numerical solutions at time $t = 1$. That is,

RelError = $\max_{i,j} |u_{i,j}^k(1) - u_{i,j}^{k-1}(1)|$ for $k = 1, 2, \dots$ and $u_{i,j}^0$ being the value of the initial approximation function at (x_i, y_j) . Importantly, the iteration process stops when the relative error is less than the provided ϵ . We note that, we did simulations with $N = 20$, $M = 20$, but the simulation results did not improve at all. However, the cpu time increased substantially. In addition, the simulations run with different initial approximation function $u^0(t, x, y)$ other than zero also did not lead to any substantial improvements in the results. Hence, throughout all simulations we only consider $N = M = 10$ and $u^0(t, x, y) = 0$. In Figure 1, we simulated the model with $\epsilon = 0.0001$ to obtain the approximate solution. It took 5 steps to achieve the relative error of the differences to be less than the given ϵ . The error distribution at time $t = 1$ is given with the approximate solution. As seen from the Figure 1 left panel, the error distribution closely traces the approximate solution in terms of the magnitude. That is, error is large at the points where the solution is large.

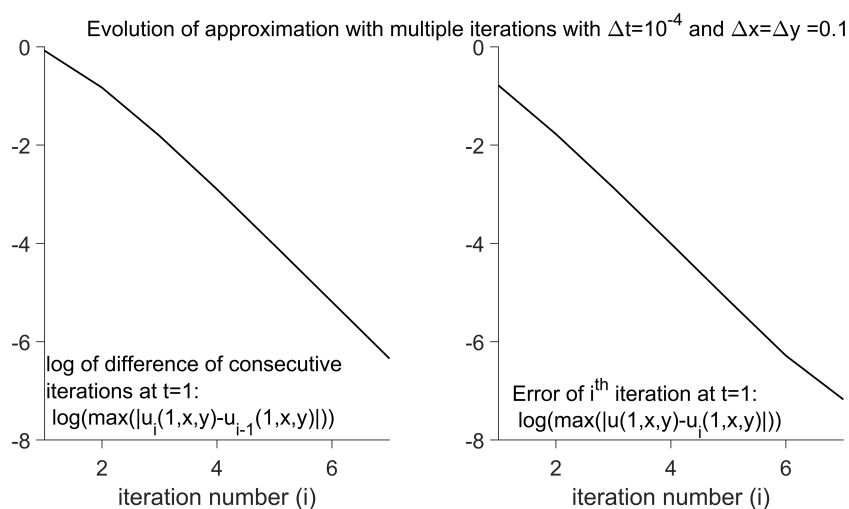


Figure 2: Absolute and relative error dependence to the iteration number. Both type of errors decay linearly in logarithm depending on the iteration number.

To observe the improvement in the error with respect to the iteration number, we next simulated the model and evaluated both absolute and relative errors in logarithmic function. The results are shown in Figure 2. It appears that, when the initial approximation function for the iteration is chosen as zero, both absolute and relative errors progress linearly with respect to the iteration number. This is important in the following way; instead of using refined step size to improve the error, we can increase and obtain much faster results with less error. This is analyzed in Figure 3 in which we consider the dependency of the results on both the time step and ϵ . Importantly, the improvement in the error with the refinement of time step size is very subtle compared to the improvement in error with ϵ . However, the computational cost of the refinement of the time step size is enormous. Finally, we considered the dependence of the error, cpu time and iteration number to the fractional derivative order α for the given $\epsilon = 0.0001$, see Figure 4. It appears that, the change in the error with decreasing α is subtle; whereas the computational cost

increases with reduction in alpha. Interestingly, number of iterations necessary to achieve the given relative error remains intact.

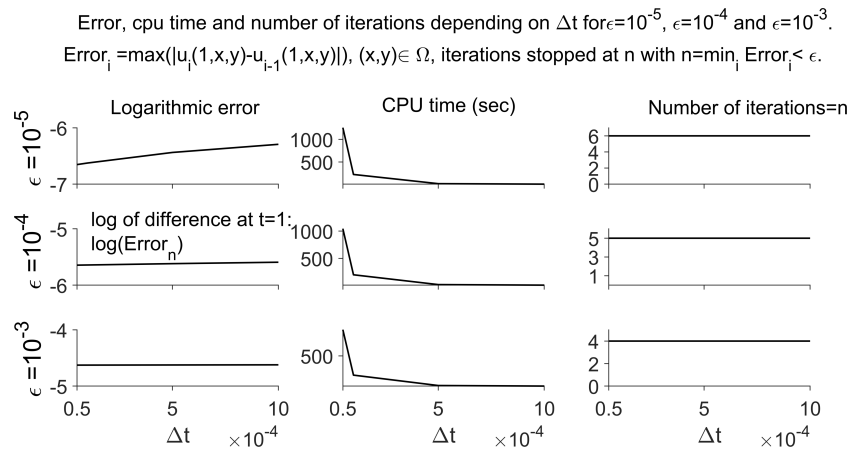


Figure 3: Absolute error, CPU time and iteration number dependence to both time step size and relative error bound ϵ

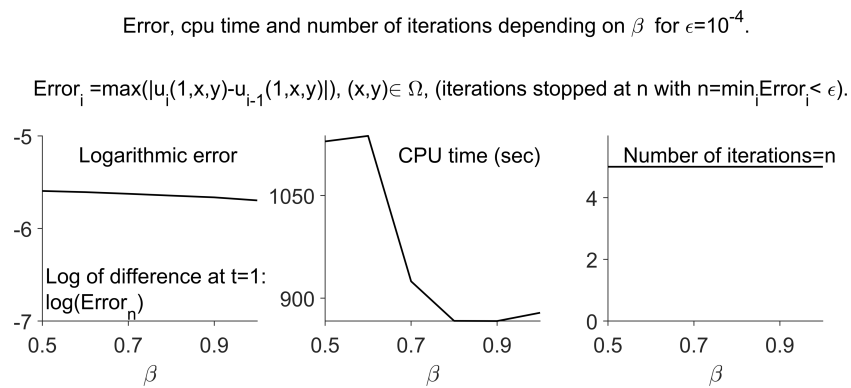


Figure 4: Absolute error, CPU time and iteration number dependence to the fractional derivative order β

References

- [1] Kendall Atkinson. *An introduction to numerical analysis*. John wiley & sons, 1991.
- [2] Kai Diethelm, Neville J Ford, and Alan D Freed. Detailed error analysis for a fractional adams method. *Numerical algorithms*, 36(1):31–52, 2004.
- [3] Kai Diethelm and Alan D Freed. The fracpece subroutine for the numerical solution of differential equations of fractional order. *Forschung und wissenschaftliches Rechnen*, 1999:57–71, 1998.
- [4] Abd El-Ghany El Abd and Jacek J Milczarek. Neutron radiography study of water absorption in porous building materials: anomalous diffusion analysis. *Journal of Physics D: Applied Physics*, 37(16):2305, 2004.
- [5] Hairer Ernst and Wanner Gerhard. *Solving ordinary differential equations ii: Stiff and differentialalgebraic problems*, 1996.
- [6] Roberto Garrappa. On linear stability of predictor–corrector algorithms for fractional differential equations. *International Journal of Computer Mathematics*, 87(10):2281–2290, 2010.
- [7] DN Gerasimov, VA Kondratieva, and OA Sinkevich. An anomalous non-self-similar infiltration and fractional diffusion equation. *Physica D: Nonlinear Phenomena*, 239(16):1593–1597, 2010.
- [8] E Gerolymatou, I Vardoulakis, and R Hilfer. Modelling infiltration by means of a non-linear fractional diffusion model. *Journal of Physics D: Applied Physics*, 39(18):4104, 2006.
- [9] Ivan A Guerrini and D Swartzendruber. Soil water diffusivity as explicitly dependent on both time and water content. *Soil Science Society of America Journal*, 56(2):335–340, 1992.
- [10] Arieh Iserles. *A first course in the numerical analysis of differential equations*. Number 44. Cambridge university press, 2009.
- [11] Anatoliĭ Aleksandrovich Kilbas, Hari M Srivastava, and Juan J Trujillo. *Theory and applications of fractional differential equations*, volume 204. elsevier, 2006.
- [12] J Manimaran, L Shangerganesh, A Debbouche, and J-C Cortés. A time-fractional hiv infection model with nonlinear diffusion. *Results in Physics*, 25:104293, 2021.
- [13] Yakov Pachepsky, Dennis Timlin, and Walter Rawls. Generalized richards’ equation to simulate water transport in unsaturated soils. *Journal of Hydrology*, 272(1-4):3–13, 2003.

- [14] Yakov Pachepsky, Dennis Timlin, and Walter Rawls. Generalized richards' equation to simulate water transport in unsaturated soils. *Journal of Hydrology*, 272(1-4):3–13, 2003.
- [15] Burhan Pektaş and Ege Tamci. The heat flux identification problem for a nonlinear parabolic equation in 2d. *Journal of Computational and Applied Mathematics*, 312:134–142, 2017.
- [16] Łukasz Płociniczak. Analytical studies of a time-fractional porous medium equation. derivation, approximation and applications. *Communications in Nonlinear Science and Numerical Simulation*, 24(1-3):169–183, 2015.
- [17] I Podlubny. Fractional differential equations academic press. *San Diego, Boston*, 6, 1999.
- [18] Igor Podlubny. Matrix approach to discrete fractional calculus. *Fractional calculus and applied analysis*, 3(4):359–386, 2000.
- [19] Igor Podlubny, Aleksei Chechkin, Tomas Skovranek, YangQuan Chen, and Blas M Vinagre Jara. Matrix approach to discrete fractional calculus ii: partial fractional differential equations. *Journal of Computational Physics*, 228(8):3137–3153, 2009.
- [20] Shujun Shen, Fawang Liu, Q Liu, and Vo Anh. Numerical simulation of anomalous infiltration in porous media. *Numerical Algorithms*, 68(3):443–454, 2015.
- [21] Salih Tatar, Ramazan Tnaztepe, and Mustafa Zeki. Numerical solutions of direct and inverse problems for a time fractional viscoelastoplastic equation. *Journal of Engineering Mechanics*, 143(7):04017035, 2017.
- [22] Salih Tatar and Süleyman Ulusoy. An inverse problem for a nonlinear diffusion equation with time-fractional derivative. *Journal of Inverse and Ill-posed Problems*, 25(2):185–193, 2017.