

Research Article

Application of Nonlinear Time-Fractional Partial Differential Equations to Image Processing via Hybrid Laplace Transform Method

B. A. Jacobs ^{1,2,3} and C. Harley ^{1,2}

¹School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, Private Bag 3, Wits 2050, South Africa

²DST-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS), South Africa

³School of Mathematics and Statistics, UNSW Australia, Sydney, NSW 2052, Australia

Correspondence should be addressed to B. A. Jacobs; byron.jacobs@wits.ac.za

Received 6 April 2018; Accepted 30 August 2018; Published 16 September 2018

Academic Editor: Morteza Khodabin

Copyright © 2018 B. A. Jacobs and C. Harley. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work considers a hybrid solution method for the time-fractional diffusion model with a cubic nonlinear source term in one and two dimensions. Both Dirichlet and Neumann boundary conditions are considered for each dimensional case. The hybrid method involves a Laplace transformation in the temporal domain which is numerically inverted, and Chebyshev collocation is employed in the spatial domain due to its increased accuracy over a standard finite-difference discretization. Due to the fractional-order derivative we are only able to compare the accuracy of this method with Mathematica's `NDSolve` in the case of integer derivatives; however, a detailed discussion of the merits and shortcomings of the proposed hybridization is presented. An application to image processing via a finite-difference discretization is included in order to substantiate the application of this method.

1. Introduction

This work examines the performance of a hybrid Laplace transform–Chebyshev collocation technique applied to the time-fractional diffusion equation in two dimensions with a nonlinear source term:

$$u_{t^\alpha} = c_d (u_{xx} + u_{yy}) + c_s u (1 - u) (u - a) \quad (1)$$

where

$$\begin{aligned} u_{t^\alpha} &= \frac{\partial^\alpha u}{\partial t^\alpha}, \\ u_{xx} &= \frac{\partial^2 u}{\partial x^2}, \\ u_{yy} &= \frac{\partial^2 u}{\partial y^2}. \end{aligned} \quad (2)$$

This model is explored subject to both Dirichlet and Neumann boundary conditions on the bounded domain $\Omega =$

$[-1, 1] \times [-1, 1]$ to satisfy the domain required by the Chebyshev polynomials with initial condition $u(x, y, 0) = f(x, y)$. This method benefits from the analyticity of the Laplace transform and efficient numerical inversion of this transform, an accurate discretization approach through Chebyshev collocation, and a convergent linearization technique, which results in a robust method for solving nonlinear time-fractional partial differential equations on a bounded domain. Following the method detailed by Jacobs and Harley in [1, 2] we also employ a finite-difference discretization in applying this method to images. This is elaborated on further in Section 5.

Recently fractional derivatives and fractional partial differential equations (FPDEs) have received great attention both in analysis and application (see [3–5] and references therein). Despite this large effort, very little attention has been paid to solving FPDEs on a bounded domain through transformation techniques. Agrawal [6] makes use of the Laplace transform and the finite sine transform to obtain an

analytic solution to the fractional diffusion-wave equation on a bounded domain. Other techniques such as the variational iteration method, Adomian decomposition method, and differential transform method have all been applied to fractional partial differential equations but with a focus on unbounded domains. More recently Hashemizadeh and Ebrahimzadeh [7] and Zaky et al. [8] present solutions to the linear and nonlinear time-fractional Klein-Gordon equations, respectively. Their solutions were obtained by using an operational matrix approach and Legendre polynomial interpolation.

The linear diffusion model has been applied in many different fields including image processing [9–14]. However, to the best of the authors' knowledge, the application of a time-fractional partial differential equation has not yet been thoroughly examined within such a context. It is from the standpoint of applying a fractional partial differential equation to an image that we examine the time-fractional diffusion equation in two dimensions. In a recent paper Jacobs and Momoniat [15, 16] show that the diffusion equation with this nonlinear source term is able to binarize a document image with great success. In extending this concept to fractional-order derivatives, we seek to preserve the symmetry of the diffusion equation. Values of $\alpha \in (0, 1]$ resolve the diffusion equation to subdiffusion, which preserves symmetry and exhibits some interesting dynamics which are discussed later on in this work. Values of $\alpha > 1$ introduce a transport effect which then breaks symmetry. It is for this reason that we impose the restriction, $\alpha \in (0, 1]$.

In this work we make use of the Laplace transform which allows us to handle the fractional-order derivative in an algebraic way and incur no error in doing so. Inversion of the Laplace transform is however difficult to obtain analytically. We therefore make extensive use of the numerical inversion procedure described by Weideman and Trefethen in [17] which defines a contour of integration that maps the domain of the Bromwich integral from the entire complex space to the real space, from which we can approximate this integral with a trapezoidal rule.

With a robust method for inverting the Laplace transform we may then hybridize the transform with a discretization technique. The Laplace transform of the temporal variable avoids the need for a time-marching scheme as well as reducing the fractional-order derivative to an algebraic expression. The transformed model is then discretized by the use of Chebyshev collocation due to its superior performance to finite differences as illustrated in [1, 2]. The resulting system is solved and the transform inverted to obtain a semianalytic solution, continuous in time and discrete in space.

In the following section we present some preliminary results which are put to use throughout the paper. In Section 3 the implemented methods are described, including the different cases for boundary conditions. Section 4 presents the solutions obtained by the proposed method as well as an error comparison with `Mathematica`'s `NDSolve` for the one- and two-dimensional cases of our model as well as both Dirichlet and Neumann boundary conditions. In Section 5 we provide a real-world application of this method and model in the form of document image binarization, illustrating the ability to obtain a useful result with the present method when coupled

with the finite-difference discretization. A discussion of the results and their relationship to work beyond this research is presented in Section 6 and some concluding remarks are made in Section 7.

2. Preliminaries

In this work we employ Caputo's definition of a fractional derivative over the Riemann-Liouville derivative due to the fact that the Caputo derivative makes use of the physical boundary conditions, whereas the Riemann-Liouville derivative requires fractional-order boundary conditions.

Definition 1. The Riemann-Liouville integral of order $\alpha > 0$ of a function $u(t)$ is

$$J^\alpha u(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} u(\tau) d\tau, \quad x > 0. \quad (3)$$

Definition 2. The fractional derivative of $u(t)$ according to the Caputo definition with $m - 1 < \alpha \leq m$, $m \in \mathbb{N}$, is

$$\begin{aligned} \frac{\partial^\alpha u(t)}{\partial t^\alpha} &= J^{m-\alpha} D^m u(t) = D_*^\alpha u(t) \\ &:= \frac{1}{\Gamma(m-\alpha)} \int_0^t (t - \tau)^{m-\alpha-1} u^{(m)}(\tau) d\tau. \end{aligned} \quad (4)$$

If $\alpha \in \mathbb{Z}$, the Caputo fractional derivative reduces to the ordinary derivative or integral. Podlubny [4] illustrates the pleasing property of the Laplace transform of a Caputo derivative, as can be seen in (5). In our case where $0 < \alpha < 1$ we have

$$\mathcal{L} \left\{ \frac{\partial^\alpha u(x, y, t)}{\partial t^\alpha} \right\} = s^\alpha U(x, y) - s^{\alpha-1} u(x, y, 0). \quad (5)$$

This property allows one to treat fractional-order derivatives algebraically.

Definition 3. The Generalized Mittag-Leffler function of the argument z is

$$E_{\alpha, \beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k\alpha + \beta)}. \quad (6)$$

The use of Laplace transform allows one to circumvent the problems that arise in the time-domain discretization. However, using the Laplace transform for the fractional-order derivative presents the problem of inverting the transform to find a solution. Analytic inversion of the transform is infeasible and hence the numerical scheme for evaluating the Bromwich integral presented by Weideman and Trefethen in [17] is put to extensive use. The Bromwich integral is of the form

$$u(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} e^{st} U(s) ds, \quad \sigma > \sigma_0, \quad (7)$$

where σ_0 is the convergence abscissa. Using the parabolic conformal map

$$s = \mu(iw + 1)^2, \quad -\infty < w < \infty, \quad (8)$$

(7) becomes

$$u(t) = \frac{1}{2\pi i} \int_{-\infty}^{\infty} e^{s(w)t} U(s(w)) s'(w) dw \quad (9)$$

which can then be approximated simply by the trapezoidal rule, or any other quadrature technique, as

$$u_{n;N_i}(t) = \frac{h}{2\pi i} \sum_{k=-N_i}^{N_i} e^{s(w_k)t} U(s(w_k)) s'(w_k), \quad (10)$$

$$w_k = kh.$$

On the parabolic contour the exponential factor in (7) forces rapid decay in the integrand making it amenable to quadrature. All that is left is to choose the parameters h and μ optimally which is done by asymptotically balancing the truncation error and discretization errors in each of the half-planes. The methodology described by Weideman and Trefethen achieves near optimal results (see [17] and figures therein) and the interested reader is directed there for a thorough description of the implementation of the method. In this work we make use of the parabolic contour due to the ease of use and the hyperbolic contour only exhibits a slight improvement in performance over the parabolic contour.

3. Methods

This section introduces the methodologies used for the two-dimensional model previously presented. We may write our model as

$$c_d u_{xx} + c_d u_{yy} + c_s (a+1) u^2 - c_s a u - c_s u^3 - u_{t^\alpha} = \Phi(u_{xx}, u_{yy}, u, u_{t^\alpha}) = 0. \quad (11)$$

The quasi-linearization technique can be viewed as a generalized Newton-Raphson method in functional space. An iterative scheme is constructed creating a sequence of linear equations that approximate the nonlinear equation (11) and boundary conditions. Furthermore, this sequence of solutions converges quadratically and monotonically [18–20]. The quasi-linear form is given by

$$c_d u_{xx}^{n+1} + c_d u_{yy}^{n+1} + X_1^n u^{n+1} + X_2^n u_{t^\alpha}^{n+1} = X_3^n, \quad (12)$$

where n indicates the index of successive approximation. Moreover, $u(x, y, t)^n$ is known entirely at the explicit index n . The coefficients are given by

$$X_1^n = \frac{\partial \Phi}{\partial u} = c_s 2(a+1) u^n - c_s a - 3c_s (u^n)^2, \quad (13)$$

and

$$X_2^n = \frac{\partial \Phi}{\partial u_{t^\alpha}} = -1. \quad (14)$$

If the elements in $\underline{u} = (u_{xx}, u_{yy}, u, u_{t^\alpha})$ are indexed by j , then for the n th iteration in the quasi-linearization we have

$$X_3^n = \sum_j \underline{u}_j^n \frac{\partial \Phi^n}{\partial \underline{u}_j},$$

$$= \left[u_{xx} \frac{\partial \Phi}{\partial u_{xx}} + u_{yy} \frac{\partial \Phi}{\partial u_{yy}} + u \frac{\partial \Phi}{\partial u} + u_{t^\alpha} \frac{\partial \Phi}{\partial u_{t^\alpha}} \right]^n, \quad (15)$$

$$= \left[c_d u_{xx} + c_d u_{yy} + 2(a+1) u^2 - a u - 3u^3 - (u_{t^\alpha}) \right]^n,$$

$$= \left[c_d u_{xx} + c_d u_{yy} + (a+1) u^2 - a u - u^3 - (u_{t^\alpha}) \right]^n$$

$$+ \left[(a+1) u^2 - 2u^3 \right]^n.$$

Since the first term above satisfies (11), it is replaced with 0 giving

$$X_3^n = \left[(a+1) u^2 - 2u^3 \right]^n. \quad (16)$$

Equation (12) can now be transformed by the Laplace transform, a linear operator, to obtain

$$c_d U_{xx}^{n+1} + c_d U_{yy}^{n+1} + X_1^n U^{n+1} - s^\alpha U^{n+1} = \frac{X_3^n}{s} - s^{\alpha-1} f(x, y), \quad (17)$$

where

$$U(x, y, s) = \mathcal{L}\{u(x, y, t)\}. \quad (18)$$

Equation (17) may be discretized by Chebyshev collocation, described in the following section.

3.1. Chebyshev Collocation. Chebyshev polynomials form a basis on $[-1, 1]$ and hence we dictate the domain of our PDE to be Ω . We note here, however, that any domain in \mathbb{R}^2 can be trivially deformed to match Ω . We discretize our spatial domain using Chebyshev-Gauss-Lobatto points:

$$x_i = \cos\left(\frac{i\pi}{N_x}\right), \quad i = 0, 1, \dots, N_x, \quad (19)$$

$$y_j = \cos\left(\frac{j\pi}{N_y}\right), \quad j = 0, 1, \dots, N_y.$$

Given this choice of spatial discretization, we have $x_0 = 1$, $x_{N_x} = -1$, $y_0 = 1$, and $y_{N_y} = -1$, indicating that the domain is in essence reversed and one must exercise caution when imposing the boundary conditions.

In mapping our domain to Ω we may assume that $N_x = N_y$; i.e., we have equal number of collocation points in each spatial direction. We now define a differentiation matrix $D^{(1)} = d_{kl}$:

$$d_{kl} = \begin{cases} \frac{c_k (-1)^{k+l}}{c_l (x_k - x_l)}, & k \neq l, \\ -\frac{x_k}{2(1-x_k^2)}, & k = l, \\ \frac{1}{6}(2N_x^2 + 1), & k = l = 0, \\ -\frac{1}{6}(2N_x^2 + 1), & k = l = N_x, \end{cases} \quad (20)$$

$$\text{where } c_k = \begin{cases} 2, & k = 0, N_x, \\ 1, & k = 1, \dots, N_x - 1. \end{cases}$$

Bayliss et al. [21] describe a method for minimizing the round-off errors incurred in the calculations of higher order differentiation matrices. Since we write $D^{(2)} = D^{(1)}D^{(1)}$, we implement the method, described in [21], in order to minimize propagation of round-off errors for the second derivative in space.

The derivative matrices in the x direction are

$$\begin{aligned} \widehat{D}_x^{(1)} &= D^{(1)}, \\ \widehat{D}_x^{(2)} &= D^{(2)}, \end{aligned} \quad (21)$$

where $D^{(1)}$ is the Chebyshev differentiation matrix of size $(N_x + 1) \times (N_y + 1)$.

Because we have assumed $N_x = N_y$, we derive the pleasing property that $\widehat{D}_y^{(1)} = (\widehat{D}_x^{(1)})^T$ and $\widehat{D}_y^{(2)} = (\widehat{D}_x^{(2)})^T$.

Writing the discretization of (17) in matrix form yields

$$\widehat{D}_x^{(2)} \mathbf{U}^{n+1} + \mathbf{U}^{n+1} \widehat{D}_y^{(2)} + \mathbf{X}_1^n \mathbf{U}^{n+1} - s^\alpha \mathbf{U}^{n+1} = \frac{\mathbf{X}_3^n}{s} - \mathbf{F}, \quad (22)$$

where

$$F_{ij} = s^{\alpha-1} f(x_i, y_j). \quad (23)$$

By expanding (22) in summation notation, we have

$$\begin{aligned} & \sum_{k=0}^{N_x} d_{ik}^{(2)} U^{n+1}(x_k, y_j) + \sum_{k=0}^{N_y} d_{kj}^{(2)} U^{n+1}(x_i, y_k) \\ & + [X_1]_{i,j}^n U^{n+1}(x_i, y_j) - s^\alpha U^{n+1}(x_i, y_j) \\ & = \frac{[X_3]_{i,j}^n}{s} - s^{\alpha-1} f(x_i, y_j), \end{aligned} \quad (24)$$

for $i = 0, 1, \dots, N_x, j = 0, 1, \dots, N_y$. By extracting the first and last terms in sums, we obtain

$$\begin{aligned} & d_{i0}^{(2)} U^{n+1}(x_0, y_j) + d_{iN_x}^{(2)} U^{n+1}(x_{N_x}, y_j) \\ & + d_{0j}^{(2)} U^{n+1}(x_i, y_0) + d_{N_y j}^{(2)} U^{n+1}(x_i, y_{N_y}) \\ & + \sum_{k=1}^{N_x-1} d_{ik}^{(2)} U^{n+1}(x_k, y_j) + \sum_{k=1}^{N_y-1} d_{kj}^{(2)} U^{n+1}(x_i, y_k) \\ & + [X_1]_{i,j}^n U^{n+1}(x_i, y_j) - s^\alpha U^{n+1}(x_i, y_j) \\ & = \frac{[X_3]_{i,j}^n}{s} - s^{\alpha-1} f(x_i, y_j), \end{aligned} \quad (25)$$

for $i = 1, \dots, N_x - 1, j = 1, \dots, N_y - 1$. We use the form of (25) to impose the boundary conditions.

The solution $\widetilde{\mathbf{U}} = \{U(x_1, y_1), U(x_1, y_2), \dots, U(x_{N_x-1}, y_{N_y-1})\}$, which is the matrix of unknown interior points of \mathbf{U} , can be found by solving the system

$$(\widetilde{\mathbf{A}} - s^\alpha \mathbf{I}) \widetilde{\mathbf{U}} + \mathbf{X}_1^n \circ \widetilde{\mathbf{U}} + \widetilde{\mathbf{U}} \widetilde{\mathbf{B}} = \frac{\mathbf{X}_3^n}{s} - \widetilde{\mathbf{F}}, \quad (26)$$

where $\widetilde{\mathbf{A}}$ is the matrix of interior points of $\widehat{D}_x^{(2)}$ and $\widetilde{\mathbf{B}}$ is the matrix of interior points of $\widehat{D}_y^{(2)}$, so that $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$ match the dimensions of $\widetilde{\mathbf{U}}$. We also use \circ to denote the Hadamard product between two matrices. Also

$$\begin{aligned} \widetilde{F}_{ij} &= s^{\alpha-1} f(x_i, y_j) + d_{i0}^{(2)} U(x_0, y_j) \\ & + d_{iN_x}^{(2)} U(x_{N_x}, y_j) + d_{0j}^{(2)} U(x_i, y_0) \\ & + d_{N_y j}^{(2)} U(x_i, y_{N_y}), \end{aligned} \quad (27)$$

for $i = 1, \dots, N_x - 1, j = 1, \dots, N_y - 1$.

3.1.1. Dirichlet Boundary Conditions. Boundary conditions may be in the form of Dirichlet conditions,

$$u(-1, y, t) = a, \quad (28)$$

$$u(1, y, t) = b,$$

$$u(x, -1, t) = c, \quad (29)$$

$$u(x, 1, t) = d,$$

and hence,

$$U(-1, y) = \mathcal{L}\{a\}, \quad (30)$$

$$U(1, y) = \mathcal{L}\{b\},$$

$$U(x, -1) = \mathcal{L}\{c\}, \quad (31)$$

$$U(x, 1) = \mathcal{L}\{d\}.$$

The parameters a, b, c , and d are potentially functions of the temporal variable and one of the spatial variables; i.e., $a = a(y, t)$. We assume that a, b, c , and d are constant.

Dirichlet boundary conditions can be imposed directly by substituting (28) and (29) into (25) and collecting all the known terms in $\widetilde{\mathbf{F}}$.

3.1.2. Neumann Boundary Conditions. Alternatively Neumann boundary conditions give

$$u_x(-1, y, t) = a, \quad (32)$$

$$u_x(1, y, t) = b,$$

$$u_y(x, -1, t) = c, \quad (33)$$

$$u_y(x, 1, t) = d,$$

with,

$$U_x(-1, y) = \mathcal{L}\{a\}, \quad (34)$$

$$U_y(1, y) = \mathcal{L}\{b\},$$

$$U_y(x, -1) = \mathcal{L}\{c\}, \quad (35)$$

$$U_y(x, 1) = \mathcal{L}\{d\}.$$

Neumann boundary conditions given by (30) are discretized as

$$\frac{\partial U}{\partial x}(x_{N_x} = -1, y_j) \approx \sum_{k=0}^{N_x} d_{N_x k} U^n(x_k, y_j) = \mathcal{L}\{a\}, \quad (36)$$

$$\frac{\partial U}{\partial x}(x_0 = 1, y_j) \approx \sum_{k=0}^{N_x} d_{0k} U^n(x_k, y_j) = \mathcal{L}\{b\}. \quad (37)$$

Similarly for equations (31)

$$\frac{\partial U}{\partial y}(x_i, y_{N_y} = -1) \approx \sum_{k=0}^{N_y} d_{kN_y} U^n(x_i, y_k) = \mathcal{L}\{c\}, \quad (38)$$

$$\frac{\partial U}{\partial y}(x_i, y_0 = 1) \approx \sum_{k=0}^{N_y} d_{k0} U^n(x_i, y_k) = \mathcal{L}\{d\}. \quad (39)$$

By extracting the first and last terms in the sum, the discretizations can be written as

$$\begin{pmatrix} d_{N_x 0} & d_{N_x N_x} \\ d_{00} & d_{0N_x} \end{pmatrix} \begin{pmatrix} U^n(x_0, y_j) \\ U^n(x_{N_x}, y_j) \end{pmatrix} \\ = \begin{pmatrix} \mathcal{L}\{b\} - \sum_{k=1}^{N_x-1} d_{0k} U^n(x_k, y_j) \\ \mathcal{L}\{a\} - \sum_{k=1}^{N_x-1} d_{N_x k} U^n(x_k, y_j) \end{pmatrix} \quad (40)$$

and

$$\begin{pmatrix} d_{0N_y} & d_{N_y N_y} \\ d_{00} & d_{N_y 0} \end{pmatrix} \begin{pmatrix} U^n(x_i, y_0) \\ U^n(x_i, y_{N_y}) \end{pmatrix} \\ = \begin{pmatrix} \mathcal{L}\{d\} - \sum_{k=1}^{N_y-1} d_{k0} U^n(x_i, y_k) \\ \mathcal{L}\{c\} - \sum_{k=1}^{N_y-1} d_{kN_y} U^n(x_i, y_k) \end{pmatrix} \quad (41)$$

and the solutions to these linear systems are then substituted into equation (25).

4. Results

In this section we consider only the results obtained by Chebyshev collocation due to the enormous increase in accuracy obtained over the finite-difference method that was presented by the authors in [1, 2].

TABLE 1: Maximum absolute error in the presented method's solution of the problem described by Example 1 at time $t = 0.5$.

N_x	Error vs NDSolve
5	0.029145584018683945
10	0.00015607552473428932
15	0.00014992336248420557
20	0.00014949081096204964
25	0.0001494889958276735
30	0.00014948859870700382

4.1. *Example 1.* The first example we consider is

$$u(x, 0) = e^{-x^2}, \quad (42)$$

with Dirichlet boundary conditions consistent with this initial condition,

$$u(-1, t) = u(1, t) = e^{-1}. \quad (43)$$

Table 1 illustrates the maximum absolute error between the solution obtained by the present method and the solution obtained by NDSolve in Mathematica 9 for $\alpha = 1$. To the best of the authors' knowledge, no exact solution exists for the fractional case and, hence, no comparison can be made.

However, we present Figure 1 which illustrates the behaviour of the mid-point of the solution as time evolves for various values of α .

4.2. *Example 2.* We now consider the one-dimensional case with initial condition

$$u(x, 0) = e^{-(x+1)} \quad (44)$$

and Neumann Boundary conditions

$$u_x(-1, t) = -1 \quad (45)$$

$$u_x(1, t) = \frac{-1}{e^2}. \quad (46)$$

Once again our solution converges with N moving from 5 to 10 but does not continue to improve. Table 2 presents the errors in our method when compared to NDSolve for $\alpha = 1$. We present the behaviour of the mid-point $u(x = 0, t)$ as time evolves for different values of α in Figure 2.

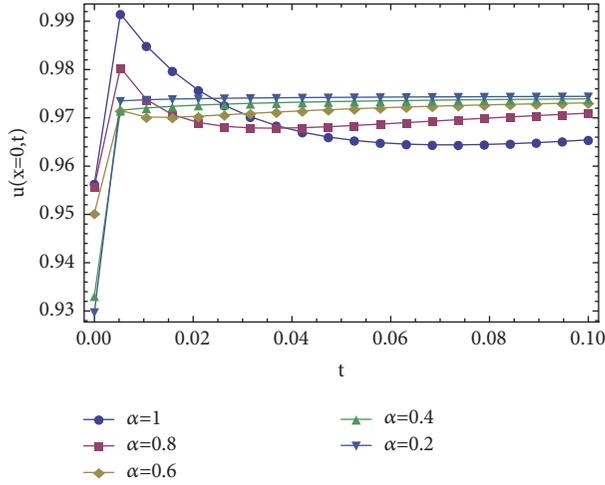
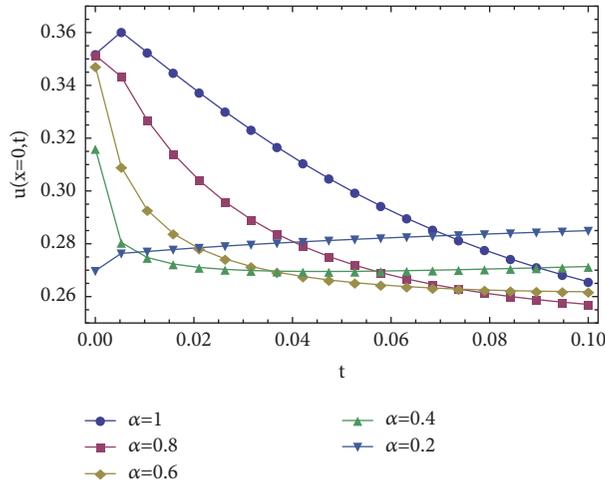
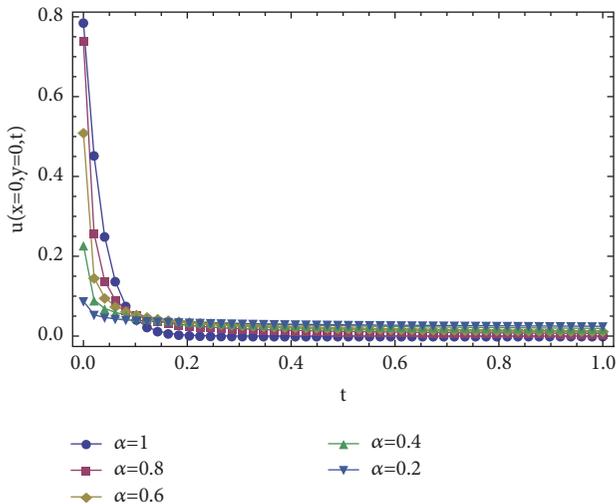
4.3. *Example 3.* This example considers the two-dimensional case with initial condition

$$u(x, y, 0) = (x - 1)(x + 1)(y - 1)(y + 1) \quad (47)$$

and consistent Dirichlet boundary conditions described as

$$u(-1, y, t) = u(1, y, t) = u(x, -1, t) = u(x, 1, t) = 0. \quad (48)$$

Table 3 presents the errors in the present method when compared to NDSolve. Figure 3 describes the evolution of the mid-point in two dimensions with time for various α values.

FIGURE 1: Plot of $u(x = 0, t)$ for various values of α .FIGURE 2: Plot of $u(x = 0, t)$ for various values of α .FIGURE 3: Plot of $u(x = 0, y = 0, t)$ for various values of α .TABLE 2: Maximum absolute error in the presented method's solution of the problem described by Example 2 at time $t = 0.5$.

N_x	Error vs NDSolve
5	0.18888552094993993
10	0.06402298769357995
15	0.06716933046894086
20	0.06886100702660541
25	0.0685232103205754
30	0.06782287252683938

TABLE 3: Maximum absolute error in the presented method's solution of the problem described by Example 3 at time $t = 0.5$.

$N_x = N_y$	Error vs NDSolve
5	0.000050581035499991776
10	0.00006478718535577835
15	0.00006310249073780343
20	0.00006478739642413641
25	0.0000642415234639226
30	0.00006478739639844785

TABLE 4: Maximum absolute error in the presented method's solution of the problem described by Example 4 at time $t = 1$.

$N_x = N_y$	Error vs NDSolve
5	0.00875011619006949
10	0.002380333748528196
15	0.002658963151695559
20	0.0027658847054590208
25	0.002819390822498269
30	0.0028497574705675377

4.4. *Example 4.* Finally we consider the case of a two-dimensional initial condition with Neumann conditions,

$$u(x, y, 0) = \cos(x) \cos(y) \quad (49)$$

and

$$u_x(-1, y, t) = \sin(1) \cos(y) \quad (50)$$

$$u_x(1, y, t) = -\sin(1) \cos(y) \quad (51)$$

$$u_y(x, -1, t) = \cos(x) \sin(1) \quad (52)$$

$$u_y(x, 1, t) = -\cos(x) \sin(1). \quad (53)$$

The errors for this example when compared to NDSolve are presented in Table 4. Figure 4 describes the evolution of the mid-point in two dimensions with time for various α values.

5. Image Processing Application

Despite the impressive accuracy obtained by Chebyshev collocation for small values of N_x , the method suffers from severe round-off error for values of $N_x > 100$ due to finite-precision arithmetic [22, 23]. In applying this scheme to

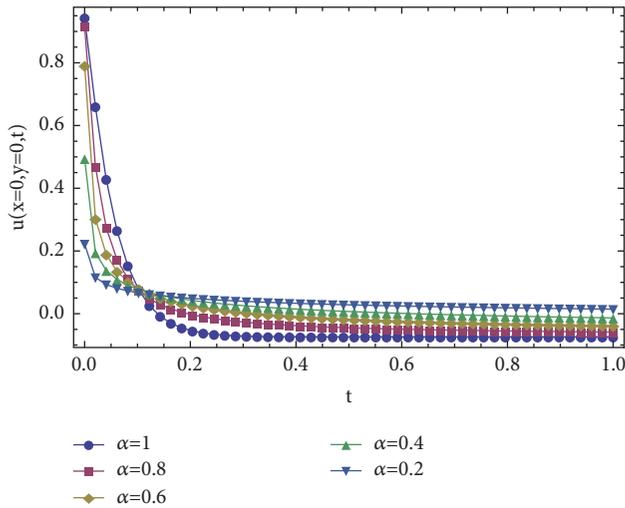


FIGURE 4: Plot of $u(x = 0, y = 0, t)$ for various values of α .

an image, the dimensions of the input image dictate the resolution of the scheme, and in most cases the input image dimensions will exceed 100. This forces us to employ a finite-difference discretization scheme instead of the Chebyshev scheme as described by Jacobs and Harley in [2]. Jacobs and Momoniat [15] present the results of applying the integer order model to an image for the purpose of document image binarization. We present here some examples of the results obtained using the hybrid-transform method. The power of the hybrid-transform method lies in the ability to immediately determine the solution at any point in time as opposed to needing to iterate to a given point in time. Figure 5 illustrates the solutions obtained, given an input image, at varying points in time. Figure 6 shows the effects of the fractional term on the resulting image, wherein the steady state has been altered as well as the transient phase being executed much more quickly. This mirrors the effects illustrated in the examples above, Figures 1, 2, 3, and 4, where the steady state of the equation is altered with changing values of α as well as the equation reaching a steady state more rapidly. This suggests the order of the derivative may be used as an extra dimension of control in image processing, where a smaller α value actually brightens the background of the image; it also reduces diffusivity of the model by altering the steady state away from a uniformly diffused image.

If we take, for example, $N_x = N_y = 100$ then the stability criteria for a standard time-marching scheme would be $\Delta t = (1/100)^{2/\alpha}$. As α decreases from 1, Δt becomes so small that the number of iterations required to attain a final time of $t = 0.003$ is unfeasibly large. This further justifies the use of a hybrid-transform method in the application of image processing.

5.1. Application Example. We present here a document image binarization application, where the input image is corrupted by noise as well as ink bleed-through from the reverse side of the page. The solution image we seek has black pixels which represent text and white pixels everywhere else. Figure 7

illustrates the input image and the resulting binary images for different values of α as well as the ground truth image against which we may quantitatively measure our result. The L_1 norm is used as an error measure and is presented at various points in time in Figure 8. Due to the changing α values we must normalize time, to accurately compare the performance of the present method over various α values. Due to the accelerated convergence to the steady state with varying α , as illustrated in the examples in Section 4, we note that $\alpha = 0.5$ results in an optimally processed image, supporting the application of fractional partial differential equations in image processing.

6. Discussion

The results above indicate a strong convergence to a solution with increasing N . The similar results obtained for $N > 5$ are attributed to the comparison being drawn between two numerical methods which may be different from the true exact solution. Results obtained in [1] indicate that, when the present method is compared with an exact solution, the accuracy increases dramatically with increasing N . Despite this, the present method attains results similar to that of an industry standard, `NDSolve`, which is encouraging.

Figures 1, 2, 3, and 4 indicate the dynamic behaviour of the subdiffusive process. Interestingly the diffusive process becomes increasingly more aggressive as α decreases from 1 to 0; however, the final “steady state” achieved by these processes is typically less severe than the standard diffusion model. In image processing terms we could achieve a semidiffused result extremely quickly by choosing α to be small, rather than using $\alpha = 1$ and diffusing over a longer period.

In both the one- and two-dimensional cases, the Neumann boundary conditioned problems result in an accuracy of two orders of magnitude worse than that of the Dirichlet cases. In the linear case, where an exact solution exists, the accuracy of the present method increases dramatically with increasing N [2]; however, due to the comparison being drawn with another numerical method, `NDSolve`, the free ends affect the comparative solution, rather than reinforcing a condition on the boundary as in the Dirichlet case.

7. Conclusions

In addition to obtaining a high accuracy, the present method is robust for $0 < \alpha \leq 1$ and is in fact exact in the transformation from a time-fractional partial differential equation to partial differential equation or differential equation depending on the dimensionality. The errors incurred are then discretization error and numerical inversion of the Bromwich integral which is $\mathcal{O}(10.2^{-N})$ [17].

The application of this method to images introduces a computational hurdle since the resolution of the image dictates the spatial discretization resolution. After the transformation one must solve a linear system that is of the same dimension as the input image. This is exacerbated by the use of Chebyshev collocation, since the derivative matrices are full and not tridiagonal or banded as they are when using a standard finite-difference scheme, where the Thomas algorithm can be put to use as in [18–20]. In

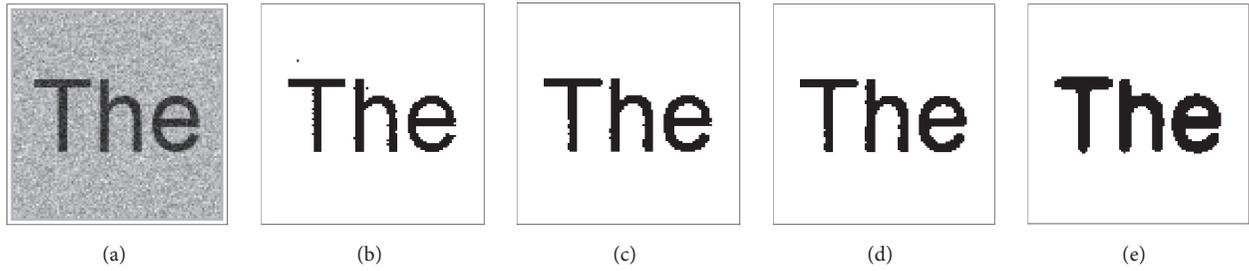


FIGURE 5: Figures depicting results obtained at increasing time using the current method and $\alpha = 1$. (a) The original image, (b) binarized original image at $t \approx 0$, (c) processed binary image at $t = 0.003$, (d) processed binary image at $t = 0.008$, (e) overdiffused image at $t = 0.01$.

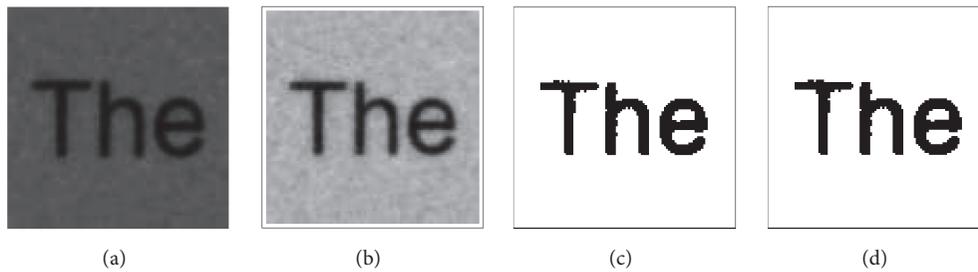


FIGURE 6: Figures depicting results obtained at increasing time using the current method and $\alpha = 0.2$. (a) The original image, (b) processed image at $t = 0.01$, (c) processed binary image at $t = 0.001$, (d) processed binary image at $t = 0.01$.

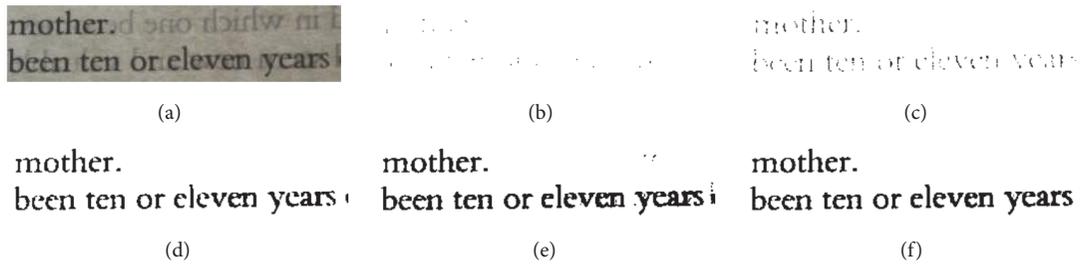


FIGURE 7: Figures illustrating processed binary images obtained for various values of α . (a) The original image, (b) $\alpha = 1$, (c) $\alpha = 0.75$, (d) $\alpha = 0.5$, (e) $\alpha = 0.25$, (f) ground truth image.

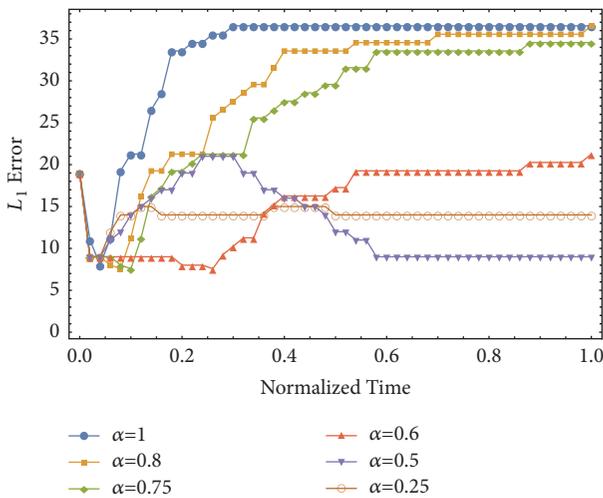


FIGURE 8: Plot L_1 errors against number of iterations for various values of α .

such a case, a numerical method should be implemented to improve the computational time required for large input data. The Chebyshev collocation method is also known to have large round-off errors for large problems, $N_x > 100$, due to finite-precision [22, 23]. Alternatively, for large systems a finite-difference discretization may be used to speed up computational time at the cost of accuracy in the solution. We have shown, for example, the images obtained by employing the hybrid-transform method to illustrate that the method does produce the expected results in a real-world application.

We have presented a method that is robust for a time-fractional diffusion equation with a nonlinear source term for one- and two-dimensional cases with both Dirichlet and Neumann boundary conditions. We have shown through numerical experiments that in the case of $\alpha = 1$ our solution obtains a result similar to that of `NDSolve` in `Mathematica 9` and extends trivially to fractional-order temporal derivative of order $\alpha \in (0, 1]$. The application to image processing

substantiates this method for real-world problems, highlighting the effectiveness of the fractional ordered derivative as a further dimension of control which has a dramatic effect both on the transition time of the model and on the final state attained. Within the realm of image processing this opens new avenues of research allowing more control over physically derived processes to construct methods that are physically substantial.

To the best of the authors' knowledge this hybridization of the Laplace transform, Chebyshev collocation, and quasi-linearization scheme has not yet been applied to FPDEs in one or two dimensions. This collaboration yields a method that is accurate and robust for time-fractional derivatives.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

Opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the CoE-MaSS. The work contained in this paper also forms part of B. A. Jacobs' Doctoral thesis.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

B. A. Jacobs acknowledges the support of the NRF under Thuthuka grant 94005. The DST-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS) is acknowledged for funding.

References

- [1] B. A. Jacobs and C. Harley, "A comparison of two hybrid methods for applying the time-fractional heat equation to a two dimensional function," in *Proceedings of the 11th International Conference of Numerical Analysis and Applied Mathematics 2013, ICNAAM 2013*, pp. 2119–2122, Greece, September 2013.
- [2] B. A. Jacobs and C. Harley, "Two Hybrid Methods for Solving Two-Dimensional Linear Time-Fractional Partial Differential Equations," *Abstract and Applied Analysis*, vol. 2014, Article ID 757204, 10 pages, 2014.
- [3] G.-h. Gao, Z.-z. Sun, and Y.-n. Zhang, "A finite difference scheme for fractional sub-diffusion equations on an unbounded domain using artificial boundary conditions," *Journal of Computational Physics*, vol. 231, no. 7, pp. 2865–2879, 2012.
- [4] I. Podlubny, *Fractional Differential Equations*, vol. 198 of *Mathematics in Science and Engineering*, Academic Press, San Diego, Calif, USA, 1999.
- [5] K. B. Oldham and J. Spanier, *The Fractional Calculus*, Academic Press, New York, NY, USA, 1974.
- [6] O. P. Agrawal, "Solution for a fractional diffusion-wave equation defined in a bounded domain," *Nonlinear Dynamics*, vol. 29, no. 1–4, pp. 145–155, 2002.
- [7] E. Hashemizadeh and A. Ebrahimzadeh, "An efficient numerical scheme to solve fractional diffusion-wave and fractional Klein–Gordon equations in fluid mechanics," *Physica A: Statistical Mechanics and its Applications*, vol. 503, pp. 1189–1203, 2018.
- [8] M. A. Zaky, D. Baleanu, J. F. Alzaidy, and E. Hashemizadeh, "Operational matrix approach for solving the variable-order nonlinear Galilei invariant advection-diffusion equation," *Advances in Difference Equations*, Paper No. 102, 11 pages, 2018.
- [9] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 182–193, 1992.
- [10] F.-C. Liu and Y. Ha, "Noise reduction based on reaction-diffusion system," in *Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, ICWAPR '07*, pp. 831–834, China, November 2007.
- [11] G.-H. Cottet and L. Germain, "Image processing through reaction combined with nonlinear diffusion," *Mathematics of Computation*, vol. 61, no. 204, pp. 659–673, 1993.
- [12] J. Kacur and K. Mikula, "Solution of nonlinear diffusion appearing in image smoothing and edge detection," *Applied Numerical Mathematics*, vol. 17, no. 1, pp. 47–59, 1995.
- [13] L. Alvarez, P.-L. Lions, and J.-M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion. II," *SIAM Journal on Numerical Analysis*, vol. 29, no. 3, pp. 845–866, 1992.
- [14] S. Morfu, "On some applications of diffusion processes for image processing," *Physics Letters A*, vol. 373, no. 29, pp. 2438–2444, 2009.
- [15] B. A. Jacobs and E. Momoniat, "A novel approach to text binarization via a diffusion-based model," *Applied Mathematics and Computation*, vol. 225, pp. 446–460, 2013.
- [16] B. A. Jacobs and E. Momoniat, "A locally adaptive, diffusion based text binarization technique," *Applied Mathematics and Computation*, vol. 269, pp. 464–472, 2015.
- [17] J. A. Weideman and L. N. Trefethen, "Parabolic and hyperbolic contours for computing the Bromwich integral," *Mathematics of Computation*, vol. 76, no. 259, pp. 1341–1356, 2007.
- [18] P. J. Singh, S. Roy, and I. Pop, "Unsteady mixed convection from a rotating vertical slender cylinder in an axial flow," *International Journal of Heat and Mass Transfer*, vol. 51, no. 5–6, pp. 1423–1430, 2008.
- [19] P. M. Patil, S. Roy, and I. Pop, "Unsteady mixed convection flow over a vertical stretching sheet in a parallel free stream with variable wall temperature," *International Journal of Heat and Mass Transfer*, vol. 53, no. 21–22, pp. 4741–4748, 2010.
- [20] P. M. Patil and S. Roy, "Unsteady mixed convection flow from a moving vertical plate in a parallel free stream: Influence of heat generation or absorption," *International Journal of Heat and Mass Transfer*, vol. 53, no. 21–22, pp. 4749–4756, 2010.
- [21] A. Bayliss, A. Class, and B. J. Matkowsky, "Roundoff Error in Computing Derivatives Using the Chebyshev Differentiation Matrix," *Journal of Computational Physics*, vol. 116, no. 2, pp. 380–383, 1995.
- [22] W. S. Don and A. Solomonoff, "Accuracy and speed in computing the Chebyshev collocation derivative," *SIAM Journal on Scientific Computing*, vol. 16, no. 6, pp. 1253–1268, 1995.
- [23] K. S. Breuer and R. M. Everson, "On the errors incurred calculating derivatives using Chebyshev polynomials," *Journal of Computational Physics*, vol. 99, no. 1, pp. 56–67, 1992.



Hindawi

Submit your manuscripts at
www.hindawi.com

