Numerical simulations of two-dimensional fractional subdiffusion problems

Hermann Brunner[1,*], Leevan Ling[2,*], Masahiro Yamamoto[3,**]

Abstract

The growing number of applications of fractional derivatives in various fields of science and engineering indicates that there is a significant demand for better mathematical algorithms for models with real objects and processes. Currently, most algorithms are designed for 1D problems due to the memory effect in fractional derivatives. In this work, the 2D fractional subdiffusion problems are solved by an algorithm that couples an adaptive time stepping and adaptive spatial basis selection approach. The proposed algorithm is also used to simulate a subdiffusion-convection equation.

Keywords: Fractional differential equations, Kansa's method, radial basis functions, collocation, adaptive greedy algorithm, geometric time grids

1. Introduction

Let Ω be a bounded domain in $\mathbb{R}^2$ with sufficiently smooth boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\Gamma_D \cap \Gamma_N = \emptyset$. We consider an initial-boundary problem for a time fractional diffusion equation with fractional-order $0 < \alpha < 1$:

*Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

**Graduate School of Mathematical Sciences, University of Tokyo, 3-8-1 Komaba Meguro Tokyo.

[1]E-mail: hbrunner@math.hkbu.edu.hk

[2]E-mail: lling@hkbu.edu.hk (Corresponding Author)

[3]E-mail: myama@ms.u-tokyo.ac.jp

$$
\begin{array}{rcll}
{}^cD_t^\alpha u(x,t) &=& \Delta u(x,t) + f(x), & x \in \Omega, \quad t \in (0,T), \\
u(x,0) &=& u_0(x), & x \in \Omega, \\
u(x,t) &=& g_0(x,t), & x \in \Gamma_D, \quad t \in (0,T), \\
\partial_\nu u(x,t) &=& g_1(x,t), & x \in \Gamma_N \quad t \in (0,T),
\end{array}
\tag{1}
$$

where ${}^cD_t^\alpha$ denotes the Caputo fractional derivative of order α with respect to t defined by

$$
{}^cD_t^\alpha u(x,t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial u(x,\eta)}{\partial \eta} \frac{d\eta}{(t-\eta)^\alpha}, \quad 0 < \alpha < 1, \tag{2}
$$

see monograph by Podlubny [1]. The operator Δ is the Laplacian in $\mathbb{R}^2$ and ∂_ν is the outward normal derivative. Note that if $\alpha = 1$, then the Caputo fractional derivative in (2) becomes $\partial_t u(x,t)$ and the problem in (1) represents the standard integer-order parabolic equation. Very recently, existence and uniqueness of the weak solution of (1) is shown in [2].

The fractional diffusion equation is related with the continuous-time random walk and is a model for anomalous diffusion in many applied fields such as diffusion processes of contaminants in porous media, see [3, 4, 5, 6] and the references therein.

In this paper, we discuss a numerical algorithm that couples an adaptive time stepping and an adaptive spatial basis selection approach, and show numerical results.

As for works on numerical methods for fractional diffusion equations appearing in (1.1), we can refer to [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. The above papers, except for [18], all treat the case where the spatial dimension is one; see also [14] for a nonlinear fractional diffusion equation. As for available numerical methods for fractional diffusion equations, see [19] and [20].

2. Methodology

2.1. Finite difference time discretization

Suppose the numerical approximation of the solution u in (1) is obtained up to some time t ($0 < t < T$). Any explicit-time-scheme requires evaluation of the Caputo fractional derivative (2). If the numerical approximation is restricted on uniform time grid, algorithms for evaluation of the convolution integrals [21] can be employed. As we will soon see, nonuniform time grids

are preferred. Hence, we employ the difference approximation in [22] for the fractional time derivative. Suppose the time interval $[0, T]$ is discretized uniformly into n subintervals; define $t_k = k \cdot \triangle t$, $k = 0, 1, \ldots, K$, where $\triangle t = T/n$ is the time step. Let $\lambda(t_k)$ be the exact value of a function $\lambda(t)$ at time step t_k. Then, the fractional time derivative can be approximated by the following scheme:

$$\begin{aligned}
{}^cD_t^\alpha \lambda(t_{k+1}) &= \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial \lambda(t)}{\partial \eta} \frac{d\eta}{(t-\eta)^\alpha} \\
&\approx \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{k} \frac{\lambda(t_{j+1}) - \lambda(t_j)}{\triangle t} \int_{j\triangle t}^{(j+1)\triangle t} \frac{d\eta}{(t_{k+1}-\eta)^\alpha} \\
&= \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{k} \frac{\lambda(t_{j+1}) - \lambda(t_j)}{\triangle t} \int_{(k-j+1)\triangle t}^{(k-j)\triangle t} \eta^{-\alpha} d\eta \\
&= \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{k} \frac{\lambda(t_{k+1-j}) - \lambda(t_{k-j})}{\triangle t} \int_{j\triangle t}^{(j+1)\triangle t} \eta^{-\alpha} d\eta \\
&= \frac{(\triangle t)^{1-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{k} \frac{\lambda(t_{k+1-j}) - \lambda(t_{k-j})}{\triangle t} \big[(j+1)^{1-\alpha} - j^{1-\alpha}\big].
\end{aligned}$$

Hence, we obtain a first-order discretization

$$ {}^cD_t^\alpha \lambda(t_{k+1}) \approx {}^c\triangle_t^\alpha \lambda(t_{k+1}) := \frac{(\triangle t)^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{k} w_j \big[\lambda(t_{k+1-j}) - \lambda(t_{k-j})\big], \quad (3)$$

for $k = 0, \ldots, K-1$ where the weight is defined as $w_j = [(j+1)^{1-\alpha} - j^{1-\alpha}]$ for $j = 0, 1, \ldots, K$. Equation (3) can be easily rewritten to obtain a fully explicit scheme for the latest approximation $\lambda(t_{k+1})$ which depends on all previous values $\lambda(t_0), \ldots, \lambda(t_k)$.

2.2. Kernel based spatial approximation

In this section, we consider a kernel-basis representation for the spatial variables. For the considered problem (1), the numerical approximation is expanded as

$$u(x,t) \approx U(x,t) = \sum_{\ell=1}^{N} \lambda_\ell(t) \Phi\big(\|x - \xi_\ell\|/c\big), \quad x \in \Omega, \quad (4)$$

where c is the scaling parameter, the set $\Xi = \{\xi_\ell\}_{\ell=1}^N$ is the trial centers and $\Phi(\cdot)$ can be any commonly used radial basis kernel; for examples, multiquadrics $\Phi(r) = (r+1)^{1/2}$, inverse multiquadrics $\Phi(r) = (r+1)^{-1/2}$, gaussian $\Phi(r) = \exp(-r^2)$, thin plate spline $\Phi(r) = r^2 \log(r)$, etc. Putting (4) into the subdiffusion equation (1) results in

$$\sum_{\ell=1}^{N} {}^cD_t^\alpha \lambda_\ell(t) \Phi\big(\|x - \xi_\ell\|/c\big) = \sum_{\ell=1}^{N} \lambda_\ell(t) \Delta\Phi\big(\|x - \xi_\ell\|/c\big) + f(x). \qquad (5)$$

Using a sufficiently dense set $X = \{x_1, x_2, \ldots, x_M\} \subset \bar{\Omega}$ for collocation, applying the finite difference ${}^c\triangle_t^\alpha$ in (3) to $\lambda_\ell(\cdot)$ and the strong form collocations at X will result in a matrix system for updating (discrete) values of the coefficient functions $\lambda_\ell(t_k)$, $\ell = 1, \ldots, N$, $k = 1, \ldots, K$.

2.3. Geometric time grids

When $t \gg 0$, the size of "memory" in the fractional-derivative approximation becomes enormously large. The "short-memory" principle [1, 23] suggests that, for large t, the role of the "history" of the behavior of the solution $u(x,t)$ near $t = 0$ can be neglected. This agrees with the fact that $w_j \searrow 0$ in (3) as $j \nearrow$ with large n. Hence, one may take into account the behavior of $u(x,t)$ in the recent past in the interval $[t - L,\ t]$ where L is the "memory length". It is shown [1, Ch.7] that

$$|\ {}^cD_t^\alpha \lambda(t) - {}_{t-L}{}^cD_t^\alpha \lambda(t)| \le \epsilon, \quad \text{if } L \ge \left(\frac{M}{\epsilon|\Gamma(1-\alpha)|}\right)^{1/\alpha},$$

where ${}_{t-L}{}^cD_t^\alpha$ is the fractional derivative with moving lower integration limit $t - L$ in the definition (2), instead of 0.

Despite of its success in 1D problems, Figure 1 shows that the short-memory principle is not particularly useful in reducing the memory requirement in 2D when $t \approx 1$. The penalty in the form of inaccuracy is too large; for example, when $\alpha = 0.5$, any memory length $L < 1$ will introduce an error $\epsilon \gg 1$. As we will see soon in the numerical experiment, small time stepping is important to capture the fast "initial drop" [1, Ch.3] accurately.

One can also see the initial drop as a boundary layer at $t = 0$. Techniques for solving boundary layer problems, e.g. [24, 25, 26], can be applied. However, these techniques do not ease the memory requirement. The nested mesh principle in [23] partitions the time interval $[0, T]$ into nonuniform subintervals; the smallest subinterval is placed at $t = 0$ in order to well-capture the

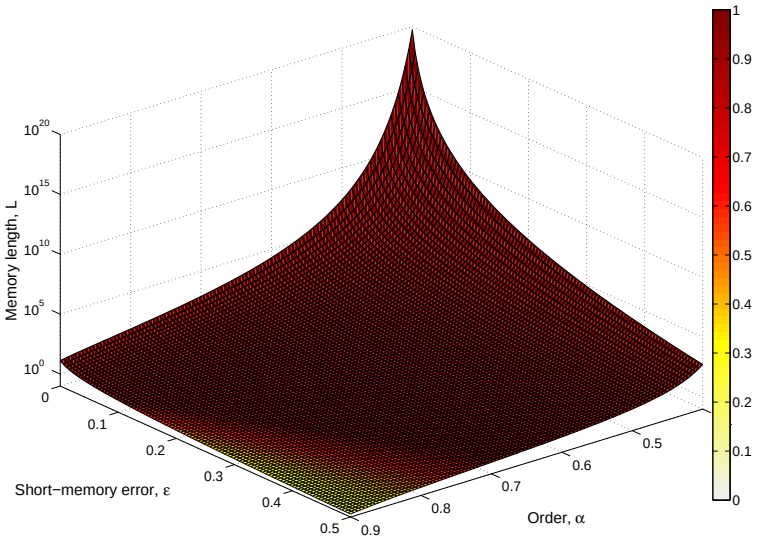

Figure 1: Memory length L required for various orders α and desired accuracy ϵ in the "short-memory" principle.

initial drop and the rest gradually widen as time increases in order to speed up the calculation. In this paper, we do not neglect outdated information completely as in the short-memory principle. Similar to [27], we make use of the fading memory property but in a different way. To do so, we turn our focus to the *geometric time grids* [28, 29].

For large t away from 0, the solutions of subdiffusion "diffuse more slowly" than the standard integer-order diffusion process. It makes sense to employ a large time step in this region. Let $U(\cdot, t_k)$ be the numerical approximation for $u(\cdot, t_k)$. To monitor this diffusion rate, we define a measure between the numerical solutions $U(x, \cdot)$ of two consecutive time steps by

$$\Delta U_{t_k} = \frac{\|U(x,t_k) - U(x,t_{k-1})\|_{L^2(\Omega)}}{\|U(x,t_{k-1})\|_{L^2(\Omega)}}, \quad \text{for} \quad k = 1, \ldots, K. \tag{6}$$

For some user-defined relaxation parameters τ, if $\Delta U_{t_i} < \tau$, the time spacing is relaxed:

$$\triangle t \leftarrow 2 \cdot \triangle t,$$

up to some prefixed value $\triangle t_{\max}$.

2.4. Adaptive kernel selections

Although the geometric time grids can help reduce the number of previous solutions needed for evaluating ${}^{c}\triangle_{t}^{\alpha}u(x,t)$ at current time, it is not possible to completely remove the memory nature as it comes directly from the fractional subdiffusion problem. To effectively minimize the overhead of computer memory, the spatial information must be carefully treated. Using kernel representation, a kind of meshless method, expansion (4) provides us a parametric description of the numerical approximation. This is the first motivation of employing an adaptive technique so that only a small subset of unknown coefficients $\lambda_\ell(t_k)$ in (4) are nonzero and stored instead of all approximation function values $U(X,t_k)$.

However, kernel representation is not at all trouble-free. For example, choosing optimal trial centers ξ_ℓ for numerical expansion is a common problem for researchers who employ various meshless methods. On one hand, high accuracy is always desired; on the other, ill-conditioning problems of the resultant matrices, that may lead to unstable algorithms, prevent some researchers from using meshless methods. For example, the optimal placements of source points in the method of fundamental solutions, or of the centers in the radial basis functions method are always unclear. Intuitively, such optimal locations will depend on many factors: the partial differential equations, the domain, the trial basis used (i.e. the employed method itself), the computational precisions, some user-defined parameters, and so on. Such complexity makes the hope of having optimal trial centers placement unpromising.

Various adaptive algorithms are devoted on the sub-optimal solution to the trial-centers-placement problem. In particular, we employ the most up-to-date algorithm in [30]. The first theoretical foundation [31] is that, in a large set of trial centers (large in the sense of $|\Xi| \geq |X|$), there exists a subset such that the meshless collocation system is solvable. To put this purely theoretical result into practice, we use more-than-necessary number of basis functions in the expansion (4). Then, by applying the adaptive algorithm with certain selection criteria, a "proper" subset of trial centers will be selected from Ξ. Equivalently, only a small subset of coefficients $\lambda_\ell(t_k)$ is *active* in (4). Before describing the algorithm, we emphasize the algorithm presented below is matrix-free in the sense that resultant matrix will not be fully evaluated or stored. Hence, the increase in number of basis functions does not impose an overhead to the memory requirement; that is one of the main concerns in solving fractional subdiffusion equations.

Without going into the how-and-why, we present the key ideas and steps of the adaptive algorithm. Readers can refer to the original articles [30, 31, 32] for details. Consider a meshless collocation system $A\lambda = b$ with $A \in \mathbb{R}^{M\times N}$ and $b \in \mathbb{R}^M$ usually with $M \leq N$ (due to the solvability theorem we mentioned above). The adaptive algorithm makes sequential *collocation–trial center* pair selection and builds up ordered indexed sets, denoted by $X_{(k)} = \{x_{(1)}, \ldots, x_{(k)}\}$ and $\Xi_{(k)} = \{\xi_{(1)}, \ldots, \xi_{(k)}\}$, for $k = 1, \ldots, M$, for collocation points and trial centers respectively.

Suppose, after the first k iterations, our algorithm selects a set of k collocation points and a set k RBF centers, respectively, and builds $X_{(k)} \subset X$ and $\Xi_{(k)} \subset \Xi$. These sets of points define a subproblem to the original one:

$$\begin{cases} A_{(k)}\check{\lambda}^{(k)} = \check{b}^{(k)}, \\ A_{(k)}^T\check{\nu}^{(k)} = -\check{\lambda}^{(k)}, \end{cases} \tag{7}$$

where $A_{(k)} \in \mathbb{R}^{k\times k}$ is a $k \times k$ square-submatrix of the full matrix A with rows associated with $X_{(k)}$ and columns associated with $\Xi_{(k)}$. Similarly, $\check{b}^{(k)} \in \mathbb{R}^k$ is the k entry of b associated with the selected collocation points $X_{(k)}$. After solving (7) for $\check{\lambda}^{(k)} \in \mathbb{R}^k$, let $\lambda^{(k)} \in \mathbb{R}^N$ be the extension of $\check{\lambda}^{(k)}$ by patching zeros into entries associated with the unselected trial centers. Similarly, $\check{\nu}^{(k)} \in \mathbb{R}^k$ can be extended to $\nu^{(k)} \in \mathbb{R}^N$.

All versions of the adaptive algorithm use the same criteria for selecting new collocation points. The $(k+1)^{\text{st}}$ collocation point $x_{(k+1)}$ can be selected from the primal residual

$$r^{(k)} = A\lambda^{(k)} - b. \tag{8}$$

In other words, we are checking the well-fitness of approximating b with only k columns of A with weight in $\check{\lambda}^{(k)}$. We pick—from the set of collocation points X_M—an (always-new) collocation point $x_{(k+1)}$ such that the corresponding entry in the primal residual $r^{(k)}$ is the largest in absolute value (that is the *greedy technique*). This is why the adaptive algorithm here is sometimes referred as the *greedy algorithm*.

For an unsymmetric matrix system, picking a row (or collocation point) provides no hint on column-selection. In [31, 32], columns (or trial centers) are selected based on the determinant function. A new column is selected such that of the resulting submatrix $A^{(k+1)}$ has a determinant closest to 1. The latest version in [30] is more cost-efficient and it uses the dual residual

$$q^{(k)} = \lambda^{(k)} + A^T\nu^{(k)}. \tag{9}$$

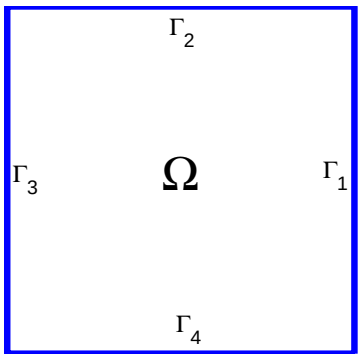

Figure 2: Boundary labels for $\Omega = [-1, 1]^2$.

Geometric interpretation of the dual residual can be found in the same article. Using the greedy technique again, the new trial center $\xi_{(k+1)}$ is selected from all candidates in Ξ_N such that $q^{(k)}$ is the largest in absolute value among all others. The adaptive algorithm terminates if either residual is smaller than some tolerances or when severe problem of ill-conditioning appears in the subproblem (7). After the adaptive algorithm terminates, the convergence analysis in [31] recommends to use all available collocation points instead of the selected ones only. To obtain the unknown coefficient λ in $A\lambda = b$, an overdetermined system containing all rows but only the selected columns are solved. In [33], if one employs the MQ-kernel, exponential spatial accuracy for the integer-order heat problem is formally proven. Numerical evidences of exponential convergence for other types of PDE can be found in [34].

3. Numerical verifications

We verify the proposed numerical scheme to solve a simplified problem with zero Dirichlet/Neumann/mixed boundary conditions:

$$\begin{array}{rcllll} {}^{c}D_t^{\alpha} u(x,t) & = & \Delta u(x,t), & x \in \Omega, & t \in (0,T), & 0 < \alpha < 1, \\ u(x,0) & = & u_0(x), & x \in \Omega, & & \\ u(x,t) & = & 0, & x \in \Gamma_D, & t \in (0,T), & \\ \partial_\nu u(x,t) & = & 0, & x \in \Gamma_N & t \in (0,T). & \end{array} \tag{10}$$

Let $\Omega = [-1, 1]^2$ whose boundaries are labeled as in Figure 2 and $x = (x_1, x_2) \in \Omega$. We consider three cases:

Dirichlet BC is imposed on the whole boundary $\Gamma_D = \partial\Omega$. The exact solution of (10) associated with initial condition

$$u_0(x) = \cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right) \tag{11}$$

is given as

$$w_1(x,t) = E_\alpha\left(-\frac{1}{2}\pi^2 t^\alpha\right)\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right).$$

Neumann BC: is imposed on the whole boundary $\Gamma_N = \partial\Omega$ with initial condition

$$u_0(x) = \sin\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right) \tag{12}$$

The exact solution is $E_\alpha\left(-\frac{1}{2}\pi^2 t^\alpha\right) u_0(x)$.

Mixed BC: with Dirichlet BC on $\Gamma_D = \Gamma_1 \cup \Gamma_3$ and Neumann BC on $\Gamma_N = \Gamma_2 \cup \Gamma_4$. Initial condition is

$$u_0(x) = \cos\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right).$$

Similarly, the exact solution is $E_\alpha\left(-\frac{1}{2}\pi^2 t^\alpha\right) u_0(x)$.

3.1. $\alpha = \frac{1}{2}$

Our first verification focuses on the half-order cases because exact solutions to (1) can be found explicitly. Recall the definition of the (one-parameter) Mittag-Leffler function

$$E_\alpha(z) := \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \quad \alpha > 0,$$

and its property

$${}^cD_t^\alpha E_\alpha(-\lambda t^\alpha) = -\lambda E_\alpha(-\lambda t^\alpha),\ 0 < \alpha < 1. \tag{13}$$

As for the unique existence of solution to (10) in the case of $\Gamma_D = \partial\Omega$, we refer to the very recent results in [35] and [36]. Note that $E_{1/2}(-z) = \mathrm{erfcx}(z)$ is the *scaled complementary error function* defined by

$$\mathrm{erfcx}(z) = \frac{2}{\sqrt{\pi}}\exp(z^2)\int_z^\infty \exp(-\eta^2)\,d\eta,$$

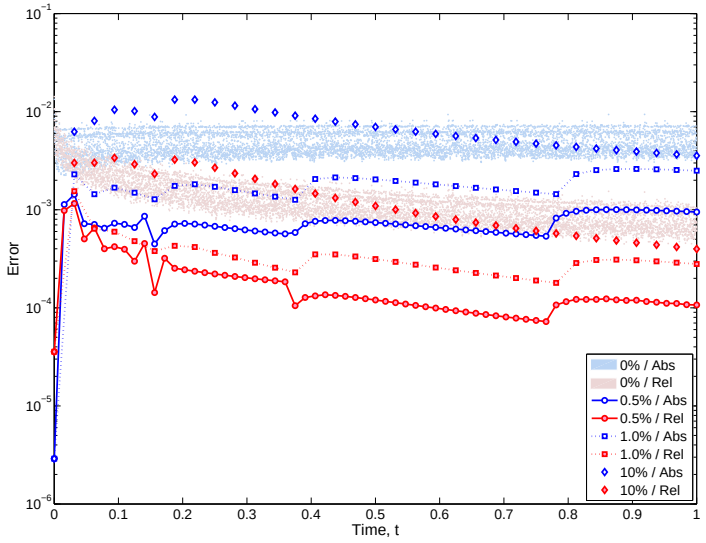

Figure 3: Dirichlet boundary conditions ($\alpha = \frac{1}{2}$): Absolute and relative errors over time for different relaxation parameters $\tau = 0\%$, 0.5%, 1.0%, and 10%.

see [1] for instance.

A total number of 1537 trial basis functions, including both interior and boundary nodes, is fed into the adaptive algorithm for all time. For all three boundary conditions and all time updates, the numbers of selected basis range between 82 to 146 that is an over 90% saving in memory requirement. The initial time step is $dt = 2^{-13}$ and it is relaxed whenever the measure in (6) is less than $\tau = 0\%$, 0.5%, 1.0%, and 10%. When $\tau = 0\%$, the time stepping is fixed at $dt = 2^{-13}$ for all time. Figure 3 to Figure 4 show the absolute and relative errors over $t = (0, 1]$.

One interesting observation (see Figure 3 and Figure 4) is that fine time stepping ($\tau = 0\%$) does not result in the best accuracy due to the presence of cancelation errors. When τ is large, e.g. 10%, the time spacing is relaxed too early and hence thus results in poor accuracy near $t = 0$. However, as t increases, we see that the numerical solutions for $\tau = 10\%$ is more accurate than those for $\tau = 0\%$. This tells how severe the cancelation errors are. Better results can be obtained by small tolerances $\tau = 0.5\%$ or 1.0%. Note that using small $\tau > 0$ requires more (but still much faster than using fixed small time step) computational time.

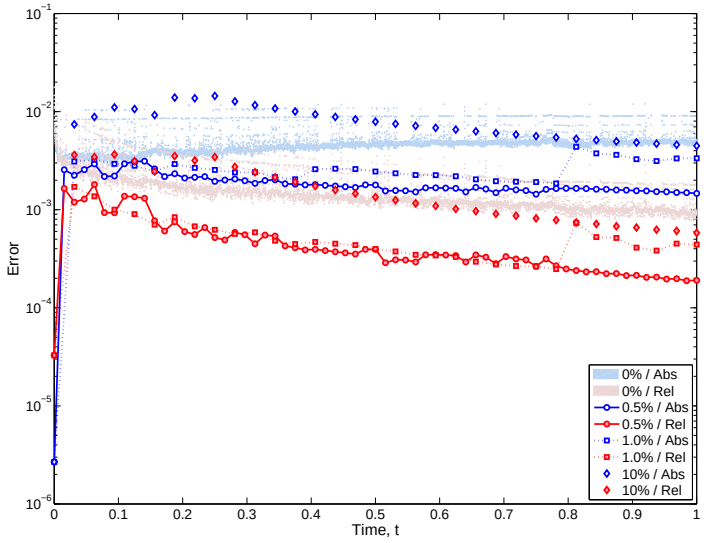

Figure 4: Neumann boundary conditions ($\alpha = \frac{1}{2}$): Absolute and relative errors over time for different relaxation parameters $\tau = 0\%$, 0.5%, 1.0%, and 10%.

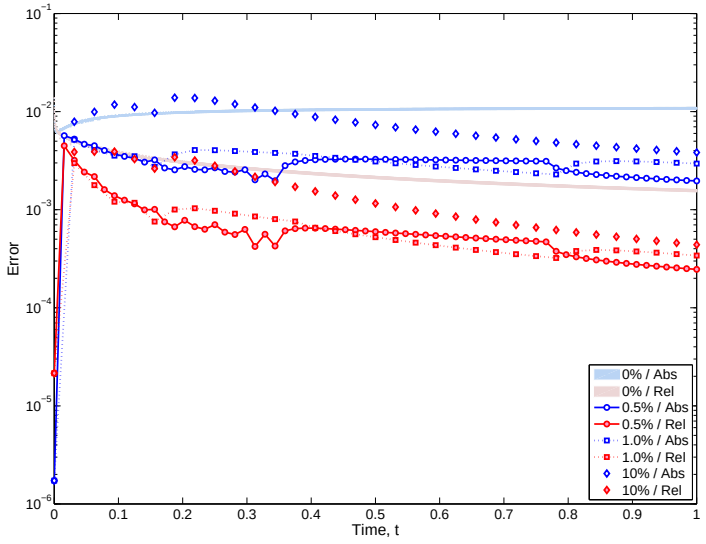

Figure 5: Mixed boundary conditions ($\alpha = \frac{1}{2}$): Absolute and relative errors over time for different relaxation parameters $\tau = 0\%$, 0.5%, 1.0%, and 10%.

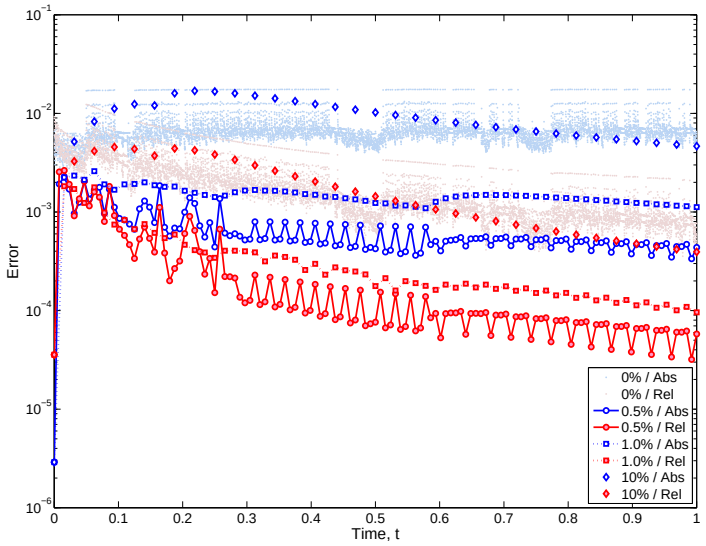

Figure 6: Dirichlet boundary conditions ($\alpha = \frac{2}{3}$): Absolute and relative errors over time for different relaxation parameters $\tau = 0\%$, 0.5%, 1.0%, and 10%.

3.2. Other α

To ensure that the proposed algorithm works beyond the special case $\alpha = 1/2$, we consider (10) again but with different order. To make use of the eigen-relation (13) with $\alpha \neq 1/2$, we numerically evaluate the Mittag-Leffler functions [37] to high accuracy (with tolerance 10^{-10}). Other settings remain the same.

For Dirichlet BC with initial condition (11) and $\alpha = 2/3$, the results are shown in Figure 6. The error profiles are oscillatory comparing to Figure 3. We still observe that using $\tau = 0.5\%$ yields the best result. Without the geometric grid (i.e. $\tau = 0\%$), the results are not only inaccurate but also very computationally costly (due to the enormous number of time steps).

For completeness, we now consider a problem with Neumann BC. We take $\alpha = 1/\sqrt{5}$ so that it is irrational and less than a-half. Initial condition is taken to be (12). The results displayed in Figure 7 should be compared with Figure 4. From these figures, we can see that the value α does not have a great effect on the proposed algorithm.

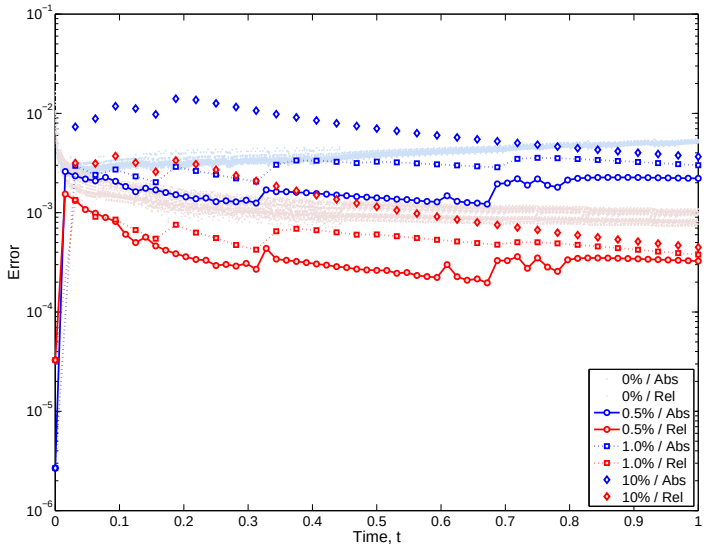

Figure 7: Neumann boundary conditions ($\alpha = \frac{1}{\sqrt{5}}$): Absolute and relative errors over time for different relaxation parameters $\tau = 0\%$, 0.5%, 1.0%, and 10%.

4. Numerical simulations

Our first simulation studies the effect of the order α on the decay rate of the subdiffusion solution. We consider (10) with Neumann boundary condition for insulated boundary. Initial time stepping is $dt = 2^{-13}$ and the relaxation parameter is set to be $\tau = 0.05\%$. Figure 8 shows the maximum norm of the numerical solution for $\alpha = 0.1, 0.2, \ldots, 0.9$ and time $t \in [0, 1]$; the dots in Figure 8 indicate all visited times in each run. For small α, say 0.1, the initial drop is enormous; in case of $\alpha = 0.1$, the maximum norm of the solution immediately drops from 1 to 0.34 after the first time update. On the other hand, when time gets large, the change in the solution is relatively minor; dt is relaxed all the way to $dt_{\max} = 2^{-5}$. For large α, the solution behaves more like the integer-order case. When $\alpha = 0.9$, the largest time stepping used is $dt = 2^{-8}$. In the experiment, we see that a very small initial time stepping is desired for small α. Whereas, when α is large, a more easygoing relaxation scheme is desired.

Our last example simulates the fractional subdiffusion-convection prob-

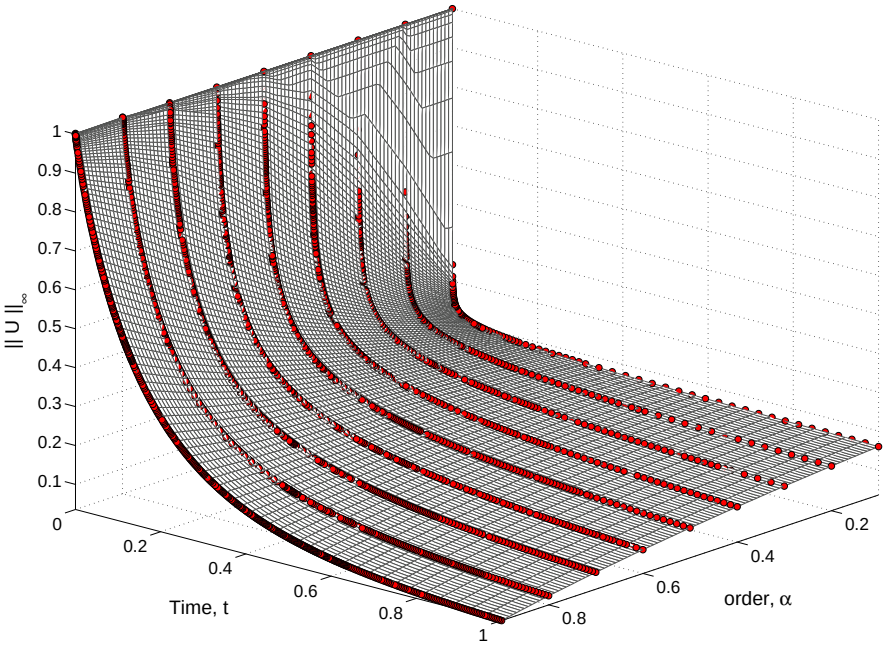

Figure 8: Maximum norm of numerical solution to the subdiffusion problem (10) with Neumann boundary conditions for $\alpha = 0.1, 0.2, \ldots, 0.9$.

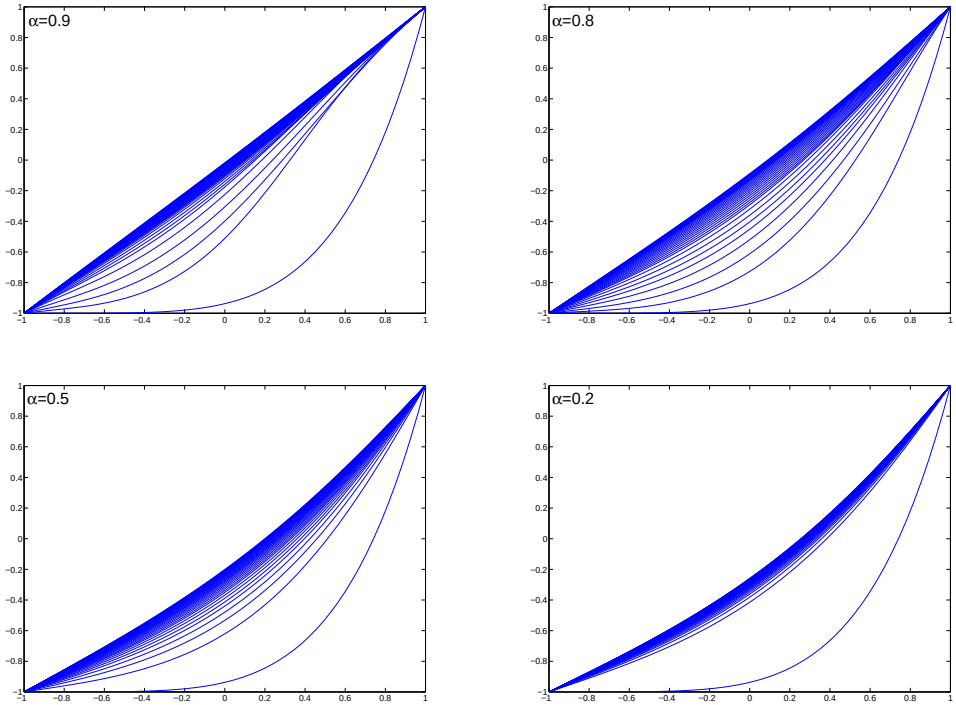

Figure 9: Numerical solution (cross section) for a fractional subdiffusion-convection problem with different α.

lem,

$$\begin{aligned} {}^{c}D_t^{\alpha} u(x,t) &= \Delta u(x,t) + \omega \frac{\partial}{\partial x} u(x,t), && x \in \Omega, \\ u(x,0) &= 2\left(\frac{x_1+1}{2}\right)^5 - 1, && x = (x_1, x_2) \in \Omega, \\ u(x,t) &= x_1, && x \in \Gamma_1 \cup \Gamma_3, \\ \partial_\nu u(x,t) &= 0, && x \in \Gamma_2 \cup \Gamma_4 \end{aligned}$$

for $\alpha = \{0.9, 0.8, 0.5, 0.2\}$, $t \in (0,1)$ and $\omega = 0.005$ is the convection coefficient. Due to the symmetry of the problem, we show the cross section of the numerical solution (parallel to the x_1-axis) in Figure 9 for every $1/32$ sec moving up from the lower-right towards the diagonal.

The effect of convection can be seen most clearly in the case of $\alpha = 0.9$; the presence of the points of inflection is obvious for small t. The effect

of convection is less clear as α decreases. After careful examination, one may still find some inflection points for the case of $\alpha = 0.8$. However, when $\alpha = 0.5$, the effect of convection becomes even less significant. For $\alpha = 0.2$, as in the previous example, we see a very rapid change in the solution between $(0, \epsilon)$; after that, the solution varies slowly. Note also that the numerical solutions for different α are more distinct near the left endpoint where fluid is being pumped out. For experiment design, it makes sense to place sensors somewhere in $-1 < x_1 < 0$ instead of in $0 < x_1 < 1$.

5. Conclusion

We present a numerical scheme, which includes geometric time grids relaxation and adaptive kernel selection, for solving the 2D fraction subdiffusion problems. The algorithm is tested with different boundary conditions, for which exact solutions are known, in order to verify its accuracy. Next, the algorithm is applied to simulate the subdiffusion problems with different fractional-orders and a subdiffusion-convection problem.

Since the kernel presentation is used for spatial discretization, we implicitly required the initial condition to be of certainly smoothness. In cases where this is not true, one should employ a finite element or finite difference scheme for the first few time steps. Once the numerical solution (sub)diffuses and becomes smooth, the kernel presentation can be re-introduced. The memory saving provided by the adaptive kernel selection will become more significant in 3D. The simulations in [33] suggest that the adaptive algorithm takes roughly about 500 trial basis for approximating smooth functions in $[-1, 1]^3$. This suggests that the proposed algorithm has a good chance in solving 3D subdiffusion problems without modification. We leave this to our future studies.

Acknowledgements

This project was supported by the CERG Grant of Hong Kong Research Grant Council and the FRG grant of Hong Kong Baptist University. This work has been mostly done during the stay of the second named author at the Graduate School of Mathematical Sciences of the University of Tokyo in December 2008–January 2009, and the stay was supported by Global COE Program "The Research and Training Center for New Development in Mathematics."

[1] I. Podlubny, Fractional differential equations, Vol. 198 of Mathematics in Science and Engineering, Academic Press Inc., San Diego, CA, 1999.

[2] X. Li, C. Xu, Existence and uniqueness of the weak solution of the space-time fractional diffusion equation and a spectral approximation, Commun. Comput. Phys. To appear. Doi: 10.4208/cicp.020709.221209a.

[3] R. Gorenflo, F. Mainardi, D. Moretti, P. Paradisi, Time fractional diffusion: a discrete random walk approach, Nonlinear Dynam. 29 (1-4) (2002) 129–143.

[4] R. Gorenflo, F. Mainardi, A. Vivoli, Continuous-time random walk and parametric subordination in fractional diffusion, Chaos Solitons Fractals 34 (1) (2007) 87–103.

[5] R. Metzler, J. Klafter, Boundary value problems for fractional diffusion equations, Phys. A 278 (1-2) (2000) 107–125.

[6] R. Metzler, J. Klafter, The random walk's guide to anomalous diffusion: a fractional dynamics approach, Phys. Rep. 339 (1) (2000) 77.

[7] K. Al-Khaled, S. Momani, An approximate solution for a fractional diffusion-wave equation using the decomposition method, Appl. Math. Comput. 165 (2) (2005) 473–483.

[8] E. Cuesta, C. Lubich, C. Palencia, Convolution quadrature time discretization of fractional diffusion-wave equations, Math. Comp. 75 (254) (2006) 673–696.

[9] E. Cuesta, C. Palencia, A numerical method for an integro-differential equation with memory in Banach spaces: qualitative properties, SIAM J. Numer. Anal. 41 (4) (2003) 1232–1241.

[10] W. Deng, Numerical algorithm for the time fractional Fokker-Planck equation, J. Comput. Phys. 227 (2) (2007) 1510–1522.

[11] T. A. M. Langlands, B. I. Henry, The accuracy and stability of an implicit solution method for the fractional diffusion equation, J. Comput. Phys. 205 (2) (2005) 719–736.

[12] Z. Odibat, S. Momani, Approximate solutions for boundary value problems of time-fractional wave equation, Appl. Math. Comput. 181 (1) (2006) 767–774.

[13] Z. Odibat, S. Momani, Numerical solution of Fokker-Planck equation with space- and time-fractional derivatives, Physics Letters A 369 (2007) 349–358.

[14] Z. Odibat, S. Momani, Numerical methods for nonlinear partial differential equations of fractional order., Appl. Math. Modelling 32 (1) (2008) 28–39.

[15] S. Shen, F. Liu, V. Anh, I. Turner, Detailed analysis of a conservative difference approximation for the time fractional diffusion equation, J. Appl. Math. Comput. 22 (3) (2006) 1–19.

[16] Z. Sun, X. Wu, A fully discrete difference scheme for a diffusion-wave system, Appl. Numer. Math. 56 (2) (2006) 193–209.

[17] S. B. Yuste, Weighted average finite difference methods for fractional diffusion equations, J. Comput. Phys. 216 (1) (2006) 264–274.

[18] P. Zhuang, F. Liu, Implicit difference approximation for the two-dimensional space-time fractional diffusion equation, J. Appl. Math. Comput. 25 (1-2) (2007) 269–282.

[19] K. Diethelm, N. J. Ford, A. D. Freed, Y. Luchko, Algorithms for the fractional calculus: a selection of numerical methods, Comput. Methods Appl. Mech. Engrg. 194 (6-8) (2005) 743–773.

[20] P. P. Valkó, J. Abate, Numerical inversion of 2-D Laplace transforms applied to fractional diffusion equations, Appl. Numer. Math. 53 (1) (2005) 73–88.

[21] K. Diethelm, A. D. Freed, An efficient algorithm for the evaluation of convolution integrals, Comput. Math. Appl. 51 (1) (2006) 51 – 72.

[22] F. Liu, P. Zhuang, V. Anh, I. Turner, A fractional-order implicit difference approximation for the space-time fractional diffusion equation, ANZIAM J. 47 (2005) C48–C68.

[23] N. J. Ford, A. C. Simpson, The numerical solution of fractional differential equations: Speed versus accuracy, Numer. Algorithms 26 (4) (2001) 333–346.

[24] T. Tang, M. R. Trummer, Boundary layer resolving pseudospectral methods for singular perturbation problems, SIAM J. Sci. Comput. 17 (2) (1996) 430–438.

[25] L. Ling, M. R. Trummer, Multiquadratic collocation method with integral formulation for boundary layer problems, Comput. Math. Appl. 48 (5-6) (2004) 927–941.

[26] L. Ling, M. R. Trummer, Adaptive multiquadric collocation for boundary layer problems, J. Comp. Appl. Math. 188 (2) (2006) 256–282.

[27] K. Diethelm, An algorithm for the numerical solution of differential equations of fractional order, Elec. Transact. Numer. Anal 5 (1997) 1–6.

[28] H. Brunner, Collocation methods for Volterra integral and related functional differential equations, Vol. 15 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2004.

[29] H. Brunner, A. Pedas, G. Vainikko, Piecewise polynomial collocation methods for linear Volterra integro-differential equations with weakly singular kernels, SIAM J. Numer. Anal. 39 (3) (2001) 957–982.

[30] L. Ling, R. Schaback, An improved subspace selection algorithm for meshless collocation methods, Int. J. Numer. Methods Eng. 80 (13) (2009) 1623–1639.

[31] L. Ling, R. Opfer, R. Schaback, Results on meshless collocation techniques, Eng. Anal. Bound. Elem. 30 (4) (2006) 247–253.

[32] L. Ling, R. Schaback, Stable and convergent unsymmetric meshless collocation methods, SIAM J. Numer. Anal. 46 (3) (2008) 1097–1115.

[33] T. O. Kwok, L. Ling, On convergence of a least-squares Kansa's method for the modified Helmholtz equations, Adv. Appl. Math. Mech. 1 (3) (2009) 367–382.

[34] C.-F. Lee, L. Ling, R. Schaback, On convergent numerical algorithms for unsymmetric collocation, Adv. Comput. Math. 30 (4) (2009) 339–354.

[35] Y. Luchko, Some uniqueness and existence results for the initial-boundary-value problems for the generalized time-fractional diffusion equation, Comput. Math. Appl. 59 (5) (2010) 1766 – 1772.

[36] K. Sakamoto, M. Yamamoto, Initial value/boundary value problems for fractional diffusion-wave equations and applications to some inverse problems, (preprint).

[37] H. J. Seybold, R. Hilfer, Numerical results for the generalized Mittag-Leffler function, Fract. Calc. Appl. Anal. 8 (2) (2005) 127–140.