

Novel numerical methods for time-space fractional reaction diffusion equations in two dimensions

Qianqian Yang¹ Tim Moroney² Kevin Burrage³
Ian Turner⁴ Fawang Liu⁵

(Received 15 December 2010; revised 29 June 2011)

Abstract

We consider time-space fractional reaction diffusion equations in two dimensions. This equation is obtained from the standard reaction diffusion equation by replacing the first order time derivative with the Caputo fractional derivative, and the second order space derivatives with the fractional Laplacian. Using the matrix transfer technique proposed by Ilić, Liu, Turner and Anh [Fract. Calc. Appl. Anal., 9:333–349, 2006] and the numerical solution strategy used by Yang, Turner, Liu, and Ilić [SIAM J. Scientific Computing, 33:1159–1180, 2011], the solution of the time-space fractional reaction diffusion equations in two dimensions can be written in terms of a matrix function vector product $f(\mathbf{A})\mathbf{b}$ at each time step, where \mathbf{A} is an approximate matrix representation of the standard Laplacian. We use the finite volume method over unstructured triangular meshes to generate the matrix \mathbf{A} ,

<http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/3791> gives this article, © Austral. Mathematical Soc. 2011. Published July 21, 2011. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

which is therefore non-symmetric. However, the standard Lanczos method for approximating $f(\mathbf{A})\mathbf{b}$ requires that \mathbf{A} is symmetric. We propose a simple and novel transformation in which the standard Lanczos method is still applicable to find $f(\mathbf{A})\mathbf{b}$, despite the loss of symmetry. Numerical results are presented to verify the accuracy and efficiency of our newly proposed numerical solution strategy.

Contents

1	Introduction	C396
2	Numerical scheme	C398
2.1	Finite volume method for spatial discretisation	C398
2.2	Matrix transfer technique for fractional Laplacian	C400
2.3	Finite difference method for the time fractional derivative	C401
2.4	Numerical scheme for the TSFRDE-2D	C401
3	Matrix function approximation and solution strategy	C402
4	Numerical results	C404
5	Conclusions	C406
	References	C407

1 Introduction

The theory of derivatives of fractional order has a long history. The concept is almost as old as its more familiar integer order counterpart. For three centuries the theory of fractional derivatives was considered a field for pure mathematicians. However, in the last few decades, fractional calculus gained

considerable popularity and importance among applied mathematicians, due to its demonstrated applications in fields such as electricity, chemistry, biology, economics, modelling, identification, control theory and signal processing [1, 4, 8]. It is important to investigate the solution behaviour of these fractional models and to explore their applications.

We consider the following two dimensional time-space fractional reaction diffusion equation (TSFRDE-2D) with homogeneous Dirichlet or Neumann boundary conditions

$${}_tD_*^\gamma \mathbf{u} = -K_\alpha (-\nabla^2)^{\alpha/2} \mathbf{u} + \mathbf{g}(\mathbf{u}), \quad (\mathbf{x}, \mathbf{y}) \in \Omega, \quad t > 0, \quad (1)$$

where $\mathbf{u}(\mathbf{x}, \mathbf{y}, t)$ is (for example) a concentration and K_α is the diffusion coefficient. The Caputo time fractional derivative of order γ ($0 < \gamma \leq 1$) with starting point at time $t = 0$ is defined [6] as

$${}_tD_*^\gamma \mathbf{u}(\mathbf{x}, \mathbf{y}, t) = \frac{1}{\Gamma(1-\gamma)} \int_0^t \frac{\mathbf{u}_t(\mathbf{x}, \mathbf{y}, \eta)}{(t-\eta)^\gamma} d\eta, \quad 0 < \gamma < 1, \quad (2)$$

with ${}_tD_*^\gamma \mathbf{u} = \partial \mathbf{u} / \partial t$ for $\gamma = 1$. The space fractional derivative $(-\nabla^2)^{\alpha/2}$ of order α ($1 < \alpha \leq 2$) is a fractional Laplacian operator defined through its eigenfunction expansion on the finite domain Ω [3, 12, 13]. The nonlinear reaction term $\mathbf{g}(\mathbf{u})$ is assumed to be Lipschitz continuous.

To solve the TSFRDE-2D, we first introduce a mesh and discretise in space to obtain an approximate matrix representation \mathbf{A} of the Laplacian $(-\nabla^2)$. Using the matrix transfer technique proposed by Ilić et al. [3], equation (1) is transformed into a system of time fractional differential equations involving the matrix \mathbf{A} raised to the fractional index $\alpha/2$

$${}_tD_*^\gamma \mathbf{u} = -K_\alpha \mathbf{A}^{\alpha/2} \mathbf{u} + \mathbf{g}(\mathbf{u}). \quad (3)$$

We use a finite difference method in time to handle the Caputo fractional derivative, which necessitates the computation of a matrix function vector product $\mathbf{f}(\mathbf{A})\mathbf{b}$ at each time step, where \mathbf{b} is a suitably defined vector.

Previously Yang et al. [13] successfully solved the two dimensional time-space fractional diffusion equation using Lanczos-based methods to approximate the action of the matrix function $f(\mathbf{A})$. When \mathbf{A} is generated from the finite difference method, it is symmetric and hence the standard Lanczos method applies. When \mathbf{A} is generated from the finite element method, it becomes non-symmetric due to the influence of the mass matrix. In this case, the M-Lanczos method [9] has been applied.

Now, we consider generating the matrix \mathbf{A} using the finite volume method. Although the matrix is again non-symmetric, we outline a novel way in which the standard Lanczos method still can be used to apply the required matrix functions. We illustrate our proposed numerical strategy on several test problems, including a fractional Fisher's equation on a disk.

2 Numerical scheme

We present the numerical scheme to simulate the solution behaviour of the TSFRDE-2D. Section 2.1 discusses the finite volume spatial discretisation for the non-fractional equation, and then Section 2.2 shows how to extend this discretisation to the fractional case using the matrix transfer technique. Section 2.3 discusses the discretisation of the Caputo time fractional derivative. Finally, Section 2.4 shows how these discretisations are combined in the final numerical scheme.

2.1 Finite volume method for spatial discretisation

The vertex centred finite volume method, also known as the control volume finite element (CV-FE) method, begins with a standard finite element mesh which is a partition of the domain Ω . We consider unstructured triangular meshes.

Let the number of nodes in the mesh be denoted N . Around each node, a control volume (CV) is constructed by connecting element centroids to face midpoints [2]. This generates a dual mesh of control volumes, and we denote by V_i the i th CV, which has volume¹ ΔV_i . These CVs form a partition of the domain Ω , so that $\Omega = \bigcup_{i=1}^N V_i$.

The CV faces are denoted by A_j , for $j = 1 \dots N_{CVF}$, where N_{CVF} is the number of CV faces in the mesh, and the area² of CV face A_j is ΔA_j . We let \mathcal{F}_i denote the set of indices of the CV faces that comprise the boundary Γ_i of V_i .

In deriving the finite volume discretisation of (1), we first consider the simpler, non-fractional equation

$$\frac{\partial \mathbf{u}}{\partial t} = -K_\alpha (-\nabla^2 \mathbf{u}) + g(\mathbf{u}), \tag{4}$$

with homogeneous Dirichlet or Neumann boundary conditions. The method proceeds by integrating (4) over control volume V_i :

$$\frac{d}{dt} \int_{V_i} \mathbf{u} \, dV = -K_\alpha \int_{\Gamma_i} (-\nabla \mathbf{u} \cdot \hat{\mathbf{n}}) \, d\sigma + \int_{V_i} g(\mathbf{u}) \, dV, \tag{5}$$

where the order of differentiation and integration has been interchanged on the left, and the divergence theorem applied to the first term on the right. Denoting by \mathbf{u}_i the (approximate) numerical solution at the i th node, substitute $\int_{V_i} \mathbf{u} \, dV = \Delta V_i \mathbf{u}_i$ and $\int_{V_i} g(\mathbf{u}) \, dV = \Delta V_i g(\mathbf{u}_i)$. Furthermore, approximate the surface integral by the sum of midpoint rule approximations over each face, to obtain

$$\Delta V_i \frac{d\mathbf{u}_i}{dt} = -K_\alpha \sum_{j \in \mathcal{F}_i} (-\nabla \mathbf{u} \cdot \hat{\mathbf{n}})_{mp_j} \Delta A_j + \Delta V_i g(\mathbf{u}_i) \tag{6}$$

where mp_j is the midpoint of the j th CV face.

¹Area, for two dimensional problems.

²Length, for two dimensional problems.

To approximate the flux at the CV face midpoint, interpolation is required. The CV-FE method uses the underlying finite element mesh as the mechanism for this interpolation. Within each element, the standard shape functions are used to construct an interpolant, and the gradient of this interpolant is evaluated at the CV face midpoint to find the necessary fluxes [2].

In this way, the expression for the total flux through the CV boundary Γ_i depends on the nodal value \mathbf{u}_i , as well as on the values \mathbf{u}_j for any node j that shares an element with node i . Applying (6) for each CV in the mesh, a system of differential equations is obtained

$$\mathbf{M} \frac{d\mathbf{u}}{dt} = -\mathbf{K}_\alpha \mathbf{K} \mathbf{u} + \mathbf{M} \mathbf{g}(\mathbf{u}) \quad (7)$$

where $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N)$ is the vector of nodal solution values. The matrix \mathbf{M} is diagonal with its diagonal entries the volumes of each CV. The matrix \mathbf{K} is sparse and symmetric, and its elements represent the contributions from each node to the total flux across each CV boundary.

Writing equation (7) as

$$\frac{d\mathbf{u}}{dt} = -\mathbf{K}_\alpha \mathbf{A} \mathbf{u} + \mathbf{g}(\mathbf{u}),$$

and comparing it with (4), we identify

$$\mathbf{A} = \mathbf{M}^{-1} \mathbf{K} \quad (8)$$

as the approximate matrix representation of the Laplacian $(-\nabla^2)$ under the CV-FE discretisation. Interestingly, although $(-\nabla^2)$ is a self-adjoint operator, its CV-FE matrix representation \mathbf{A} is non-symmetric.

2.2 Matrix transfer technique for fractional Laplacian

Under the matrix transfer technique proposed by Ilić et al. [3], the fractional Laplacian is approximated by

$$(-\nabla^2)^{\alpha/2} \mathbf{u} \approx \mathbf{A}^{\alpha/2} \mathbf{u}. \quad (9)$$

Hence, the TSFRDE-2D (1) is transformed into a system of time fractional differential equations as described in (3).

Although \mathbf{A} is sparse, $\mathbf{A}^{\alpha/2}$ will typically be dense. Fortunately, as discussed by Section 3, this matrix never needs to be formed, as only the action of a matrix function on a suitably chosen vector is required. We also emphasise that this representation assumes that homogeneous Dirichlet or Neumann conditions are imposed on the boundary. Extensions of this approach to non-homogeneous boundary conditions are discussed by Ilić et al. [3], but are not considered here.

2.3 Finite difference method for the time fractional derivative

Define times $t_n := n\tau$, $n = 0, 1, 2, \dots$, where τ is the time step. The Caputo time fractional derivative ${}_t D_*^\gamma \mathbf{u}$ is discretised using the implicit scheme [5]

$${}_t D_*^\gamma \mathbf{u}(t_n) \approx \frac{1}{\mu_0} \sum_{j=0}^{n-1} b_j [\mathbf{u}(t_{n-j}) - \mathbf{u}(t_{n-j-1})], \tag{10}$$

where $\mu_0 = \tau^\gamma \Gamma(2 - \gamma)$ and $b_j = (j + 1)^{1-\gamma} - j^{1-\gamma}$, $j = 0, 1, 2, \dots, n - 1$.

2.4 Numerical scheme for the TSFRDE-2D

Incorporating the approximations of the fractional Laplacian (9) and of the Caputo time fractional derivative (10), and treating the source term explicitly [11], we obtain the numerical approximation of the TSFRDE-2D (1)

$$\frac{1}{\mu_0} \sum_{j=0}^{n-1} b_j [\mathbf{u}^{n-j} - \mathbf{u}^{n-1-j}] = -K_\alpha \mathbf{A}^{\alpha/2} \mathbf{u}^n + \mathbf{g}(\mathbf{u}^{n-1}), \tag{11}$$

where $\mathbf{u}^n \approx \mathbf{u}(\mathbf{t}_n)$. After some further manipulations, (11) reads

$$\mathbf{u}^n = [\mathbf{I} + \mu_0 \mathbf{K}_\alpha \mathbf{A}^{\alpha/2}]^{-1} \left[\sum_{j=0}^{n-2} (\mathbf{b}_j - \mathbf{b}_{j+1}) \mathbf{u}^{n-1-j} + \mathbf{b}_{n-1} \mathbf{u}^0 + \mathbf{g}(\mathbf{u}^{n-1}) \right]. \quad (12)$$

Defining the scalar function $f(\xi) = [1 + \mu_0 \mathbf{K}_\alpha \xi^{\alpha/2}]^{-1}$, we obtain the numerical scheme for approximating the TSFRDE-2D (1) as

$$\mathbf{u}^n = f(\mathbf{A}) \mathbf{b}^n, \quad \text{with } \mathbf{b}^n = \sum_{j=0}^{n-2} (\mathbf{b}_j - \mathbf{b}_{j+1}) \mathbf{u}^{n-1-j} + \mathbf{b}_{n-1} \mathbf{u}^0 + \mathbf{g}(\mathbf{u}^{n-1}), \quad (13)$$

where \mathbf{u}^0 is the discrete representation of the initial value $\mathbf{u}(\mathbf{x}, \mathbf{y}, 0)$.

Remark Equation (13) is for the case $0 < \gamma < 1$. For the non-fractional case $\gamma = 1$, the temporal discretisation (10) should be replaced with the standard backward Euler scheme. Alternatively, a value $\gamma = 1 - \varepsilon$, for some small ε , can be used in the numerical scheme (13).

3 Matrix function approximation and solution strategy

We show how the matrix function vector product $f(\mathbf{A}) \mathbf{b}^n$ can be approximated at each time step without forming the (dense) matrix function $f(\mathbf{A})$. The prevailing method for approximating the matrix function vector product $f(\mathbf{A}) \mathbf{b}$ for a scalar, analytic function $f : \mathcal{D} \subset \mathbb{C} \rightarrow \mathbb{C}$ is the Lanczos approximation [7, 10]

$$f(\mathbf{A}) \mathbf{b} \approx \|\mathbf{b}\| \mathbf{V}_m f(\mathbf{T}_m) \mathbf{e}_1, \quad \mathbf{b} = \|\mathbf{b}\| \mathbf{V}_m \mathbf{e}_1, \quad (14)$$

where

$$\mathbf{A} \mathbf{V}_m = \mathbf{V}_m \mathbf{T}_m + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^T \quad (15)$$

is the Lanczos decomposition and the columns of \mathbf{V}_m form an orthonormal basis for the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{m-1}\mathbf{b}\}$, with $m \ll N$.

If \mathbf{A} is symmetric, then the matrix $\mathbf{T}_m \in \mathbb{R}^{m \times m}$ is symmetric and tridiagonal. Hence the required matrix function $f(\mathbf{T}_m)$ is efficiently found by diagonalising \mathbf{T}_m . However, equation (8) shows that under the CV-FE spatial discretisation the matrix \mathbf{A} is not symmetric, even though both \mathbf{M} and \mathbf{K} are symmetric. We now outline a simple and novel transformation that allows the standard Lanczos method to be used to find $f(\mathbf{A})\mathbf{b}$, despite the loss of symmetry.

First, we observe that \mathbf{A} is similar to a symmetric matrix $\tilde{\mathbf{A}}$, since

$$\mathbf{A} = \mathbf{M}^{-1}\mathbf{K} = \mathbf{M}^{-1/2}\mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}\mathbf{M}^{1/2} = \mathbf{M}^{-1/2}\tilde{\mathbf{A}}\mathbf{M}^{1/2},$$

where $\tilde{\mathbf{A}} = \mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$ is symmetric. Let $\tilde{\mathbf{A}} = \tilde{\mathbf{P}}\mathbf{D}\tilde{\mathbf{P}}^T$ be an orthogonal diagonalisation of $\tilde{\mathbf{A}}$. Then we find a diagonalisation of \mathbf{A} as

$$\mathbf{A} = \mathbf{M}^{-1/2}\tilde{\mathbf{A}}\mathbf{M}^{1/2} = \mathbf{M}^{-1/2}\tilde{\mathbf{P}}\mathbf{D}\tilde{\mathbf{P}}^T\mathbf{M}^{1/2} = (\mathbf{M}^{-1/2}\tilde{\mathbf{P}})\mathbf{D}(\mathbf{M}^{-1/2}\tilde{\mathbf{P}})^{-1},$$

and hence $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ where $\mathbf{P} = \mathbf{M}^{-1/2}\tilde{\mathbf{P}}$. With this result, we now relate the two matrix functions $f(\mathbf{A})$ and $f(\tilde{\mathbf{A}})$

$$\begin{aligned} f(\mathbf{A}) &= \mathbf{P}f(\mathbf{D})\mathbf{P}^{-1} = \mathbf{M}^{-1/2}\tilde{\mathbf{P}}f(\mathbf{D})(\mathbf{M}^{-1/2}\tilde{\mathbf{P}})^{-1} \\ &= \mathbf{M}^{-1/2}\tilde{\mathbf{P}}f(\mathbf{D})\tilde{\mathbf{P}}^T\mathbf{M}^{1/2} = \mathbf{M}^{-1/2}f(\tilde{\mathbf{A}})\mathbf{M}^{1/2}. \end{aligned} \tag{16}$$

Hence, the solution (13) is computed as

$$\mathbf{u}^n = f(\mathbf{A})\mathbf{b}^n = \mathbf{M}^{-1/2}f(\tilde{\mathbf{A}})\mathbf{M}^{1/2}\mathbf{b}^n = \mathbf{M}^{-1/2}f(\tilde{\mathbf{A}})\tilde{\mathbf{b}}, \tag{17}$$

where $\tilde{\mathbf{b}} = \mathbf{M}^{1/2}\mathbf{b}^n$ and the symmetric matrix function vector product $f(\tilde{\mathbf{A}})\tilde{\mathbf{b}}$ is computed using the standard Lanczos method. We emphasise again that neither \mathbf{A} nor $\tilde{\mathbf{A}}$ is formed during the algorithm.

TABLE 1: Spatial and temporal errors at time $t = 0.01$ with $\alpha = 1.3$ and $\gamma = 0.5$.

h	Spatial error	τ	Temporal error
0.1253	1.7E-4	0.001	4.8E-4
0.0676	5.7E-5	0.0005	2.3E-4
0.0342	1.7E-5	0.00025	1.1E-4
0.0176	4.6E-6	0.000125	5.6E-5
0.0091	1.2E-6	0.0000625	2.7E-5
Order	2.0	Order	1.0

4 Numerical results

Numerical experiments were carried out to assess the computational performance and accuracy of our solution strategy, as well as to illustrate the effect of the fractional order on the solution.

Example 1 Consider the TSFRDE-2D (1) on the domain $\Omega = [0, 1] \times [0, 1]$ with $K_\alpha = 1$, $g(\mathbf{u}) = 0$, initial condition $\mathbf{u}_0(x, y) = xy(1 - x)(1 - y)$ and boundary conditions $\mathbf{u} = 0$ on $\partial\Omega$.

With $g(\mathbf{u}) = 0$, the TSFRDE-2D becomes linear. The analytical solution of this linear time-space fractional diffusion equation is [13]

$$u(x, y, t) = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} E_\gamma(-\lambda_{n,m}^{\alpha/2} t^\gamma) c_{n,m} \varphi_{n,m}, \tag{18}$$

where $\lambda_{n,m} = n^2\pi^2 + m^2\pi^2$ and $\varphi_{n,m} = 2 \sin(n\pi x) \sin(m\pi y)$ are the eigenvalues and eigenfunctions of the two dimensional Laplacian $(-\nabla^2)$, $c_{n,m} = \int_0^1 \int_0^1 xy(1 - x)(1 - y)\varphi_{n,m} \, dy \, dx$ are the Fourier coefficients of the initial condition, and $E_\gamma(\cdot)$ is the Mittag-Leffler function.

To identify the order of convergence in space, in the second column of Table 1, we compute the maximum errors in the numerical solution at time $t = 0.01$,

with $\alpha = 1.3$ and $\gamma = 0.5$, for a sequence of refined meshes with 328, 1 342, 5 422, 21 962 and 88 790 elements, and a small time-step ($\tau = 5 \times 10^{-6}$). The convergence in space is estimated to be $\mathcal{O}(\mathbf{h}^2)$, where \mathbf{h} is the maximum element diameter. This is consistent with the standard result for CV-FE applied to the non-fractional Laplacian [2].

To identify the order of convergence in time, in the fourth column of Table 1, we present the maximum errors in the numerical solution at time $t = 0.01$ on the finest mesh with 88 790 elements ($\mathbf{h} = 0.0091$), $\alpha = 1.3$, and $\gamma = 0.5$. The convergence in time-step is estimated to be $\mathcal{O}(\tau)$.

Example 2 To illustrate the effect of the fractional order in space, we consider the population dynamics of bacteria growing in a Petri dish. One model of this process is Fisher's equation, for which we take the TSFRDE-2D (1) on the domain $\Omega = \{(x, y) \mid x^2 + y^2 \leq 1\}$, $g(\mathbf{u}) = \mathbf{u}(1 - \mathbf{u})$, and $\partial \mathbf{u} / \partial \mathbf{n} = 0$ on $\partial \Omega$. Here, \mathbf{u} represents the concentration of bacteria.

We discretise the unit disk using an unstructured triangular mesh with 2900 elements, and place a small initial concentration of 0.01 at the node closest to $(0.1, -0.1)$. Holding $\gamma = 1$, we compare the evolution of the fractional diffusion model ($\alpha = 1.5$) and the standard diffusion model ($\alpha = 2.0$). The diffusion coefficients are taken to be $K_\alpha = 5 \times 10^{-4}$ and $K_\alpha = 16 \times 10^{-4}$ respectively, so that both models reach 80% of their carrying capacity at approximately the same time.

Figure 1 illustrates the spatial distribution of concentration of both models at two points in time. At early time, the sharper profile of the fractional diffusion model is evident, while the standard diffusion model exhibits the expected Gaussian-like profile. The late time profiles correspond to the time at which both models reached 80% of their carrying capacity. The increased growth of the fractional diffusion model near the boundaries is due to its heavier tailed diffusion, resulting in more concentration reaching the extremities of the domain and promoting growth there. For the standard diffusion model, there is a larger region of high concentration in the interior, but less growth

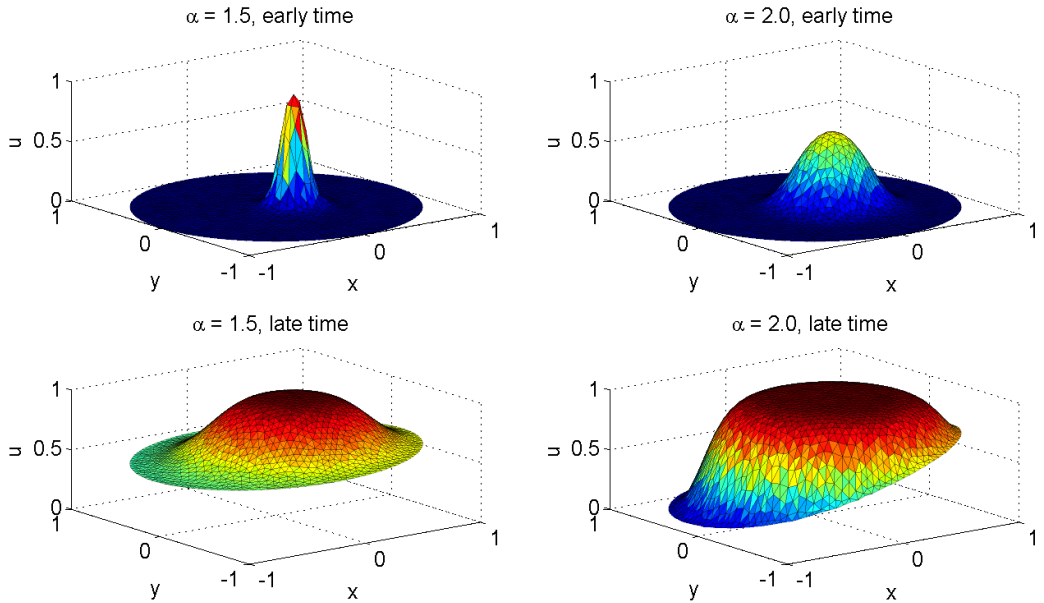


FIGURE 1: Spatial distribution of concentration at early time (top) and late time (bottom) for $\alpha = 1.5$, $K_\alpha = 5 \times 10^{-4}$ (left) and $\alpha = 2.0$, $K_\alpha = 16 \times 10^{-4}$ (right). The movie [video.avi](http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/downloadSuppFile/3791/832) shows a side-by-side comparison of the two [<http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/downloadSuppFile/3791/832>].

towards the boundaries. The impact of the initial concentration being placed slightly off-centre is also much more evident in the standard model.

5 Conclusions

We demonstrate a novel approach for solving time-space fractional reaction diffusion equations in two dimensions. In particular, our approach allows the standard Lanczos method to be used to approximate the required matrix

function vector products, even though the CV-FE matrix representation of the Laplacian is nonsymmetric. The method recovers the expected second order spatial and first order temporal accuracy when applied to a linear test problem. A model simulating bacteria growing in a Petri dish illustrates some interesting population dynamics that can be modelled using a fractional reaction diffusion equation.

References

- [1] D. Baleanu, Z. B. Guvenc, J. A. T. Machado. (Eds.) *New Trends in Nanotechnology and Fractional Calculus Applications*. Springer, 2010. [C397](#)
- [2] R. E. Ewing, T. Lin and Y. Lin. On the accuracy of the finite volume element method based on piecewise linear polynomials. *SIAM J. Numerical Analysis* 39(6):1865–1888, 2002. [doi:10.1137/S0036142900368873](#) [C399](#), [C400](#), [C405](#)
- [3] M. Ilić, F. Liu, I. Turner, and V. Anh. Numerical approximation of a fractional-in-space diffusion equation (II)—with nonhomogeneous boundary conditions. *Fract. Calc. Appl. Anal.*, 9:333–349, 2006. <http://hdl.handle.net/10525/1287> [C397](#), [C400](#), [C401](#)
- [4] R. Klages, G. Radons, I. M. Sokolov. (Eds.) *Anomalous Transport: Foundations and Applications*. Wiley, 2008. [C397](#)
- [5] Y. Lin and C. Xu. Finite difference/spectral approximations for the time-fractional diffusion equation. *J. Comp. Phys.*, 225:1533–1552, 2007. [doi:10.1016/j.jcp.2007.02.001](#) [C401](#)
- [6] I. Podlubny. *Fractional Differential Equations*. Academic Press, New York, 1999. [C397](#)

- [7] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29:209–228, 1992. doi:10.1137/0729014 C402
- [8] J. Sabatier, O. P. Agrawal, J. A. T. Machado. (Eds.) *Advances in fractional calculus: Theoretical developments and applications in physics and engineering*. Springer, 2007. C397
- [9] V. Simoncini and D. B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numer. Linear Algebra Appl.*, 14:1–59, 2007. doi:10.1002/nla.499 C398
- [10] H. A. van der Vorst. An iterative solution method for solving $f(A)x = b$ using Krylov subspace information obtained for the symmetric positive definite matrix A . *J. Comput. Appl. Math.*, 18:249–263, 1987. doi:10.1016/0377-0427(87)90020-3 C402
- [11] Q. Yang, F. Liu, and I. Turner. Stability and convergence of an effective numerical method for the time-space fractional Fokker–Planck equation with a nonlinear source term. *International Journal of Differential Equations*, Article ID 464321, 22 pages, 2010. doi:10.1155/2010/464321 C401
- [12] Q. Yang, F. Liu, and I. Turner. Numerical methods for fractional partial differential equations with Riesz space fractional derivatives. *Applied Mathematical Modelling*, 34:200–218, 2010. doi:10.1016/j.apm.2009.04.006 C397
- [13] Q. Yang, I. Turner, F. Liu, and M. Ilić. Novel numerical methods for solving the time-space fractional diffusion equation in 2D. *SIAM J. Scientific Computing*, 33:1159–1180, 2011. doi:10.1137/100800634 C397, C398, C404

Author addresses

1. **Qianqian Yang**, Discipline of Mathematical Sciences, Queensland University of Technology, Brisbane, Queensland 4001, AUSTRALIA and Institute for Molecular Bioscience, University of Queensland, Brisbane, Queensland 4072, AUSTRALIA.
<mailto:q.yang@qut.edu.au>
2. **Tim Moroney**, Discipline of Mathematical Sciences, Queensland University of Technology, Brisbane, Queensland 4001, AUSTRALIA.
<mailto:t.moroney@qut.edu.au>
3. **Kevin Burrage**, Discipline of Mathematical Sciences, Queensland University of Technology, Brisbane, Queensland 4001, AUSTRALIA; Institute for Molecular Bioscience, University of Queensland, Brisbane, Queensland 4072, AUSTRALIA; and Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK.
<mailto:kevin.burrage@comlab.ox.ac.uk>, kevin.burrage@qut.edu.au
4. **Ian Turner**, Discipline of Mathematical Sciences, Queensland University of Technology, Brisbane, Queensland 4001, AUSTRALIA.
<mailto:i.turner@qut.edu.au>
5. **Fawang Liu**, Discipline of Mathematical Sciences, Queensland University of Technology, Brisbane, Queensland 4001, AUSTRALIA.
<mailto:f.liu@qut.edu.au>