AMSI
AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

National Collaboration in the Mathematical Sciences

Efficient preconditioning of Krylov subspace methods with applications for fractional PDEs

Alex Simmons
Supervisor: Dr. Qianqian Yang
Queensland University of Technology

February 2014

1 Introduction

The theory of fractional derivatives and integrals dates back hundreds of years to a letter from Leibniz discussing with L'Hopital the meaning of a derivative of order one half. It has only been in the last few decades that there has been considerable interest in utilising fractional calculus for important practical applications. Due to the recent explosion in popularity there is a significant demand for efficient techniques to stimulate insight into the behaviors of fractional models.

Fractional derivatives can be used to model diffusion-type processes where the underlying particle motion deviates from Brownian motion [3]. Due to this, typical applications are models based on transport in porous media such as seepage flow [29], coastal aquifer salt-water intrusion [7] or, wood drying [36]. Models are not restricted to transport in porous media, work is being conducted in drug delivery models [45], heart physiology [21, 35], bioengineering [24] and optimal image processing [6].

It has been well established that analytical methods exist for only a small number of special, simple, mostly linear fractional models. To determine solutions to more complicated fractional models, efficient numerical methods are preferred and, in most cases, required. Compared to integer-order methods, fractional models pose significant computational challenges. In the last decade substantial work has been completed in determining efficient numerical methods for approximating fractional models, including finite difference methods [26, 25, 23, 5], finite volume methods [40, 31], finite elements methods [44, 37], spectral methods [39, 1] and mesh-free methods [41, 30].

Postal Address: 111 Barry Street
c/- The University of Melbourne
Victoria 3010 Australia

Email: enquiries@amsi.org.au
Phone: +61 3 8344 1777
Fax: +61 3 9349 4106
Web: www.amsi.org.au

In this paper we consider the space-fractional diffusion equation (SFDE) with non-linear source term

$$\frac{\partial u(x,t)}{\partial t} = -\kappa \left(-\nabla^2\right)^{\alpha/2} u(x,t) + S(u(x,t),x,t) \tag{1}$$

on the finite domain $0 \leq x \leq L$ with homogeneous Dirichlet boundary conditions and initial condition $u(x,0) = u_0(x)$ with the fractional order satisfying $1 < \alpha \leq 2$.

Definition 1 [17]. Suppose the Laplacian $(-\nabla^2)$ has a complete set of orthonormal eigenfunctions φ_n corresponding to eigenvalues λ_n^2 on a bounded region $\mathcal{D}$, i.e, $(-\nabla^2)\varphi_n = \lambda_n^2 \varphi_n$ on a bounded region $\mathcal{D}$; $\mathcal{B}(\varphi) = 0$ on $\partial\mathcal{B}$, where $\mathcal{B}(\varphi)$ is one of the standard three homogeneous boundary conditions. Let

$$\mathcal{F}_\gamma = \left\{ f = \sum_{n=1}^{\infty} c_n \varphi_n, c_n = \langle f, \varphi_n \rangle, \sum_{n=1}^{\infty} |c_n||\lambda|_n^\gamma < \infty, \gamma = \max(\alpha, 0) \right\}$$

then for any $f \in \mathcal{F}_\gamma$, $(-\nabla^2)^{\alpha/2} f$ is defined by

$$(-\nabla^2)^{\alpha/2} f = \sum_{n=1}^{\infty} c_n (\lambda_n^2)^{\alpha/2} \varphi_n.$$

The most significant challenge faced by researchers in this area is the high computational-expense of obtaining numerical solutions to fractional models, such as equation (1). When using techniques such as the Matrix Transfer Technique proposed by Ilić et al [18] to convert a SFDE into a system of differential equations, due to the non-local nature of the fractional derivative, the resulting systems have a fully dense nature. This can be seen when we spatially discretise our model in equation (2) and signifies a need for high-efficiency numerical methods that can significantly reduce the amount of computer time required to approximate these dense fractional models.

To spatially discretise our system we introduce a mesh with $N+1$ uniformly distributed nodes of width $h = L/N$, and $N+1$ nodes x_i for $i = 0 \ldots N$, where $x_i = ih$. To transform our SFDE from equation (1) into a system of time ordinary differential equations we use the Matrix Transfer Technique introduced by Ilić et al [18]. To apply the Matrix Transfer Technique we use the spatial discretisation within a method of lines framework such that $(-\nabla^2)^{\alpha/2} u \approx \mathbf{A}^{\alpha/2}\mathbf{u}(t)$ with homogeneous Dirichlet boundary conditions where $\mathbf{A}$ is $\texttt{tridiag}(-1, 2, -1)$ and $\mathbf{u}(t)$ is the spatially discretised approximation to the solution $u(x,t)$; from this it follows that the system of non-linear time ODEs has the form

$$\dot{\mathbf{u}} = \mathbf{F}(\mathbf{u}, t) := -\kappa \mathbf{A}^{\alpha/2}\mathbf{u}(t) + \mathbf{S}(\mathbf{u}, \mathbf{x}, t), \quad \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}). \tag{2}$$

For homogeneous Dirichlet boundary conditions the discretisation in equation (2) applies only to the inner nodes $x_i \; \forall \; 0 < i < N+1$, the outer boundary nodes x_0 and x_{N+1} are set to zero. When $1 < \alpha < 2$ then it is important to highlight that $\mathbf{A}^{\alpha/2}$ has a fully dense nature. This presents computational challenges that stimulates our discussion in Section 3, most significantly, and the justification for our new preconditioner, the Jacobian matrix $\mathbf{J} = \partial \mathbf{F}/\partial \mathbf{u}$ is fully dense.

One of the computational techniques that we implement is Newton's method to handle the non-linearity within equation (2). Due to the non-local nature of the fractional derivative we require the solution to dense linear systems owing to the dense Jacobians that result from the fractional Laplacian. To avoid explicitly forming these dense Jacobians, the Jacobian-Free Newton-Krylov (JFNK) method [20] is employed. This method has been considered previously by Moroney and Yang [28, 27]. We also introduce high order timestepping through backwards differentiation formulas (BDF).

We implement the Generalised Minimal Residual method from the class of Krylov subspace techniques to form the Krylov component of the JFNK method because of its ability to approximate linear systems without the need to operate directly on the matrix. We also investigate the Lanczos method for approximating matrix-function-vector-products $f(\mathbf{A})\mathbf{v} = \mathbf{A}^{\alpha/2}\mathbf{v}$ without the need to explicitly form the memory intensive matrix function $\mathbf{A}^{\alpha/2}$. This technique has been heavily considered by many authors such as Van der Vorst [38], Saad [32], Druskin and Knizhnerman [8], Hochbruck and Lubich [14], Sidje [34], van den Eshof [10], Eiermann and Ernst [9], Lopez and Simoncini [22], Ilić, Turner, and Anh [15] and Ilić, Turner, and Simpson [16] as well as many other researchers during the last twenty years. The primary concern with using Krylov subspace techniques is the requirement of an effective preconditioner to accelerate convergence of the technique.

Implicitly applied preconditioners has been a particular interest in literature due to their ability to avoid the high cost of dense matrix multiplication for significantly large problems. Due to the prevalence of Krylov subspace methods for approximating the solutions to linear systems or matrix-function-vector-products Erhel et al [19] developed a preconditioner based on eigenvalue deflation. Burrage et al. [4] considered both algebraic multigrid and incomplete LU preconditioning. For preconditioning two-sided nonlinear SFDE Moroney and Yang developed a fast Poisson preconditioner [28] and a banded preconditioner [27].

The focus of this paper is the development of a new banded preconditioner for accelerating the Generalised Minimal Residual method component of the JFNK method, without which, runtimes are orders of magnitude larger than what can be achieved. To put this work in context, we begin by discussing the key numerical concepts that

we utilise to derive approximate solutions to our nonlinear SFDE. Once this has been conducted, arguments are formed that justify the requirements of a new preconditioner before moving on to discussing the processes undertaken. Numerical results are then derived that justify the effectiveness of our preconditioner and conclusions are then drawn.

While Krylov subspace techniques is a continuous theme within this paper, we also consider the contour Integral Method refined by Hale et al [12] for approximating matrix-function-vector-products. Through investigation we arrive at a preference of the contour integral method over the Lanczos method for approximating matrix-function-vector-products due to significant runtime reductions in 1-D problems.

The remainder of this paper is as follows. In Section 2 we introduce the BDF for integrating the initial value problem that results from equation (2), we also introduce the JFNK method [20]. In Section 3 we introduce the banded preconditioner that we have developed. We use a discretisation of the Riesz fractional derivative and a summation over a few local neighbours using the shifted Grünwald-Letnikov approximation to approximate the discretised fractional Laplacian. We then consider two methods for approximating $f(\mathbf{A})\mathbf{b}$, the Lanczos method from the class of Krylov subspace methods as well and the contour integral method. We discuss briefly the theory behind the two methods before using numerical experiments as a justification for a preference towards the contour integral method for our purposes. In Section 5 we conduct some numerical experiments where we solve our space-fractional nonlinear diffusion equation. We demonstrate the infeasibility of not implementing our banded preconditioner as well as the significant acceleration that is gained from its implementation. We conclude this paper in Section 6.

1.1 Backwards differentiation formulas

Backwards differentiation formulas are a family of implicit multistep methods for solving initial value problems [11]. Given a solution at time t_{n-1}, where $\mathbf{u}(t_{n-1}, \mathbf{x}) \approx \mathbf{u}_{n-1}(\mathbf{x})$ is known then the next point in time $t_n = t_{n-1} + \tau_n$ where τ_n is the stepsize [13]. The generalised class of BDF formulas for determining an approximation to the derivative $\dot{\mathbf{u}}(t)$ is

$$\dot{\mathbf{u}} \approx \frac{1}{\tau_n} \sum_{p=0}^{q} \beta_{n,p} \mathbf{u}_{n-p}. \tag{3}$$

where q is the order of the BDF, and the coefficients $\beta_{n,p}$ depend on the recent order and stepsize. One of the most common and best-know BDF is the backwards Euler method which corresponds to $q = 1$ and $\beta_{n,0} = 1$ and $\beta_{n,1} = -1$.

In our implementation we utilise the powerful CVODE package for stiff IVP-BDF problems from the SUNDIALS (Suite of Nonlinear and Differential/Algebraic Equation Solvers) suite [13]. The justification for using modern IVP-BDF solvers such as CVODE is due to their use of sophisticated algorithms to adaptively adjust the timestep and order size within a predetermined tolerance level. This allows the stepsize to be kept as large as possible while still maintaining accuracy, providing significant performance gains over hard-coded methods such as backwards Euler or similar.

Substituting our BDF (3) into (2)

$$\frac{1}{\tau_n}\sum_{p=0}^{q}\beta_{n,p}\mathbf{u}_{n-p} = -\kappa\mathbf{A}^{\alpha/2}\mathbf{u} + \mathbf{S}(\mathbf{u},\mathbf{x},t).$$

Rearranging for the unknown solution $\mathbf{u}_n$,

$$\mathbf{G}_n(\mathbf{u}_n) := \mathbf{u}_n - \gamma_n\mathbf{F}(\mathbf{u}_n,t_n) + \mathbf{a}_n = 0 \tag{4}$$

where $\gamma_n = \tau_n/\beta_{n,0}$ and $\mathbf{a}_n = \sum_{p=1}^{q}(\beta_{n,p}/\beta_{n,0})\mathbf{u}_{n-p}$. At this point we introduce Newton's method applied to (4). Let $\mathbf{u}_n^k$ be the kth iterate in the sequence $\{\mathbf{u}_n^k\}_{k=0}^{\infty} \to \mathbf{u}_n$ with correction vector $\delta\mathbf{u}_n^k$:

$$\mathbf{u}_n^{k+1} = \mathbf{u}_n^k + \delta\mathbf{u}_n^k. \tag{5}$$

To obtain $\delta\mathbf{u}_n^k$ we must solve

$$\Big(\mathbf{I} - \gamma_n\mathbf{J}(\mathbf{u}_n^k)\Big)\delta\mathbf{u}_n^k = -\mathbf{G}_n(\mathbf{u}_n^k). \tag{6}$$

$\mathbf{J} = \partial\mathbf{F}/\partial\mathbf{u}$ has a dense nature and in-turn poses significant memory requirements if we were to explicitly form it. To avoid this issue we choose to implement the Jacobian-Free Newton-Krylov method to avoid the forming and solving $\mathbf{J}(\mathbf{u}_n^k)$.

1.2 Jacobian-free Newton-Krylov methods

To avoid explicitly solving (6) we approximate a solution to $\delta\mathbf{u}_n^k$ by introducing Krylov subspace methods. As a brief introduction, Krylov subspace methods for solving $\mathbf{A}_J\mathbf{x} = \mathbf{b}$ is a projection method onto the Krylov subspace

$$\mathcal{K}_m(\mathbf{A}_J,\mathbf{b}) = \text{span}\{\mathbf{b},\mathbf{A}_J\mathbf{b},\ldots,\mathbf{A}_J^{m-1}\mathbf{b}\}. \tag{7}$$

Applied to (6) such that $\mathbf{A}_J = \mathbf{I}-\gamma_n\mathbf{J}(\mathbf{u}_n^k)$, $\mathbf{x} = \delta\mathbf{u}_n^k$ and $\mathbf{b} = -\mathbf{G}_n(\mathbf{u}_n)$ then the action of the matrix $\mathbf{A}_J$ on some suitably chosen vector $\mathbf{v}$ is required

$$\mathbf{A}_J\mathbf{v} = \mathbf{v} - \gamma_n\mathbf{J}(\mathbf{u}_n^k)\mathbf{v}.$$

It is important to notice that the Jacobian-vector product can be approximated [20] without the explicit formation of the Jacobian $\mathbf{J}$ with

$$\mathbf{J}(\mathbf{u}_n^k)\mathbf{v} \approx \frac{\mathbf{F}(\mathbf{u}_n^k + \epsilon\mathbf{v}, t_n) - \mathbf{F}(\mathbf{u}_n^k, t_n)}{\epsilon} \tag{8}$$

and a suitably chosen shift value ϵ. With the formation of $\mathbf{F}(\mathbf{u}_n^k, t_n)$, each successive Jacobian-vector product only requires one additional formulation of the RHS vector $\mathbf{F}$. This Jacobian-free approach, combined with Newton's method, leads to a class of Jacobian-free Newton-Krylov methods for solving (4).

The introduction of Krylov subspace methods presents a significant difficulty. Due to the wide range of time scales present in the semidiscrete system, we are faced with the well known problem of "stiffness". This significantly hinders the performance of the Krylov subspace method [13, 20, 33]. To overcome this, the standard procedure is to precondition the system before applying the Krylov subspace method.

Rather than solve $\mathbf{A}_J\mathbf{x} = \mathbf{b}$ directly, we solve

$$\mathbf{M}^{-1}\mathbf{A}_J\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$$

by projecting onto the Krylov subspace

$$\mathcal{K}_m(\mathbf{M}^{-1}\mathbf{A}_J, \mathbf{b}) = \text{span}\{\mathbf{b}, (\mathbf{M}^{-1}\mathbf{A}_J)\mathbf{b}, \ldots, (\mathbf{M}^{-1}\mathbf{A}_J)^{m-1}\mathbf{b}\}. \tag{9}$$

The idea is that, if $\mathbf{M}^{-1}$ closely resembles $\mathbf{A}^{-1}$ or captures a significant amount of the "stiffness" in the problem then convergence of the Krylov subspace method should be significantly accelerated such that $m \ll n$ where $\mathbf{A} \in \mathbb{R}^{n\times n}$. For our preconditioner $\mathbf{M}^{-1}$ it should be easy to form and factorise yet still capture the most significant amount of underlying physics within our problem. How we achieve this is covered in Section 4.

2 Banded Preconditioner

In this section we define the form of our preconditioner $\mathbf{M}^{-1}$. This is achieved by expanding the Jacobian in $\mathbf{A}_J$

$$\mathbf{A}_J = \mathbf{I} - \gamma_n\mathbf{J} = \mathbf{I} - \gamma_n\frac{\partial\mathbf{F}}{\partial\mathbf{u}} = \mathbf{I} - \gamma_n\left(-\kappa\mathbf{A}^{\alpha/2} + \frac{\partial\mathbf{S}}{\partial\mathbf{u}}\right). \tag{10}$$

It is evident from (10) that the dense nature owes to the matrix $\mathbf{A}_J^{\alpha/2}$. What we propose is to form some banded matrix $\mathbf{B} \approx \mathbf{A}_J^{\alpha/2}$. If the bandwidth of $\mathbf{B}$ is sufficiently small

Postal Address: 111 Barry Street
c/- The University of Melbourne
Victoria 3010 Australia

Email: enquiries@amsi.org.au
Phone: +61 3 8344 1777
Fax: +61 3 9349 4106
Web: www.amsi.org.au

yet still captures a significant proportion of the physical dynamics in our problem then our preconditioner

$$\mathbf{M} = \mathbf{I} - \gamma_n \left(-\kappa \mathbf{B} + \frac{\partial \mathbf{S}}{\partial \mathbf{u}} \right) \tag{11}$$

should be easily factorisable. To form the banded approximation to the discretised fractional Laplacian we employ the Riesz fractional derivative [42]. We then utilise the shifted-Grünwald Letnikov approximation to generate a discrestised representation of the Riesz fractional derivative

$$g_{\alpha,0} = 1, \quad g_{\alpha,j} = (-1)^j \frac{\alpha(\alpha-1)\dots(\alpha-j+1)}{j!}, \tag{12}$$

$$\mathbf{B}_{ij} = \frac{1}{2\cos\left(\frac{\pi\alpha}{2}\right) h^\alpha} \begin{cases} g_{\alpha,i-j+1}, & j < i-1, \\ g_{\alpha,2} + g_{\alpha,0}, & j = i-1, \\ g_{\alpha,1}, & j = i, \\ g_{\alpha,2} + g_{\alpha,0}, & j = i+1, \\ g_{\alpha,i-j+1}, & j > i+1. \end{cases} \tag{13}$$

Recalling from earlier, we argued that if the preconditioner $\mathbf{M}^{-1}$ has a significantly small bandwidth than it is easily factorised. By forming our matrix $\mathbf{B}$ from only the central diagonal elements defined by the Riesz fractional derivative this is achievable while still capturing the significant physical processes. Shown in 1 is evidence that the most significant elements in the discretised fractional Laplacian $\mathbf{A}^{\alpha/2}$ lay around the diagonal.

We demonstrate using Figure 2 that the matrix $\mathbf{B}$ defined from (12)-(13) captures the most significant elements of $\mathbf{A}^{\alpha/2}$ while still maintaining a relatively small bandwidth. It is known that there are two sources of error when attempting to approximate $\mathbf{A}^{\alpha/2}$ with our matrix $\mathbf{B}$; the banded nature of $\mathbf{B}$ and the difference in the operator definitions.

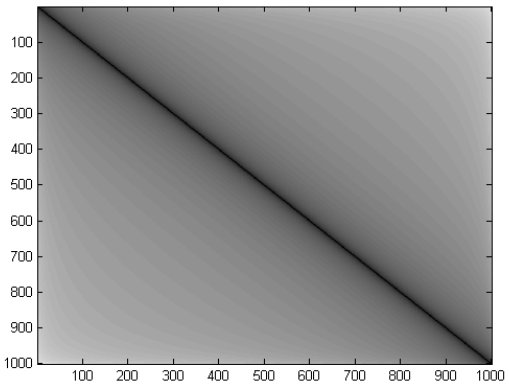

Figure 1: Magnitude of the elements of $\log_{10}(|\mathbf{A}^{\alpha/2}|)$ with dimension 1000×1000. Demonstrates that the points that are most significant in modelling the system lay around the diagonal.

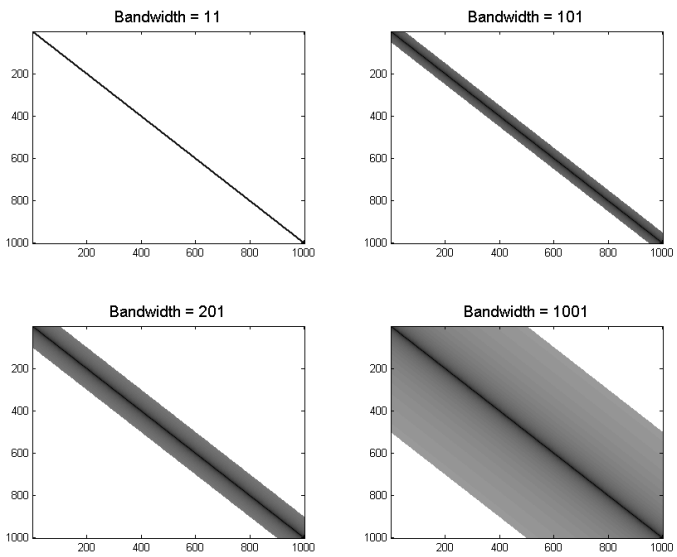

Figure 2: Magnitude of the elements of $\log_{10}(|\mathbf{B}|)$ with varying bandwidths of $b = \{11, 101, 201, 1001\}$. The colour mapping used in Figure 1 is reused in all subplots presented here.

3 Approximating $f(\mathbf{A})\mathbf{b}$

A significant issue that needs to be highlighted is the evaluation of $\mathbf{A}^{\alpha/2}\mathbf{b}$ required in the JFNK method. To maintain a consistent theme, we wish to avoid the explicit

formulation of the dense matrix $\mathbf{A}^{\alpha/2}$. Covered in this paper are two methods, the Lanczos method from the class of Krylov subspace methods, and the contour integral method. In this section, we briefly cover the theory behind these two methods before justifying our preference for the contour integral method with some numerical results.

3.1 Preconditioned Lanczos Method

In this section we briefly cover the Lanczos method with the application of an adaptive preconditioner, a prevailing method in literature for approximating the maxtrix-function-vector product $f(\mathbf{A})\mathbf{b}$ approximation. For a scalar, analytic function $f : D \subset \mathbb{C} \to \mathbb{C}$ then

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

where

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{T}_m + \beta_m\mathbf{v}_{m+1}\mathbf{e}_m^T. \tag{15}$$

is the Lanczos decomposition with the columns of $\mathbf{V}_m$ forming an orthonormal basis for the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{m-1}\mathbf{b}\}$ and $\mathbf{T}_m$ is symmetric and tridiagonal when $\mathbf{A}$ is symmetric [38]. To accelerate convergence of the Lanczos method we implement an adaptive preconditioner $\mathbf{Z}^{-1}$ [19]. This adaptive preconditioner has been considered by Yang et al. [44, 43], Ilić et al. [15, 16], Baglama et al. [2] and, others. This adaptive preconditioner shifts the k smallest eigenvalues which are known to hinder convergence of the Lanczos method.

We compute the k smallest eigenvalues $\{\lambda_i\}_{i=1}^k$ and corresponding eigenvectors $\{\mathbf{q}_i\}_{i=1}^k$ of the matrix $\mathbf{A}$. Methods exists to do this efficiently but for our problem, they are known exactly. Setting $\mathbf{Q}_k = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k]$ and $\mathbf{\Lambda}_k = \text{diag}\{\lambda_1, \ldots, \lambda_k\}$ then Erhel et al. [19] proposes the preconditioner

$$\mathbf{Z}^{-1} = \theta^*\mathbf{Q}_k\mathbf{\Lambda}_k\mathbf{Q}_k^T + \mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^T. \tag{16}$$

where $\theta^* = \frac{\lambda_{min}+\lambda_{max}}{2}$. λ_{min} and λ_{max} are the smallest and largest eigenvalues of $\mathbf{A}$ respectively. Baglama et al. [2] demonstrates that $\mathbf{A}\mathbf{Z}^{-1}$ maintains the same eigenvectors of $\mathbf{A}$ but has the k smallest eigenvalues mapped to θ^*, eliminating their influence in hindering convergence of the Lanczos method. Ilić et al. [15] defines the relationship between $f(\mathbf{A})\mathbf{b}$ and $f(\mathbf{A}\mathbf{Z}^{-1})\mathbf{b}$

$$f(\mathbf{A})\mathbf{b} = \mathbf{Q}_k f(\mathbf{\Lambda}_k)\mathbf{Q}_k^T\mathbf{b} + f(\mathbf{A}\mathbf{Z}^{-1})\hat{\mathbf{b}}. \tag{17}$$

where $\hat{\mathbf{b}} = (\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^T)\mathbf{b}$ and

$$\mathbf{A}\mathbf{Z}^{-1}\mathbf{V}_m = \mathbf{V}_m\mathbf{T}_m + \beta_m\mathbf{v}_{m+1}\mathbf{e}_m^T, \tag{18}$$

where $\mathbf{v}_1 = \hat{\mathbf{b}}/||\hat{\mathbf{b}}||$ and the columns of $\mathbf{V}_m$ form an orthonormal basis for the Krylov subspace $\mathcal{K}_m(\mathbf{A}\mathbf{Z}^{-1}, \mathbf{b}) = \text{span}\{\hat{\mathbf{b}}, \mathbf{A}\mathbf{Z}^{-1}\hat{\mathbf{b}}, \ldots, (\mathbf{A}\mathbf{Z}^{-1})^m\hat{\mathbf{b}}\}$. The Lanczos approximation (14) gives

$$f(\mathbf{A}\mathbf{Z}^{-1})\hat{\mathbf{b}} \approx \mathbf{V}_m f(\mathbf{T}_m)\mathbf{V}_m^T\hat{\mathbf{b}} \tag{19}$$

where the much smaller matrix function $f(\mathbf{T}_m)$ can be easily calculated by finding the diagonalisation of $\mathbf{T}_m$.

3.2 Contour Integral Method

It is known that for an analytic function f of a square matrix $\mathbf{A}$ then $f(\mathbf{A})$ can be represented as a contour integral,

$$f(\mathbf{A}) = \frac{1}{2\pi i}\int_\Gamma f(z)(z\mathbf{I} - \mathbf{A})^{-1}\,\mathrm{d}z. \tag{20}$$

where Γ is a closed contour lying in the region of analyticity of f and winding once around the spectrum $\sigma(\mathbf{A})$. Hale et al [12] found a better method to obtain $f(\mathbf{A})$ be computing $\mathbf{A}\cdot\mathbf{A}^{-1}f(\mathbf{A})$

$$f(\mathbf{A}) = \frac{\mathbf{A}}{2\pi i}\int_\Gamma z^{-1}f(z)(z\mathbf{I} - \mathbf{A})^{-1}\,\mathrm{d}z. \tag{21}$$

The algorithms proposed by Hale et al. [12] constructs the matrix $f(\mathbf{A})$ rather than the matrix function-vector product. The methods can be reworked to efficiently determine $f(\mathbf{A})\mathbf{b}$. The MATLAB code used in this paper is provided below

```
I = speye(size(A));
uu = zeros(size(A,1),1);
m = eigs(A,1,'SM'); M = eigs(A,1,'LM');   % only for toy problems
k = ((M/m)^(1/4)-1)/((M/m)^(1/4)+1);      % Useful constant
L = -log(k)/pi;
[K,Kp] = ellipkkp(L);                     % Elliptic integrals
t = .5i*Kp - K + (.5:J)*2*K/J;            % Midpoint rule points
[u cn dn] = ellipjc(t,L);                 % Jacobi elliptic functions
w = (m*M)^(1/4)*((1/k+u)./(1/k-u));       % Quadrature nodes
dzdt = cn.*dn./(1/k-u).^2;                % Derivative wrt t
```

```
for j = 1:J
    gamma = (f(w(j)^2)/w(j))*dzdt(j);      % Node specific constant
    uu = uu + (gamma*((w(j)^2*I-A)\b));    % Solve using backslash
end
sigma = -8*K*(m*M)^(1/4)*A/(k*pi*J);       % Scaling
uj = imag(sigma*uu);
```

Code 1: Contour integral method adapted from method2.m [12]

3.3 Comparison of Methods

So far we have looked at two methods for approximating linear systems, the Lanczos method and the contour integral method (CIM). Before proceeding into applying these methods to linear and non-linear fractional diffusion equations a small comparison of the effectiveness of these two methods shall be carried out.

Our $\mathbf{A}$ matrix is of size $N \times N$ and is constructed via MATLAB's gallery$('tridiag', N)$ function. The Contour integral method (CIM) utilising MATLAB's backslash operator is compared against the Lanczos method with a deflation preconditioner. In Table 1 the Lanczos method uses the deflation preconditioner with 10 and 50 predetermined eigenvalues and eigenvectors; however, CIM is significantly more efficient, especially as the size of the system was increased.

Table 1 demonstrates that the condition number of the preconditioned $\mathbf{A}^{\alpha/2}\mathbf{Z}^{-1}$, where $\mathbf{Z}^{-1}$ is the deflation preconditioner (16), was significantly lower than the condition number of $\mathbf{A}^{\alpha/2}$. This change is reflected in a reduction in the number of basis vectors m needed by the Lanczos algorithm. For the contour integral method, 35 points were used, it was determined that for systems with the dimensions in Table 1 (the N column) that this was sufficient. The efficiency of the method was determined by the runtime. For our 1-D problem, it is demonstrated in Table 1 the Contour Integral Method is the preferred method for approximating matrix-function vector-products.

$k = 10$			CIM		Lanczos Methods		
N	$\texttt{cond}(\mathbf{A})$	$\texttt{cond}(\mathbf{AZ}^{-1})$	Time	Accuracy	m	Time	Accuracy
16	1.16e+02	1.97e+00	0.01	7.83e-14	7	0.00	1.22e-13
64	1.71e+03	1.45e+01	0.01	5.07e-12	40	0.01	5.06e-12
256	2.68e+04	2.22e+02	0.01	2.15e-10	162	0.09	2.21e-10
1024	4.26e+05	3.52e+03	0.02	1.26e-08	664	1.86	1.28e-08
4096	6.80e+06	5.62e+04	0.04	8.29e-07	2728	137.14	8.30e-07
$k = 50$			CIM		Lanczos Methods		
N	$\texttt{cond}(\mathbf{A})$	$\texttt{cond}(\mathbf{AZ}^{-1})$	Time	Accuracy	m	Time	Accuracy
64	1.71e+03	2.00e+00	0.01	5.07e-12	8	0.00	5.56e-12
256	2.68e+04	1.06e+01	0.01	2.12e-10	43	0.02	2.19e-10
1024	4.26e+05	1.64e+02	0.02	1.26e-08	181	0.18	1.29e-08
4096	6.80e+06	2.62e+03	0.04	8.28e-07	755	9.84	8.35e-07

Table 1: Comparison of CIM and Lanczos algorithm for computing $\mathbf{A}^{\alpha/2}\mathbf{b}$ where $\mathbf{b}$ is the discretisation of $x^2(1-x)\exp(x)$. Deflation preconditioner of $k = 10$ and $k = 50$ was applied the Lanczos method.

4 Numerical Results

To test the effectiveness of our banded preconditioner we use a space-fractional diffusion equation with a logistic growth source term on the finite domain $0 < x < 1$ with homogeneous Dirichlet boundary conditions.

$$\frac{\partial u}{\partial t} = -\kappa(-\nabla^2)^{\alpha/2}u + u(1-u), \quad u(x,0) = x(1-x), \quad \kappa = 0.1, \quad \alpha = 1.5 \tag{22}$$

In Table 2 it is demonstrated that the banded preconditioner is effective at efficiently reducing the runtime and RHS function evaluations. With $N = 16384$ and 285 $\mathbf{F}$ evaluations and a minimum runtime of 37.4s with a bandwidth of 201. For $N = 65536$ has a minimum runtime of 259.7s with a bandwidth of 401.

In Table 2, N refers to the number of nodes used in our spatial discretisation. Bandwidth is the bandwidth of our preconditioner used. Time is measured in seconds and $\mathbf{F}$ evals refers to the number of evaluations of the right hand side function $\mathbf{F}$ from Equation (2), this is significant due to requiring the approximation of $\mathbf{A}^{\alpha/2}\mathbf{v}$ that was covered in Section 5.

	$N = 16384$		$N = 65536$	
bandwidth	Time	$\mathbf{F}$ evals	Time	$\mathbf{F}$ evals
–	5098.7	47059	99453	197907
41	65.1	592	1478	2799
101	38.6	333	574.2	1067
201	37.4	285	440.5	795
401	64.5	356	259.7	349
801	149.3	316	404.8	276

Table 2: Time taken and number of evaluations of $\mathbf{F}$ to generate approximate solutions to Equation (22) from $t = 0 \ldots 5$ with $\kappa = 0.1$ and $\alpha = 1.5$ using the BDF-IVP solver CVODE. Different bandwidths of the banded preconditioner are used with the Contour Integral method used to evaluate $f(\mathbf{A})\mathbf{b}$.

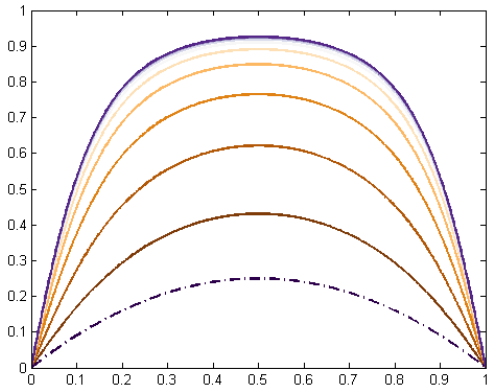

Figure 3: Evolution of equation (22) solved from $t = 0$ (dashed line) up to $t = 5$ with $\alpha = 1.5$ and $\kappa = 0.1$.

5 Conclusions

In this report we have developed a banded preconditioner for Krylov subspace methods with applications in space-fractional diffusion equations in one-dimension. The preconditioner exploits the close relation between the fractional Laplacian and the Riesz fractional derivative to form a banded matrix $\mathbf{B}$ that captured the significant amount of physical processes within the discretised model. By acknowledging that the most significant physical processes occurred around the diagonal of the discretised fractional Laplacian $\mathbf{A}^{\alpha/2}$ we could maintain a relatively small bandwidth in $\mathbf{B}$ while still capturing the most significant physical processes. We discretised our nonlinear space-fractional diffusion equation using a finite difference method. We introduced the Backwards Differentiation Methods as a means to handle the temporal discretisation as well as the class of Jacobian-free Newton Krylov methods due to the nonlinearity in our problem and the difficulty in explicitly solving the dense Jacobian. For evaluating the matrix-function vector-product $\mathbf{A}^{\alpha/2}\mathbf{u}$ we compared the Lanczos method from the class of Krylov subspace methods as well as the Contour Integral method, numerical tests demonstrated for our purposes, the Contour Integral Method was more computationally efficient.

Numerical experiments confirmed that our preconditioner significantly accelerates the convergence on GMRES methods while still being able to be easily factorisable and applied to vectors. Without our preconditioner, experiments determined that it was infeasible to determine numerical solutions on a reasonable timescale. Our

preconditioner also has the appealing characteristic that it is easily implemented into modern high-order IVP-BDF software such as CVODE from the SUNDIALS suite. It is also easily applicable to nonlinear cases where previous work is restricted to simple linear cases.

References

[1] K. Burrage A. Bueno-Orovio D. Kay. "Fourier spectral methods for fractional-in-space reaction-diffusion equations". In: *Journal of Computational Physics* (Submitted) (2012).

[2] J. Baglama et al. "Adaptively Preconditioned GMRES Algorithms". In: *SIAM J. Sci. Comput* 20 (1998), pp. 243–269.

[3] David A Benson, Stephen W Wheatcraft, and Mark M Meerschaert. "Application of a fractional advection-dispersion equation". In: *Water Resources Research* 36.6 (2000), pp. 1403–1412.

[4] Kevin Burrage, Nicholas Hale, and David Kay. "An Efficient Implicit FEM Scheme for Fractional-in-Space Reaction-Diffusion Equations". In: *SIAM Journal on Scientific Computing* 34.4 (2012), A2145–A2172.

[5] M. M. Meerschaert C. Tadjeran. "A second-order accurate numerical method for the two-dimensional fractional diffusion equation". In: *J. Comput. Phys.* 220 (2) (2007), pp. 813–823.

[6] Minghua Chen and Weihua Deng. "A second-order numerical method for two-dimensional two-sided space fractional convection diffusion equation". In: *arXiv preprint arXiv:1304.3788* (2013).

[7] S. Wheatcraft D. Benson and M. Meerschaert. "Application of a fractional advection-dispersion equation". In: *Water Resources Research* 36.6 (2000).

[8] Vladimir Druskin. "Krylov Subspace Approximation of Eigenpairs and Matrix Functions in Exact and Computer Arithmetic". In: *Numerical Linear Algebra with Applications* 2.3 (1995), pp. 205–217.

[9] M. Eiermann and O. G. Ernst. "A restarted Krylov subspace method for the evaluation of matrix functions". In: *SIAM J. Numer. Anal.* 44 (2006), pp. 2481–2504.

[10] J. van den Eshof et al. "Numerical methods for the QCD overlap operator. I. Sign-function and error bounds". In: *Comput. Phys. Commun.* 146 (2002), pp. 203–224.

[11] C William Gear. *Numerical initial value problems in ordinary differential equations*. Prentice Hall PTR, 1971.

[12] Nicholas Hale, Nicholas J. Higham, and Lloyd N. Trefethen. "Computing $A^\alpha, \log(A)$, and Related Matrix Functions by Contour Integrals". In: *SIAM Journal on Numerical Analysis* 46.5 (2008), pp. 2505–2523.

[13] Alan C Hindmarsh et al. "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers". In: *ACM Transactions on Mathematical Software (TOMS)* 31.3 (2005), pp. 363–396.

[14] M. Hochbruck and C. Lubich. "On Krylov subspace approximations to the matrix exponential operator". In: *SIAM J. Numer. Anal.* 34 (1997), pp. 1911–1925.

[15] M. Ilić, I. W. Turner, and V. Anh. "A Numerical Solution Using an Adaptively Preconditioned Lanczos Method for a Class of Linear Systems Related with the Fractional Poisson Equation". In: *Journal of Applied Mathematics and Stochastic Analysis* 2008 (2008), pp. 1–26.

[16] M. Ilić et al. "Analytical and numerical solutions of a one-dimensional fractional-in-space diffusion equation in a composite medium". In: *Applied Mathematics and Computation* 216.8 (2010), pp. 2248–2262.

[17] Milos Ilić et al. "Numerical Approximation of a Fractional-In-Space Diffusion Equation, I". In: *Fractional Calculus and Applied Analysis* 8.3 (2005), pp. 323–341.

[18] Milos Ilić et al. "Numerical approximation of a fractional-in-space diffusion equation (II) - with nonhomogeneous boundary conditions". In: *Fractional Calculus and Applied Analysis* 9.4 (2006), pp. 333–349.

[19] K. Burrage J. Erhel and B. Pohl. "Restarted GMRES preconditioned by deflation". In: *J. Comput. Appl. Math.* 69 (1996).

[20] D.A. Knoll and D.E. Keyes. "Jacobian-free Newton-Krylov methods: a survey of approaches and applications". In: *Journal of Computational Physics* 193.2 (2004), pp. 357–397.

[21] Fawang Liu et al. "Numerical methods for solving the multi-term time-fractional wave-diffusion equation". In: *Fractional Calculus and Applied Analysis* 16.1 (2013), pp. 9–25.

[22] L. Lopez and V. Simoncini. “Analysis of projection methods for rational function approximation to the matrix exponential”. In: *SIAM J. Numer. Anal.* 44 (2006), pp. 613–635.

[23] C. Tadjeran M. Meerschaert H.-P. Scheffer. “Finite difference methods for two-dimensional fractional dispersion equations”. In: *Journal of Computational Physics* 211 (1) (2006), pp. 249–261.

[24] Richard L Magin. *Fractional calculus in bioengineering*. Begell House Redding, 2006.

[25] M. Meerschaert and E. Scalas. “Coupled continuous time random walks in finance”. In: *Physica A: Statistical Mechanics and its Applications* 370 (2006), pp. 114–118.

[26] Mark M Meerschaert and Charles Tadjeran. “Finite difference approximations for fractional advection–dispersion flow equations”. In: *Journal of Computational and Applied Mathematics* 172.1 (2004), pp. 65–77.

[27] Timothy Moroney and Qianqian Yang. “A banded preconditioner for the two-sided, nonlinear space-fractional diffusion equation”. In: *Computers & Mathematics with Applications* 66.5 (2013), pp. 659–667.

[28] Timothy Moroney and Qianqian Yang. “Efficient solution of two-sided nonlinear space-fractional diffusion equations using fast Poisson preconditioners”. In: *Journal of Computational Physics* 246 (2013), pp. 304–317.

[29] I. Turner Q. Liu F. Liu and V. Anh. “Numerical simulation for the 3D seepage flow with fractional derivatives in porous media”. In: *IMA Journal of Applied Mathematics* 74 (2008).

[30] P. Zhuang F. Liu Y. Nie Q. Liu Y. T. Gu. “An implicit RBF meshless approach for time fractional diffusion equations”. In: *Computational Mechanics* 48 (2011), pp. 1 –12.

[31] K.Burrage I. Turner F. Liu Q. Yang T. Moroney. “Novel numerical methods for time-space fractional diffusion equations in two dimensions”. In: *ANZIAM J.* 52 (2011), pp. C395 –C409.

[32] Yousef Saad. “Analysis of some Krylov subspace approximations to hte matrix exponential operator.” In: *SIAM J. Sci. Comput* (1992).

[33] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.

[34] R. B. Sidje. “Expokit: A software package for computing matrix exponentials”. In: *ACM Trans. Math. Softw.* 24 (1998), pp. 130–156.

[35] Wenchang Tan et al. “An anomalous subdiffusion model for calcium spark in cardiac myocytes”. In: *Applied Physics Letters* 91.18 (2007), p. 183901.

[36] I. Turner, M. Ilic, and P. Perré. “Modelling Non-Fickian Behavior in the Cell Walls of Wood Using a Fractional-In-Space Diffusion Equation”. In: *Drying Technology* 29.16 (2011), pp. 1932–1940.

[37] J. P. Roop V. J. Ervin. “Variational formulation for the stationary fractional advection dispersion equation”. In: *Numerical Methods for Partial Differential Equations* 22 (2006), pp. 558–576.

[38] 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”. In: *Journal of Computational and Applied Mathematics* 18.2 (1987), pp. 249 –263. ISSN: 0377-0427.

[39] C. Xu X. Li. “A Space-Time Spectral Method for the Time Fractional Diusion Equation”. In: *SIAM Journal on Numerical Analysis* 47 (2009), pp. 2108 –2131.

[40] L. K. Deeks M. I. Stutter A. G. Bengough I. M. Young X. Zhang J. W. Crawford. “A mass balance based numerical method for the fractional advection-dispersion equation: Theory and application”. In: *Water Resources Research* 41 (2005).

[41] Q. Liu Y. T. Gu P. Zhuang. “An Advanced Meshless Method for Time Fractional Diffusion Equations”. In: *International Journal of Computational Methods* 8 (2011), pp. 653 –665.

[42] Q. Yang, F. Liu, and I. Turner. “Numerical methods for fractional partial differential equations with Riesz space fractional derivatives”. In: *Applied Mathematical Modelling* 34.1 (2010), pp. 200–218.

[43] Qianqian Yang et al. “A preconditioned Lanczos method for space-fractional reaction-diffusion equations on two-dimensional unstructured meshes”. In: *4th International Conference on Computational Methods (ICCM2012)*. Ed. by Yuan-Tong Gu. Crowne Plaza, Gold Coast, QLD, 2012.

[44] Qianqian Yang et al. “Novel Numerical Methods for Solving the Time-Space Fractional Diffusion Equation in Two Dimensions”. In: *SIAM Journal on Scientific Computing* 33.3 (2011), pp. 1159–1180.

Postal Address: 111 Barry Street
c/- The University of Melbourne
Victoria 3010 Australia

Email: enquiries@amsi.org.au
Phone: +61 3 8344 1777
Fax: +61 3 9349 4106
Web: www.amsi.org.au

[45] Chen Yin and Xicheng Li. “Anomalous diffusion of drug release from a slab matrix: Fractional diffusion models”. In: *International journal of pharmaceutics* 418.1 (2011), pp. 78–87.