Report no. 11/08

An efficient implementation of an implicit FEM scheme for fractional-in-space reaction-diffusion equations

Kevin Burrage[a] Nicholas Hale[b] David Kay[c]

[a]Computational Biology Group, Wolfson Building, Parks Road, Oxford OX1 3QD, UK, and the School of Mathematical Sciences, Queensland University of Technology, Brisbane, Australia (kevin.burrage@cs.ox.ac.uk, http://www.cs.ox.ac.uk/people/kevin.burrage/).

[b]Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, Oxford, OX1 3LB, UK (hale@maths.ox.ac.uk, http://people.maths.ox.ac.uk/hale/).

[c]Corresponding Author, Computational Biology Group, Wolfson Building, Parks Road, Oxford OX1 3QD, UK. (dkay@cs.ox.ac.uk, http://www.cs.ox.ac.uk/people/david.kay/).

Fractional differential equations are becoming increasingly used as a modelling tool for processes with anomalous diffusion or spatial heterogeneity. However, the presence of a fractional differential operator causes memory (time fractional) or nonlocality (space fractional) issues, which impose a number of computational constraints. In this paper we develop efficient, scalable techniques for solving fractional-in-space reaction diffusion equations using the finite element method on both structured and unstructured grids, and robust techniques for computing the fractional power of a matrix times a vector. Our approach is show-cased by solving the fractional Fisher and fractional Allen–Cahn reaction-diffusion equations in two and three spatial dimensions, and analysing the speed of the travelling wave and size of the interface in terms of the fractional power of the underlying Laplacian operator.

Oxford University Mathematical Institute
Numerical Analysis Group
24-29 St Giles'
Oxford, England OX1 3LB

August, 2011

1 Introduction

Fractional models, in which a standard time or space differential operator is replaced by a corresponding fractional differential operator, have a long history in physics, finance and hydrology, with such models being use to represent so-called anomalous diffusion. In the area of water resources, fractional models have been used to describe chemical and contaminant transport in heterogeneous aquifers [2, 10, 20, 47, 51]. In finance they have been used because of the relationship with certain option pricing mechanisms and heavy-tailed stochastic processes [58]. More recently, fractional models of the Bloch–Torrey equation for describing anomalous diffusion have been considered in the context of magnetic resonance [46], and fractional-in-space diffusion models used to describe microscale diffusion in the cell wall lining of plants [63]. In cell biology anomalous diffusion has been measured in fluorescence photobleaching recovery [57] and neural cell adhesion [62], and fractional-in-time models have been developed for simple types of chemical reaction-diffusion equations [70]. Liu and Burrage [43] have considered the question of parameter estimation for time fractional models of the dynamics of a genetic regulatory feedback component of Bacillus Subtilis, a bacterium found in soil.

In terms of the numerical solution of fractional differential equations (FDEs) a number of approaches have been considered for both the time fractional and space fractional forms. In the time fractional setting, a common approach is to approximate the fractional-in-time differential operator by a finite difference scheme that has memory all the way back to the initial condition. Such an approximation can result in explicit [12,69] or implicit schemes [39]. Fourier series approximations have also been considered [13]. More recently the variable order fractional operator problem has been invstigated [41,72].

In the space fractional setting, Liu et al. [42] have considered the numerical approximation of the fractional Fokker–Planck equation. Meerchaert and Tadjern [49] have used finite difference approximations and Turner and coauthors have investigated various stability and convergence issues based on finite difference approximations [32,45,68]. Liu and Turner have also have considered a finite volume approach [44]. However, many of these approaches have limitations in terms of computational efficiency when two or three spatial dimensions are considered. For example [53] has considered a finite element approach but computes directly the fractional power of the discrete Laplacian and this approach does not scale well. The aim of this paper is to obtain a robust, efficient approach that can be equally applicable to fractional-in-space problems in two or three spatial dimensions using a finite element (FEM) approach. The computational heart of this approach is the efficient computation of the fractional power of a matrix times a vector. Yang et al. [67] have considered the numerical solution of time-space fractional diffusion equations in two spatial dimensions with homogeneous Dirichlet boundary conditions using the matrix transfer technique. Their approach is to compute the function of a matrix times a vector either by a preconditioned Lanczos or M-Lanczos technique, depending on whether the matrix is symmetric or not. In our paper we will go into much greater depth in comparing and contrasting scalable techniques for computing $f(\mathbf{A})b$.

We will consider three techniques: the contour integral method (CIM), Extended Krylov subspace methods (ESKM), and the pre-assigned poles and interpolation nodes method (PAIN). We will see that in the first and last cases we can find preconditioners that will allow almost mesh independent convergence.

We will show-case our approach by solving fractional-in-space reaction-diffusion problems of the form

$$\frac{\partial u}{\partial t} + K(-\Delta)^{\alpha} u = g(u), \quad \underline{x} \in \Omega = [0,1]^d, \tag{1.1}$$

using the FEM and a semi-implicit Euler scheme for various $\alpha \in (1/2, 1]$. Standard equations and models that fall into this class include the fractional Allen–Cahn and Fisher equations, as well as, of course, the fractional heat equation when $g = 0$. We have chosen this class of problems as recent work [21] has established how the speed of the travelling wave and the thickness of the interface changes when $\alpha \neq 1$ — see Section 6. Our approach is intended to be robust and scalable and equally applicable for structured or unstructured grids.

The outline of the paper is as follows. In Section 2 we review some important issues associated with modelling via fractional differential equations. In Section 3 we review the matrix transfer technique that allows us to approximate the fractional Laplacian by the fractional power of a matrix and give a review of relevant computational techniques for computing appropriate matrix functions $f(\mathbf{A})b$. These include the Contour integral, Extended Krylov, and PAIN methods. In Section 4 we investigate the Neumann formulations of (1.1), which give rise to different numerical issues than Dirichlet formulations as the discrete Laplacian is now singular. Section 5 outlines our approach in terms of iterative methods and preconditioning for the CIM and PAIN methods in terms of mesh independent convergence, while Section 6 gives detailed numerical results and analysis of the fractional Fisher and fractional Allen–Cahn models, with a particular focus on the relationship between α and the speed of the travelling wave and the size of the interface. Section 7 gives some conclusions and thoughts for future work.

2 Background to Fractional Modelling

In order to understand why fractional models are useful and how they might arise, consider the classical advection-diffusion equation which can be derived from two different equations. Let $u(x,t)$ be the particle mass density of some species of particles diffusing and let ν be the constant average velocity of those particles. Fick's Law then states that the flux V is the rate at which mass is transported through a unit area and is given by

$$V = \nu u - D\nabla u, \tag{2.1}$$

where D is the diffusion tensor. Conservation of mass requires

$$\frac{\partial u}{\partial t} = -\nabla \cdot V, \tag{2.2}$$

and combining (1) and (2) gives the classical advection-diffusion equation:

$$\frac{\partial u}{\partial t} = -\nu u + \nabla \cdot D\nabla u. \tag{2.3}$$

However, both of the assumptions that underlie this approach can be questioned for heterogeneous media [65]. The continuum hypothesis uses local averaged values, but averaged quantities fluctuate wildly as the averaging volume becomes smaller. This is also related to the Homogenisation principle which is a fundamental principle for predicting macroscopic properties from microscopic features [31], which are often assumed to be independent. But in some settings this independence between the microscopic and the macroscopic may not hold and the Homogenisation principle can fail. It is in this setting that fractional models can offer insights that traditional approaches do not offer. This is especially the case of diffusion in heterogeneous environments.

In heterogenous structures, such as those possessing spatial connectivity, movement of particles may be facilitated within a certain scale – so called super-diffusion. Thus the spatial complexity of the environment imposes geometric constraints on the transport processes on all length scales, which can be interpreted as temporal correlations on all time scales. Non-homogeneities of the medium may fundamentally alter the laws of Markov diffusion, leading to long range fluxes, and non-Gaussian, heavy tailed profiles [5, 7, 50], and these motions may no longer obey Fick's Law [71].

Schumer at al. [59] demonstrated that a fractional Fick's Law is a governing equation for solute transport in porous media where temporally correlated velocity fields do not dominate transport processes. A fractional Fick's Law naturally implies spatial and temporal nonlocality and can be derived from rigorous approaches using spatial averaging theorems and the concept of a measurable function [11]. It is possible to assign certain quantities via integrals of these measurable functions and then approximate them by differentiable fluctuations and then apply a Divergence Theorem. This approach is consistent with the use of fractional derivatives.

We can relate these fractional derivatives to the underlying behaviour of an ensemble of particles undergoing a continuous time random walk [5]. When a fractional derivative replaces the Laplacian operator in the diffusion equation, the resulting solution describes the probability density function of particles undertaking a heavy tailed random walk, where occasional large jumps dominate the more common smaller jumps. On the other hand, a fractional time derivative leads to subdiffusion where the mean square displacement of an ensemble of particles behaves as $t^\alpha, \alpha \in (0, 1]$. This equates to a random walk where the waiting time between particle jumps has a long tailed probability.

Meerschaert et al. [48] define a fractional Fick's Law as

$$V = \nu u - D\nabla^\beta u, \tag{2.4}$$

where $\beta = 2\alpha - 1$ and $\alpha \in (1/2, 1]$. Combined with (1.2) this gives

$$\frac{\partial u}{\partial t} = -\nu u + \nabla \cdot D\nabla^{\beta} u. \tag{2.5}$$

In three spatial dimensions

$$\nabla^{\beta} = \left(\frac{\partial_1^{\beta}}{\partial x_1^{\beta}}, \frac{\partial_2^{\beta}}{\partial x_2^{\beta}}, \frac{\partial_3^{\beta}}{\partial x_3^{\beta}} \right)^T \tag{2.6}$$

and this is known as the Riemann–Louiville fractional gradient vector. The components of this vector are fractional derivatives, where, for $\beta \in (0, 1)$,

$$\frac{\partial f^{\beta}}{\partial x^{\beta}} = \frac{1}{\Gamma(1-\beta)} \frac{\partial}{\partial x} \int_0^x \frac{f(s, y, z)}{(x-s)^{\beta}} ds. \tag{2.7}$$

There has recently been substantial interest in general problems with nonlocal operators, and these represent an even more general framework than fractional formulations. Andreu et al. [3, 4] have considered $p-$Laplacian diffusion equations with Dirichlet and Neumann boundary conditions, respectively. Du and Zhou [18, 19] have shown through peridynamic nonlocal contiuum theory that nonlocal diffusion exhibits multiscale diffuuion beyond that of standard diffusion, and Gilboa et al. [25] have given a nice application of nonlocal operators to image processing. Finally Gunzberger et al. [27] have developed a calculus for nonlocal operators and have defined a weak formulation of nonlocal boundary value problems that mimic Dirichlet and Neumann conditions for second order elliptic partial differential equations.

The main rationale for using FDEs is their relationship to particles undergoing anomalous diffusion in a heterogeneous environment. The fractional-in-space diffusion equation

$$\frac{\partial p}{\partial t} = -K(-\Delta)^{\alpha} p, \tag{2.8}$$

can be viewed as describing the probability density function of particles undergoing superdiffusion where $(-\Delta)^{\alpha}$ is the fractional Laplacian operator. Thus it makes sense that this equation should play the fundamental role in using fractional models.

However, another interpretation is to work with the fractional gradient vector ∇^{β} as defined in (1.6) and (1.7). Then the natural modelling approach is via a fractional Fick's Law of the form $\nabla \cdot (D\nabla^{\beta})$. However, there is no natural relationship between $\nabla \cdot (\nabla^{\beta})$ and $-(-\Delta)^{\alpha}$ as there is in the pure diffusion case when $\beta = \alpha = 1$. There is a relationship between the fractional Laplacian and the Riesz derivative but only on unbounded domains. In three dimensions this takes the form

$$-(-\Delta)^{\alpha} = \frac{\partial^{\alpha}}{\partial |x|^{\alpha}} + \frac{\partial^{\alpha}}{\partial |y|^{\alpha}} + \frac{\partial^{\alpha}}{\partial |z|^{\alpha}}. \tag{2.9}$$

This can be extended to finite domains by assuming that the function is zero outside of the domain, but even then this interpretation only holds in certain settings [66].

However, Turner at al. [63] have recently argued a different approach. They go back to the underlying physical processes and use measure theory and measurable functions and work with the concept of a potential. They define a potential

$$\bar{u} = (-\Delta)^{\alpha-1} u \tag{2.10}$$

and note that since $\alpha - 1 < 0$, the expression for the potential can be interpreted as an integral. Thus the fractional gradient ∇^α can be interpreted as

$$\nabla^\alpha = \nabla(-\Delta)^{\alpha-1} \tag{2.11}$$

so

$$\nabla^\alpha u = \nabla \bar{u}. \tag{2.12}$$

When substituted into (1.5) this gives

$$\frac{\partial u}{\partial t} = -\nu u + \nabla \cdot D \nabla \bar{u} \tag{2.13}$$

or

$$\frac{\partial u}{\partial t} = -\nu u + \nabla \cdot D \nabla^\alpha u. \tag{2.14}$$

In the case that $D = KI$ this becomes from (2.4)

$$\frac{\partial u}{\partial t} = -\nu u - K(-\Delta)^\alpha u. \tag{2.15}$$

Thus in this setting there is a natural relationship between a fractional Fick's Law and a fractional Laplacian. However, we have to be careful with boundary conditions when dealing with problems of the form (2.15). Guann and Ma [26] note that a $(-\Delta)^\alpha$-harmonic function on a finite domain cannot be determined by its value on the boundary of that domain, but on the whole space minus the domain. However, Ilić et al. [32] have shown that problems of the form (2.15) with $\nu = 0$ are well defined on finite domains with homogeneous boundary conditions of Dirichlet, Neumann and Robin type. They show this by noting that $(-\Delta)^\alpha$ has the same interpretation as $-\Delta$ in terms of its spectral decomposition for these homogeneous boundary conditions. In this case the fractional Laplacian can be dealt with by using the matrix transfer technique (MTT) that was introduced in [33, 34]. In essence, the spatial discretisation of the fractional Laplacian is obtained by first finding a matrix representation of the Laplacian (whether it is by finite difference, finite element, or finite volume and which we will denote by $\mathbf{A}$) and then raising it to the same fractional power $\mathbf{A}^\alpha$.

The case of non-homogeneous boundary conditions is more complicated, and there is much confusion in the literature. Ilić et al. [33] decompose the operator as

$$(-\Delta)^\alpha = (-\Delta)^{\alpha-1}(-\Delta),$$

where te right-most operator is used to deal with the non-homogeneous boundary conditions, for which it is well defined. They then use the MTT to solve an ordinary differential equation which requires both the computation of $\mathbf{A}^{\alpha}$ and $\mathbf{A}^{\alpha-1}$ operating on different vectors. Recently Gunzberger at al. [27] have considered a nonlocal second order scalar elliptic boundary value problem

$$L(u)(x) := 2\int_{\Omega}(u(y)-u(x))v(x,y)dy = b(x) \tag{2.16}$$

augmented with nonlocal Dirichlet or Neumann boundary conditions. Here $v(x,y)$ is a symmetric function, and the boundary value problem (2.16) characterises a solution of the minimisation of

$$\int_{\Omega}\int_{\Omega}(u(y)-u(x))^2 v(x,y)dydx - \int_{\Omega} b(x)u(x)dx$$

augmented with nonlocal boundary conditions on nonzero volumes. Under appropriate conditions this reduces to the standard setting.

In this paper we will adopt a finite element (FEM) approach for solving (1.8) as such problems often arise in heterogeneous settings and FEMs are appropriate for unstructured meshes.

3 Modelling and Computational Methods

Let $\mathbf{L}$ be the standard FEM stiffness matrix with appropriate boundary conditions, and $\mathbf{M}$ the corresponding mass matrix. Following the work of Ilić at al. [33], we employ the matrix transfer technique, which states that the error introduced by approximating the fractional Laplacian as defined in (2.7) by a fractional power of the matrix $\mathbf{A} = \mathbf{M}^{-1}\mathbf{L}$ converges at the same rate as the underlying FEM discretisation. Both $\mathbf{L}$ and $\mathbf{M}$ are real and symmetric, $\mathbf{L}$ is non-negative definite, and $\mathbf{M}$ is positive definite. Noting that $\mathbf{M}^{1/2}\mathbf{A}\mathbf{M}^{-1/2} = \mathbf{M}^{-1/2}\mathbf{L}\mathbf{M}^{-1/2}$ is real, symmetric, and similar to $\mathbf{A}$, it follows that the spectrum $\sigma(\mathbf{A})$ of $\mathbf{A}$ is real. Furthermore the non-negative definiteness of $\mathbf{L}$ implies $x^T\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{1/2}x \geq 0$, so that $\mathbf{A}$ is also non-negative definite and $\sigma(\mathbf{A}) \subset [0,\infty)$.

When $\mathbf{A} \in \mathbb{R}^{N\times N}$ is diagonalisable and a function f is defined on the spectrum $\sigma(\mathbf{A})$, the matrix function $f(\mathbf{A})$ can be defined [30, Definition 1.2] as

$$f(\mathbf{A}) = \mathbf{Q}f(\mathbf{\Lambda})\mathbf{Q}^{-1}, \tag{3.1}$$

where $\mathbf{\Lambda} = \mathtt{diag}([\lambda_1, \lambda_2, \ldots, \lambda_N])$ is a diagonal matrix of the eigenvalues $\lambda_1 \leq \lambda_2, \ldots, \leq \lambda_N$ of $\mathbf{A}$, $\mathbf{Q}$ is a matrix of corresponding eigenvectors, and $[f(\mathbf{\Lambda})]_{j,j} = f(\lambda_j)$. If f is analytic in a connected open subset $\mathcal{H}(f)$ of the complex plane such that $\sigma(\mathbf{A}) \subset \mathcal{H}(f)$, an equivalent definition [30, Definition 1.11 & Theorem 1.12] is given by

$$f(\mathbf{A}) = \frac{1}{2\pi i}\int_{\Gamma} f(z)\,(z\mathbf{I}-\mathbf{A})^{-1}\,dz, \tag{3.2}$$

where $\Gamma \subset \mathcal{H}(f)$ is a closed contour winding once around the spectrum $\sigma(\mathbf{A})$ in the anticlockwise direction.

Discretising (1.1) semi-implicitly in time with a backward Euler approximation leads, at the m^{th} time step, to linear systems of the form

$$(\mathbf{I} + K\Delta t(\mathbf{M}^{-1}\mathbf{L})^{\alpha})u_{m+1} = u_m + \Delta t g(u_m). \tag{3.3}$$

The matrix function we then require is

$$f(z) = 1/(1 + K\Delta t z^{\alpha}), \tag{3.4}$$

with $\mathcal{H}(f) \subset \mathbb{C} \setminus (-\infty, 0]$. Note that functions of the same form arise when using finite difference approximations to fractional-in-time operators [66, 68], and so the methods we discuss in the next two sections are also applicable time-and-space fractional setting . When $\mathbf{A}$ is nonsingular either definition (3.1) or (3.2) can be applied, but when the smallest eigenvalue $\lambda_1 = 0$ the hypothesis that $\sigma(\mathbf{A})$ is contained in $\mathcal{H}(f)$, and (3.2) cannot be used directly.

Higham has recently published an extensive monograph [30] on the computation of matrix functions, however the problem $f(\mathbf{M}^{-1}\mathbf{L})$ we consider is so large and dense (even, for example, when $f(x) = x$) that forming it explicitly is infeasible, specialised techniques are required. Fortunately it is not $f(\mathbf{M}^{-1}\mathbf{L})$ we need in (3.3), but rather its action on a vector, and in this section we consider recent methods which allow efficient (and scalable) computation of $f(\mathbf{M}^{-1}\mathbf{L})b$ by taking advantage of the sparsity and structure of $\mathbf{M}$ and $\mathbf{L}$.

We consider three different but related approaches. The first is a contour integral method (CIM) [29], based upon a quadrature discretisation of (3.2). The other two, the extended Krylov subspace method (EKSM) [17, 37] and the preassigned poles and interpolation nodes (PAIN) method [28], are rational Krylov subspace-based techniques. Further comparisons and far more detailed descriptions of these and other rational $f(\mathbf{A})b$ methods can be found in two recent PhD theses, [28] and [61].

In what follows we assume the matrix function f is analytic in $\mathcal{H}(f) \subset \mathbb{C} \setminus (-\infty, 0]$, that the matrix $\mathbf{A}$ has a known decomposition $\mathbf{A} = \mathbf{M}^{-1}\mathbf{L}$ where $\mathbf{M}$ and $\mathbf{L} \in \mathbb{R}^{N\times N}$ are symmetric positive definite, and that the spectrum $\sigma(\mathbf{A})$ of $\mathbf{A}$ is contained in an interval $[\lambda_1, \lambda_N] \subset (0, \infty)$ where λ_1 and λ_N are known (or can at least be computed). In Section 4 we extend the discussion to the case where $\lambda_1 = 0$. Furthermore, we suppose that both the action of $\mathbf{A}$ and shifted solves of the form $(\mathbf{A} - \sigma\mathbf{I})x = b$ can be computed efficiently, and further discussion of these linear solves is given in Section 5.

3.1 Contour Integral Method (CIM)

Until recently, Definition 3.2 was rarely exploited for numerical computation, but Hale et al. [29] propose efficient quadrature approximations for computing $f(\mathbf{A})b$ when $\sigma(\mathbf{A})$ lies on or near the positive real axis and $\mathcal{H}(f) = \mathbb{C} \setminus (-\infty, 0]$. They show that although a naïve discretisation of (3.2) leads to methods requiring a number of quadrature points that increases linearly with the condition number λ_1/λ_N, choosing the contour Γ wisely can lead to methods where the number of quadrature nodes needed to obtain a specified accuracy increases asymptotically as $\log(\lambda_1/\lambda_N)$ [29, Thm. 2.1]. Since the number of nodes equates to the number of linear solves required, this makes the CIM approximation to $f(\mathbf{A})b$ 'almost' optimal.

The contour is chosen by a conformal map from an annulus $\mathcal{A}_{1,R}$ to $\mathbb{C} \setminus \{(-\infty, 0] \cup [\lambda_1, \lambda_N]\}$ (Figure 1), and the quadrature nodes and weights given by mapping the trapezium or midpoint rule applied in the annulus using the same map[1]. This leads to an approximation of the form

$$f(\mathbf{A})b \approx r_n(\mathbf{A})b = \sum_{j=1}^{n} w_j \left(\xi_j \mathbf{I} - \mathbf{A}\right)^{-1} b, \tag{3.5}$$

where the ξ_j are complex shifts lying on Γ. Equation (3.5) can be interpreted as a rational approximation to $f(\mathbf{A})b$ given directly in partial fraction form. When $\mathbf{A} = \mathbf{M}^{-1}\mathbf{L}$ this can be rewritten as

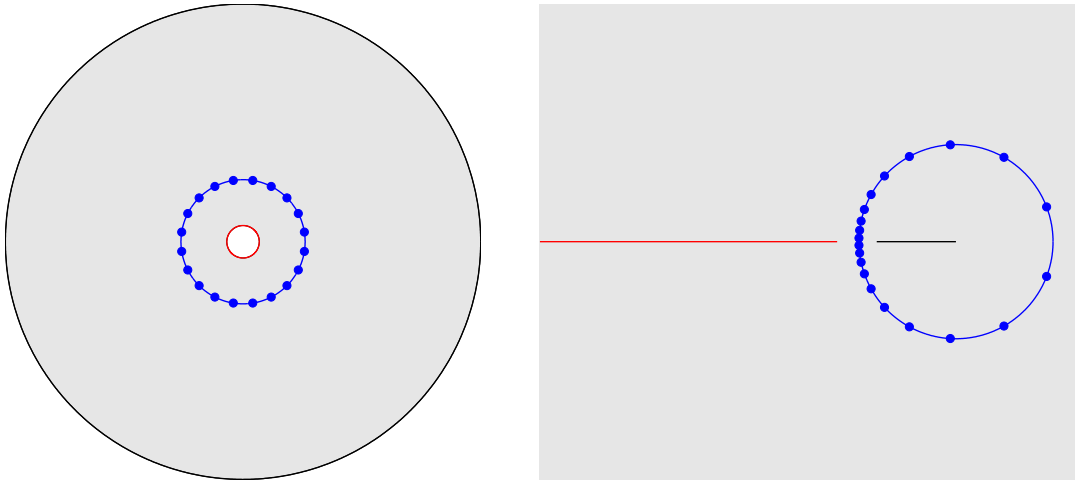
$$r_n(\mathbf{M}^{-1}\mathbf{L})b = \sum_{j=1}^{n} w_j \left(\xi_j \mathbf{M} - \mathbf{L}\right)^{-1} \mathbf{M} b, \tag{3.6}$$

and $\mathbf{A}$ itself needs never be formed explicitly. A particular advantage of the CIM is that each of the solves on the righthand side of (3.6) are independent, making this method easily parallelisable.

Furthermore, since $\mathbf{A}$ is real the integrand in (3.2) is real-symmetric and $f(\mathbf{A})$ is twice the real part of the value obtained by integrating over the first half of the contour. This means we need only half of the quadrature points (i.e., those in the upper half-plane) and therefore compute only half of the shifted solves. To simplify later discussion we will denote by the subscript of r the number of linear solves in the computation, rather than the degree of the underlying rational interpolant, and refer to the n-point CIM approximation in the same manner.

Given $\mathtt{L}, \mathtt{M} \in \mathbb{R}^{\mathtt{N}\mathtt{x}\mathtt{N}}$, $\mathtt{b} \in \mathbb{R}^{\mathtt{N}}$, a function `f`, and an integer `n`, the following MATLAB code returns `v`, the `n`-point CIM quadrature approximation to `f(M\L)b`. The routines `ellipkkp.m` and `ellipjc.m` on lines 3 and 5 are from Driscoll's Schwarz–Christoffel toolbox [16], but can be replaced by built-in MATLAB routines as described in [29]. Line 11

[1] In practice we can skip the first step of the conformal map, and begin with equally spaced points from the trapezium or midpoint rule in a rectangle as in the MATLAB code that follows.



(a) Preimage (b) Image

Figure 1: Conformal map from the annulus $\mathcal{A}_{1,R}$ (a) to the slit domain $\mathbb{C} \setminus \{(-\infty, 0] \cup [\lambda_1, \lambda_N]\}$ (b). The blue dots are the quadrature points in the CIM, i.e., the shifts used in the linear solves, that can be shown to lie on a circle of radius $\lambda_N\sqrt{1+\lambda_1/\lambda_N}$ about the point λ_N. For real matrices, we only need the quadrature points in the upper half-plane.

uses MATLAB's \ to solve the linear systems in (3.6), although more practical methods of doing this when `N` is large are discussed in Section 5. The `parfor` command on line 10 allows each of the shifted linear solves to be solved in parallel when the MATLAB parallel toolbox is available.

```
0   % Contour Integral Method, adapted from [30, method1.m]
1   l = eigs(L,M,2,'BE'); l1 = l(1); lN = l(2); % Spectrum of M\L
2   k = (sqrt(lN/l1)-1)/(sqrt(lN/l1)+1);   % A convenient constant
3   [K Kp] = ellipkkp(-log(k)/pi);         % Elliptic integrals
4   t = .5i*Kp-K+(n-.5:-1:0)*2*K/n;        % Midpoint rule points
5   [sn cn dn] = ellipjc(t,-log(k)/pi);    % Jacobi elliptic functions
6   xi = sqrt(l1*lN)*(1/k+sn)./(1/k-sn);   % Quadrature nodes
7   dxidt = cn.*dn./(1/k-sn).^2;           % Derivative wrt t
8   wts = f(xi).*dxidt;                    % Quadrature weights
9   v = zeros(length(b),1);                % Initialise output
10  parfor j = 1:n                         % Parallel solves
11      y = (xi(j)*M-L)\(M*b);             % Solve using backslash
12      v = v + wts(j)*y;                  % Update solution vector
13  end
14  v = -4*K*sqrt(l1*lN)*imag(v)/(k*pi*n); % Scaling
```

3.2 Krylov subspace methods

Krylov subspace methods are well-known for their application in solving linear systems (see, for example, [14, 15]). Although rarely considered as an $f(\mathbf{A})b$ problem, i.e., the application of a matrix function to a vector, such linear solves can certainly be viewed as the special case where $f(z) = z^{-1}$. The same subspace projection ideas can and have been extended to more general functions [14].

From a *polynomial Krylov space* $\mathcal{K}_r$ of order r

$$\mathcal{K}_r(\mathbf{A}, b) = \mathtt{span}\{b, \mathbf{A}b, \ldots, \mathbf{A}^{r-1}b\}, \tag{3.7}$$

the *Rayleigh–Ritz approximation* to $f(\mathbf{A})b$ is given by

$$f_r(\mathbf{A})b = \mathbf{V}_r f(\mathbf{A}_r)\mathbf{V}_r^* b,$$

where $\mathbf{A}_r$ is the *Rayleigh quotient* $\mathbf{A}_r = \mathbf{V}_r^*\mathbf{A}\mathbf{V}_r$, and the columns of $\mathbf{V}_r$, which form a basis of $\mathcal{K}_r(\mathbf{A}, b)$, are computed by a stabilised Gram–Schmidt process. It is typically assumed that r is not large, so that $f(\mathbf{A}_r)b$ can be computed directly by, say, an eigenvalue decomposition as in Definition 3.1 or by other methods described by Higham [30]. The Krylov spaces (3.7) are nested, i.e., $\mathcal{K}_{r+1}(\mathbf{A}, b) \subset \mathcal{K}_r(\mathbf{A}, b)$, so the basis can be updated efficiently and the approximation refined iteratively.

The Rayleigh–Ritz approach can be interpreted as a polynomial approximation

$$f_r(\mathbf{A})b = p_{r-1}(\mathbf{A})b \approx f(\mathbf{A})b,$$

where the degree $r-1$ polynomial p_{r-1} interpolates f at the Ritz values $\sigma(\mathbf{A}_r)$ [56]. A natural extension to *Rational Krylov methods* based upon rational interpolation was introduced by Ruhe [54, 55]. The motivation here is that the additional cost of needing to solve linear systems in the approximation is more than compensated for by increased rates of convergence, and the overall computational work required can be significantly reduced.

Given a polynomial q_{r-1} of degree $r-1$ with no poles in $\sigma(\mathbf{A})$, the *rational Krylov space* of order r associated with $(\mathbf{A}, b, q_{r-1})$ is given by

$$\mathcal{Q}_r(\mathbf{A}, b; q_{r-1}) = \mathcal{K}_r(\mathbf{A}, q_{r-1}(\mathbf{A})^{-1}b).$$

Again it is convenient in computations if the $\mathcal{Q}_j$ are nested, so typically a sequence of *poles* $\{\xi_1, \xi_2, \ldots, \} \subset \{\mathbb{C} \cup \{\infty\}\} \setminus \sigma(\mathbf{A})$ is defined and

$$q_r(z) = \prod_{j=1}^{r}(z - \xi_j), \quad r = 1, 2, \ldots. \tag{3.8}$$

Once the rational functions q_{r-1}, and hence the subspaces $\mathcal{Q}_r$, have been chosen, we then obtain the *rational Rayleigh–Ritz* approximation

$$f_r(\mathbf{A})b = \mathbf{V}_r f(\mathbf{A}_r)\mathbf{V}_r^* b,$$

where $\mathbf{A}_r = \mathbf{V}_r^* \mathbf{A} \mathbf{V}_r$, and the orthogonal columns of V_r form a basis of the now rational Krylov subspace $\mathcal{Q}_r(\mathbf{A}, b)$.

If all $\xi_j = \infty$ in (3.8), the rational Krylov method reduces to the polynomial Rayleigh–Ritz method described above. Choosing $\xi_{2j} = \infty, \xi_{2j+1} = 0$ gives the *Extended Krylov subspace method* (EKSM) [17, 37], discussed below.

3.2.1 Extended Krylov Subspace Method (EKSM)

The EKSM method, originally introduced by Druskin and Knizhnerman [17] and recently revived by Knizhnerman and Simoncini [37], is the special case of the rational Krylov methods described above where the poles $\{\xi_{2j+1}\}$ are all chosen to be at the origin. This simple choice of pole locations means EKSM approximations to $f(\mathbf{A})b$ are generated from the rational subspace

$$\mathcal{K}_{2r}(\mathbf{A}, \mathbf{A}^{-r} b),$$

of which an orthogonal basis $\mathbf{V}_{2r} = [V_2, V_4, \ldots, V_{2r}]$ can be constructed by applying Gram–Schmidt orthogonalisation to

$$\hat{V}_2 = [b, \mathbf{A}^{-1} b], \quad \hat{V}_{2r} = [\mathbf{A} V_{2r-2}^{(1)}, \mathbf{A}^{-1} V_{2r-2}^{(2)}], \quad r > 1.$$

Here $\mathbf{A} = \mathbf{M}^{-1}\mathbf{L}$, so the $\hat{V}$ are given by

$$\hat{V}_{2r} = [\mathbf{M}^{-1}\mathbf{L} V_{2r-2}^{(1)}, \mathbf{L}^{-1}\mathbf{M} V_{2r-2}^{(2)}],$$

and solves with both $\mathbf{M}$ and $\mathbf{L}$ are required for each dimension of the subspace (except the first). To remain consistent with the CIM method, we therefore refer to the *degree* n EKSM approximation $r_n(\mathbf{A})b$ of $f(\mathbf{A})b$ as that which requires n linear solves to compute, and so $n = 2r - 1$ in the above.

For real symmetric positive definite matrices $\mathbf{A}$ with spectrum in $[\lambda_1, \lambda_N]$, and functions analytic on $\mathbb{C} \setminus (-\infty, 0]$, it can be shown [9, 37] that convergence of the EKSM satisfies

$$||f(A)b - r_n(\mathbf{A})b|| = O(\exp(-n \sqrt[4]{\lambda_1/\lambda_N})). \tag{3.9}$$

Whilst this bound suggests that the number of linear solves required to obtain a certain accuracy will increase more rapidly here than in the CIM as the condition number rises (i.e., to the $1/4^{\text{th}}$ power, rather than logarithmically), it is possible to show that, for symmetric matrices, this is an optimal rate for a single pole location [9].

Pseudocode for computing the EKSM was given in the original paper [17], and Simoncini gives MATLAB code in [60] which is easily modified to account for $\mathbf{A} = \mathbf{M}^{-1}\mathbf{L}$ without forming $\mathbf{A}$ explicitly.

3.2.2 Preassigned poles and interpolation nodes (PAIN[2])

In addition to prescribing the poles $\{\xi_j\}$ that determine the rational Krylov subspace, we can also select the interpolation nodes $\{\alpha_j\}$. The problem then becomes even more closely related to that of best rational approximation, which is itself related to potential theory. Whilst the task of determining a best approximation is known to be a difficult one, there are a number of ways of determining asymptotically optimal points. Levin and Saff [40] give a detailed discussion of the varying approaches, and we recall some of that information here.

Fekete points, which give the supremum of

$$\prod_{\substack{j\neq k\\ 1\leq j,k\leq n}} \frac{|\xi_j-\xi_k||\alpha_j-\alpha_k|}{|\xi_j-\alpha_k|^2} \tag{3.10}$$

for $\xi_k \in (-\infty, 0], \alpha_k \in [\lambda_1, \lambda_N]$, are considered the 'cleanest' mathematically but are difficult to compute numerically and rarely used in practice. *Leja–Bagby points* are an approximation to Fekete points whereby an initial pole and node $\{\xi_1, \alpha_1\}$ are chosen arbitrarily and $\{\xi_k, \alpha_k\}, k = 2, \ldots, n$ are determined incrementally to minimise the the energy

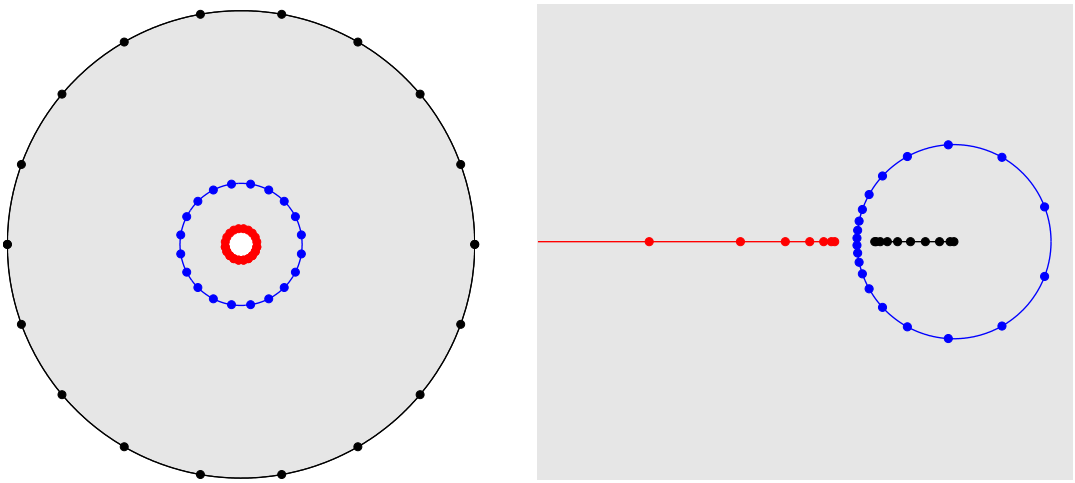
$$\prod_{j=1}^{k-1} \frac{|\xi-\xi_j||\alpha-\alpha_j|}{|\xi-\alpha_j||\alpha-\xi_j|} \tag{3.11}$$

for each k. Still these points are cumbersome to compute, and typically the allowed values for $\{\xi_k\}$ and $\{\alpha_k\}$ are restricted to discrete subsets of $(-\infty, 0]$ and $[\lambda_1, \lambda_N]$. A code for solving this discrete problem, `lejadc`, appears in the appendix.

When $\mathcal{H}(f)$ is closed, connected, and disjoint from $[\lambda_1, \lambda_N]$, the *Fejér–Walsh points* are defined by the image of equally spaced points on the boundary of an annulus $\mathcal{A}_{1,R}$ under the unique conformal map from $\mathcal{A}_{1,R}$ to $\mathcal{S} = \mathcal{H}(f) \setminus [\lambda_1, \lambda_N]$. In the problem we consider $\mathcal{S} = \mathbb{C} \setminus \{(-\infty, 0] \cup [\lambda_1, \lambda_N]\}$ and the map is precisely that given for the CIM appearing in Figure 1. We repeat the map in Figure 2, but now contains the Fejér–Walsh interpolation nodes (black dots) and poles (red dots) alongside those from the CIM (blue dots). Since this map is available, appearing in [29] and [1, §49], we follow the suggestion of Levin and Saff [40, 7.c] that these Fejér–Walsh points can be "more efficient" than the other possibilities discussed above. Some brief experiments, at least with discretely chosen Leja–Bagby points, gave similar results.

As with the polynomial Krylov approximation, the PAIN method can be thought of as the evaluation of an interpolant (now rational) in Newton form. As such, certain stability issues arise. In particular we must sensibly order the nodes (and poles) that define the denominators $q_r(z)$, and hence the subspaces $\mathcal{Q}_r$. We follow the suggestion of Reichel [52] by ordering the Fejér–Walsh points so that (3.11) is satisfied for each

[2]This acronym seems to have been coined by Güttel [28]



(a) Preimage (b) Image

Figure 2: Fejér–Walsh poles (red dots) and interpolation nodes (black dots). Importantly, these are both real.

$k = 1, \ldots, n$, which can be done using the `lejadc` code mentioned above.

The convergence properties of the PAIN method are directly related to those of the underlying rational approximant r_n on the spectrum of $\mathbf{A}$. Güttel [28] shows that, when choosing the nodes and poles of the rational approximant in any of the manners described above, we have

$$\limsup_{n\to\infty} ||f - r_n||^{1/n}_{[\lambda_1,\lambda_N]} \leq R^{-1}, \tag{3.12}$$

where R is the outer radius of the annulus $\mathcal{A}_{1,R}$ used to define the Fejér–Walsh points, appearing in Figures 1a and 2a. By the same argument as in [29, Theorem 2.1], one can show that the degree of the rational function needed to obtain a specified accuracy increases only logarithmically with the condition number λ_N/λ_1. Güttel also shows that this convergence rate is not more than a factor of two worse than the rational best uniform approximation r_n^* of type $(m-1, m-1)$ to f on $[\lambda_1, \lambda_N]$ [28, Remark 7.7]. For complex matrices, the CIM yields a similar result to (3.12), but with the R being replaced by $\sqrt{R}$. However, as described in the previous section, when the matrix $\mathbf{A}$ is real we need only use the quadrature points in the upper half-plane, which effectively replaces n with $2n$ in the exponent and recovers (3.12).

Given `L,M` $\in \mathbb{R}^{\mathtt{N}\times\mathtt{N}}$, `b` $\in \mathbb{R}^{\mathtt{N}}$, a matrix function `f`, and an integer `n`, the MATLAB code below returns `v`, the degree `n` Fejér–Walsh PAIN approximation to `f(M\L)b`. Here `f` is a matrix version of the function f suitable for evaluating the lower triangular subspace projection `Ar`, such as that returned by the MATLAB routine `funm`. The inverse action of $f(\mathbf{A})$ can be applied efficiently by uncommenting line `9`. The vectors `xi` and `alpha` are given by line `6` in the CIM code, but with line `4` replaced by `t=1i*Kp-K+(n-.5:-1:0)*2*K/n` and `t-K+(n-1:-1:0)*2*K/(n-1)` respectively. The

function `lejadc` (which can be found in the appendix) Leja-sorts the nodes and poles as described above. Again we use MATLAB's \ command to solve the linear systems in line 4, but discuss in Section 5 more practical methods for large `N`.

```
0  % PAIN method, adapted from Guettel's thesis [29]
1  v = zeros(length(b),1); K = eye(n); H = diag(alpha); % Initialise
2  [alpha,xi] = lejadc(alpha,xi);           % Leja-sort the nodes.
3  for r = 1:n
4     w = (M*xi(r)-L)\((L-alpha(r)*M)*b);   % New basis vector
5     K(r+1,r) = norm(w);                   % Update K
6     H(r+1,r) = xi(r)*norm(w);             % Update H
7     Ar = H(1:r,1:r)/K(1:r,1:r);           % Rayleigh quotient
8     fr = f(Ar);                           % Matrix function
9     % fr = fr\[1 ; zeros(r-1,1)];         % Uncomment for f(A)\b
10    v = v + fr(r,1)*b;                    % Update solution vector
11    b = w/norm(w);                        % Renormalise b
12 end
```

3.3 Comparison of CIM, EKSM, and PAIN

We end this section by testing the three $f(\mathbf{A})b$ methods discussed above on the problem

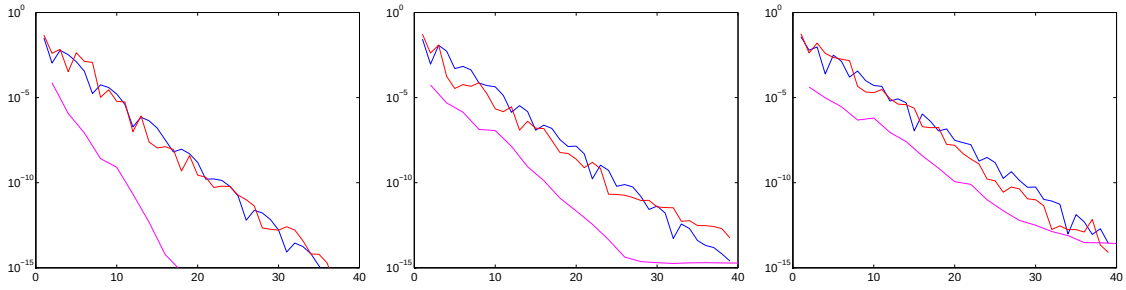
$$u - K\left(-\Delta u\right)^{\alpha} = x(x-1)y(y-1) \text{ in } \Omega = (0,1)\times(0,1), \tag{3.13}$$

with the Dirichlet boundary condition

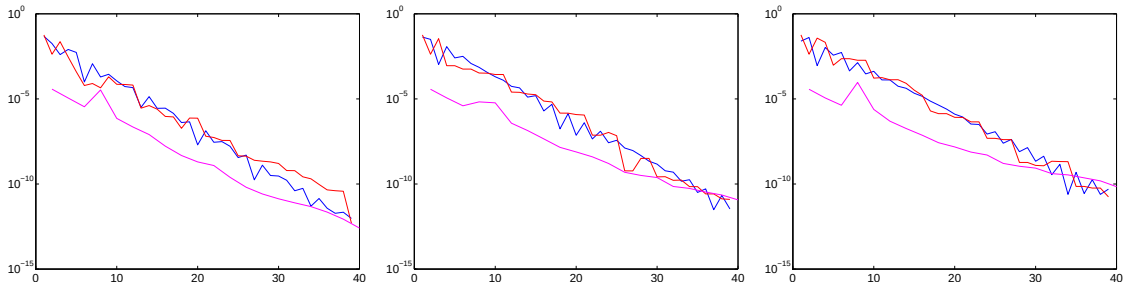
$$u(\partial\Omega) = 0. \tag{3.14}$$

We use a sequence of 6 uniformly refined triangulations of the domain Ω, and solve all linear systems using MATLAB's \ command. We take $K = 0.01$, and show in Fig. 3 results for $\alpha = 0.75$, which seem representative of the results for all $\alpha \in (0.5, 1)$. The x-axis shows n, the number of linear solves, which are assumed to be the expensive stage of the computation, and the y-axis the infinity norm of the error. For the first two levels the reference solution is computed using MATLAB's `mpower` command, and for the more refined grids where this becomes too expensive we choose the 100-point CIM quadrature approximation. Timings are given in Table 1, and all computations were performed on an AMD 2.6GHz tri-core desktop in MATLAB 7.12.

Fig. 3 shows that, as a function of the number of linear solves required, the convergence of the CIM and the PAIN method are very similar for this problem, and closely follow the bound (3.12). Although degrading more as the size of the mesh increases, as suggested by (3.9), the curves for the EKSM typically lie below those for the CIM and PAIN methods. In particular, the EKSM gives a surprisingly accurate solution even after just two solves, as it captures the contribution from the unitary eigenvalues arising from the Dirichlet conditions – a phenomenon described in [8]. Recalling that the



(a) level 1 (b) level 2 (c) level 3



(d) level 4 (e) level 5 (f) level 6

Figure 3: Error in the CIM (blue), EKSM (magenta), and PAIN (red) approximations to the matrix function $f(\mathbf{A})b$ arising from a FEM approximation to (3.13-3.14) as a function of n, the number of linear solves required, on a series of increasingly finer meshes.

level	N	cond. no.	CIM time	PAIN time	EKSM time
1	145	3,894	0.0262	0.0291	0.0193
2	545	14,488	0.1407	0.3592	0.0563
3	2,113	57,486	0.8678	0.6704	0.2848
4	8,321	230,020	4.7929	2.7610	1.7202
5	33,025	920,038	27.218	15.499	8.5777
6	131,585	3,680,178	165.01	90.953	52.112

Table 1: Timing results for solving equation (3.13) using the CIM, PAIN, and EKSM methods of degree $n = 40$ (i.e., involving 40 linear solves with backslash). The complex-valued shifts of the CIM makes it up to a factor of 2 slower than PAIN, whereas the simplicity of the solves in EKSM makes it around twice as fast.

shifts, and hence all the arithmetic, in the PAIN and EKSM methods are real, it is to be expected that the CIM has the longer computational time when n is large and the linear solves dominate the computational cost. However, it is slower only by a factor of a little less than 2, and later in the next section we solve this problem on a 3 core machine, where the timings are comparable. There are no shifts involved in the EKSM method, and this seems to make the solves faster than those in the PAIN method by around a factor of 2.

The CIM is based upon a midpoint rule quadrature discretisation, so for each n in Fig. 3 the shifts used in the linear solves can only be reused each time n is tripled[3]. Conversely, the results of the PAIN and EKSM methods are iterative, and so need only be applied once. This allows computing the interpolation points and poles corresponding to some given higher degree rational function, and iterating until some convergence criterion, such as [61, Theorem 3.7], is achieved. However, for the application we have in mind where the same matrix function will be computed for many different right-hand sides in semi-implicit time-stepping scheme, the number of quadrature points (or degree of the rational function) needed to obtain a specified accuracy can be computed offline in advance at little cost.

4 Neumann Problems

So far our discussion has been restricted to the Dirichlet problem in equation (1.1), where the matrix $\mathbf{A}$ is nonsingular. However, many applications require Neumann-type conditions, such as $\frac{\partial u}{\partial n} = 0$ on $\partial\Omega$, making the discretised matrix $\mathbf{A}$ singular. In this section we describe extensions to the methods discussed in the previous which allow practical computation of $f(\mathbf{A})b$ in such a situation.

[3]By using the trapezium rule rather than the midpoint they repeat each time n is doubled.

We begin by describing a modification of the CIM for singular $\mathbf{A}$. The contour integral Definition 3.2 of $f(\mathbf{A})$ cannot be applied here, as $(-\infty, 0]$ and $\sigma(\mathbf{A}) = [0, \lambda_N]$ are no longer disjoint and we cannot find a contour Γ in $\mathcal{H}(f)$ surrounding $\sigma(\mathbf{A})$. However, $\mathbf{A}$ is still diagonalisable and Definition (3.1) still holds. Theorem 1 below suggests how $f(\mathbf{A})b$ might be computed up to the addition of a constant vector simply by a contour integral which simply ignores the zero eigenvalue.

Theorem 1 *Let* $\mathbf{A} = \mathbf{Q}\mathtt{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)\mathbf{Q}^{-1}$ *be a diagonalisable matrix. If* f *is a function analytic in an open connected set* $\mathcal{H}(f) \subset \mathbb{C}$ *containing* $\lambda_2, \lambda_3, \ldots, \lambda_N$ *and defined at* λ_1*, then*

$$f(\mathbf{A}) = \frac{1}{2\pi i}\int_{\Gamma_2} f(z)(z\mathbf{I} - \mathbf{A})^{-1}dz + f(\lambda_1)\mathbf{Q}(1,:)\mathbf{Q}^{-1}(:,1),$$

where $\Gamma_2 \subset \mathcal{H}(f)$ *is a closed contour containing* $\lambda_2, \lambda_3, \ldots, \lambda_N$*, but not* λ_1*.*

Proof To simplify notation let $\mathbf{S} = \mathbf{Q}^{-1}$, $s^T = \mathbf{S}(:,1)$, and $q = \mathbf{Q}(:,1)$ (the eigenvector corresponding to the eigenvalue λ_1, which spans the null-space if $\lambda_1 = 0$). Definition (3.1) implies

$$\begin{aligned} f(\mathbf{A}) &= \mathbf{Q}f(\mathbf{\Lambda})\mathbf{S} \\ &= \begin{pmatrix} q & \underline{\mathbf{Q}} \end{pmatrix}\begin{pmatrix} f(\lambda_1) & \\ & f(\underline{\mathbf{\Lambda}}) \end{pmatrix}\begin{pmatrix} s^T \\ \overline{\mathbf{S}} \end{pmatrix} \\ &= f(\lambda_1)qs^T + \underline{\mathbf{Q}}f(\underline{\mathbf{\Lambda}})\overline{\mathbf{S}}, \end{aligned} \tag{4.1}$$

where $\underline{\mathbf{Q}} = \mathbf{Q}(:, 2{:}\mathtt{end}), \overline{\mathbf{S}} = \mathbf{S}(2{:}\mathtt{end}, :)$ and $\underline{\mathbf{\Lambda}} = \mathbf{\Lambda}(2{:}\mathtt{end}, 2{:}\mathtt{end})$. We can expand the inverse of $(z\mathbf{I} - \mathbf{A})$ in the same way to find

$$(z\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{z - \lambda_1}qs^T + \underline{\mathbf{Q}}(z\underline{\mathbf{I}} - \underline{\mathbf{\Lambda}})^{-1}\overline{\mathbf{S}},$$

which, multiplying by $f(z)$ and integrating around Γ_2, gives

$$\int_{\Gamma_2} f(z)(z\mathbf{I} - \mathbf{A})^{-1}dz = qs^T\int_{\Gamma_2}\frac{f(z)}{z - \lambda_1}dz + \underline{\mathbf{Q}}\int_{\Gamma_2} f(z)(z\underline{\mathbf{I}} - \underline{\mathbf{\Lambda}})^{-1}dz\overline{\mathbf{S}}. \tag{4.2}$$

The first integral on the right-hand side of (4.2) vanishes by the Cauchy integral theorem, and since the second is precisely Definition 3.2 of $f(\underline{\mathbf{\Lambda}})$ (scaled by $2\pi i$), we have

$$\frac{1}{2\pi i}\int_{\Gamma_2} f(z)(z\mathbf{I} - \mathbf{A})^{-1}dz = \underline{\mathbf{Q}}f(\underline{\mathbf{\Lambda}})\overline{\mathbf{S}}. \tag{4.3}$$

Combining (4.1) and (4.3) gives the required result. ∎

Remark 1: Theorem 1 and its proof generalise to the case when $\mathbf{A}$ is not diagonalisable but can be expressed in Jordan canonical form where λ_1 is semisimple with

multiplicity 1. Furthermore, the result can be easily extended to a contour which avoids multiple eigenvalues of $\mathbf{A}$.

Remark 2: When computing $f(\mathbf{A})b$ the correction $f(\lambda_1)qs^Tb$ is a scalar multiple of q. If $\mathbf{A}$ is symmetric then $s = q$ and the correction can be computed explicitly if q and λ_1 are known.

Remark 3: In the case of Neumann boundary conditions in our FEM discretisation, $\lambda_1 = 0$ and the null-space of $\mathbf{L}$, and hence $\mathbf{A}$, is a constant vector. Although $\mathbf{A} = \mathbf{M}^{-1}\mathbf{L}$ is not symmetric both $\mathbf{M}$ and $\mathbf{L}$ are, and so the null-space of $\mathbf{A}^T$ is spanned by $s = \mathbf{M}q/q^T\mathbf{M}q$ (where the scaling ensures $q^Ts = 1$). Therefore the correction $f(\lambda_1)\mathbf{Q}(1,:)\mathbf{Q}^T(:,1)b$ when applying Theorem 1 to a vector b is given by

$$q\frac{f(0)q^T\mathbf{M}b}{q^T\mathbf{M}q} = e\frac{f(0)e^T\mathbf{M}b}{e^T\mathbf{M}e}, \tag{4.4}$$

where e is the constant vector of ones. This correction has a further interpretation in terms of mass conservation. That is, if $f_2(\mathbf{A})b$ is defined by the integral in (4.3), then

$$f(\mathbf{A})b = f_2(\mathbf{A})b + \frac{e^T\mathbf{M}(b - f_2(\mathbf{A})b)}{e^T\mathbf{M}e}e \tag{4.5}$$

ensures the masses $e^T\mathbf{M}f(\mathbf{A})b$ and $e^T\mathbf{M}b$ are equal (as should be the case when $f(0) = 1$ as in equation (3.4)). Furthermore, since the vector e is orthogonal to each column of $\mathbf{M}\underline{\mathbf{Q}} = \underline{\mathbf{M}\mathbf{Q}} = \overline{\mathbf{S}}^T$, we find $e^T\mathbf{M}f_2(\mathbf{A})b = e^T\mathbf{M}\underline{\mathbf{Q}}f(\underline{\overline{\mathbf{\Lambda}}})\overline{\mathbf{S}}b = 0$, and so (4.5) agrees with (4.4).

The approach of simply ignoring the zero eigenvalue is related to the *spectral splitting* idea introduced by Ilić and Turner for symmetric positive definite matrices [35]. They discuss a method of decomposing $f(\mathbf{A})$ whereby the spectrum $\sigma(\mathbf{A})$ is split into a 'regular' part Λ, where f can be well approximated by a low-degree polynomial p_n, and a 'singular' part Λ^c, which is less easily approximated. This leads to an approximation of the form

$$f(\mathbf{A})b \approx \hat{\mathbf{Q}}f(\mathbf{H})\hat{\mathbf{S}}b + p_n(\mathbf{A})(\mathbf{I} - \hat{\mathbf{Q}}\hat{\mathbf{S}})b, \tag{4.6}$$

where $\hat{\mathbf{Q}}$ is an orthanormal basis for the invariant subspace corresponding to Λ^c such that $\mathbf{A}\hat{\mathbf{Q}} = \hat{\mathbf{Q}}\mathbf{H}$, and $\hat{\mathbf{S}} = \hat{\mathbf{Q}}^T$. Ilić and Turner suggest one of the drawbacks of the method is the difficulty in computing the basis $\hat{\mathbf{Q}}$, but go some way to addressing this in a subsequent paper [36]. In our case $\mathbf{A} = \mathbf{M}^{-1}\mathbf{L}$ is no longer symmetric, but we can define $\hat{\mathbf{Q}} = (q \ \texttt{zeros}(\texttt{N}, \texttt{N}-1))$ and $\hat{\mathbf{S}} = (\mathbf{M}\hat{\mathbf{Q}})^T$ with $\mathbf{H} = 0$ so that the first term in the sum (4.6) reduces to a familiar $f(0)qs^T$. Some further algebra shows that

$$\begin{aligned} p_n(\mathbf{A})\hat{\mathbf{Q}}\hat{\mathbf{S}}b &= \mathbf{Q}p_n(\mathbf{\Lambda})\mathbf{S}\hat{\mathbf{Q}}\hat{\mathbf{S}}b \\ &= \mathbf{Q}p_n(\mathbf{\Lambda})e_1s^Tb \\ &= p_n(\lambda_1)qs^Tb \end{aligned}$$

and so

$$f(\mathbf{A})b \approx p_n(\mathbf{A})b + (f(\lambda_1) - p_n(\lambda_1))\, qs^T b. \tag{4.7}$$

Although in [35] the original motivation was to let p_n be a low-degree polynomial which approximates f well on Λ, there is no need to stick with polynomials. In particular, we can replace $p_n(x)$ in (4.7) with $r_n(x;[\lambda_2,\lambda_N])$, the degree n PAIN approximation assuming the spectrum of $\mathbf{A}$ is contained in $[\lambda_2,\lambda_N]$. In this way the PAIN method can now also be applied to Neumann problems. Similarly, replacing p_n with f_2 from (4.3) (or its CIM quadrature approximation), we recover the approach suggested by Theorem 1.

For the EKSM, things are not so straightforward. Whereas we could apply the PAIN method to $\mathbf{A}$ even when it is singular (as the only solves needed have negative shifts), the EKSM builds a space containing $(\mathbf{M}^{-1}\mathbf{L})^{-k}b$ which is not well-defined if b contains any components in the null-space of $\mathbf{L}$. However, we propose the following idea, similar to those for the CIM and PAIN methods and related to the well known Sherman–Morrison formula. Let

$$\mathbf{B} = \mathbf{A} + \mu qs' = \mathbf{M}^{-1}\left(\mathbf{L} + \mu\frac{\mathbf{M}ee^T\mathbf{M}}{e^T\mathbf{M}e}\right) \tag{4.8}$$

for some $\mu > 0$, then

$$f(\mathbf{A})b = f(\mathbf{B})b + (f(0) - f(\mu))qs^T, \tag{4.9}$$

by Equation (4.1). Now $\mathbf{B}$ is nonsingular and $f(\mathbf{B})b$ can be computed by EKSM. However the rank 1 update in (4.8) means $\mathbf{B}$ is now dense, and a naïve implementation will be impractical. Noting however that $\mathbf{MB}$ is simply the stiffness matrix $\mathbf{L}$ plus a multiple of the rank 1 matrix $(\mathbf{M}e)(\mathbf{M}e)^T$, and this structure can be exploited to solve $\mathbf{B}^{-k}b$ efficiently. In particular, it can be shown that if y is a solution to

$$\mathbf{L}y = \mathbf{M}\left(c - \frac{e^T\mathbf{M}c}{e^T\mathbf{M}e}e\right),$$

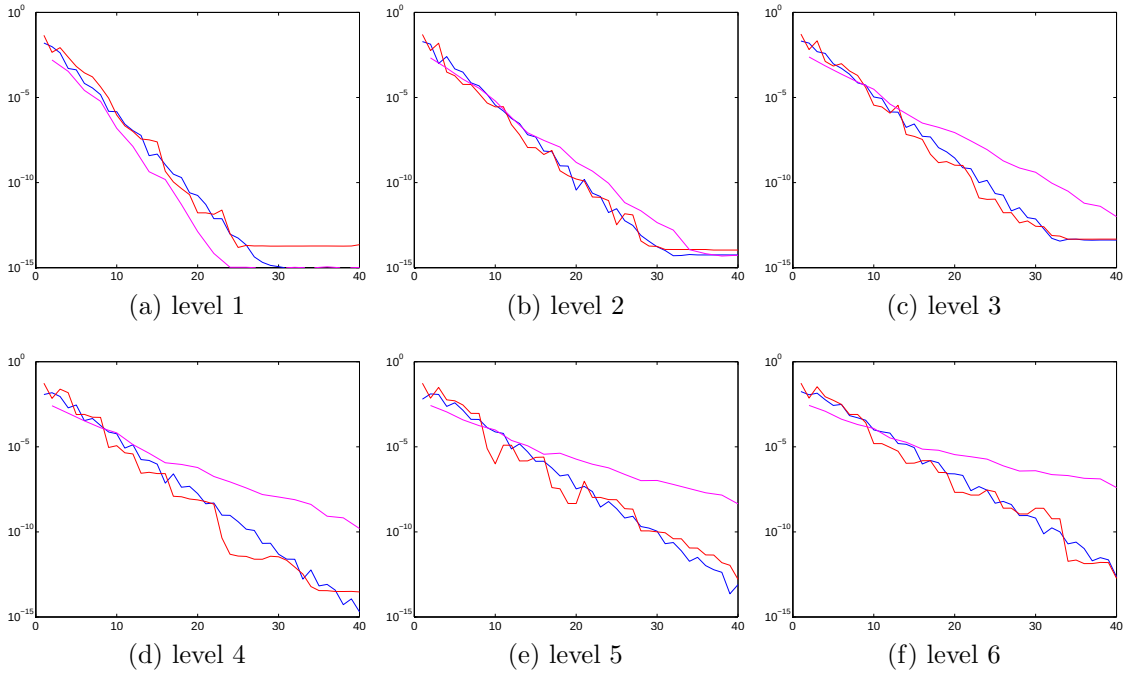
(which is well-posed as the righthand side has no component in the kernel of $\mathbf{L}$) then

$$x = y + \frac{e^T\mathbf{M}}{e^T\mathbf{M}e}\left(\frac{c}{\mu} - y\right)e$$

solves

$$\mathbf{B}x = c.$$

Fig. 4 and Table 2 show results for the same example problem (3.13) from Section 3.3, but now with Neumann boundary conditions. For CIM and PAIN this is done by applying the relevant boundary conditions to the stiffness matrix $\mathbf{L}$, defining `l = eigs(L,M,2,'SM'); l1 = l(2); lN = eigs(L,M,1,'LM');` in the codes given previously, and adjusting the output `v` subject to (4.5) for the CIM or (4.7) for the PAIN method. For EKSM, $f(\mathbf{B})b$ was computed with μ in (4.8) chosen as $\lambda_1/2$, and adjusted subject to (4.9). In solving these Neumann problems we use the MATLAB Parallel Computing Toolbox on a 3 core machine, still with \, and find in Table 2 that as well as



(a) level 1 (b) level 2 (c) level 3

(d) level 4 (e) level 5 (f) level 6

Figure 4: As in Fig. 3, showing the convergence of the CIM (blue), EKSM (magenta), and PAIN (red) approximations to the $f(\mathbf{A})b$ for the Neumann version of (3.13). Here the less-optimal dependence of the EKSM convergence on the condition number of $\mathbf{M}^{-1}\mathbf{L}$ is more evident as we move to the more refined levels.

convergence as a function of the number of solves n, the timing for the CIM an for PAIN are now comparable. This suggests that on multi-core or clustered machines the penalty taken by the CIM in using complex arithmetic might be compensated by the better scalability of its independent solves. Theoretically the EKSM should also benefit here by a factor of almost two, as the multiplication and solves with $\mathbf{A}$ (corresponding to solves with $\mathbf{M}$ and $\mathbf{L}$ respectively) can be solved independently, but unfortunately our attempts to exploit this in the MATLAB code lead to a decrease in speed rather than an increase.

As before, the convergence curves for the CIM and PAIN methods are fairly closely matched. However, the stronger dependence of EKSM on the condition number of $\mathbf{A}$ is more evident in this Neumann case, and furthermore, as there are no longer repeated eigenvalues or gaps in the spectrum close to the origin which it can exploit, the EKSM does not benefit from the instant accuracy we saw in the Dirichlet case. Nonetheless, the simplicity of the linear solves and the trivial parallelisation of the two independent solves in the EKSM mean, as we see in Table 2, that the same accuracy may be obtained with a comparabile computational effort by taking more iterations.

level	N	cond. no.	CIM time	PAIN time	EKSM time
1	145	405	0.0445	0.0361	0.0352
2	545	1480	0.1037	0.3985	0.0671
3	2113	5822	0.4077	0.7482	0.3058
4	8321	23301	2.4767	2.3300	1.8584
5	33025	93215	14.779	14.926	9.4105
6	131585	372878	96.589	90.468	54.131

Table 2: Results for solving equation (3.13) from the previous section, but with homogeneous Neumann boundary conditions $\partial u/\partial n = 0$. Here we solve on a three core machine using the MATLAB Parallel Computing Toolbox, and see that the CIM method scales much better than before, and the computational time is now comparable with PAIN. Although the EKSM should theoretically benefit from a factor of 2 speed up in the parallelisation of the two independent solves at each iteration, we did not manage to find a way to produce this in practice.

5 Practical Solution Methods

In the previous section we provided CPU timings when calculating the fractional Laplacian using the three methods CIM, PAIN, and EKSM. In all cases the majority of this time was taken up solving the required linear systems. Hence, to improve these methods an efficient solution method for large sparse systems is required. In the following two sections we will investigate preconditioning choices for both CIM and PAIN. All results presented are for Neumann type problems, although when applied to Dirichlet type problems results were seen to be similar. Finally, we recall that there are no shifts in the EKSM, and so standard preconditioners for mass and stiffness matrices can be used.

5.1 Iterative Methods

For both the PAIN and CIM methods we are required to solve systems of the form

$$\mathbf{A}_z := (\mathbf{L} - zk^2\mathbf{M})\mathbf{x} = \mathbf{b},$$

where $k \in R$ and $z \in C$ with $|z| = 1$. Furthermore, $Re(z), Im(z) > 0$ for the CIM and z is real and negative for the PAIN method. In both cases these systems are large and very sparse. For such systems using suitably preconditioned Krylov subspace methods may lead to reliable and efficient numerical schemes.

For $\mathbf{A}_z$ we wish to apply a preconditioner, P_z, such that:

1. $\mathbf{P}_z$ is fast to construct.

2. The action of $\mathbf{P}_z^{-1}$ is fast and requires little storage.

3. The Krylov subspace method acting on $\mathbf{P}_z^{-1}\mathbf{A}_z$ provides mesh independent convergence rates.

5.2 Preconditioning CIM

Due to the complex shifts required in the CIM we are left with an indefinite system. In this case BiCGstab [15] is an appropriate iterative solver to consider. In Table 3 we see that applying either an incomplete LU factorisation with no fill in, ILU0($\mathbf{A}_z$), or an algebraic multigrid, AMG($\mathbf{A}_z$), directly to $\mathbf{A}_z$ leads to, at best mesh dependent convergence and at worst total failure. For the ILU preconditioner this degradation with respect to mesh refinement is due to the fact that a high proportion of the shifts result in diffusion dominated systems, where it is well known that incomplete factorisation performs poorly [64]. In the AMG case the action of the inverse of the complex indefinite systems is not approximated well and results in no convergence [23]. Finally, when using $\mathbf{P} = \mathbf{L}_0$, where $\mathbf{L}_0$ is $\mathbf{L}$ with one node pinned to remove the singularity, motivated by the fact that it is well known that AMG($\mathbf{L}_0$) provides an optimal approximation of $\mathbf{L}_0^{-1}$, iteration counts increase dramatically with mesh refinement [6].

	Level 2	Level 3	Level 4	Level 5
ILU0($\mathbf{A}_z$)	13 (30)	16(50)	21(80)	29(142)
AMG($\mathbf{A}_z$)	19 (70)	(–)	(–)	(–)
AMG($\mathbf{L}_0$)	60 (155)	100 (288)	180 (570)	(–)

Table 3: Average and maximum, in brackets, number of BiCGStab iterations per quadrature point for the z_i shifted Laplacian solves required in the CIM(20). Here (–) denotes failure to converge. Levels correspond to Table 1.

We investigate the use of the five possible preconditioners: $\mathbf{P}_1 := \mathbf{L} + k^2 z_1 \mathbf{M}$, $P_2 := \mathbf{L} + ik^2 z_1 \mathbf{M}$, $\mathbf{P}_3 := \mathbf{L} + (1 - 0.5i) z_1 k^2 \mathbf{M}$ and $\mathbf{P}_4 := \mathbf{L} - k^2(z_1 - 1)\mathbf{M}$, where $z = z_1 + iz_2$. Note, $\mathbf{P}_1$ was originally considered by Laird [38], whilst $\mathbf{P}_2$ and $\mathbf{P}_3$ were considered by Erlangga et al. [22, 23].

Assuming w is not an eigenvalue of the generalised eigenvalue problem

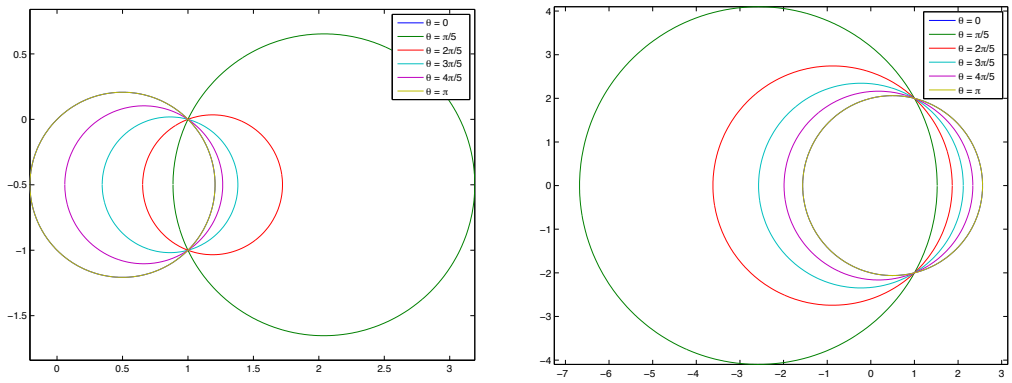
$$\mathbf{L}x = \lambda \mathbf{M}x, \quad \lambda \in \mathbb{R}^+,$$

then any eigenvalue $\sigma := \sigma_1 + i\sigma_2$ of the preconditioned system, $\mathbf{P}_j^{-1}\mathbf{A}_z$, satisfies

$$w_2\sigma_1^2 - (z_2 + w_2)\sigma_1 + w_2\sigma_2^2 + (z_1 - w_1)\sigma_2 + z_2 = 0, \tag{5.1}$$

where $\mathbf{P}_j = \mathbf{L} - (w_1 + iw_2)\mathbf{M}$. Firstly, for the preconditioning choices $\mathbf{P}_1$ and $\mathbf{P}_4$ where $w_2 = 0$ we see that the spectrum of the preconditioned system lies on the line in the complex plane satisfying

$$-z_2\sigma_1 + (z_1 - w_1)\sigma_2 + z_2 = 0.$$



(a) Spectrum for $P_2^{-1}A_z$ (b) Spectrum for $P_3^{-1}A_z$

Figure 5: Spectrum of preconditioned system for various positions on the CIM shift circle.

Secondly, for the preconditioners $\mathbf{P}_2$ and $\mathbf{P}_3$ where $w_2 \neq 0$, it can be shown that σ lies on a circle in the complex plain with centre C and radius R given by

$$C = \left(\frac{w_2 + z_2}{2w_2}, \frac{w_1 - z_1}{2w_2} \right) \quad \text{and} \quad R = \sqrt{\frac{(w_2 - z_2)^2 + (w_1 - z_1)^2}{4w_2^2}}. \tag{5.2}$$

(See Van Gijzen [24] for details).

In particular, in the CIM we use a modification of `method1.m` as described in [29], which gives rise to a Γ which is precisely a circle of radius $\lambda_N\sqrt{1-\lambda_1/\lambda_N}$ about the point λ_N (Figure 1). From here on we will call this the CIM shift circle. Writing the shift z in the form

$$z(\theta) = (\lambda_N + \lambda_N\sqrt{1-\lambda_1/\lambda_N}\cos(\theta)) + i(\lambda_N\sqrt{1-\lambda_1/\lambda_N}sin(\theta)),$$

we see that the preconditioned system remains bounded independently of the magnitude of the shift. Furthermore, when considering $\mathbf{P}_2$ we see that as $\theta \to \pi$ the CIM shift circle approaches the origin. Figure 5 shows the circles that the preconditioned systems lie on for several position on the CIM shift circles. In practice $\theta \to \pi$ as $\lambda_N \to \infty$ or the number of CIM quadrature points is increased, for accuracy reasons, the consequence can be seen in Table 4, where the maximum iteration counts for $\mathbf{P}_2$ increases as the mesh is refined, where this refinement has lead to an increase in λ_N.

The first four rows of Table 4 show iteration counts when using exact inverses and the final four rows show iteration counts when we apply just one AMG V-cycle to the preconditioners. From this table we see that $\mathbf{P}_2$ is not only mesh dependent when used exactly but also suffers when used inexactly via AMG. Furthermore, it can clearly be seen that $\mathbf{P}_1$, $\mathbf{P}_3$, and $\mathbf{P}_4$ perform well when used exactly and more importantly inexactly, where only a few more iterations are required compared to the exact solves. In Table 5 we

	Level 2	Level 3	Level 4	Level 5
$\mathbf{P}_1$	17 (23)	18 (24)	17 (23)	18 (24)
$\mathbf{P}_2$	28 (48)	31 (50)	33 (56)	38 (66)
$\mathbf{P}_3$	15 (18)	15 (17)	15 (16)	14 (16)
$\mathbf{P}_4$	12 (16)	12 (14)	12 (14)	11 (13)
AMG($\mathbf{P}_1$)	18 (22)	18 (22)	18 (22)	18 (23)
AMG($\mathbf{P}_2$)	35 (70)	40 (99)	96 (567)	61 (133)
AMG($\mathbf{P}_3$)	16 (19)	16 (19)	15 (18)	15(17)
AMG($\mathbf{P}_4$)	13 (16)	13 (16)	12 (16)	12 (15)

Table 4: Average and maximum, in brackets, number of BiCGStab iterations for various preconditioners for the z_i shifted Laplacian solve required in the 10-point CIM quadrature discretisation, CIM(10). Levels correspond to Table 1.

		Level 2	Level 3	Level 4	Level 5
CIM(20)	AMG($\mathbf{P}_1$)	17(21)	18 (22)	19 (24)	19 (24)
	AMG($\mathbf{P}_3$)	19(24)	19 (23)	19(22)	18 (22)
	AMG($\mathbf{P}_4$)	13 (23)	13 (22)	13 (21)	12 (20)
CIM(40)	AMG($\mathbf{P}_1$)	17 (21)	18 (23)	19 (24)	19 (23)
	AMG($\mathbf{P}_3$)	23 (31)	24 (30)	23 (30)	23 (30)
	AMG($\mathbf{P}_4$)	13 (36)	13 (40)	13 (39)	13 (38)

Table 5: Average and maximum, in brackets, number of BiCGStab iterations for various preconditioners for the z_i shifted Laplacian solve required in CIM with various numbers of quadrature points.

show the effects of improving the CIM accuracy by increasing the number of quadrature points in the numerical contour integration on the preconditioners. On average $\mathbf{P}_4$ requires the least number of iterations per fractional calculation, although the maximum number of iterations is larger than that of $\mathbf{P}_1$. In all cases the maximum iteration count occurs for the largest shift and in this case we could replace the preconditioner in $\mathbf{P}_4$ to that of $\mathbf{P}_1$, although no significant improvement will be obtained.

5.3 Preconditioning the PAIN Method

The solves required within the PAIN method are of the form

$$\mathbf{A}_z := (\mathbf{L} - zk^2\mathbf{M})\mathbf{x} = \mathbf{b},$$

where z are real negative. Hence, the system is symmetric positive definite and preconditioned conjugate gradients (PCG) is the iterative method of choice. In Table 6 we see that both algebraic AMG and ILU0 lead to unsatisfactory methods. In the case of ILU0($\mathbf{A}_i$) we see strong mesh dependence on the iteration counts, while in the AMG case the method fails to converge. In the case of AMG($\mathbf{A}_i$) this failure is due to the large shifts required within the PAIN method as either PAIN iterations increase or mesh size decreases. While ILU0 is not a reliable method for small shifts where it exhibits mesh dependent convergence [64], if we combine these two preconditioners into a hybrid scheme, using AMG for shifts z less than the spectral radius of $\mathbf{L}\mathbf{M}^{-1}$ and ILU0 otherwise, we may obtain a reliable mesh and iteration number independent solver. This can be seen in the bottom block of Table 6.

		Level 2	Level 3	Level 4	Level 5
AMG(A_i)	PAIN(10)	8 (23)	9 (26)	9 (25)	10 (35)
	PAIN(20)	–	–	–	–
	PAIN(40)	–	–	–	–
ILU0($\mathbf{A}_i$)	PAIN(10)	10 (21)	12 (29)	15 (40)	19 (57)
	PAIN(20)	8 (22)	9 (31)	12 (46)	19 (66)
	PAIN(40)	9 (22)	12 (33)	15 (48)	19 (71)
HYBRID($\mathbf{A}_i$)	PAIN(10)	6(10)	7(8)	7(8)	7 (8)
	PAIN(20)	6 (9)	6(8)	7 (8)	7 (8)
	PAIN(40)	6 (8)	6 (8)	7(8)	7 (8)

Table 6: Average and maximum, in brackets, number of PCG iterations for both ILU0($\mathbf{A}_i$), AMG($\mathbf{A}_i$), and HYBRID($\mathbf{A}_i$) preconditioners for the z_i shifted Laplacian solve required when using PAIN(n), $n = 10, 20, 40$.

6 Numerical Results

We have given a thorough discussion and analysis on numerical methods for scalably computing a matrix function with a fractional power, and in this section we apply these

to simulate some classes of fractional reaction-diffusion equations. We consider two types of reaction terms:

$$g_1(u) := u(1-u) \text{ and } g_2(u) := u(u-0.5)(1-u),$$

for the equation

$$\begin{aligned} \frac{\partial u}{\partial t} + \nu(-\Delta)^\alpha u = g_i(u), \qquad & \text{in } \Omega \\ \frac{\partial u}{\partial n} = 0, \qquad & \text{on } \partial\Omega, \\ u(x,0) = u_0(x), \qquad & \forall x \text{ in } \Omega, \end{aligned}$$

where $\nu > 0$. The function $g_1(u)$ represents the Fisher reaction equation leading to logistic growth, with $u = 0$ being an unstable equilibrium point. The second function $g_2(u)$ represents an Allen–Cahn equation with a quartic double well potential, which results in two stable modes with motion driven by curvature.

When $\alpha = 1$ the Fisher reaction produces a travelling wave with a smooth interface between the $u = 1$ stable state and the unstable $u = 0$ state. The thickness of this interface and the speed of the wave are determined by ν. For the Allen–Cahn reaction the interface thickness between the two stable states $u = 0$ and $u = 1$ and the speed of the curvature driven flow are determined by ν. For $0 < \alpha < 1$, Engler [21] has shown, in the case $\Omega = \mathbf{R}$, that for Fisher type reactions, with $g > 0$ on $(0, 1)$, the speed of spread of the wave front is unbounded. On the other hand for Allen–Cahn type reactions, with $g'(0) > 0$ and $g \leq 0$ in a neighbourhood of 0, the motion of any interfaces remains finite. In the following we will produce numerical results for both types of model. These results show that the computational models behave in a similar fashion to the bounds given by Engler [21].

Within each backward Euler time step $[t_n, t_{n+1}]$ we will treat the nonlinear term, $g_i(u^{n+1})$, using the following simple fixed point iteration: Given u^n, define $u^{n+1,0} := u^n$ and for $k = 1, 2, \ldots, N$ find $u^{n+1,k+1}$ such that

$$\frac{u^{n+1,k} - u^n}{\Delta t} + \nu(-\Delta)^\alpha u^{n+1,k} = g_i(u^{n+1,k-1}),$$

where N is to be chosen. Clearly, $N = 1$ leads to the fully explicit treatment of the nonlinear term and for sufficiently large N the method is fully implicit.

6.1 Fisher Equation

Here we define $\Omega_1 = (0, 20) \times (0, 0.1)$, $u_0(x, y) = e^{-5x}$, and $\nu = 10^{-3}$. This domain is chosen so as to accommodate the resulting unbounded speed of spread of the travelling front. The meshes used within the following two dimensional example have between $40,000$ and $140,000$ degrees of freedom. Figure 6 shows several evenly spaced time slices

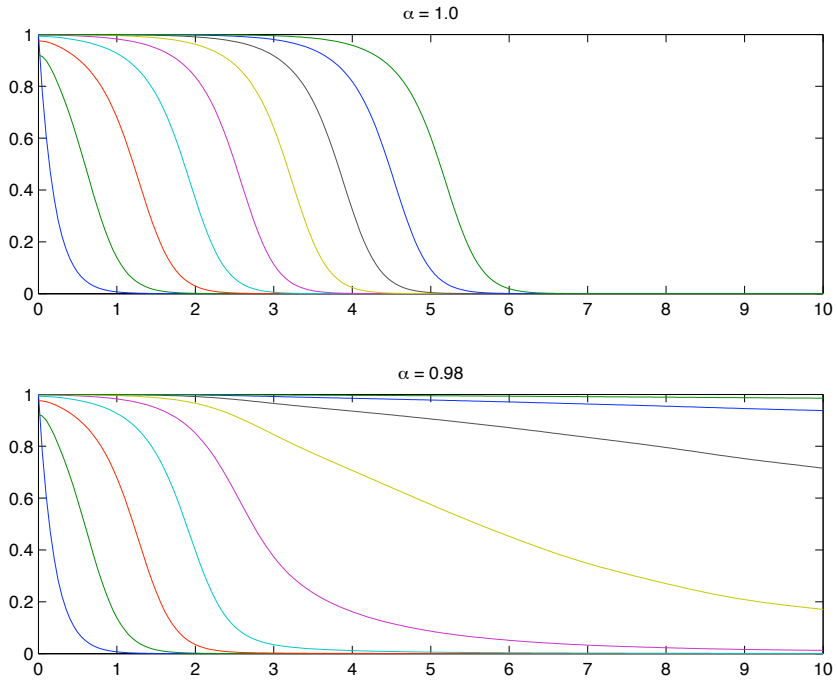


Figure 6: Fisher reaction for standard diffusion, $\alpha = 1$, top and Fractional diffusion, $\alpha = 0.98$, bottom.

for the evolution of the model for $\alpha = 1$ and $\alpha = 0.98$ through the line $y = 0$. From the top figure the steady speed of the standard Fisher equation with $\alpha = 1$ is visible. The bottom figure shows the effects of fractional diffusion, in this case $\alpha = 0.98$. The exponential spread, derived by Engler [21], of the interface is apparent.

6.2 Allen–Cahn Equation – Interfacial Properties

In this first example we are interested in the effects on the interface when fractional diffusion is used for Allen–Cahn reactions. To enable us to focus on this issue we consider the long thin domain Ω_1 and use the initial condition $u_0(x, y) = \max(\text{sign}(10 - y, 0))$. This leads to an interface centred along the line $x = 10$, the thickness of which is a function of ν. Since it is only within a region of this interface that the solution changes rapidly we use a highly non-uniform mesh, with the smallest elements around the region $5 - 1/\nu < x < 5 + 1/\nu$. Figure 7 shows the effects of α on the profile of the steady state solution for fixed $\nu = 0.01$. As expected, in the case $\alpha = 1$ we obtain a well defined interface with the profile attaining the extreme values of $u = 1$ and $u = 0$ around $x = 8$ and $x = 12$, respectively (Figure 7, top and bottom). For the cases where $\alpha < 1$ we see that the solution changes significantly faster near the centre of the interface with respect to the $\alpha = 1$ case, see the middle plot. Away from the centre the solutions become less steep, and as α is reduced the whole interface becomes thicker (Figure 7, centre). Figure 8 shows three choices for $\nu = 0.1, 0.01, 0.001$ with standard $\alpha = 1$ diffusion compared

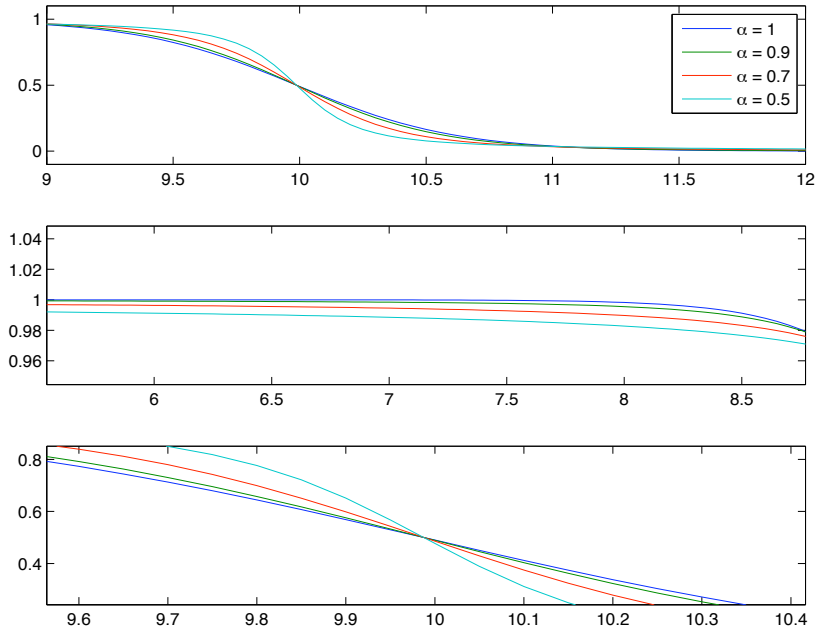


Figure 7: Allen–Cahn reactions for various powers of diffusion, $\nu = 0.01$.

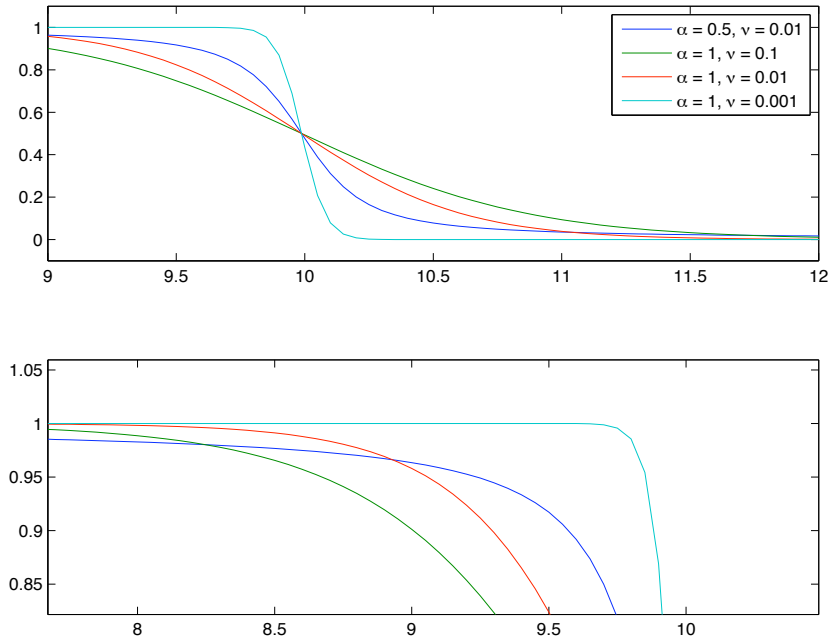


Figure 8: Allen–Cahn reactions for various ν with $\alpha = 1$ compared to $\alpha = 0.5$ with $\nu = 0.01$

to the choice $\alpha = 0.5, \nu = 0.01$. As ν decreases the upper plot shows how the interface becomes thinner and steeper at the centre. From the lower plot it is clear that the tail of the interface behaves very differently when fractional diffusion is considered. All solutions obtained using standard diffusion pass the fractional wave at some stage, due to the long tail in the fractional solution.

6.3 Allen–Cahn Equation – Motion From an Initial Cross

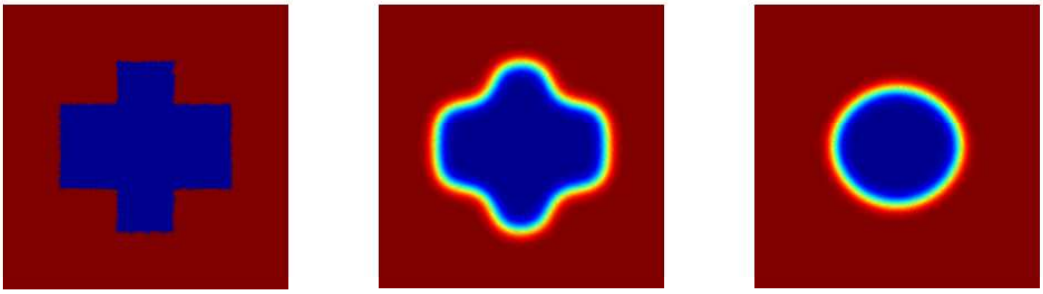


Figure 9: Allen–Cahn diffusion with $\alpha = 1.0$ and $\nu = 0.001$ at times $t = 0, 2, 100$.

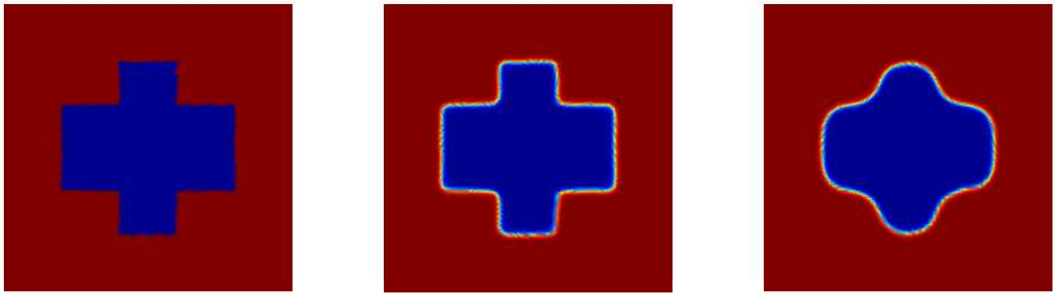


Figure 10: Allen–Cahn diffusion with $\alpha = 0.75$ and $\nu = 0.001$ at times $t = 0, 2, 100$.

In this model we define $\Omega = (0, 1) \times (0, 1)$, $u_0(x, y)$ such that the $u = 0$ phase is in the shape of a cross (Figure 9, left), and $\nu = 10^{-3}$. With standard diffusion, i.e., $\alpha = 1$, we see in Figure 9 that the curvature drives the $u = 0$ phase towards a circle (constant curvature) and shrinks. The motion for fractional diffusion, with $\alpha = 0.75$, is similar, although the rate is slower and the interface is thinner (Figure 10).

6.4 Allen–Cahn Equation – Spinodal Decomposition

In this final two dimensional example we investigate the effects of fractional diffusion when spinodal decomposition is considered. The initial state is well mixed, with $u_0(x, y)$ drawn from a random normal distribution about 0.5. For $\alpha = 1$ the early stages of phase

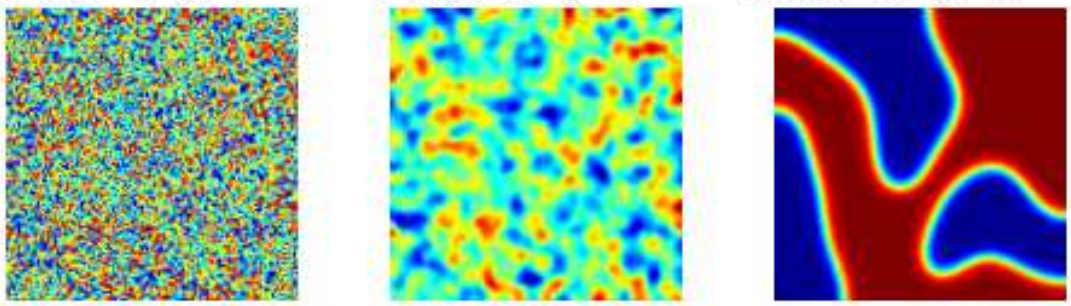


Figure 11: Allen–Cahn diffusion with $\alpha = 1.0$ and $\nu = 0.0001$ at times $t = 0.2, 16, 100$.

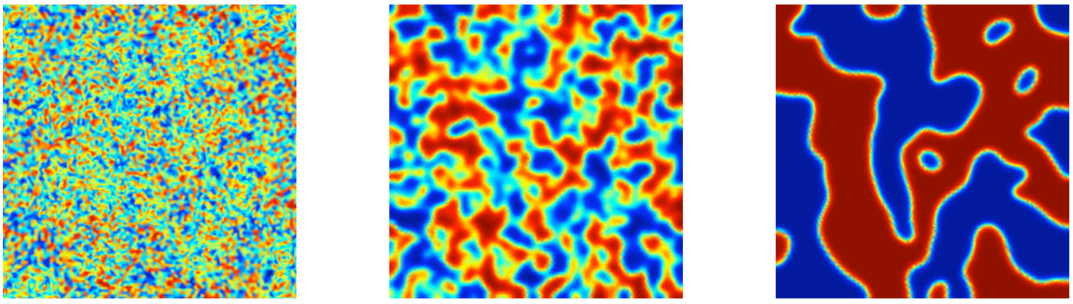


Figure 12: Allen–Cahn diffusion with $\alpha = 0.8$ and $\nu = 0.001$ at times $t = 0.2, 16, 100$.

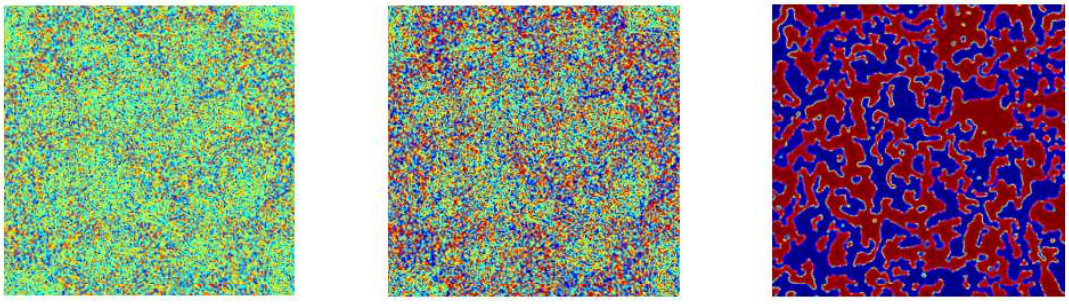


Figure 13: Allen–Cahn diffusion with $\alpha = 0.6$ and $\nu = 0.001$ at times $t = 0.2, 16, 100$.

transition produce a rapid movement to bulk regions of both phases and then motion slows (Figure 11). Reducing the fractional power in Figures 12 and 13 leads to thinner interfaces that allow for smaller bulk regions and a much more heterogeneous phase structure. Furthermore, motion to large bulk regions is vastly increased for fractional models.

6.5 Allen–Cahn Equation in Three Dimensional Space

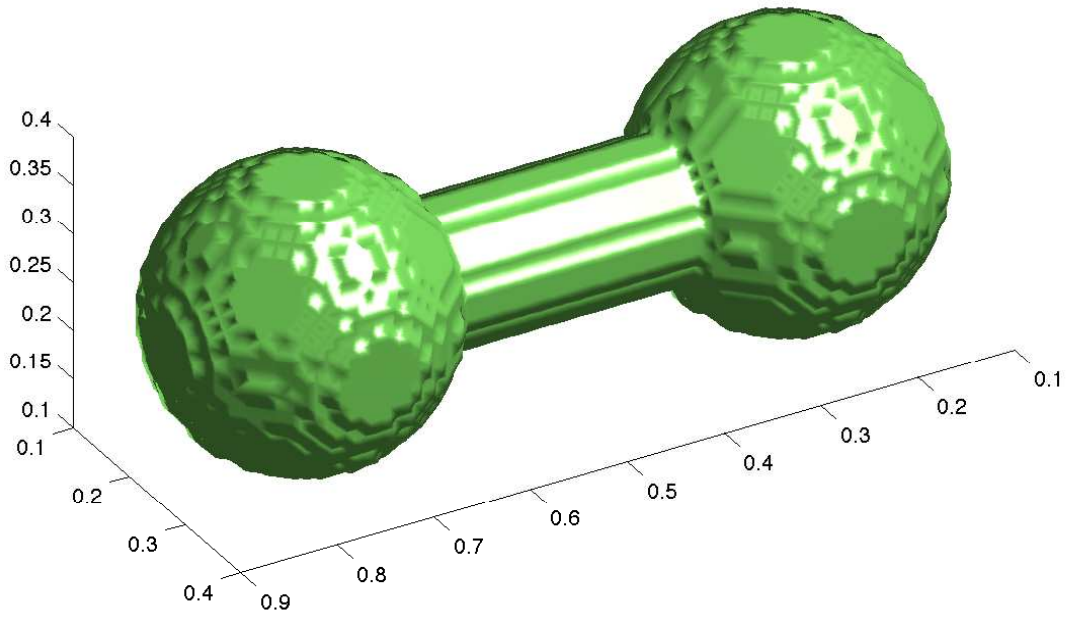
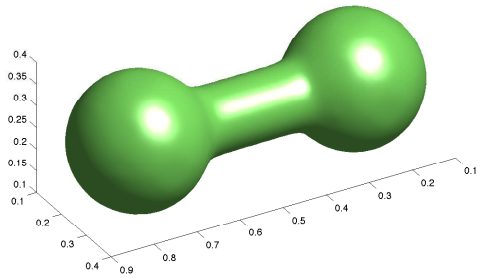


Figure 14: Initial three dimensional dumbbell.

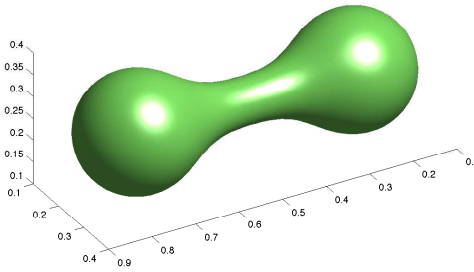
We now compare standard diffusion with fractional diffusion in three spatial dimensions. Due to the significantly large number of degrees of freedom required to obtain accurate approximations in three dimensions the methods considered in this paper are essential when considering fractional models. The following two examples are of fractional Allen–Cahn models with diffusion coefficient $\nu = 0.02$. The approximation is on a structured cuboidal mesh with $262,701$ degrees of freedom.

For the first example the initial condition is given by $u(x, y) = 0$ if $(x, y) \in D$ and 1 otherwise, where D is the dumbbell occupying the region given in Figure 14, and $\Omega = (0, 1) \times (0, 0.5) \times (0, 0.5)$. We take the The time slices in Figures 15 and 16 show the level line $u = 0.5$ for standard diffusion $\alpha = 1$ and fractional diffusion with $\alpha = 0.8$, respectively. From these we see the motion is slowed when a fractional power is considered. We observe also that, as was the case in two dimensions, the interfacial region $0 < u < 1$ in the fractional case is sharper.

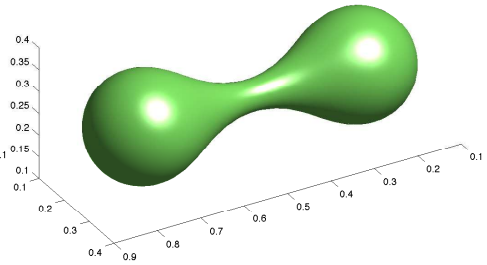
In the second example we look at spinodal decomposition. Figures 17 and 18 show similar characteristics to those seen in the two dimensional case. In particular, for



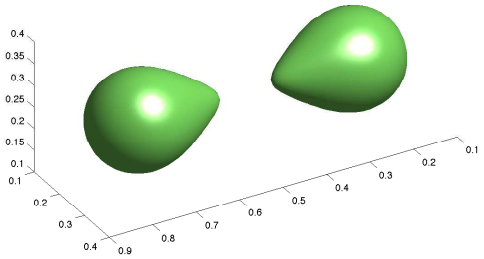
(a) $\alpha = 1$, time $= 0.4$.



(b) $\alpha = 1$, time $= 15$.

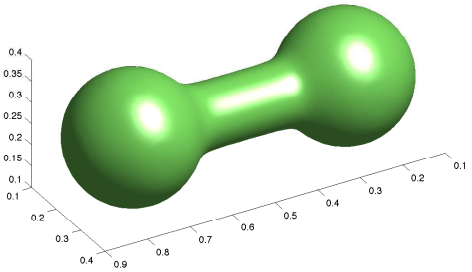


(c) $\alpha = 1$, time $= 21$.

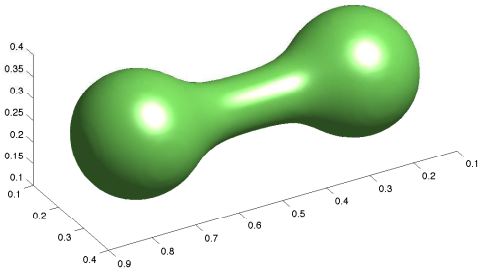


(d) $\alpha = 1$, time $= 30$.

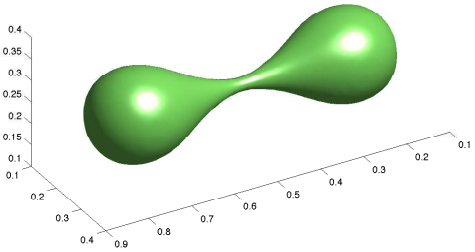
Figure 15: Level lines $u = 0.5$ for $\alpha = 1$ at times $0.4, 15, 21$ and 30.



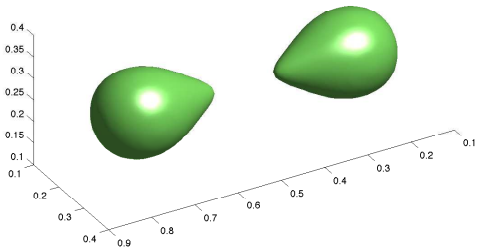
(a) $\alpha = 0.8$, time $= 15$.



(b) $\alpha = 0.8$, time $= 60$.

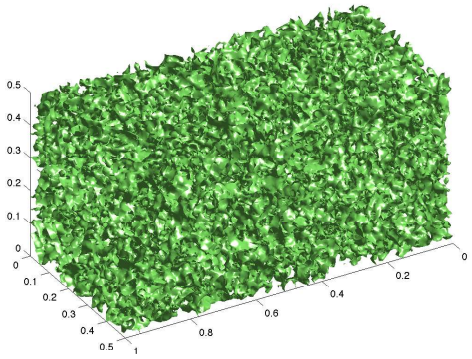


(c) $\alpha = 0.8$, time $= 147$.

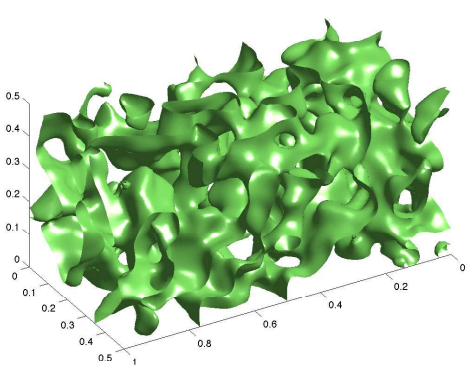


(d) $\alpha = 0.8$, time $= 165$.

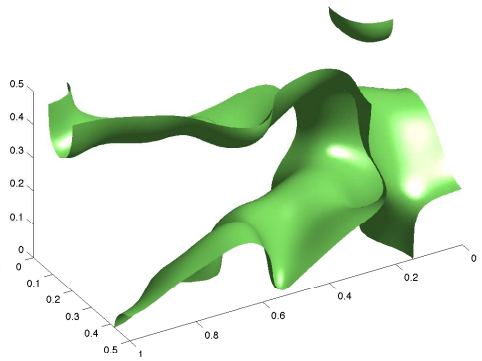
Figure 16: Level lines $u = 0.5$ for $\alpha = 0.8$ at times $15, 21, 60$ and 100.



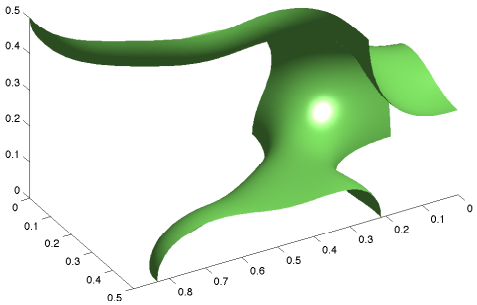
(a) $\alpha = 1$, time $= 0.4$.



(b) $\alpha = 1$, time $= 4$.

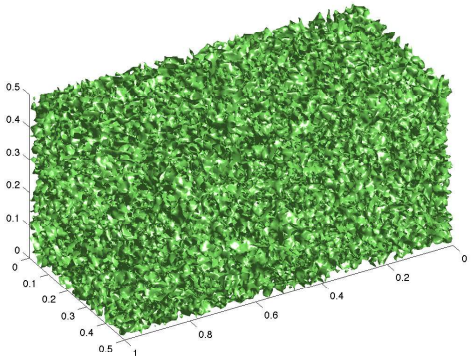


(c) $\alpha = 1$, time $= 38$.

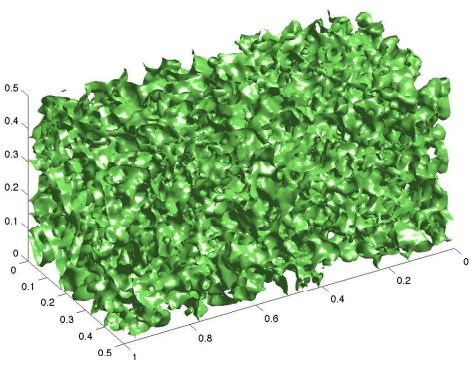


(d) $\alpha = 1$, time $= 100$.

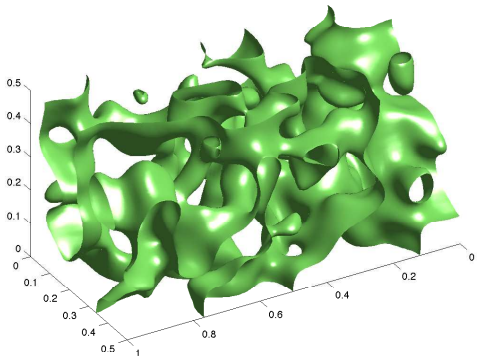
Figure 17: Level lines $u = 0.5$ for $\alpha = 1$ at times $0.4, 4, 38$ and 100.



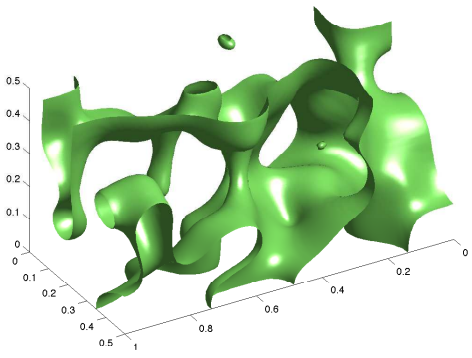
(a) $\alpha = 0.8$, time $= 15$.



(b) $\alpha = 0.8$, time $= 4$.



(c) $\alpha = 0.8$, time $= 38$.



(d) $\alpha = 0.8$, time $= 100$.

Figure 18: Level lines $u = 0.5$ for $\alpha = 0.8$ at times $0.4, 4, 38$ and 100.

$\alpha = 0.8$ the bulk regions persist for longer when compared to $\alpha = 1.0$, where at time $t = 100$ the $\alpha = 1$ case has only a couple of bulk regions whilst the fractional model has numerous bulk regions.

7 Conclusion

In this paper we have developed robust, efficient and scalable techniques for solving fractional-in-space reaction diffusion equations using finite elements on unstructure grids. We have considered three approaches: the Contour integral method, the PAIN method and the Extended Krylov subspace method. CIM is trivially parallelisable through a set of shifted linear solves and effective preconditioners can be found that lead to mesh independent convergence. The effects of inexact solves are well understood but a negative aspect is that complex arithmetic must be used. In the case of the PAIN method all computations are real and there are no issues with using effective iteration schemes. On the other hand the PAIN method is not so easily parallelisable and the effect of inexact solves is less well understood. Finally, in the case of EKSM no shifted solves are required, it is trivially parallelisable on two cores only, the convergence may not be mesh independent and the effect of inexact solves is unknown at this stage.

Simulation results on the fractional-in-space Fisher and Allen–Cahn equations show that such problems can have very different dynamics to standard diffusion models and as such represent a powerful modelling approach for understanding the many aspects of heterogeneity. Finally, many of the ideas considered in this paper can be extended to the more general framework of nonlocal operators.

Acknowledgments

The second author's contribution to this work was supported by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST). The authors are most grateful to Stefan Güttel for his insight on the PAIN method and for the MATLAB codes in Section 3.2.2 and A1 which were adapted from his thesis [28], to James Lottes for his suggestion on how to solve the stiffness plus rank 1 matrix in the EKSM method for Neumann problems, and to Ian Turner for many discussions on fractional modelling and simulation.

8 Appendix 1

```
0  %LEJADC Leja-Bagby points from two discrete sets. Adapted from [24]
1  function [a, b] = lejadc(A, B)
2  sA = ones(size(A)); sB = sA; % Initialise sum vectors
3  a = sA; a(1) = max(A);       % Initialise a and b
4  b = sB; b(1) = min(B);       % Initialise b
5  for j = 1:length(A)-1        % Loop over the discrete sets
```

```
6   sA = sA .* ((A-a(j)) ./ (A-b(j)));                % Update sum A
7   sB = sB .* ((B-a(j)) ./ (B-b(j)));                % Update Sum B
8   [~,indA] = max(abs(sA)); [~,indB] = min(abs(sB)); % Find min & max
9   a(j+1) = A(indA);        b(j+1) = B(indB);        % Update outputs
10 end
```

References

[1] N. I. Acheiser. Elements of the theory of elliptic functions. *Amer. Math, Society*, 1990.

[2] E. E. Adams and L.W. Gelhar. Field study of dispersion in a heterogeneous aquifer: 2. Spatial moment analysis. *Water Resources Res.*, 28:3293–3307, 1992.

[3] F. Andreu, J. M. Mazon, J. D. Rossi, and J. Toledo. A nonlocal p-Laplacian evolution equation with Neumann boundary conditions. *J. Math. Pures Appl.*, 90:201–227, 2008.

[4] F. Andreu, J. M. Mazon, J. D. Rossi, and J. Toledo. A nonlocal p-Laplacian evolution equation with nonhomogeneous Dirichlet boundary conditions. *SIAM J. Math. Anal.*, 40:1815–1851, 2009.

[5] E. Barkai, R. Metzler, and J. Klafter. From continuous time random walks to the fractional Fokker–Planck equation. *Phys. Rev. E*, 61:132–138, 2000.

[6] A. Bayliss, C. I. Goldstein, and E. Turkel. An iterative method for the Helmholtz equation. *J. Comp. Phys.*, 49:443–457, 1983.

[7] P. Becker-Kern, M. M. Meerschaert, and H. P. Scheffler. Limit theorem for continuous time random walks with two time scales. *J. App. Prob.*, 41:455–466, 2004.

[8] B. Beckermann and S. Güttel. Superlinear convergence of the rational Arnoldi method for the approximation of matrix functions. *Numerische Mathematik*, submitted.

[9] B. Beckermann and L. Reichel. Error estimates and evaluation of matrix functions via the Faber transform. *SIAM J. Numer. Anal.*, 47:3849–3883, 2009.

[10] D. A. Benson, S.W. Wheatcraft, and M. M. Meerschaert. Application of a fractional advection-dispersion equation. *Water Resources Res.*, 36:1403–1412, 2000.

[11] J. P. Bouchaud and A. Georges. Anomalous diffusion in disordered media: Statistical mechanisms, models and physical applications. *Phy. Rep.*, 195:127–293, 1990.

[12] C. Chen, F. Liu, and K. Burrage. Finite difference methods and a Fourier analysis for the fractional reaction-subdiffusion equation. *Appl. Math. Comp.*, 198:754–769, 2008.

[13] C. Chen, F. Liu, I. Turner, and V. Anh. Fourier method for the fractional diffusion equation describing sub-diffusion. *J. Comp. Phys.*, 227:886–897, 2007.

[14] 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, May 1987.

[15] H. A. Van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 13:631–644, 1992.

[16] T. A. Driscoll. The Schwarz–Christoffel toolbox. available online at http://www.math,udel.edu/ driscoll/software/SC/.

[17] V. Druskin and L. Knizhnerman. Extended Krylov subspaces: approximation of the matrix square root and related functions. *SIAM J. Matrix Anal. Appl.*, 19:755–771, 1998.

[18] Q. Du and K. Zhou. Mathematical analysis for the peridynamic nonlocal continuum theory. *M2AN Math. Model. Numer. Anal.*, to appear.

[19] Q. Du and K. Zhou. Mathematical and numerical analysis of linear peridynamic models with nonlocal boundary conditions. *SIAM J. Numer. Anal.*, to appear.

[20] J. Eggleston and S. Rojstaczer. Identification of large-scale hydraulic conductivity trends and the influence of trends on contaminant transport. *Water Resources Res.*, 34:2155–2168, 1998.

[21] H. Engler. On the speed of spread for fractional reaction-diffusion equations. *Int. J. Diff. Eqn.*, 2010.

[22] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik. Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Appl. Numer. Math.*, 56:648–666, 2006.

[23] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee. On a class of preconditioners for the Helmholtz equation. *Appl. Numer. Math.*, 20:409–425, 2005.

[24] M. B. Van Gijzen, Y. A. Erlangga, and C. Vuik. Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian. *SIAM J. Sci Comp.*, 29:1942–1958, 2007.

[25] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Model. Simul.*, 7:1005–1028, 2008.

[26] Q. Y. Guan and Z. M. Ma. Boundary problems for fractional Laplacians. *Stoch. Dyn.*, 5:385–424, 2005.

[27] M. Gunzburger and R. B. Lehoucq. A nonlocal vector calculus with application to nonlocal boundary value problems. *Multiscale Model. Simul.*, 8:1581–1598, 2010.

[28] S. Güttel. *Rational Krylov Methods for Operator Functions.* PhD thesis, TU Bergakademie Freiberg, 2010.

[29] N. Hale, N. J. Higham, and L. N. Trefethen. Computing $A^\alpha, \log(A)$, and related matrix functions by contour integrals. *SIAM J. Num. Anal.*, 46:2505–2523, 2008.

[30] N. J. Higham. Functions of matrices: Theory and computation. *SIAM*, 2008.

[31] U. Hornung. Homogenization and porous media. *Springer-Verlag, New York*, 1997.

[32] M. Ilić, F. Liu, I. Turner, and V. Anh. Numerical approximation of a fractional-in-space diffusion equation. *I. Frac. Calc. and App. Anal.*, 8:323–341, 2005.

[33] M. Ilić, F. Liu, I. Turner, and V. Anh. Numerical approximation of a fractional-in-space diffusion equation (II) - with nonhomogeneous boundary conditions. *Frac. Calc. and App. Anal.*, 9:333–349, 2006.

[34] M. Ilić, I. Turner, F. Liu, and V. Anh. Analytical and numerical solutions of one-dimensional fractional-in-space diffusion equation in a composite medium. *Appl. Math. Comput*, 2010.

[35] M. Ilić and I. W. Turner. Approximating functions of a large sparse positive definite matrix using a spectral splitting method. In Rob May and A. J. Roberts, editors, *Proc. of 12th Computational Techniques and Applications Conference CTAC-2004*, volume 46, pages C472–C487, June 2005. `http://anziamj.austms.org.au/V46/CTAC2004/Ilic` [June 9, 2005].

[36] 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. *J. of App. Math. and Stoch. Anal.*, 2008.

[37] L. Knizhnerman and V. Simoncini. A new investigation of the extended Krylov subspace method for matrix function evaluations. *Lin. Alg. Appl.*, 17:615–638, 2010.

[38] A. L. Laird and M. B. Giles. Preconditioning harmonic unsteady potential flow calculations. *Oxford NA Tech. Rep.*, 02/12, 2002.

[39] T. A. M. Langlands and B.I. Henry. The accuracy and stability of an implicit solution method for the fractional diffusion equation. *J. Comp. Phys.*, 205:719–736, 2005.

[40] A. Levin and E. Saff. Optimal ray sequences of rational functions connected with the Zolotarev problem. *Constructive Approximation*, 10:235–273, 1994. 10.1007/BF01263066.

[41] R. Lin, F. Liu, V. Anh, and I. Turner. Stability and convergence of a new explicit finite-difference approximation for the variable-order nonlinear fractional diffusion equation. *App. Math. and Comp.*, 212:435–445, 2009.

[42] F. Liu, V. Anh, and I. Turner. Numerical solution of the space fractional Fokker–Planck equation. *J. Comp. and App. Math.*, 166:209–219, 2004.

[43] F. Liu and K. Burrage. Parameter estimation for fractional dynamical models in biological systems. *Proceedings of FDA10, the 4th IFAC Workshop Fractional Differentiation and its Applications, Badajoz*, Eds: I. Podlubny, B. M. Vinagre Jara, YQ. Chen, V. Feliu Batlle, I. Tejado Balsera, 2010.

[44] F. Liu, I. Turner, and M. Cox. A finite volume method for saltwater intrusion with heavy-tailed motions. *In Review.*

[45] F. Liu, P. Zhuang, V. Anh, I. Turner, and K. Burrage. Stability and convergence of the finite difference method for the space-time fractional advection-diffusion equation. *App. Num. and Comp.*, 91:12–20, 2007.

[46] R. L. Magin, O. Abdullah, D. Baleanu, and X. J. Zhou. Anomalous diffusion expressed through fractional order differential operators in the Bloch–Torrey equation. *J. of Mag. Res.*, 190:255–270, 2008.

[47] M. M. Meerschaert, D. A. Benson, and S. W. Wheatcraft. Subordinated advection-dispersion equation for contaminant transport. *Water Resource Res.*, 37:1543–1550, 2001.

[48] M. M. Meerschaert, J. Mortensenb, and S. W. Wheatcraft. Fractional vector calculus for fractional advection-dispersion. *Physica A*, 367:181–190, 2006.

[49] M. M. Meerschaert and C. Tadjeran. Finite difference approximations for two-sided space-fractional partial differential equations. *App. Num. Math*, 56:80–90, 2006.

[50] R. Metzler and J. Klafter. The random walk's guide to anomalous diffusion: A fractional dynamics approach. *Phys. Rep.*, 339:1–77, 2000.

[51] C. Ray, T. R. Ellsworth, A. J. Valocchi, and C. W. Boast. An improved dual porosity model for chemical transport in macroporous soils. *J. of Hydrology*, 193:270–292, 1997.

[52] L. Reichel. Newton interpolation at Leja points. *BIT*, 30:332–346, 1990.

[53] J. Roop. Computational aspects of fem approximations of fractional advection dispersion equations on bounded domains on r^2. *J. Comp. Appl. Math.*, 193:243–268, 2005.

[54] A. Ruhe. Rational Krylov sequences for eigenvalue computations. *Lin. Alg. Appl*, 58:391–405, 1984.

[55] A. Ruhe. Rational Krylov: a practical algorithm for large sparse nonsymmetric matrix pencils. *SIAM J. Sci. Comput.*, 19:1535–1551, 1998.

[56] Y. Saad. Preconditioning techniques for non symmetric and indefinite linear system. *J. Comp. Appl. Math.*, 24:89–105, 1988.

[57] M. Saxton. Anomalous subdiffusion in fluorescence photobleaching recovery: A Monte Carlo study. *J. Biophys*, 81:2226–2240, 2001.

[58] E. Scalas, R. Gorenflo, and F. Mainardi. Fractional calculus and continuous time finance. *Physica A*, 284:376–384, 2000.

[59] R. Schumer, D. A. Benson, M. M. Meerschaert, and B. Baeumer. Fractal mobile/immobile solute transport. *Water Res.*, 39:1296–1307, 2003.

[60] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.*, 29:1268–1288, May 2007.

[61] D. P. Simpson. *Krylov subspace methods for approximating functions of symmetric positive definite matrices with applications to applied statistics and anomalous diffusion*. PhD thesis, QUT, 2008.

[62] P. Smith, I. Morrison, K. Wilson, N. Fernandez, and R. Cherry. Anomalous diffusion of major histocompatability complex class I molecules on HeLa cells determined by single particle tracking. *J. Biophys*, 76:3331–3344, 1999.

[63] I. Turner, M. Ilić, and P. Perr. The use of fractional-in-space diffusion equations for describing microscale diffusion in porous media. In *11th International Drying Conference, Magdeburg, Germany*, 2010.

[64] A. J. Wathen. Realistic eigenvalue bounds for the galerkin mass matrix. *IMA J. Numer. Anal.*, 7:449–457, 1987.

[65] S. Whitaker. Flow in porous media I: A theoretical derivation of Darcys law. *Transp. Porous Media*, 1:3–25, 1986.

[66] Q. Yang. Novel analytical and numerical methods for solving fractional dynamical systems. *PhD Thesis, QUT*, 2005.

[67] Q. Yang, F. Liu I. Turner, and M. Ilic. Novel numerical methods for solving the time-space fractional diffusion equation in 2d. *SIAM J. Sci. Comp.*, 33:1159–1180, 2011.

[68] Q. Yang, F. Liu, and I. Turner. Numerical methods for fractional partial differential equations with Riesz space fractional derivatives. *App. Num. Mod.*, 34:200–218, 2010.

[69] S. B. Yuste and L. Acedo. An explicit finite difference method and a new Von Neumann-type stability analysis for fractional diffusion equations. *SIAM J. Numer. Anal.*, 42:1862–1874, 2005.

[70] S. B. Yuste, L. Acedo, and K. Lindenberg. Reaction front in an A + B - C reaction-subdiffusion process. *Phys. Rev. E*, 69, 2004.

[71] Y. Zhang, D. A. Benson, and D. M. Reeves. Time and space nonlocalities underlying fractional-derivative models: Distinction and literature review of field applications. *Adv. in Water Res.*, 32:561–581, 2009.

[72] P. Zhuang, F. Liu, V. Anh, and I. Turner. Numerical methods for the variable-order fractional advection-diffusion equation with a nonlinear source term. *SIAM J. Numer. Anal.*, 47:1760–1781, 2009.