

A Fast Gradient Projection Method for a Constrained Fractional Optimal Control

Ning Du¹ · Hong Wang² · Wenbin Liu³

Received: 17 October 2014 / Accepted: 20 October 2015 / Published online: 14 November 2015
© Springer Science+Business Media New York 2015

Abstract Optimal control problems governed by a fractional diffusion equation tends to provide a better description than one by a classical second-order Fickian diffusion equation in the context of transport or conduction processes in heterogeneous media. However, the fractional control problem introduces significantly increased computational complexity and storage requirement than the corresponding classical control problem, due to the nonlocal nature of fractional differential operators. We develop a fast gradient projection method for a pointwise constrained optimal control problem governed by a time-dependent space-fractional diffusion equation, which requires the computational cost from $O(MN^3)$ of a conventional solver to $O(MN \log N)$ and memory requirement from $O(N^2)$ to $O(N)$ for a problem of size N and of M time steps. Numerical experiments show the utility of the method.

Keywords Constrained optimal control · Fractional diffusion equation · Fast Fourier transform · Preconditioned conjugate gradient method

This work was supported by the National Natural Science Foundation of China under Grants 11371229, 91130010, and 11471194, by the National Science Foundation under Grants EAR-0934747 and DMS-1216923 and, by the China Scholarship Council (File No. 201308370102).

✉ Hong Wang
hwang@math.sc.edu

Ning Du
duning@sdu.edu.cn

Wenbin Liu
W.B.Liu@ukc.ac.uk

¹ School of Mathematics, Shandong Univeristy, Jinan, Shandong, China

² Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

³ KBS, University of Kent, Canterbury CT2 7NF, UK

1 Introduction

It is well known that numerical approximation of optimal control problems has long been an important topic in engineering design work. There has been extensive researches on developing fast numerical algorithms for these optimal control problems. Some recent progress in this area can be found in [5, 8, 12–17, 21–23, 25, 29] and the references cited therein.

In this paper we consider the model of a pointwise-constrained time-dependent optimal control problem imposed on the space interval $(0, 1)$ over the time period $[0, T]$: Find a control $u \in K := \{v \in L^2(0, T; (0, 1)), v \geq u_0\}$ with a prescribed function $u_0(x, t)$ to minimize the objective functional

$$J(u) := \frac{1}{2} \int_0^T \int_0^1 (y - \tilde{y})^2 dx dt + \frac{1}{2} \int_0^T \int_0^1 u^2 dx dt, \quad (1)$$

which is governed by the initial-boundary value problem of a diffusion equation for the state variable $y(x, t)$

$$\begin{aligned} Ly &= f + u, \quad (x, t) \in (0, 1) \times (0, T), \\ y(0, t) &= y(1, t) = 0, \quad t \in [0, T], \quad y(x, 0) = y_0(x), \quad x \in [0, 1]. \end{aligned} \quad (2)$$

Here $\tilde{y}(x, t)$ represents a prescribed or observed value of the state variable y , $f(x, t)$ is a prescribed source and sink term, u is the control variable, and L represents a diffusion operator.

This type of problems occurs widely in engineering design and applications. For instance, in the temperature control in a thermo conduction process, y represents the temperature in a medium and u the heating or cooling source. The goal of the control problem is to optimize the input of the heating or cooling source u such that the temperature distribution u is as close as to the prescribed temperature distribution \tilde{y} . Due to the limitation of the surrounding environments or equipments, the input control u might be subject to certain constraint which leads to a constrained control problem such as that with a pointwise obstacle constraint as described by the admissible set K . Similar models also occur in the optimal control in mass diffusive transport process in porous media and in the photocopier and the laser printer in the modern office that rely on the transport of electrons in amorphous semiconductors in an electric field. The resulting optimality conditions from these models consist of non-smooth nonlinear systems of partial differential equations.

The canonical optimal control problems of this type, in which the differential operator L is a classical second-order Fickian diffusion operator, have been studied extensively in the literature and many efficient and reliable methods have been developed [5, 12–15, 17, 21, 25]. There exist at least three general classes of computational approaches to solving the resulting nonlinear systems of partial differential equations derived from the optimality conditions. The primal-dual active set method employs the primal-dual active set strategy based on the augmented Lagrangian method in [12, 13]. This method has widely been used in solving numerical solutions of different kinds of optimal control in the literature. However in some cases, e.g. when the Lagrange multipliers are irregular, the active sets are not easy to be computed numerically. The second class is to apply the semi-Newton algorithm to solve the resulting optimality conditions, which involves computations of second derivatives, see, e.g., [14, 15]. There exists an extensive body of research on the above two methods in the literature, and it is even impossible to give a brief review here. The third kind is the class of gradient projection methods that use a line search method to minimize the objective functional, guided by the gradient whose computation needs repeatedly solving the state and

the co-state equations. This class of methods has been found robust in practice [17], but its effectiveness relies on very efficient solvers for the state and the co-state equations.

In the last few decades an increasingly more number of evidence suggests that the classical second-order Fickian diffusion equation or heat conduction equation does not necessarily provides a proper description of the diffusion or heat conduction process in a heterogeneous medium. A microscopic explanation is as follows: In a normal diffusion process modeled by a classical second-order Fickian diffusion or heat conduction equation, a particle’s motion in the underlying transport or conduction process has virtually no spatial correlation and long walks in the same direction are rare, so the variance of a particle excursion distance is finite. The central limit theorem ensures that the probability of finding a particle somewhere in space satisfies a normal distribution, which gives a probabilistic description of a normal diffusion. However, in a diffusion or heat conduction process in a heterogeneous medium, a particle’s motion is better described by steps that are not independent and that can take vastly different times to perform. A particle’s long walk may have long correlation length so the processes can have large deviations from the stochastic process of Brownian motion or even have an infinite variance. Thus, the assumptions of (i) the existence of a mean free path and (ii) the existence of a mean time taken to perform a step, which were made by Einstein in his derivation of diffusion equation or Brownian motion [9], are violated. As a matter of fact, it was shown [3, 20] that the heavy tail behavior of the transport processes in heterogeneous media can be described accurately by Levy distribution, which can be viewed as a probability description of fractional diffusion equations. This is the fundamental reason why fractional diffusion or heat conduction equations provide a better description of transport processes in heterogeneous media than classical Fickian diffusion or heat conduction equations do.

Motivated by these facts, in this paper we consider the optimal control problem of (1)–(2) except that L is a fractional diffusion operator given by

$$Ly := \frac{\partial y}{\partial t} - d(x, t) \left(r {}_0D_x^{2-\beta} + (1-r) {}_xD_1^{2-\beta} \right) y, \quad (x, t) \in (0, 1) \times (0, T] \quad (3)$$

where $d(x, t)$ is the diffusivity coefficient with positive lower and upper bounds, $0 \leq r \leq 1$ indicates the relative weight of forward versus backward transition probability, $2 - \beta$ with $0 < \beta < 1$ is the order of the anomalous diffusion in the governing equation. The left and right fractional integrals are defined by Podlubny [24]

$$\begin{aligned} {}_0D_x^{-\beta} w(x) &:= \frac{1}{\Gamma(\beta)} \int_0^x \frac{w(s)}{(x-s)^{1-\beta}} ds, \\ {}_xD_1^{-\beta} w(x) &:= \frac{1}{\Gamma(\beta)} \int_x^1 \frac{w(s)}{(s-x)^{1-\beta}} ds \end{aligned}$$

For any positive integer m , the left and right Riemann–Liouville fractional derivatives of order $m - \beta$ are defined by

$$\begin{aligned} {}_0D_x^{m-\beta} w(x) &:= D^m {}_0D_x^{-\beta} w(x) = \frac{1}{\Gamma(\beta)} \frac{d^m}{dx^m} \int_0^x \frac{w(s)}{(x-s)^{1-\beta}} ds, \\ {}_xD_1^{m-\beta} w(x) &:= (-D)^m {}_xD_1^{-\beta} w(x) = \frac{(-1)^m}{\Gamma(\beta)} \frac{d^m}{dx^m} \int_x^1 \frac{w(s)}{(s-x)^{1-\beta}} ds. \end{aligned}$$

Because of the improved modeling capability of the fractional differential equations, there has been increasingly more research on both mathematical analysis and numerical approximation of the fractional control problems [1, 7, 10]. However, because of the nonlocal nature of fractional differential operators, the corresponding numerical methods for solving

the fractional state and costate equations generate full stiffness matrices. Consequently, the resulting computational complexity is of $O(N^3)$ and the memory requirement is of $O(N^2)$ for a problem of size N , which is in sharp contrast to the computational work and memory requirement of $O(N)$ for numerical methods for second-order Fickian diffusion equations. Recall that the fractional state and costate equations have to be iteratively solved for many times. Hence, the computational bottleneck becomes a really severe burden. To our best knowledge, there is no work in the literature that has addressed the significantly increased computational cost issue.

The goal of this paper is to develop a fast gradient projection method for the fractional optimal control problem (1)–(3), which significantly reduces the computational cost of the solution of the fractional state and costate equations to $O(MN \log N)$ from $O(MN^3)$, with M being the number of time steps, per iterative solve of the two equations, and the memory requirement to $O(N)$ from $O(N^2)$. Numerical experiments justify these. The rest of the paper is organized as follows. In Sect. 2 we derive the optimality condition for the fractional optimal control problem. In Sect. 3 we present a fractional gradient projection method. In Sect. 4 we present a fast and faithful solution projection method. In Sect. 5 we carry out numerical experiments to validate the fast gradient projection method developed.

2 The Optimality Condition of the Fractional Control Problem

We frequently use the following properties in this section. The left and right Riemann–Liouville fractional integral operators are adjoints in the L^2 sense [24, 26]:

$$\int_0^1 ({}_0D_x^{-\beta} w) v dx = \int_0^1 w ({}_x D_1^{-\beta} v) dx, \quad \forall v, w \in L^2(0, 1). \quad (4)$$

For any v, w with $v(0) = 0$ and $w(1) = 0$, the fractional integral operator and the classical first-order differential operators commute [33]:

$${}_0D_x^{-\beta} Dv = D{}_0D_x^{-\beta} v, \quad {}_x D_1^{-\beta} Dw = D{}_x D_1^{-\beta} w. \quad (5)$$

To derive the adjoint costate equation for the costate function we rewrite the state equation (2) as

$$\begin{aligned} d^{-1} \frac{\partial y}{\partial t} - (r {}_0D_x^{2-\beta} + (1-r) {}_x D_1^{2-\beta}) y &= d^{-1} (f + u), \\ y(0, t) = y(1, t) &= 0, \quad t \in [0, T], \\ y(x, 0) = y_0(x), \quad x &\in [0, 1]. \end{aligned} \quad (6)$$

We then multiply the equation by any test function p with appropriately specified boundary condition and terminal conditions [as in (7)] and utilize the adjoint property (4) to obtain the following terminal-boundary value problem of the adjoint costate equation for the costate function $p(x, t)$

$$\begin{aligned} -\frac{\partial (d^{-1} p)}{\partial t} - ((1-r) {}_0D_x^{2-\beta} + r {}_x D_1^{2-\beta}) p &= y - \tilde{y}, \\ p(0, t) = p(1, t) &= 0, \quad t \in [0, T], \\ p(x, T) = 0, \quad x &\in [0, 1]. \end{aligned} \quad (7)$$

To derive the optimality condition for (1), we calculate the Gâteaux derivative of the objective functional $J(u)$. Let $u \in K$ be the minimizer of the objective functional $J(u)$. Let

$v \in K$ be any admissible function and $\delta u = v - u$. For any arbitrarily small positive number ε , we note that

$$\delta y := \lim_{\varepsilon \rightarrow 0^+} \frac{y(u + \varepsilon \delta u) - y(u)}{\varepsilon} = y'(u) \delta u \tag{8}$$

exists, provided that y is differentiable with respect to u . Consequently,

$$\begin{aligned} y(u + \varepsilon \delta u) &= y(u) + \varepsilon y'(u) \delta u + o(|\varepsilon \delta u|) \\ &= y(u) + \varepsilon \delta y + o(|\varepsilon \delta u|) \end{aligned}$$

as ε tends to zero. Therefore, we expand $J(u + \varepsilon \delta u)$ as follows

$$\begin{aligned} J(u + \varepsilon \delta u) &= \frac{1}{2} \int_0^T \int_0^1 (y(u + \varepsilon \delta u) - \tilde{y})^2 dx dt \\ &\quad + \frac{1}{2} \int_0^T \int_0^1 (u + \varepsilon \delta u)^2 dx dt \\ &= \frac{1}{2} \int_0^T \int_0^1 (y(u) - \tilde{y})^2 + 2\varepsilon(y(u) - \tilde{y})\delta y dx dt \\ &\quad + \frac{1}{2} \int_0^T \int_0^1 u^2 + 2\varepsilon u \delta u dx dt + o(\varepsilon). \end{aligned}$$

We accordingly find that

$$\begin{aligned} J'(u) \delta u &= \lim_{\varepsilon \rightarrow 0^+} \frac{J(u + \varepsilon \delta u) - J(u)}{\varepsilon} \\ &= \int_0^T \int_0^1 (y - \tilde{y}) \delta y dx dt + \int_0^T \int_0^1 u \delta u dx dt \\ &= \int_0^T \int_0^1 \left[-\frac{\partial(d^{-1}p)}{\partial t} - \left((1-r) {}_0D_x^{2-\beta} + r {}_x D_1^{2-\beta} \right) p \right] \delta y dx dt \\ &\quad + \int_0^T \int_0^1 u \delta u dx dt. \end{aligned} \tag{9}$$

In the last step we have replaced $y - \tilde{y}$ in the integrand of the first integral by the left-hand side of the costate equation (7).

We see from (8) that δy satisfies the homogeneous initial and boundary conditions. We further utilize the fact that $p(x, T) = 0$ and integrate the first integrand in the first integral on the right-hand side with respect to time by parts to get

$$\int_0^T \int_0^1 \frac{\partial(d^{-1}p)}{\partial t} \delta y dx dt = - \int_0^T \int_0^1 p d^{-1} \frac{\partial \delta y}{\partial t} dx dt. \tag{10}$$

We intend to utilize the facts that δy and p both vanish at the boundary to integrate the second integrand in the first integral on the right-hand side with respect to x by parts in a similar fashion. However because the fractional integral operators and the classical differential operators do not commute in general, we have to take extra care by utilizing (4) and (5) repeatedly to obtain

$$\begin{aligned} &\int_0^T \int_0^1 \left[\left((1-r) {}_0D_x^{2-\beta} + r {}_x D_1^{2-\beta} \right) p \right] \delta y dx dt \\ &= \int_0^T \int_0^1 \left[\left((1-r) D^2 {}_0D_x^{-\beta} + r D^2 {}_x D_1^{-\beta} \right) p \right] \delta y dx dt \end{aligned}$$

$$\begin{aligned}
&= \int_0^T \int_0^1 \left[\left((1-r) D_0 D_x^{-\beta} D + r D_x D_1^{-\beta} D \right) p \right] \delta y dx dt \\
&= - \int_0^T \int_0^1 \left[\left((1-r) {}_0 D_x^{-\beta} D + r {}_x D_1^{-\beta} D \right) p \right] (D \delta y) dx dt \\
&= - \int_0^T \int_0^1 \left[\left(r {}_0 D_x^{-\beta} + (1-r) {}_x D_1^{-\beta} \right) D \delta y \right] D p dx dt \\
&= - \int_0^T \int_0^1 \left[\left(r D {}_0 D_x^{-\beta} + (1-r) D {}_x D_1^{-\beta} \right) \delta y \right] D p dx dt \\
&= \int_0^T \int_0^1 \left[\left(r D^2 {}_0 D_x^{-\beta} + (1-r) D^2 {}_x D_1^{-\beta} \right) \delta y \right] p dx dt \\
&= \int_0^T \int_0^1 \left[\left(r {}_0 D_x^{2-\beta} + (1-r) {}_x D_1^{2-\beta} \right) \delta y \right] p dx dt. \tag{11}
\end{aligned}$$

We incorporate (10) and (11) into (9) to obtain

$$\begin{aligned}
J'(u) \delta u &= \int_0^T \int_0^1 \left[d^{-1} \frac{\partial \delta y}{\partial t} - \left(r {}_0 D_x^{2-\beta} + (1-r) {}_x D_1^{2-\beta} \right) \delta y \right] p dx dt \\
&\quad + \int_0^T \int_0^1 u \delta u dx dt.
\end{aligned}$$

We observe from (6) and (8) that

$$d^{-1} \frac{\partial \delta y}{\partial t} - \left(r {}_0 D_x^{2-\beta} + (1-r) {}_x D_1^{2-\beta} \right) \delta y = d^{-1} \delta u.$$

Recall that u is the minimizer of the objective functional $J(\cdot)$ and $\delta u = v - u$, the fact that $J'(u) \delta u \geq 0$ leads to the following variational inequality

$$\begin{aligned}
J'(u) \delta u &= \int_0^T \int_0^1 (d^{-1} p + u) \delta u dx dt \\
&= \int_0^T \int_0^1 (d^{-1} p + u) (v - u) dx dt \geq 0 \tag{12}
\end{aligned}$$

for all $v \in K$. This variational inequality has the following closed form solution [17]

$$u = \max \{u_0, -d^{-1} p\}. \tag{13}$$

As a matter of fact, if $-d^{-1} p > u_0$ on the entire interval $(0, 1)$, then (13) implies $u = -d^{-1} p$ and so $d^{-1} p + u = 0$ on the entire interval $(0, 1)$. The variational inequality (12) becomes identically zero for all $v \in K$. If $-d^{-1} p \leq u_0$ on the entire interval $(0, 1)$, (13) implies $u = u_0$. Thus, $d^{-1} p + u = d^{-1} p + u_0 \geq 0$. Since $v - u = v - u_0 \geq 0$ for all $v \in K$, the variational inequality (12) holds.

In the general case, the maximum in (13) may be reached by either on different subdomains of $[0, 1] \times [0, T]$. Suppose that $u = -d^{-1} p$ on Ω_1 and $u = u_0$ on Ω_2 with $\Omega_1 \cup \Omega_2 = [0, 1] \times [0, T]$. It follows that the integral $\int_{\Omega_1} (d^{-1} p + u) (v - u) dx dt = 0$. Hence, the integral in (12) reduces to an integral on Ω_2 , on which $u = u_0$. Then the same argument in the preceding paragraph shows that this integral remains nonnegative for all $v \in K$. Therefore, (12) still holds.

3 A Fractional Gradient Projection Method

The optimality condition of the constrained optimal control problem (1) and (2) reduces the problem to the solution of a coupled system of the state equation (6), the costate equation (7), and the optimal control condition (13). A commonly used method for the efficient solution of the system is the following sequentially decoupled gradient projection method [17]:

- (i) Pick up an guess of the control u^0 .
For $n = 0, 1, \dots$, do
- (ii) With the known control u^n , solve the state equation forward in time for the state variable y^n

$$d^{-1} \frac{\partial y^n}{\partial t} - \left(r {}_0 D_x^{2-\beta} + (1-r) {}_x D_1^{2-\beta} \right) y^n = d^{-1} \left(f + u^n \right).$$

- (iii) With the known state variable y^n , solve the costate equation backward in time for the costate variable p^n

$$-\frac{\partial (d^{-1} p^n)}{\partial t} - \left((1-r) {}_0 D_x^{2-\beta} + r {}_x D_1^{2-\beta} \right) p^n = y^n - \tilde{y}.$$

- (iv) Update the optimal control

$$u^{n+1} = \max \{ u_0, -d^{-1} p^n \}.$$

- (v) If $\|u^{n+1} - u^n\|_\infty$ is within the tolerance, then stop
end do

It is clear that in the gradient projection method, the major computational cost comes from the repeated solution of the state equation (6) and the costate equation (7). In the rest of this section, we present two finite difference methods for these equations and discuss related computational issues.

Let M and N be positive integers. Define the time step size $\tau := T/M$ and the spatial mesh size $h := 1/(N + 1)$, respectively. We define the temporal partition $t^k := k\tau$ for $k = 0, 1, \dots, M$ and the spatial partition $x_i := ih$ for $i = 0, 1, \dots, N + 1$. Let y_i^k, p_i^k, u_i^k be the finite difference approximations to $y(x_i, t^k), p(x_i, t^k), u(x_i, t^k)$, respectively, and denote $(d^{-1})_i^k := d^{-1}(x_i, t^k), f_i^k := f(x_i, t^k), \tilde{y}_i^k := \tilde{y}(x_i, t^k)$. Then the Meerschaert finite difference scheme for the state equation can be formulated as follows [19]: For $k = 1, 2, \dots, M$, solve the following equation forward in time to find $\{y_i^k\}_{i=1}^N$ such that

$$\begin{aligned} (d^{-1})_i^k \frac{y_i^k - y_i^{k-1}}{\tau} - \left[r \sum_{j=0}^{i-1} g_j^{(2-\beta)} y_{i-j+1}^k + (1-r) \sum_{j=0}^{N-i+1} g_j^{(2-\beta)} y_{i+j-1}^k \right] h^{\beta-2} \\ = (d^{-1})_i^k (f_i^k + u_i^k). \end{aligned} \tag{14}$$

Here the fractional binomial coefficients $g_j^{(\alpha)}$ can be evaluated recursively

$$g_0^{(\alpha)} = 1, \quad g_j^{(\alpha)} = \left(1 - \frac{\alpha + 1}{j} \right) g_{j-1}^{(\alpha)} \quad \text{for } j \geq 1.$$

We can similarly formulate the Meerschaert finite difference scheme for the costate equation: For $k = M, M - 1, \dots, 1$, solve the following equation backward in time to find $\{p_i^k\}_{i=1}^N$ such that

$$\begin{aligned} & \frac{(d^{-1})_i^{k-1} p_i^{k-1} - (d^{-1})_i^k p_i^k}{\tau} - \left[(1-r) \sum_{j=0}^{i+1} g_j^{(2-\beta)} p_{i-j+1}^k + r \sum_{j=0}^{N-i+1} g_j^{(2-\beta)} p_{i+j-1}^k \right] h^{\beta-2} \\ & = y_i^{k-1} - \tilde{y}_i^{k-1}. \end{aligned} \tag{15}$$

Because a shifted Grünwald approximation was used, the Meerschaert scheme has only first-order accuracy in both space and time [19].

Second-order finite difference methods were developed to improve the computational efficiency. A Crank-Nicolson temporal discretization plus a Richardson extrapolation of the spatial discretization in (14) results in a scheme that has second-order accuracy in both space and time [18]. In this paper we use an alternative second-order scheme, the CN-WSGD scheme, developed in [28]: For $k = 1, 2, \dots, M$, solve the following equation forward in time to find $\{y_i^k\}_{i=1}^N$ such that

$$\begin{aligned} & (d^{-1})_i^{k-\frac{1}{2}} \frac{y_i^k - y_i^{k-1}}{\tau} - \left[r \sum_{j=0}^{i+1} w_j^{(2-\beta)} y_{i-j+1}^{k-\frac{1}{2}} + (1-r) \sum_{j=0}^{N-i+1} w_j^{(2-\beta)} y_{i+j-1}^{k-\frac{1}{2}} \right] h^{\beta-2} \\ & = (d^{-1})_i^{k-\frac{1}{2}} \left(f_i^{k-\frac{1}{2}} + u_i^{k-\frac{1}{2}} \right). \end{aligned} \tag{16}$$

Here the superscript $k - \frac{1}{2}$ denotes the mean value of the function on k and $k - 1$ levels and $w_j^{(\alpha)}$ can be evaluated by

$$w_0^{(\alpha)} = \frac{\alpha}{2} g_0^{(\alpha)}, \quad w_j^{(\alpha)} = \frac{\alpha}{2} g_j^{(\alpha)} + \frac{2-\alpha}{2} g_{j-1}^{(\alpha)} \quad \text{for } j \geq 1.$$

We can similarly formulate the CN-WSGD scheme for the costate equation as follows: For $k = M, M - 1, \dots, 1$, solve the following equation backward in time to find $\{p_i^k\}_{i=1}^N$ such that

$$\begin{aligned} & \frac{(d^{-1})_i^{k-1} p_i^{k-1} - (d^{-1})_i^k p_i^k}{\tau} - \left[(1-r) \sum_{j=0}^{i+1} w_j^{(2-\beta)} p_{i-j+1}^{k-\frac{1}{2}} + r \sum_{j=0}^{N-i+1} w_j^{(2-\beta)} p_{i+j-1}^{k-\frac{1}{2}} \right] h^{\beta-2} \\ & = y_i^{k-\frac{1}{2}} - \tilde{y}_i^{k-\frac{1}{2}}. \end{aligned} \tag{17}$$

The nonlocal property of fractional differential operators results in a significant increase in the already expensive computational and storage cost in the gradient projection method for the constrained optimal control problem. Despite that the optimal control constrained by space-fractional diffusion equations provides a more appropriate description than its analogue constrained by second-order diffusion equations, the significantly increased computational and storage cost poses a severe computational challenge to the new model. More precisely, within each outer loop in the gradient projection method, the computational cost of the space-fractional state and costate equations is of order $O(MN^3)$, which is a very sharp increase of the computational cost of second-order state and costate equations that has a computational cost of $O(MN)$. This significantly increased computational cost gets even worse as the state and costate equations have to be solved iteratively in the gradient projection method. Therefore, the development of a fast and faithful numerical method is of fundamental importance.

The same idea can be applied to other schemes in the previous section. The Meerschaert scheme (15) for the costate equation can be expressed in the matrix form

$$\left(\mathbf{D}^{k-1} + \gamma \mathbf{A}^T\right) \mathbf{p}^{k-1} = \mathbf{D}^k \mathbf{p}^k + \tau \left(\mathbf{y}^{k-1} - \tilde{\mathbf{y}}^{k-1}\right). \tag{23}$$

It was shown in [30,32] that the matrix \mathbf{G} is diagonal-dominant and positive-definite for $0 < \beta < 1$, and so does \mathbf{A} . Clearly the matrix \mathbf{A} is Toeplitz, and the coefficient matrices $\mathbf{D}^k + \gamma \mathbf{A}$, $\mathbf{D}^{k-1} + \gamma \mathbf{A}^T$ are of diagonal-plus-Toeplitz type. Thus, the memory requirements to store them are $O(N)$, instead of a traditional storage for them that uses $O(N^2)$ of storage. Furthermore, if $r = 1/2$, \mathbf{A} is symmetric and positive-definite, so the conjugate gradient can be used to solve the linear systems (18) and (23) iteratively. For $r \neq 1/2$, \mathbf{A} is nonsymmetric, the conjugate gradient squared type of methods can be employed to solve the systems.

Similarly, the CN-WSGD schemes (16) and (17) can be expressed in the matrix forms

$$\left(\mathbf{D}^{k-\frac{1}{2}} + \frac{1}{2} \gamma \mathbf{B}\right) \mathbf{y}^k = \left(\mathbf{D}^{k-\frac{1}{2}} - \frac{1}{2} \gamma \bar{\mathbf{B}}\right) \mathbf{y}^{k-1} + \tau \mathbf{D}^{k-\frac{1}{2}} \left(\mathbf{f}^{k-\frac{1}{2}} + \mathbf{u}^{k-\frac{1}{2}}\right) \tag{24}$$

and

$$\left(\mathbf{D}^{k-1} + \frac{1}{2} \gamma \mathbf{B}^T\right) \mathbf{p}^{k-1} = \left(\mathbf{D}^k - \frac{1}{2} \gamma \mathbf{B}^T\right) \mathbf{p}^k + \tau \left(\mathbf{y}^{k-\frac{1}{2}} - \tilde{\mathbf{y}}^{k-\frac{1}{2}}\right). \tag{25}$$

Here $\mathbf{B} = r \bar{\mathbf{G}} + (1 - r) \bar{\mathbf{G}}^T$ and $\bar{\mathbf{G}}$ is also given by the expression in (20) with g_i being replaced by $w_i^{(2-\beta)}$. $\mathbf{D}^{k-\frac{1}{2}}$, $\mathbf{f}^{k-\frac{1}{2}}$, $\mathbf{u}^{k-\frac{1}{2}}$, $\mathbf{y}^{k-\frac{1}{2}}$, and $\tilde{\mathbf{y}}^{k-\frac{1}{2}}$ are their corresponding average at time step t^{k-1} and t^k . The matrix $\bar{\mathbf{G}}$ was shown to be positive definite [28], and so is \mathbf{B} . Hence, the fast method described earlier in the subsection still applies.

4.2 A Fast Preconditioned Conjugate Gradient Algorithm

Although the fast Krylov subspace iterative solvers described in the previous subsection significantly reduced the computational cost to $O(N \log N)$ per iteration, large number of iterations might be needed if the condition number of the coefficient matrix is large. In addition, the round off errors accumulated due to large number of iterations might deteriorate the convergence behavior of the iterative solvers [30]. This motivates us to use preconditioned Krylov subspace solvers.

For $r = 1/2$, the coefficient matrices of the systems (18), (23), (24), and (25) are of diagonal-plus-symmetric and positive-definite Toeplitz type. We modify the Strang preconditioner as follows [27]: For the Toeplitz matrix \mathbf{T} given in the previous subsection, the diagonals s_j of the Strang circulant preconditioner $\mathbf{S}(\mathbf{T}) = (s_{i-j})_{0 \leq i, j \leq N}$ are given by

$$s_l = \begin{cases} t_l, & 0 \leq l \leq N/2, \\ t_{l-N}, & N/2 < l < N, \\ t_{l+N}, & -N < l < 0. \end{cases} \tag{26}$$

Then, the diagonal-plus-symmetric and positive-definite Toeplitz coefficient matrices in the systems in (18), (23), (24), and (25) are preconditioned by the following circulant preconditioner

$$\mathbf{M}^k = D^k \mathbf{I} + \gamma \mathbf{S}(\mathbf{T}), \tag{27}$$

where $D^k = \frac{1}{N} \sum_{i=1}^N (d^{-1})_i^k$ is the mean value of the diagonal elements of \mathbf{D}^k , \mathbf{I} is the identity matrix of order N , and $\mathbf{T} = \mathbf{A}$ for the systems (18) and (23) or $\mathbf{T} = \mathbf{B}$ for (24) and (25). We use the system (18) as an example to formulate the fast preconditioned conjugate gradient method as follows [2]:

For a given initial guess $\mathbf{y}^{(0)}$, compute
 $\mathbf{r}^{(0)} = \mathbf{D}^k \mathbf{y}^{k-1} + \tau \mathbf{D}^k (\mathbf{f}^k + \mathbf{u}^k) - (\mathbf{D}^k + \gamma \mathbf{A}) \mathbf{y}^{(0)}$
 Solve $\mathbf{M}^k \mathbf{q}^{(0)} = \mathbf{r}^{(0)}$
 $\mathbf{p}^{(0)} = \mathbf{q}^{(0)}$
 For $i = 1, 2, \dots$
 $\alpha_{i-1} = \frac{(\mathbf{r}^{(i-1)})^T \mathbf{p}^{(i-1)}}{(\mathbf{p}^{(i-1)})^T (\mathbf{D}^k + \gamma \mathbf{A}) \mathbf{p}^{(i-1)}}$
 $\mathbf{y}^{(i)} = \mathbf{y}^{(i-1)} + \alpha_{i-1} \mathbf{p}^{(i-1)}$
 $\mathbf{r}^{(i)} = \mathbf{r}^{(i-1)} - \alpha_{i-1} (\mathbf{D}^k + \gamma \mathbf{A}) \mathbf{p}^{(i-1)}$
 $\delta = \|\mathbf{r}^{(i)}\|$
 Check for convergence: if δ is small enough, exit
 Solve $\mathbf{M}^k \mathbf{q}^{(i)} = \mathbf{r}^{(i)}$
 $\beta_{i-1} = \frac{(\mathbf{q}^{(i)})^T \mathbf{r}^{(i)}}{(\mathbf{q}^{(i-1)})^T \mathbf{r}^{(i-1)}}$
 $\mathbf{p}^{(i)} = \mathbf{q}^{(i)} + \beta_{i-1} \mathbf{p}^{(i-1)}$
 end
 $\mathbf{y}^k = \mathbf{y}^{(i)}$

In the above algorithm, all steps need $O(N)$ operations except for two matrix-vector multiplications $\mathbf{A} \mathbf{y}^{(0)}$, $\mathbf{A} \mathbf{p}^{(i-1)}$, and the solution of the system $\mathbf{M}^k \mathbf{q}^{(i)} = \mathbf{r}^{(i)}$. The matrix-vector multiplications can be carried out in $O(N \log N)$ operations as shown in Sect. 4.1. The preconditioner \mathbf{M}^k is an $N \times N$ circulant matrix and so can be decomposed similarly as (22)

$$\mathbf{M}^k = \mathbf{F}_N^{-1} \text{diag}(\mathbf{F}_N \mathbf{s}) \mathbf{F}_N,$$

where \mathbf{s} is the first column vector of \mathbf{M}^k and \mathbf{F}_N is the $N \times N$ discrete Fourier transform matrix. It can be inverted in $O(N \log N)$ operations.

4.3 A Fast Preconditioned Conjugate Gradient Squared Algorithm

For $r \neq 1/2$, the stiffness matrices \mathbf{A} and \mathbf{B} in the systems (18), (23), (24), and (25) are positive-definite but nonsymmetric Toeplitz matrices. We again use the Meerschaert scheme (18) as an example to demonstrate the idea. We still use a circulant preconditioner of the structure $\mathbf{M}^k = D^k \mathbf{I} + \gamma \mathbf{S}(\mathbf{A})$. However, note that $\mathbf{S}(\mathbf{A})$ is different from that of the symmetric case. Of course, we can also replace $\mathbf{S}(\mathbf{A})$ by a T. Chan preconditioner [4]. Then a fast preconditioned conjugate gradient squared algorithm for (18) can be formulated as follows [2]:

For a given initial guess $\mathbf{y}^{(0)}$, compute
 $\mathbf{r}^{(0)} = \mathbf{D}^k \mathbf{y}^{k-1} + \tau \mathbf{D}^k (\mathbf{f}^k + \mathbf{u}^k) - (\mathbf{D}^k + \gamma \mathbf{A}) \mathbf{y}^{(0)}$
 Choose $\tilde{\mathbf{r}}$ (for example, $\tilde{\mathbf{r}} = \mathbf{r}^{(0)}$)
 for $i = 1, 2, \dots$
 $\rho_{i-1} = \tilde{\mathbf{r}}^T \mathbf{r}^{(i-1)}$
 if $\rho_{i-1} = 0$ method fails
 if $i = 1$
 $\mathbf{w}^{(1)} = \mathbf{r}^{(0)}$
 $\mathbf{e}^{(1)} = \mathbf{w}^{(1)}$
 else
 $\beta_{i-1} = \rho_{i-1} / \rho_{i-2}$
 $\mathbf{w}^{(i)} = \mathbf{r}^{(i-1)} + \beta_{i-1} \mathbf{q}^{(i-1)}$
 $\mathbf{e}^{(i)} = \mathbf{w}^{(i)} + \beta_{i-1} (\mathbf{q}^{(i-1)} + \beta_{i-1} \mathbf{e}^{(i-1)})$
 end if
 solve $\mathbf{M}^k \hat{\mathbf{e}} = \mathbf{e}^{(i)}$
 $\hat{\mathbf{v}} = (\mathbf{D}^k + \gamma \mathbf{A}) \hat{\mathbf{e}}$
 $\alpha_i = \rho_{i-1} / \tilde{\mathbf{r}}^T \hat{\mathbf{v}}$
 $\mathbf{q}^{(i)} = \mathbf{w}^{(i)} - \alpha_i \hat{\mathbf{v}}$
 solve $\mathbf{M}^k \hat{\mathbf{w}} = \mathbf{w}^{(i)} + \mathbf{q}^{(i)}$
 $\mathbf{y}^{(i)} = \mathbf{y}^{(i-1)} + \alpha_i \hat{\mathbf{w}}$
 $\hat{\mathbf{q}} = (\mathbf{D}^k + \gamma \mathbf{A}) \hat{\mathbf{w}}$
 $\mathbf{r}^{(i)} = \mathbf{r}^{(i-1)} - \alpha_i \hat{\mathbf{q}}$
 $\delta = \|\mathbf{r}^{(i)}\|$
 Check for convergence: if δ is small enough, exit
 end
 $\mathbf{y}^k = \mathbf{y}^{(i)}$

5 Numerical Experiments

In this section we carry out numerical experiments to investigate the performance of the fast gradient projection method. These methods were implemented using Compaq Visual Fortran 6.6 on a ThinkPad T410 Laptop.

5.1 An Optimal Control Problem with $r = 0.5$

We consider the optimal control problem (1) and (2) with the following data

$$\begin{aligned} \beta &= 0.2, \quad r = 0.5, \quad T = 1, \quad u_0 = 1, \\ \tilde{y} &= \frac{100x^3(1-x)^2 \sin(T-t)}{(1+xt)^2} + \frac{100x^2(1-x)^2 \cos(T-t)}{1+xt} \\ &\quad - \frac{\sin(T-t)}{\Gamma(1.2)} \left[(x^{0.2} + (1-x)^{0.2}) - 5(x^{1.2} + (1-x)^{1.2}) + \frac{50}{11}(x^{2.2} + (1-x)^{2.2}) \right], \\ f &= -\max \left\{ \frac{100x^2(1-x)^2 \sin(T-t)}{1+xt}, 1 \right\}, \\ d &= \frac{1+xt}{100}. \end{aligned} \tag{28}$$

The analytical solution of this problem is given by

$$\begin{aligned}
 y &= 0, \\
 p &= -x^2(1-x)^2 \sin(T-t), \\
 u &= \max \left\{ \frac{100x^2(1-x)^2 \sin(T-t)}{1+xt}, 1 \right\}.
 \end{aligned}
 \tag{29}$$

Since $r = 0.5$, the coefficient matrices for the states equations are symmetric. We solve the Meerschaert schemes (18) and (23) as well as the CN-WSGD systems (24) and (25) by Gaussian elimination (Gauss), the fast conjugate gradient (CG) method, and the fast preconditioned conjugate gradient (PCG) method. We present the l^2 errors of the approximations to y , p , and u , and the convergence rates of the PCG scheme in Tables 1 and 2, respectively. Since Gauss elimination consumes too much CPU time, we skipped the numerical solution by Gauss elimination for the finest mesh $N = 2^{10}$. We present the consumed CPU times of the two schemes in Table 3.

Table 1 Performance of the Meerschaert scheme solved by Gauss, the CG, and the PCG solvers for the example in Sect. 5.1

| $N = M$ | Gauss | CG | PCG | |
|----------|-------------------------|-------------------------|-------------------------|------------|
| | $\ u - u_h\ _{l^2}$ | $\ u - u_h\ _{l^2}$ | $\ u - u_h\ _{l^2}$ | Conv. rate |
| 2^5 | 8.7775×10^{-3} | 8.7775×10^{-3} | 8.7775×10^{-3} | |
| 2^6 | 4.3653×10^{-3} | 4.3653×10^{-3} | 4.3653×10^{-3} | 1.0077 |
| 2^7 | 2.1765×10^{-3} | 2.1766×10^{-3} | 2.1765×10^{-3} | 1.0040 |
| 2^8 | 1.0866×10^{-3} | 1.0866×10^{-3} | 1.0866×10^{-3} | 1.0022 |
| 2^9 | 5.4289×10^{-4} | 5.4289×10^{-4} | 5.4289×10^{-4} | 1.0011 |
| 2^{10} | N/A | 2.7135×10^{-4} | 2.7135×10^{-4} | 1.0005 |
| $N = M$ | $\ y - y_h\ _{l^2}$ | $\ y - y_h\ _{l^2}$ | $\ y - y_h\ _{l^2}$ | Conv. rate |
| 2^5 | 4.8164×10^{-3} | 4.8164×10^{-3} | 4.8164×10^{-3} | |
| 2^6 | 2.4410×10^{-3} | 2.4410×10^{-3} | 2.4410×10^{-3} | 0.9805 |
| 2^7 | 1.2294×10^{-3} | 1.2294×10^{-3} | 1.2294×10^{-3} | 0.9895 |
| 2^8 | 6.1668×10^{-4} | 6.1668×10^{-4} | 6.1668×10^{-4} | 0.9953 |
| 2^9 | 3.0889×10^{-4} | 3.0889×10^{-4} | 3.0889×10^{-4} | 0.9974 |
| 2^{10} | N/A | 1.5458×10^{-4} | 1.5458×10^{-4} | 0.9988 |
| $N = M$ | $\ p - p_h\ _{l^2}$ | $\ p - p_h\ _{l^2}$ | $\ p - p_h\ _{l^2}$ | Conv. rate |
| 2^5 | 1.0563×10^{-4} | 1.0563×10^{-4} | 1.0563×10^{-4} | |
| 2^6 | 5.2743×10^{-5} | 5.2743×10^{-5} | 5.2743×10^{-5} | 1.0020 |
| 2^7 | 2.6355×10^{-5} | 2.6355×10^{-5} | 2.6355×10^{-5} | 1.0009 |
| 2^8 | 1.3177×10^{-5} | 1.3177×10^{-5} | 1.3177×10^{-5} | 1.0001 |
| 2^9 | 6.5885×10^{-6} | 6.5884×10^{-6} | 6.5884×10^{-6} | 1.0000 |
| 2^{10} | N/A | 3.2942×10^{-6} | 3.2942×10^{-6} | 1.0000 |

Table 2 Performance of the CN-WSGD scheme solved by Gauss, the CG, and the PCG solvers for the example in Sect. 5.1

| $N = M$ | Gauss | CG | PCG | |
|----------|-------------------------|-------------------------|-------------------------|------------|
| | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | Conv. rate |
| 2^5 | 1.8638×10^{-4} | 1.8638×10^{-4} | 1.8638×10^{-4} | |
| 2^6 | 4.7115×10^{-5} | 4.7116×10^{-5} | 4.7116×10^{-5} | 1.9840 |
| 2^7 | 1.1823×10^{-5} | 1.1826×10^{-5} | 1.1825×10^{-5} | 1.9943 |
| 2^8 | 2.9603×10^{-6} | 2.9667×10^{-6} | 2.9647×10^{-6} | 1.9959 |
| 2^9 | 7.4318×10^{-7} | 7.5945×10^{-7} | 7.5488×10^{-7} | 1.9736 |
| 2^{10} | N/A | 2.5293×10^{-7} | 2.2979×10^{-7} | 1.7159 |
| $N = M$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | Conv. rate |
| 2^5 | 9.5396×10^{-5} | 9.5396×10^{-5} | 9.5396×10^{-5} | |
| 2^6 | 2.4422×10^{-5} | 2.4422×10^{-5} | 2.4422×10^{-5} | 1.9657 |
| 2^7 | 6.1533×10^{-6} | 6.1531×10^{-6} | 6.1531×10^{-6} | 1.9888 |
| 2^8 | 1.5344×10^{-6} | 1.5344×10^{-6} | 1.5343×10^{-6} | 2.0038 |
| 2^9 | 3.7645×10^{-7} | 3.7782×10^{-7} | 3.7709×10^{-7} | 2.0246 |
| 2^{10} | N/A | 9.7558×10^{-8} | 9.3049×10^{-8} | 2.0188 |
| $N = M$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | Conv. rate |
| 2^5 | 5.5368×10^{-6} | 5.5369×10^{-6} | 5.5369×10^{-6} | |
| 2^6 | 1.7756×10^{-6} | 1.7758×10^{-6} | 1.7758×10^{-6} | 1.6406 |
| 2^7 | 5.2820×10^{-7} | 5.2849×10^{-7} | 5.2845×10^{-7} | 1.7486 |
| 2^8 | 1.5095×10^{-7} | 1.5137×10^{-7} | 1.5130×10^{-7} | 1.8043 |
| 2^9 | 4.2087×10^{-8} | 4.2672×10^{-8} | 4.2596×10^{-8} | 1.8286 |
| 2^{10} | N/A | 1.2560×10^{-8} | 1.2312×10^{-8} | 1.7906 |

Table 3 The consumed CPU times of the two schemes solved by Gauss, the CG, and the PCG solvers for the example in Sect. 5.1

| $N = M$ | Euler-Grünwald | | | CN-WSGD | | |
|----------|----------------|--------|--------|------------|--------|--------|
| | Gauss | CG | PCG | Gauss | CG | PCG |
| 2^5 | 0.09s | 0.22s | 0.19s | 0.14s | 0.25s | 0.25s |
| 2^6 | 0.89s | 1.00s | 0.81s | 1.11s | 0.98s | 0.92s |
| 2^7 | 20s | 4.88s | 3.64s | 20s | 4.16s | 3.77s |
| 2^8 | 8m 18s | 23s | 14s | 9m7s | 18s | 15s |
| 2^9 | 2h 47m 50s | 1m 45s | 58s | 2h 53m 18s | 1m 19s | 1m 2s |
| 2^{10} | N/A | 8m 24s | 3m 55s | N/A | 6m 14s | 4m 17s |

We observe from Tables 1 and 2 that Gauss, the CG and the PCG solvers generate almost identical numerical solutions. Thus, we only compute the convergence rates of the PCG solutions as an indicator. The fast gradient projection method, as an iterated method for the

Table 4 Performance of the Meerschaert scheme solved by Gauss, the CGS, and the PCGS solvers for the example in Sect. 5.2

| $N = M$ | Gauss | CGS | PCGS | |
|----------|-------------------------|-------------------------|-------------------------|------------|
| | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | Conv. rate |
| 2^5 | 8.4267×10^{-3} | 8.4267×10^{-3} | 8.4267×10^{-3} | |
| 2^6 | 4.1978×10^{-3} | 4.1978×10^{-3} | 4.1978×10^{-3} | 1.0053 |
| 2^7 | 2.0950×10^{-3} | 2.0950×10^{-3} | 2.0950×10^{-3} | 1.0027 |
| 2^8 | 1.0464×10^{-3} | 1.0464×10^{-3} | 1.0464×10^{-3} | 1.0015 |
| 2^9 | 5.2296×10^{-4} | 5.2296×10^{-4} | 5.2296×10^{-4} | 1.0007 |
| 2^{10} | N/A | N/A | 2.6142×10^{-4} | 1.0003 |
| $N = M$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | Conv. rate |
| 2^5 | 4.6956×10^{-3} | 4.6956×10^{-3} | 4.6956×10^{-3} | |
| 2^6 | 2.3799×10^{-3} | 2.3799×10^{-3} | 2.3799×10^{-3} | 0.9804 |
| 2^7 | 1.1986×10^{-3} | 1.1986×10^{-3} | 1.1986×10^{-3} | 0.9896 |
| 2^8 | 6.0125×10^{-4} | 6.0125×10^{-4} | 6.0125×10^{-4} | 0.9953 |
| 2^9 | 3.0117×10^{-4} | 3.0117×10^{-4} | 3.0117×10^{-4} | 0.9974 |
| 2^{10} | N/A | N/A | 1.5071×10^{-4} | 0.9988 |
| $N = M$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | Conv. rate |
| 2^5 | 1.0103×10^{-4} | 1.0103×10^{-4} | 1.0103×10^{-4} | |
| 2^6 | 5.0549×10^{-5} | 5.0549×10^{-5} | 5.0549×10^{-5} | 0.9990 |
| 2^7 | 2.5288×10^{-5} | 2.5288×10^{-5} | 2.5288×10^{-5} | 0.9992 |
| 2^8 | 1.2651×10^{-5} | 1.2651×10^{-5} | 1.2651×10^{-5} | 0.9992 |
| 2^9 | 6.3275×10^{-6} | 6.3275×10^{-6} | 6.3275×10^{-6} | 0.9996 |
| 2^{10} | N/A | N/A | 3.1644×10^{-6} | 0.9997 |

coupled nonlinear system, retains the first order convergence rate of Meerschaert scheme and the second order convergence rate of the CN-WSGD scheme, respectively. These results, together with the results in Table 3, show the utility of the fast preconditioned conjugate gradient method.

5.2 An Optimal Control Problem with $r = 0.8$

We consider the same example as in Sect. 5.1 except that $r = 0.8$ and

$$\begin{aligned}
 \tilde{y} = & \frac{100x^3(1-x)^2 \sin(T-t)}{(1+xt)^2} + \frac{100x^2(1-x)^2 \cos(T-t)}{1+xt} \\
 & - \frac{2 \sin(T-t)}{5\Gamma(1.2)} \left[(x^{0.2} + 4(1-x)^{0.2}) \right. \\
 & \left. - 5(x^{1.2} + 4(1-x)^{1.2}) + \frac{50}{11}(x^{2.2} + 4(1-x)^{2.2}) \right]. \tag{30}
 \end{aligned}$$

The true solution of this problems is the same as the one in Sect. 5.1.

Table 5 Performance of the CN-WSGD scheme solved by Gauss, the CGS, and the PCGS solvers for the example in Sect. 5.2

| $N = M$ | Gauss | CGS | PCGS | |
|----------|-------------------------|-------------------------|-------------------------|------------|
| | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | Conv. rate |
| 2^5 | 1.9119×10^{-4} | 1.9119×10^{-4} | 1.9119×10^{-4} | |
| 2^6 | 4.8584×10^{-5} | 4.8584×10^{-5} | 4.8578×10^{-5} | 1.9766 |
| 2^7 | 1.2228×10^{-5} | 1.2228×10^{-5} | 1.2219×10^{-5} | 1.9912 |
| 2^8 | 3.0667×10^{-6} | 3.0669×10^{-6} | 3.0558×10^{-6} | 1.9995 |
| 2^9 | 7.7081×10^{-7} | 7.7084×10^{-7} | 7.6673×10^{-7} | 1.9948 |
| 2^{10} | N/A | 1.9603×10^{-7} | 1.9599×10^{-7} | 1.9679 |
| $N = M$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | Conv. rate |
| 2^5 | 9.5732×10^{-5} | 9.5732×10^{-5} | 9.5730×10^{-5} | |
| 2^6 | 2.4534×10^{-5} | 2.4534×10^{-5} | 2.4530×10^{-5} | 1.9644 |
| 2^7 | 6.1905×10^{-6} | 6.1905×10^{-6} | 6.1847×10^{-6} | 1.9878 |
| 2^8 | 1.5459×10^{-6} | 1.5456×10^{-6} | 1.5377×10^{-6} | 2.0079 |
| 2^9 | 3.7990×10^{-7} | 3.7983×10^{-7} | 3.7699×10^{-7} | 2.0282 |
| 2^{10} | N/A | 8.8457×10^{-8} | 8.8383×10^{-8} | 2.0927 |
| $N = M$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | Conv. rate |
| 2^5 | 6.7384×10^{-6} | 6.7384×10^{-6} | 6.7385×10^{-6} | |
| 2^6 | 2.0756×10^{-6} | 2.0756×10^{-6} | 2.0758×10^{-6} | 1.6988 |
| 2^7 | 5.9744×10^{-7} | 5.9744×10^{-7} | 5.9767×10^{-7} | 1.7963 |
| 2^8 | 1.6612×10^{-7} | 1.6612×10^{-7} | 1.6639×10^{-7} | 1.8448 |
| 2^9 | 4.5253×10^{-8} | 4.5253×10^{-8} | 4.5386×10^{-8} | 1.8742 |
| 2^{10} | N/A | 1.2157×10^{-8} | 1.2178×10^{-8} | 1.8980 |

As $r = 0.8$, the stiffness matrices of the numerical schemes are nonsymmetric, so we solve the Meerschaert schemes (18) and (23) and the CN-WSGD schemes (24) and (25) by Gauss elimination, the fast conjugate gradient squared (CGS) method, and the fast preconditioned conjugate gradient squared (PCGS) method. We present the corresponding results in Tables 4, 5 and 6, respectively.

We observe roughly the same convergence behavior as those in Sect. 5.1. One important exception is that for the Meerschaert scheme with the finest mesh $M = N = 2^{10}$ the CGS solver diverged. As a matter of fact, the CGS solver does not converge within 4 hours of CPU time. In contrast, the PCGS solver still converged in 6 minutes and 15 seconds of CPU time. In other words, the PCGS solver not only reduced the CPU time of the CGS solver, but also improved its convergence behavior. Furthermore, the CGS solver converged for the CN-WSGD scheme at $M = N = 2^{10}$. This showed an additional computational benefit of the second-order CN-WSGD scheme over the first-order Meerschaert scheme.

Table 6 The consumed CPU times of the two schemes solved by Gauss, the CGS, and the PCGS solvers for the example in Sect. 5.2

| $N = M$ | Meerschaert scheme | | | CN-WSGD | | |
|----------|--------------------|-------|--------|-----------|---------|--------|
| | GAUSS | CGS | PCGS | GAUSS | CGS | PCGS |
| 2^5 | 0.09s | 0.28s | 0.23s | 0.13s | 0.30s | 0.27s |
| 2^6 | 0.91s | 1.34s | 0.98s | 1.09s | 1.30s | 1.05s |
| 2^7 | 20s | 6.91s | 4.53s | 21s | 5.84s | 4.25s |
| 2^8 | 8m 12s | 35s | 20s | 8m 17s | 27s | 17s |
| 2^9 | 2h 41m 23s | 3m 1s | 1m 29s | 2h 43m 2s | 2m 15s | 1m 13s |
| 2^{10} | N/A | N/A | 6m 15s | N/A | 11m 22s | 5m 51s |

Table 7 Performance of the Meerschaert scheme solved by Gauss, the CGS, and the PCGS solvers for the example in Sect. 5.3

| $N = M$ | GAUSS | CGS | PCGS | |
|----------|-------------------------|-------------------------|-------------------------|------------|
| | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | Conv. rate |
| 2^5 | 1.5656×10^{-2} | 1.5656×10^{-2} | 1.5656×10^{-2} | |
| 2^6 | 7.8740×10^{-3} | 7.8740×10^{-3} | 7.8740×10^{-3} | 0.9915 |
| 2^7 | 3.9488×10^{-3} | 3.9488×10^{-3} | 3.9488×10^{-3} | 0.9957 |
| 2^8 | 1.9774×10^{-3} | 1.9774×10^{-3} | 1.9774×10^{-3} | 0.9978 |
| 2^9 | 9.8945×10^{-4} | 9.8945×10^{-4} | 9.8945×10^{-4} | 0.9989 |
| 2^{10} | N/A | N/A | 4.9491×10^{-4} | 0.9995 |
| $N = M$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | Conv. rate |
| 2^5 | 7.9224×10^{-3} | 7.9224×10^{-3} | 7.9224×10^{-3} | |
| 2^6 | 4.0504×10^{-3} | 4.0504×10^{-3} | 4.0504×10^{-3} | 0.9679 |
| 2^7 | 2.0486×10^{-3} | 2.0486×10^{-3} | 2.0486×10^{-3} | 0.9834 |
| 2^8 | 1.0302×10^{-3} | 1.0302×10^{-3} | 1.0302×10^{-3} | 0.9917 |
| 2^9 | 5.1661×10^{-4} | 5.1661×10^{-4} | 5.1661×10^{-4} | 0.9958 |
| 2^{10} | N/A | N/A | 2.5869×10^{-4} | 0.9979 |
| $N = M$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | Conv. rate |
| 2^5 | 1.8107×10^{-4} | 1.8107×10^{-4} | 1.8107×10^{-4} | |
| 2^6 | 9.1388×10^{-5} | 9.1388×10^{-5} | 9.1388×10^{-5} | 0.9864 |
| 2^7 | 4.5912×10^{-5} | 4.5912×10^{-5} | 4.5912×10^{-5} | 0.9931 |
| 2^8 | 2.3012×10^{-5} | 2.3012×10^{-5} | 2.3012×10^{-5} | 0.9965 |
| 2^9 | 1.1520×10^{-5} | 1.1520×10^{-5} | 1.1520×10^{-5} | 0.9982 |
| 2^{10} | N/A | N/A | 5.7637×10^{-6} | 0.9991 |

Table 8 Performance of the CN-WSGD scheme solved by Gauss, the CGS, and the PCGS solvers for the example in Sect. 5.3

| $N = M$ | GAUSS | CGS | PCGS | |
|----------|-------------------------|-------------------------|-------------------------|------------|
| | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | $\ u - u_h\ _{l_2}$ | Conv. rate |
| 2^5 | 2.5443×10^{-4} | 2.5443×10^{-4} | 2.5444×10^{-4} | |
| 2^6 | 6.5173×10^{-5} | 6.5174×10^{-5} | 6.5179×10^{-5} | 1.9648 |
| 2^7 | 1.6528×10^{-5} | 1.6528×10^{-5} | 1.6531×10^{-5} | 1.9792 |
| 2^8 | 4.1939×10^{-6} | 4.1939×10^{-6} | 4.1948×10^{-6} | 1.9785 |
| 2^9 | 1.0650×10^{-6} | 1.0651×10^{-6} | 1.0657×10^{-6} | 1.9768 |
| 2^{10} | N/A | N/A | 2.7541×10^{-7} | 1.9521 |
| $N = M$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | $\ y - y_h\ _{l_2}$ | Conv. rate |
| 2^5 | 1.3542×10^{-4} | 1.3542×10^{-4} | 1.3542×10^{-4} | |
| 2^6 | 3.4955×10^{-5} | 3.4955×10^{-5} | 3.4957×10^{-5} | 1.9538 |
| 2^7 | 8.8796×10^{-6} | 8.8796×10^{-6} | 8.8813×10^{-6} | 1.9767 |
| 2^8 | 2.2334×10^{-6} | 2.2334×10^{-6} | 2.2338×10^{-6} | 1.9913 |
| 2^9 | 5.4987×10^{-7} | 5.4987×10^{-7} | 5.5013×10^{-7} | 2.0216 |
| 2^{10} | N/A | N/A | 1.2568×10^{-7} | 2.1300 |
| $N = M$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | $\ p - p_h\ _{l_2}$ | Conv. rate |
| 2^5 | 8.4539×10^{-6} | 8.4539×10^{-6} | 8.4540×10^{-6} | |
| 2^6 | 2.5212×10^{-6} | 2.5212×10^{-6} | 2.5213×10^{-6} | 1.7455 |
| 2^7 | 7.0685×10^{-7} | 7.0685×10^{-7} | 7.0693×10^{-7} | 1.8345 |
| 2^8 | 1.9205×10^{-7} | 1.9205×10^{-7} | 1.9207×10^{-7} | 1.8799 |
| 2^9 | 5.1264×10^{-8} | 5.1264×10^{-8} | 5.1277×10^{-8} | 1.9053 |
| 2^{10} | N/A | N/A | 1.3535×10^{-8} | 1.9216 |

Table 9 The consumed CPU times of the two schemes solved by Gauss, the CGS, and the PCGS solvers for the example in Sect. 5.3

| $N = M$ | Euler–Grünwald | | | CN-WSGD | | |
|----------|----------------|----------|----------|----------|---------|----------|
| | GAUSS | CGS | PCGS | GAUSS | CGS | PCGS |
| 2^5 | 0.09 s | 0.36 s | 0.28 s | 0.13 s | 0.36 s | 0.31 s |
| 2^6 | 0.97 s | 1.72 s | 1.17 s | 1.11 s | 1.56 s | 1.27 s |
| 2^7 | 21 s | 8.48 s | 5.14 s | 21 s | 7.23 s | 5.41 s |
| 2^8 | 8 m 27 s | 45 s | 22 s | 8 m 32 s | 37 s | 24 s |
| 2^9 | 2 h 51 m 41 s | 4 m 10 s | 1 m 45 s | 3 h 27 s | 3 m 7 s | 1 m 47 s |
| 2^{10} | N/A | N/A | 9 m 10 s | N/A | N/A | 8 m 19 s |

5.3 An Optimal Control Problem with $r = 1$

We consider the optimal control problem (1) and (2) with the following data

$$\begin{aligned}
 T &= 1, \quad u_0 = 1, \quad \beta = 0.2, \quad r = 1, \\
 \tilde{y} &= x^2(1-x)\sin(T-t) + \frac{100x^2(1-x)^2\sin(T-t)}{(1+xt)^2} \\
 &\quad + \frac{100x(1-x)^2\cos(T-t)}{1+xt} - \frac{\sin(T-t)}{\Gamma(1.2)} \left[2(1-x)^{0.2} - 5(1-x)^{1.2} \right], \\
 f &= -x^2(1-x)\cos(T-t) - \frac{(1+xt)\sin(T-t)}{100\Gamma(1.2)} \left[2x^{0.2} - 5x^{1.2} \right] \\
 &\quad - \max \left\{ \frac{100x(1-x)^2\sin(T-t)}{1+xt}, 1 \right\}, \\
 d &= \frac{1+xt}{100}.
 \end{aligned} \tag{31}$$

The true solution to the problem is

$$\begin{aligned}
 y &= x^2(1-x)\sin(T-t) \\
 p &= -x(1-x)^2\sin(T-t) \\
 u &= \max \left\{ \frac{100x(1-x)^2\sin(T-t)}{1+xt}, 1 \right\}.
 \end{aligned} \tag{32}$$

This example has even stronger nonsymmetry, as the containing fractional diffusion equations are one-sided equations. Namely, the state equation is a left-sided fractional diffusion equation while the costate equation is a right-sided equation. We solve the Meerschaert schemes (18) and (23) and the CN-WSGD schemes (24) and (25) by Gauss elimination, the CGS method, and the PCGS method. We present the corresponding results in Tables 7, 8 and 9, respectively.

We observe similar convergence behavior as in Sect. 5.2 except that the CGS solver diverged even for the CN-WSGD scheme at $M = N = 2^{10}$. In contrast, the PCGS method still converges for both the Meerschaert scheme and the CN-WSGD scheme. This again shows the utility of the PCGS method.

References

1. Agrawal, O.: A general formulation and solution scheme for fractional optimal control problems. *Non-linear Dyn.* **38**, 323–337 (2004)
2. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J.M., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., Van der Vorst, H.: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia (1994)
3. Benson, D., Wheatcraft, S.W., Meerschaert, M.M.: The fractional-order governing equation of Lévy motion. *Water Resour. Res.* **36**, 1413–1423 (2000)
4. Chan, T.: An optimal circulant preconditioner for Toeplitz systems. *SIAM J. Sci. Stat. Comput.* **9**, 766–771 (1988)
5. Chen, Y., Lu, Z.: Error estimates of fully discrete mixed finite element methods for semilinear quadratic parabolic optimal control problem. *Comput. Method. Appl. Mech. Eng.* **199**(23–24), 1415–1423 (2010)
6. Davis, P.J.: *Circulant Matrices*. American Mathematical Society, Providence (1979)
7. Dorville, R., Mophou, G.M., Valmorin, V.S.: Optimal control of a nonhomogeneous Dirichlet boundary fractional diffusion equation. *Comput. Math. Appl.* **62**(3), 1472–1481 (2011)

8. Du, N., Ge, L., Liu, W.: Adaptive finite element approximation for an elliptic optimal control problem with both pointwise and integral control constraints. *J. Sci. Comput.* **60**(1), 160–183 (2014)
9. Einstein, A.E.: *Investigations on the Theory of the Brownian Movement*, Translation. Dover, Mineola (1956)
10. Frederico, G., Torres, D.: Fractional optimal control in the sense of caputo and the fractional Noethers theorem. *Int. Math. Forum* **3**, 479–493 (2008)
11. Gray, R.M.: *Toeplitz and Circulant Matrices: A Review*. Now Publishers Inc, Hanover (2006)
12. Ito, K., Kunisch, K.: Augmented Lagrangian methods for nonsmooth convex optimization in Hilbert spaces. *Nonlinear Anal.* **41**, 573–589 (2000)
13. Ito, K., Kunisch, K.: The primal-dual active set method for nonlinear optimal control problems with bilateral constraints. *SIAM J. Control Optim.* **43**, 357–376 (2004)
14. Ito, K., Kunisch, K.: Semismooth Newton methods for time-optimal control for a class of ODEs. *SIAM J. Control Optim.* **48**, 3997–4013 (2010)
15. Ito, K., Kunisch, K.: Minimal effort problems and their treatment by semismooth Newton methods. *SIAM J. Control Optim.* **49**, 2083–2100 (2011)
16. Li, R., Liu, W., Ma, H., Tang, T.: Adaptive finite element approximation for distributed elliptic optimal control problems. *SIAM J. Control Optim.* **41**(5), 1321–1349 (2002)
17. Liu, W., Yan, N.: *Adaptive Finite Element Methods for Optimal Control Governed by PDEs: C Series in Information and Computational Science 41*. Science Press, Beijing (2008)
18. Meerschaert, M.M., Scheffler, H.P., Tadjeran, C.: Finite difference methods for two-dimensional fractional dispersion equation. *J. Comput. Phys.* **211**, 249–261 (2006)
19. Meerschaert, M.M., Tadjeran, C.: Finite difference approximations for two-sided space-fractional partial differential equations. *Appl. Numer. Math.* **56**, 80–90 (2006)
20. Metzler, R., Klafter, J.: The random walk’s guide to anomalous diffusion: a fractional dynamics approach. *Phys. Rep.* **339**, 1–77 (2000)
21. Neittaanmaki, P., Tiba, D., Dekker, M.: *Optimal Control of Nonlinear Parabolic Systems: Theory: Algorithms and Applications*. Marcel Dekker, New York (1994)
22. Niu, H., Yang, D.: Finite element analysis of optimal control problem governed by Stokes equations with L^2 -norm state-constraints. *J. Comput. Math.* **29**, 589–604 (2011)
23. Pironneau, O.: *Optimal Shape Design for Elliptic Systems*. Springer, Berlin (1984)
24. Podlubny, I.: *Fractional Differential Equations*. Academic Press, New York (1999)
25. Roos, H., Reibiger, C.: Numerical analysis of a system of singularly perturbed convection–diffusion equations related to optimal control. *Numer. Math. Theor. Meth. Appl.* **4**, 562–575 (2011)
26. Samko, S., Kilbas, A., Marichev, O.: *Fractional Integrals and Derivatives: Theory and Applications*. Gordon and Breach, London (1993)
27. Strang, G.: A proposal for Toeplitz matrix calculations. *Stud. Appl. Math.* **74**(2), 171–176 (1986)
28. Tian W., Zhou H., Deng W.: A class of second order difference approximations for solving space fractional diffusion equations. [arXiv:1201.5949](https://arxiv.org/abs/1201.5949) [math.NA]
29. Vallejos, M.: Multigrid methods for elliptic optimal control problems with pointwise state constraints. *Numer. Math. Theor. Meth. Appl.* **5**, 99–109 (2012)
30. Wang, H., Du, N.: A superfast-preconditioned iterative method for steady-state space-fractional diffusion equations. *J. Comput. Phys.* **240**, 49–57 (2013)
31. Wang, H., Du, N.: A fast finite difference method for three-dimensional time-dependent space-fractional diffusion equations and its efficient implementation. *J. Comput. Phys.* **253**, 50–63 (2013)
32. Wang, H., Wang, K., Sircar, T.: A direct $O(N \log^2 N)$ finite difference method for fractional diffusion equations. *J. Comput. Phys.* **229**, 8095–8104 (2010)
33. Wang, H., Yang, D.: Wellposedness of variable-coefficient conservative fractional elliptic differential equations. *SIAM J. Numer. Anal.* **51**, 1088–1107 (2013)