FAST SOLUTION METHODS FOR CONVEX QUADRATIC OPTIMIZATION OF FRACTIONAL DIFFERENTIAL EQUATIONS

Citation for published version:
Pougkakiotis, S, Pearson, JW, Leveque, S & Gondzio, J 2020, 'FAST SOLUTION METHODS FOR CONVEX QUADRATIC OPTIMIZATION OF FRACTIONAL DIFFERENTIAL EQUATIONS', *SIAM Journal on Matrix Analysis and Applications*, vol. 41, no. 3, pp. 1443–1476. https://doi.org/10.1137/19M128288X

Digital Object Identifier (DOI):
10.1137/19M128288X

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
SIAM Journal on Matrix Analysis and Applications

FAST SOLUTION METHODS FOR CONVEX QUADRATIC OPTIMIZATION OF FRACTIONAL DIFFERENTIAL EQUATIONS

SPYRIDON POUGKAKIOTIS*, JOHN W. PEARSON†, SANTOLO LEVEQUE‡, AND JACEK GONDZIO§

Abstract. In this paper, we present numerical methods suitable for solving convex quadratic Fractional Differential Equation (FDE) constrained optimization problems, with box constraints on the state and/or control variables. We develop an Alternating Direction Method of Multipliers (ADMM) framework, which uses preconditioned Krylov subspace solvers for the resulting sub-problems. The latter allows us to tackle a range of Partial Differential Equation (PDE) optimization problems with box constraints, posed on space-time domains, that were previously out of the reach of state-of-the-art preconditioners. In particular, by making use of the powerful Generalized Locally Toeplitz (GLT) sequences theory, we show that any existing GLT structure present in the problem matrices is preserved by ADMM, and we propose some preconditioning methodologies that could be used within the solver, to demonstrate the generality of the approach. Focussing on convex quadratic programs with time-dependent 2-dimensional FDE constraints, we derive multilevel circulant preconditioners, which may be embedded within Krylov subspace methods, for solving the ADMM sub-problems. Discretized versions of FDEs involve large dense linear systems. In order to overcome this difficulty, we design a recursive linear algebra, which is based on the Fast Fourier Transform (FFT). We manage to keep the storage requirements linear, with respect to the grid size N, while ensuring an order $N \log N$ computational complexity per iteration of the Krylov solver. We implement the proposed method, and demonstrate its scalability, generality, and efficiency, through a series of experiments over different setups of the FDE optimization problem.

1. Introduction. Optimization problems with Differential Equations (Partial (PDEs) or Ordinary (ODEs)) as constraints have received a great deal of attention within the applied mathematics and engineering communities, due in particular to their wide applicability across many fields of science. In addition to classical differential equation constraints, one may also use Fractional Differential Equations (FDEs) in order to model processes that could not otherwise be modeled using integer derivatives. In fact, there is a wide and increasing use of FDEs in the literature. Among other processes, FDEs have been used to model viscoelasticity (e.g. [43]), chaotic systems (e.g. [76]), turbulent flow, or anomalous diffusion (e.g. [7]). In particular, since the fractional operator is *non-local*, problems with non-local properties can frequently be modeled accurately using FDEs (see [62] for an extended review).

Availability of closed form solutions for FDEs is rare, and hence various numerical schemes for solving them have been developed and analyzed in the literature (see [50, 51, 52] for finite difference, and [19, 31] for finite element methods). Such numerical schemes produce dense matrices, making the solution or even the storage of FDE-constrained optimization problems extremely difficult for fine grids. Naturally, this behavior is even more severe in the case of multidimensional FDEs. In light of the previous, employing standard (black-box) direct approaches for solving such problems requires $O(N^3)$ operations and $O(N^2)$ storage, where N is the number of grid points. Iterative methods with general purpose preconditioners also suffer from similar issues.

Various specialized solution methods have been proposed in the literature, aiming at lowering the computational and storage cost of solving such problems (see for instance [17, 23, 24, 40, 53, 77]). One popular and effective approach is to employ tensor product solvers. Such specialized methods have been proposed for solving

*School of Mathematics, University of Edinburgh (S.Pougkakiotis@sms.ed.ac.uk).
†School of Mathematics, University of Edinburgh (J.Pearson@ed.ac.uk).
‡School of Mathematics, University of Edinburgh (S.Leveque@sms.ed.ac.uk).
§School of Mathematics, University of Edinburgh (J.Gondzio@ed.ac.uk).

high-dimensional FDE-constrained inverse problems with great success, even for very fine discretizations (see for example [23, 40] and the references therein). While these solvers are highly scalable (with respect to the grid size), to date they have been tailored solely to problems with specific cost functionals and without additional algebraic constraints. Another popular approach is based on the observation that multidimensional FDEs possess a *multilevel Toeplitz*-like structure. It is well known that such matrices can be very well approximated by banded multilevel Toeplitz (see for example [25, 53]) or *multilevel circulant* matrices (see [14, 15, 32, 44, 45]). The former are usually sparse and can be inverted using specially designed multigrid or factorization methods, while the latter can be inverted or applied to a vector in only $O(N \log N)$ operations using the Fast Fourier Transform (FFT) (e.g. [75]). The idea is to apply a Krylov subspace solver, supported by a banded Toeplitz or circulant preconditioner, in order to solve the optimality conditions of the problem. One is able to redesign the underlying linear algebra, in order to achieve an $O(N \log N)$ iteration complexity for the Krylov solver, with $O(N)$ overall storage requirements (see for example [44, 45, 46]). While such solution methods are certainly more general (although usually slower), when compared to tensor product solvers, they remain rather sensitive in terms of the underlying structure. In particular, to the authors' knowledge, no such method has been proposed for the solution of more general FDE optimization problems, for instance those which include box constraints on the state and control variables. We highlight that a time-independent problem, with box constraints on the control, is studied in [29], and the authors attempt to solve it using a Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L–BFGS) method.

In this paper, we present an optimization method suitable for solving convex quadratic PDE-constrained optimization problems with box constraints on the state and control variables. In particular, we assume that we are given an arbitrary PDE constrained inverse problem, an associated discretization method, and that the resulting sequences of discrete matrices belong to the class of Generalized Locally Toeplitz (GLT) sequences (we refer the reader to [34, 35, 70] for a comprehensive overview of the powerful GLT theory). Then, we propose the use of an Alternating Direction Method of Multipliers (ADMM) for solving the discretized optimization problems. We employ ADMM in order to separate the equality from the inequality constraints. As a consequence of this choice, we show that the linear systems required to be solved during the iterations of ADMM preserve the GLT structure of the initial problem matrices. Using this structure, we present and analyze some general methodologies for efficiently preconditioning such linear systems, and solving them using an appropriate Krylov subspace method. The Krylov subspace method is in turn, under certain mild assumptions, expected to converge in a number of iterations independent of the grid size. Subsequently, we focus on a certain class of convex quadratic optimization problems with FDE constraints. In particular, we consider time-dependent 2-dimensional FDEs, and we precondition the associated discretized matrices using multilevel circulant preconditioners. We manage to keep the storage requirements linear, with respect to the grid size N, while ensuring an order $N \log N$ computational complexity per iteration of the Krylov solver inside ADMM. We implement the proposed method, and demonstrate its robustness and efficiency, through a series of experiments over different setups of the FDE optimization problem.

This paper is structured as follows. In Section 2, we provide the relevant theoretical background as well as the notation used throughout the paper. Subsequently, in Section 3 we present the proposed ADMM framework, as well as possible preconditioning strategies that could be used to accelerate the solution of the ADMM sub-problems, given the assumption that the associated matrices possess a GLT structure. In Section 4 we present the FDE-constrained optimization problem under consideration. Then, in Section 5, we propose the use of a multilevel circulant preconditioner for approximating multilevel Toeplitz matrices arising from the discretization of the FDE under consideration, while demonstrating that such a preconditioner is effective for the problem at hand. In Section 6, we discuss the implementation details of the proposed approach and present some numerical results. Finally, in Section 7, we state our conclusions.

2. Notation and Theoretical Background. In this section, we introduce some notation and provide the theoretical background that will be used in the rest of this manuscript. Firstly, we introduce the notion of d-indices which will allow us to compactly represent multilevel matrices. For brevity of presentation, we only discuss the crucial notions that will be used in this paper. A more complete presentation of the notation and theory of this section can be found in [34, 35]. A reader familiar with the theory of GLT sequences can skip directly to Section 3.

DEFINITION 2.1. *A multi-index $\boldsymbol{i}$ of size d (d-index) is a row vector in $\mathbb{Z}^d$ with components $i_1, \ldots, i_d$. Using this notation, we define the following notions:*

- *$\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots,$ are the row vectors of all zeros, ones, twos, etc.*
- *$N(\boldsymbol{i}) = \prod_{j=1}^{d} i_j$ and we write $\boldsymbol{i} \to \infty$ to indicate that $\min(\boldsymbol{i}) \to \infty$.*
- *Given two d-indices $\boldsymbol{h}, \boldsymbol{k}$, we write $\boldsymbol{h} \leq \boldsymbol{k}$ to express that $h_j \leq k_j$, for all $j \in \{1, \ldots, d\}$. The d-index range $\boldsymbol{h}, \ldots, \boldsymbol{k}$ is a set of cardinality $N(\boldsymbol{k}-\boldsymbol{h}+\mathbf{1})$ given by $\{\boldsymbol{j} \in \mathbb{Z}^d : \boldsymbol{h} \leq \boldsymbol{j} \leq \boldsymbol{k}\}$. The latter set is assumed to be ordered under the lexicographical ordering, that is:*

$$\left[\ldots \left[[(j_1, \ldots, j_d)]_{j_d = h_d, \ldots, k_d} \right]_{j_{d-1} = h_{d-1}, \ldots, k_{d-1}} \ldots \right]_{j_1 = h_1, \ldots, k_1}.$$

- *Let a d-index $\boldsymbol{m} \in \mathbb{N}^d$, and define $\boldsymbol{x} = [x_{\boldsymbol{i}}]_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}}$ ($\boldsymbol{X} = [x_{\boldsymbol{i},\boldsymbol{j}}]_{\boldsymbol{i},\boldsymbol{j}=1}^{\boldsymbol{m}}$, respectively). Then $\boldsymbol{x}$ ($\boldsymbol{X}$, respectively) is a vector of size $N(\boldsymbol{m})$ (a matrix of size $N(\boldsymbol{m}) \times N(\boldsymbol{m})$, respectively).*
- *Any operation involving d-indices that has no meaning in the vector space $\mathbb{Z}^d$ will be interpreted in a componentwise sense.*

A matrix A of size N is a d-level matrix with level orders $n_1, \ldots, n_d$ if $N = n_1 n_2 \cdots n_d$ and it is partitioned into n_1^2 square blocks of size $\frac{N}{n_1}$, each of which is partitioned into n_2^2 blocks of size $\frac{N}{n_1 n_2}$, and so on until the last n_d^2 blocks of size 1. Then, A can be written as $A = [A_{\boldsymbol{i}\boldsymbol{j}}]_{\boldsymbol{i},\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{n}}$, where $A_{\boldsymbol{i}\boldsymbol{j}} = A_{i_1 j_1; \ldots; i_d j_d}$, for $\boldsymbol{i}, \boldsymbol{j} = \mathbf{1}, \ldots, \boldsymbol{n}$.

Next, we define the notion of a matrix-sequence, which is a fundamental element for studying the asymptotic spectral behavior of structured matrices arising from some discretization of a physical process. In the rest of this manuscript, given an arbitrary matrix A, $\sigma(A)$ denotes the set of singular values of the matrix, while $\lambda(A)$ denotes the set of eigenvalues of the matrix A (given that they exist).

DEFINITION 2.2. *A matrix-sequence is a sequence of the form $\{A_n\}_n$, where n varies over some infinite subset of $\mathbb{N}$, A_n is a square matrix of size d_n, and $d_n \to \infty$ as $n \to \infty$. In particular, a d-level matrix sequence is a sequence of the form $\{A_{\boldsymbol{n}}\}_n$, where $A_{\boldsymbol{n}}$ is a matrix of size $N(\boldsymbol{n}) \times N(\boldsymbol{n})$, n varies over some infinite subset of $\mathbb{N}$, and $\boldsymbol{n} = \boldsymbol{n}(n) \in \mathbb{N}^d$ is such that $\boldsymbol{n} \to \infty$, as $n \to \infty$. Given a d-level matrix-sequence $\{A_{\boldsymbol{n}}\}_n$, we say that it is sparsely unbounded (and denote that as s.u.) if:*

$$\lim_{M \to \infty} \limsup_{n \to \infty} \frac{\#\{i \in \{1, \ldots, N(\boldsymbol{n})\} : \sigma_i(A_{\boldsymbol{n}}) > M\}}{N(\boldsymbol{n})} = 0,$$

where $\#S$ denotes the cardinality of a set S. Similarly, we say that $\{A_{\boldsymbol{n}}\}_n$ is sparsely vanishing (and denote that as s.v.) if:

$$\lim_{M\to\infty}\limsup_{n\to\infty}\frac{\#\{i\in\{1,\ldots,N(\boldsymbol{n})\}:\sigma_i(A_{\boldsymbol{n}})<1/M\}}{N(\boldsymbol{n})}=0,$$

where we assume that $1/\infty=0$.

An important notion is that of clustering. In order to define it we let, for every $z\in\mathbb{C}$ and any $\epsilon>0$, $D(z,\epsilon)$ represent the disk with center z and radius ϵ. If $S\subseteq\mathbb{C}$ and $\epsilon>0$, $D(S,\epsilon)$ denotes the ϵ-expansion of S, defined as $D(S,\epsilon)=\bigcup_{z\in S}D(z,\epsilon)$.

Definition 2.3. *Let $\{A_n\}_n$ be a sequence of matrices, with A_n of size $d_n\times d_n$, and let $S\subseteq\mathbb{C}$ be a non-empty subset of $\mathbb{C}$. We say that $\{A_n\}_n$ is strongly clustered at S (in the sense of eigenvalues) if $\forall\ \epsilon>0$ we have:*

$$\#\{j\in\{1,\ldots,d_n\}:\lambda_j(A_n)\notin D(S,\epsilon)\}=O(1),$$

and weakly clustered at S if $\forall\ \epsilon>0$,

$$\#\{j\in\{1,\ldots,d_n\}:\lambda_j(A_n)\notin D(S,\epsilon)\}=o(d_n).$$

Clustering in the sense of singular values is defined analogously.

Let $f_m,\ f:D\subseteq\mathbb{R}^d\mapsto\mathbb{C}$ be measurable functions, with respect to the Lebesgue measure μ_d in $\mathbb{R}^d$. We say that $f_m\to f$ in measure if, for every $\epsilon>0$, $\lim_{m\to\infty}\mu_d\big(\{|f_m-f|>\epsilon\}\big)=0$. Furthermore, $f_m\to f$ a.e. (almost everywhere) if $\mu_d\big(\{f_m\nrightarrow f\}\big)=0$.

Lemma 2.4. *Let $f_m,\ g_m,\ f,\ g:D\subseteq\mathbb{R}^d\mapsto\mathbb{C}$ be measurable functions.*

1. *If $f_m\to f$ in measure, then $|f_m|\to|f|$ in measure.*
2. *If $f_m\to f$ in measure and $g_m\to g$ in measure, then $\alpha f_m+\beta g_m\to\alpha f+\beta g$ in measure for all $\alpha,\ \beta\in\mathbb{C}$.*
3. *If $f_m\to f$ in measure, $g_m\to g$ in measure, and $\mu_d(D)<\infty$, then $f_mg_m\to fg$ in measure.*

Proof. This is stated in [34, Lemma 2.3] and proved in [4, Corollary 2.2.6]. ∎

Let $C_c(\mathbb{C})$ ($C_c(\mathbb{R})$, respectively) be the space of complex-(real-)valued continuous functions defined on $\mathbb{C}$ (or $\mathbb{R}$) with compact support. Given a field $\mathbb{K}$ ($=\mathbb{C}$ or $\mathbb{R}$) and a measurable function $g:D\subset\mathbb{R}^d\mapsto\mathbb{K}$, with $0<\mu_d(D)<\infty$, define the functional:

$$\phi_g:C_c(\mathbb{K})\mapsto\mathbb{C},\quad \phi_g(F)=\frac{1}{\mu_d(D)}\int_D F(g(\boldsymbol{x}))\ \mathrm{d}\boldsymbol{x}.$$

Definition 2.5. *Let $\{A_n\}_n$ be a matrix-sequence, with A_n of size $d_n\times d_n$. We say that $\{A_n\}_n$ has an asymptotic eigenvalue (spectral) distribution described by a functional $\phi:C_c(\mathbb{C})\mapsto\mathbb{C}$, and we write $\{A_n\}_n\sim_\lambda\phi$, if:*

$$\lim_{n\to\infty}\frac{1}{d_n}\sum_{j=1}^{d_n}F(\lambda_j(A_n))=\phi(F),\quad\forall\ F\in C_c(\mathbb{C}).$$

If $\phi=\phi_f$ for some measurable function $f:D\subset\mathbb{R}^d\mapsto\mathbb{C}$, where $0<\mu_d(D)<\infty$, we say that $\{A_n\}_n$ has an asymptotic spectral distribution described by f and we write $\{A_n\}_n\sim_\lambda f$. Then, f is referred to as the eigenvalue (spectral) symbol of $\{A_n\}_n$.

We can define the asymptotic singular value distribution of a matrix sequence similarly to Definition 2.5 (see [33, Definition 2.1]). In that case, we write $\{A_n\}_n \sim_\sigma f$.

Below we define two important classes of matrix sequences, namely diagonal sampling and Toeplitz matrix-sequences.

DEFINITION 2.6. *Let a function $v : [0,1]^d \mapsto \mathbb{C}$ be given. The $\boldsymbol{n}$-th diagonal sampling matrix generated by v is denoted by $D_{\boldsymbol{n}}(v)$ and is defined by the following $N(\boldsymbol{n}) \times N(\boldsymbol{n})$ diagonal matrix:*

$$D_{\boldsymbol{n}}(v) = \operatorname*{diag}_{\boldsymbol{i}=\boldsymbol{1},\dots,\boldsymbol{n}} v\left(\frac{\boldsymbol{i}}{\boldsymbol{n}}\right).$$

DEFINITION 2.7. *Given a d-index $\boldsymbol{n} \in \mathbb{N}^d$, a matrix of the form $[a_{\boldsymbol{i}-\boldsymbol{j}}]_{\boldsymbol{i},\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{n}} \in \mathbb{C}^{N(\boldsymbol{n})\times N(\boldsymbol{n})}$ is called a d-level Toeplitz matrix. Unilevel Toeplitz matrices ($d = 1$) are also defined as matrices that are constant along all of their diagonals.*

A characterization of Toeplitz matrix-sequences is given by the following Theorem, the proof of which can be found in [35, Sections 3.1, 3.5]. Before that, let us define a useful matrix. Given an arbitrary $n \in \mathbb{N}$ and $k \in \mathbb{Z}$, define the $n \times n$ matrix $J_n^{(k)}$ such that $[J_n^{(k)}]_{ij} = 1$ if $i - j = k$ and $[J_n^{(k)}]_{ij} = 0$ otherwise. Given two d-indices $\boldsymbol{n} \in \mathbb{N}^d$ and $\boldsymbol{k} \in \mathbb{Z}^d$, we define $J_{\boldsymbol{n}}^{(\boldsymbol{k})} = J_{n_1}^{(k_1)} \otimes J_{n_2}^{(k_2)} \otimes \cdots \otimes J_{n_d}^{(k_d)}$, where $\otimes$ denotes the Kronecker product between two matrices.

THEOREM 2.8. *Let a function $f : [-\pi,\pi]^d \mapsto \mathbb{C}$ belonging to $L^1([-\pi,\pi]^d)$ be given, with Fourier coefficients denoted by:*

$$f_{\boldsymbol{k}} = \frac{1}{(2\pi)^d} \int_{[-\pi,\pi]^d} f(\boldsymbol{\theta}) e^{-i\langle \boldsymbol{k},\boldsymbol{\theta}\rangle}\, \mathrm{d}\boldsymbol{\theta}, \quad \boldsymbol{k} \in \mathbb{Z}^d,$$

where $\langle \boldsymbol{k},\boldsymbol{\theta}\rangle = \sum_{i=1}^d k_i\theta_i$. The $\boldsymbol{n}$-th (d-level) Toeplitz matrix associated with f is defined as:

$$T_{\boldsymbol{n}}(f) = [f_{\boldsymbol{i}-\boldsymbol{j}}]_{\boldsymbol{i},\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{n}} = \sum_{\boldsymbol{k}=-(\boldsymbol{n}-\boldsymbol{1})}^{\boldsymbol{n}-\boldsymbol{1}} f_{\boldsymbol{k}} J_{\boldsymbol{n}}^{(\boldsymbol{k})}.$$

Every d-level matrix sequence of the form $\{T_{\boldsymbol{n}}(f)\}_n$, with $\{\boldsymbol{n} = \boldsymbol{n}(n)\}_n \subseteq \mathbb{N}^d$ such that $\boldsymbol{n} \to \infty$ as $n \to \infty$, is called a (d-level) Toeplitz sequence generated by f, which in turn is referred to as the generating function of $\{T_{\boldsymbol{n}}(f)\}_n$. Furthermore, $\{T_{\boldsymbol{n}}(f)\}_n \sim_\sigma f$. If moreover f is real, then $\{T_{\boldsymbol{n}}(f)\}_n \sim_\lambda f$.

A special type of Toeplitz matrices are the circulant matrices, as defined below.

DEFINITION 2.9. *A matrix of the form $\left[a_{(\boldsymbol{i}-\boldsymbol{j}) \bmod \boldsymbol{n}}\right]_{\boldsymbol{i},\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{n}} \in \mathbb{C}^{N(\boldsymbol{n})\times N(\boldsymbol{n})}$, for some d-index $\boldsymbol{n} \in \mathbb{N}^d$, is called a multilevel (d-level) circulant matrix.*

Given an arbitrary $n \in \mathbb{N}$, define the $n \times n$ matrix C_n such that $[C_n]_{ij} = 1$ if $(i - j) \bmod n = 1$ and $[C_n]_{ij} = 0$ otherwise. Then, for $\boldsymbol{n} \in \mathbb{N}^d$ and $\boldsymbol{k} \in \mathbb{Z}^d$, let $C_{\boldsymbol{n}}^{\boldsymbol{k}} = C_{n_1}^{k_1} \otimes C_{n_2}^{k_2} \otimes \ldots \otimes C_{n_d}^{k_d}$, where $C_{n_i}^{k_i}$ is the previously defined matrix C_n raised to the power k_i. Let F_n denote the unitary discrete Fourier transform of order n. For any $\boldsymbol{n} \in \mathbb{N}^d$ let $F_{\boldsymbol{n}} = F_{n_1} \otimes \ldots \otimes F_{n_d}$. Below we provide a Theorem characterizing multilevel circulant matrices; its proof can be found in [35, Section 3.4].

THEOREM 2.10. *The d-level circulant matrix admits the following expression:*

$$\left[a_{(\boldsymbol{i}-\boldsymbol{j}) \bmod \boldsymbol{n}}\right]_{\boldsymbol{i},\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{n}} = \sum_{\boldsymbol{k}=\boldsymbol{0}}^{\boldsymbol{n}-\boldsymbol{1}} a_{\boldsymbol{k}} C_{\boldsymbol{n}}^{\boldsymbol{k}},$$

where $C_{\boldsymbol{n}}^{\boldsymbol{k}}$ is as defined earlier. Furthermore, letting any $\boldsymbol{r} \in \mathbb{N}^d$ and $c_{-\boldsymbol{r}}, \ldots, c_{\boldsymbol{r}} \in \mathbb{C}$, we have that any linear combination of the form $\sum_{\boldsymbol{d}=-\boldsymbol{r}}^{\boldsymbol{r}} c_{\boldsymbol{k}} C_{\boldsymbol{n}}^{\boldsymbol{k}}$ is a d-level circulant matrix. Then,

$$\sum_{\boldsymbol{k}=-\boldsymbol{r}}^{\boldsymbol{r}} c_{\boldsymbol{k}} C_{\boldsymbol{n}}^{\boldsymbol{k}} = F_{\boldsymbol{n}}^* \left(\operatorname*{diag}_{\boldsymbol{j}=\boldsymbol{0},\ldots,\boldsymbol{n}-\boldsymbol{1}} c\left(\frac{2\pi \boldsymbol{j}}{\boldsymbol{n}}\right)\right) F_{\boldsymbol{n}},$$

where $c(\boldsymbol{\theta}) = \sum_{\boldsymbol{k}=-\boldsymbol{r}}^{\boldsymbol{r}} c_{\boldsymbol{k}} e^{i\langle \boldsymbol{k},\boldsymbol{\theta}\rangle}$, and $F_{\boldsymbol{n}}$ is the multilevel discrete Fourier transform. Moreover, $\sum_{\boldsymbol{k}=-\boldsymbol{r}}^{\boldsymbol{r}} c_{\boldsymbol{k}} C_{\boldsymbol{n}}^{\boldsymbol{k}}$ is a normal matrix the spectrum of which is given by:

$$\lambda\left(\sum_{\boldsymbol{k}=-\boldsymbol{r}}^{\boldsymbol{r}} c_{\boldsymbol{k}} C_{\boldsymbol{n}}^{\boldsymbol{k}} \right) = \left\{ c\left(\frac{2\pi \boldsymbol{j}}{\boldsymbol{n}}\right) : \boldsymbol{j} = \boldsymbol{0}, \ldots, \boldsymbol{n}-\boldsymbol{1} \right\}.$$

Let $\mathcal{C}_{\boldsymbol{n}}$ be the set of all d-level circulant matrices of size $N(\boldsymbol{n}) \times N(\boldsymbol{n})$. In light of Theorem 2.10 we can see that the set $\mathcal{C}_{\boldsymbol{n}}$, together with matrix addition and multiplication, is a commutative ring. For more about circulant matrices see [18].

A very important notion of the theory of GLT sequences is that of the approximating class of sequences, which will be denoted as a.c.s.. In particular, it is very common in practice to approximate a "difficult" matrix-sequence by an "easier" sequence of matrix-sequences, which has the same asymptotic singular value or eigenvalue distribution. For example, such an "easier" sequence can be used to construct effective preconditioners inside a suitable Krylov subspace method. For the rest of this manuscript, given a matrix X, we denote its spectral norm by $\|X\|$.

Definition 2.11. *Let $\{A_n\}_n$ be a matrix-sequence, with A_n of size $d_n \times d_n$, and let $\{\{B_{n,m}\}_n\}_m$ be a sequence of matrix-sequences, with $B_{n,m}$ of size $d_n \times d_n$. We say that $\{\{B_{n,m}\}_n\}_m$ is an approximating class of sequences (a.c.s.) for $\{A_n\}_n$ if for every m, there exists n_m such that, for all $n \geq n_m$, we can write:*

$$A_n = B_{n,m} + R_{n,m} + N_{n,m}, \quad \operatorname{rank}(R_{n,m}) \leq c(m) d_n, \quad \|N_{n,m}\| \leq \omega(m),$$

where n_m, $c(m)$, and $\omega(m)$ depend only on m, and are such that:

$$\lim_{m\to\infty} c(m) = \lim_{m\to\infty} \omega_m = 0.$$

In that case, we write $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$.

Below, we provide a result, as reported in [35, Theorem 2.9], which will be very useful when constructing suitable preconditioners later in this paper.

Theorem 2.12. *Let two matrix-sequences $\{A_n\}_n$, $\{A'_n\}_n$ be given, with A_n, A'_n of size $d_n \times d_n$, and suppose that $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ and $\{B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A'_n\}_n$. The following properties hold:*

1. *$\{B^*_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A^*_n\}_n$.*
2. *$\{c_1 B_{n,m} + c_2 B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{c_1 A_n + c_2 A'_n\}_n$, for all c_1, $c_2 \in \mathbb{C}$.*
3. *If $\{A_n\}_n$ and $\{A'_n\}_n$ are s.u., then $\{B_{n,m}B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n A'_n\}_n$.*
4. *Suppose $\{A_n\}_n$ is s.v.. If $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ then $\{B^\dagger_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A^\dagger_n\}_n$.*

All the previous definitions are used to define the notion of Locally Toeplitz (LT) sequences, which in turn are generalized to define the notion of GLT sequences. We briefly define this class of matrix-sequences here, and refer the reader to [34, 35] for a complete derivation of this class, and a vast amount of results concerning sequences belonging in the GLT class. This theory was originally developed in [70].

DEFINITION 2.13. *Let $m,\ n \in \mathbb{N}$, $v : [0,1] \mapsto \mathbb{C}$, and $f \in L^1([-\pi,\pi])$. The 1-level locally Toeplitz operator is defined as the following $n \times n$ matrix:*

$$LT_n^m(v,f) = \Big(D_m(v) \otimes T_{\lfloor n/m \rfloor}(f)\Big) \oplus O_{n \bmod m},$$

where $D_m(v)$ is a diagonal sampling matrix generated by v, $T_{\lfloor n/m \rfloor}(f)$ a Toeplitz matrix generated by f, and $O_{n \bmod m}$ a zero matrix. Let also $\boldsymbol{m},\ \boldsymbol{n} \in \mathbb{N}^d$, $v : [0,1]^d \mapsto \mathbb{C}$, and $f \in L^1([-\pi,\pi]^d)$. The d-level locally Toeplitz operator is recursively defined as the following $N(\boldsymbol{n}) \times N(\boldsymbol{n})$ matrix:

$$LT_{\boldsymbol{n}}^{\boldsymbol{m}}(v, f_1 \otimes \ldots \otimes f_d) = LT_{n_1,\ldots,n_d}^{m_1,\ldots,m_d}\big(v(x_1,\ldots,x_d), f_1 \otimes \ldots \otimes f_d\big).$$

Definition 2.13 allows us to recall the notion of a multilevel locally Toeplitz sequence.

DEFINITION 2.14. *Let $\{A_{\boldsymbol{n}}\}_n$ be a d-level matrix-sequence, let $v : [0,1]^d \mapsto \mathbb{C}$ be Riemann-integrable and let $f \in L^1([-\pi,\pi]^d)$. We say that $\{A_{\boldsymbol{n}}\}_n$ is a (d-level) locally Toeplitz sequence with symbol $v \otimes f$, and we write $\{A_{\boldsymbol{n}}\}_n \sim_{LT} v \otimes f$, if:*

$$\{LT_{\boldsymbol{n}}^{\boldsymbol{m}}(v,f)\}_n \xrightarrow{\text{a.c.s.}} \{A_{\boldsymbol{n}}\}_n, \text{ as } \boldsymbol{m} \to \infty.$$

We are now able to define generalized locally Toeplitz sequences.

DEFINITION 2.15. *Let a d-level matrix-sequence $\{A_{\boldsymbol{n}}\}_n$, and a measurable function $\kappa : [0,1]^d \times [-\pi,\pi]^d \mapsto \mathbb{C}$ be given. Suppose that $\forall\ \epsilon > 0$ there exists a finite number of d-level LT sequences $\{A_{\boldsymbol{n}}^{(i,\epsilon)}\}_n \sim_{LT} v_{i,\epsilon} \otimes f_{i,\epsilon}$, $i = 1,\ldots,N_\epsilon$, such that as $\epsilon \to 0$:*

$$\sum_{i=1}^{N_\epsilon} v_{i,\epsilon} \otimes f_{i,\epsilon} \to \kappa \text{ in measure, and } \Big\{\sum_{i=1}^{N_\epsilon} A_{\boldsymbol{n}}^{(i,\epsilon)}\Big\}_n \xrightarrow{\text{a.c.s.}} \{A_{\boldsymbol{n}}\}_n.$$

Then $\{A_{\boldsymbol{n}}\}_n$ is a d-level GLT sequence with symbol κ, and we write $\{A_{\boldsymbol{n}}\}_n \sim_{GLT} \kappa$.

The GLT class contains a wide range of matrix-sequences arising from various discretization methods of numerous differential equations. In the following Theorem we present some important properties of GLT sequences that will be used later in this paper. This is only a subset of the properties of multilevel GLT sequences, and the reader is referred to [34, 35] for a detailed derivation of all the results presented in this section. Given a measurable function κ, we denote its complex conjugate by $\bar{\kappa}$.

THEOREM 2.16. *Let $\{A_{\boldsymbol{n}}\}_n$ and $\{B_{\boldsymbol{n}}\}_n$ be two d-level matrix-sequences and $\kappa,\ \xi : [0,1]^d \times [-\pi,\pi]^d \mapsto \mathbb{C}$ two measurable functions. Assume that $\{A_{\boldsymbol{n}}\}_n$ is a GLT sequence with symbol κ, while $\{B_{\boldsymbol{n}}\}_n$ a GLT sequence with symbol ξ. Then:*

1. *If $A_{\boldsymbol{n}}$ are Hermitian then $\{A_{\boldsymbol{n}}\}_n \sim_\lambda \kappa$.*
2. $\{A_{\boldsymbol{n}}^*\}_n \sim_{GLT} \bar{\kappa}$.
3. $\{c_1 A_{\boldsymbol{n}} + c_2 B_{\boldsymbol{n}}\}_n \sim_{GLT} c_1\kappa + c_2\xi$, for all $c_1,\ c_2 \in \mathbb{C}$.
4. $\{A_{\boldsymbol{n}} B_{\boldsymbol{n}}\}_n \sim_{GLT} \kappa\xi$.
5. *If $\kappa \neq 0$ almost everywhere, then $\{A_{\boldsymbol{n}}^\dagger\}_n \sim_{GLT} \kappa^{-1}$.*
6. *Let a sequence of d-level matrix-sequences $\{B_{\boldsymbol{n},m}\}_n \sim_{GLT} \kappa_m$. Then, we have that $\{B_{\boldsymbol{n},m}\}_n \xrightarrow{\text{a.c.s.}} \{A_{\boldsymbol{n}}\}_n$ if and only if $\kappa_m \to \kappa$ in measure.*
7. *If $\{A_{\boldsymbol{n}}\}_n \sim_{GLT} \kappa$ and each $A_{\boldsymbol{n}}$ is Hermitian, then $\{f(A_{\boldsymbol{n}})\}_n \sim_{GLT} f(\kappa)$ for every continuous function $f : \mathbb{C} \mapsto \mathbb{C}$.*

3. A Structure Preserving Method. In this section, we will derive an optimization method suitable for solving convex quadratic optimization problems, with linear constraints arising from the discretization of some continuous process. The assumption on the constraints is that the generated (multilevel) matrix-sequence is a GLT sequence. Let us consider the following generic Differential Equation (DE):

$$\mathrm{Dy}(\boldsymbol{x},t) = \mathrm{g}(\boldsymbol{x},t),$$

where D denotes some linear differential operator associated with the DE, $\boldsymbol{x}$ is a $(d-1)$-dimensional spatial variable and $t \geq 0$ is the time variable. Since analytical solutions are not readily available for various differential operators, we discretize the previous equation given an arbitrary numerical method, and instead solve a sequence of linear systems of the form:

$$\{D_{\boldsymbol{n}} y_{\boldsymbol{n}}\}_n = \{g_{\boldsymbol{n}}\}_n, \tag{3.1}$$

with size $d_n \times d_n$ and $d_n = N(\boldsymbol{n})$, such that $\boldsymbol{n} \to \infty$ as $n \to \infty$.

Concerning the objective of the studied model, we assume that it may be summarized by a convex functional $\mathrm{J}_1(\mathrm{y}(\boldsymbol{x},t))$. Usually, such a functional measures the misfit between the state $\mathrm{y}(\boldsymbol{x},t)$ and some desired state $\bar{\mathrm{y}}(\boldsymbol{x},t)$, and we will focus our attention on this class of (inverse) problems. In other words, we expect that the discretized version of this functional will be of the form $\frac{1}{2}(y-\bar{y})^* J_1 (y-\bar{y})$, with J_1 a symmetric positive (semi-)definite matrix. As is common in such problems, the linear systems in (3.1) usually admit more than one solution and hence a regularization functional is usually employed to guarantee that the chosen solution will have some desired properties, depending on the initial DE under consideration. In other words, we introduce a control variable $\mathrm{u}(\boldsymbol{x},t)$ which is linked to the state variable as follows:

$$\mathrm{Dy}(\boldsymbol{x},t) + \mathrm{u}(\boldsymbol{x},t) = \mathrm{g}(\boldsymbol{x},t).$$

The size of the control is measured using some convex functional $\mathrm{J}_2(\mathrm{u}(\boldsymbol{x},t))$.

Finally, we allow further restrictions on the state and control variables in the form of inequality constraints (which depend on the problem under consideration). By combining all the previous, we obtain the following generic model that is studied in this paper:

$$\begin{aligned} \min_{\mathrm{y},\mathrm{u}} \ & \mathrm{J}(\mathrm{y}(\boldsymbol{x},t),\mathrm{u}(\boldsymbol{x},t)) = \mathrm{J}_1(\mathrm{y}(\boldsymbol{x},t)) + \mathrm{J}_2(\mathrm{u}(\boldsymbol{x},t)) \\ \text{s.t.} \ & \mathrm{Dy}(\boldsymbol{x},t) + \mathrm{u}(\boldsymbol{x},t) = \mathrm{g}(\boldsymbol{x},t), \\ & \mathrm{y_a}(\boldsymbol{x},t) \leq \mathrm{y}(\boldsymbol{x},t) \leq \mathrm{y_b}(\boldsymbol{x},t), \quad \mathrm{u_a}(\boldsymbol{x},t) \leq \mathrm{u}(\boldsymbol{x},t) \leq \mathrm{u_b}(\boldsymbol{x},t). \end{aligned} \tag{3.2}$$

The problem is considered on a given compact space-time domain $\Omega \times (0,T)$, for some $T > 0$, where $\Omega \subset \mathbb{R}^{d-1}$ has boundary $\partial\Omega$. The algebraic inequality constraints are assumed to hold a.e. on $\Omega \times (0,T)$. We further note that the restrictions $\mathrm{y_a}$, $\mathrm{y_b}$, $\mathrm{u_a}$, and $\mathrm{u_b}$ may take the form of constants, or functions in spatial and/or temporal variables. The boundary conditions are not specified since they do not affect the analysis in this section. Notice that problem (3.2) includes the case of equality-constrained optimization, by allowing unbounded restriction functions.

We discretize problem (3.2), using an arbitrary numerical method, to find an approximate solution by solving a sequence of optimization problems of the form:

$$\begin{aligned} \min_{y_{\boldsymbol{n}}, u_{\boldsymbol{n}}} \quad & \left(\frac{1}{2}(y_{\boldsymbol{n}} - \bar{y}_{\boldsymbol{n}})^* J_{1_{\boldsymbol{n}}} (y_{\boldsymbol{n}} - \bar{y}_{\boldsymbol{n}}) + \frac{1}{2} u_{\boldsymbol{n}}^* J_{2_{\boldsymbol{n}}} u_{\boldsymbol{n}}\right) \\ \text{s.t.} \quad & D_{\boldsymbol{n}} y_{\boldsymbol{n}} + u_{\boldsymbol{n}} = g_{\boldsymbol{n}}, \\ & y_{a_{\boldsymbol{n}}} \le y_{\boldsymbol{n}} \le y_{b_{\boldsymbol{n}}}, \quad u_{a_{\boldsymbol{n}}} \le u_{\boldsymbol{n}} \le u_{b_{\boldsymbol{n}}}, \end{aligned} \tag{3.3}$$

in which the associated matrices are of size $d_n \times d_n$, where $d_n = N(\boldsymbol{n})$ and $d_n \to \infty$ as $n \to \infty$. Notice that we only assume $J_{1_{\boldsymbol{n}}}$ and $J_{2_{\boldsymbol{n}}}$ to be symmetric positive semi-definite. Hence, the presented methodology is applicable to a wide range of convex quadratic programming problems. An entry n_j of the multi-index $\boldsymbol{n}$ corresponds to the number of discretization points along dimension j, with $j \in \{1, \ldots, d\}$, where n_d corresponds to the time dimension. Below, we summarize our assumptions for the associated matrices in problem (3.3).

Assumption 1. *Given the sequence of problems in (3.3), we assume that:*

- *The sequence $\{D_{\boldsymbol{n}}\}_n$ is a d-level matrix sequence with spectral norm uniformly bounded with respect to n, i.e. there exists a constant C_D such that $\|D_{\boldsymbol{n}}\| \le C_D$ for all n. Furthermore, there exists a measurable function $\kappa : [0,1]^d \times [-\pi,\pi]^d \mapsto \mathbb{C}$, which is the symbol of $\{D_{\boldsymbol{n}}\}_n$, so that $\{D_{\boldsymbol{n}}\}_n \sim_{GLT} \kappa$.*
- *The sequences $\{J_{1_{\boldsymbol{n}}}\}_n$ and $\{J_{2_{\boldsymbol{n}}}\}_n$ are two d-level matrix sequences, with uniformly bounded spectral norms with respect to n. Furthermore, there exist two measurable functions $\xi_1,\ \xi_2 : [0,1]^d \times [-\pi,\pi]^d \mapsto \mathbb{R}$, such that $\xi_1 \ge 0$, $\xi_2 \ge 0$, $\{J_{1_{\boldsymbol{n}}}\}_n \sim_{GLT} \xi_1$, and $\{J_{2_{\boldsymbol{n}}}\}_n \sim_{GLT} \xi_2$.*

We note that a wide range of numerical discretizations of DEs satisfy this assumption (see [34, 35] for a plethora of applications). Notice also that the requirement that ξ_1 and ξ_2 are real and non-negative follows from the positive semi-definiteness of $J_{1_{\boldsymbol{n}}}$ and $J_{2_{\boldsymbol{n}}}$. Towards the end of this section we discuss how one could still apply the presented methodology successfully without requiring the GLT structure of the discretized objective function (i.e. by requiring only boundedness and convexity).

Before presenting the proposed optimization method for solving problems of the form (3.3), we note a negative result concerning a large class of optimization methods. More specifically, problems like (3.3) are often solved using an Interior Point Method (IPM), or some Active-Set (AS) type of method. However, such problems are usually highly structured, and this structure must be exploited, given that the problem size increases indefinitely as one refines the discretization. Obviously, any AS method would fail in maintaining the structure, as only a subset of the constraints of (3.3) is considered at each AS iteration and hence the structure of the AS sub-problems will be unknown. In fact, any optimization method whose sub-problems arise by projecting the variables of the problem in a subspace would face this issue.

On the other hand, IPMs deal with the inequality constraints by introducing logarithmic barriers in the objective (see for example [38]). Then, at every IPM iteration, one forms the optimality conditions of the barrier sub-problem, and approximately solves them using Newton's method. If we assume that there exists a symbol f which describes the asymptotic eigenvalue distribution of the sequence of Hessian matrices of the logarithmic barrier, we arrive at a contradiction. Indeed, the sequence of Hessian matrices arising from the logarithmic barriers introduced by IPM are not s.u.. This in turns contradicts the assumption that f is the symbol of this matrix sequence, since if an arbitrary matrix sequence is such that $\{L_n\}_n \sim_\sigma f$, for some measurable function f, then $\{L_n\}_n$ must be s.u. (see [34, Section 9–S1]). In particular, any GLT sequence is s.u., and hence the sequence of Hessian matrices of the logarithmic barrier functions cannot be a GLT sequence. As a consequence, the system matrix of the optimality conditions of each barrier sub-problem, within the IPM, will not be in the GLT class.

3.1. Alternating Direction Method of Multipliers. In order to overcome the previous issues, we propose the use of an alternating direction method of multipliers (see [5, Section 5] and the references therein), which separates the equality from the inequality constraints, thus allowing us to preserve the structure found in the matrices associated with (3.3). We should mention here that while ADMM allows us to retain the underlying structure of the problem, it comes at a cost. It is well-known (see e.g. [5]) that ADMM leads to relatively slow convergence and hence is not suitable for finding very accurate solutions. Nevertheless, a 4-digit accurate solution can generally be found in reasonable CPU time. Furthermore, the linear system solved at each ADMM iteration does not change, and hence, if a suitable preconditioner exploiting the problem structure is found, it only needs to be computed once. Finally, linear convergence can also be shown, under certain assumptions on the problem under consideration (such as strong convexity, see [20]).

We begin by rewriting problem (3.3), after introducing some auxiliary variables $z_{y_{\boldsymbol{n}}}, z_{u_{\boldsymbol{n}}}$ of size $N(\boldsymbol{n})$:

$$\begin{aligned} &\min_{y_{\boldsymbol{n}},u_{\boldsymbol{n}},z_{y_{\boldsymbol{n}}},z_{u_{\boldsymbol{n}}}} \left(\frac{1}{2}(y_{\boldsymbol{n}}-\bar{y}_{\boldsymbol{n}})^* J_{1_{\boldsymbol{n}}}(y_{\boldsymbol{n}}-\bar{y}_{\boldsymbol{n}}) + \frac{1}{2}u_{\boldsymbol{n}}^* J_{2_{\boldsymbol{n}}} u_{\boldsymbol{n}}\right) \\ &\text{s.t.} \quad D_{\boldsymbol{n}} y_{\boldsymbol{n}} + u_{\boldsymbol{n}} = g_{\boldsymbol{n}}, \\ &\qquad y_{\boldsymbol{n}} = z_{y_{\boldsymbol{n}}},\ u_{\boldsymbol{n}} = z_{u_{\boldsymbol{n}}}, \\ &\qquad y_{a_{\boldsymbol{n}}} \le z_{y_{\boldsymbol{n}}} \le y_{b_{\boldsymbol{n}}},\ u_{a_{\boldsymbol{n}}} \le z_{u_{\boldsymbol{n}}} \le u_{b_{\boldsymbol{n}}}. \end{aligned} \tag{3.4}$$

Next, we define the augmented Lagrangian function corresponding to (3.4):

$$\begin{aligned} \mathcal{L}_\delta(y_{\boldsymbol{n}}, u_{\boldsymbol{n}}, z_{y_{\boldsymbol{n}}}, z_{u_{\boldsymbol{n}}}, p_{\boldsymbol{n}}, w_{y_{\boldsymbol{n}}}, w_{u_{\boldsymbol{n}}}) = &\ \frac{1}{2}(y_{\boldsymbol{n}}-\bar{y}_{\boldsymbol{n}})^* J_{1_{\boldsymbol{n}}}(y_{\boldsymbol{n}}-\bar{y}_{\boldsymbol{n}}) + \frac{1}{2}u_{\boldsymbol{n}}^* J_{2_{\boldsymbol{n}}} u_{\boldsymbol{n}} \\ &+\ p_{\boldsymbol{n}}^*(D_{\boldsymbol{n}} y_{\boldsymbol{n}} + u_{\boldsymbol{n}} - g_{\boldsymbol{n}}) + w_{y_{\boldsymbol{n}}}^*(y_{\boldsymbol{n}} - z_{y_{\boldsymbol{n}}}) + w_{u_{\boldsymbol{n}}}^*(u_{\boldsymbol{n}} - z_{u_{\boldsymbol{n}}}) \\ &+\ \frac{1}{2\delta}\left(\|D_{\boldsymbol{n}} y_{\boldsymbol{n}} + u_{\boldsymbol{n}} - g_{\boldsymbol{n}}\|_2^2 + \|y_{\boldsymbol{n}} - z_{y_{\boldsymbol{n}}}\|_2^2 + \|u_{\boldsymbol{n}} - z_{u_{\boldsymbol{n}}}\|_2^2\right), \end{aligned} \tag{3.5}$$

where $p_{\boldsymbol{n}},\ w_{y_{\boldsymbol{n}}}$, and $w_{u_{\boldsymbol{n}}}$ are the dual variables corresponding to each of the equality constraints of (3.4). An ADMM applied to model (3.4) is given in Algorithm 3.1. We omit specific details of the algorithm. The reader is referred to [5] for a basic proof of convergence of Algorithm 3.1, as well as a detailed overview of ADMM. For a convergence proof for the case where complex variables and matrices are allowed, the reader is referred to [49]. Linear convergence of a generalization of this algorithm, under certain assumptions, can be found in [20] and the references therein. We should note that the step-length ρ in (3.6c) and (3.6d) plays an important role in the convergence behavior of ADMM, and in fact, convergence of Algorithm 3.1 is guaranteed for any $\rho \in (0, \frac{\sqrt{5}+1}{2})$ (see [36]).

One can easily observe that the most challenging step of Algorithm 3.1, is that of solving (3.6a). The optimality conditions of (3.6a), at iteration j, read as follows:

$$\begin{bmatrix} J_{1_{\boldsymbol{n}}} + \frac{1}{\delta}(D_{\boldsymbol{n}}^* D_{\boldsymbol{n}} + I_{\boldsymbol{n}}) & \frac{1}{\delta}D_{\boldsymbol{n}}^* \\ \frac{1}{\delta}D_{\boldsymbol{n}} & J_{2_{\boldsymbol{n}}} + \frac{2}{\delta}I_{\boldsymbol{n}} \end{bmatrix} \begin{bmatrix} y_{\boldsymbol{n}} \\ u_{\boldsymbol{n}} \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}, \tag{3.7}$$

Algorithm 3.1 (2-Block) Standard ADMM

Input: Let $y_{\boldsymbol{n}}^0, u_{\boldsymbol{n}}^0, z_{y_{\boldsymbol{n}}}^0, z_{u_{\boldsymbol{n}}}^0, p_{\boldsymbol{n}}^0, w_{y_{\boldsymbol{n}}}^0, w_{u_{\boldsymbol{n}}}^0 \in \mathbb{C}^{N(\boldsymbol{n})}$, $\delta > 0$, $\rho \in (0, \frac{\sqrt{5}+1}{2})$.

for $(j = 0, 1, \dots)$ **do**

$$(y_{\boldsymbol{n}}^{j+1}, u_{\boldsymbol{n}}^{j+1}) = \operatorname*{arg\,min}_{y_{\boldsymbol{n}}, u_{\boldsymbol{n}}} \left\{ \mathcal{L}_\delta(y_{\boldsymbol{n}}, u_{\boldsymbol{n}}, z_{y_{\boldsymbol{n}}}^j, z_{u_{\boldsymbol{n}}}^j, p_{\boldsymbol{n}}^j, w_{y_{\boldsymbol{n}}}^j, w_{u_{\boldsymbol{n}}}^j) \right\} \tag{3.6a}$$

$$(z_{y_{\boldsymbol{n}}}^{j+1}, z_{u_{\boldsymbol{n}}}^{j+1}) = \operatorname*{arg\,min}_{z_y \in [y_a, y_b],\ z_u \in [u_a, u_b]} \left\{ \mathcal{L}_\delta(y_{\boldsymbol{n}}^{j+1}, u_{\boldsymbol{n}}^{j+1}, z_{y_{\boldsymbol{n}}}, z_{u_{\boldsymbol{n}}}, p_{\boldsymbol{n}}^j, w_{y_{\boldsymbol{n}}}^j, w_{u_{\boldsymbol{n}}}^j) \right\} \tag{3.6b}$$

$$p_{\boldsymbol{n}}^{j+1} = p_{\boldsymbol{n}}^j + \frac{\rho}{\delta}(D_{\boldsymbol{n}} y_{\boldsymbol{n}}^{j+1} + u_{\boldsymbol{n}}^{j+1} - g_{\boldsymbol{n}}) \tag{3.6c}$$

$$(w_{y_{\boldsymbol{n}}}^{j+1}, w_{u_{\boldsymbol{n}}}^{j+1}) = \left(w_{y_{\boldsymbol{n}}}^j + \frac{\rho}{\delta}(y_{\boldsymbol{n}}^{j+1} - z_{y_{\boldsymbol{n}}}^{j+1}), w_{u_{\boldsymbol{n}}}^j + \frac{\rho}{\delta}(u_{\boldsymbol{n}}^{j+1} - z_{u_{\boldsymbol{n}}}^{j+1}) \right) \tag{3.6d}$$

end for

where

$$\eta_1 = J_{1_{\boldsymbol{n}}} \bar{y}_{\boldsymbol{n}} - D_{\boldsymbol{n}}^* p_{\boldsymbol{n}}^j - w_{y_{\boldsymbol{n}}}^j + \frac{1}{\delta}(D_{\boldsymbol{n}}^* g_{\boldsymbol{n}} + z_{y_{\boldsymbol{n}}}^j), \quad \eta_2 = -p_{\boldsymbol{n}}^j - w_{u_{\boldsymbol{n}}}^j + \frac{1}{\delta}(g_{\boldsymbol{n}} + z_{u_{\boldsymbol{n}}}^j).$$

Solving (3.7) directly is not a good idea in our case, since its coefficient matrix is not expected to be cheap or convenient to work with. Instead, we can merge steps (3.6a) and (3.6c) to obtain a more flexible saddle point system. More specifically, to take (3.6c) into account, we substitute $p_{\boldsymbol{n}} = p_{\boldsymbol{n}}^j + \frac{\rho}{\delta}(D_{\boldsymbol{n}} y_{\boldsymbol{n}} + u_{\boldsymbol{n}} - g_{\boldsymbol{n}})$ into (3.7), and the optimality conditions of (3.6a) and (3.6c) can then be written as:

$$\begin{bmatrix} \rho(J_{1_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}}) & 0 & D_{\boldsymbol{n}}^* \\ 0 & \rho(J_{2_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}}) & I_{\boldsymbol{n}} \\ D_{\boldsymbol{n}} & I_{\boldsymbol{n}} & -\frac{\delta}{\rho} I_{\boldsymbol{n}} \end{bmatrix} \begin{bmatrix} y_{\boldsymbol{n}} \\ u_{\boldsymbol{n}} \\ p_{\boldsymbol{n}} \end{bmatrix} = \begin{bmatrix} \rho(J_{1_{\boldsymbol{n}}} \bar{y}_{\boldsymbol{n}} - w_{y_{\boldsymbol{n}}}^j + \frac{1}{\delta} z_{y_{\boldsymbol{n}}}^j) + (1-\rho) D_{\boldsymbol{n}}^* p_{\boldsymbol{n}}^j \\ \rho(-w_{u_{\boldsymbol{n}}}^j + \frac{1}{\delta} z_{u_{\boldsymbol{n}}}^j) + (1-\rho) p_{\boldsymbol{n}}^j \\ g_{\boldsymbol{n}} - \frac{\delta}{\rho} p_{\boldsymbol{n}}^j \end{bmatrix}. \tag{3.8}$$

At this point, we have to decide how to solve (3.8). For simplicity of exposition, we present here only one way of solving system (3.8), by forming the normal equations and then employing the Preconditioned Conjugate Gradient method (PCG) to solve the resulting positive definite system, assuming that its $(2,2)$ block will be easily invertible. We note that the developments in this section hold for any Schur complement of the matrix in (3.8) (the choice of which Schur complement to use heavily depends on the problem under consideration). The case where neither the $(1,1)$ nor the $(2,2)$ block is easily invertible, will be treated at the end of this section. Pivoting the second and then the third block equation of this system, yields:

$$u_{\boldsymbol{n}} = \left(\rho \left(J_{2_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}} \right) \right)^{-1} \left(-p_{\boldsymbol{n}} - \rho w_{u_{\boldsymbol{n}}}^j + \frac{\rho}{\delta} z_{u_{\boldsymbol{n}}}^j + (1-\rho) p_{\boldsymbol{n}}^j \right),$$

$$p_{\boldsymbol{n}} = \left(\left(\rho J_{2_{\boldsymbol{n}}} + \frac{\rho}{\delta} I_{\boldsymbol{n}} \right)^{-1} + \frac{\delta}{\rho} I_{\boldsymbol{n}} \right)^{-1} (D_{\boldsymbol{n}} y_{\boldsymbol{n}} + r),$$

$$r = -g_{\boldsymbol{n}} + \frac{\delta}{\rho} p_{\boldsymbol{n}}^j - \left(\rho \left(J_{2_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}} \right) \right)^{-1} \left(\rho \left(-w_{u_{\boldsymbol{n}}}^j + \frac{1}{\delta} z_{u_{\boldsymbol{n}}}^j \right) + (1-\rho) p_{\boldsymbol{n}}^j \right),$$

and the resulting normal equations read as follows:

(3.9)
$$S_{\boldsymbol{n}} y_{\boldsymbol{n}} \coloneqq \left(\rho \left(J_{1_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}} \right) + D_{\boldsymbol{n}}^* \left(\left(\rho J_{2_{\boldsymbol{n}}} + \frac{\rho}{\delta} I_{\boldsymbol{n}} \right)^{-1} + \frac{\delta}{\rho} I_{\boldsymbol{n}} \right)^{-1} D_{\boldsymbol{n}} \right) y_{\boldsymbol{n}} =$$
$$\rho \left(J_{1_{\boldsymbol{n}}} \bar{y}_{\boldsymbol{n}} - w_{y_{\boldsymbol{n}}}^j + \frac{1}{\delta} z_{y_{\boldsymbol{n}}}^j \right) + (1-\rho) D_{\boldsymbol{n}}^* p_{\boldsymbol{n}}^j - D_{\boldsymbol{n}}^* \left(\left(\rho J_{2_{\boldsymbol{n}}} + \frac{\rho}{\delta} I_{\boldsymbol{n}} \right)^{-1} + \frac{\delta}{\rho} I_{\boldsymbol{n}} \right)^{-1} r.$$

Finally, we should mention that problem (3.6b) of Algorithm 3.1 is trivial, as it admits a closed form solution. More specifically, we perform the optimization by ignoring the box constraints and then projecting the solution onto the box.

In what follows, using Assumption 1, we present some results concerning the asymptotic behavior of the matrix sequence $\{S_{\boldsymbol{n}}\}_n$, by making use of the Theorems presented in the previous section. The latter is produced by refining an arbitrary discretization applied to (3.2) (assuming it satisfies Assumption 1), employing ADMM to the discretized problem, and forming a certain Schur complement of the joint optimality conditions of (3.6a) and (3.6c). The solution of (3.9) delivers the solution to (3.8), and the remaining ADMM sub-problems can be trivially solved in $O(N(\boldsymbol{n}))$ operations. Following practical applications, we assume δ and ρ to be $\Theta(1)$ and constant along the iterations of ADMM (usually $\delta \in [0.01, 100]$ and $\rho \in [1, 1.618]$) .

THEOREM 3.1. *Given Assumption 1, and the sequence $\{S_{\boldsymbol{n}}\}_n$, with $S_{\boldsymbol{n}}$ given in (3.9), we have that there exists a measurable function $\tau : [0,1]^d \times [-\pi,\pi]^d \mapsto \mathbb{R}$ such that $\tau \geq 0$, $\tau \neq 0$ a.e., and $\{S_{\boldsymbol{n}}\}_n \sim_{GLT} \tau$. Moreover, $S_{\boldsymbol{n}}$ are Hermitian positive definite, $\{S_{\boldsymbol{n}}\}_n \sim_\lambda \tau$, and $\{S_{\boldsymbol{n}}^{-1}\}_n \sim_\lambda \tau^{-1}$.*

Proof. Let Assumption 1 hold. Then, we have that there exist three measurable functions $\kappa,\ \xi_1,\ \xi_2 : [0,1]^d \times [-\pi,\pi]^d \mapsto \mathbb{C}$, such that $\xi_1 \geq 0$, $\xi_2 \geq 0$, $\{J_{1_{\boldsymbol{n}}}\}_n \sim_{GLT} \xi_1$, $\{J_{2_{\boldsymbol{n}}}\}_n \sim_{GLT} \xi_2$, and $\{D_{\boldsymbol{n}}\}_n \sim_{GLT} \kappa$. Furthermore, we can notice that, for any constant $C > 0$, $\{C I_{\boldsymbol{n}}\}_n \sim_{GLT} C$, where C can be considered as a positive constant function (e.g. as a constant on the domain $[-\pi,\pi]^d$, generating a diagonal Toeplitz matrix). This, combined with Theorem 2.16 (conditions (2.)–(4.)), yields that:

$$\{M_{1_{\boldsymbol{n}}}\}_n \coloneqq \left\{ \rho \left(J_{1_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}} \right) \right\}_n \sim_{GLT} \rho(\xi_1 + \delta^{-1}),$$
$$\{M_{2_{\boldsymbol{n}}}\}_n \coloneqq \left\{ \left(\rho \left(J_{2_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}} \right) \right)^{-1} + \frac{\delta}{\rho} I_{\boldsymbol{n}} \right\}_n \sim_{GLT} \left(\rho(\xi_2 + \delta^{-1}) \right)^{-1} + \frac{\delta}{\rho}.$$

Similarly, from Theorem 2.16 (conditions (2.)–(5.)), we have that:

$$\left\{ M_{1_{\boldsymbol{n}}} + D_{\boldsymbol{n}}^* M_{2_{\boldsymbol{n}}}^{-1} D_{\boldsymbol{n}} \right\}_n \sim_{GLT} \rho(\xi_1 + \delta^{-1}) + |\kappa|^2 \left(\left(\rho(\xi_2 + \delta^{-1}) \right)^{-1} + \frac{\delta}{\rho} \right)^{-1},$$

where we used that $\bar{\kappa}\kappa = |\kappa|^2$. Setting $\tau = \rho(\xi_1 + \delta^{-1}) + |\kappa|^2 \left(\left(\rho(\xi_2 + \delta^{-1}) \right)^{-1} + \frac{\delta}{\rho} \right)^{-1}$ and noticing that $\tau > 0$ completes the proof. ∎

Subsequently we present some possible approaches that could allow one to take advantage of the structure preserving property of ADMM. In particular, three possible ways of exploiting the preserved structure are discussed here. However, other approaches could be possible. For this analysis, we will make use of the following proposition:

PROPOSITION 3.2. *Let Assumption 1 hold. Then, there exist sequences of d-level matrix-sequences $\{\{\tilde{D}_{\boldsymbol{n},m}\}_n\}_m$, $\{\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n\}_m$, $\{\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n\}_m$, with uniformly bounded spectral norms with respect to n and m, and sequences of measurable functions $\{\kappa_m\}_m$, $\{\xi_{1_m}\}_m$, and $\{\xi_{2_m}\}_m$ such that $\kappa_m,\ \xi_{1_m},\ \xi_{2_m} : [0,1]^d \times [-\pi,\pi]^d \mapsto \mathbb{C}$, $\xi_{1_m},\ \xi_{2_m}$ are real a.e., non-negative, and:*

- $\{\tilde{D}_{\boldsymbol{n},m}\}_n \xrightarrow{\text{a.c.s.}} \{D_{\boldsymbol{n}}\}_n$, $\{\tilde{D}_{\boldsymbol{n},m}\}_n \sim_{GLT} \kappa_m$, *with* $\kappa_m \to \kappa$ *in measure,*
- $\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n \xrightarrow{\text{a.c.s.}} \{J_{1_{\boldsymbol{n}}}\}_n$, $\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n \sim_{GLT} \xi_{1_m}$, *with* $\xi_{1_m} \to \xi_1$ *in measure,*
- $\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n \xrightarrow{\text{a.c.s.}} \{J_{2_{\boldsymbol{n}}}\}_n$, $\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n \sim_{GLT} \xi_{2_m}$, *with* $\xi_{2_m} \to \xi_2$ *in measure.*

Proof. The proof can be found in [34, Theorem 8.6] for the unilevel case and [35, Theorem 5.6] for the multilevel case. ∎

For the rest of this section we will assume that we have such sequences of d-level GLT sequences available, satisfying the conditions stated in Proposition 3.2. We further assume that these approximate sequences are comprised of matrices that are easy to compute and invert (whenever possible).

3.2. Schur complement approximations. In what follows we present various Schur complement approximations that could potentially serve as preconditioners inside PCG, for solving systems of the form of (3.9) (or any other Schur complement of system (3.8)). The viability of each of the following approaches depends on the structure of the problem, as well as the choice of the discretization. We note that the different approaches are presented for completeness, as well as an indicator of the generality of the proposed methodology. In particular, as the convergence behavior of ADMM does not depend on the choice of preconditioner (assuming that the PCG converges to a desired accuracy), we will only make use of one of the following Schur complement approximations when presenting computational results.

3.2.1. A Schur complement block approximation. Given three sequences of GLT sequences $\{\{\tilde{D}_{\boldsymbol{n},m}\}_n\}_m$, $\{\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n\}_m$, $\{\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n\}_m$, satisfying the conditions of Proposition 3.2, we define the following approximation for the matrix in (3.9):

$$\tilde{S}_{\boldsymbol{n},m} = \rho\left(\tilde{J}_{1_{\boldsymbol{n},m}} + \frac{1}{\delta}I_{\boldsymbol{n}}\right) + \tilde{D}^*_{\boldsymbol{n},m}\left(\left(\rho\tilde{J}_{2_{\boldsymbol{n},m}} + \frac{\rho}{\delta}I_{\boldsymbol{n}}\right)^{-1} + \frac{\delta}{\rho}I_{\boldsymbol{n}}\right)^{-1}\tilde{D}_{\boldsymbol{n},m}. \tag{3.10}$$

THEOREM 3.3. *Let Assumption 1 hold, and assume that we have available the sequences $\{\{\tilde{D}_{\boldsymbol{n},m}\}_n\}_m$, $\{\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n\}_m$, and $\{\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n\}_m$, satisfying the conditions of Proposition 3.2. By defining $\tilde{S}_{\boldsymbol{n},m}$ as in (3.10), we have:*

- $\{\tilde{S}_{\boldsymbol{n},m}\}_n \xrightarrow{\text{a.c.s.}} \{S_{\boldsymbol{n}}\}_n$, $\{\tilde{S}_{\boldsymbol{n},m}\}_n \sim_{GLT} \tau_m$, *and* $\tau_m \to \tau$ *in measure, where* τ *is given in Theorem 3.1 and:*

$$\tau_m = \rho(\xi_{1_m} + \delta^{-1}) + |\kappa_m|^2\left(\left(\rho(\xi_{2_m} + \delta^{-1})\right)^{-1} + \frac{\delta}{\rho}\right)^{-1}. \tag{3.11}$$

- *The sequence* $\{\tilde{S}^{-1}_{\boldsymbol{n},m}S_{\boldsymbol{n}}\}_n$ *is weakly clustered at* 1.
- *For any* $n,\ m$, *the eigenvalues of* $\tilde{S}^{-1}_{\boldsymbol{n},m}S_{\boldsymbol{n}}$ *lie in the interval* $\left[\frac{1}{C_s}, C_s\right]$, *where* C_s *is a positive constant uniformly bounded with respect to* n *and* m.

Proof. For the first condition, by Proposition 3.2 as well as Theorem 2.12 (conditions (1.)–(4.)), we get that $\{\tilde{S}_{\boldsymbol{n},m}\}_n \xrightarrow{\text{a.c.s.}} \{S_{\boldsymbol{n}}\}_n$. Using Proposition 3.2 again, this time combined with Theorem 2.16 (conditions (2.)–(5.)) and Lemma 2.4, yields that τ_m is given by (3.11), and $\tau_m \to \tau$ in measure.

For the second condition, we firstly note that the sequence under consideration is Hermitian and positive definite by Assumption 1. Then, using that $\{\tilde{S}_{\boldsymbol{n},m}\}_n \xrightarrow{\text{a.c.s.}} \{S_{\boldsymbol{n}}\}_n$ implies that, for every m, there exists n_m such that for all $n \geq n_m$:

$$S_{\boldsymbol{n}} = \tilde{S}_{\boldsymbol{n},m} + R_{\boldsymbol{n},m} + N_{\boldsymbol{n},m}, \quad \operatorname{rank}(R_{\boldsymbol{n},m}) \leq c(m)d_n, \quad \|N_{\boldsymbol{n},m}\| \leq \omega(m), \tag{3.12}$$

where n_m, $c(m)$ and $\omega(m)$ depend only on m and are such that:

$$\lim_{m\to\infty} c(m) = \lim_{m\to\infty} \omega(m) = 0.$$

By assumption, it is easy to see that any $\tilde{S}_{\boldsymbol{n},m}$ has a spectral norm uniformly bounded in n and m. Furthermore, since $\delta = \Theta(1)$ and $\rho = \Theta(1)$, we observe that $\|\tilde{S}_{\boldsymbol{n},m}^{-1}\|$ is also uniformly bounded in n. Hence, by multiplying both sides of (3.12) by $\tilde{S}_{\boldsymbol{n},m}^{-1}$:

$$\tilde{S}_{\boldsymbol{n},m}^{-1} S_{\boldsymbol{n}} = I_{\boldsymbol{n}} + \tilde{R}_{\boldsymbol{n},m} + \tilde{N}_{\boldsymbol{n},m},$$

where $\tilde{R}_{\boldsymbol{n},m} = \tilde{S}_{\boldsymbol{n},m}^{-1} R_{\boldsymbol{n},m}$ (thus $\operatorname{rank}(\tilde{R}_{\boldsymbol{n},m}) \leq \operatorname{rank}(R_{\boldsymbol{n},m}) \leq c(m)d_n$) and $\tilde{N}_{\boldsymbol{n},m} = \tilde{S}_{\boldsymbol{n},m}^{-1} N_{\boldsymbol{n},m}$ (and hence $\|\tilde{N}_{\boldsymbol{n},m}\| \leq C\omega(m)$, for some constant $C > 0$, independent of m and n). This, along with the definition of a weak cluster in Definition 2.3, proves the second condition.

For the third condition, let us take some constant $C_\dagger$ of $O(1)$, such that:

$$\max\left\{\|D_{\boldsymbol{n}}\|, \|\tilde{D}_{\boldsymbol{n},m}\|, \|J_{1_{\boldsymbol{n}}}\|, \|J_{2_{\boldsymbol{n}}}\|, \|\tilde{J}_{1_{\boldsymbol{n},m}}\|, \|\tilde{J}_{2_{\boldsymbol{n},m}}\|\right\} \leq C_\dagger, \ \forall\ n,\ m.$$

We know that such a constant exists by Assumption 1. Then, we have that $\lambda_{\min}(S_{\boldsymbol{n}}) \geq \frac{\rho}{\delta}$ and $\lambda_{\max}(S_{\boldsymbol{n}}) \leq \rho C_\dagger + \frac{\rho}{\delta} + \frac{\rho}{\delta} C_\dagger^2$, for any n. The exact same bounds hold also for $\tilde{S}_{\boldsymbol{n},m}$, for every n and m. Using these bounds, we can easily show that:

$$\lambda_{\min}(\tilde{S}_{\boldsymbol{n},m}^{-1} S_{\boldsymbol{n}}) \geq \frac{1}{C_\dagger^2 + \delta C_\dagger + 1}, \quad \lambda_{\max}(\tilde{S}_{\boldsymbol{n},m}^{-1} S_{\boldsymbol{n}}) \leq C_\dagger^2 + \delta C_\dagger + 1,$$

for all $n,\ m$. Upon noticing that $\delta = \Theta(1)$ and $\rho = \Theta(1)$, there exists a constant $C_s = C_\dagger^2 + \delta C_\dagger + 1$ uniformly bounded with respect to n, satisfying the third condition of the Theorem. ∎

REMARK 3.1. *Notice that in order to obtain a strong clustering at 1, we would have to employ some extra assumptions. In particular, we would have to require that the sequences given in Assumption 1 are strongly clustered in the essential range of their symbols, which in turn are required to be different from zero a.e.. Furthermore, we would have to assume that the condition in (3.12) is such that* $c(m)d_n = O(1)$.

REMARK 3.2. *While Assumption 1 holds for a wide range of problems, and one is able to find easily computable sequences satisfying the conditions in Proposition 3.2, it is not often the case that the preconditioner in (3.10) is easy to compute or invert. If this is the case, then Theorem 3.3 guarantees that such a preconditioner will provide a weak cluster of the eigenvalues of the preconditioned matrix at* 1. *We note that while this is not optimal, it is expected to be good enough. This is because of the penalty parameter introduced by ADMM (i.e. $\delta = \Theta(1)$), which (along with the assumption that the involved matrices are uniformly bounded in n) guarantees that the normal equations matrix will be relatively well-conditioned, and hence PCG will converge in a number of iterations independent of the grid size (however, possibly depending on the conditioning of the problem matrix as well as the problem parameters).*

The use of the preconditioner in (3.10) becomes more obvious in the following example. If the d-level approximating matrix sequences $\{\{\tilde{D}_{\boldsymbol{n},m}\}_n\}_m$, $\{\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n\}_m$, *and* $\{\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n\}_m$ *satisfy the conditions of Proposition 3.2, and belong to the set of d-level circulant matrices of size* $N(\boldsymbol{n}) \times N(\boldsymbol{n})$*, i.e.* $\mathcal{C}_{\boldsymbol{n}}$*, then the preconditioner in (3.10) will be cheap to form and store, using the fast Fourier transform (requiring* $O\big(N(\boldsymbol{n})\log(N(\boldsymbol{n}))\big)$ *operations and* $O\big(N(\boldsymbol{n})\big)$ *memory). This is because* $\mathcal{C}_{\boldsymbol{n}}$ *is a commutative ring under standard matrix addition and multiplication (see Theorem 2.10).*

3.2.2. A matching Schur complement approximation. As mentioned earlier, many approximating sequences based on the GLT theory would not allow for an easy computation or storage of the preconditioner in (3.10). While the numerical results of this paper will not focus on this case, we present an alternative to the preconditioner in (3.10), which could allow one to use various GLT approximations for the blocks of the matrix in (3.8), and form an easily computable Schur complement approximation for a matrix of the form of (3.9).

In what follows, we define a Schur complement approximation based on the matching strategy, which was proposed in [59] and has been applied in a wide range of applications (e.g. [23, 58, 60]). While this approach can be very general, it is particularly effective under some additional assumptions imposed on problem (3.3). More specifically, we study the properties of this approximation using the GLT theory, and give certain assumptions under which such an approach would be optimal.

Given three sequences $\{\{\tilde{D}_{\boldsymbol{n},m}\}_n\}_m$, $\{\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n\}_m$, $\{\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n\}_m$, satisfying the conditions of Proposition 3.2, we define the following matrix:

$$\hat{D}_{\boldsymbol{n},m} = \tilde{D}^*_{\boldsymbol{n},m} + \rho^{\frac{1}{2}} \left(\tilde{J}_{1_{\boldsymbol{n},m}} + \frac{1}{\delta} I_{\boldsymbol{n}}\right)^{\frac{1}{2}} \left(\left(\rho \tilde{J}_{2_{\boldsymbol{n},m}} + \frac{\rho}{\delta} I_{\boldsymbol{n}}\right)^{-1} + \frac{\delta}{\rho} I_{\boldsymbol{n}}\right)^{\frac{1}{2}},$$

using which we can define an approximation for the matrix in (3.9) as

$$\hat{S}_{\boldsymbol{n},m} = \hat{D}_{\boldsymbol{n},m} \left(\left(\rho \tilde{J}_{2_{\boldsymbol{n},m}} + \frac{\rho}{\delta} I_{\boldsymbol{n}}\right)^{-1} + \frac{\delta}{\rho} I_{\boldsymbol{n}}\right)^{-1} \hat{D}^*_{\boldsymbol{n},m}. \tag{3.13}$$

For simplicity of exposition let us define the following matrices:

$$M_{1_{\boldsymbol{n}}} = \rho\left(J_{1_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}}\right), \quad \tilde{M}_{1_{\boldsymbol{n}}} = \rho\left(\tilde{J}_{1_{\boldsymbol{n},m}} + \frac{1}{\delta} I_{\boldsymbol{n}}\right),$$

$$M_{2_{\boldsymbol{n}}} = \left(\rho\left(J_{2_{\boldsymbol{n}}} + \frac{1}{\delta} I_{\boldsymbol{n}}\right)\right)^{-1} + \frac{\delta}{\rho} I_{\boldsymbol{n}}, \quad \tilde{M}_{2_{\boldsymbol{n}}} = \left(\rho\left(\tilde{J}_{2_{\boldsymbol{n},m}} + \frac{1}{\delta} I_{\boldsymbol{n}}\right)\right)^{-1} + \frac{\delta}{\rho} I_{\boldsymbol{n}}.$$

Under Assumption 1, we have that $\{M_{1_{\boldsymbol{n}}}\}_n \sim_{GLT} \rho(\xi_1 + \delta^{-1})$, $\{M_{2_{\boldsymbol{n}}}\}_n \sim_{GLT} (\rho(\xi_2 + \delta^{-1}))^{-1} + \frac{\delta}{\rho})$, and $\tilde{M}_{1_{\boldsymbol{n}}} \sim_{GLT} \rho(\xi_{1_m} + \delta^{-1})$ with $\xi_{1_m} \to \xi_1$ in measure, while $\tilde{M}_{2_{\boldsymbol{n}}} \sim_{GLT} (\rho(\xi_{2_m} + \delta^{-1}))^{-1} + \frac{\delta}{\rho})$, where $\xi_{2_m} \to \xi_2$ in measure. Further, notice that all four previous matrix-sequences are comprised of Hermitian and positive definite matrices, each of which admits a square root.

Lemma 3.4. *Let* $\boldsymbol{n} \in \mathbb{N}^d$ *be a d-index and* $\{A_{\boldsymbol{n}}\}_n$ *be a multilevel matrix-sequence with* $A_{\boldsymbol{n}}$ *being Hermitian positive definite of size* $N(\boldsymbol{n}) \times N(\boldsymbol{n})$ *and* $\{A_{\boldsymbol{n}}\}_n \sim_{GLT} \chi$*, where* χ *is a measurable function* $\chi : [0,1]^d \times [-\pi,\pi]^d \mapsto \mathbb{R}$ *such that* $\chi \geq 0$ *and* $\chi \neq 0$ *a.e.. Then, the matrices* $A_{\boldsymbol{n}}$ *(*$A_{\boldsymbol{n}}^{-1}$*, respectively) admit a square root* $A_{\boldsymbol{n}}^{\frac{1}{2}}$ *(*$A_{\boldsymbol{n}}^{-\frac{1}{2}}$*, respectively), such that* $\{A_{\boldsymbol{n}}^{\frac{1}{2}}\}_n \sim_{GLT} \chi^{\frac{1}{2}}$ *(*$\{A_{\boldsymbol{n}}^{-\frac{1}{2}}\}_n \sim_{GLT} \chi^{-\frac{1}{2}}$*, respectively).*

Proof. Let the function $f : (0, \infty) \mapsto (0, \infty)$, be defined as $f(x) = x^{\frac{1}{2}}$. Then, from Theorem 2.16 (condition (7.)), we have that $\{f(A_{\boldsymbol{n}})\}_n \sim_{GLT} f(\chi)$, where $f(A_{\boldsymbol{n}})$ is interpreted as a matrix function applied to the eigenvalues of matrix $A_{\boldsymbol{n}}$. ∎

THEOREM 3.5. *Let Assumption 1 hold, and assume that we have available the sequences* $\{\{\tilde{D}_{\boldsymbol{n},m}\}_n\}_m$, $\{\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n\}_m$, *and* $\{\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n\}_m$, *satisfying the conditions of Proposition 3.2. By defining* $\hat{S}_{\boldsymbol{n},m}$ *as in (3.13), we have:*

- $\{\hat{S}_{\boldsymbol{n},m}\}_n \xrightarrow{\text{a.c.s.}} \{S_{\boldsymbol{n}} + E_{\boldsymbol{n}}\}_n$, *where*

$$\{E_{\boldsymbol{n}}\}_n \coloneqq \left\{ M_{1_{\boldsymbol{n}}}^{\frac{1}{2}} M_{2_{\boldsymbol{n}}}^{-\frac{1}{2}} D_{\boldsymbol{n}} + D_{\boldsymbol{n}}^* M_{2_{\boldsymbol{n}}}^{-\frac{1}{2}} M_{1_{\boldsymbol{n}}}^{\frac{1}{2}} \right\}_n \sim_{GLT} \epsilon,$$

  with

$$\epsilon \coloneqq \rho(\xi_1 + \delta^{-1})^{\frac{1}{2}}((\xi_2 + \delta^{-1})^{-1} + \delta)^{-\frac{1}{2}}(\kappa + \bar{\kappa}).$$

  Furthermore, $\{\hat{S}_{\boldsymbol{n},m}\}_n \sim_{GLT} \tau_m + \epsilon_m$ *and* $\tau_m + \epsilon_m \to \tau + \epsilon$ *in measure, where* τ *is defined in Theorem 3.1, and* $\tau_m,\ \epsilon_m,\ \epsilon$ *are measurable functions having the same domain as* τ. *If* $\tilde{E}_{\boldsymbol{n},m} \coloneqq \tilde{M}_{1_{\boldsymbol{n},m}}^{\frac{1}{2}} \tilde{M}_{2_{\boldsymbol{n},m}}^{-\frac{1}{2}} \tilde{D}_{\boldsymbol{n},m} + \tilde{D}_{\boldsymbol{n},m}^* \tilde{M}_{2_{\boldsymbol{n},m}}^{-\frac{1}{2}} \tilde{M}_{1_{\boldsymbol{n},m}}^{\frac{1}{2}}$ *is positive semi-definite for all* m *and* n, *then the sequence of preconditioned normal equations' matrices is such that:*

$$\{\hat{S}_{\boldsymbol{n},m}^{-1} S_{\boldsymbol{n}}\}_n \sim_{GLT} \tau(\tau_m + \epsilon_m)^{-1} \to \tau(\tau + \epsilon)^{-1}, \text{ as } m \to \infty,$$

  and there exist positive constants $C_{\dagger_1},\ C_{\dagger_2}$, *independent of* $n,\ m$, *such that* $\lambda(\hat{S}_{\boldsymbol{n},m}^{-1} S_{\boldsymbol{n}}) \in [C_{\dagger_1}, C_{\dagger_2}]$, *for all* $n,\ m$.
- *If the matrix sequences* $\{J_{1_{\boldsymbol{n}}}\}_n$, $\{J_{2_{\boldsymbol{n}}}\}_n$, *and* $\{D_{\boldsymbol{n}}\}_n$ *are such that* $J_{1_{\boldsymbol{n}}}$ *and* $J_{2_{\boldsymbol{n}}}$ *are scaled identities or zero matrices, while* $D_{\boldsymbol{n}} + D_{\boldsymbol{n}}^*$ *is Hermitian positive semi-definite, then the matrix-sequence* $\{\hat{S}_{\boldsymbol{n},m}^{-1} S_{\boldsymbol{n}}\}_n$ *is weakly clustered at* $[\frac{1}{2}, 1]$. *If furthermore the matrix-sequence* $\{\tilde{D}_{\boldsymbol{n},m}^{-1} D_{\boldsymbol{n}}\}_n$ *is strongly clustered at* 1, *then the matrix-sequence* $\{\hat{S}_{\boldsymbol{n},m}^{-1} S_{\boldsymbol{n}}\}_n$ *is strongly clustered at* $[\frac{1}{2}, 1]$.

Proof. Firstly, notice that from (3.13) we obtain the following expression:

$$\hat{S}_{\boldsymbol{n},m} = \tilde{S}_{\boldsymbol{n},m} + \tilde{M}_{1_{\boldsymbol{n}}}^{\frac{1}{2}} \tilde{M}_{2_{\boldsymbol{n}}}^{-\frac{1}{2}} \tilde{D}_{\boldsymbol{n}} + \tilde{D}_{\boldsymbol{n}}^* \tilde{M}_{2_{\boldsymbol{n}}}^{-\frac{1}{2}} \tilde{M}_{1_{\boldsymbol{n}}}^{\frac{1}{2}},$$

where $\tilde{S}_{\boldsymbol{n},m}$ is defined as in (3.10). Then, the first part of the Theorem can be proved by employing Lemma 3.4 and by performing a similar analysis to that of the proof of the first and third conditions of Theorem 3.3. For brevity, the latter is omitted.

We proceed by proving the second condition. Notice that if $J_{1_{\boldsymbol{n}}}$ and $J_{2_{\boldsymbol{n}}}$ are scaled identities or zero matrices (the latter being mostly of theoretical interest), then we can represent them exactly, that is $\tilde{J}_{1_{\boldsymbol{n},m}} = J_{1_{\boldsymbol{n}}}$ and $\tilde{J}_{2_{\boldsymbol{n},m}} = J_{2_{\boldsymbol{n}}}$, for all m and n. The latter implies that $M_{1_{\boldsymbol{n}}},\ M_{2_{\boldsymbol{n}}}$ are scaled identities and we can write $M_{\boldsymbol{n}} = M_{1_{\boldsymbol{n}}} = \frac{1}{c_s} M_{2_{\boldsymbol{n}}}$, for some positive constant c_s. We define the following matrix:

$$\bar{S}_{\boldsymbol{n}} = \frac{1}{c_s}(D_{\boldsymbol{n}}^* + \sqrt{c_s} M_{\boldsymbol{n}}) M_{\boldsymbol{n}}^{-1} (D_{\boldsymbol{n}} + \sqrt{c_s} M_{\boldsymbol{n}}).$$

Following exactly the developments in [60, Theorem 4.1] (since $D + D^* \succeq 0$), we can consider the generalized eigenproblem $\bar{S}_{\boldsymbol{n}}^{-1} S_{\boldsymbol{n}} x = \mu x$, and show that $\lambda(\bar{S}_{\boldsymbol{n}}^{-1} S_{\boldsymbol{n}}) \in [\frac{1}{2}, 1]$, where $S_{\boldsymbol{n}}$ is defined as in (3.9), μ is an arbitrary eigenvalue of the preconditioned matrix $\bar{S}_{\boldsymbol{n}}^{-1} S_{\boldsymbol{n}}$ and x the corresponding eigenvector.

Let us now notice that by Assumption 1, the matrix-sequence $\{\bar{S}_{\boldsymbol{n}}\}_n$ is a GLT sequence. In particular, it is easy to see that $\{S_{\boldsymbol{n}} + E_{\boldsymbol{n}}\}_n \equiv \{\bar{S}_{\boldsymbol{n}}\}_n$ and hence its symbol is $\tau+\epsilon$, where ϵ is defined in the first condition of this Theorem. Again, from the first condition of this Theorem, we have that the preconditioner defined in (3.13) is such that $\{\hat{S}_{\boldsymbol{n},m}\}_n \xrightarrow{\text{a.c.s.}} \{S_{\boldsymbol{n}}+E_{\boldsymbol{n}}\}_n \equiv \{\bar{S}_{\boldsymbol{n}}\}_n$, and $\{\hat{S}_{\boldsymbol{n},m}\}_n \sim_{GLT} \tau_m+\epsilon_m$ with $\tau_m+\epsilon_m \to \tau+\epsilon$ in measure. Then, from Theorem 2.12 we know that $\{\hat{S}^{-1}_{\boldsymbol{n},m}\}_n \xrightarrow{\text{a.c.s.}} \{\bar{S}^{-1}_{\boldsymbol{n}}\}_n$. Using Definition 2.11, we have that for all $n \geq n_m$, we can write:

$$\bar{S}^{-1}_{\boldsymbol{n}} = \hat{S}^{-1}_{\boldsymbol{n},m} + R_{\boldsymbol{n},m} + N_{\boldsymbol{n},m}, \quad \operatorname{rank}(R_{\boldsymbol{n},m}) \leq c(m)N(\boldsymbol{n}), \quad \|N_{\boldsymbol{n},m}\| \leq \omega(m), \tag{3.14}$$

where n_m, $c(m)$ and $\omega(m)$ depend only on m and are such that:

$$\lim_{m\to\infty} c(m) = \lim_{m\to\infty} \omega(m) = 0.$$

In view of the previous, we can analyze the sequence $\{\bar{S}^{-1}_{\boldsymbol{n}}S_{\boldsymbol{n}} - \hat{S}^{-1}_{\boldsymbol{n},m}S_{\boldsymbol{n}}\}_n$ as follows:

$$\bar{S}^{-1}_{\boldsymbol{n}}S_{\boldsymbol{n}} - \hat{S}^{-1}_{\boldsymbol{n},m}S_{\boldsymbol{n}} = (\bar{S}^{-1}_{\boldsymbol{n}} - \hat{S}^{-1}_{\boldsymbol{n},m})S_{\boldsymbol{n}} = (R_{\boldsymbol{n},m}+N_{\boldsymbol{n},m})S_{\boldsymbol{n}},$$

where $\operatorname{rank}(R_{\boldsymbol{n},m}S_{\boldsymbol{n}}) \leq \operatorname{rank}(R_{\boldsymbol{n},m}) \leq c(m)N(\boldsymbol{n})$ and $\|N_{\boldsymbol{n},m}S_{\boldsymbol{n}}\| \leq \omega(m)\|S_{\boldsymbol{n}}\| = \Theta(\omega(m))$. In other words, we have that $\{\bar{S}^{-1}_{\boldsymbol{n}}S_{\boldsymbol{n}} - \hat{S}^{-1}_{\boldsymbol{n},m}S_{\boldsymbol{n}}\}_n$ is weakly clustered at zero. Furthermore, as $\lambda(\bar{S}^{-1}_{\boldsymbol{n}}S_{\boldsymbol{n}}) \in [\frac{1}{2},1]$, we conclude that $\{\hat{S}^{-1}_{\boldsymbol{n},m}S_{\boldsymbol{n}}\}_n$ is weakly clustered at $[\frac{1}{2},1]$.

Finally, if we assume that $\{\tilde{D}^{-1}_{\boldsymbol{n},m}D_{\boldsymbol{n}}\}_n$ is strongly clustered at 1, and by noting that $M_{\boldsymbol{n}}$ is a scaled identity (and hence $\tilde{M}_{\boldsymbol{n},m} = M_{\boldsymbol{n}}$, for all n, m), we can conclude that (3.14) holds for $R_{\boldsymbol{n},m}$ such that $\operatorname{rank}(R_{\boldsymbol{n},m}) = O(1)$. By employing a similar methodology as before, this yields that $\{\hat{S}^{-1}_{\boldsymbol{n},m}S_{\boldsymbol{n}}\}_n$ is strongly clustered at $[\frac{1}{2},1]$. ◻

REMARK 3.3. *Let us now briefly discuss the applicability of the preconditioner in* (3.13). *Firstly, it is important to notice that such a preconditioner is generally only sensible when the approximating matrices* $\tilde{J}_{1_{\boldsymbol{n},m}}$ *and* $\tilde{J}_{2_{\boldsymbol{n},m}}$ *are diagonal, circulant or zero. If this is not the case, we discuss a remedy in Section* 3.3.2. *In many applications of interest, the preconditioner* $\tilde{D}_{\boldsymbol{n},m}$ *has a diagonal times a multilevel banded Toeplitz structure (e.g.* [17, 53]*). The application of the preconditioner in* (3.13) *consists in a single (LU or, if applicable, Cholesky) factorization of* $\hat{D}_{\boldsymbol{n},m}$ *at the beginning of the optimization, and subsequently two backward solves for every ADMM iteration. Such an approach should be feasible, in terms of memory and computational requirements, as long as the problem dimensions are not very large and the bandwidth of the matrix* $\tilde{D}_{\boldsymbol{n},m}$ *is small. For high-dimensional problems, one could employ an incomplete or specialized factorization (e.g.* [17]*), possibly assisted by suitable low-rank updates, if necessary. In some cases, replacing the factorization with a specialized iterative solver (such as a multigrid method as in* [53]*) could be beneficial. However, it is important to note that factorization (complete or incomplete) needs to be computed only once. An alternative employing low-rank approximations of the associated matrices is discussed in the following Remark. The suitability of each of the aforementioned approaches depends heavily on the problem under consideration.*

REMARK 3.4. *As mentioned earlier, the proposed preconditioner in* (3.13) *allows one to use a variety of approximations for the blocks of the matrix in* (3.8)*, based on the GLT theory, under certain conditions (which hold for a wide class of problems, such as the problem considered in Section* 4*). In fact, this preconditioner can be seen as an approximation of the preconditioner in* (3.10)*, which in turn has limited applicability unless the approximating blocks have a multilevel circulant structure.*

The limitations of preconditioner (3.13) depend on the problem under consideration. In particular, if the assumptions of the second condition of Theorem 3.5 hold, then it can serve as a basis for constructing easily computable optimal preconditioners. Furthermore, if the aforementioned assumptions hold for the problem under consideration, one might be able to use a tensor product approach with low-rank approximations of the matrices in (3.8) to solve problem (3.3). Such solvers can be extremely effective, allowing one to solve high-dimensional problems, however they tend to require that various features of the problem (e.g. initial conditions, desired state, boundary conditions, discrete solutions) are approximated in a low-rank format, which is not always the case. The proposed approach would allow one to create a rather general tensor product solver for inverse problems measuring the discrepancy of the state variable y *from a desired state* $\bar{\mathrm{y}}$ *as well as the size of the control* u *in the* L^2*-norm, where the structure of the problem allows this. Such solvers have been proposed in [23, 40] for the equality constrained case, and hence the proposed methodology could allow one to further generalize these approaches. Additionally, many low-rank tensor product solvers in the literature require that the objective function has a scaled identity Hessian. This can be alleviated here, by making use of the generalized ADMM presented in Section 3.3.2, alongside the preconditioner in (3.13).*

3.2.3. Element-wise Schur complement approximation. In the context of finite element methods, a popular preconditioner is the so-called element-wise (also known as additive or element-by-element) Schur complement approximation. As this approach has been analyzed multiple times, we only mention it here as a viable alternative for preconditioning the normal equations in (3.9) and refer the interested reader to the available literature. In particular, such preconditioners have been analyzed using the GLT theory in [27, 28]. An analysis for general problems can be found in [55] and the references therein. These preconditioners can be very effective (in fact optimal under reasonable and general assumptions). Furthermore, they can efficiently be implemented in a parallel environment, allowing one to solve huge-scale problem instances (see e.g. [26, 28]).

3.3. General quadratic objective function. As we stressed earlier, it could be the case that both $J_{1_{\boldsymbol{n}}}$ and $J_{2_{\boldsymbol{n}}}$ are general positive semi-definite matrices, whose inverses (if they exist) are expensive to compute. As a consequence, the normal equations could be prohibitively expensive to form. In order to tackle such problems, we propose two alternatives. The former simply avoids forming the normal equations and solves (3.8) instead, using an appropriate Krylov subspace method. The latter approach generalizes the algorithmic framework in Algorithm 3.1, allowing us to simplify the resulting sub-problems. Then, the simplified sub-problems can be solved using PCG alongside any of the previously presented Schur complement approximations.

3.3.1. A saddle point approximation. In many applications, forming a Schur complement of system (3.8) would be very costly. Instead, one could solve system (3.8), which can be seen as a regularized saddle point system. Among many other iterative methods, one could employ preconditioned MINRES to solve systems of this form. The aforementioned method allows only the use of a positive definite preconditioner, hence, many block preconditioners for (3.8) are not applicable. For instance, block-triangular preconditioners, motivated by the work in [41, 54], would generally require a non-symmetric solver such as GMRES [64]. However, block-diagonal preconditioners have been shown to be very effective and efficient in practice for systems of the form of (3.8) (see for example [2, 57, 69]). To that end, we can define the following positive definite block-diagonal preconditioner:

$$\tilde{K}_{\boldsymbol{n},m} = \begin{bmatrix} \rho(\tilde{J}_{1_{\boldsymbol{n},m}} + \frac{1}{\delta} I_{\boldsymbol{n}}) & 0 & 0 \\ 0 & \rho(\tilde{J}_{2_{\boldsymbol{n},m}} + \frac{1}{\delta} I_{\boldsymbol{n}}) & 0 \\ 0 & 0 & \tilde{S}_{\boldsymbol{n},m} \end{bmatrix}, \tag{3.15}$$

where $\tilde{S}_{\boldsymbol{n},m}$ can be defined as in (3.10) or as in (3.13) (and indeed any other suitable Schur complement approximation), assuming that we have available two sparse sequences of d-level GLT sequences $\{\{\tilde{J}_{1_{\boldsymbol{n},m}}\}_n\}_m$ and $\{\{\tilde{J}_{2_{\boldsymbol{n},m}}\}_n\}_m$, satisfying the conditions of Proposition 3.2. We note that preconditioners similar to (3.15) have been analyzed multiple times in the literature and hence such an analysis is omitted here (see for example [2, 57, 61, 68, 69]). It is important to notice that the quality of the preconditioner in (3.15) depends heavily on the quality of the Schur complement approximation, as well as on the approximations of the $(1, 1)$ and $(2, 2)$ blocks of the matrix in (3.8), which can be computed by making use of the GLT theory.

3.3.2. Generalized ADMM. Instead of solving the saddle point system in (3.8), one could derive the following generalized ADMM algorithm, as described in Algorithm 3.2. The following methodology is presented for completeness and is focused on the case where all the associated matrices as well as state and control variables are real. One could apply it to the complex case, however, in that case the theory derived in [20] to support such methods, would no longer hold.

Algorithm 3.2 (2-Block) Generalized ADMM

Input: Let $y^0_{\boldsymbol{n}}, u^0_{\boldsymbol{n}}, z^0_{y_{\boldsymbol{n}}}, z^0_{u_{\boldsymbol{n}}}, p^0_{\boldsymbol{n}}, w^0_{y_{\boldsymbol{n}}}, w^0_{u_{\boldsymbol{n}}} \in \mathbb{R}^{N(\boldsymbol{n})}$, $\delta > 0$, $\rho \in (0, 1]$, $R_y \succ 0$, $R_u \succ 0$.

for $(j = 0, 1, \dots)$ **do**

$$\begin{aligned}(y^{j+1}_{\boldsymbol{n}}, u^{j+1}_{\boldsymbol{n}}) = \underset{y_{\boldsymbol{n}}, u_{\boldsymbol{n}}}{\arg\min} \Big\{ & \mathcal{L}_\delta(y_{\boldsymbol{n}}, u_{\boldsymbol{n}}, z^j_{y_{\boldsymbol{n}}}, z^j_{u_{\boldsymbol{n}}}, p^j_{\boldsymbol{n}}, w^j_{y_{\boldsymbol{n}}}, w^j_{u_{\boldsymbol{n}}}) \\ & + \frac{1}{2}(y_{\boldsymbol{n}} - y^j_{\boldsymbol{n}})^T R_y (y_{\boldsymbol{n}} - y^j_{\boldsymbol{n}}) + (u_{\boldsymbol{n}} - u^j_{\boldsymbol{n}})^T R_u (u_{\boldsymbol{n}} - u^j_{\boldsymbol{n}}) \Big\}\end{aligned} \tag{3.16a}$$

$$(z^{j+1}_{y_{\boldsymbol{n}}}, z^{j+1}_{u_{\boldsymbol{n}}}) = \underset{z_y \in [y_a, y_b],\ z_u \in [u_a, u_b]}{\arg\min} \big\{ \mathcal{L}_\delta(y^{j+1}_{\boldsymbol{n}}, u^{j+1}_{\boldsymbol{n}}, z_{y_{\boldsymbol{n}}}, z_{u_{\boldsymbol{n}}}, p^j_{\boldsymbol{n}}, w^j_{y_{\boldsymbol{n}}}, w^j_{u_{\boldsymbol{n}}}) \big\} \tag{3.16b}$$

$$p^{j+1}_{\boldsymbol{n}} = p^j_{\boldsymbol{n}} + \frac{\rho}{\delta}(D_{\boldsymbol{n}} y^{j+1}_{\boldsymbol{n}} + u^{j+1}_{\boldsymbol{n}} - g_{\boldsymbol{n}}) \tag{3.16c}$$

$$(w^{j+1}_{y_{\boldsymbol{n}}}, w^{j+1}_{u_{\boldsymbol{n}}}) = \Big(w^j_{y_{\boldsymbol{n}}} + \frac{\rho}{\delta}(y^{j+1}_{\boldsymbol{n}} - z^{j+1}_{y_{\boldsymbol{n}}}), w^j_{u_{\boldsymbol{n}}} + \frac{\rho}{\delta}(u^{j+1}_{\boldsymbol{n}} - z^{j+1}_{u_{\boldsymbol{n}}}) \Big) \tag{3.16d}$$

end for

There are two major differences between Algorithm 3.1 and Algorithm 3.2. In the latter method, we have added an extra proximal term in problem (3.6a), which belongs to the class of Bregman distances, and indeed is produced by the Bregman function $\|\cdot\|_R$, where $R = R_y \oplus R_u$, $R_y \succ 0$, and $R_u \succ 0$. For a detailed derivation of proximal methods using Bregman distances, the reader is referred to [30] and the references therein. The second difference is that, in the general case, Algorithm 3.2 requires that the step-size ρ lies in a smaller interval than that allowed in Algorithm 3.1. In fact, the allowed values for ρ depend on the choice of R_y and R_u. We refer the reader to [20] for a more general derivation of methods similar to Algorithm 3.2, in which a precise condition is given for the maximum allowed values of ρ, so that the method converges globally. Furthermore, the authors in [20] prove linear convergence of the method under different sets of conditions, one of which requires that $J_{1_{\boldsymbol{n}}} \succ 0$ and $J_{2_{\boldsymbol{n}}} \succ 0$.

In light of the previous discussion, we can choose:

$$R_{y_{\boldsymbol{n}}} = c_y I_{\boldsymbol{n}} - J_{1_{\boldsymbol{n}}}, \quad R_{u_{\boldsymbol{n}}} = c_u I_{\boldsymbol{n}} - J_{2_{\boldsymbol{n}}},$$

where $c_y,\ c_u > 0$ are such that $R_{y_{\boldsymbol{n}}} \succ 0$, $R_{u_{\boldsymbol{n}}} \succ 0$. With these choices of $R_{y_{\boldsymbol{n}}}$ and $R_{u_{\boldsymbol{n}}}$, the optimality conditions of (3.16a) and (3.16c) involve the coefficient matrix:

$$\begin{bmatrix} \rho(c_y + \frac{1}{\delta})I_{\boldsymbol{n}} & 0 & D_{\boldsymbol{n}}^* \\ 0 & \rho(c_u + \frac{1}{\delta})I_{\boldsymbol{n}} & I_{\boldsymbol{n}} \\ D_{\boldsymbol{n}} & I_{\boldsymbol{n}} & -\frac{\delta}{\rho}I_{\boldsymbol{n}} \end{bmatrix}. \tag{3.17}$$

As one can easily observe, the normal equations operator of (3.17) can be efficiently applied to a vector, and all the previously presented Schur complement approximations can be used within PCG to accelerate the solution of the new simplified sub-problems. Furthermore, notice that this way we can ensure that the $(1, 1)$ and $(2, 2)$ blocks of the matrix in (3.17) are scaled identities, and hence the preconditioner in (3.13) can be particularly effective (see Theorem 3.5).

We should note at this point that similar methodologies can be employed to enforce certain structure on the associated matrices of problem (3.4). While this can be very effective in some cases, by making the ADMM sub-problems easy to solve, it should be used with caution. On the one hand ρ is required, in general, to take values in the interval $(0, 1]$. In practice, the larger the value of ρ, the faster the convergence of ADMM. More importantly, if the constants c_y and c_u are large, we essentially regularize the problem strongly (i.e. we force a large δ, in the case of Algorithm 3.1). This means that tuning δ in Algorithm 3.2 will not allow us to accelerate the algorithm significantly (which is not the case for Algorithm 3.1).

4. The FDE-Constrained Optimization Model. In this section, we present the FDE-constrained optimization problem studied hereon and provide details as to the FDE discretization used. We then highlight some important properties of the resulting discretized matrices.

We define the *Caputo derivative* of a function $\mathrm{f}(t)$ defined on $t \in [t_0, t_1]$, of real order α such that $n-1 < \alpha < n$ with $n \in \mathbb{N}$, as follows:

$${}^{\mathrm{C}}_{t_0}D^{\alpha}_{t}\ \mathrm{f}(t) = \frac{1}{\Gamma(n-\alpha)} \int_{t_0}^{t} \frac{\mathrm{d}^n \mathrm{f}(s)}{\mathrm{d}s^n} \frac{\mathrm{d}s}{(t-s)^{\alpha-n+1}},$$

assuming convergence of the above [21, 51, 62]. We also define the left-sided and right-sided *Riemann–Liouville derivatives* of a function $\mathrm{f}(x)$ defined on $x \in [x_0, x_1]$, of real order β such that $n-1 < \beta < n$ with $n \in \mathbb{N}$, as

$$\begin{aligned} {}^{\mathrm{RL}}_{x_0}D^{\beta}_{x}\ \mathrm{f}(x) &= \frac{1}{\Gamma(n-\beta)} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \int_{x_0}^{x} \frac{\mathrm{f}(s)\,\mathrm{d}s}{(x-s)^{\beta-n+1}}, \\ {}^{\mathrm{RL}}_{x}D^{\beta}_{x_1}\ \mathrm{f}(x) &= \frac{(-1)^n}{\Gamma(n-\beta)} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \int_{x}^{x_1} \frac{\mathrm{f}(s)\,\mathrm{d}s}{(s-x)^{\beta-n+1}}, \end{aligned}$$

respectively. From this, we define the *symmetric Riesz derivative* as follows [62, 65]:

$$ {}^{\mathrm{R}}D_x^{\beta}\ \mathrm{f}(x) = \frac{-1}{2\cos(\frac{\beta\pi}{2})}\left({}^{\mathrm{RL}}_{x_0}D_x^{\beta}\ \mathrm{f}(x) + {}^{\mathrm{RL}}_{x}D_{x_1}^{\beta}\ \mathrm{f}(x)\right). \tag{4.1} $$

We highlight that Caputo derivatives are frequently used for discretization of FDEs in time, given initial conditions, with Riemann–Liouville derivatives correspondingly considered for spatial derivatives, given boundary conditions. We consider the minimization problem:

$$ \begin{aligned} &\min_{\mathrm{y},\mathrm{u}}\ \mathrm{J}(\mathrm{y}(\boldsymbol{x},t),\mathrm{u}(\boldsymbol{x},t)) \\ &\quad \text{s.t.}\ \left({}^{\mathrm{C}}_{0}D_t^{\alpha} - {}^{\mathrm{R}}D_{x_1}^{\beta_1} - {}^{\mathrm{R}}D_{x_2}^{\beta_2}\right)\mathrm{y}(\boldsymbol{x},t) + \mathrm{u}(\boldsymbol{x},t) = \mathrm{g}(\boldsymbol{x},t), \\ &\qquad\ \mathrm{y}_a(\boldsymbol{x},t) \le \mathrm{y}(\boldsymbol{x},t) \le \mathrm{y}_b(\boldsymbol{x},t), \quad \mathrm{u}_a(\boldsymbol{x},t) \le \mathrm{u}(\boldsymbol{x},t) \le \mathrm{u}_b(\boldsymbol{x},t), \end{aligned} \tag{4.2} $$

where the fractional differential equation and additional algebraic constraints are given on the space-time domain $\Omega \times (0,T)$, where $\Omega \subset \mathbb{R}^2$ has boundary $\partial\Omega$, and the spatial coordinates are given by $\boldsymbol{x} = [x_1, x_2]^T$. We impose the initial condition $\mathrm{y}(\boldsymbol{x},0) = 0$ at $t = 0$, and the Dirichlet condition $\mathrm{y} = 0$ on $\partial\Omega \times (0,T)$. We assume that the orders of differentiation satisfy $0 < \alpha < 1$, $1 < \beta_1 < 2$, $1 < \beta_2 < 2$.

The cost functional $J(\mathrm{y},\mathrm{u})$ measures the misfit between the *state variable* y and a given *desired state* $\bar{\mathrm{y}}$ in some given norm, and also measures the 'size' of the *control variable* u. In this paper we consider the cost functional $J(\mathrm{y},\mathrm{u})$ corresponding to L^2-norms measuring both terms:

$$ \mathrm{J}(\mathrm{y},\mathrm{u}) = \frac{1}{2}\int_0^T \int_\Omega (\mathrm{y} - \bar{\mathrm{y}})^2\ \mathrm{d}x\mathrm{d}t + \frac{\gamma}{2}\int_0^T \int_\Omega \mathrm{u}^2\ \mathrm{d}x\mathrm{d}t. \tag{4.3} $$

Here $\gamma > 0$ denotes a regularization parameter on the control variable. We note that other variants for $\mathrm{J}(\mathrm{y},\mathrm{u})$ are possible, including measuring the state misfit and/or the control variable in other norms, as well as alternative weightings within the cost functionals. We also emphasize that it is perfectly reasonable to consider such problems involving FDEs in one or three spatial dimensions (or indeed higher dimensions), rather than in two dimensions as in (4.2), and the methodology in this paper could be readily tailored to such problems.

Upon discretization, we consider the non-shifted Grünwald–Letnikov formula [23, 62, 65, 66] to approximate the Caputo derivative in time:

$$ {}^{\mathrm{C}}_{t_0}D_t^{\alpha}\ \mathrm{y}(t) = \frac{1}{h_t^{\alpha}}\sum_{k=0}^{n_t-1} g_k^{\alpha}\,\mathrm{y}(t - kh_t) + O(h_t), \tag{4.4} $$

where h_t is the step-size in time, and $g_k^{\alpha} = \frac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}$ may be computed recursively via $g_k^{\alpha} = (1 - \frac{\alpha+1}{k})g_{k-1}^{\alpha}$, $k = 1,2,...,\nu$, with $g_0^{\alpha} = 1$ and $\nu \in \mathbb{N}$. This leads to the Caputo derivative matrix for all grid points in the time variable:

$$ \mathscr{C}_{n_t}^{\alpha} = \frac{1}{h_t^{\alpha}}\begin{bmatrix} g_0^{\alpha} & 0 & \cdots & \cdots & 0 \\ g_1^{\alpha} & g_0^{\alpha} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & g_1^{\alpha} & g_0^{\alpha} & 0 \\ g_{n_t-1}^{\alpha} & \cdots & \cdots & g_1^{\alpha} & g_0^{\alpha} \end{bmatrix}. \tag{4.5} $$

For the (left-sided) spatial derivative we use the p-shifted Grünwald–Letnikov formula [3, 50, 52, 62], with shift parameter $p = 1$, to minimize the local truncation error:

$$ {}^{\mathrm{RL}}_{x_0}D^{\beta}_{x}\ \mathrm{y}(x) = \frac{1}{h_x^{\beta}} \sum_{k=0}^{n} g_k^{\beta}\, \mathrm{y}(x-(k-1)h_x) + O(h_x), \tag{4.6} $$

where h_x is the step-size in space, leading to the matrix

$$ \mathscr{L}_n^{\beta,l} = \frac{1}{h_x^{\beta}} \begin{bmatrix} g_1^{\beta} & g_0^{\beta} & 0 & \cdots & 0 \\ g_2^{\beta} & g_1^{\beta} & g_0^{\beta} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & g_2^{\beta} & g_1^{\beta} & g_0^{\beta} \\ g_n^{\beta} & \cdots & \cdots & g_2^{\beta} & g_1^{\beta} \end{bmatrix}, $$

whereby using the formula (4.1) leads to the following Riemann–Liouville derivative matrix for the symmetrized Riesz derivative:

$$ \mathscr{L}_n^{\beta} = \frac{-1}{2\cos(\frac{\beta\pi}{2})} \left(\mathscr{L}_n^{\beta,l} + (\mathscr{L}_n^{\beta,l})^T \right). \tag{4.7} $$

Using all the previous definitions, we can write the discretized version of the FDE constraint within (4.2) as

$$ D_{\boldsymbol{n}} y_{\boldsymbol{n}} + u_{\boldsymbol{n}} = g_{\boldsymbol{n}}, \tag{4.8} $$

where $y_{\boldsymbol{n}}$, $u_{\boldsymbol{n}}$, $g_{\boldsymbol{n}}$ represent the discretized variants of y, u, g, $\boldsymbol{n} = [n_x, n_y, n_t]$ is a 3-index containing the grid sizes along each dimension, and

$$ D_{\boldsymbol{n}} = \mathscr{C}_{n_t}^{\alpha} \otimes I_{n_{x_1}\cdot n_{x_2}} - I_{n_t} \otimes \left(\mathscr{L}_{n_{x_1}}^{\beta_1} \otimes I_{n_{x_2}} + I_{n_{x_1}} \otimes \mathscr{L}_{n_{x_2}}^{\beta_2} \right). \tag{4.9} $$

For simplicity of exposition, in the rest of the paper we assume that $h_{x_1} = h_{x_2} = h_x$, where h_{x_i} is the discretization step in the respective spatial direction, noting that the method readily generalizes to problems where this is not the case.

By using the trapezoidal rule we approximate the two terms in the objective functional (4.3), by

$$ J_{\boldsymbol{n}} = J_{1_{\boldsymbol{n}}} = \frac{1}{\gamma} J_{2_{\boldsymbol{n}}} = \begin{bmatrix} I_{(n_t-1)\cdot n_{x_1}\cdot n_{x_2}} & 0 \\ 0 & \frac{1}{2} I_{n_{x_1}\cdot n_{x_2}} \end{bmatrix}, \tag{4.10} $$

which is applied to vectors arising from every time-step, apart from the initial time $t = 0$. Notice that matrix $J_{\boldsymbol{n}}$ is diagonal with only two different values on the diagonal, and hence can be almost exactly approximated by a scaled identity.

We should mention that we assume constant diffusion coefficients in the FDE constraints. In turn, this yields that the discretized constraint matrix has a multilevel Toeplitz structure. As we discuss in the following section, such matrices can be approximated by circulant preconditioners, which in turn allow us to use the preconditioner in (3.10) for accelerating the solution of the resulting ADMM sub-problems. In the presence of non-constant diffusion coefficients, the discretized constraint matrices would belong to the class of multilevel GLT sequences. In this case, circulant preconditioners would no longer be effective and we would have to approximate such matrices using diagonal times multilevel banded Toeplitz matrices (see for example [24, 53]). In light of the discussion in Section 3, we can observe that one could extend the results presented in this paper to the non-constant diffusion coefficient case, by making use of the preconditioner in (3.13) (upon noting that the discretization of the functional in (4.3) yields a diagonal matrix). For brevity of presentation this is left to a future study.

In the following proposition, we summarize some well-known properties of the fractional binomial coefficients that arise above when constructing the matrices $\mathscr{C}_\alpha$ and $\mathscr{L}_\beta$ (see for example [39, page 397], or [51, 72]):

Proposition 4.1. *Let $0 < \alpha < 1$ and $1 < \beta < 2$, with $g_k^\alpha,\ g_k^\beta$ as in (4.4), (4.6). Then, we have that:*

$$g_0^\alpha > 0,\ g_k^\alpha < 0,\ \forall\ k \geq 1, \quad \sum_{k=0}^{n_t} g_k^\alpha > 0,\ \forall\ n_t \geq 1, \tag{4.11}$$

$$g_0^\beta = 1,\ g_1^\beta = -\beta,\ g_2^\beta > g_3^\beta > \ldots > 0, \quad \sum_{k=0}^{\infty} g_k^\beta = 0,\ \sum_{k=0}^{n} g_k^\beta < 0,\ \forall\ n \geq 1. \tag{4.12}$$

5. Toeplitz Matrices and Circulant Preconditioners. In this section, we propose a multilevel circulant preconditioner, suitable for approximating multilevel Toeplitz matrices, and then examine the quality of such a preconditioner for the problem at hand, showing that the preconditioner is in fact a.c.s. for a scaled version of the coefficient matrix in (4.9).

Toeplitz and multilevel Toeplitz matrices appear often when (numerically) solving partial, integral, or fractional differential equations, problems in time series analysis, as well as in signal processing (see for example [1, 46, 63, 74], and the references therein). An active area of research is that of solving a huge-scale systems of linear equations, $Ax = b$, where the matrix A has some specific structure, such as Toeplitz, multilevel Toeplitz, or it can be written as a combination of Toeplitz and other structured matrices. There are two major approaches for solving such systems. One alternative is to solve them directly by exploiting the matrix structure (see for example [6, 16, 47, 73]). A more popular approach is to employ some iterative method to solve the system, assisted by an appropriately designed preconditioner, to ensure that the iterative method achieves fast convergence (as in [8, 9, 10, 11, 12, 13, 14, 15, 42, 44, 45, 46, 56, 71]). An equally rich literature exists for preconditioning Toeplitz-like linear systems arising specifically from the discretization of fractional diffusion equations (see [17, 24, 25, 29, 32, 45, 46, 48, 53], among others).

In this paper, we follow the simplest possible approach: that of approximating multilevel Toeplitz matrices using multilevel circulant preconditioners. To do so, we first have to derive a unilevel circulant approximation of an arbitrary unilevel Toeplitz matrix. Given a unilevel Toeplitz matrix $T_n \in \mathbb{R}^{n\times n}$, we employ the circulant approximation proposed for the first time in [14] (also called the T. Chan preconditioner for T_n). More specifically, we define the optimal circulant approximation of T_n, as the solution of the following optimization problem:

$$C_1(T_n) = \min_{C_n \in \mathcal{C}_n} \|C_n - T_n\|_F, \tag{5.1}$$

where $\mathcal{C}_n$ is the set of all $n \times n$ circulant matrices, and $\|\cdot\|_F$ the *Frobenius norm.* It turns out that (5.1) admits the following closed form solution:

$$c_i = \frac{(n-i)\cdot t_i + i \cdot t_{-n+i}}{n}, \quad i \in \{0, \ldots, n-1\}.$$

Then, we can write $C_1(T_n) = F_n^* \Lambda_n F_n$, where F_n is the discrete Fourier transform of size n and Λ_n is a diagonal matrix containing the eigenvalues of $C_1(T_n)$, which can be computed as $\Lambda_n = \text{diag}(F_n c_1)$, where c_1 is the first column of $C_1(T_n)$. Other unilevel circulant approximations are possible, such as those proposed in [12, 13, 71], however, the T. Chan preconditioner seems (empirically) to behave better for the problem under consideration.

We now focus on the discretized FDE given in (4.8). By multiplying this equation on both sides by $\psi = \min\{h_t^\alpha, h_x^{\beta_1}, h_x^{\beta_2}\}$, we have:

$$B_{\boldsymbol{n}} y_{\boldsymbol{n}} + \psi u_{\boldsymbol{n}} = \psi g_{\boldsymbol{n}},$$

where $y_{\boldsymbol{n}}$, $u_{\boldsymbol{n}}$, $g_{\boldsymbol{n}}$ represent the discretized variants of y, u, g, $B_{\boldsymbol{n}} = \psi D_{\boldsymbol{n}}$, with $D_{\boldsymbol{n}}$ defined as in (4.9), h_t, h_x the time and spatial mesh-sizes, and $\boldsymbol{n} = [n_{x_1}, n_{x_2}, n_t]$. We observe that the matrix $D_{\boldsymbol{n}}$ (and hence $B_{\boldsymbol{n}}$) enjoys a 3-level Toeplitz structure. In particular, each block of $D_{\boldsymbol{n}}$ ($B_{\boldsymbol{n}}$) enjoys a quadrantally symmetric block Toeplitz structure (such matrices are analyzed for example in [10]). Given the matrix $B_{\boldsymbol{n}}$, we can define its T. Chan-based 3-level circulant preconditioner as:

(5.2)
$$\begin{aligned} C_3(B_{\boldsymbol{n}}) &= \psi C_1(\mathscr{C}_{n_t}^{\alpha}) \otimes I_{n_{x_1} \cdot n_{x_2}} - \psi I_{n_t} \otimes \big(C_1(\mathscr{L}_{n_{x_1}}^{\beta_1}) \otimes I_{n_{x_2}} + I_{n_{x_1}} \otimes C_1(\mathscr{L}_{n_{x_2}}^{\beta_2})\big) \\ &= (F_{n_{x_1}} \otimes F_{n_{x_2}} \otimes F_{n_t})^* \Lambda_{\boldsymbol{n}} (F_{n_{x_1}} \otimes F_{n_{x_2}} \otimes F_{n_t}), \end{aligned}$$

where $\Lambda_{\boldsymbol{n}}$ is the diagonal eigenvalue matrix of the preconditioner, computed as:

$$\Lambda_{\boldsymbol{n}} = \psi \Lambda_\alpha \otimes I_{n_{x_1} \cdot n_{x_2}} - \psi I_{n_t} \otimes \big(\Lambda_{\beta_1} \otimes I_{n_{x_2}} + I_{n_{x_1}} \otimes \Lambda_{\beta_2}\big),$$

with Λ_α, Λ_{β_1}, Λ_{β_2} being the diagonal matrices containing the eigenvalues of the T. Chan approximations of the matrices $\mathscr{C}_{n_t}^{\alpha}$, $\mathscr{L}_{n_{x_1}}^{\beta_1}$, and $\mathscr{L}_{n_{x_2}}^{\beta_2}$, respectively.

The preconditioner in (5.2) can be computed efficiently in $O(N(\boldsymbol{n}) \log N(\boldsymbol{n}))$ operations, using the fast Fourier transform. The storage requirements are $O(N(\boldsymbol{n}))$ since we only need to store the eigenvalue matrix, that is $\Lambda_{\boldsymbol{n}}$. Clearly, the preconditioner in (5.2) can be defined similarly for FDEs of arbitrary dimension, say d. Given a d-index $\boldsymbol{n}$, containing the level sizes of an arbitrary d-level Toeplitz $T_{\boldsymbol{n}}$ or circulant matrix $C_{\boldsymbol{n}}$, we summarize the computational and storage costs of various recursive linear algebra operations in Table 5.1.

Table 5.1: Summary of computational and storage complexity

Structure	**Operation**	**Computations**	**Storage**
d-level circulant	$C_{\boldsymbol{n}} x$	$O(N(\boldsymbol{n}) \log N(\boldsymbol{n}))$	$O(N(\boldsymbol{n}))$
d-level circulant	$C_{\boldsymbol{n}}^{-1} x$	$O(N(\boldsymbol{n}) \log N(\boldsymbol{n}))$	$O(N(\boldsymbol{n}))$
d-level circulant	$C_{\boldsymbol{n}}^{(1)} C_{\boldsymbol{n}}^{(2)}$	$O(N(\boldsymbol{n}))$	$O(N(\boldsymbol{n}))$
d-level circulant	$C_{\boldsymbol{n}}^{(1)} + C_{\boldsymbol{n}}^{(2)}$	$O(N(\boldsymbol{n}))$	$O(N(\boldsymbol{n}))$
d-level Toeplitz	$T_{\boldsymbol{n}} x$	$O(2^d N(\boldsymbol{n}) \log N(\boldsymbol{n}))$	$O(2^d N(\boldsymbol{n}))$
d-level circulant	Construct $C_d(T_{\boldsymbol{n}})$	$O(N(\boldsymbol{n}) \log N(\boldsymbol{n}))$	$O(N(\boldsymbol{n}))$

Using the definition of the matrices used to construct matrix $B_{\boldsymbol{n}}$ (see (4.5) and (4.7)), we are now able to derive the generating function of this 3-level Toeplitz matrix.

To that end, let us define the following scalars:

$$\nu_1 = \frac{\psi}{h_x^{\beta_1}}, \quad \nu_2 = \frac{\psi}{h_x^{\beta_2}}, \quad \nu_3 = \frac{\psi}{h_t^{\alpha}}, \tag{5.3}$$

which are obviously bounded above by 1, from the definition of ψ. Of course in order for these to be theoretically meaningful, we have to assume that $h_t^{\alpha} \propto h_x^{\beta_1} \propto h_x^{\beta_2}$.

LEMMA 5.1. *Let $\boldsymbol{n} = [n_{x_1}, n_{x_2}, n_t]$ be a 3-index and define the matrix $D_{\boldsymbol{n}}$ as in (4.9). Then, the symbol generating the matrix-sequence $\{B_{\boldsymbol{n}}\}_n = \{\psi D_{\boldsymbol{n}}\}_n$, can be expressed as:*

$$\begin{gathered}\phi_{\beta_1,\beta_2,\alpha}(\boldsymbol{\theta}) = \nu_3 \sum_{k=0}^{\infty} g_k^{\alpha} e^{ik\theta_3} - \\ \sum_{k=-1}^{\infty} \left(\frac{-\nu_1}{2\cos(\frac{\beta_1\pi}{2})} \big(g_{k+1}^{\beta_1} (e^{ik\theta_1} + e^{-ik\theta_1}) \big) + \frac{-\nu_2}{2\cos(\frac{\beta_2\pi}{2})} \big(g_{k+1}^{\beta_2} (e^{ik\theta_2} + e^{-ik\theta_2}) \big) \right),\end{gathered} \tag{5.4}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]$, and g_k^c is the fractional binomial coefficient, for some $c \in (0,1) \cup (1,2)$ and an arbitrary $k \geq 0$. Thus, we can write $B_{\boldsymbol{n}} = T_{\boldsymbol{n}}(\phi_{\beta_1,\beta_2,\alpha})$.

Proof. We omit the proof, which follows easily from the definition of the matrices within $B_{\boldsymbol{n}}$, that is using the definition of $\mathscr{C}_n^{\alpha}$ in (4.5) and $\mathscr{L}_n^{\beta}$ in (4.7). The reader is referred to [24, 46, 53], among others, for derivations of similar results. The alternative representation of matrix $B_{\boldsymbol{n}}$ follows directly from Theorem 2.8. $\square$

To analyze the effectiveness of the proposed 3-level circulant preconditioner for $B_{\boldsymbol{n}}$, we prove that the trigonometric polynomial generating function of matrix $B_{\boldsymbol{n}}$ belongs to the Wiener class (that is, it has absolutely summable coefficients).

LEMMA 5.2. *Assume that β_1 and β_2 are bounded away from 1. Then, the generating function $\phi_{\beta_1,\beta_2,\alpha}(\boldsymbol{\theta})$ defined in (5.4) belongs to the Wiener class, that is:*

$$\phi_{\beta_1,\beta_2,\alpha}(\boldsymbol{\theta}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \phi_{\boldsymbol{k}} e^{i\langle \boldsymbol{k},\boldsymbol{\theta}\rangle}, \text{ such that } \sum_{\boldsymbol{k}\in\mathbb{Z}^3} |\phi_{\boldsymbol{k}}| < \infty.$$

Proof. For brevity of presentation, we provide an outline of the proof. Firstly, one has to transform (5.4) to the form $\phi_{\beta_1,\beta_2,\alpha}(\boldsymbol{\theta}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \phi_{\boldsymbol{k}} e^{i\langle \boldsymbol{k},\boldsymbol{\theta}\rangle}$, by matching the coefficients of the associated trigonometric polynomials. By taking the absolute values of the matched coefficients, applying the triangle inequality, and using the properties of the fractional binomial coefficients, summarized in Proposition 4.1, we obtain:

$$\begin{aligned}\sum_{\boldsymbol{k}\in\mathbb{Z}^3} |\phi_{\boldsymbol{k}}| &\leq \nu_3 \sum_{k=0}^{\infty} |g_k^{\alpha}| + \sum_{k=-1}^{\infty} \left(\frac{\nu_1}{|\cos(\frac{\beta_1\pi}{2})|} |g_{k+1}^{\beta_1}| + \frac{\nu_2}{|\cos(\frac{\beta_2\pi}{2})|} |g_{k+1}^{\beta_2}| \right) \\ &\leq (2\nu_3) \cdot g_0^{\alpha} + \left(\frac{2\nu_1}{|\cos(\frac{\beta_1\pi}{2})|} \right) \beta_1 + \left(\frac{2\nu_2}{|\cos(\frac{\beta_2\pi}{2})|} \right) \beta_2.\end{aligned}$$

The latter completes the proof. $\square$

Using the results presented in [10, 44, 45], we can derive the following Theorem, which in fact shows that the 3-level circulant approximation of matrix $B_{\boldsymbol{n}}$ defined in (5.2) is an a.c.s. for it.

THEOREM 5.3. *Let $B_{\boldsymbol{n}} = \psi D_{\boldsymbol{n}}$ where $\boldsymbol{n} = [n_{x_1}, n_{x_2}, n_t]$, and $C_3(B_{\boldsymbol{n}})$ its circulant approximation defined in (5.2). For every $\epsilon(m) > 0$, such that $\epsilon(m) \to 0$ as $m \to \infty$, there exist constants N_{x_1}, N_{x_2}, N_t, such that for all $n_{x_1} > N_{x_1}$, $n_{x_2} > N_{x_2}$, $n_t > N_t$:*

$$B_{\boldsymbol{n}} - C_3(B_{\boldsymbol{n}}) = U_{\boldsymbol{n},\epsilon(m)} + V_{\boldsymbol{n},\epsilon(m)},$$

where

$$\mathrm{rank}(U_{\boldsymbol{n},\epsilon(m)}) = O(n_{x_2}n_{x_1} + n_{x_1}n_t + n_t n_{x_2}), \quad \|V_{\boldsymbol{n},\epsilon(m)}\|_2 < \epsilon(m).$$

Proof. The proof is omitted since it follows exactly the developments in [45, Theorem 3.2 and Theorem 4.1], with the only difference being that the Strang unilevel circulant approximation is used there (for example see [12]) instead of the T. Chan approximation. Notice that the authors in [45] assume invertibility of $C_3(B_{\boldsymbol{n}})$, using which they prove a weak clustering result. Hence, to prove the result stated here, one needs to follow only part of the proof outlined in [45, Theorem 3.2]. ∎

REMARK 5.1. *Following* [45, Remark 4.1]*, assuming that $d = O(1)$, we can recursively extend the result of Theorem 5.3 to the d-level case, using induction. In other words, the developments discussed in this paper can be extended trivially to higher dimensional FDEs. As expected, the circulant approximation becomes weaker as the dimension of the associated FDE is increased. In particular, the result in* [67] *shows that in the general case, any multilevel circulant preconditioner for multilevel Toeplitz matrices is not a superlinear preconditioner. Superlinear preconditioners are important, in that they allow preconditioned Conjugate Gradient-like methods to converge in a constant number of iterations, independently of the size of the problem. In general, one could not hope of achieving a strong clustering when preconditioning multilevel Toeplitz matrices using multilevel circulant preconditioners. In light of that, it comes as no surprise that a preconditioner like the one in* (5.2) *does not asymptotically capture all of the eigenvalues of the approximated multilevel Toeplitz matrix.*

REMARK 5.2. *Let us now notice that a scaled identity approximation for the discretized objective Hessian matrix in* (4.10) *yields (trivially) a GLT sequence. Similarly, the approximation $C_3(B_{\boldsymbol{n}})$ in* (5.2) *for the matrix $B_{\boldsymbol{n}} = \psi D_{\boldsymbol{n}}$, where $D_{\boldsymbol{n}}$ is defined in* (4.9)*, is also a GLT sequence (since it can be considered as a multilevel Toeplitz matrix). In view of the previous, as well as Theorem 2.16 (condition (6.)), we can see that the proposed approximations for the matrices associated to the discretized version of* (4.2) *satisfy the conditions of Proposition 3.2. Hence, we are able to invoke Theorem 3.3 for the preconditioner in* (3.10)*, which is constructed by using the aforementioned multilevel circulant approximations. Thus, we are able to show that the resulting preconditioned ADMM system matrix, corresponding to the normal equations in* (3.9)*, is weakly clustered at 1. Furthermore, by the same Theorem, we expect convergence of PCG in a number of iterations independent of the grid-size.*

6. Implementation Details and Numerical Results.

In this section we discuss specific implementation details and present the numerical results obtained by running the implementation of the proposed method over a variety of settings of the FDE optimization problem.

6.1. Test problem and implementation details.

We assess the performance of the proposed method on the following test problem. We attempt to numerically solve problem (4.2). The state and the control are defined on the domain $\Omega \times (0,T) = (0,1)^2 \times (0,1)$. For some $n \in \mathbb{N}$, the discretized grid contains $n \times n \times n$ uniform points, in space and time (i.e. we make use of the 3-index $\boldsymbol{n} = [n, n, n]$), which yields:

$$x_1^i = ih_x,\ x_2^j = jh_x,\ t^k = kh_t,\ i, j = 1, \ldots, n,\ k = 1, \ldots, n,\ h_x = \frac{1}{n+1},\ h_t = h_x.$$

It is worth mentioning that the choice of the number of discretization points in time should depend on the value of the fractional derivative orders. In particular, in the theory we had to assume that $h_t^{\alpha} \propto h_x^{\beta_1} \propto h_x^{\beta_2}$. Of course, this could be difficult to satisfy for certain values of α, β_1, and β_2. In terms of discretization error, $n_t = n$ suffices, as we employ first-order numerical schemes for the space and time fractional derivatives. In what follows, we choose to use $n_t = n$ throughout all the experiments, noting that for very large values of n, this should be adjusted to take into consideration the values of the fractional derivative orders. Such an increase in the number of discretization points in time, could potentially be tackled by the use of higher-order numerical methods for the space fractional derivatives (see [22] and the references therein for higher-order approximations for the Riemann–Liouville and Riesz fractional derivatives). This is omitted for a future study.

As a desired state function, we set $\bar{\mathrm{y}}(x_1, x_2, t) = 10\cos(10x_1)\sin(x_1x_2)(1 - e^{-5t})$, as in [23, Section 5.1], with homogeneous boundary and initial conditions. Throughout this section, we employ the convention that $n_{x_1} = n_{x_2} = n_t$, and we only present the overall size of the discretized state vector, that is $N(\boldsymbol{n}) = n_{x_1} \cdot n_{x_2} \cdot n_t = n_{x_1}^3$. As an indicator of convergence of the numerical method, we apply the trapezoidal rule to roughly approximate the discrepancy between the solution for the state and the desired state on the discrete level, i.e.:

$$\mathcal{E}_{L^2}(y - \bar{y}) \approx \|\mathrm{y} - \bar{\mathrm{y}}\|_{L^2}.$$

We should note that the previous measure approximates the misfit between the state and the desired state of the continuous problem, and hence it is not expected to converge to zero. Due to the Dirichlet boundary conditions, there is a mismatch between y and $\bar{y}$ on the boundary. Hence a refinement in the grid size is expected to result in slight increase in the approximate discrepancy measure.

We implement a standard 2-Block ADMM for solving problems of the form of (3.4). The implementation follows exactly the developments in Section 3. We solve system (3.9) using the MATLAB function `pcg`. We note that while various potential acceleration strategies for ADMMs have been studied in the literature (see for example [5, 37]), the focus of the paper is to illustrate the viability of the proposed approach, and hence the simplest possible ADMM scheme is adopted. The step-size of ADMM is chosen to be close to the maximum allowed one in all computations, that is $\rho = 1.618$. The termination criteria of the ADMM are summarized as follows:

$$\left(\|B_{\boldsymbol{n}} y_{\boldsymbol{n}}^j + \psi(u_{\boldsymbol{n}}^j - g_{\boldsymbol{n}})\|_\infty \leq 10^{-4}\right) \wedge \left(\|y_{\boldsymbol{n}}^j - z_{y_{\boldsymbol{n}}}^j\|_\infty \leq 10^{-4}\right) \wedge \left(\|u_{\boldsymbol{n}}^j - z_{u_{\boldsymbol{n}}}^j\|_\infty \leq 10^{-4}\right).$$

In order to avoid unnecessary computations, we do not require a specific tolerance for the dual infeasibility. Instead, we report the dual infeasibility at the accepted optimal point. The Krylov solver tolerance is set dynamically, based on the accuracy attained at the respective ADMM iteration. In particular, the required tolerance for the Krylov solver is set to:

$$\text{In. Tol.} = 0.05 \cdot \max\left\{ \min\{\|B_{\boldsymbol{n}} y_{\boldsymbol{n}}^j + \psi(u_{\boldsymbol{n}}^j - g_{\boldsymbol{n}})\|_\infty, \|y_{\boldsymbol{n}}^j - z_{y_{\boldsymbol{n}}}^j\|_\infty, \|u_{\boldsymbol{n}}^j - z_{u_{\boldsymbol{n}}}^j\|_\infty\}, 10^{-4}\right\},$$

at every iteration j. Hence, we present the average number of inner iterations in the results to follow. Furthermore, we employ the convention that the discretized restricting functions are of the form $y_{b_{\boldsymbol{n}}} = -y_{a_{\boldsymbol{n}}} = c \cdot e_{\boldsymbol{n}}$ (or $u_{b_{\boldsymbol{n}}} = -u_{a_{\boldsymbol{n}}} = c \cdot e_{\boldsymbol{n}}$), where $e_{\boldsymbol{n}}$ is the $N(\boldsymbol{n})$-dimensional vector of ones and $c > 0$. Thus, we present only the value of the entries of $y_{a_{\boldsymbol{n}}}$ ($u_{a_{\boldsymbol{n}}}$, respectively).

As we discussed earlier, the FDE constraints were scaled by the constant ψ, since this was required from the theory (see Theorem 5.3). By doing this, we ensure that the elements of the matrix $B_{\boldsymbol{n}}$ are of order 1 (assuming that $h_t^{\alpha} \propto h_{x_i}^{\beta_i}$, for $i = 1, 2$). As a result, the discretized control in the FDE constraints is multiplied by ψ. In ADMM such a scaling translates to a scaled step of the dual variables corresponding to the FDE constraints. In order to improve the balance of the algorithm, we multiply by ψ the constraints linking $u_{\boldsymbol{n}}$ with its copy variables $z_{u_{\boldsymbol{n}}}$, thus scaling all the dual multipliers corresponding to these constraints.

The penalty parameter of ADMM, δ, is chosen from a pool of five values which deliver reasonably good behavior of the method. More specifically, for the experiments to follow we choose $\delta \in \{0.1, 0.4, 2, 10, 100\}$. We note here that one could tune this parameter for each problem instance and obtain significantly better results. However, as this is not practical, we restrict ourselves to a small set of possible values.

The experiments were conducted on a PC with a 2.2 GHz Intel (hexa-) core i7 processor, run under the Windows 10 operating system. The code is written in MATLAB R2019a.

6.2. Numerical Results.

We distinguish three types of problems:

- Problems with box constraints on the state y,
- problems with box constraints on the control u, and
- problems with box constraints on both variables.

As expected and verified in practice, the third type of problem is the most difficult one. Hence, we will focus our attention on problems with box constraints on both variables, while presenting a few experiments on problems of the other two types.

***Box constraints on the state* $\mathbf{y}$.** Let us briefly focus on the case where the state variable is required to stay in a box, while the control is free, that is $\mathrm{y_a} \leq \mathrm{y} \leq \mathrm{y_b},\ -\infty \leq \mathrm{u} \leq \infty$. Using similar arguments as in [23, 29], we can see that an optimal solution in this case is guaranteed to exist. We run the method for different inequality bounds on the state y. The results are summarized in Table 6.1. All fixed parameters are provided at the title of the respective Table.

Table 6.1: Inequalities on the state: Varying restriction bounds (with $N = 50^3$, $\beta_x = \beta_y = 1.3$, $\alpha = 0.7$, $\gamma = 10^{-4}$, $\delta = 0.1$).

$\boldsymbol{y_a}$	$\boldsymbol{\mathcal{E}_{L^2}(y - \bar{y})}$	**Dual Inf.**	**Iterations**		**Time (s)**
			PCG	ADMM	
-7	5.60×10^{-1}(*)[1]	2.13×10^{-3}	9	75	142.31
-5	5.80×10^{-1}	3.14×10^{-3}	10	105	206.73
-3	7.88×10^{-1}	4.11×10^{-3}	10	100	193.77
-1	1.38×10^{0}	8.74×10^{-4}	10	86	173.49

***Box constraints on the control* $\mathbf{u}$.** We now focus on the case with $-\infty \leq \mathrm{y} \leq \infty,\ \mathrm{u_a} \leq \mathrm{u} \leq \mathrm{u_b}$. Again, it is straightforward to show that such a problem admits an optimal solution (see [29]). We run the method for different inequality bounds on the

[1](*) means that the solution coincides with the equality constrained solution; all the variables lie strictly within the restriction bounds.

control u. The results are summarized in Table 6.2 (including all the values of the parameters used to perform the experiment).

Table 6.2: Inequalities on the control: Varying restriction bounds (with $N = 50^3$, $\beta_x = \beta_y = 1.3$, $\alpha = 0.7$, $\gamma = 10^{-4}$, $\delta = 0.4$).

u_a	$\mathcal{E}_{L^2}(y-\bar{y})$	Dual Inf.	Iterations		Time (s)
			PCG	ADMM	
−400	5.60×10^{-1}(*)	8.43×10^{-4}	16	30	84.46
−300	5.65×10^{-1}	8.79×10^{-4}	19	22	72.77
−200	6.25×10^{-1}	4.39×10^{-4}	17	28	85.55
−100	8.90×10^{-1}	1.49×10^{-4}	18	65	205.69

Box constraints on both variables. Let us now consider the case where $\mathrm{y_a} \le \mathrm{y} \le \mathrm{y_b}$, $\mathrm{u_a} \le \mathrm{u} \le \mathrm{u_b}$. In general, in this case one is not able to conclude that the problem admits an optimal solution. Thus, we run the method on instances for which a solution is known to exist.

First, we present the runs of the method for different inequality bounds in Table 6.3. Next, we present the runs of the method for varying grid size in Table 6.4. As one can observe in Table 6.4, the grid size does not affect the average number of inner PCG iterations. This comes in line with our observations in Section 3. Nevertheless, it is expected that ADMM requires more iterations as the size of the problem increases. Furthermore, we can observe the first-order convergence of the numerical method, as n is increased.

Table 6.3: Inequalities on both variables: Varying restriction bounds (with $\beta_x = \beta_y = 1.3$, $\alpha = 0.7$, $\gamma = 10^{-4}$, $\delta = 0.4$, $N = 50^3$).

y_a	u_a	$\mathcal{E}_{L^2}(y-\bar{y})$	Dual Inf.	Iter.		Time (s)
				PCG	ADMM	
−7	−400	5.60×10^{-1}(*)	4.88×10^{-3}	10	36	74.20
−7	−200	5.94×10^{-1}	2.35×10^{-3}	11	38	80.14
−4	−350	6.45×10^{-1}	1.99×10^{-4}	18	126	412.66
−1	−400	1.38×10^{0}	2.56×10^{-4}	19	109	377.86

Table 6.4: Inequalities on both variables: Varying grid size (with $\beta_x = \beta_y = 1.3$, $\alpha = 0.7$, $\gamma = 10^{-4}$, $y_a = -4$, $u_a = -350$).

N	δ	$\mathcal{E}_{L^2}(y-\bar{y})$	Dual Inf.	Iter.		Time (s)
				PCG	ADMM	
8^3	2	3.87×10^{-1}	5.23×10^{-4}	12	86	1.89
16^3	2	5.02×10^{-1}	8.68×10^{-5}	13	58	6.06
32^3	0.4	6.09×10^{-1}	2.94×10^{-4}	16	62	75.04
50^3	0.4	6.45×10^{-1}	1.99×10^{-4}	18	126	412.66
64^3	0.1	6.58×10^{-1}	3.34×10^{-3}	17	97	987.12
80^3	0.1	6.65×10^{-1}	4.31×10^{-3}	17	102	1,135.83
100^3	0.1	6.70×10^{-1}	4.91×10^{-3}	17	119	2,436.17
128^3	0.1	6.73×10^{-1}	3.49×10^{-3}	17	169	9,077.08

Subsequently, we run the method for various values of the fractional derivative orders. The results are summarized in Table 6.5. As one can observe, the constraint matrix becomes ill-conditioned when β is close to 1, due to the scaling factor in the definition of the Riesz derivative $\left(\text{that is, } \frac{-1}{2\cos(\frac{\beta\pi}{2})}\right)$. In turn, this results in an increase of the PCG iterations in the case where $\beta = 1.1$.

Table 6.5: Inequalities on both variables: Varying fractional derivative orders (with $N = 32^4$, $y_a = -4$, $u_a = -350$, $\gamma = 10^{-4}$).

α	β	δ	$\mathcal{E}_{L^2}(y-\bar{y})$	Dual Inf.	Iter. PCG	Iter. ADMM	Time (s)
0.1	1.3	0.4	6.46×10^{-1}	1.60×10^{-4}	17	126	380.75
0.3	1.3	0.4	6.46×10^{-1}	2.56×10^{-4}	17	126	385.96
0.5	1.3	0.4	5.12×10^{-1}	2.83×10^{-4}	18	126	408.47
0.9	1.3	0.4	6.44×10^{-1}	3.10×10^{-4}	19	125	419.46
0.7	1.1	0.4	6.48×10^{-1}	1.21×10^{-3}	30	100	508.59
0.7	1.5	0.1	7.79×10^{-1}	4.23×10^{-4}	15	96	275.03
0.7	1.7	0.4	1.04×10^{0}	2.46×10^{-4}	13	113	275.21
0.7	1.9	0.1	1.36×10^{0}	1.36×10^{-3}	8	108	180.85

Finally, we present the runs of the method for various values of the regularization parameter γ. We note at this point that as γ is changed, the solution of the equality constrained problem is significantly altered. In light of this, we adjust the inequality constraints of the problem for each value of γ, in order to ensure that the optimal solution will lie *strictly* within the bounds. That way, we are able to compare the convergence behavior of ADMM, for instances with different regularization values, γ. The results are summarized in Table 6.6.

Table 6.6: Inequalities on both variables: Varying regularization (with $N = 50^3$, $\alpha = 0.7$, $\beta = 1.3$).

γ	y_a	u_a	δ	$\mathcal{E}_{L^2}(y-\bar{y})$	Dual Inf.	Iter. PCG	Iter. ADMM	Time (s)
10^{-2}	-2	-100	0.1	1.77×10^{0}(*)	1.38×10^{-3}	11	87	190.99
10^{-4}	-7	-400	0.4	5.60×10^{-1}(*)	4.88×10^{-3}	10	36	74.20
10^{-6}	-9	$-2,800$	10	1.28×10^{-1}(*)	6.03×10^{-4}	8	47	71.13
10^{-8}	-9	$-4,000$	100	1.13×10^{-1}(*)	2.18×10^{-4}	6	32	44.70
10^{-10}	-9	$-4,000$	100	1.13×10^{-1}(*)	2.18×10^{-4}	5	32	40.77

We can observe that the proposed approach is sufficiently robust with respect to the problem parameters. The linear systems that have to be solved within ADMM require a small number of PCG iterations for a wide range of parameter choices. Furthermore, ADMM achieves convergence to a 4-digit accurate primal solution in a reasonable number of iterations, making the method overall efficient. In light of the generality of the approach (established in Section 3), the numerical results are very promising, and we conjecture that the proposed method can be equally effective for a very wide range of FDE optimization problems.

7. Conclusions. In this paper, we proposed the use of an Alternating Direction Method of Multipliers, for the solution of a large class of PDE-constrained convex quadratic optimization problems. Firstly, under some general assumptions, and by using the theory of Generalized Locally Toeplitz sequences, we showed that the linear system arising at every ADMM iteration preserves the GLT structure of the PDE constraints. We then associated a symbol to the aforementioned linear system while providing and analyzing some alternatives for preconditioning it efficiently. Subsequently, we focused on solving two-dimensional, time-dependent FDE-constrained optimization problems, with box constraints on the state and/or control variables. Using the Grünwald–Letnikov finite difference method, and by employing a discretize-then-optimize approach, we solved the resulting problem in the discretized variables. Given the underlying structure of such discretized problems, we designed a recursive linear algebra based on FFTs, using which we solved the associated ADMM linear systems through a Krylov subspace solver alongside a multilevel circulant preconditioner. We demonstrated how one can restrict the storage requirements to order of N (where N is the grid size), while requiring only $O(N \log N)$ operations for every iteration of the Krylov solver. We further verified that the number of Krylov iterations required at each ADMM iteration is independent of the grid size. As a proof of concept, we implemented the method, and demonstrated its scalability, efficiency, and generality.

While the paper is focused on a special type of FDE optimization problems, we have provided a suitable methodology that has a significantly wider range of applicability. As a future research direction, we would like to employ the method, and the associated preconditioners, to various extensions of the current model, by allowing non-constant diffusion coefficients, employing higher-order discretization methods, or by solving FDEs posed in higher space-time dimensions.

Acknowledgements. SP acknowledges financial support from a Principal's Career Development PhD scholarship at the University of Edinburgh, as well as a scholarship from A. G. Leventis Foundation. JWP acknowledges support from the Engineering and Physical Sciences Research Council (EPSRC) grant EP/S027785/1 and a Fellowship of The Alan Turing Institute. SL acknowledges financial support from a School of Mathematics PhD studentship at the University of Edinburgh, and JG acknowledges support from the EPSRC grant EP/N019652/1.

REFERENCES

[1] H. Bart, I. Gohberg, and M. A. Kaashoek, *Toeplitz Matrices and Linear Systems*, I. Gohberg (ed.), Toeplitz Centennial. Operator Theory: Advances and Applications, Vol. 4. Birkhäuser, Basel, 1982.
[2] M. Benzi, G. H. Golub, and J. Liesen, *Numerical Solutions of Saddle Point Problems*, Acta Numer., Vol. 14 (2005), pp. 1–137.
[3] D. Bertaccini and F. Durastante, *Limited Memory Block Preconditioners for Fast Solution of Fractional Partial Differential Equations*, J. Sci. Comp., Vol. 77 (2018), pp. 950–970.
[4] V. I. Bogachev, *Measure Theory (Volume I)*, Springer-Verlag Berlin Heidenberg, 2007.
[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, Found. Trends Mach. Learn., Vol. 3 (2010), pp. 1–122.
[6] J. Bunch, *Stability of Methods for Solving Toeplitz Systems of Equations*, SIAM J. Sci. Stat. Comp., Vol. 6 (1985), pp. 349–364.
[7] B. A. Carreras, V. E. Lynch, and G. M. Zaslavsky, *Anomalous Diffusion and Exit Time Distribution of Particle Tracers in Plasma Turbulance Models*, Phys. Plasma, Vol. 8 (2001), pp. 5096–5103.
[8] R. H. Chan, *Circulant Preconditioners for Hermitian Toeplitz Systems*, SIAM J. Mat. Anal. Appl., Vol. 10 (1989), pp. 542–550.
[9] R. H. Chan, *An Introduction to Iterative Toeplitz Systems*, SIAM, 2007.
[10] R. H. Chan and X. Q. Jin, *A Family of Block Preconditioners for Block Systems*, SIAM J. Sci. Stat. Comp., Vol. 13 (1992), pp. 1218–1235.
[11] R. H. Chan, J. G. Nagy, and R. J. Plemmons, *Circulant Preconditioned Toeplitz Least Squares Iterations*, SIAM J. Mat. Anal. Appl., Vol. 15 (1994), pp. 80–97.

[12] R. H. Chan and G. Strang, *Toeplitz Equations by Conjugate Gradient with Circulant Preconditioner*, SIAM J. Sci. Stat. Comp., Vol. 10 (1989), pp. 104–119.
[13] R. H. Chan and C. K. Wong, *Best-Conditioned Circulant Preconditioners*, Linear Alg. Appl., Vol. 218 (1995), pp. 205–211.
[14] T. F. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Stat. Comp., Vol. 9 (1988), pp. 766–771.
[15] T. F. Chan and J. A. Olkin, *Circulant Preconditioners for Toeplitz-Block Matrices*, Numerical Alg., Vol. 6 (1994), pp. 89–101.
[16] S. Chandrasekaran, M. Gu, X. Sun, J. Xia, and J. Zhu, *A Superfast Algorithm for Toeplitz Systems of Linear Equations*, SIAM J. Mat. Anal. Appl., Vol. 29 (2007), pp. 1247–1266.
[17] S. Cipolla and F. Durastante, *Fractional PDE Constrained Optimization: An Optimize-then-Discretize Approach with L–BFGS and Approximate Inverse Preconditioning*, Appl. Numer. Math., Vol. 123 (2018), pp. 43–57.
[18] P. J. Davis, *Circulant Matrices. Second Edition*, AMS Chelsea Publishing, 1994.
[19] W. Deng, *Finite Element Method for the Space and Time Fractional Fokker–Planck Equation*, SIAM J. Numer. Anal., Vol. 47 (2008), pp. 204–226.
[20] W. Deng and W. Yin, *On Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers*, J. Sci. Comp., Vol. 66 (2016), pp. 889–916.
[21] K. Diethelm, N. J. Ford, A. D. Freed, and Y. Luchko, *Algorithms for the Fractional Calculus: A Selection of Numerical Methods*, Comput. Method Appl. Math., Vol. 194 (2005), pp. 743–773.
[22] H. Ding and Y. Q. Li, C. Chen, *Higher-Order Algorithms for Riesz Derivative and Their Applications (I)*, Abstr. Appl. Anal., Article ID 653797 (2014).
[23] S. Dolgov, J. W. Pearson, D. V. Savostyanov, and M. Stoll, *Fast Tensor Product Solvers for Optimization Problems with Fractional Differential Equations as Constraints*, Appl. Math. Comp., Vol. 273 (2016), pp. 604–623.
[24] M. Donatelli, M. Mazza, and S. Serra-Capizzano, *Spectral Analysis and Structure Preserving Preconditioners for Fractional Diffusion Equations*, J. Comp. Phys., Vol. 307 (2017), pp. 262–279.
[25] M. Donatelli, M. Mazza, and S. Serra-Capizzano, *Spectral Analysis and Multigrid Methods for Finite Volume Approximations of Space-Fractional Diffusion Equations*, SIAM J. Sci. Comp., Vol. 40 (2018), pp. A4007–A4039.
[26] A. Dorostkar, M. Neytcheva, and B. Lund, *Numerical and Computational Aspects of Some Block-Preconditioners for Saddle Point Systems*, Parallel Comp., Vol. 49 (2015), pp. 164–178.
[27] A. Dorostkar, M. Neytcheva, and S. Serra-Capizzano, *Schur Complement Matrix and its (Elementwise) Approximation: A Spectral Analysis Based on GLT Sequences*, Large-Scale Sci. Comp. (LSSC 2015), (2015), pp. 419–426.
[28] A. Dorostkar, M. Neytcheva, and S. Serra-Capizzano, *Spectral Analysis of Coupled PDEs and of their Schur Complements via Generalized Locally Toeplitz Sequences in 2D*, Comput. Meth. Appl. Mech. Engrg., Vol. 309 (2016), pp. 74–105.
[29] F. Durastante and S. Cipolla, *Fractional PDE Constrained Optimization: Box and Sparse Constrained Problems, in Numerical Methods for Optimal Control Problems (M. Falcone, R. Ferretti, L. Grune, W. M. McEneaney, eds.)*, Springer, 2018.
[30] J. Eckstein, *Nonlinear Proximal Point Algorithms using Bregman Functions, with Applications to Convex Programming*, Math. Op. Res., Vol. 18 (1993), pp. 202–226.
[31] V. J. Ervin, N. Heuer, and J. P. Roop, *Numerical Approximation of a Time Dependent, Nonlinear, Space-Fractional Diffusion Equation*, SIAM J. Numer. Anal., Vol. 45 (2007), pp. 572–591.
[32] Z. Fang, M. K. Ng, and H. W. Sun, *Circulant Preconditioners for a kind of Spatial Fractional Diffusion Equations*, Numer. Alg., Vol. 82 (2019), pp. 729–747.
[33] C. Garoni, M. Mazza, and S. Serra-Capizzano, *Block Generalized Locally Toeplitz Sequences: From the Theory to the Applications*, Axioms, Vol. 7 (2018), 49 (29 pages).
[34] C. Garoni and S. Serra-Capizzano, *Generalized Locally Toeplitz Sequences: Theory and Applications (Volume I)*, Springer International Publishing, 2017.
[35] C. Garoni and S. Serra-Capizzano, *Generalized Locally Toeplitz Sequences: Theory and Applications (Volume II)*, Springer International Publishing, 2018.
[36] R. Glowinski, *Numerical Methods for Non-linear Variational Problems*, Springer Series in Computational Physics, 1984.
[37] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, *Fast Alternating Direction Optimization Methods*, SIAM J. Imaging Sci., Vol. 7 (2014), pp. 1588–1623.
[38] J. Gondzio, *Interior Point Methods 25 Years Later*, Eur. J. Oper. Res., Vol. 218 (2013),

pp. 587–601.

[39] M. Hazewinkel, *Encyclopedia of Mathematics: A-Integral – Coordinates*, Springer, 1995.

[40] G. Heidel, V. Khoromskaia, B. N. Khoromskij, and V. Schulz, *Tensor Product Method for Fast Solution of Optimal Control Problems with Fractional Multidimensional Laplacian in Constraints*, 2018, https://arxiv.org/abs/arxiv:1809.01971.

[41] I. C. F. Ipsen, *A Note on Preconditioning Non-Symmetric Matrices*, SIAM J. Sci. Comp., Vol. 23 (2001), pp. 1050–1051.

[42] X. Q. Jin, *A Preconditioner for Constrained and Weighted Least Squares Problems with Toeplitz Structure*, BIT Num. Math., Vol. 36 (1996), pp. 101–109.

[43] R. C. Koeller, *Applications of Fractional Calculus to the Theory of Viscoelasticity*, J. Appl. Mech., Vol. 51 (1984), pp. 299–307.

[44] S. L. Lei, X. Chen, and X. Zhang, *FFT-Based Preconditioners for Toeplitz-Block Least Squares Problems*, SIAM J. Numer. Anal., Vol. 30 (1992), pp. 1740–1768.

[45] S. L. Lei, X. Chen, and X. Zhang, *Multilevel Circulant Preconditioners for High-Dimensional Fractional Diffusion Equations*, East Asian J. Appl. Math., Vol. 6 (2016), pp. 109–130.

[46] S. L. Lei and H. W. Sun, *A Circulant Preconditioner for Fractional Diffusion Equations*, J. Comp. Phys., Vol. 242 (2013), pp. 715–725.

[47] N. Levinson, *The Wiener RMS Error Criterion in Filter Design and Prediction*, J. Math. Phys., Vol. 25 (1947), pp. 261–278.

[48] X. Lin, M. K. Ng, and H. W. Sun, *A Splitting Preconditioner for Toeplitz-like Linear Systems Arising from Fractional Diffusion Equations*, SIAM J. Mat. Anal. Appl., Vol. 38 (2017), pp. 1580–1614.

[49] L. Lu, X. Wang, and G. Wang, *Alternating Direction Method of Multipliers for Separable Convex Optimization of Real Functions in Complex Variables*, Math. Probl. Eng., Article ID 104531, 14 pages, doi: http://dx.doi.org/10.1155/2015/104531 (2015).

[50] M. M. Meerschaert, H.-P. Scheffler, and C. Tadjeran, *Finite Difference Methods for Two-Dimensional Fractional Dispersion Equation*, J. Comput. Phys., Vol. 211 (2006), pp. 249–261.

[51] M. M. Meerschaert and C. Tadjeran, *Finite Difference Approximations for Fractional Advection–Dispersion Flow Equations*, J. Comput. Appl. Math., Vol. 172 (2004), pp. 65–77.

[52] M. M. Meerschaert and C. Tadjeran, *Finite Difference Approximations for Two-Sided Space-Fractional Partial Differential Equations*, Appl. Numer. Math., Vol. 56 (2006), pp. 80–90.

[53] H. Moghaderi, M. Dehghan, M. Donatelli, and M. Mazza, *Spectral Analysis and Multigrid Preconditioner for Two-Dimensional Space-Fractional Diffusion Equations*, J. Comp. Phys., Vol. 350 (2017), pp. 992–1011.

[54] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A Note on Preconditioning for Indefinite Linear Systems*, SIAM J. Sci. Comp., Vol. 21 (2000), pp. 1969–1972.

[55] M. Neytcheva, *On Element-by-Element Schur Complement Approximations*, Linear Alg. Appl., Vol. 434 (2011), pp. 2308–2324.

[56] M. K. Ng, *Iterative Methods for Toeplitz Systems (Numerical Mathematics and Scientific Computation)*, Oxford University Press, New York, 2004.

[57] Y. Notay, *A New Analysis of Block Preconditioners for Saddle Point Problems*, SIAM J. Mat. Anal. and Appl., Vol. 35 (2014), pp. 143–173.

[58] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-Robust Preconditioners for Time-Dependent PDE-Constrained Optimization Problems*, SIAM J. Mat. Anal. Appl., Vol. 33 (2012), pp. 1126–1152.

[59] J. W. Pearson and A. J. Wathen, *A New Approximation of the Schur Complement in Preconditioners for PDE-constrained Optimization*, Numer. Linear Alg. Appl., Vol. 19 (2012), pp. 816–829.

[60] J. W. Pearson and A. J. Wathen, *Fast Iterative Solvers for Convection-Diffusion Control Problems*, Electon. Trans. Numer. Anal., Vol. 40 (2013), pp. 294–310.

[61] I. Perguia and V. Simoncini, *Block–Diagonal and Indefinite Symmetric Preconditioners for Mixed Finite Element Formulations*, Numer. Linear Alg., Vol. 7 (2000), pp. 585–616.

[62] I. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of their Solution and Some of their Applications*, Mathematics in Science and Engineering, Volume 198 (1st Edition), Elsevier, 1999.

[63] S. S. Reddi, *Eigenvector Properties of Toeplitz Matrices and their Application to Spectral Analysis of Time Series*, Signal Processing, Vol. 7 (1984), pp. 45–56.

[64] Y. Saad and M. H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving*

Nonsymmetric Linear Systems, SIAM J. Sci. Stat. Comp., Vol. 7 (1986), pp. 856–869.

[65] S. G. Samko, A. A. Kilbas, and O. O. I. Marichev, *Fractional Integrals and Derivatives*, Gordon and Breach Science Publishers Yverdon, 1993.

[66] R. Scherer, L. K. Shyam, Y. Tang, and J. Huang, *The Grünwald–Letnikov Method for Fractional Diffusion Equations*, Computers Math. Appl., Vol. 62 (2011), pp. 902–917.

[67] S. Serra-Capizzano and E. Tyrtyshnikov, *Any Circulant-Like Preconditioner for Multilevel Matrices is Not Superlinear*, SIAM J. Mat. Anal. Appl., Vol. 21 (2000), pp. 431–439.

[68] C. Siefert and E. de Sturler, *Preconditioners for Generalized Saddle-Point Problems*, SIAM J. Numer. Anal., Vol. 44 (2006), pp. 1275–1296.

[69] D. Silvester and A. Wathen, *Fast Iterative Solution of Stabilized Stokes Systems, Part II: Using General Block Preconditioners*, SIAM J. Numer. Anal., Vol. 31 (1994), pp. 1352–1367.

[70] P. Tilli, *Locally Toeplitz Sequences: Spectral Properties and Applications*, Linear Alg. Appl., Vol. 278 (1998), pp. 91–120.

[71] E. E. Tyrtyshnikov, *Optimal and Superoptimal Circulant Preconditioners*, SIAM J. Mat. Anal. Appl., Vol. 13 (1992), pp. 459–473.

[72] H. Wang, K. Wang, and T. Sircar, *A Direct $O(N \log^2 N)$ Finite Difference Method for Fractional Diffusion Equations*, J. Comp. Phys., Vol. 229 (2010), pp. 8095–8104.

[73] J. Xia, Y. Xi, and M. Gu, *A Superfast Structured Solver for Toeplitz Linear Systems via Randomized Sampling*, SIAM J. Mat. Anal. Appl., Vol. 33 (2012), pp. 837–858.

[74] H. Xiao and W. B. Wu, *Covariance Matrix Estimation for Stationary Time Series*, Ann. Statist., Vol. 40 (2012), pp. 466–493.

[75] K. Ye and L. H. Lim, *Algorithms for Structured Matrix–Vector Product of Optimal Bilinear Complexity*, IEEE ITW, (2016).

[76] G. M. Zaslavsky, D. Stevens, and H. Weitzner, *Self-Similar Transport in Incomplete Chaos*, Phys. Rev. E, Vol. 48 (1993), pp. 1683–1694.

[77] M.-L. Zeng and G.-F. Zhang, *An MHSS-Like Iteration Method for Two-by-Two Linear Systems with Application to FDE Optimization Problems*, J. Comput. Appl. Math., Vol. 352 (2019), pp. 368–381.