Sparse grid approximation of the Riccati equation

Helmut Harbrecht, Ilja Kalmykov

Departement Mathematik und Informatik
Fachbereich Mathematik
Universität Basel
CH-4051 Basel

Preprint No. 2024-01
January 2024

dmi.unibas.ch

SPARSE GRID APPROXIMATION OF THE RICCATI EQUATION

HELMUT HARBRECHT AND ILJA KALMYKOV

Abstract. In this article, we study the sparse grid discretization for the numerical solution of the algebraic Riccati equation (ARE). This approach is of particular interest for the solution of large scale AREs. Such AREs arise, for example, from the discretization of the operator Riccati equations associated with the linear quadratic control of systems evolving in a Hilbert space H. Following [4, 45], we formulate the ARE as a nonlinear operator equation on the space of Hilbert–Schmidt operators and derive the matrix equation for the sparse grid discretization. Provided that $\mathcal{O}(N)$ degrees of freedom are used to discretize the space H, the sparse grid approximation of the ARE requires $\mathcal{O}(N \log N)$ degrees of freedom. Especially, we propose an algorithm that evaluates the approximated ARE with $\mathcal{O}(N^{3/2})$ operations. This considerably reduces the cost of solving the ARE compared to the $\mathcal{O}(N^2)$ memory requirement and $\mathcal{O}(N^3)$ complexity of the regular tensor product discretization. Numerical results are presented to validate the approach.

1. Introduction

Many problems in mathematics, physics and engineering can be traced back to the solution of a Riccati equation. Well-known examples are linear quadratic (LQ) optimal control problems ([5, 9, 36, 39]) and optimal linear filtering problems ([32, 47]). Other important applications include model reduction ([22, 31]), scattering theory ([40]), optimal placement of sensors and actuators ([14, 33]), or resilience analysis of critical infrastructure ([49]). Of particular importantance is the case where the Riccati equation admits a stationary solution. Such setting arises, for example, in the infinite-horizon LQ problem. The corresponding nondynamical equation is called the algebraic Riccati equation (ARE) (see [9, Proposition 2.2, p. 482] or [37, Theorem 2.3.3.1, p. 134] for example).

Dynamical systems evolving in a Hilbert space H generally lead to an infinite-dimensional ARE. Hereby, H is usually referred to as the state space. The solution of such equations depends on various approximation and truncation techniques. A number of methods is available here – see e.g. [8] or [10] for a survey. We introduce the algebraic Riccati equation in Section 2. There, we closely follow the formulation presented, for example, in [4, 44, 45, 47], where the authors consider the ARE as an abstract nonlinear operator equation in the space of Hilbert–Schmidt operators on H.

One common approach is to project the Riccati equation onto a sequence of finite-dimensional subspaces of H. In this way, the exact solution is approximated by a sequence of solutions of finite-dimensional Riccati equations. Then, provided that we use $\mathcal{O}(N)$ degrees of freedom for the discretization of the state space H, the

Date: December 30, 2023.

discretization of an operator acting on H by a regular tensor product approach requires $\mathcal{O}(N^2)$ degrees of freedom. This generally leads to $\mathcal{O}(N^3)$ overall complexity for the solution of the approximate ARE.

The cubic complexity and the quadratic growth of the memory requirements are major bottlenecks in the numerical treatment of large scale operator Riccati equations. Discretization in the regular tensor product space becomes prohibitively expensive, if not even impossible, at least for $d \geqslant 3$ spatial dimensions. This is one example of a more general problem known as curse of dimensionality. At the same time however, theoretical results on the regularity of the Riccati operator (see e.g. [33, 36, 41]) indicate that more efficient numerical methods can be developed.

Various approaches, such as multigrid methods ([20, 46]) or $\mathcal{H}$-matrices ([21]), have been studied to overcome the $\mathcal{O}(N^3)$ complexity (see also [23, 42]). In the present article, we discretize the Riccati equation by using a *sparse grid* (SG) – a numerical technique, which allows to overcome the curse of dimensionality to a certain extent. The construction of the SG space is based on the cost-benefit analysis of the finite-dimensional subspaces of H ([13, 19]). Assuming a certain regularity, this method requires only $\mathcal{O}(N \log N)$ degrees of freedom with essentially no loss of the approximation power. We will introduce the sparse grid space and discuss the corresponding discretization of the ARE in Section 3.

An immediate advantage of the SG method is the $\mathcal{O}(N \log N)$ memory requirement, which is nearly linear. Nonetheless, a straightforward evaluation of the ARE approximated in the SG space results in the complexity of $\mathcal{O}(N^2 \log N)$ (see [27]). Interestingly, this can be improved considerably. In the present article, we derive an algorithm that requires $\mathcal{O}(N^{3/2})$ operations and prove its complexity bound. This is our main result and subject of Section 4.

The $\mathcal{O}(N^{3/2})$ complexity is the square root of the $\mathcal{O}(N^3)$ cost of the regular tensor product approach. This result, combined with the $\mathcal{O}(N \log N)$ memory usage, presents the SG discretization as a viable alternative for solving large-scale algebraic Riccati equations. We demonstrate our approach by numerical experiments in Section 5 in order to validate our theoretical findings. Finally, in Section 6, we state concluding remarks.

2. Algebraic Riccati equation in Hilbert spaces

This section introduces the algebraic Riccati equation on spaces of Hilbert–Schmidt operators. We closely follow the presentations in [4, Chapter II, Section 3.3] and [44, 45]. This approach is also described, for example, in [47].

Suppose we are given Hilbert spaces $Z \subset H \subset Z'$, whereby Z' is the dual space of Z, and a linear operator $A \in \mathcal{L}(Z, Z')$ with the following properties:

(i) Z is densely, continuously and compactly (compare e.g. [24, Chapter 6]) embedded into H,
(ii) the operator A is Z-elliptic (cf. [24, p. 154]), i.e. for the inner product $\langle \cdot, \cdot \rangle_H$ on H there holds

$$\exists \alpha > 0 : \langle Au, u \rangle_H \geqslant \alpha \|u\|_Z^2 \text{ for all } u \in Z.$$

Let $\mathcal{H} = HS(H)$ denote the spaces of Hilbert–Schmidt operators on H and $[\cdot, \cdot]_{\mathcal{H}}$ the corresponding inner product. Furthermore, let $\mathcal{K}$ be the cone of self-adjoint and positive definite operators from $\mathcal{H}$, i.e.

$$\mathcal{K} = \{\Phi \in \mathcal{H} : \Phi = \Phi^\star,\ \Phi \geqslant 0\}.$$

We are interested in the following algebraic Riccati equation in the space $\mathcal{H}$

$$A^\star P + PA + P^2 = Q, \quad Q \in \mathcal{K}. \tag{ARE}$$

Equations of this form appear, among others, in the context of linear quadratic optimal control problems with infinite time horizon. As an example, let us take the spaces $Z = H^1_0(\Omega)$, $H = L^2(\Omega)$, and an $H^1_0(\Omega)$-elliptic, second order differential operator

$$A \in \mathcal{L}\big(H^1_0(\Omega), H^{-1}(\Omega)\big),\ Az = \sum_{i,j=1}^{d} \partial_i(a_{i,j}(x)\partial_j z) + \sum_{i=1}^{d} b_i(x)\partial_i z + c(x)z. \tag{1}$$

Then, we consider the abstract differential equation

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t} z(t) = Az(t) + u(t), & t \in (0,T], \\ z(0) = z_0, & z_0 \in H, \end{cases} \tag{2}$$

and an observation operator $C \in \mathcal{L}(H,H)$ with the property $C^\star C \in \mathcal{H}$. Our goal is to find a function $u \in L^2\big((0,T);H\big)$ which minimizes the quadratic cost functional

$$\int_0^\infty \left\{\|Cz(t)\|_H^2 + \|u(t)\|_H^2\right\} \mathrm{d}t.$$

It is well known that the minimizer u_{opt} is given by the feedback formula $u_{\mathrm{opt}}(t) = Pz_{\mathrm{opt}}(t)$ (cf. [9, Part V, Chapter 1], [18, 37] and [39, Chapter III, Section 4]), where z_{opt} is the solution of the closed loop system associated to (2) (see e.g. [9, p. 480]), and P satisfies (ARE) for the data $Q = C^\star C$.

The discussion of the solvability of (ARE) can be conducted within the framework of the theory of monotone operators on Banach spaces. To translate (ARE) into this setting, let $HS(Z',H)$ and $HS(H,Z)$ be the spaces of Hilbert-Schmidt operators from Z' to H and from H to Z, respectively. Moreover, we define

$$\mathcal{G} = HS(Z',H) \cap HS(H,Z).$$

Let us write the linear and quadratic parts of (ARE) as the operators

$$\mathcal{A} : \mathcal{D}(\mathcal{A}) \to \mathcal{G}',\ \mathcal{A}(\varPhi) = A^\star\varPhi + \varPhi A,\ \mathcal{D}(\mathcal{A}) = \{\varPhi \in \mathcal{G} : \mathcal{A}(\varPhi) \in \mathcal{H}\} \subset \mathcal{G},$$

and

$$\mathcal{B} : \mathcal{D}(\mathcal{B}) \to \mathcal{H},\ \mathcal{B}(\varPhi) = \varPhi^2,\ \mathcal{D}(\mathcal{B}) = \mathcal{K} \subset \mathcal{H}.$$

Here, we denote with $\mathcal{D}$ the domain of an operator.

With the above definitions at hand, the problem of solving (ARE) becomes a problem of finding the solution to the nonlinear operator equation

$$\mathcal{F}(P) = \mathcal{A}(P) + \mathcal{B}(P) = A^\star P + PA + P^2 = Q, \quad Q \in \mathcal{K}. \tag{3}$$

The existence and uniqueness of a solution to (3) is guaranteed by the following theorem.

Theorem 2.1. *([4, Theorem 3.9, p.91]) Let $Q \in \mathcal{K}$. Under the above hypotheses, there exists a unique $P \in \mathcal{H}$ such that*

$$\mathcal{F}(P) = Q,\ P = P^\star,\ P \geqslant 0,\ P \in \mathcal{G}.$$

Remark 2.2. *An optimal control problem usually involves a control operator B. The framework considered here can be generalised to such cases. To do so, we take the quadratic part*

$$\mathcal{B} : \mathcal{D}(\mathcal{B}) \to \mathcal{H},\ \mathcal{B}(\Phi) = \Phi BB^\star\Phi,\ \mathcal{D}(\mathcal{B}) = \mathcal{K} \subset \mathcal{H},$$

and assume that $\mathcal{B}$ is bounded from $\mathcal{K}$ to $\mathcal{H}$, monotone on $\mathcal{K}$, i.e.

$$\forall \Phi, \Psi \in \mathcal{K} : [\mathcal{B}(\Phi) - \mathcal{B}(\Psi), \Phi - \Psi]_{\mathcal{H}} \geqslant 0,$$

and satisfies $\mathcal{K} \subset [\mathrm{Id}_{\mathcal{H}} + \lambda\mathcal{B}](\mathcal{K})$ for every $\lambda > 0$. These assumptions are fulfilled, for example, if $BB^\star \in \mathcal{L}(H, H)$ (cf. [4, p. 94]). The corresponding algebraic Riccati equation becomes in this case

$$A^\star P + PA + PBB^\star P = C^\star C.$$

For the sake of clarity in presentation, we will stick to the case $B = \mathrm{Id}_H$.

3. Sparse grid discretization

Sparse grids are a numerical discretization technique that is of particular interest for high-dimensional problems. This section recalls the main ideas by following the presentation [51]. A detailed presentation can be found in [1, 13, 17, 19, 43], see also [12], [24, p. 260], [25, p. 280], and [26, 29, 30]. We will use sparse grids to construct appropriate ansatz spaces of finite dimensional operators for the approximation of (ARE).

Throughout the following, we denote by small bold letters, e.g. $\boldsymbol{i} \in \mathbb{N}^2$, a two-dimensional multi-index, i.e. $\boldsymbol{i} = (i_1, i_2)$. In contrast, cursive letters, e.g. $i \in \mathbb{N}$, are used as usual indices. If not stated otherwise, we assume $J \in \mathbb{N}$ and $i, i_1, j, j_1, k, \ell \in \{0, 1, \ldots, J\}$.

3.1. Construction of the sparse grid.

Suppose we are given a nested sequence of finite dimensional subspaces $Z_j \subset Z$, that is

$$Z_0 \subset Z_1 \subset Z_2 \subset \cdots \subset Z_J \subset Z.$$

Consider a space V with $Z \otimes Z \subset V$, whereby $\otimes$ denotes the algebraic tensor product, and the completion can be taken with respect to an appropriate norm. Our goal is to construct a finite dimensional subspace of V using the spaces Z_j.

In accordance with [13, 19, 29], let us introduce the hierarchical difference spaces W_j via

$$W_j := Z_j \ominus Z_{j-1},\ \text{where } Z_{-1} := \{0\},$$

and set $N_j := \dim W_j$. We will refer to the index j as *level*.

Assumption 3.1. *We shall assume that N_j behaves like an increasing geometric sequence. This is, for example, the case if the sequence $\{Z_j\}$ is constructed from dyadic subdivisions of a given coarse grid triangulation or tetrahedralization of the underlying domain. In this particular case, we obtain $|N_j| = \mathcal{O}(2^{dj})$.*

Let $W_{\boldsymbol{j}} = W_{(j_1, j_2)}$ denote the tensor product of the two spaces W_{j_1} and W_{j_2}

$$W_{\boldsymbol{j}} := W_{j_1} \otimes W_{j_2} = (Z_{j_1} \ominus Z_{j_1-1}) \otimes (Z_{j_2} \ominus Z_{j_2-1}).$$

The dimension of $W_{\boldsymbol{j}}$ is $N_{\boldsymbol{j}} := \dim W_{\boldsymbol{j}} = N_{j_1} N_{j_2}$. With these spaces at hand, we first introduce the full tensor product space V_J by the direct sum

$$V_J := \bigoplus_{j_1, j_2 \leqslant J} W_{(j_1, j_2)} = \bigoplus_{\|\boldsymbol{j}\|_\infty \leqslant J} W_{\boldsymbol{j}}. \tag{4}$$

The discretization on V_J suffers from the curse of dimensionality, i.e. the number of degrees of freedom in the space V_J is N_J^2. However, provided certain regularity, a function $f \in V$ can be approximated by sparse grids at essentially the same rate with only $\mathcal{O}(N_J \log N_J)$ degrees of freedom.

The idea of a *sparse grid* is to consider only those basis functions in the space V_J, which have a large contribution to the representation of a function $f \in V$ to be approximated, cf. [13, 19]. We denote the sparse grid function space with $\widehat{V}_J$ and give the following formal definition

$$\widehat{V}_J := \bigoplus_{j_1+j_2 \leqslant J} W_{(j_1,j_2)} = \bigoplus_{\|\boldsymbol{j}\|_1 \leqslant J} W_{\boldsymbol{j}}. \tag{5}$$

In order to illustrate this construction, we shall consider the representation of the sparse grid space $\widehat{V}_J$ in terms of the basis of the space Z_J. To this end, we assume the space Z_J to be spanned by some hierarchical basis $\{\phi_k\}_{k=1}^{N_J}$. Then, if the functions $\{\phi_{(j_1,j_2)}\}_{j_2=1}^{N_{j_1}}$ span the spaces W_{j_1}, that is

$$Z_{j_1} \ominus Z_{j_1-1} = W_{j_1} = \operatorname{span}\{\phi_{(j_1,j_2)} : j_2 = 1, \ldots, N_{j_1}\},$$

we can write

$$\widehat{V}_J = \operatorname{span}\left\{\phi_{(i_1,i_2)} \otimes \phi_{(j_1,j_2)} : i_1 + j_1 \leqslant J,\ i_2 = 1, \ldots, N_{i_1},\ j_2 = 1, \ldots, N_{j_1}\right\}. \tag{6}$$

By comparison of (4) and (5), one figures out that $\widehat{V}_J$ is obtained from V_J by discarding the hierarchical difference spaces with $j_1 + j_2 > J$. This construction leads to a much smaller number of degrees of freedom

$$\dim \widehat{V}_J = \mathcal{O}(N_J \log N_J).$$

In general, for sparse grids on m-fold tensor product spaces, there holds $\dim \widehat{V}_J = \mathcal{O}(N_J \log N_J^{m-1})$ while essentially no approximation power is lost provided that the function to be approximated exhibits extra smoothness in terms of bounded mixed derivatives. In other words, the exponential dependency on the dimension is only in the $\log N_J$ factor, which substantially reduces the dimension of the sparse grid space compared to the full tensor product space.

3.2. **Orthogonal projection onto the sparse grid.** In the following, our aim is to use operator spaces associated to $\widehat{V}_J$ to discretize (ARE). In order to achieve this, we first discuss the orthogonal projections onto the spaces $\widehat{V}_J$ in this subsection.

Let $\Pi_{Z_j} \in \mathcal{L}(H, H)$ denote the orthogonal projection with respect to $\|\cdot\|_H$ onto the space Z_j. We define

$$\Delta\Pi_{Z_j} := \Pi_{Z_j} - \Pi_{Z_{j-1}} = \Pi_{Z_i}\left(\mathrm{Id}_H - \Pi_{Z_{j-1}}\right), \tag{7}$$

where we set $\Pi_{Z_0} := 0$. The following lemma recalls some properties of the operators $\Delta\Pi_{Z_j}$.

Lemma 3.2. *The operators $\Delta\Pi_{Z_j} \in \mathcal{L}(H, H)$ are H-orthogonal projections with*

$$\operatorname{im}\left(\Delta\Pi_{Z_j}\right) = Z_j \cap Z_{j-1}^{\perp}, \quad \ker\left(\Delta\Pi_{Z_j}\right) = Z_j^{\perp} \oplus Z_{j-1}.$$

Proof. Using the definition (7), we have $\Delta\Pi_{Z_j} = \Delta\Pi_{Z_j}^{\star}$ and $\Delta\Pi_{Z_j}\Delta\Pi_{Z_j} = \Delta\Pi_{Z_j}$. This proves that $\Delta\Pi_{Z_j}$ is an orthogonal projection.

To characterize the image and the kernel of $\Delta\Pi_{Z_j}$, consider an arbitrary function $g \in \operatorname{im}\left(\Delta\Pi_{Z_j}\right)$. There holds

$$g = \Delta\Pi_{Z_j} g = \Pi_{Z_j}\left(\mathrm{Id}_Z - \Pi_{Z_{j-1}}\right) g.$$

This implies $g \in \operatorname{im}\left(\Delta\Pi_{Z_j}\right) \Leftrightarrow g \in Z_i \cap Z_{j-1}^{\perp}$ (see [16, Lemma 2 (d), p. 481]). Because $\Delta\Pi_{Z_j}$ is an orthogonal projection, we conclude that (cf. [11, Section 2.5])

$$\ker\left(\Delta\Pi_{Z_j}\right) = \left[\operatorname{im}\left(\Delta\Pi_{Z_j}\right)\right]^{\perp} = \left(Z_j \cap Z_{j-1}^{\perp}\right)^{\perp} = Z_j^{\perp} \oplus Z_{j-1}.$$

□

In the next step, let us recall the connection between the operator space $\mathcal{H}$ and the tensor product $Z \otimes Z$. To this end, we consider the scalar product

$$\langle g_1 \otimes g_2, h_1 \otimes h_2 \rangle := \langle g_1, h_1 \rangle_H \langle g_2, h_2 \rangle_H, \quad g_1, g_2, h_1, h_2 \in H,$$

and the induced norm $\|g\| = \sqrt{\langle g, g \rangle}$. We define $V^H := H \otimes_{\|\cdot\|} H \simeq Z \otimes_{\|\cdot\|} Z$, i.e. V^H is the tensor product of the Hilbert space H with itself (see [25, Section 4.5, p. 142] or [38, p. 20]). Recall that V^H is isometric to the space $\mathcal{H}$ (cf. [25, Lemma 4.119] or [3, p. 296]). This allows to construct orthogonal projections in $\mathcal{H}$ by using orthogonal projections in V^H, which are discussed in the next lemma.

Lemma 3.3. *Let $\Pi_{\widehat{V}_J}$ denote the V^H-orthogonal projection onto $\widehat{V}_J$. Then*

$$\Pi_{\widehat{V}_J} = \sum_{i+j \leqslant J} \Delta\Pi_{Z_i} \otimes \Delta\Pi_{Z_j} = \sum_{j \leqslant J} \Pi_{Z_{J-j+1}} \otimes \Delta\Pi_{Z_j} = \sum_{j \leqslant J} \Delta\Pi_{Z_k} \otimes \Pi_{Z_{J-j+1}}.$$

Proof. See [16, p. 514], [2, p. 321], and a similar result in [28]. □

3.3. **Discretization of the algebraic Riccati equation.** Our goal in this subsection is to derive a discrete version of (ARE). To this end, let $g, h \in H$ and consider the map

$$K(g \otimes h) = g\langle h, \cdot \rangle_H \in \mathcal{H}. \tag{8}$$

There holds $\|g \otimes h\| = \|g\langle h, \cdot\rangle_H\|_{\mathcal{H}}$ (see [25, Lemma 4.119] or [3, p. 296] for example), which means that K extends (by linearity and continuity) to a bijective isometry from V^H to $\mathcal{H}$. We shall denote this extension again by K.

Let us define for $\|\boldsymbol{i}\|_\infty \leqslant J$

$$\mathcal{W}_{\boldsymbol{i}} = \left\{\Phi\Pi_{W_{i_2}} : \Phi \in \mathcal{L}(W_{i_2}, W_{i_1})\right\} = K\left(W_{\boldsymbol{i}}\right),$$

and

$$\widehat{\mathcal{V}}_J = \bigoplus_{\|\boldsymbol{i}\|_1 \leqslant J} \mathcal{W}_{\boldsymbol{i}} = K\left(\widehat{V}_J\right), \quad \mathcal{V}_J = \bigoplus_{\|\boldsymbol{i}\|_\infty \leqslant J} \mathcal{W}_{\boldsymbol{i}} = K\left(V_J\right).$$

Then, to compute an approximate solution to (ARE), we are going to consider the Galerkin discretization with $\widehat{\mathcal{V}}_J$ as the trial and ansatz space. The following lemma and Theorem 3.5 allow to give an explicit representation of the discrete version of (ARE) in terms of the $\mathcal{H}$-orthogonal projection to $\widehat{\mathcal{V}}_J$.

Lemma 3.4. *Let G_1, G_2 be Hilbert spaces and $K \in \mathcal{L}(G_1, G_2)$ be a bijective isometry. Then, any orthogonal projection $\Pi_1 : G_1 \to G_1$ satisfies the identity $\Pi_1 = K^{-1}\Pi_2 K = K^\star \Pi_2 K$, whereby $\Pi_2 : G_2 \to G_2$ is the orthogonal projection such that*

$$\operatorname{im}(\Pi_2) = K\big(\Pi_1(G_1)\big), \ \ker(\Pi_2) = K\big(\Pi_1(G_1)^{\perp}\big).$$

Proof. K is a surjective isometry and therefore unitary (see [50, p. 259] for example). Recall that there holds $K^{-1} = K^\star$ for bijective unitary operators. By using this result, we obtain

$$\varPi_2^\star = \left(K\varPi_1K^\star\right)^\star = K\varPi_1K^\star = \varPi_2, \quad \varPi_2^2 = \left(K\varPi_1K^\star\right)^2 = K\varPi_1K^\star = \varPi_2,$$

which shows that $\varPi_2$ is an orthogonal projection on G_2.

In order to characterize the kernel and the image of $\varPi_2$, we compute

$$g \in \ker(\varPi_2) \;\Leftrightarrow\; 0 = \varPi_2 g = K\varPi_1K^\star g \;\Leftrightarrow\; K^\star g \in \ker(\varPi_1) \;\Leftrightarrow\; g \in K\big(\varPi_1(G_1)^\perp\big),$$

and

$$g \in \operatorname{im}(\varPi_2) \;\Leftrightarrow\; g = \varPi_2 g = K\varPi_1K^\star g \;\Leftrightarrow\; K^\star g \in \operatorname{im}(\varPi_1) \;\Leftrightarrow\; g \in K\big(\varPi_1(G_1)\big).$$

□

With the help of this lemma, we arrive at the following theorem.

Theorem 3.5. *Let $\varPhi \in \mathcal{H}$. Then, the $\mathcal{H}$-orthogonal projection of $\varPhi$ onto $\widehat{\mathcal{V}}_J$ is given by*

$$\varPi_{\widehat{\mathcal{V}}_J}(\varPhi) = \sum_{i+j\leqslant J} \Delta\varPi_{Z_i}\varPhi\,\Delta\varPi_{Z_j}.$$

Proof. Application of Lemma 3.4 yields

$$\varPi_{\widehat{\mathcal{V}}_J}(\varPhi) = \left[K\varPi K^\star\right](\varPhi) = K\left(\varPi K^\star\left(\sum_{i=1}^\infty \sigma_i u_i \langle v_i, \cdot\rangle_H\right)\right),$$

whereby $\sum_{i=1}^\infty \sigma_i u_i \langle v_i, \cdot\rangle_H$ is the singular value decomposition of $\varPhi$, and $\varPi$ is the orthogonal projection with

$$\operatorname{im}(\varPi) = K^\star\left(\varPi_{\widehat{\mathcal{V}}_J}(\mathcal{H})\right) = K^\star\left(\widehat{\mathcal{V}}_J\right) = \widehat{V}_J,\ \ker(\varPi) = K^\star\left(\varPi_{\widehat{\mathcal{V}}_J}(\mathcal{H})^\perp\right) = \widehat{V}_J^\perp.$$

Therefore $\varPi = \varPi_{\widehat{V}_J}$.

By continuity of $\varPi_{\widehat{V}_J}$ and Lemma 3.3, we have

$$\varPi_{\widehat{V}_J}\left(\sum_{i=1}^\infty \sigma_i u_i \otimes v_i\right) = \sum_{i=1}^\infty \sigma_i \varPi_{\widehat{V}_J}(u_i \otimes v_i) = \sum_{i=1}^\infty \sigma_i\left[\sum_{k+l\leqslant J} \Delta\varPi_{Z_k} \otimes \Delta\varPi_{Z_l}\right](u_i \otimes v_i)$$
$$= \sum_{k+l\leqslant J}\sum_{i=1}^\infty \sigma_i\big(\Delta\varPi_{Z_k}u_i\big) \otimes \big(\Delta\varPi_{Z_l}v_i\big).$$

Mapping back to $\mathcal{H}$ gives

$$K\left(\sum_{k+l\leqslant J}\sum_{i=1}^\infty \sigma_i\big(\Delta\varPi_{H_k}u_i\big) \otimes \big(\Delta\varPi_{H_l}v_i\big)\right) = \sum_{k+l\leqslant J}\sum_{i=1}^\infty \sigma_i\,\big(\Delta\varPi_{H_k}u_i\big)\big\langle \Delta\varPi_{H_l}v_i, \cdot\big\rangle_H$$
$$= \sum_{k+l\leqslant J} \Delta\varPi_{H_k}\sum_{i=1}^\infty \sigma_i u_i \big\langle v_i, \Delta\varPi_{H_l}\cdot\big\rangle_H = \sum_{k+l\leqslant J}\Delta\varPi_{H_k}\left(\sum_{i=1}^\infty \sigma_i u_i \langle v_i, \cdot\rangle_H\right)\Delta\varPi_{H_l}.$$

□

Corollary 3.6. *The Galerkin discretization of* (ARE) *is given by*

(ARE-P) *find $P \in \widehat{\mathcal{V}}_J$ such that* $\displaystyle\sum_{i+j\leqslant J} \Delta\varPi_{Z_i}\left(A^\star P + PA + P^2 - Q\right)\Delta\varPi_{Z_j} = 0.$

Proof. Recall that the Galerkin discretization is defined via

$$\text{find } P \in \widehat{\mathcal{V}}_J, \text{ so that } \forall \Psi \in \widehat{\mathcal{V}}_J : \left[A^\star P + PA + P^2 - Q, \Psi\right]_{\mathcal{H}} = 0,$$

compare e.g. [24, p. 184]. The second condition is equivalent to considering the $\mathcal{H}$-orthogonal projection of $A^\star P + PA + P^2 - Q$ onto $\widehat{\mathcal{V}}_J$. The statement follows now by applying Theorem 3.5.

□

3.4. **Matrix equation.** In this subsection, we derive a matrix equation associated to (ARE-P). To this end, similar to (6), let us fix a hierarchical basis $\{\phi_k\}_{k=1}^{N_J}$ of Z_J, and denote the corresponding bases of the hierarchical increments W_i with $\{\phi_{(i,j)}\}_{j=1}^{N_i}$, i.e.

$$Z_J = \operatorname{span}\{\phi_k : k = 1, \ldots, N_J\}, \quad W_i = \operatorname{span}\{\phi_{(i,j)} : j = 1, \ldots, N_i\},$$

and

$$\{\phi_k\}_{k=1}^{N_J} = \bigcup_{i \leqslant J} \{\phi_{(i,j)}\}_{j=1}^{N_i}.$$

We start the derivation of the matrix equation by introducing the prolongation operator (see [24, p. 184] for example) associated to $\{\phi_k\}_{k=1}^{N_J}$,

$$I : \mathbb{R}^{N_J} \to H, \quad (\alpha_1, \alpha_2, \ldots, \alpha_{N_J}) \mapsto \sum_{k=1}^{N_J} \alpha_k \phi_k.$$

Its adjoint $I^\star : H \simeq H' \to \left(\mathbb{R}^{N_J}\right)' \simeq \mathbb{R}^{N_J}$ with respect to $\langle \cdot, \cdot \rangle_H$ is called restriction operator. It is defined canonically by

$$(I^\star v, \alpha) = \langle v, I\alpha \rangle_H \quad \text{for all } v \in H, \ \alpha \in \mathbb{R}^{N_J}.$$

Here, $(\cdot, \cdot)$ denotes the Euclidean scalar product of $\mathbb{R}^{N_J}$. Recall that for the image of $I^\star$ there holds (see [24, p. 187] for example)

$$[I^\star v]_{k=1}^{N_J} = \left[\langle v, \varphi_k \rangle_H\right]_{k=1}^{N_J} \in \mathbb{R}^{N_J}. \tag{9}$$

In order to introduce the prolongation and restriction operators associated to the spaces W_j, let us use the identification

$$\mathbb{R}^{N_J} = \bigoplus_{j \leqslant J} \mathbb{R}^{N_j},$$

i.e. the spaces $\mathbb{R}^{N_j}$ correspond to the coefficients associated to the hierarchical increments W_j. The prolongations and restrictions for the spaces W_j are defined via

$$I_j : \mathbb{R}^{N_J} \to H, \quad I_j\alpha \mapsto \begin{cases} I\alpha, & \alpha \in \mathbb{R}^{N_j}, \\ 0, & \text{else}, \end{cases}$$

and

$$I_j^\star : H \to \mathbb{R}^{N_J}, \quad \left[I_j^\star v\right]_{k=1}^{N_J} = \begin{cases} \langle v, \varphi_k \rangle_H, & \varphi_k \in W_j, \\ 0, & \text{else}. \end{cases}$$

To simplify the notation in the following, we shall set

$$\widetilde{I}_n^\star := \sum_{j \leqslant n} I_j^\star, \quad \widetilde{I}_n := \sum_{j \leqslant n} I_j, \quad n \leqslant J. \tag{10}$$

Recall that I is bijective from $\mathbb{R}^{N_J}$ to Z_J, and $I^\star$ is bijective from Z_J to $\mathbb{R}^{N_J}$. Therefore, we can consider the inverse operators

$$I^{-1} : Z_J \to \mathbb{R}^{N_J}, \quad I^{-1}v = \alpha \Leftrightarrow I\alpha = v,$$

and

$$I^{-\star} : \mathbb{R}^{N_J} \to Z_J, \quad I^{-\star}\alpha = v \Leftrightarrow I^\star v = \alpha.$$

By using the bijectivity of I and $I^\star$, it is possible to identify an operator from $\mathcal{V}_J$ with a matrix. To be more precise, we will consider the following spaces of matrices

$$\mathbb{R}^{\mathcal{V}_J} := \bigoplus_{\|\boldsymbol{j}\|_\infty \leqslant J} \mathbb{R}^{N_{j_1} \times N_{j_2}} \simeq \mathcal{V}_J, \quad \mathbb{R}^{\widehat{\mathcal{V}}_J} := \bigoplus_{\|\boldsymbol{j}\|_1 \leqslant J} \mathbb{R}^{N_{j_1} \times N_{j_2}} \simeq \widehat{\mathcal{V}}_J.$$

In particular, $\mathbb{R}^{\widehat{\mathcal{V}}_J} \subset \mathbb{R}^{\mathcal{V}_J}$ is the linear subspace associated to the operators from $\widehat{\mathcal{V}}_J$.

As the first step to derive a matrix equation for (ARE-P), we give a characterization of the $\mathcal{H}$-orthogonal projection onto $\widehat{\mathcal{V}}_J$ in terms of the prolongation and restriction operators. This is the statement of Lemma 3.8. For the sake of clear representation, we prove the following intermediate result first.

Lemma 3.7. *For all $\ell \leqslant J$ there holds*

$$\widetilde{I}^\star_{J-\ell} \Pi_{\widehat{\mathcal{V}}_J}(\Phi) I_\ell = \widetilde{I}^\star_{J-\ell} \Phi I_\ell.$$

Proof. Recall that according to Lemma 3.2 there holds

$$\Delta \Pi_{Z_j} I_\ell = 0 \text{ for } j > \ell. \tag{11}$$

Moreover, by using the representation (9), we obtain $I^\star_\ell \Pi_{Z_j} = I^\star_\ell$ for $j \geqslant \ell$, and therefore

$$\widetilde{I}^\star_{J-\ell} \Pi_{Z_{J-j}} = \widetilde{I}^\star_{J-\ell} \text{ for } j \leqslant \ell. \tag{12}$$

By applying the formulas (11) and (12), we finally compute

$$\begin{aligned}
\widetilde{I}^\star_{J-\ell} \Pi_{\widehat{\mathcal{V}}_J}(\Phi) I_\ell &= \widetilde{I}^\star_{J-\ell} \sum_{j \leqslant J} \Pi_{Z_{J-j}} \Phi \, \Delta \Pi_{Z_j} I_\ell \overset{(11)}{=} \widetilde{I}^\star_{J-\ell} \sum_{j \leqslant \ell} \Pi_{Z_{J-j}} \Phi \, \Delta \Pi_{Z_j} I_\ell \\
&\overset{(12)}{=} \widetilde{I}^\star_{J-\ell} \sum_{j \leqslant \ell} \Phi \, \Delta \Pi_{Z_j} I_\ell = \widetilde{I}^\star_{J-\ell} \Phi \sum_{j \leqslant \ell} \Delta \Pi_{Z_j} I_\ell = \widetilde{I}^\star_{J-\ell} \Phi \Pi_{Z_\ell} I_\ell = \widetilde{I}^\star_{J-\ell} \Phi I_\ell.
\end{aligned}$$

□

Lemma 3.8. *Let $\Phi, \Psi \in \mathcal{H}$. Then*

$$\Pi_{\widehat{\mathcal{V}}_J}(\Phi) = \Pi_{\widehat{\mathcal{V}}_J}(\Psi) \tag{13}$$

if and only if

$$\forall k + \ell \leqslant J : I^\star_k \Phi I_\ell = I^\star_k \Psi I_\ell. \tag{14}$$

Proof. Assume that (14) is true and note that it is equivalent to

$$\forall \ell \leqslant J : \widetilde{I}^\star_{J-\ell} \Phi I_\ell = \widetilde{I}^\star_{J-\ell} \Psi I_\ell.$$

Consequently, according to Lemma 3.7, we obtain that (14) holds if and only if

$$\forall k + \ell \leqslant J : I^\star_k \Pi_{\widehat{\mathcal{V}}_J}(\Phi) I_\ell = I^\star_k \Pi_{\widehat{\mathcal{V}}_J}(\Psi) I_\ell$$

is true. Finally, by using that the mapping

$$\widehat{\mathcal{V}}_J \ni \Phi \mapsto \left[I_k^\star \Pi_{\widehat{\mathcal{V}}_J}(\Phi) I_\ell \right]_{k+\ell \leqslant J} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$$

is bijective from $\widehat{\mathcal{V}}_J$ to $\mathbb{R}^{\widehat{\mathcal{V}}_J}$, we conclude that (13) is equivalent to (14).

□

The condition (14) from Lemma 3.8 can be reformulated with the help of projections in the space $\mathbb{R}^{\mathcal{V}_J}$. This allows us to use the Galerkin matrices of the operators A and Q as well as the mass matrix in the discussion of the algorithms for the Riccati equation.

In order to define these projections, we first introduce the matrices $\Pi_k \in \mathbb{R}^{\mathcal{V}_J}$. To this end, let us write $X_{(j_1,j_2)}$ for the block of $X \in \mathbb{R}^{\mathcal{V}_J}$ corresponding to the subspace $\mathbb{R}^{N_{j_1} \times N_{j_2}}$, i.e. we have

$$X = \left[X_{(j_1,j_2)} \right]_{\|(j_1,j_2)\|_\infty \leqslant J}.$$

We define Π_k according to

$$[\Pi_k]_{(j_1,j_2)} = \begin{cases} \mathrm{Id}_{\mathbb{R}^{N_k}}, & j_1 = j_2 = k, \\ \mathbf{0}, & \text{else.} \end{cases}$$

Multiplication of a matrix $X \in \mathbb{R}^{\mathcal{V}_J}$ with Π_k from the left (right) yields the slice of the k^{th} row (column) of blocks of X, i.e.

$$[\Pi_k X]_{(j_1,j_2)} = \begin{cases} X_{(j_1,j_2)}, & j_1 = k, \\ \mathbf{0}, & \text{else,} \end{cases} \quad \text{and} \quad [X \Pi_k]_{(j_1,j_2)} = \begin{cases} X_{(j_1,j_2)}, & j_2 = k, \\ \mathbf{0}, & \text{else.} \end{cases}$$

Similar to (10), let us use the notation

$$\widetilde{\Pi}_n := \sum_{k \leqslant n} \Pi_k \quad \text{for } n \leqslant J. \tag{15}$$

With the help of (15), we then define the operator

$$\Pi_{SG} : \mathbb{R}^{\mathcal{V}_J} \to \mathbb{R}^{\mathcal{V}_J}, \quad X \mapsto \sum_{k \leqslant J} \Pi_k X \widetilde{\Pi}_{J-k} = \sum_{k \leqslant J} \widetilde{\Pi}_{J-k} X \Pi_k, \tag{16}$$

which is the orthogonal (with respect to the Frobenius norm) projection from $\mathbb{R}^{\mathcal{V}_J}$ onto $\mathbb{R}^{\widehat{\mathcal{V}}_J}$. In other words, Π_{SG} restricts the coefficients to the subspace $\mathbb{R}^{\widehat{\mathcal{V}}_J}$ which corresponds to a sparse grid. Figure 1 illustrates the action of Π_{SG}.

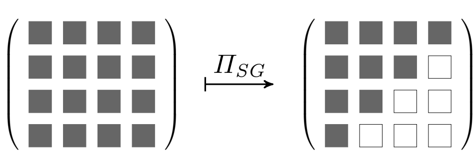

FIGURE 1. Application of Π_{SG} to an element from $\mathbb{R}^{\mathcal{V}_J}$ yields an element of $\mathbb{R}^{\widehat{\mathcal{V}}_J}$.

We obtain the following equivalent characterisation of the result of Lemma 3.8 using Π_{SG}.

Lemma 3.9. *Let $\Phi \in \mathcal{V}$. Denote by Ξ the mapping*

$$\Xi : \mathcal{V} \to \mathbb{R}^{\widehat{\mathcal{V}}_J}, \quad \Psi \mapsto \sum_{k+\ell \leqslant J} I_k^\star \Psi I_\ell.$$

Then, there holds

$$\Pi_{SG}(I^\star \Phi I) = \Xi(\Phi) = \Xi\big(\Pi_{\widehat{\mathcal{V}}_J}(\Phi)\big),$$

i.e. the following diagram commutes:

$$\begin{tikzcd}
\mathcal{V} \arrow[r, "\Phi \mapsto I^\star \Phi I"] \arrow[d, "\Pi_{\widehat{\mathcal{V}}_J}"'] \arrow[dr, "\Xi"] & \mathbb{R}^{\mathcal{V}_J} \arrow[d, "\Pi_{SG}"] \\
\widehat{\mathcal{V}}_J \arrow[r, "\simeq", "\Xi|_{\widehat{\mathcal{V}}_J}"'] & \mathbb{R}^{\widehat{\mathcal{V}}_J}
\end{tikzcd}$$

Proof. We compute

$$\Pi_{SG}(I^\star \Phi I) \overset{(\star)}{=} \underbrace{\sum_{k+\ell \leqslant J} I_k^\star \Phi I_\ell}_{\Xi(\Phi)} \overset{(\star\star)}{=} \underbrace{\sum_{k+\ell \leqslant J} I_k^\star \Pi_{\widehat{\mathcal{V}}_J}(\Phi) I_\ell}_{\Xi(\Pi_{\widehat{\mathcal{V}}_J}(\Phi))},$$

whereby $(\star)$ holds by the definition (16) of Π_{SG} and $(\star\star)$ due to Lemma 3.7. □

With the above results at hand, we can reformulate (ARE-P) in terms of matrices.

Theorem 3.10. *Let $Q \in \mathcal{K}$ and $A_J = I^\star A I$, $E_J = I^\star I$, $Q_J = I^\star Q I$. Then, $\Phi \in \widehat{\mathcal{V}}_J$ is a solution of* (ARE-P) *if and only if $\widehat{X} = I^{-1} \Phi I^{-\star} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$ is a solution of*

$$\Pi_{SG}\left(A_J \widehat{X} E_J + E_J \widehat{X} A_J + E_J \widehat{X} E_J \widehat{X} E_J - Q_J\right) = \mathbf{0}. \tag{ARE-M}$$

Proof. Let $\Phi \in \widehat{\mathcal{V}}_J$ be a solution of (ARE-P). According to Lemma 3.9, this is equivalent to

$$\Pi_{SG}\left(I^\star (A\Phi + \Phi A + \Phi^2 - Q) I\right) = \mathbf{0}. \tag{17}$$

Using $\widehat{X} = I^{-1} \Phi I^{-\star} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$, we can rewrite (ARE) as

$$A\Phi + \Phi A + \Phi^2 = A I \widehat{X} I^\star + I \widehat{X} I^\star A + I \widehat{X} I^\star I \widehat{X} I^\star.$$

Multiplying with $I^\star$ from the left and with I from the right gives

$$\begin{aligned}
&I^\star A I \widehat{X} I^\star I + I^\star I \widehat{X} I^\star A I + I^\star I \widehat{X} I^\star I \widehat{X} I^\star I \\
&= A_J \widehat{X} E_J + E_J \widehat{X} A_J + E_J \widehat{X} E_J \widehat{X} E_J.
\end{aligned}$$

Thus, we conclude that (17) is equivalent to (ARE-M). □

4. Evaluation of the algebraic Riccati equation in the sparse grid space

In this section, we present algorithms for evaluating the Riccati equation in the sparse grid space. We consider the linear part in Subsection 4.2 and the quadratic part in Subsection 4.3. Both parts are combined to derive the final approach for solving the algebraic Riccati equation in Subsection 4.4.

To better understand the results presented, recall that there is a close relation between expressions involving operators from $\widehat{\mathcal{V}}_J$ and those involving functions from $\widehat{V}_J$. This connection is rooted in the tensor product structure of the discretization space. Consequently, there are two equivalent approaches to discuss the algorithms for the sparse grids. In our presentation, we formulate the algorithms from the perspective of the operator approach using the associated discretization matrices. The relevant connection with the formulation in terms of function from the space $\widehat{V}_J$ is briefly outlined in Subsection 4.2.

4.1. Notation.

Throughout the following, let $S : Z \to Z'$ be a linear operator. Likewise to Theorem 3.10, we denote the Galerkin matrix of S with respect to the space Z_J by $S_J = I^\star S I \in \mathbb{R}^{N_J \times N_J}$, and use the notation

$$\widetilde{S}_k = \sum_{j \leqslant k} \widetilde{\Pi}_{J-j} S_J \Pi_j = \sum_{j \leqslant k} \widetilde{I}^\star_{J-j} S I_j,$$

which is analagous to that introduced in (10) and (15). In other words, $\widetilde{S}_k$ is the Galerkin matrix of S with respect to the space Z_k extended by 0 such that $\widetilde{S}_k \in \mathbb{R}^{N_J \times N_J}$ holds. Note that we also use the same notation for other matrices.

We will repeatedly use coefficient matrices associated with the elements from the space V_J, respectively from $\mathcal{V}_J$, in our presentation. To be more precise, let us consider a function $x \in V_J$. The associated coefficient matrix $X \in \mathbb{R}^{\mathcal{V}_J}$ is defined by

$$X = I^{-1} K(x) I^{-\star}.$$

Here, K denotes the isomorphism between V^H and $\mathcal{H}$ introduced in (8). We will refer to elements of $\widehat{V}_J$ as sparse grid functions and to elements of $\mathbb{R}^{\widehat{\mathcal{V}}_J}$ as sparse grid matrices. Note that the coefficients of matrices are stored in blocks corresponding to the spaces $W_{\boldsymbol{j}}$ (cf. [30] for example). In particular, this means that for $\widehat{X} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$ there holds

$$\forall j_1 + j_2 > J : \widehat{X}_{(j_1, j_2)} = \mathbf{0} \in \mathbb{R}^{N_{j_1} \times N_{j_2}}.$$

To simplify the notation, we make two assumptions throughout this section. First, we assume S to be self-adjoint. The results presented can be easily adapted to the non-self-adjoint case by replacing S with $S^\intercal$ where necessary. Second, for the sake of simplicity, we will restrict ourselves to the one-dimensional case, i.e. $d = 1$. This affects only the derivation of the complexity bounds, which however can be generalized in a straightforward manner to the desired setting in arbitrary dimension $d \in \mathbb{N}$ by adapting the proofs of Lemmas 4.3 and 4.7.

4.2. UniDir algorithm and evaluation of the linear part.

In this subsection, we discuss the evaluation of the linear part of the discretized Riccati equation by means of the UniDir algorithm (cf. [13, 51]). Originally, UniDir was introduced to compute the matrix-vector product in the space $\widehat{V}_J$ for a tensor product operator, i.e. the expression $(S \otimes S)\widehat{x}$, where $\widehat{x} \in \widehat{V}_J$. Algorithms which employ similar techniques were developed in [26, 29, 30]. Assuming that the complexity of the evaluation of S on the spaces Z_k is linear, UniDir performs the matrix-vector multiplication $(S \otimes S)\widehat{x}$ again with linear complexity $\mathcal{O}(\dim \widehat{V}_J) = \mathcal{O}(N_J \log N_J)$.

Various representations of UniDir are already available in the literature. Our main motivation for giving a different derivation is to discuss several intermediate results which are not present in the literature in this form. These results, such as Lemma 4.2 for example, are used in Subsection 4.3 and allow for a shorter and more concise derivation of the algorithm for the quadratic part.

The need to develop specialized algorithms for the matrix–vector multiplications in the space $\widehat{V}_J$ arises from a typical situation in which the spaces Z_j describe a multi-resolution analysis in d dimensions (cf. [13, 29]). In this case, the resulting discretization matrix $S_J \otimes S_J$ is not sparse even for finitely supported ansatz functions and local operators S because of the overlapping of ansatz functions from different levels. However, the matrix $S_J \otimes S_J$ has block tensor product structure that can be utilized for fast matrix–vector multiplication (cf. [29, 51]).

Let us consider a function $x \in V_J$ first. By using the tensor product structure of the operator $S \otimes S$, we get the equivalent representation

$$Y = S_J X S_J \tag{18}$$

for the product $(S \otimes S)x$. Here, $X \in \mathbb{R}^{\mathcal{V}_J}$ is the coefficient matrix associated with x. The formulation (18) provides an important link between the spaces V_J and $\mathcal{V}_J$. It allows the algorithms developed for the matrix-vector multiplication to be adapted to evaluate products of operators.

The situation is different in the case of $\widehat{x} \in \widehat{V}_J$. A representation as in (18) is not possible, as the sparse grid on $\Omega \times \Omega$ is not a tensor product of grids on Ω, but rather a sum of tensor products (cf. [51] for example). However, we can express $(S \otimes S)\widehat{x}$ in accordance with

$$[Y]_{\boldsymbol{j}} = \sum_{\|\boldsymbol{i}\|_1 \leqslant J} S_{(j_1,i_1)} \widehat{X}_{(i_1,j_2)} S_{(j_2,i_2)}, \quad \|\boldsymbol{j}\|_1 \leqslant J, \tag{19}$$

where $\widehat{X}$ is the coefficient matrix of $\widehat{x}$.

Roughly speaking, UniDir is based on two ideas. First, note that each summand on the right–hand side of (19) can be evaluated by computing the left or right product in the first place, resulting in different computation costs. A detailed discussion of associated complexities can be found in [29]. Second, although each summand has unique indices $\boldsymbol{i}$ and $\boldsymbol{j}$, some of the products $S_{(j_1,i_1)}\widehat{X}_{(i_1,j_2)}$ or $\widehat{X}_{(i_1,j_2)}S_{(j_2,i_2)}$ appear multiple times. UniDir avoids repeated evaluation of these products by properly parenthesizing the sum in (19) and factoring out common terms.

Both ideas are reflected in the following splitting of the matrix S_J in the lower (top down) part L_J and the upper (bottom up) part U_J

$$S_J = L_J + U_J, \quad L_J = \big[S_{(j_1,j_2)}\big]_{\substack{j_1,j_2 \leqslant J,\\ j_1 \geqslant j_2}}, \quad U_J = \big[S_{(j_1,j_2)}\big]_{\substack{j_1,j_2 \leqslant J,\\ j_1 < j_2}}. \tag{20}$$

With (20), we compute the product $S_J \widehat{X} S_J$ as

$$S_J \widehat{X} S_J = U_J \widehat{X} S_J + L_J \widehat{X} S_J. \tag{21}$$

This approach leads to linear complexity, provided the following assumption holds.

Assumption 4.1. *The matrix–vector products for the matrices $\widetilde{S}_k$, $\widetilde{L}_k$, $\widetilde{U}_k$ can be evaluated for $k \leqslant J$ with complexity $\mathcal{O}(N_k)$.*

To derive UniDir formally, let us start with the following lemma regarding the interaction between the matrices L_J and U_J and the operator Π_{SG}. We will use this result for the derivation of the UniDir as well as for the quadratic part algorithm.

Lemma 4.2. *Let $\widehat{X} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$ and $X \in \mathbb{R}^{\mathcal{V}_J}$. Define $L_J, U_J \in \mathbb{R}^{N_J \times N_J}$ as in (20) and Π_{SG} as in* (16)*. Then, there holds*

(a) $\Pi_{SG}(U_J\widehat{X}) = U_J\widehat{X}$ and $\Pi_{SG}(\widehat{X}L_J) = \widehat{X}L_J$, i.e. $U_J\widehat{X}$ and $\widehat{X}L_J$ are again elements of $\mathbb{R}^{\widehat{\mathcal{V}}_J}$.

(b) $\Pi_{SG}(L_JX) = \Pi_{SG}\big(L_J\Pi_{SG}(X)\big)$ and $\Pi_{SG}(XU_J) = \Pi_{SG}\big(\Pi_{SG}(X)U_J\big)$.

Proof. To prove the first identity in (a), we compute

$$U_J\widehat{X} = \sum_{k \leqslant J} U_J\widehat{X}\Pi_k \overset{(\star)}{=} \sum_{k \leqslant J} \widetilde{\Pi}_{J-k}U_J\widehat{X}\Pi_k \overset{(16)}{=} \Pi_{SG}(U_J\widehat{X}),$$

where $(\star)$ holds because blocks of coefficients corresponding to the subspaces $W_{(\ell,k)}$, $\ell > k$, are equal zero, that is $\widehat{X}\Pi_k \in \bigoplus_{i \leqslant k} \mathbb{R}^{N_i \times N_k}$. Figure 2 illustrates this statement. The proof of $\Pi_{SG}(\widehat{X}L_J) = \widehat{X}L_J$ is similar.

To prove the first statement in (b), we use the representation (16) of Π_{SG}

$$\Pi_{SG}(L_JX) = \Pi_{SG}\left(\sum_{k \leqslant J} L_JX\Pi_k\right) \overset{(16)}{=} \sum_{k \leqslant J} \widetilde{\Pi}_{J-k}L_JX\Pi_k.$$

Next, note that because of the shape of L_J we have $\widetilde{\Pi}_{J-k}L_J = \widetilde{\Pi}_{J-k}L_J\widetilde{\Pi}_{J-k}$. Therefore,

$$\widetilde{\Pi}_{J-k}L_JX\Pi_k = \widetilde{\Pi}_{J-k}L_J\widetilde{\Pi}_{J-k}X\Pi_k = \widetilde{\Pi}_{J-k}L_J\Pi_{SG}(X)\Pi_k$$

holds for all $k \leqslant J$. By using this result, we arrive at

$$\Pi_{SG}(L_JX) = \sum_{k \leqslant J} \widetilde{\Pi}_{J-k}L_J\Pi_{SG}(X)\Pi_k = \Pi_{SG}\big(L_J\Pi_{SG}(X)\big).$$

The proof of $\Pi_{SG}(XU_J) = \Pi_{SG}\big(\Pi_{SG}(X)U_J\big)$ is similar. □

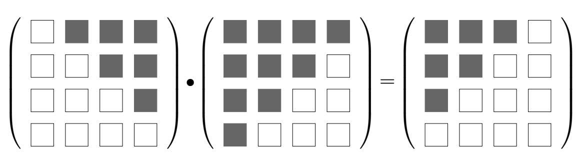

FIGURE 2. Schematic representation of the application of the bottom up part U_J of the operator $\mathcal{S}$ from the left to a sparse grid matrix.

In order to compute the expression (21), we have to consider the complexity of applying the matrices L_J, U_J, and S_J to elements from $\mathbb{R}^{\widehat{\mathcal{V}}_J}$ and $\mathbb{R}^{\mathcal{V}_J}$. For the proofs, we will use repeatedly two facts. First, the blocks of a sparse grid matrix which correspond to the spaces $W_{(k,\ell)}$ with $\ell + k > J$ are zero. Second, we can discard the computation of the blocks which are in the kernel of the operator $\varPi_{SG}$. We start by examining the terms $\varPi_{SG}\big(\widehat{X}S_J\big)$ and $\varPi_{SG}\big(S_J\widehat{X}\big)$ in the next lemma.

Lemma 4.3. *Let the operator S fulfil Assumption 4.1 and let $\widehat{X} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$. Then, the terms*

$$\varPi_{SG}\big(S_J\widehat{X}\big), \quad \varPi_{SG}\big(\widehat{X}S_J\big)$$

can be evaluated with complexity $\mathcal{O}(N_J \log N_J)$.

Proof. By using the representation (16) of $\varPi_{SG}$, we have

$$\varPi_{SG}\big(\widehat{X}S_J\big) = \sum_{k\leqslant J} \varPi_k \widehat{X} S_J \widetilde{\varPi}_{J-k}.$$

Note that $\varPi_k\widehat{X} = \varPi_k\widehat{X}\widetilde{\varPi}_{J-k}$. With this result, we obtain

$$\varPi_{SG}\big(\widehat{X}S_J\big) = \sum_{k\leqslant J} \varPi_k \widehat{X} \widetilde{\varPi}_{J-k} S \widetilde{\varPi}_{J-k} = \sum_{k\leqslant J} \varPi_k \widehat{X} \widetilde{S}_{J-k}.$$

Figure 3 illustrates this representation.

By Assumption 3.1, each term $\varPi_k\widehat{X}$ consists of $\mathcal{O}(2^k)$ vectors with $\mathcal{O}(N_{J-k}) = \mathcal{O}(2^{J-k})$ non-zero entries. Therefore, using Assumption 4.1, we can evaluate $\varPi_k\widehat{X}\widetilde{S}_{J-k}$ with complexity $\mathcal{O}(N_J)$. For the complete sum $\sum_{k\leqslant J} \varPi_k\widehat{X}\widetilde{S}_{J-k}$, we obtain the complexity

$$\sum_{k\leqslant J} \mathcal{O}\left(N_J\right) = \mathcal{O}\left(N_J \log N_J\right).$$

The proof for $\varPi_{SG}\big(S_J\widehat{X}\big)$ is similar. $\square$

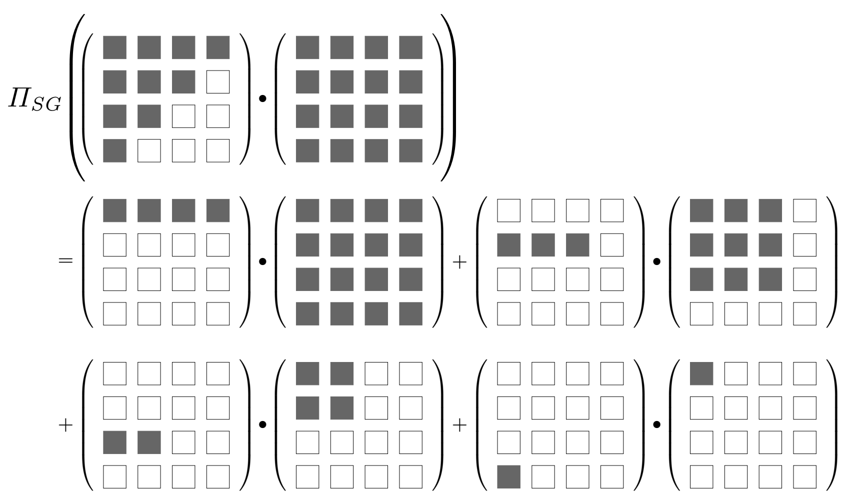

FIGURE 3. Illustration of applying S_J from the right to a sparse grid matrix $\widehat{X}$, followed by the projection onto $\mathbb{R}^{\widehat{\mathcal{V}}_J}$. The result can be computed as a sum of products $\widetilde{S}_{J-k}\widehat{X}\varPi_k$, $k = 0, 1, \ldots, J$.

The next two lemmas state results regarding the computational complexities of the terms on the right-hand side of the splitting (21). We will use these lemmas to derive the UniDir algorithm as well as to compute the quadratic term of the Riccati equation in Subsection 4.3. We start with the complexity of evaluating $\mathit{\Pi}_{SG}\big(U_J\widehat{X}S_J\big)$.

Lemma 4.4. *Let the operator S fulfil Assumption 4.1 and let $\widehat{X} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$. Define $U_J \in \mathbb{R}^{N_J \times N_J}$ as in* (20)*. Then, the expression*

$$\mathit{\Pi}_{SG}\big(U_J\widehat{X}S_J\big)$$

can be evaluated with complexity $\mathcal{O}(N_J \log N_J)$.

Proof. The product $U_J\widehat{X}$ is again an element of $\mathbb{R}^{\widehat{\mathcal{V}}_J}$ by Lemma 4.2 (a). We conclude the proof by applying Lemma 4.3 to $\mathit{\Pi}_{SG}\big(U_J\widehat{X}S_J\big)$. □

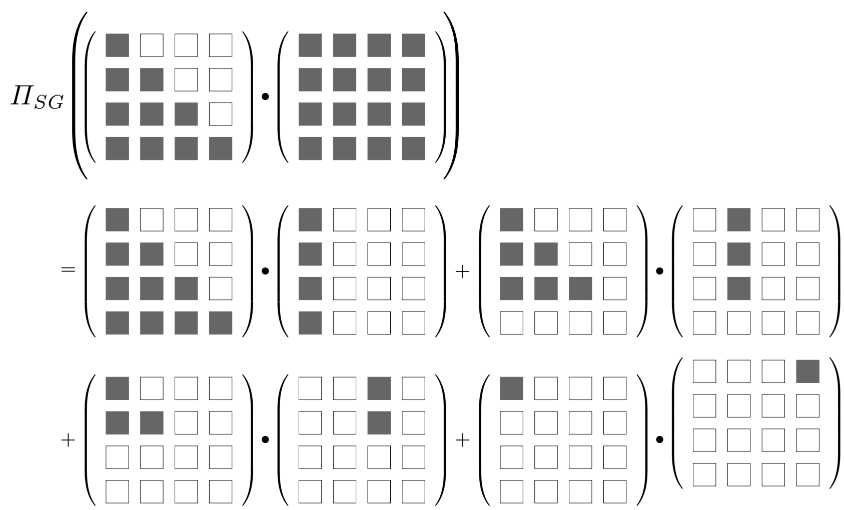

FIGURE 4. Schematic representation of applying the top down part L_J from left to a regular grid matrix X, followed by the projection onto $\mathbb{R}^{\widehat{\mathcal{V}}_J}$. The result can be computed as a sum of products $\widetilde{L}_{J-k}\widehat{X}\mathit{\Pi}_k$, $k = 0, 1, \ldots, J$.

The next lemma proves a result regarding the complexity of the application of the matrix L_J.

Lemma 4.5. *Let the operator S fulfil Assumption 4.1 and $\widehat{X} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$ be a sparse grid matrix. Define $L_J \in \mathbb{R}^{N_J \times N_J}$ as in (20). Then, the expression*

$$\mathit{\Pi}_{SG}\big(L_J\widehat{X}S_J\big) \tag{22}$$

can be evaluated with complexity $\mathcal{O}(N_J \log N_J)$.

Proof. The product $\widehat{X}S_J$ is not an element of $\mathbb{R}^{\widehat{\mathcal{V}}_J}$ in general. However, by Lemma 4.2 (b), there still holds $\mathit{\Pi}_{SG}\big(L_J\widehat{X}S_J\big) = \mathit{\Pi}_{SG}\big(L_J\mathit{\Pi}_{SG}\big(\widehat{X}S_J\big)\big)$.

By Lemma 4.3, the expression $\widehat{X}' = \mathit{\Pi}_{SG}\big(\widehat{X}S_J\big)$ can be evaluated with complexity $\mathcal{O}(N_J \log N_J)$. Furthermore, due to the form of the matrix L_J, we can write

$$\varPi_{SG}\big(L_J\widehat{X}'\big) = \sum_{k\leqslant J}\widetilde{\varPi}_{J-k}L_J\widehat{X}'\varPi_k = \sum_{k\leqslant J}\widetilde{\varPi}_{J-k}L_J\widetilde{\varPi}_{J-k}\widehat{X}'\varPi_k = \sum_{k\leqslant J}\widetilde{L}_{J-k}\widehat{X}'\varPi_k.$$

Figure 4 illustrates this representation, which is also true for elements of $\mathbb{R}^{\mathcal{V}_J}$.

As S_J fulfils Assumption 4.1, the products

$$\widetilde{L}_{J-k}\widehat{X}'\varPi_k, \quad k = 0, 1, \ldots, J,$$

can be computed with complexities $\mathcal{O}(N_{J-k})$. Therefore, similar to the proof of Lemma 4.3, the overall complexity is $\mathcal{O}\left(N_J \log N_J\right)$. □

In the next theorem, we summarize the observations of this section and estimate the complexity of applying linear tensor product operators to a sparse grid matrix. This statement can also be found in [1, 12, 51].

Theorem 4.6 (UniDir)**.** *Let the operator S fulfil Assumption 4.1 and let $\widehat{X} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$. Then, the expression*

$$\varPi_{SG}\big(S_J\widehat{X}S_J\big)$$

can be evaluated with complexity $\mathcal{O}(N_J \log N_J)$.

Proof. We use the decomposition (21)

$$\varPi_{SG}\big(S_J\widehat{X}S_J\big) = \varPi_{SG}\big(L_J\widehat{X}S_J\big) + \varPi_{SG}\big(U_J\widehat{X}S_J\big),$$

and apply Lemma 4.4 to $\varPi_{SG}\big(U_J\widehat{X}S_J\big)$ and Lemma 4.5 to $\varPi_{SG}\big(L_J\widehat{X}S_J\big)$, respectively. □

Alltogether, we arrive at Algorithm 1, which evaluates the expression $\varPi_{SG}\big(S_J\widehat{X}S_J\big)$ with linear complexity $\mathcal{O}(N_J \log N_J)$. This algorithm is used to evaluate the linear part of the Riccati equation.

Data : A sparse grid matrix $\widehat{X} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$, for $k \leqslant J$ the matrices $\widetilde{S}_k$, $\widetilde{U}_k$, $\widetilde{L}_k$ as defined in (15) and (20).

Result: $Y = \varPi_{SG}\big(S_J\widehat{X}S_J\big)$.

1 $\widehat{X}_U \leftarrow U_J\widehat{X} = \sum_{k\leqslant J}\widetilde{U}_{J-k}\widehat{X}\varPi_k$

2 $\widehat{X}_U \leftarrow \varPi_{SG}\big(\widehat{X}_U S_J\big) = \sum_{k\leqslant J}\varPi_k\widehat{X}_U\widetilde{S}_{J-k}$

3 $\widehat{X}_L \leftarrow \varPi_{SG}\big(\widehat{X}S_J\big) = \sum_{k\leqslant J}\varPi_k\widehat{X}\widetilde{S}_{J-k}$

4 $\widehat{X}_L \leftarrow \varPi_{SG}\big(L_J\widehat{X}_L\big) = \sum_{k\leqslant J}\widetilde{L}_{J-k}\,\widehat{X}_L\varPi_k$

5 $Y \leftarrow \widehat{X}_U + \widehat{X}_L$

Algorithm 1: (UniDir) Evaluation of the linear part of the Riccati equation.

4.3. **Quadratic part.** In this subsection, we will consider the evaluation of expressions of the form

$$\varPi_{SG}\big(S_J \widehat{X} R \widehat{Z} S_J\big), \tag{23}$$

where S_J is the discretization matrix of the operator S, $\widehat{X}$ and $\widehat{Z}$ are sparse grid matrices, and $R \in \mathbb{R}^{N_J \times N_J}$. Our goal is to derive an algorithm for the computation of the quadratic part of the Riccati equation. In this case, the matrix R is the discretization matrix of the operator $BB^\star$ with respect to the space Z_J and $\widehat{Z} = \widehat{X}$.

The main ideas of the algorithm are, similar to UniDir, to utilize the sparsity of elements from $\mathbb{R}^{\widehat{\mathcal{V}}_J}$ and the properties of the projector $\varPi_{SG}$. In the first step, we interpret the expression (23) as the application of the tensor product operator $S_J \otimes S_J$ to $\widehat{X} R \widehat{Z}$. Next, we split the matrix R into its lower R^L and upper R^U parts similar to (20), i.e. $R = R^L + R^U$. In view of Lemma 4.2 (a), we see that the products $\widehat{X} R^L$ and $R^U \widehat{Z}$ are again sparse grid matrices. Therefore, we can split (23) as

$$S_J \widehat{X} R \widehat{Z} S_J = S_J \widehat{X}\big(R^L + R^U\big)\widehat{Z} S_J = S_J\big(\widehat{X} R^L\big)\widehat{Z} S_J + S_J \widehat{X}\big(R^U \widehat{Z}\big) S_J. \tag{24}$$

By this means, we have to apply the operator $S \otimes S$ to products of sparse grid matrices in order to to evaluate (23).

Let us first state the following lemma regarding the projection onto $\mathbb{R}^{\widehat{\mathcal{V}}_J}$ of a product of two sparse grid matrices.

Lemma 4.7. *Let $\widehat{X}, \widehat{Z} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$. Then, the expression*

$$\varPi_{SG}\big(\widehat{X}\widehat{Z}\big) \tag{25}$$

can be evaluated with complexity $\mathcal{O}(N_J^{3/2})$.

Proof. Let us write $Y = \widehat{X}\widehat{Z}$. The blocks of the matrix Y are scalar products of the blocks of the rows of $\widehat{X}$ and the columns of $\widehat{Z}$:

$$Y_{(j_1,j_2)} = \sum_{k \leqslant J} \widehat{X}_{(j_1,k)} \widehat{Z}_{(k,j_2)} = \sum_{k \leqslant \min\{J-j_1, J-j_2\}} \widehat{X}_{(j_1,k)} \widehat{Z}_{(k,j_2)}.$$

Recall that we assume $d = 1$ in this section. Therefore, by Assumption 3.1, the complexity for the computation of a matrix product $\widehat{X}_{(j_1,k)}\widehat{Z}_{(k,j_2)}$ is $\mathcal{O}(2^{k+j_1+j_2})$. Consequently, the computation of a block $Y_{(j_1,j_2)}$ requires

$$\sum_{0 \leqslant k \leqslant \min\{J-j_1,J-j_2\}} \mathcal{O}\left(2^{k+j_1+j_2}\right) = \mathcal{O}\left(2^{\min\{J-j_1,J-j_2\}+j_1+j_2}\right)$$

operations.

We have the following cases:

$$\min\{J-j_1, J-j_2\} = \begin{cases} J - j_2, & j_1 < j_2, \\ J - j_1, & j_1 \geqslant j_2. \end{cases} \tag{26}$$

To use the expression (26), consider the partition of the set $\big\{\|\boldsymbol{j}\|_1 \leqslant J\big\}$ along the lines $\|\boldsymbol{j}\|_1 = const.$, that is

$$
\begin{aligned}
\big\{\|\boldsymbol{j}\|_1 \leqslant J\big\} &= \bigcup_{\ell=1}^{J} \big\{\|\boldsymbol{j}\|_1 = \ell\big\} \\
&= \bigcup_{\ell=1}^{J} \Big[\big\{\|\boldsymbol{j}\|_1 = \ell,\, j_1 < j_2\big\} \cup \big\{\|\boldsymbol{j}\|_1 = \ell,\, j_1 \geqslant j_2\big\}\Big].
\end{aligned}
\tag{27}
$$

Note that we have the equivalent representations

$$
\begin{aligned}
\big\{\|\boldsymbol{j}\|_1 = \ell,\, j_1 < j_2\big\} &= \big\{\|\boldsymbol{j}\|_1 = \ell,\, j_1 \leqslant \lfloor \ell/2 \rfloor\big\}, \\
\big\{\|\boldsymbol{j}\|_1 = \ell,\, j_1 \geqslant j_2\big\} &= \big\{\|\boldsymbol{j}\|_1 = \ell,\, j_2 \leqslant \lceil \ell/2 \rceil\big\}.
\end{aligned}
\tag{28}
$$

This results in the following estimate for the complexity of computation of $\Pi_{SG}(Y)$:

$$
\begin{aligned}
&\sum_{\|\boldsymbol{j}\|_1 \leqslant J} \mathcal{O}\left(2^{\min\{J-j_1, J-j_2\}+j_1+j_2}\right) \\
&\overset{(27)}{=} \sum_{\ell \leqslant J} \Bigg[\sum_{\substack{j_1+j_2=\ell,\\ j_1<j_2}} \mathcal{O}\left(2^{J-j_2+j_1+j_2}\right) + \sum_{\substack{j_1+j_2=\ell,\\ j_1 \geqslant j_2}} \mathcal{O}\left(2^{J-j_1+j_1+j_2}\right) \Bigg] \\
&= \sum_{\ell \leqslant J} \Bigg[\sum_{\substack{j_1+j_2=\ell,\\ j_1<j_2}} \mathcal{O}\left(2^{J+j_1}\right) + \sum_{\substack{j_1+j_2=\ell,\\ j_1 \geqslant j_2}} \mathcal{O}\left(2^{J+j_2}\right) \Bigg] \\
&\overset{(28)}{=} \mathcal{O}\left(2^J\right) \cdot \sum_{\ell \leqslant J} \Bigg[\sum_{\substack{j_1+j_2=\ell,\\ j_1 \leqslant \lfloor \ell/2 \rfloor}} \mathcal{O}\left(2^{j_1}\right) + \sum_{\substack{j_1+j_2=\ell,\\ j_2 \leqslant \lceil \ell/2 \rceil}} \mathcal{O}\left(2^{j_2}\right) \Bigg] \\
&= \mathcal{O}\left(2^J\right) \cdot \sum_{\ell \leqslant J} \mathcal{O}\left(2^{\ell/2}\right) = \mathcal{O}\left(2^{3J/2}\right) = \mathcal{O}\left(N_J^{3/2}\right).
\end{aligned}
$$

□

The next theorem estimates the computation costs for the expression $S_J \widehat{X} \widehat{Z} S_J$, where $\widehat{X}$ and $\widehat{Z}$ are sparse grid matrices.

Theorem 4.8. *Let the operator S fulfil Assumption 4.1. Let $\widehat{X}, \widehat{Z} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$. Then, the expression*

$$
\Pi_{SG}\big(S_J \widehat{X} \widehat{Z} S_J\big) \tag{29}
$$

can be evaluated with complexity $\mathcal{O}(N_J^{3/2})$.

Proof. We define $L_J, U_J \in \mathbb{R}^{N_J \times N_J}$ as in (20) and consider the splitting

$$
\begin{aligned}
\Pi_{SG}\big(S_J \widehat{X} \widehat{Z} S_J\big) &= \Pi_{SG}\big(L_J \widehat{X} \widehat{Z} L_J\big) + \Pi_{SG}\big(L_J \widehat{X} \widehat{Z} U_J\big) \\
&\quad + \Pi_{SG}\big(U_J \widehat{X} \widehat{Z} L_J\big) + \Pi_{SG}\big(U_J \widehat{X} \widehat{Z} U_J\big).
\end{aligned}
$$

We prove that each summand on the right hand side can be evaluated with complexity $\mathcal{O}(N_J^{3/2})$.

$\underline{\Pi_{SG}\big(L_J \widehat{X} \widehat{Z} L_J\big)}$: By Lemma 4.2 (b), there holds

$$
\Pi_{SG}\big(L_J \widehat{X} \widehat{Z} L_J\big) = \Pi_{SG}\big(L_J \Pi_{SG}\big(\widehat{X} \widehat{Z} L_J\big)\big).
$$

By Lemma 4.2 (a), the product $\widehat{Z}L_J$ is a sparse grid matrix. Therefore, by Lemma 4.7, the computation of $\widehat{X}' = \Pi_{SG}(\widehat{X}\widehat{Z}L_J)$ requires $\mathcal{O}(N_J^{3/2})$ operations. Note that $\widehat{X}' \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$, i.e. $\widehat{X}'$ is a sparse grid function. Consequently, the evaluation of the expression $\Pi_{SG}(L_J\widehat{X}')$ is of complexity $\mathcal{O}(N_J \log N_J)$ by Lemma 4.5. We conclude that the evaluation of $\Pi_{SG}(L_J\widehat{X}\widehat{Z}L_J)$ is $\mathcal{O}(N_J^{3/2})$.

$\underline{\Pi_{SG}(L_J\widehat{X}\widehat{Z}U_J)}$: We use Lemma 4.2 (b) twice and get

$$\begin{aligned}\Pi_{SG}(L_J\widehat{X}\widehat{Z}U_J) &= \Pi_{SG}(L_J\Pi_{SG}(\widehat{X}\widehat{Z}U_J))\\ &= \Pi_{SG}(L_J\Pi_{SG}(\Pi_{SG}(\widehat{X}\widehat{Z})U_J)).\end{aligned}$$

The complexity for the evaluation of $\widehat{X}' = \Pi_{SG}(\widehat{X}\widehat{Z})$ is of order $\mathcal{O}(N_J^{3/2})$. The terms $\widehat{X}'' := \Pi_{SG}(\widehat{X}'U_J)$ and $\Pi_{SG}(L_J\widehat{X}'')$ can be computed with $\mathcal{O}(N_J \log N_J)$ by Lemma 4.4 and Lemma 4.5, respectively. Therefore, the total complexity for the evaluation of $\Pi_{SG}(L_J\widehat{X}\widehat{Z}U_J)$ is $\mathcal{O}(N_J^{3/2})$.

$\underline{\Pi_{SG}(U_J\widehat{X}\widehat{Z}L_J)}$: We note that $U_J\widehat{X}$ and $\widehat{Z}L_J$ are again sparse grid matrices by Lemma 4.2 (a). Therefore, the complete expression can be evaluated with complexity $\mathcal{O}(N_J^{3/2})$ by Lemma 4.7.

$\underline{\Pi_{SG}(U_J\widehat{X}\widehat{Z}U_J)}$: The argumentation is similar to the case $\Pi_{SG}(L_J\widehat{X}\widehat{Z}L_J)$. There holds

$$\Pi_{SG}(U_J\widehat{X}\widehat{Z}U_J) = \Pi_{SG}(\Pi_{SG}(U_J\widehat{X}\widehat{Z})U_J).$$

The result of the product $U_J\widehat{X}$ is a sparse grid matrix. Therefore, $\Pi_{SG}(U_J\widehat{X}\widehat{Z})$, and consequently also the complete expression, can be computed with complexity $\mathcal{O}(N_J^{3/2})$.

By combining the single cases, we conclude that the overall complexity for the evaluation of (29) is $\mathcal{O}(N_J^{3/2})$. □

Finally, we apply Theorem 4.8 to the expression $\Pi_{SG}(S_J\widehat{X}R\widehat{Z}S_J)$.

Corollary 4.9. *Let the assumption of Theorem 4.8 be fulfilled. and let $R \in \mathbb{R}^{\mathcal{V}_J}$. Denote by R^L and R^U the lower and upper part of R, i.e.*

$$R^L = \left[R_{(j_1,j_2)}\right]_{\substack{j_1,j_2 \leqslant J,\\ j_1 \geqslant j_2}}, \quad R^U = \left[R_{(j_1,j_2)}\right]_{\substack{j_1,j_2 \leqslant J,\\ j_1 < j_2}}, \quad R = R^L + R^U.$$

Assume that the products $\widehat{X}R^L$ and $R^U\widehat{Z}$ can be evaluated with complexity $\mathcal{O}(N_J^{3/2})$. Then, the expression

$$\Pi_{SG}(S_J\widehat{X}R\widehat{Z}S_J)$$

can be evaluated with complexity $\mathcal{O}(N_J^{3/2})$.

Proof. We use the splitting (24)

$$S_J\widehat{X}R\widehat{Z}S_J = S_J(\widehat{X}R^L)\widehat{Z}S_J + S_J\widehat{X}(R^U\widehat{Z})S_J.$$

By our assumptions, the expressions $\widehat{X}' = \widehat{X}R^L$ and $\widehat{Z}' = R^U\widehat{Z}$ can be computed with complexity $\mathcal{O}(N_J^{3/2})$, and are sparse grid matrices according to Lemma 4.2 (a). Therefore, we can apply Theorem 4.8 to the terms $\Pi_{SG}(S_J\widehat{X}'\widehat{Z}S_J)$ and $\Pi_{SG}(S_J\widehat{X}\widehat{Z}'S_J)$. □

Theorem 4.8 and Corollary 4.9 give rise to the following algorithm for the computation of the quadratic part of the Riccati equation.

Data : A sparse grid matrix $\widehat{X} \in \mathbb{R}^{\widehat{\mathcal{V}}_J}$, for $k \leqslant J$ the matrices $\widetilde{S}_k$, $\widetilde{U}_k$, $\widetilde{L}_k$, matrices R^L, R^U.

Result: $Y = \Pi_{SG}\big(S_J \widehat{X} R \widehat{X} S_J\big)$.

$\widehat{X}_L \leftarrow \widehat{X} R^L$

$\widehat{X}_U \leftarrow R^U \widehat{X}$

for $(\widehat{Z}, \widehat{Z}') \in \{(\widehat{X}_L, \widehat{X}), (\widehat{X}, \widehat{X}_U)\}$ **do**

  $Y_1 \leftarrow \Pi_{SG}\big(L_J \Pi_{SG}\big(\widehat{Z}\big(\widehat{Z}' L_J\big)\big)\big)$

  $Y_2 \leftarrow \Pi_{SG}\big(L_J \Pi_{SG}\big(\Pi_{SG}\big(\widehat{Z}\widehat{Z}'\big) U_J\big)\big)\big)$

  $Y_3 \leftarrow \Pi_{SG}\big(\big(U_J \widehat{Z}\big)\big(\widehat{Z}' L_J\big)\big)$

  $Y_4 \leftarrow \Pi_{SG}\big(\Pi_{SG}\big(\big(U_J \widehat{Z}\big)\widehat{Z}'\big) U_J\big)$

  $Y \leftarrow Y + \sum_{i=1}^{4} Y_i$

Algorithm 2: Evaluation of the quadratic part of the Riccati equation.

4.4. **Application of the algorithms and the Netwon's method.** The complexity of the presented algorithms is estimated based on Assumption 3.1 and Assumption 4.1. A finitely supported multi-resolution analysis with all involved operators being local is a typical situation where these assumptions are fulfilled. Important examples of suitable sequences $\{Z_k\}$ include hierachical bases ([13]), wavelets ([15]), multilevel frames ([29, 51]), or polynomials of different degrees ([1, 13]). In this article, we consider Z_k to be spanned by the hierarchical basis of standard hat functions and take A to be a second order differential operator. A precise definition is provided in Section 5.

The algebraic Riccati equation is a nonlinear equation that depends on the unknown operator P in a quadratic manner. To find a solution, various methods for nonlinear equations can be considered (see e.g. [6, 7, 8, 35]). We have implemented the Newton's method as proposed by [34]. In each iteration $n \in \mathbb{N}$, we solve the Sylvester type equation of the form

$$\begin{aligned}\Pi_{SG}\Big(\big(E_J \widehat{X}^{(n)} E_J - A_J\big)\widehat{X}^{(n+1)} E_J + E_J \widehat{X}^{(n+1)}\big(E_J \widehat{X}^{(n)} E_J - A_J\big)\Big) \\ = \Pi_{SG}\Big(E_J \widehat{X}^{(n)} E_J \widehat{X}^{(n)} E_J + Q_J\Big)\end{aligned} \tag{30}$$

for the unknown sparse grid matrix $\widehat{X}^{(n+1)}$. Algorithm 1 and Algorithm 2 are respectively applied to evaluate the linear and non-linear parts of the equation (30). Note that, using an optimal preconditioner such as the multigrid method, enables the equation (30) to be solved with over-all complexity of $\mathcal{O}(N_J^{3/2})$. Therefore, the total cost of solving the Riccati equation results in $\mathcal{O}(N_{\text{iter}} N_J^{3/2})$ operations, where N_{iter} represents the number of iterations of the Newton's method.

5. Numerical results

For the demonstration of our algorithm, let us consider the domain $\Omega = [0,1]^d$, whereby $d = 1,2,3$, and take $Z = H_0^1(\Omega)$ and $H = L^2(\Omega)$. In order to construct the sparse grid ansatz space $\widehat{V}_J$ on $\Omega \times \Omega$, we use piecewise linear hat functions (see e.g. [13]). Our starting point is the standard linear hat function on $\mathbb{R}$:

$$\phi(x) := \max\{1 - |x|, 0\}.$$

By translation and dilatation of $\phi(x)$, we define the functions

$$\phi_{(\ell,k)}(x) := \phi\left(\frac{x - k \cdot 2^\ell}{2^\ell}\right) = \phi(2^{-\ell}x - k), \quad \ell \in \mathbb{N}_0,\ k \leqslant 2^\ell,$$

whereby $\phi_{(0,0)}$ and $\phi_{(0,1)}$ are restricted to Ω. The integers ℓ and k are usually termed level and index of the function $\phi_{(\ell,k)}(x)$.

Next, let $\boldsymbol{\ell}, \boldsymbol{k} \in \mathbb{N}^d$ be multi-indices and $x \in \mathbb{R}^d$. We define a piecewise d-linear function on Ω by the tensor product

$$\phi_{(\boldsymbol{\ell},\boldsymbol{k})}(x) := \prod_{i=1}^d \phi_{(\ell_i,k_i)}(x_i),$$

and introduce the spaces Z_j as

$$Z_j := \operatorname{span}\left\{\phi_{(\boldsymbol{\ell},\boldsymbol{k})} : \|\boldsymbol{\ell}\|_\infty \leqslant j \text{ and } k_i \leqslant 2^{\ell_i} \text{ for } i = 1, \ldots, d\right\}.$$

Given the spaces Z_j, we can construct W_j, $W_{\boldsymbol{j}}$, and the sparse grid space $\widehat{V}_J$ as described in Section 3. Note that the spaces $\widehat{V}_J$ are defined on 2-, 4- and 6-dimensional domains. The algorithms for the solution of the Riccati equation are implemented based on the sparse grid library SG++, see [43, 48] for the details.

The operator A under consideration will be

$$A : H_0^1(\Omega) \cap H^2(\Omega) \to L^2(\Omega), \quad Az = \sum_{i=1}^d \partial_{x_i x_i} z - \sum_{i=1}^d \partial_{x_i} z + 2 \cdot z.$$

For the right-hand side, we take the operator

$$Q : L^2(\Omega) \to L^2(\Omega), \quad Qz = \int_\Omega q(\cdot,\xi) z(\xi) \,\mathrm{d}\xi,$$

with

$$q(x,\xi) = \prod_{i=1}^d (1 - |2x_i - 1|)(1 - |2\xi_i - 1|)$$

for $x, \xi \in \mathbb{R}^d$.

As the reference solution we take the operator

$$P_{\text{ref}} z = \int_\Omega p_{\text{ref}}(\cdot,\xi) z(\xi) \,\mathrm{d}\xi,$$

whereby we use the ansatz

$$p_{\text{ref}}(x,\xi) = \sum_{\|\boldsymbol{v}\|_\infty, \|\boldsymbol{w}\|_\infty = 1}^{N_{\text{ref}}} p_{\boldsymbol{v},\boldsymbol{w}} \prod_{i=1}^d \sqrt{2} \sin(v_i \pi x_i)\, \mathrm{e}^{\frac{1}{2} x_i} \sqrt{2} \sin(w_i \pi \xi_i)\, \mathrm{e}^{\frac{1}{2} \xi_i} \tag{31}$$

for the kernel p_{ref}. The parameter N_{ref} is $N_{\text{ref}} = 4000$ for $d = 1$, $N_{\text{ref}} = 150$ for $d = 2$, and $N_{\text{ref}} = 30$ for $d = 3$.

Let p_{approx} denote the kernel of the Riccati operator computed by using the sparse grid discretization. Recall that $\mathcal{H}$ is isometric to $L^2(\Omega \times \Omega)$. Using this, we estimate the $\mathcal{H}$-error by considering the pointwise differences of p_{approx} and p_{ref} on the mesh

$$X_{\text{eval}} := \left\{(x,\xi) \in [0,1]^2 \,:\, (x,\xi) = (i,j)\cdot 1/5000,\ i,j = 0,\ldots,5000\right\},$$

i.e. we compute

$$e^2 = \sqrt{\frac{\sum_{(x,\xi)\in X_{\text{eval}}} (p_{\text{approx}}(x,\xi) - p_{\text{ref}}(x,\xi))^2}{|X_{\text{eval}}|}}. \tag{32}$$

A quadrature on a full grid is too expensive to compute the error estimate for four or six dimensional Riccati kernels. Therefore, in these cases, we estimate the L^2-error as

$$\begin{aligned} e^2 &= \|p_{\text{ref}} - p_{\text{approx}}\|_{L^2(\Omega\times\Omega)} \\ &= \left[\|p_{\text{ref}}\|^2_{L^2(\Omega\times\Omega)} - 2\left\langle p_{\text{ref}}, p_{\text{approx}}\right\rangle_{L^2(\Omega\times\Omega)} + \|p_{\text{approx}}\|^2_{L^2(\Omega\times\Omega)}\right]^{1/2}, \end{aligned} \tag{33}$$

whereby we use the ansatz (31) to evaluate the scalar products directly. Note that the error estimate (32) we use for the one-dimensional case can be obtained from (33) by virtue of a numerical quadrature. Thus, the difference between the formulas (32) and (33) is neglectable for sufficiently large number of evaluation points.

The results are presented in Figure 5 and Tables 1, 2, and 3. Herein, 'DoF' is the number of degrees of freedom for the approximation, i.e., the dimension of the ansatz spaces $\widehat{\mathcal{V}}_J$. Especially, we tabulated the convergence rates $\rho_i = \operatorname{ld}\left(e_i^2/e_{i-1}^2\right)$, where e_i^2 is the value of the error estimator e^2 on the level i.

For the present example, we observe nearly the optimal rate of convergence for sparse grids, i.e., the convergence rate is the same as for the discretization on a full tensor grid. The cost per degrees of freedom is however significantly smaller in the case of the sparse grid approach. With the suggested algorithm, the computational cost to find a solution is $\mathcal{O}(N_J^{3/2})$, compared to $\mathcal{O}(N_J^3)$ for the full grid discretization.

level	e^2	$\rho_i(e^2)$	DoF	level	e^2	$\rho_i(e^2)$	DoF
2	1.47_{-3}	$\star$	5	10	3.27_{-8}	1.65	9,217
3	3.88_{-4}	1.93	17	11	6.11_{-9}	2.42	20,481
4	9.87_{-5}	1.97	49	12	1.63_{-9}	1.91	45,057
5	2.49_{-5}	1.99	129	13	4.23_{-10}	1.94	98,305
6	6.28_{-6}	1.99	321	14	1.08_{-10}	1.97	212,993
7	1.58_{-6}	1.99	769	15	2.81_{-11}	1.95	458,753
8	3.99_{-7}	1.99	1,793	16	7.39_{-12}	1.93	983,041
9	1.03_{-7}	1.96	4,097	$\star$	$\star$	$\star$	$\star$

TABLE 1. Estimations e^2 of the $\mathcal{H}$-error and the convergence rates $\rho_i(e^2) = \operatorname{ld}(e_{i-1}^2/e_i^2)$ for $d = 1$.

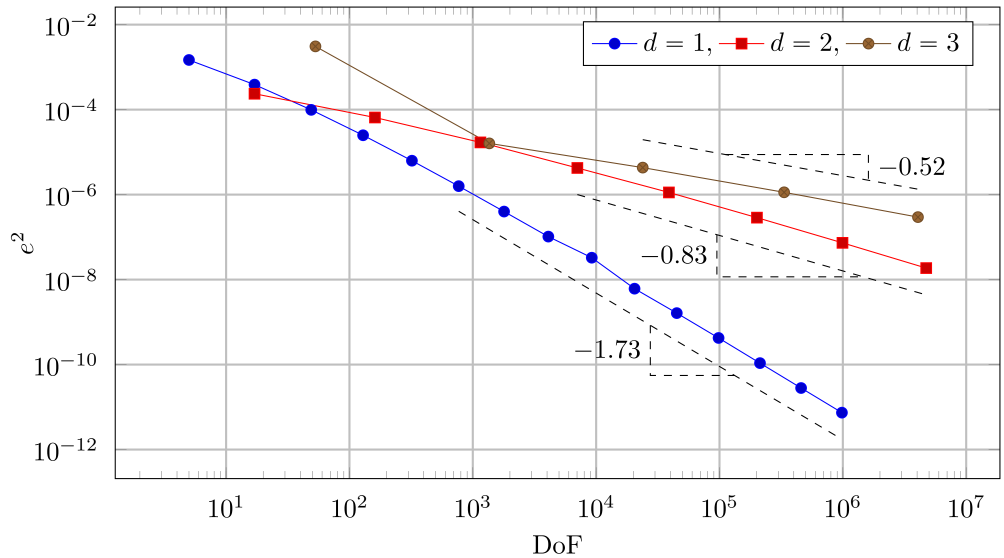

Figure 5. Estimation e^2 of the $\mathcal{H}$-error versus the number of degrees of freedom (DoF) of the ansatz space.

level	e^2	$\rho_i(e^2)$	DoF	level	e^2	$\rho_i(e^2)$	DoF
2	2.36_{-4}	⋆	17	6	1.12_{-6}	1.91	38,913
3	6.53_{-5}	1.86	161	7	2.87_{-7}	1.97	201,217
4	1.69_{-5}	1.95	1,153	8	7.32_{-8}	1.97	993,281
5	4.24_{-6}	2	7,041	9	1.87_{-8}	1.97	4,741,121

Table 2. Estimations e^2 of the $\mathcal{H}$-error and the convergence rates $\rho_i(e^2) = \mathrm{ld}(e^2_{i-1}/e^2_i)$ for $d = 2$.

level	e^2	$\rho_i(e^2)$	DoF	level	e^2	$\rho_i(e^2)$	DoF
2	3.08_{-3}	⋆	53	5	1.13_{-6}	1.93	334,209
3	1.61_{-5}	7.58	1,361	6	2.95_{-7}	1.94	4,064,577
4	4.32_{-6}	1.89	23,857	⋆	⋆	⋆	⋆

Table 3. Estimations e^2 of the $\mathcal{H}$-error and the convergence rates $\rho_i(e^2) = \mathrm{ld}(e^2_{i-1}/e^2_i)$ for $d = 3$.

6. Conclusions

In this article, we developed an efficient solver for large-scale Riccati equations based on a sparse grid discretization. Both, the overall complexity in computation time and the memory requirement, are basically only the square root of those required by a regular tensor product approach. We demonstrated the feasibility of the present approach by means of numerical example for a parabolic control problem with distributed control in one, two, and three spatial dimensions.

References

[1] Stefan Achatz. "Adaptive finite Dünngitter-Elemente höherer Ordnung für elliptische partielle Differentialgleichungen mit variablen Koeffzienten". Dissertation. München, Germany: Technische Universität München, 2003.

[2] Hans Wilhelm Alt. *Lineare Funktionalanalysis*. Berlin-Heidelberg: Springer, 2012.

[3] Jean-Pierre Aubin. *Applied Functional Analysis*. John Wiley & Sons, 2000.

[4] Viorel Barbu. *Nonlinear semigroups and differential equations in Banach spaces*. Leyden: Noordhoff International Publishing, 1976.

[5] Richard Bellman. *Introduction to Matrix Analysis*. McGraw-Hill, 1970.

[6] Peter Benner. *Computational Methods for Linear-Quadratic Optimization*. Research report 98-04. Zentrum für Technomathematik, Universität Bremen, 1998.

[7] Peter Benner and Jens Saak. "Numerical Solution of Large and Sparse Continuous Time Algebraic Matrix Riccati and Lyapunov Equations: A State of the Art Survey". In: *GAMM-Mitt.* 36.1 (2013), pp. 32–52.

[8] Peter Benner et al. "A Numerical Comparison of Different Solvers for Large-Scale, Continuous-Time Algebraic Riccati Equations". In: *SIAM Journal on Scientific Computing* 42.2 (2020), A957–A996.

[9] Alain Bensoussan et al. *Representation and Control of Infinite Dimensional Systems*. Boston: Birkhäuser, 2007.

[10] Dario A. Bini, Bruno Iannazzo, and Beatrice Meini. *Numerical Solution of Algebraic Riccati Equations*. Vol. 9. SIAM, 2012.

[11] Haim Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Berlin-Heidelberg: Springer, 2011.

[12] Hans-Joachim Bungartz. "A Multigrid Algorithm for Higher Order Finite Elements on Sparse Grids". In: *Electronic Transactions on Numerical Analysis* 6 (1997), pp. 63–77.

[13] Hans-Joachim Bungartz and Michael Griebel. "Sparse Grids". In: *Acta Numerica* 13 (2004), pp. 1–123.

[14] J. A. Burns and B. B. King. "A Note on the Regularity of Solutions of Infinite Dimensional Riccati Equations". In: *Applied Mathematics Letters* 7.6 (1994), pp. 13–17.

[15] Wolfgang Dahmen. "Wavelet and Multiscale Methods for Operator Equations". In: *Acta Numerica* 6 (1997), pp. 55–228.

[16] Nelson Dunford and Jacob T. Schwartz. *Linear Operators*. Vol. I: General Theory. John Wiley & Sons, 1988.

[17] Christian Feuersänger. "Sparse Grids Methods for Higher Dimensional Approximation". Dissertation. Bonn, Germany: Rheinische Friedrich–Wilhelms–Universität Bonn, 2010.

[18] Franco Flandoli. "Algebraic Riccati equation arising in boundary control problems". In: *SIAM Journal on Control and Optimization* 25.3 (1987), pp. 612–636.

[19] Jochen Garcke. "Sparse Grids in a Nutshell". In: *Sparse Grids and Applications*. Ed. by Jochen Garcke and Michael Griebel. Vol. 88. Lecture Notes in Computational Science and Engineering. Berlin-Heidelberg: Springer, 2013, pp. 57–80.

[20] Lars Grasedyck and Wolfgang Hackbusch. “A Multigrid Method to Solve Large Scale Sylvester Equations”. In: *SIAM Journal on Matrix Analysis and Applications* 29.3 (2007), pp. 870–894. ISSN: 1095-7162.

[21] Lars Grasedyck, Wolfgang Hackbusch, and Boris N. Khoromskij. “Solution of Large Scale Algebraic Matrix Riccati Equations by Use of Hierarchical Matrices”. In: *Computing* 70.2 (2003), pp. 121–165.

[22] Serkan Gugercin and Athanasios C. Antoulas. “A Survey of Model Reduction by Balanced Truncation and Some New Results”. In: *International Journal of Control* 77.8 (2004), pp. 748–766.

[23] Bernard Haasdonk and Andreas Schmidt. “Reduced Basis Approximation of Large Scale Parametric Algebraic Riccati Equations”. In: *ESAIM: COCV* 24.1 (2018), pp. 129–151.

[24] Wolfgang Hackbusch. *Elliptic Differential Equations. Theory and Numerical Treatment*. Germany: Springer, 2017.

[25] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Berlin-Heidelberg: Springer, 2012.

[26] Helmut Harbrecht. “A Finite Element Method for Elliptic Problems with Stochastic Input Data”. In: *Applied Numerical Mathematics* 60.227–244 (2010).

[27] Helmut Harbrecht and Ilja Kalmykov. “Sparse Grid Approximation of the Riccati Operator for Closed Loop Parabolic Control Problems with Dirichlet Boundary Control”. In: *SIAM Journal on Control and Optimization* 56.6 (2021).

[28] Helmut Harbrecht and Michael Peters. “Combinations Technique Based Second Moment Analysis for Elliptic PDEs on Random Domains”. In: *Sparse Grids and Applications – Stuttgart 2014*. Ed. by Jochen Garcke and Dirk Pflüger. Vol. 109. Lecture Notes in Computational Science and Engineering. Switzerland: Springer International Publishing, 2016, pp. 51–77.

[29] Helmut Harbrecht, Reinhold Schneider, and Christoph Schwab. “Multilevel Frames for Sparse Tensor Product Spaces”. In: *Numerische Mathematik* 110.199–220 (2008).

[30] Helmut Harbrecht and Christoph Schwab. “Sparse Tensor Finite Elements for Elliptic Multiple Scale Problems”. In: *Computer Methods in Applied Mechanics and Engineering* 200.45–46 (2011), pp. 3100–3110.

[31] Edmond A. Jonckheere and Leonard M. Silverman. “A New Set of Invariants for Linear Systems – Application to Reduced Order Compensator Design”. In: *IEEE Transactions on Automatic Control* 28.10 (1983), pp. 953–964.

[32] Rudolf E. Kálmán and Richard S. Bucy. “New results in linear filtering and prediction theory”. In: *Journal of Basic Engineering* 83.1 (1961), pp. 95–108.

[33] Belinda B. King. “Representation of Feedback Operators for Parabolic Control Problems”. In: *Proceedings of the American Mathematical Society* 128.5 (2000), pp. 89–100.

[34] David L. Kleinman. “On an Iterative Technique for Riccati Equation Computations”. In: *IEEE Transactions on Automatic Control* 13.1 (1968), pp. 114–115.

[35] Vladimiír Kučera. “A review of the matrix Riccati equation”. In: *Kybernetika* 9.2 (1973), pp. 42–61.

[36] Irena Lasiecka and Roberto Triggiani. *Control Theory for Partial Differential Equations: Continuous and Approximation Theories*. Vol. I: Abstract Parabolic Systems. Encyclopedia of Mathematics and Its Applications. Cambridge University Press, 1999.

[37] Irena Lasiecka and Roberto Trigging. "The Regulator Problem for Parabolic Equations with Dirichlet Boundary Control; Part I: Riccati's Feedback Synthesis and Regularity of Optimal Solutions". In: *Applied Mathematics and Optimization* 16 (1987), pp. 147–168.

[38] William Allan Light and Elliot Ward Cheney. *Approximation Theory in Tensor Produc Spaces*. Berlin-Heidelberg: Springer, 1985.

[39] Jacques-Louis Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Berlin-Heidelberg: Springer, 1971.

[40] Lennart Ljung, Thomas Kailath, and Benjamin Friedlander. "Scattering Theory and Linear Least Squares Estimation – Part I: Continuous-Time Problems". In: *Proceedings of the IEEE* 64.1 (1976), pp. 131–139.

[41] Mark R. Opmeer. "Decay of Singular Values of the Gramians of Infinite-Dimensional Systems". In: *ECC15 – European Control Conference*. Institute of Electrical and Electronics Engineers Inc., 2015, pp. 1183–1188.

[42] Mark R. Opmeer, Timo Reis, and Winnifried Wollner. "Finite-Rank ADI Iteration for Operator Lyapunov Equations". In: *SIAM Journal on Control and Optimization* 51.5 (2013), pp. 4084–4117.

[43] Dirk Pflüger. "Spatially Adaptive Sparse Grids for High-Dimensional Problems". Dissertation. München, Germany: Institut für Informatik, Technische Universität München, 2010.

[44] I. Gary Rosen. "Convergence of Galerkin Approximations for Operator Riccati Equations – A Nonlinear Evolution Equations Approach". In: *Journal of Mathematical Analysis and Applications* 155 (1991), pp. 226–248.

[45] I. Gary Rosen. "On Hilbert–Schmidt Norm Convergence of Galerkin Approximation for Operator Riccati Equations". In: *International Series of Numerical Mathematics* 91 (1989), pp. 335–349.

[46] I. Gary Rosen and Chunming Wang. "A multilevel technique for the approximate solution of operator Lyapunov and algebraic Riccati equations". In: *SIAM Journal of Numerical Analysis* 32.2 (1995), pp. 514–541.

[47] A. De Santis, A. Germani, and L. Jetto. "Approximation of the Algebraic Riccati Equation in the Hilbert Space of Hilbert-Schmidt Operators". In: *SIAM Journal on Control and Optimization* 31.4 (1993), pp. 847–874.

[48] Julian Valentin and Dirk Pflüger. "Hierarchical Gradient-Based Optimization with B-Splines on Sparse Grids". In: *Sparse Grids and Applications – Stuttgart 2014*. Ed. by Jochen Garcke and Dirk Pflüger. Vol. 109. Lecture Notes in Computational Science and Engineering. Switzerland: Springer International Publishing, 2016, pp. 315–336.

[49] Eric D. Vugrin, Chris Camphouse, and Daniel Sunderland. *Quantitative Resilience Analysis Through Control Design*. Research report SAND2009-5957. Sandia National Laboratories, 2009.

[50] Dirk Werner. *Funktionalanalysis*. Berlin-Heidelberg: Springer, 2018.

[51] Andreas Zeiser. "Fast Matrix-Vector Multiplication in the Sparse-Grid Galerkin Method". In: *Journal of Scientific Computing* 47.3 (2011), pp. 328–346.

Helmut Harbrecht and Ilja Kalmykov, Departement für Mathematik und Informatik, Universität Basel, Spiegelgasse 1, 4051 Basel

Helmut Harbrecht and Ilja Kalmykov, Departement für Mathematik und Informatik, Universität Basel, Spiegelgasse 1, 4051 Basel
Email address: {helmut.harbrecht,ilja.kalmykov}@unibas.ch