A Multigrid Solver for PDE-Constrained Optimization with Uncertain Inputs

Journal of Scientific Computing (2024) 101:13
https://doi.org/10.1007/s10915-024-02646-7

# A Multigrid Solver for PDE-Constrained Optimization with Uncertain Inputs

**Gabriele Ciaramella**[1] · **Fabio Nobile**[2] · **Tommaso Vanzan**[3]

Received: 6 October 2023 / Revised: 16 July 2024 / Accepted: 24 July 2024


**Abstract**
In this manuscript, we present a collective multigrid algorithm to solve efficiently the large saddle-point systems of equations that typically arise in PDE-constrained optimization under uncertainty, and develop a novel convergence analysis of collective smoothers and collective two-level methods. The multigrid algorithm is based on a collective smoother that at each iteration sweeps over the nodes of the computational mesh, and solves a reduced saddle-point system whose size is proportional to the number $N$ of samples used to discretized the probability space. We show that this reduced system can be solved with optimal $O(N)$ complexity. The multigrid method is tested both as a stationary method and as a preconditioner for GMRES on three problems: a linear-quadratic problem, possibly with a local or a boundary control, for which the multigrid method is used to solve directly the linear optimality system; a nonsmooth problem with box constraints and $L^1$-norm penalization on the control, in which the multigrid scheme is used as an inner solver within a semismooth Newton iteration; a risk-averse problem with the smoothed CVaR risk measure where the multigrid method is called within a preconditioned Newton iteration. In all cases, the multigrid algorithm exhibits excellent performances and robustness with respect to the parameters of interest.

**Keywords** Multigrid · Optimization under uncertainty · Random PDEs

**Mathematics Subject Classification** 65M55 · 65F10 · 65K10 · 49J55

✉ Tommaso Vanzan
tommaso.vanzan@polito.it

Gabriele Ciaramella
gabriele.ciaramella@polimi.it

Fabio Nobile
fabio.nobile@epfl.ch

[1] MOX, Dipartimento di Matematica, Politecnico di Milano, Milan, Italy

[2] CSQI Chair, Institute of Mathematics, Ecole Polytecnique Fédérale de Lausanne, Lausanne, Switzerland

[3] Dipartimento di Scienze Matematiche, Politecnico di Torino, Turin, Italy

Published online: 24 August 2024

## 1 Introduction

In this work, we present a multigrid method to solve the saddle point system

$$\mathcal{S}\mathbf{x} = \mathbf{f}, \tag{1}$$

where $\mathbf{x} = (\mathbf{y}, \mathbf{u}, \mathbf{p}) = (\mathbf{y}_1, \ldots, \mathbf{y}_N, \mathbf{u}, \mathbf{p}_1, \ldots, \mathbf{p}_N)^\top$, $\mathcal{S}$ has the block structure

$$\mathcal{S} = \begin{pmatrix} C_1 & & & & A_1^\top & & \\ & \ddots & & & & \ddots & \\ & & C_N & & & & A_N^\top \\ & & & G & D_1 & \ldots & D_N \\ A_1 & & & E_1 & & & \\ & \ddots & & \vdots & & & \\ & & A_N & E_N & & & \end{pmatrix}, \tag{2}$$

and all submatrices involved represent the discretization of some differential operators. More details on each block are provided in Sect. 2. Matrices such as (2) are often encountered while solving PDE-constrained optimization problems under uncertainty of the form

$$\begin{aligned} &\min_{u \in U} \mathcal{R}\left[Q(y(\omega), u)\right] \\ &\text{s.t. } y(\omega) \in V \text{ satisfies} \\ &\langle e(y(\omega), u, \omega), v\rangle = 0 \quad \forall v \in V, \text{ a.e. } \omega \in \Omega, \end{aligned} \tag{3}$$

where $u$ is the unknown *deterministic* control, $y(\omega)$ is the state variable which satisfies a random PDE constraint expressed by $e(\cdot, \cdot, \omega)$ for almost every realization $\omega$ of the randomness, $Q$ is a real-valued quantity of interest (cost functional) and $\mathcal{R}$ is a risk measure. The vectors $\{\mathbf{y}_j\}_{j=1}^N$ and $\{\mathbf{p}_j\}_{j=1}^N$ are the discretizations of the state and adjoint variables $y(\omega)$ and $p(\omega)$ at the $N$ samples in which the random PDE constraint is collocated. The vector $\mathbf{u}$ is the discretization of the deterministic control $u$. Problems of the form (3) are increasingly employed in applications. The PDE constraints typically represent some underlying physical model whose behaviour should be optimally controlled, and the randomness in the PDE allows one to take into account the intrinsic variability or lack of knowledge on some parameters entering the model. The introduction of a risk measure in (3) allows one to construct robust controls that take into account the distribution of the cost over all possible realizations of the random parameters. Therefore, the topic has received a lot of attention in the last years, see, e.g. [1–9].

However, few works have focused on efficient solvers for the optimality systems (1). A popular approach is to perform a Schur complement on $\mathbf{u}$ and solve the reduced system with a Krylov method (possibly with Conjugate Gradient), despite each iteration would then require the solution of $2N$ PDEs, with $A_j$ and $A_j^\top$ for $j = 1, \ldots, N$ [10]. For a full-space formulation, block diagonal preconditioners have been proposed in [11] and analyzed in [12], using both an algebraic approach based on Schur complement approximations and an operator preconditioning framework.

In this manuscript, we design a multigrid method to solve general problems of the form (1), present a detailed convergence analysis which, although in a simplified setting, is nontrivial and requires technical arguments, and show how this strategy can be used for the efficient solution of three different Optimal Control Problems Under Uncertainty (OCPUU). First, we consider a linear-quadratic OCPUU and use the multigrid algorithm directly to solve the

linear optimality system. Second, we consider a nonsmooth OCPUU with box constraints and $L^1$ regularization on the control. To solve such problem, we use the collective multigrid method as an inner solver within an outer semismooth Newton iteration. Incidentally, we show that the theory developed for the deterministic OCPs with $L^1$ regularization can be naturally extended to the class of OCPUU considered here. Third, we study a risk-averse OCPUU involving the smoothed Conditional Value at Risk (CVaR) and test the performance of the multigrid scheme in the context of a nonlinear preconditioned Newton method.

The multigrid algorithm is based on a collective smoother [13–15] that, at each iteration, loops over all nodes of the computational mesh (possibly in parallel), collects all the degrees of freedom related to a node, and updates them collectively by solving a reduced saddle-point problem. For classical (deterministic) PDE-constrained optimization problems with a distributed control, this reduced system has size $3 \times 3$, thus its solution is immediate [14]. In our context, the reduced problem has size $(2N + 1) \times (2N + 1)$, which can be large when dealing with a large number of samples. Fortunately, we show that it can be solved with optimal $O(N)$ complexity.

From the theoretical point of view, there are very few convergence analyses of collective smoothers even in the deterministic setting, namely [14] based on a local Fourier analysis, and [15] which relies on an algebraic approach. Notably, the presence of a low-rank block matrix in the reduced optimality system (obtained by eliminating the control) as well as the need to have stiffness and mass matrices with specific structure make it difficult to extend the analysis of [15]. We therefore present in this manuscript a fully new convergence analysis of collective smoothers and two-level collective multigrid methods in a simplified setting, which also covers the deterministic setting as a particular instance.

Let us remark that collective multigrid strategies have been applied to OCPUU in [16, 17] and in [18]. This manuscript differs from the mentioned works since, on the one hand, [16, 17] considers a *stochastic* control $u$, therefore for (almost) every realization of the random parameters a different control $u(\omega)$ is computed through the solution of a standard deterministic OCP. On the other hand, [18] considers a stochastic Galerkin discretization, and hence the corresponding optimality system has a structure which is very different from (2).

The multigrid algorithm presented here assumes that all state and adjoint variables are discretized on the same finite element mesh. The control can instead live on a subregion of the computational mesh, so that the algorithm is applicable also to optimization problems with local or boundary controls.

Finally, we remark that the multigrid solver proposed is based on a hierarchy of spatial discretizations corresponding to different levels of approximation, but the discretization of the probability space remains fixed, that is, the number of samples remains constant across the multigrid hierarchy. The extension of the multigrid algorithm to coarsening procedures also in the probability space will be the subject of future endeavours. We hint at possible approaches and challenges in Sect. 3 (see Remark 1). Nevertheless, we stress that the multigrid algorithm can already be incorporated within outer optimization routines that take advantange of different levels of approximations of the probability space, see, e.g., [7, 10, 19].

The rest of the manuscript is organized as follows. In Sect. 2 we introduce the notation, a classical linear-quadratic OCPUU, and interpret (2) as the matrix associated to the optimality system of a discretized OCPUU. Section 3 presents the collective multigrid algorithm, discusses implementation details and develops the convergence analysis. Further, the algorthm is numerically tested on the linear-quadratic OCPUU. In Sect. 4, we consider a nonsmooth OCPUU with box constraints and a $L^1$ regularization on the control. Section 5 deals with a risk-averse OCPUU. For each of these cases, we first show how the multigrid approach can

be integrated into the solution process, by detailing concrete algorithms, and then we present extensive numerical experiments to show the efficiency of the proposed framework. Finally, we draw our conclusions in Sect. 6.

## 2 A Linear-Quadratic Optimal Control Problem Under Uncertainty

Let $\mathcal{D} \subset \mathbb{R}^d$ be a Lipschitz bounded domain, $V \subset L^2(\mathcal{D})$ a Sobolev space (e.g. $H^1(\mathcal{D})$ equipped with suitable boundary conditions), and $(\Omega, \mathcal{F}, \mathbb{P})$ a complete probability space. Given a function $u$ belonging to a Hilbert space $U$, we consider the linear elliptic random PDE

$$a_\omega(y, v) = \langle \mathcal{B}u, v\rangle, \forall v \in V, \quad \mathbb{P}\text{-a-e. } \omega \in \Omega, \tag{4}$$

where $a_\omega(\cdot,\cdot) : V \times V \to \mathbb{R}$ is a bilinear form and $\langle\cdot,\cdot\rangle$ denotes the duality between $V$ and $V'$. $\mathcal{B} : U \to V'$ is a continuous control operator allowing possibly for a local control (i.e. a control acting only on a subset $\mathcal{D}_0 \subset \mathcal{D}$) or a boundary control (i.e. a control acting as Neumann condition on a subset of $\partial\mathcal{D}$). To assure uniqueness and sufficient integrability of the solution of (4), we make the following additional assumption.

**Assumption 1** There exist two random variables $a_{\min}(\omega)$ and $a_{\max}(\omega)$ such that

$$0 < a_{\min}(\omega)\|v\|_V^2 \le a_\omega(v, v) \le a_{\max}(\omega)\|v\|_V^2, \quad \forall v \in V, \ \mathbb{P}\text{-a.e. } \omega \in \Omega,$$

and further $a_{\min}^{-1}$ and $a_{\max}$ are in $L^p(\Omega)$ for some $p \ge 4$.

Under Assumption 1, it is well-known (see, e.g., [20, 21]) that (4) admits a solution in $V$ for $\mathbb{P}$-a.e. $\omega$, and the solution $y$, interpreted as a $V$-valued random variable $y : \omega \in \Omega \mapsto y(\omega) \in V$, lies in the Bochner space $L^q(\Omega; V)$, $q \le p$, [22]. We often use the shorthand notation $y_\omega = y(\cdot, \omega)$ when the dependence on $x$ is not needed, or $y_\omega(u)$ if we wish to highlight the dependence on the control function $u$.

In this manuscript, we consider the minimization of functionals constrained by (4). Let us first focus on the linear-quadratic problem

$$\begin{aligned} &\min_{u\in U,\, y\in L^2(\Omega;V)} \frac{1}{2}\mathbb{E}\left[\|\mathcal{I}y_\omega - y_d\|^2_{L^2(\mathcal{D})}\right] + \frac{\nu}{2}\|u\|_U^2, \\ &\text{subject to} \\ &a_\omega(y_\omega, v) = \langle \mathcal{B}u + f, v\rangle, \quad \forall v \in V, \ \mathbb{P}\text{-a.e. } \omega \in \Omega, \end{aligned} \tag{5}$$

where $y_d \in L^2(\mathcal{D})$ is a target state, $f \in V'$, $\mathbb{E} : L^1(\Omega) \to \mathbb{R}$ is the expectation operator, $\nu > 0$, and $\mathcal{I}$ is the embedding operator from $V$ to $L^2(\mathcal{D})$. Introducing the linear control-to-state map $S : g \in V' \to y_\omega(g) \in L^2(\Omega; V)$, the reduced formulation of (5) is

$$\min_{u\in U} \frac{1}{2}\mathbb{E}\left[\|\mathcal{I}S(\mathcal{B}u + f) - y_d\|^2_{L^2(\mathcal{D})}\right] + \frac{\nu}{2}\|u\|_U^2. \tag{6}$$

Existence and uniqueness of the minimizer of (6) follows directly from standard variational arguments [1, 23–25]. Furthermore, due to Assumption 1, the optimal control $\overline{u}$ satisfies the variational equality

$$(\nu\overline{u} - \Lambda_U \mathcal{B}^\star S^\star \mathcal{I}^\star (y_d - S(\mathcal{B}\overline{u} + f)), v)_U = 0, \quad \forall v \in U, \tag{7}$$

where $\Lambda_U$ is the Riesz operator of $U$. The adjoint operator $S^\star : L^2(\Omega; V') \to V$ is characterized by $S^\star z = \mathbb{E}[p]$ where $p = p_\omega(x)$ is the solution of the adjoint equation

$$a_\omega(v, p_\omega) = \langle z(\omega), v\rangle, \quad \forall v \in V, \ \mathbb{P}\text{-a-e. } \omega \in \Omega. \tag{8}$$

The optimality condition (7) can thus be formulated as the optimality system

$$\begin{aligned} &a_\omega(y_\omega, v) = \langle \mathcal{B}\overline{u} + f, v\rangle, \quad \forall v \in V, \quad \mathbb{P}\text{-a-e. } \omega \in \Omega, \\ &a_\omega(v, p_\omega) = \langle \mathcal{I}^\star(y_d - y_\omega), v\rangle, \quad \forall v \in V, \ \mathbb{P}\text{-a-e. } \omega \in \Omega, \\ &(\nu\overline{u} - \Lambda_U \mathcal{B}^\star \mathbb{E}[p_\omega], v)_U = 0, \quad \forall v \in U. \end{aligned} \tag{9}$$

To solve numerically (5), we replace the exact expectation operator $\mathbb{E}$ of the objective functional by a quadrature formula $\widehat{\mathbb{E}}$ with $N$ nodes $\{\omega_i\}_{i=1}^N$ and positive weights $\{\zeta_i\}_{i=1}^N$, namely

$$\mathbb{E}[X] \approx \widehat{\mathbb{E}}[X] := \sum_{i=1}^N \zeta_i X(\omega_i), \quad \text{with} \quad \sum_{i=1}^N \zeta_i = 1.$$

Common quadrature formulae are Monte Carlo, Quasi-Monte Carlo and Gaussian formulae. The latter requires that the probability space can be parametrized by a (finite or countable) sequence of random variables $\{\chi_j\}_j$, each with distribution $\mu_j$, and the existence of a complete basis of tensorized $L^2_{\mu_j}$-orthonormal polynomials. Hence for the semi-discrete OCP, the $\mathbb{P}$-a.e. PDE-constraint is naturally collocated onto the nodes of the quadrature formula.

Concerning the space domain, we consider a family of regular triangulations $\{\mathcal{T}_h\}_{h>0}$ of $\mathcal{D}$, and a Galerkin projection onto a conforming finite element space $V^h \subset V$ of continuous piecewise polynomial functions of degree $r$ over $\mathcal{T}_h$. $N_h$ is the dimension of $V^h$ and $\{\phi_i\}_{i=1}^{N_h}$ is a nodal Lagrangian basis. We discretize the state and adjoint variables on the same finite element space. The control variable is discretized on the finite element space $U_h = \operatorname{span}\{\psi_i\}_{i=1}^{N_u}$, where $N_u$ is possibly strictly smaller than $N_h$ in case of a local or a boundary control.

Once fully discretized, (9) can be expressed as

$$\begin{pmatrix} M & & & & A_1^\top & & & \\ & \ddots & & & & \ddots & & \\ & & M & & & & A_N^\top & \\ & & & \nu M_U & -\zeta_1 B^\top & \dots & -\zeta_N B^\top & \\ A_1 & & & -B & & & & \\ & \ddots & & \vdots & & & & \\ & & A_N & -B & & & & \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{u} \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{pmatrix} = \begin{pmatrix} M\mathbf{y}_d \\ \vdots \\ M\mathbf{y}_d \\ \mathbf{0} \\ M\mathbf{f} \\ \vdots \\ M\mathbf{f} \end{pmatrix}, \tag{10}$$

where $A_j$ are the stiffness matrices associated to the bilinear forms $a_{\omega_j}(\cdot,\cdot)$, $M$ and $M_U$ are mass matrices corresponding to the finite element spaces $V_h$ and $U_h$, $B$ is the discretization of the control operator, $\mathbf{y}_d$ and $\mathbf{f}$ are the finite element discretizations of $y_d$ and $f$ respectively, while $\mathbf{y}_j$ and $\mathbf{p}_j$ are the discretizations of $y_{\omega_j}$ and $p_{\omega_j}$. Notice that the matrix in (10) could be symmetrized by multipling the first and the last $N$ rows by the weights $\{\zeta_i\}_{i=1}^N$. This would be also consistent with the theoretical interpretation of the blocks of the saddle point system as discretizations of continuous inner products. From the numerical point of view, we have not observed relevant advantanges in maintaining the weights. Since for more general problems (see, e.g., Sec. 4) the symmetry of the saddle point system cannot be recovered by multiplying some equations by the quadrature weights, we do not consider the symmetrized version in this work.

## 3 Collective Multigrid Scheme

In this section, we describe the multigrid algorithm to solve the full space optimality system (10). First, we consider a distributed control, so that $u$ lives on the whole computational mesh and $B = M$. Local and boundary controls are discussed at the end of the section. Second, for the sake of generality, we consider the more general matrix (2), so that our discussion covers also the different saddle-point matrices obtained in Sects. 4 and 5.

For each node of the triangulation, let us introduce the vectors $\widetilde{\mathbf{y}}_i$ and $\widetilde{\mathbf{p}}_i$,

$$\widetilde{\mathbf{y}}_i = \begin{pmatrix} (\mathbf{y}_1)_i \\ \vdots \\ (\mathbf{y}_N)_i \end{pmatrix} \in \mathbb{R}^N, \quad \widetilde{\mathbf{p}}_i = \begin{pmatrix} (\mathbf{p}_1)_i \\ \vdots \\ (\mathbf{p}_N)_i \end{pmatrix} \in \mathbb{R}^N, \quad i = 1, \dots, N_h,$$

which collect the degrees of freedom associated to the $i$-th node, the scalar $u_i = (\mathbf{u})_i$, and the restriction operators $R_i \in \mathbb{R}^{(2N+1)\times((2N+1)N_h)}$ such that

$$R_i \begin{pmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \widetilde{\mathbf{y}}_i \\ u_i \\ \widetilde{\mathbf{p}}_i \end{pmatrix} =: \mathbf{x}_i. \tag{11}$$

The prolongation operators are $P_i := R_i^\top$, while the reduced matrices $\widetilde{S}_i := R_i S P_i \in \mathbb{R}^{(2N+1)\times(2N+1)}$ represent a condensed saddle-point matrix on the $i$-th node, and satisfy

$$\widetilde{S}_i = \begin{pmatrix} \mathrm{diag}(\mathbf{c}_i) & 0 & \mathrm{diag}(\mathbf{a}_i) \\ 0 & (G)_{i,i} & \mathbf{d}_i^\top \\ \mathrm{diag}(\mathbf{a}_i) & \mathbf{e}_i & 0 \end{pmatrix}$$

with $\mathbf{c}_i := ((C_1)_{i,i}, \dots, (C_N)_{i,i})^\top$, $\mathbf{a}_i := ((A_1)_{i,i}, \dots, (A_N)_{i,i})^\top$, $\mathbf{e}_i = ((E_1)_{i,i}, \dots, (E_N)_{i,i})^\top$, $\mathbf{d}_i = ((D_1)_{i,i}, \dots, (D_N)_{i,i})^\top$, where $\mathrm{diag}(\mathbf{v})$ denotes a diagonal matrix with the components of $\mathbf{v}$ on the main diagonal.

Given an initial vector $\mathbf{x}^0$, a Jacobi-type collective smoothing iteration computes for $n = 1, \dots, n_1$,

$$\mathbf{x}^n = \mathbf{x}^{n-1} + \theta \sum_{i=1}^{N_h} P_i \widetilde{S}_i^{-1} R_i \left(\mathbf{f} - S\mathbf{x}^{n-1}\right), \tag{12}$$

where $\theta \in (0, 1]$ is a damping parameter. Gauss-Seidel variants can straightforwardly be defined. Next, we consider a sequence of meshes $\left\{\mathcal{T}_{h_\ell}\right\}_{\ell=\ell_{\min}}^{\ell_{\max}}$, which we assume for simplicity to be nested, and restriction and prolongator operators $R_{\ell-1}^{\ell}$, $P_{\ell-1}^{\ell}$ which map between grids $\mathcal{T}_{h_{\ell-1}}$ and $\mathcal{T}_{h_\ell}$. In the numerical experiments, the coarse matrices are defined recursively in a Galerkin fashion starting from the finest one, namely $S_\ell := R_\ell^{\ell+1} S_{\ell+1} P_\ell^{\ell+1}$ for $\ell \in \{1, \dots, \ell_{\max} - 1\}$. Nevertheless it is obviously possible to define $S_\ell$ as the discretization of the continuous saddle-point system onto the mesh $\mathcal{T}_{h_\ell}$. With this notation, the V-cycle collective multigrid is described by Algorithm 1, which can be repeated until a certain stopping criterion is satisfied. We used the notation *Collective_Smoothing*$(\cdot, \cdot, \cdot)$ to denote possible variants of (12) (e.g. Gauss-Seidel).

Notice that (12) requires to invert the matrices $S_i$ for each computational node. We now show that this can be done with optimal $O(N)$ complexity. Indeed, performing a Schur complement on $u_i$, the system $\widetilde{S}_i \mathbf{x}_i = \mathbf{f}_i$, with $\mathbf{f}_i = (\mathbf{f}_{p_i}, b_{u_i}, \mathbf{f}_{y_i})^\top$ can be solved exclusively

**Algorithm 1** V-cycle Collective Multigrid Algorithm - V-cycle($\mathbf{x}^0$,$\mathbf{f}$,$\ell$)

**if** $\ell = \ell_{\min}$, **then**
  set $\mathbf{x}^0 = S_{\ell_{\min}}^{-1}\mathbf{f}$. (direct solver)
**else**
  $\mathbf{x}^{n_1}$=Collective_Smoothing($\mathbf{x}^0$, $S_\ell$, $n_1$) ($n_1$ steps of coll. smoothing)
  $\mathbf{r} = \mathbf{f} - S_\ell \mathbf{x}^{n_1}$ (compute the residual)
  $\mathbf{e}_c$ =V-cycle($\mathbf{0}$, $R_{\ell-1}^{\ell}\mathbf{r}$, $\ell-1$). (recursive call)
  $\mathbf{x}^0 = \mathbf{x}^{n_1} + P_{\ell-1}^{\ell}\mathbf{e}_c$ (coarse correction)
  $\mathbf{x}^{n_2}$=Collective_Smoothing($\mathbf{x}^0$, $S_\ell$, $n_2$) ($n_2$ steps of coll. smoothing)
  Set $\mathbf{x}^0 = \mathbf{x}^{n_2}$ (update)
**end if**
**return** $\mathbf{x}^0$.

computing inverses of diagonal matrices and scalar products between vectors through

$$\begin{aligned} u_i &= \frac{b_{u_i} + \mathbf{d}_i^\top(\mathrm{diag}(\mathbf{a}_i)^{-1}\mathrm{diag}(\mathbf{c}_i)\mathrm{diag}(\mathbf{a}_i)^{-1}\mathbf{f}_{y_i} - \mathrm{diag}(\mathbf{a}_i)^{-1}\mathbf{f}_{p_i})}{(G)_{i,i} + \mathbf{d}_i^\top \mathrm{diag}(\mathbf{a}_i)^{-1}\mathrm{diag}(\mathbf{c}_i)\mathrm{diag}(\mathbf{a}_i)^{-1}\mathbf{e}_i}, \\ \widetilde{\mathbf{y}}_i &= (\mathrm{diag}(\mathbf{a}_i))^{-1}(\mathbf{f}_{y_i} - \mathbf{e}_i u_i), \\ \widetilde{\mathbf{p}}_i &= (\mathrm{diag}(\mathbf{a}_i))^{-1}(\mathbf{f}_{p_i} - \mathrm{diag}(\mathbf{c}_i)\widetilde{\mathbf{y}}_i). \end{aligned} \tag{13}$$

Notice that we should guaranteee that $\mathrm{diag}(\mathbf{a}_i)$ admits an inverse and that $(G)_{i,i} + \mathbf{d}_i^\top \mathrm{diag}(\mathbf{a}_i)^{-1}\mathrm{diag}(\mathbf{c}_i)\mathrm{diag}(\mathbf{a}_i)^{-1}\mathbf{e}_i \neq 0$. This has to be verified case by case, so we now focus on the specific matrix (10). On the one hand, the vectors $\mathbf{a}_i$ are strictly positive componentwise, since $(\mathbf{a}_i)_j = a_{\omega_j}(\phi_i, \phi_i) > 0 \; \forall i = 1, \ldots, N_h, \; j = 1, \ldots, N$ (due to Assumption 1). On the other hand, $(G)_{i,i} = \int_{\mathcal{D}} \psi_i^2(x)\, dx > 0$, while a direct calculation shows that

$$\mathbf{d}_i^\top \mathrm{diag}(\mathbf{a}_i)^{-1}\mathrm{diag}(\mathbf{c}_i)\mathrm{diag}(\mathbf{a}_i)^{-1}\mathbf{e}_i = (M)_{i,i}^3 \sum_{j=1}^{N} \zeta_j (A_j)_{i,i}^{-2} > 0,$$

which implies that the denominator in the first equation of (13) is strictly positive.

The collective smoother can be easily adjusted to accomodate local or boundary controls as discussed in [26] for deterministic OCPs. For all nodes $i$ for which a control basis function is present, the smoothing procedure remains that of (13). For all others computational nodes for which there is not a control basis function associated, the smoothing procedure becomes

$$\begin{aligned} \widetilde{\mathbf{y}}_i &= (\mathrm{diag}(\mathbf{a}_i))^{-1}\mathbf{f}_{yi}, \\ \widetilde{\mathbf{p}}_i &= (\mathrm{diag}(\mathbf{a}_i))^{-1}(\mathbf{f}_{pi} - \mathrm{diag}(\mathbf{c}_i)\widetilde{\mathbf{y}}_i), \end{aligned}$$

which is consistently obtained from (13) setting $u_i = 0$.

To conclude this section, we remark that the computational complexity of the smoothing procedure is of order $O(N_h N)$, thus linear with respect to the size of the saddle point-system. Provided that the V-cycle algorithm requires a constant number of iterations to converge as the number of levels increases, and that $N$ is not too large (so that the cost of the coarse solver is not dominant), the complexity of the multigrid algorithm can also be considered linear. In the next numerical experiments sections (Sects. 3.2, 4.1, 5.2), we show indeed that the number of iterations remains constant for several test cases.

***Remark 1*** (Extension to a hierarchy of samples) The multigrid algorithm presented is based on a hierarchy of spatial discretizations. However, the sample to discretize the probability

space remains fixed among the levels. If one relies on the stochastic collocation method to discretize the probability space, it is possible to envisage a multigrid algorithm that also involves a coarsening of the sample size, since for each sample set one could consider the associated stable interpolator which can then be evaluated onto a coarser or finer set of samples. Nevertheless, it is not clear at the moment the interplay between the smoothing and coarsening procedures, which is key for the efficient behaviour of a multigrid scheme. Future endeavours will investigate this interesting direction. For the rest of the manuscript we restrict oursevels to a hierarchy of spatial discretizations since on the one hand, the multigrid algorithm can already be embedded in other outer optimization algorithms that involve a hierarchy of samples [7, 10, 19, 27]. On the other hand, the reduced system can be solved with optimal $O(N)$ linear complexity, so that a coarsening in the number of samples may be superfluous.

### 3.1 Convergence Analysis

In this subsection, we present a convergence analysis of the collective multigrid algorithm in a simplified setting. Let $\mathcal{D} = (0, 1)$, and consider the random PDE

$$\eta(\omega) \int_0^1 \partial_x y(x, \omega) \partial_x v(x)\, dx = \int_0^1 (f(x) + u(x)) v(x)\, dx, \forall v \in V,\ \mathbb{P}\text{-a.e. } \omega \in \Omega, \tag{14}$$

where $\eta : \Omega \to \mathbb{R}^+$ is a positive valued random variable such that $\mathbb{E}\left[\eta^{-2}\right] < \infty$. Our goal is to minimize the objective functional of (5) constrained by (14). A discretization using finite differences and with $N$ Monte Carlo samples leads to the optimality system

$$\begin{pmatrix} \frac{\widetilde{I}}{N} & & & & \frac{\eta_1(\omega)}{N} A & & \\ & \ddots & & & & \ddots & \\ & & \frac{\widetilde{I}}{N} & & & & \frac{\eta_N(\omega)}{N} A \\ & & & \nu \widetilde{I} & -\frac{\widetilde{I}}{N} & \cdots & -\frac{\widetilde{I}}{N} \\ \frac{\eta_1(\omega)}{N} A & & & -\frac{\widetilde{I}}{N} & & & \\ & \ddots & & \vdots & & & \\ & & \frac{\eta_N(\omega)}{N} A & -\frac{\widetilde{I}}{N} & & & \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{u} \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{pmatrix} = \begin{pmatrix} \frac{\mathbf{y}_d}{N} \\ \vdots \\ \frac{\mathbf{y}_d}{N} \\ \mathbf{0} \\ \frac{\mathbf{f}}{N} \\ \vdots \\ \frac{\mathbf{f}}{N} \end{pmatrix}, \tag{15}$$

where $A$ is the tridiagonal matrix associated with the 1D Laplacian, with $2/h^2$ on the main diagonal, and $-1/h^2$ on the two adjacent diagonals, $h$ being the mesh size, $\widetilde{I} \in \mathbb{R}^{N_h \times N_h}$ is the identity matrix, and, compared to (10), the first and last blocks of $N$ equations are divided by $\frac{1}{N}$ to get a symmetric system. Despite the simplifying assumptions on the spatial discretization and on the random coefficient, the setting considered is illustrative as system (15) preserves the main features of (10), namely the specific block structure and the presence of random stiffness matrices.

To perform our analysis, we first eliminate the variable $\mathbf{u}$, and obtain the reduced matrix

$$\begin{pmatrix} \frac{\widetilde{I}}{N} & & & & \frac{\eta_1(\omega)}{N}A & & \\ & \ddots & & & & \ddots & \\ & & \frac{\widetilde{I}}{N} & & & & \frac{\eta_N(\omega)}{N}A \\ \frac{\eta_1(\omega)}{N}A & & & -\frac{\widetilde{I}}{\nu N^2} & \cdots & \cdots & -\frac{\widetilde{I}}{\nu N^2} \\ & \ddots & & \vdots & \vdots & \vdots & \vdots \\ & & \frac{\eta_N(\omega)}{N}A & -\frac{\widetilde{I}}{\nu N^2} & \cdots & \cdots & -\frac{\widetilde{I}}{\nu N^2} \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{pmatrix} = \begin{pmatrix} \frac{\mathbf{y}_d}{N} \\ \vdots \\ \frac{\mathbf{y}_d}{N} \\ \frac{\mathbf{f}}{N} \\ \vdots \\ \frac{\mathbf{f}}{N} \end{pmatrix}. \tag{16}$$

Next, let $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_{N_h})^\top \in \mathbb{R}^{(2NN_h)\times 1}$, where $\mathbf{z}_j = ((\mathbf{y}_1)_j, \ldots, (\mathbf{y}_N)_j, (\mathbf{p}_1)_j, \ldots, (\mathbf{p}_N)_j))^\top \in \mathbb{R}^{2N\times 1}$. Notice that $\mathbf{z}_j$ corresponds to the application of $R_i$ to $\mathbf{x}$ (see (11)), except for $u_i$ which has been previously eliminated. By reordering the unknowns as in $\mathbf{z}$, (16) can be written as $S\mathbf{z} = \widetilde{\mathbf{b}}$ for a suitable $\widetilde{\mathbf{b}}$ and

$$S = \begin{pmatrix} \widetilde{B} & B & & & & \\ B & \widetilde{B} & B & & & \\ & B & \widetilde{B} & B & & \\ & & \ddots & \ddots & \ddots & \\ & & & B & \widetilde{B} & B \\ & & & & B & \widetilde{B} \end{pmatrix} = \widetilde{I} \otimes \widetilde{B} + H \otimes B,$$

$$\widetilde{B} := \begin{pmatrix} \frac{I}{N} & D \\ D & -\frac{\mathbf{1}\mathbf{1}^\top}{\nu N^2} \end{pmatrix}, \quad B := \begin{pmatrix} 0 & -\frac{D}{2} \\ -\frac{D}{2} & 0 \end{pmatrix}, \quad H = \begin{pmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & 1 \\ & & & 1 & 0 \end{pmatrix},$$

where $I \in \mathbb{R}^{N\times N}$ is the identity matrix, $D$ is a diagonal matrix with $d_j := \frac{2\eta_j(\omega)}{h^2 N}$ on the diagonal, and $\mathbf{1} = (1, \ldots, 1)^\top \in \mathbb{R}^{N\times 1}$. In particular, a direct calculation verifies that the iteration matrix of (12) with $\theta = 1$ and with this new order of unknowns is equal to

$$\mathcal{G} = \mathcal{I} - (\widetilde{I} \otimes \widetilde{B}^{-1})(\widetilde{I} \otimes \widetilde{B} + H \otimes B) = -H \otimes C,$$

with $C := \widetilde{B}^{-1}B$, and $\mathcal{I} \in \mathbb{R}^{(2N_hN)\times(2N_hN)}$ being the identity matrix. We will next characterize precisely the spectrum of $\mathcal{G}$, which in turns gives an exact description of the convergence on the one-level collective smoother. To do so, we first study the spectrum of $C$ denoted by $\sigma(C)$.

**Lemma 2** *(*Spectrum of $C$*) The matrix $C$ has the spectrum*

$$\sigma(C) = -\frac{1}{2}\left\{1, 1 - r \pm i\sqrt{(1-r)r}\right\},$$

*with* $r = \frac{\widehat{\mathbb{E}}[\widetilde{\mathbf{d}}^{-2}]}{\nu + \widehat{\mathbb{E}}[\widetilde{\mathbf{d}}^{-2}]}$, $\widetilde{\mathbf{d}} \in \mathbb{R}^{N\times 1}$, $(\widetilde{\mathbf{d}})_j = Nd_j$, *and* $\widehat{\mathbb{E}}\left[\widetilde{\mathbf{d}}^{-2}\right] := \frac{1}{N}\sum_{j=1}^{N}(\widetilde{\mathbf{d}})_j^{-2}$. *The eigenvalue* $\lambda = -\frac{1}{2}$ *has algebraic multiplicity* $2N-2$ *and geometric multiplicity* $N-1$.

***Proof*** Since $\frac{I}{N}$ and $D$ are non singular, to compute $C$ we use the exact formula for the inverse of $\widetilde{B}$. Setting $\Gamma := \frac{1}{\nu N^2 + \mathbf{1}^\top \frac{D^{-2}}{N}\mathbf{1}} = \frac{1}{\nu N^2 + N^2 \widehat{\mathbb{E}}[\widetilde{\mathbf{d}}^{-2}]}$, with $(\widetilde{\mathbf{d}})_j = \frac{2\eta_j(\omega)}{h^2}$, we obtain

$$
\begin{aligned}
C = \widetilde{B}^{-1}B &= \frac{1}{2}\begin{pmatrix} -I + \frac{\Gamma}{N} D^{-1}\mathbf{1}\mathbf{1}^\top D^{-1} & -\Gamma D^{-1}\mathbf{1}\mathbf{1}^\top \\ \frac{D^{-1}}{N} - \frac{\Gamma}{N^2} D^{-2}\mathbf{1}\mathbf{1}^\top D^{-1} & -I + \frac{\Gamma}{N} D^{-2}\mathbf{1}\mathbf{1}^\top \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} -I & 0 \\ \frac{D^{-1}}{N} & -I \end{pmatrix} + \frac{\Gamma N}{2}\begin{pmatrix} \widetilde{\mathbf{d}}^{-1}\widetilde{\mathbf{d}}^{-\top} & -\widetilde{\mathbf{d}}^{-1}\mathbf{1}^\top \\ -\widetilde{\mathbf{d}}^{-2}\widetilde{\mathbf{d}}^{-\top} & \widetilde{\mathbf{d}}^{-2}\mathbf{1}^\top \end{pmatrix}.
\end{aligned}
$$

For simplicity, we focus on $\widehat{C} := -2C$, which can be written as

$$
\widehat{C} = \underbrace{\begin{pmatrix} I & 0 \\ -\frac{D^{-1}}{N} & I \end{pmatrix}}_{L} + \mathbf{a}\mathbf{c}^\top, \quad \text{with} \quad \mathbf{a} := \Gamma N \begin{pmatrix} -\widetilde{\mathbf{d}}^{-1} \\ \widetilde{\mathbf{d}}^{-2} \end{pmatrix}, \ \mathbf{c} := \begin{pmatrix} \widetilde{\mathbf{d}}^{-1} \\ -\mathbf{1} \end{pmatrix},
$$

that is, $\widehat{C}$ is the sum of a lower triangular matrix plus a rank-one perturbation. Notice that $L$ has eigenvalue $\lambda = 1$ with algebraic multiplicity $2N$ and geometric multiplicity $N$. The eigenspace associated to $\lambda = 1$ is $E_{\lambda=1}(L) := \operatorname{span}\{\mathbf{e}_j,\ j = N+1, \dots, 2N\}$, $\mathbf{e}_j$ being the $j$-th canonical vector. Next, if $N > 2$, $\widehat{C}$ has still eigenvalue $\lambda = 1$ since for any vector $\mathbf{v} = (0, \mathbf{v}_2)$, $\mathbf{v}_2 \in \mathbb{R}^{N\times 1}$, such that $\mathbf{1}^\top \mathbf{v}_2 = 0$, we have

$$
(L + \mathbf{a}\mathbf{c}^\top)\mathbf{v} = L\mathbf{v} = \mathbf{v}.
$$

Therefore, $\lambda = 1$ is an eigenvalue of $\widehat{C}$ with geometric multiplicity $N - 1$.

To find the remaining eigenvalues, we take a $\lambda \neq 1$ and consider

$$
\begin{aligned}
\det(L - \lambda I_{2N\times 2N} + \mathbf{a}\mathbf{c}^\top) &= \det(L - \lambda I_{2N\times 2N}) \det(I_{2N\times 2N} + (L - \lambda I_{2N\times 2N})^{-1}\mathbf{a}\mathbf{c}^\top) \\
&= (1-\lambda)^{2N}\left(1 + \mathbf{c}^\top (L - \lambda I_{2N\times 2N})^{-1}\mathbf{a}\right).
\end{aligned}
$$

A direct calculation leads to

$$
\mathbf{c}^\top (L - \lambda I_{2N\times 2N})^{-1}\mathbf{a} = (\mathbf{c}_1, \mathbf{c}_2)^\top \begin{pmatrix} \frac{I}{1-\lambda} & 0 \\ \frac{D^{-1}}{N(1-\lambda)^2} & \frac{I}{1-\lambda} \end{pmatrix} \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{pmatrix}, \tag{17}
$$

so that

$$
\det(L - \lambda I_{2N\times 2N} + \mathbf{a}\mathbf{c}^\top) = (1-\lambda)^{2N-2}\left(\lambda^2 - (2 + \mathbf{a}^\top\mathbf{c})\lambda + 1 + \mathbf{a}^\top\mathbf{c} + \mathbf{c}_2^\top \frac{D^{-1}}{N}\mathbf{a}_1\right),
$$

from which we conclude that $\lambda = 1$ has algebraic multiplicity $2(N-1)$. The remaining eigenvalues must be solutions of the second order equation. Using $\mathbf{a}^\top\mathbf{c} = -2\Gamma N \sum_{i=j}^{N} \widetilde{d}_j^{-2}$, $\mathbf{c}_2^\top \frac{D^{-1}}{N}\mathbf{a}_1 = \Gamma N \sum_{j=1}^{N} \widetilde{d}_j^{-2}$, recalling the definition of $\Gamma$ and $r$, and dividing by $-\frac{1}{2}$, one obtains the solutions $\lambda_{2N-1,2N} = -\frac{1}{2}\left\{1 - r \pm i\sqrt{(1-r)r}\right\}$, and the claim follows.

***Remark 2*** (Dependence on the regularization parameter) The regularization parameter $\nu$ enters into our convergence analysis only in the definition of $r$. In particular as $\nu \to 0$, $r \to 1$ and $|\lambda_{2N-1,2N}| \to 0$, and the convergence of the collective multigrid does not deteriorate (see Lemma 2). The robustness of the algorithm with respect to the (often troublesome) $\nu \to 0$ limit will be observed in the numerical experiments.

From Lemma 2, we deduce that $C$ admits the Jordan decomposition $CV = VJ$, with

$$J = \begin{pmatrix} -0.5 & 1 & & & & & & \\ & -0.5 & & & & & & \\ & & -0.5 & 1 & & & & \\ & & & -0.5 & & & & \\ & & & & \ddots & \ddots & & \\ & & & & & \lambda_{2N-1} & & \\ & & & & & & \lambda_{2N} \end{pmatrix}, \tag{18}$$

$$V = [\mathbf{v}_1, \widehat{\mathbf{v}}_1, \mathbf{v}_2, \widehat{\mathbf{v}}_2, \ldots, \mathbf{v}_{2N-1}, \mathbf{v}_{2N}],$$

where $\mathbf{v}_j$, $j = 1, \ldots, N-1$, are the eigenvectors of $C$, $\widehat{\mathbf{v}}_j$, $j = 1, \ldots, N-1$, are the generalized eigenvectors satisfying $(C - \lambda_j I)\widehat{\mathbf{v}}_j = \mathbf{v}_j$, and $\mathbf{v}_{2N-1}$ and $\mathbf{v}_{2N}$ are the eigenvectors associated to the two remaining eigenvalues $\lambda_{2N-1,2N}$.

Exploiting the Kronecker structure of $\mathcal{G}$, we obtain immediately the following two corollaries.

**Corollary 3** *(*Similarity transformation of $\mathcal{G}$*) For $i = 1, \ldots, N_h$ and $j = 1, \ldots 2N$, let $\delta_{i,j} := -\mu_j \lambda_i$, where $\lambda_i$ is an eigenvalue of $C$, and $\mu_j = 2\cos\left(\frac{j\pi}{N_h+1}\right)$. Then, $\mathcal{G}$ satisfies $\mathcal{G}Y = Y\widetilde{J}$, where $\widetilde{J}$ is an upper triangular matrix with $\delta_{i,j}$ on the diagonal, and the $k$-th column of $Y$, with $k = i + j - 1$ for some $i$ and $j$, is $Y_k = \boldsymbol{\varphi}_j \otimes V_i$, $V_i$ being the $i$-th column of $V$ defined in (18), and $(\boldsymbol{\varphi}_j)_i := \sin\left(\frac{ij\pi}{N_h+1}\right)$.*

***Proof*** We first notice that $H$ is a tridiagonal Toeplitz matrix, and it is well-known (see [28]) that has eigenvalues $\mu_j = 2\cos\left(\frac{j\pi}{N_h+1}\right)$ and eigenvectors of the form $(\boldsymbol{\varphi}_j)_i = \sin\left(\frac{ij\pi}{N_h+1}\right)$. Due to the properties of the Kronecker product, it is trivial to verify that

$$\mathcal{G}(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i) = -(H\boldsymbol{\varphi}_j) \otimes (C\mathbf{v}_i) = -\mu_j \lambda_i (\boldsymbol{\varphi}_j \otimes \mathbf{v}_i).$$

If instead we consider a generalized eigenvector $\widehat{\mathbf{v}}_i$, using the Jordan decomposition, we have

$$\mathcal{G}(\boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i) = -(H\boldsymbol{\varphi}_j) \otimes (C\widehat{\mathbf{v}}_i) = -\mu_j \lambda_i (\boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i) - \mu_j (\boldsymbol{\varphi}_j \otimes \mathbf{v}_i),$$

and the claim follows.

**Corollary 4** *(*Spectral radius of $\mathcal{G}$*) The spectral radius of $\mathcal{G}$ is strictly smaller than* 1*, and satisfies $\rho(\mathcal{G}) \leq 1 - \mathcal{O}\left(\frac{1}{N_h^2}\right)$. Therefore, the collective smoothing iteration converges.*

***Proof*** Corollary 3 shows that $\mathcal{G}$ is similar to the upper triangular matrix $\widetilde{J}$. Thus, its eigenvalues are equal to $\delta_{i,j} = -\mu_j \lambda_i$. Observing that $|\mu_j| < 2|\cos\left(\frac{\pi}{N_h+1}\right)|$ and $|\lambda_i| \leq 0.5$ for any $j, i$, the claim follows.

***Remark 3*** (Damping) The analysis has been carried out for the relaxation parameter $\theta = 1$. It is trivial to consider $\theta \neq 1$, since the iteration matrix is then $\mathcal{G}_\theta := (1 - \theta)\mathcal{I} + \theta\mathcal{G}$.

We next study the spectrum of the two-level collective multigrid algorithm, and assume that $N_h = 2^\ell - 1$ and $N_h^C = 2^{\ell-1} - 1$ for a $\ell \in \mathbb{N}$. As maps between the fine and coarse meshes, we choose the full weighting restriction matrix,

$$\widetilde{R} := \frac{1}{2}\begin{pmatrix} \frac{1}{2} & 1 & \frac{1}{2} & & \\ & \frac{1}{2} & 1 & \frac{1}{2} & \\ & & \cdots & & \\ & & & \frac{1}{2} & 1 & \frac{1}{2} \end{pmatrix} \in \mathbb{R}^{N_h^C \times N_h},$$

and the linear interpolation operator $\widetilde{P} := 2\widetilde{R}^\top$. In particular, the action of $\widetilde{R}$ and $\widetilde{P}$ on the frequencies $\boldsymbol{\varphi}_j$ can be characterized rigorously (see, e.g., [29, Lemma 4.17]). Let $\boldsymbol{\phi}_j \in \mathbb{R}^{N_h^C \times 1}$ with $(\boldsymbol{\phi}_j)_i = \sin\left(\frac{2ij\pi}{N_h+1}\right)$, $j = 1, \ldots, N_h$ and $i = 1, \ldots, N_h^C$. Further define $c_j := \cos\left(\frac{j\pi}{2(N_h+1)}\right)$ and $s_j := \sin\left(\frac{j\pi}{2(N_h+1)}\right)$. Then, for any $e_j,\ e_{\widetilde{j}} \in \mathbb{R}$, with $\widetilde{j} := N_h+1-j$ and $j = 1, \ldots, \frac{N_h+1}{2} - 1$,

$$\begin{aligned} \widetilde{R}\begin{pmatrix} \boldsymbol{\varphi}_j & \boldsymbol{\varphi}_{\widetilde{j}} \end{pmatrix}\begin{pmatrix} e_j \\ e_{\widetilde{j}} \end{pmatrix} &= \widetilde{R}\left(e_j\boldsymbol{\varphi}_j + e_{\widetilde{j}}\boldsymbol{\varphi}_{\widetilde{j}}\right) = (e_j c_j^2 - e_{\widetilde{j}} s_j^2)\boldsymbol{\phi}_j = \boldsymbol{\phi}_j\begin{pmatrix} c_j^2 & -s_j^2 \end{pmatrix}\begin{pmatrix} e_j \\ e_{\widetilde{j}} \end{pmatrix}, \\ \widetilde{P}\boldsymbol{\phi}_j &= (c_j^2\boldsymbol{\varphi}_j - s_j^2\boldsymbol{\varphi}_{\widetilde{j}}) = \begin{pmatrix} \boldsymbol{\varphi}_j & \boldsymbol{\varphi}_{\widetilde{j}} \end{pmatrix}\begin{pmatrix} c_j^2 \\ -s_j^2 \end{pmatrix}. \end{aligned} \tag{19}$$

Furthermore, $R\boldsymbol{\varphi}_{\overline{j}} = 0$ for $\overline{j} := \frac{N_h+1}{2}$. The iteration matrix of the two-level algorithm with one-step of presmoothing and no post-smoothing is

$$T := (I - RS_c^{-1}PS)\mathcal{G},$$

where $R = \widetilde{R} \otimes I$, $P = \widetilde{P} \otimes I$, and $S_C = RSP$.

**Lemma 5** *The two-level operator $T$ is similar to a block diagonal matrix whose diagonal blocks are:*

1 *The matrices $T_{ji} := \mathcal{G}_{ji} - R_j^\top \Pi_{ji}^{-1} R_j S_{ji} \mathcal{G}_{ji} \in \mathbb{R}^{4\times 4}$ for $j = 1, \ldots, \frac{N_h+1}{2} - 1$ and $i = 1, \ldots, N-1$, with*

$$\mathcal{G}_{ji} := \begin{pmatrix} \delta_{ji} & & -\mu_j & \\ & \delta_{\widetilde{j}i} & & -\mu_{\widetilde{j}} \\ & & \delta_{ji} & \\ & & & \delta_{\widetilde{j}i} \end{pmatrix}, \quad S_{ji} := \begin{pmatrix} (1-\delta_{ji}) & & & \\ & (1-\delta_{\widetilde{j}i}) & & \\ & & (1-\delta_{ji}) & \\ & & & (1-\delta_{\widetilde{j}i}) \end{pmatrix},$$

$$R_j := \begin{pmatrix} c_j^2 & -s_j^2 & & \\ & & c_j^2 & -s_j^2 \end{pmatrix}, \quad P_j = R_j^\top, \quad \Pi_{ji} := R_j S_{ji} R_j^\top.$$

2 *The matrices $\mathcal{G}_{\overline{j}i} = \begin{pmatrix} \delta_{\overline{j}i} & -\mu_{\overline{j}} \\ & \delta_{\overline{j}i} \end{pmatrix} \in \mathbb{R}^{2\times 2}$ for $\overline{j} = \frac{N_h+1}{2}$, and $i = 1, \ldots, N-1$.*

3 *The matrices $\widehat{T}_{ji} := \widehat{\mathcal{G}}_{ji} - \widehat{R}_j^\top \widehat{\Pi}_{ij}^{-1} \widehat{R}_j \widehat{S}_{ji} \widehat{\mathcal{G}}_{ij} \in \mathbb{R}^{2\times 2}$ for $j = 1, \ldots, \frac{N_h+1}{2} - 1$ and $i = 2N-1, 2N$, with*

$$\widehat{\mathcal{G}}_{ji} := \begin{pmatrix} \delta_{ji} & \\ & \delta_{\widetilde{j}i} \end{pmatrix}, \quad \widehat{S}_{ji} := \begin{pmatrix} (1-\delta_{ji}) & \\ & (1-\delta_{\widetilde{j}i}) \end{pmatrix},$$
$$\widehat{R}_j := \begin{pmatrix} c_j^2 & -s_j^2 \end{pmatrix}, \quad \widehat{P}_j = \widehat{R}_j^\top, \quad \widehat{\Pi}_{ji} := c_j^4(1-\delta_{ji}) + s_j^4(1-\delta_{ji}).$$

*4 The matrices* $\widehat{\mathcal{G}}_{\overline{j}i} = \begin{pmatrix} \delta_{\overline{j}i} & \\ & \delta_{\overline{j}i} \end{pmatrix} \in \mathbb{R}^{2\times 2}$ *for* $\overline{j} = \frac{N_h+1}{2}$, *and* $i = 2N-1, 2N$.

***Proof*** The proof follows closely the arguments presented in [30, 31] for the study of two-level iterative methods. It consists in studying the action of $T$ onto suitably defined subspaces, showing that these subspaces are invariant, and finally deriving a matrix representation of $T$ into a new basis. We start with the four dimensional subspaces $\mathcal{V}_{ji} := \text{span}\left\{\boldsymbol{\varphi}_j \otimes \mathbf{v}_i,\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i,\ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i,\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right\}$, for $j = 1, \dots, \frac{N_h+1}{2} - 1$, $i = 1, \dots, N-1$. For any quadruple of real numbers $e_j, e_{\widetilde{j}}, \widehat{e}_j, \widehat{e}_{\widetilde{j}}$, using $H\boldsymbol{\varphi}_j = \mu_j \boldsymbol{\varphi}_j$ and the Jordan decomposition of $C$, we obtain

$$\mathcal{G}\left(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} e_j \\ e_{\widetilde{j}} \\ \widehat{e}_j \\ \widehat{e}_{\widetilde{j}} \end{pmatrix}$$
$$= \left(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} \delta_{ji} & & -\mu_j & \\ & \delta_{\widetilde{j}i} & & -\mu_{\widetilde{j}} \\ & & \delta_{ji} & \\ & & & \delta_{\widetilde{j}i} \end{pmatrix} \begin{pmatrix} e_j \\ e_{\widetilde{j}} \\ \widehat{e}_j \\ \widehat{e}_{\widetilde{j}} \end{pmatrix}.$$

Next, since $\widehat{\mathbf{v}}_i$ satisfies $(C - \lambda_i I)\widehat{\mathbf{v}}_i = \mathbf{v}_i$, it holds $B\widehat{\mathbf{v}}_i = \widetilde{B}(\mathbf{v}_i + \lambda_i \widehat{\mathbf{v}}_i)$, hence,

$$S\mathcal{G}\left(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} e_j \\ e_{\widetilde{j}} \\ \widehat{e}_j \\ \widehat{e}_{\widetilde{j}} \end{pmatrix}$$
$$= \left(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} (1-\delta_{ji}) & & -\mu_j & \\ & (1-\delta_{\widetilde{j}i}) & & -\mu_{\widetilde{j}} \\ & & (1-\delta_{ji}) & \\ & & & (1-\delta_{\widetilde{j}i}) \end{pmatrix} G_{ji} \begin{pmatrix} e_j \\ e_{\widetilde{j}} \\ \widehat{e}_j \\ \widehat{e}_{\widetilde{j}} \end{pmatrix},$$

and recalling (19),

$$RS\mathcal{G}\left(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i\ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i\ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} e_j \\ e_{\widetilde{j}} \\ \widehat{e}_j \\ \widehat{e}_{\widetilde{j}} \end{pmatrix}$$
$$= \left(\boldsymbol{\phi}_j \otimes \mathbf{v}_i\ \boldsymbol{\phi}_j \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} c_j^2 & -s_j^2 & & \\ & & c_j^2 & -s_j^2 \end{pmatrix} S_{ji} G_{ji} \begin{pmatrix} e_j \\ e_{\widetilde{j}} \\ \widehat{e}_j \\ \widehat{e}_{\widetilde{j}} \end{pmatrix}.$$

We now consider the coarse correction.

$$
\begin{aligned}
&S_c \left(\boldsymbol{\phi}_j \otimes \mathbf{v}_i \ \boldsymbol{\phi}_j \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} e_j^c \\ \widehat{e}_j^c \end{pmatrix} = RSP \left(\boldsymbol{\phi}_j \otimes \mathbf{v}_i \ \boldsymbol{\phi}_j \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} e_j^c \\ \widehat{e}_j^c \end{pmatrix} \\
&= RS \left(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i \ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i \ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i \ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right) R_j^\top \begin{pmatrix} e_j^c \\ \widehat{e}_j^c \end{pmatrix} \\
&= \left(\boldsymbol{\phi}_j \otimes \mathbf{v}_i \ \boldsymbol{\phi}_j \otimes \widehat{\mathbf{v}}_i\right) R_j S_{ji} R_j^\top \begin{pmatrix} e_j^c \\ \widehat{e}_j^c, \end{pmatrix} = \left(\boldsymbol{\phi}_j \otimes \mathbf{v}_i \ \boldsymbol{\phi}_j \otimes \widehat{\mathbf{v}}_i\right) \Pi_{ji} \begin{pmatrix} e_j^c \\ \widehat{e}_j^c, \end{pmatrix}
\end{aligned}
$$

which implies

$$
S_c^{-1} \left(\boldsymbol{\phi}_j \otimes \mathbf{v}_i \ \boldsymbol{\phi}_j \otimes \widehat{\mathbf{v}}_i\right) = \left(\boldsymbol{\phi}_j \otimes \mathbf{v}_i \ \boldsymbol{\phi}_j \otimes \widehat{\mathbf{v}}_i\right) \Pi_{ij}^{-1}.
$$

Putting all together, we get

$$
\begin{aligned}
&T \left(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i \ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i \ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i \ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} e_j \\ e_{\widetilde{j}} \\ \widehat{e}_j \\ \widehat{e}_{\widetilde{j}} \end{pmatrix} \\
&= \left(\boldsymbol{\varphi}_j \otimes \mathbf{v}_i \ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \mathbf{v}_i \ \boldsymbol{\varphi}_j \otimes \widehat{\mathbf{v}}_i \ \boldsymbol{\varphi}_{\widetilde{j}} \otimes \widehat{\mathbf{v}}_i\right) \underbrace{\left(\mathcal{G}_{ji} - R_j^\top \Pi_{ji}^{-1} R_j S_{ji} \mathcal{G}_{ji}\right)}_{T_{ji}} \begin{pmatrix} e_j \\ e_{\widetilde{j}} \\ \widehat{e}_j \\ \widehat{e}_{\widetilde{j}} \end{pmatrix}.
\end{aligned}
$$

This conclude the first part of the proof. We now consider the subspaces spanned by $\boldsymbol{\varphi}_{\overline{j}} \otimes \mathbf{v}_i$, $\boldsymbol{\varphi}_{\overline{j}} \otimes \widehat{\mathbf{v}}_i$ for $i = 1, \ldots, N-1$. Since $R\boldsymbol{\varphi}_{\overline{j}} = 0$, we immediately have

$$
T \left(\boldsymbol{\varphi}_{\overline{j}} \otimes \mathbf{v}_i \ \boldsymbol{\varphi}_{\overline{j}} \otimes \widehat{\mathbf{v}}_i\right) \begin{pmatrix} e_{\overline{j}} \\ \widehat{e}_{\overline{j}} \end{pmatrix} = \left(\boldsymbol{\varphi}_{\overline{j}} \otimes \mathbf{v}_i \ \boldsymbol{\varphi}_{\overline{j}} \otimes \widehat{\mathbf{v}}_i\right) \mathcal{G}_{\overline{j}i} \begin{pmatrix} e_{\overline{j}} \\ \widehat{e}_{\overline{j}} \end{pmatrix}, \quad \mathcal{G}_{\overline{j}i} := \begin{pmatrix} \delta_{\overline{j}i} & -\mu_{\overline{j}} \\ & \delta_{\overline{j}i} \end{pmatrix},
$$

and this proves the second claim. As third set of subspaces, we consider those spanned by respectively $(\boldsymbol{\varphi}_j \otimes v_{2N-1}, \boldsymbol{\varphi}_{\widetilde{j}} \otimes v_{2N-1})$, and $(\boldsymbol{\varphi}_j \otimes v_{2N}, \boldsymbol{\varphi}_{\widetilde{j}} \otimes v_{2N})$. Following the same calculations of the first part of the proof we obtain for $i = 2N-1$ and $i = 2N$,

$$
\begin{aligned}
&T \left(\boldsymbol{\varphi}_j \otimes v_i \ \boldsymbol{\varphi}_{\widetilde{j}} \otimes v_i\right) \begin{pmatrix} e_j \\ e_{\overline{j}} \end{pmatrix} \\
&= \left(\boldsymbol{\varphi}_j \otimes v_i \ \boldsymbol{\varphi}_{\widetilde{j}} \otimes v_i\right) \left(\widehat{\mathcal{G}}_{ji} - \widehat{R}_j^\top \widehat{\Pi}_{ji}^{-1} \widehat{R}_j \widehat{S}_{ji} \mathcal{G}_{ji}\right) \begin{pmatrix} e_j \\ e_{\overline{j}} \end{pmatrix}
\end{aligned}
$$

The proof of the fourth claim is identical to that of the second part and it is skipped for the sake of brevity. By considering a matrix $V$ that has column-block wise the basis for the subspaces we considered, it is immediate to deduce that $TV = V\widetilde{T}$, where $\widetilde{T}$ is a block diagonal matrix with the blocks we computed.

***Remark 4*** (Generalization to arbitrary pre- and post-smoothing steps) Lemma (5) can be readily generalized to cover $n_1$ pre-smoothing steps and $n_2$ post-smoothing steps, but taking suitable powers of the matrices $\mathcal{G}_{ji}$, $\widehat{\mathcal{G}}_{\bar{j}i}$, $\widehat{\mathcal{G}}_{ji}$ and $\widehat{\mathcal{G}}_{\bar{j}i}$. For instance, the matrix $T_{ji}$ of part one becomes

$$T_{ji} := \mathcal{G}_{ji}^{n_2}(I_{4\times 4} - R_j^\top \Pi_{ji}^{-1} R_j S_{ji})\mathcal{G}_{ji}^{n_1}.$$

**Theorem 6** (Spectrum and convergence of the two-level algorithm) *The spectrum of the matrix* $T = \mathcal{G}^{n_2}(I - RS_c^{-1}PS)\mathcal{G}^{n_1}$ *is*

$$\sigma(T) = \{0\} \cup \left\{ \frac{c_j^4(1-\delta_{ji})\delta_{\widetilde{j}i}^{n_1+n_2} + s_j^4(1-\delta_{\widetilde{j}i})\delta_{ji}^{n_1+n_2}}{c_j^4(1-\delta_{ji}) + s_j^4(1-\delta_{\widetilde{j}i})},\ j = 1,\dots,\frac{N_h+1}{2} - 1,\ i = 1,\dots,2N \right\}. \tag{20}$$

*Further, the spectral radius of* $T$ *is strictly smaller than 1, hence the two-level collective multigrid algorithm converges.*

***Proof*** Since $T$ is similar to a block diagonal matrix, with blocks defined in Lemma 5, it is sufficient to compute the spectrum of each block. Further, the spectrum of $T$ is equal to that of $(I - RS_c^{-1}PS)\mathcal{G}^{n_1+n_2}$. Hence, we start considering the blocks $T_{ji} = (I_{4\times 4} - R_j^\top \Pi_{ji}^{-1} R_j S_{ji})\mathcal{G}_{ji}^{n_1+n_2}$. Direct calculations show that

$$(I_{4\times 4} - R_j^\top \Pi_{ji}^{-1} R_j S_{ji}) = \left(\begin{array}{cc|c} 1 - \frac{c_j^4(1-\delta_{ji})}{\gamma} & \frac{c_j^2 s_j^2(1-\delta_{\widetilde{j}i})}{\gamma} & \\ \frac{c_j^2 s_j^2(1-\delta_{ji})}{\gamma} & 1 - \frac{s_j^4(1-\delta_{\widetilde{j}i})}{\gamma} & X \\ \hline & & 1 - \frac{c_j^4(1-\delta_{ji})}{\gamma} \quad \frac{c_j^2 s_j^2(1-\delta_{\widetilde{j}i})}{\gamma} \\ & & \frac{c_j^2 s_j^2(1-\delta_{ji})}{\gamma} \quad 1 - \frac{s_j^4(1-\delta_{\widetilde{j}i})}{\gamma} \end{array}\right), \tag{21}$$

where the expression of $X \in \mathbb{R}^{4\times 4}$ will not be needed in the following and $\gamma := c_j^4(1 - \delta_{ji}) + s_j^4(1-\delta_{\widetilde{j}i})$. Since the product of two upper triangular matrices is still upper triangular, it follows that

$$(I_{4\times 4} - R_j^\top \Pi_{ji}^{-1} R_j S_{ji})\mathcal{G}_{ji}^{n_1+n_2} = \begin{pmatrix} K & \widetilde{X} \\ & K \end{pmatrix},$$

with

$$K := \frac{1}{\gamma}\begin{pmatrix} s_j^4(1-\delta_{\widetilde{j}i})\delta_{ji}^{n_1+n_2} & c_j^2 s_j^2(1-\delta_{\widetilde{j}i})\delta_{\widetilde{j}i}^{n_1+n_2} \\ c_j^2 s_j^2(1-\delta_{ji})\delta_{ji}^{n_1+n_2} & c_j^4(1-\delta_{ji})\delta_{\widetilde{j}i}^{n_1+n_2} \end{pmatrix},$$

and whose eigenvalues are $\kappa_1^{ji} = \frac{c_j^4(1-\delta_{ji})\delta_{\widetilde{j}i}^{n_1+n_2} + s_j^4(1-\delta_{\widetilde{j}i})\delta_{ji}^{n_1+n_2}}{c_j^4(1-\delta_{ji}) + s_j^4(1-\delta_{\widetilde{j}i})}$ and $\kappa_2 = 0$. Next, $\mathcal{G}_{\bar{j}i}^{n_1+n_2}$ and $\widehat{\mathcal{G}}_{\bar{j}i}^{n_1+n_2}$ have trivially eigenvalues equal to $\delta_{\bar{j}i}^{n_1+n_2}$, which are all equal to zero since $\mu_{\bar{j}} = 0$.

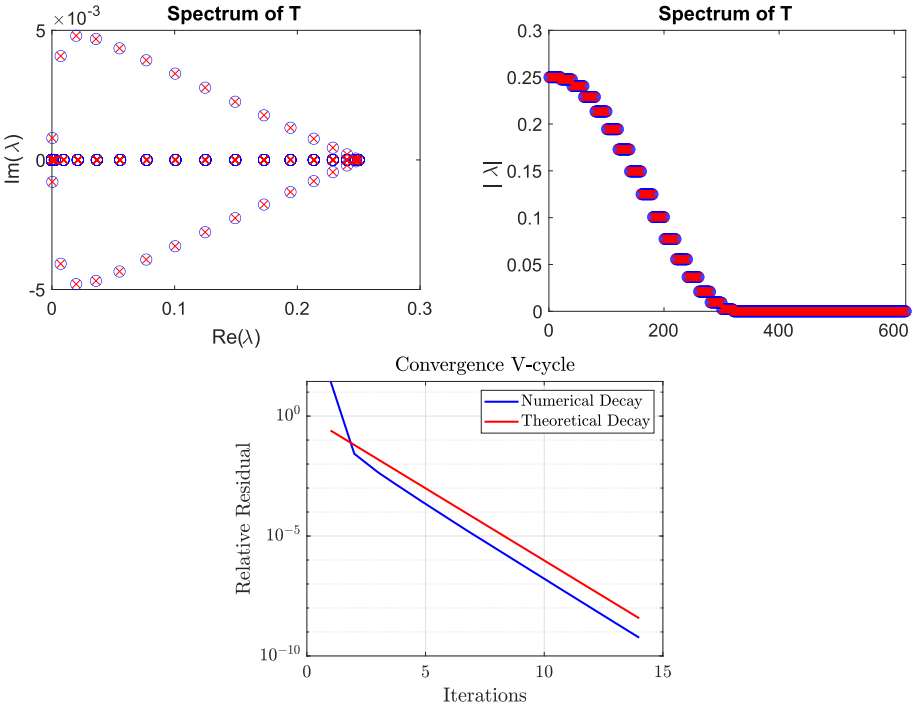


**Fig. 1** Top row: graphical representation of the spectrum of $T$ for $N_h = 31$, $N = 10$, $\nu = 10^{-2}$ and $n_1 = n_2 = 1$. The blue circles are obtained by computing numerically the eigenvalues of $T$. The red crosses are obtained through the formulae of Theorem 6. Bottom row: comparison between the numerical and theoretical convergence of the two-level algorithm

Further, direct calculations show that $\widehat{T}_{ji}$ has also two eigenvalues equal, again, to $\kappa_1^{ji}$ and $\kappa_2$. Taking into account the range of the indices of $j$ and $i$ for each blocks, we obtain the characterization of the spectrum, and since $|\delta_{ji}| < 1$ and $|\delta_{\widetilde{ji}}| < 1$, we conclude that the spectral radius of $T$ is smaller than one.

Figure 1 shows the spectrum of $T$ where, for visualization purposes, we set $N_h = 31$ and $N = 10$. In particular, the right panel shows that the spectrum is grouped into $\frac{N_h-1}{2}$ clusters, in which each eigenvalue is repeated approximately $2N$ times (approximately, because $C$ has two eigenvalues, $\lambda_{2N-1,2N}$ slightly different from 0.5.)

***Remark 5*** (Extension of the analysis to the deterministic setting) Our analysis also represents a novel approach to study the convergence of collective smoothing iterations in the case of a deterministic PDE constraint by setting $N = 1$. Retracing the analysis, we observe that $C$ has only two eigenvalues equal to $\lambda_{2N-1,2N}$ and $\mathcal{G}$ is diagonal. $T$ can then be diagonalized more easily, and its spectrum is still characterized by (20), where the index $i$ assumes only the values $2N - 1$ and $2N$.

***Remark 6*** (Extension to the two and three dimensional physical space) The analysis could be extended to square or cube domains. Due to the Kronecker product structure between spatial and probability quantities, only the matrix $H$ would have to change, and its eigenvectors would be the tensorized product of sine functions. Similarly, the action of the operators $\widetilde{R}$ and $\widetilde{P}$ would be represented by more complicated matrices.

This concludes our theoretical study of the convergence of the two-level collective multigrid algorithm. The next sections will focus on analyzing its numerical performances in different cases.

### 3.2 Numerical Experiments

We now show the performance of Algorithm 1 and its robustness with respect to several parameters for the solution of (10). We first consider the state equation

$$\begin{aligned} a_\omega(y_\omega, v) &= \int_{\mathcal{D}} \kappa(x,\omega)\nabla y(x,\omega)\cdot\nabla v(x)\,dx \\ &= \int_{\mathcal{D}} u(x)v(x)\,dx, \quad \forall v \in V,\ \mathbb{P}\text{-a.e. } \omega \in \Omega, \end{aligned} \tag{22}$$

in the L-shaped domain $\mathcal{D} = (0,1)^2 \setminus \overline{(0.5,1)}^2$ discretized with a regular mesh of squares of edge $h_\ell = 2^{-\ell}$, which are then decomposed into two right triangles. We choose $\kappa(x,\omega)$ as an approximated log-normal diffusion field

$$\kappa(x,\omega) = e^{\sigma \sum_{j=1}^{M} \sqrt{\lambda_j} b_j(x) N_j(\omega)} \approx e^{g(x,\omega)}, \tag{23}$$

where $g(x,\omega)$ is a mean zero Gaussian field with Covariance function $Cov_g(x,y) = \sigma^2 e^{\frac{-\|x-y\|_2^2}{L^2}}$. The parameter $\sigma^2$ tunes the variance of the random field, while $L$ denotes the correlation length. The pairs $(b_j(x), \sigma^2\lambda_j)$ are the eigenpairs of $T : L^2(\mathcal{D}) \to L^2(\mathcal{D})$, $(Tf)(x) = \int_{\mathcal{D}} Cov_g(x,y) f(y)\,dy$, and $N_j \overset{iid}{\sim} \mathcal{N}(0,1)$. Assumption 1 is satisfied since $a_{\min}(\omega) = (\operatorname{ess\,inf}_{x\in\mathcal{D}} \kappa(x,\omega))^{-1}$ and $a_{\max}(\omega) = \|\kappa(\cdot,\omega)\|_{L^\infty(\mathcal{D})}$ are in $L^p(\Omega)$ for every $p < \infty$ [32]. The target state is $y_d = e^{y^2}\sin(2\pi x)\sin(2\pi y)$.

Table 1 shows the number of V-cycle iterations (Algorithm 1) and of GMRES iterations preconditioned by the V-cycle to solve (10) up to a tolerance of $10^{-9}$ on the relative (unpreconditioned) residual. Inside the V-cycle algorithm, we use $n_1 = n_2 = 2$ pre- and post-smoothing iterations based on the Jacobi relaxation (12) with a damping parameter $\theta = 0.5$ (the same value will be used for all numerical experiments in this manuscript). Numerically, we observed that Gauss-Seidel relaxations lead to very similar results. The number of levels of the V-cycle hierachy is denoted with $N_L$. The size of the largest linear system solved per sub-table is denoted by $N_{\max} = (2N+1)N_h$.

The first four sub-tables are based on a discretization of the probability space using the Stochastic Collocation method [33] on Gauss-Hermite tensorized quadrature nodes, since for $L^2 = 0.5$, setting $M = 3$ into (23) is enough to preserve 99% of the variance. In the fifth sub-table we set $L^2 = 0.1$ and use the Monte Carlo method, since we need $M = 15$ random variables to preserve 99% of the variance of the random field, and the Stochastic

**Table 1** Number of V-cycle (left) and preconditioned GMRES (right) iterations to solve (10) for a linear quadratic problem on the L-shaped domain $\mathcal{D} = (0, 1)^2 \backslash \overline{(0.5, 1)}^2$ with a distributed control

| $\nu$ | $10^{-2}$ | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|---|
| It. | 18 \| 11 | 19 \| 13 | 19 \| 15 | 19 \| 15 |
| $N_h = 705, N = 125, N_L = 3, \sigma^2 = 0.5, L^2 = 0.5, N_{\max} = 1.7710^5.$ | | | | |
| $\sigma^2$ | 0.1 | 0.5 | 1 | 1.5 |
| It. | 19 \| 13 | 19 \| 13 | 20 \| 13 | 20 \| 13 |
| $N_h = 705, N = 125, N_L = 3, \nu = 10^{-4}, L^2 = 0.5, N_{\max} = 1.7710^5.$ | | | | |
| $N_h(N_L)$ | 161 (2) | 705 (3) | 2945 (4) | |
| It. | 19 \| 13 | 19 \| 13 | 20 \| 13 | |
| $N = 125, \nu = 10^{-4}, \sigma^2 = 0.5, L^2 = 0.5, N_{\max} = 7.3910^5.$ | | | | |
| $N$ | 8 | 27 | 64 | 125 |
| It. | 19 \| 13 | 19 \| 13 | 19 \| 13 | 19 \| 13 |
| $N_h = 705, N_L = 3, \nu = 10^{-4}, \sigma^2 = 0.5, L^2 = 0.5, N_{\max} = 1.7710^5.$ | | | | |
| $N$ | 100 | 500 | 1000 | 2000 |
| It. | 22 \| 16 | 22 \| 15 | 22 \| 15 | 22 \| 15 |
| $N_h = 705, \nu = 10^{-4}, N_L = 3, \sigma^2 = 1.5, L^2 = 0.1, N_{\max} = 2.8210^6$ | | | | |

Collocation method suffers the curse of dimensionality. Remark that the multigrid algorithm is robust with respect to all parameters considered, namely the regularization parameter, the variance of the random field, the number of levels as the fine grid is refined, and the number of samples to discretize the probability space.

We mention that a family of block diagonal preconditioners for saddle-point matrices such as (2) were recently proposed in [11]. A detailed theoretical analysis was developed in [12] for distributed controls, in a more general setting than the one considered in this manuscript that covers a general finite element discretization of a $d$-dimensional domain, a general elliptic bilinear form, and an additional variance term in the cost functional. Their main attractive feature is the possibility to precondition fully in parallel the $2N$ PDEs. Nevertheless, their convergence deteriorates as $\nu \to 0$ (as several preconditioners built on the same technique see, e.g., [34, 35]), so that these preconditioners are hardly effective when $\nu$ is smaller than, say, $10^{-3}/10^{-4}$. The robustness of the multigrid algorithm as $\nu \to 0$ is definitely one of its most interesting properties. In terms of mesh refinement, both approaches are robust, provided that the $2N$ PDEs constraints are suitable preconditioned (e.g., with multigrid) in the approach of [11, 12]. Concerning the refinement of the discretization of the probability space, both methods are robust, and interestingly, both convergence analyses show a dependence on the approximated expected value of the square inverse of the coercivity constants of the stiffness matrices. One current disadvantage of the multigrid algorithm is the lack of coarsening with respect to the number of samples $N$, since the solution of the coarse problem might represent a bottleneck for very fine discretizations. In these circumstances, the capability of [11, 12] to handle the PDE constraints in parallel may be beneficial.

**Table 2** Number of V-cycle (left) and preconditioned GMRES (right) iterations to solve (10) for a linear quadratic problem on the square domain $\mathcal{D} = (0, 1)^2$ with a local control acting on $\mathcal{D}_0 = (0.25, 0.75)^2$

| | | | | |
|---|---|---|---|---|
| $\nu$ | $10^{-2}$ | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
| It. | 17 \| 11 | 20 \| 13 | 26 \| 16 | 26 \| 18 |
| $N_h = 961, N = 125, N_L = 3, \sigma^2 = 0.5, L^2 = 0.5.$ | | | | |
| $\sigma^2$ | 0.1 | 0.5 | 1 | 1.5 |
| It. | 20 \| 13 | 20 \| 13 | 20 \| 13 | 19 \| 13 |
| $N_h = 961, N = 125, N_L = 3, \nu = 10^{-4}, L^2 = 0.5.$ | | | | |
| $N_h(N_L)$ | 225 (2) | 961 (3) | 3969 (4) | |
| It. | 19 \| 12 | 20 \| 13 | 20 \| 13 | |
| $N = 125, \nu = 10^{-4}, \sigma^2 = 0.5, L^2 = 0.5.$ | | | | |
| $N$ | 8 | 27 | 64 | 125 |
| It. | 20 \| 13 | 20 \| 13 | 20 \| 13 | 20 \| 13 |
| $N_h = 961, N_L = 3, \nu = 10^{-4}, \sigma^2 = 0.5, L^2 = 0.5.$ | | | | |
| $N$ | 100 | 1000 | 2000 | |
| It. | 20 \| 14 | 21 \| 15 | 21 \| 14 | |
| $N_h = 961, \nu = 10^{-4}, N_L = 3, \sigma^2 = 1.5, L^2 = 0.1.$ | | | | |

Next, we consider the same problem (22)-(23) posed in the unit square domain $\mathcal{D} = (0, 1)^2$ with either a local control acting on the subset $\mathcal{D}_0 = (0.25, 0.75)^2 \subset \mathcal{D}$, or a Neumann boundary control acting on $\Gamma = (0, 1) \times \{0\} \subset \partial\mathcal{D}$. Tables 2 and 3 report the performances of the multigrid algorithm for these two cases. We stress once more the excellent robustness and efficiency of the multigrid algorithm in all regimes.

## 4 An Optimal Control Problem Under Uncertainty with Box-constraints and $L^1$ Penalization

In this section, we consider the nonsmooth OCPUU[1]

$$\begin{aligned} &\min_{u \in U_{ad}} \frac{1}{2}\mathbb{E}\left[\|y_\omega(u) - y_d\|^2_{L^2(\mathcal{D})}\right] + \frac{\nu}{2}\|u\|^2_{L^2(\mathcal{D})} + \beta\|u\|_{L^1(\mathcal{D})}, \\ &\quad \text{subject to} \\ &a_\omega(y_\omega(u), v) = (u + f, v)_{L^2(\mathcal{D})}, \quad \forall v \in V, \ \mathbb{P}\text{-a-e. } \omega \in \Omega, \\ &U_{ad} := \left\{v \in L^2(\mathcal{D}) : a \le u \le b \quad \text{almost everywhere in } \mathcal{D}\right\}, \end{aligned} \tag{24}$$

with $a < 0 < b$ and $\nu, \beta > 0$. Deterministic OCPs with a $L^1$ penalization lead to optimal controls which are sparse, i.e. they are nonzero only on certain regions of the domain $\mathcal{D}$ [36, 37]. Sparse controls can be of great interest in applications, because it is often not desirable, or even impossible, to control the system over the whole domain $\mathcal{D}$. For sparse OCPUU, we mention [38] where the authors considered both a simplified version of (24) in which the randomness enters linearly into the state equation as a force term, and a different optimization problem whose goal is to find a stochastic control $u(\omega)$ which has a similar sparsity pattern

[1] To keep a light notation, we omitted the continuous embedding operator from $L^2(\Omega; V)$ to $L^2(\Omega; L^2(\mathcal{D}))$ and from $L^2(\mathcal{D})$ to $V'$, see Sect. 2 and, e.g., [25, Sect. 2.13].

**Table 3** Number of V-cycle (left) and preconditioned GMRES (right) iterations to solve (10) for a linear quadratic problem on the square domain $\mathcal{D} = (0, 1)^2$ with a boundary control acting on $\Gamma = (0, 1) \times \{0\}$

| | | | | |
|---|---|---|---|---|
| $\nu$ | $10^{-2}$ | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
| It. | 17 \| 14 | 22 \| 15 | 23 \| 17 | 21 \| 16 |
| $N_h = 992, N = 125, N_L = 3, \sigma^2 = 0.5, L^2 = 0.5.$ | | | | |
| $\sigma^2$ | 0.1 | 0.5 | 1 | 1.5 |
| It. | 16 \| 13 | 17 \| 14 | 17 \| 14 | 17 \| 14 |
| $N_h = 992, N = 125, N_L = 3, \nu = 10^{-4}, L^2 = 0.5.$ | | | | |
| $N_h(N_L)$ | 240 (2) | 992 (3) | 4032 (4) | |
| It. | 16 \| 12 | 17 \| 14 | 21 \| 16 | |
| $N = 125, \nu = 10^{-4}, \sigma^2 = 0.5, L^2 = 0.5.$ | | | | |
| $N$ | 8 | 27 | 64 | 125 |
| It. | 16 \| 13 | 17 \| 14 | 17 \| 14 | 17 \| 14 |
| $N_h = 992, N_L = 3, \nu = 10^{-4}, \sigma^2 = 0.5, L^2 = 0.5.$ | | | | |
| $N$ | 100 | 1000 | 2000 | |
| It. | 18 \| 15 | 19 \| 16 | 20 \| 16 | |
| $N_h = 992, \nu = 10^{-4}, N_L = 3, \sigma^2 = 1.5, L^2 = 0.1.$ | | | | |

regardless of the realization $\omega$. Note further that the assumption $\nu > 0$ does not eliminate the nonsmoothness of the objective functional, but it regularizes the optimal solution $u$, and is needed to use the fast optimation algorithm described in the following.

The well-posedness of (24) follows directly from standard variational arguments [24, 25], being $U_{ad}$ a convex set, $\varphi(u) := \beta\|u\|_{L^1(\mathcal{D})}$ a convex function and the objective functional coercive. In particular, the optimal solution $\overline{u}$ satisfies the variational inequality ([39, Proposition 2.2])

$$(\nu\overline{u} - S^\star(y_d - S(\overline{u} + f)), \overline{u} - v) + \varphi(\overline{u}) - \varphi(v) \geq 0, \quad \forall v \in U_{ad}. \tag{25}$$

Through a pointwise discussion of the box constraints and an analysis of a Lagrange multiplier belonging to the subdifferential of $\varphi$ in $\overline{u}$, [36] showed that (25) can be equivalently formulated as the nonlinear equation $\mathcal{F}(\overline{u}) = 0$, with $\mathcal{F} : L^2(\mathcal{D}) \to L^2(\mathcal{D})$ defined as

$$\begin{aligned}\mathcal{F}(u) := u - \frac{1}{\nu}\Big( &\max(0, \mathcal{T}u - \beta) + \min(0, \mathcal{T}u + \beta) \\ &- \max(0, \mathcal{T}u - \beta - \nu b) - \min(0, \mathcal{T}u + \beta - \nu a)\Big),\end{aligned} \tag{26}$$

where $\mathcal{T} : L^2(\mathcal{D}) \ni u \to -S^\star(Su) + S^\star(y_d - Sf) \in L^2(\mathcal{D})$. Notice that $\mathcal{F}$ is nonsmooth due to the presence of the Lipschitz functions $\max(\cdot)$ and $\min(\cdot)$. Nevertheless, $\mathcal{F}$ can be shown to be semismooth [24], provided that $\mathcal{T}$ is continuously Fréchet differentiable, and further Lipschitz continuous interpreted as map from $L^2(\mathcal{D})$ to $L^r(\mathcal{D})$, with $r > 2$ [24, 40]. These conditions are satisfied also in our settings since $\mathcal{T}$ is affine and further the adjoint variable $p_\omega$, solution of (8) with $z = y_d - S(u + f)$, lies in $L^2(\Omega, H_0^1(\mathcal{D}))$ so that $\mathcal{T}u = \mathbb{E}[p_\omega] \in H_0^1(\mathcal{D}) \subset L^r(\mathcal{D})$, where $r > 2$ follows from Sobolev embeddings.

Hence, to solve (26) we use the semismooth Newton method whose iteration reads for $k = 1, 2, \ldots$ until convergence,

$$u^{k+1} = u^k + du^k, \quad \text{with} \quad \mathcal{G}(u^k)du^k = -\mathcal{F}(u^k), \tag{27}$$

$\mathcal{G}(u) : L^2(\mathcal{D}) \to L^2(\mathcal{D})$ being the generalized derivative of $\mathcal{F}$. Using the linearity of $\mathcal{T}$ and considering the supports of the weak derivatives of $\max(0, x)$ and $\min(0, x)$, we obtain that

$$\mathcal{G}(u)[v] = v + \frac{1}{\nu}\chi_{(I^+\cup I^-)}S^\star S v,$$

where $\chi$ is the charateristic function of the union of the disjoint sets

$$I^+ = \{x \in \mathcal{D} : 0 \le \mathcal{T}u - \beta \le \nu b\} \text{ and } I^- = \{x \in \mathcal{D} : \nu a \le \mathcal{T}u + \beta \le 0\}.$$

It is possible to show that the generalized derivative $\mathcal{G}(u)$ is invertible with bounded inverse for all $u$, the proof being identical to the deterministic case treated in [41]. This further implies that the semismooth Newton method (27) converges locally superlinearly [40]. We briefly summarize these results in the following proposition.

**Proposition 7** *Let the initialization $u^0$ be sufficiently close to the solution $\overline{u}$ of (24). Then the iterates $u^k$ generated by (27) converge superlinearly to $\overline{u} \in L^2(\mathcal{D})$.*

Introducing the supporting variables $dy_\omega^k$ and $dp_w^k$ in $L^2(\Omega; H_0^1(\mathcal{D}))$, the semismooth Newton equation $\mathcal{G}(u^k)du^k = -\mathcal{F}(u^k)$ may be rewritten as the equivalent saddle point system

$$\begin{aligned}
&a_\omega(dy_\omega^k, v) - (du^k, v) = 0, \quad \forall v \in V, \quad \mathbb{P}\text{-a-e. } \omega \in \Omega,\\
&a_\omega(v, dp_\omega^k) + (dy_\omega^k, v) = 0, \quad \forall v \in V, \ \mathbb{P}\text{-a-e. } \omega \in \Omega,\\
&(\nu\, du^k - \chi_{(I^+\cup I^-)}\mathbb{E}\left[dp_\omega^k\right], v)_{L^2(\mathcal{D})} = -\mathcal{F}(u^k), \quad \forall v \in L^2(\mathcal{D}).
\end{aligned} \tag{28}$$

Further, if we set $y^0 = S(f + u^0)$ and $p^0 = S^\star(y_d - y^0)$, due to the linearity of $S$ and $S^\star$, it holds $y^{k+1} = S(u^{k+1}) = y^k + dy^k$ and similarly $p^{k+1} = p^k + dp^k$. Once fully discretized and using the notation $\widehat{\mathbb{E}}[p_\omega] = \sum_{j=1}^N \zeta_j \mathbf{p}_j$, the optimality condition (26) can be expressed through the nonlinear finite-dimensional map $\mathbf{F} : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$,

$$\begin{aligned}
\mathbf{F}(\mathbf{u}) =& \mathbf{u} - \frac{1}{\nu}\Big(\max(0, \widehat{\mathbb{E}}\left[\mathbf{p}_\omega\right] - \beta) + \min(0, \widehat{\mathbb{E}}\left[\mathbf{p}_\omega\right] + \beta)\\
&- \max(0, \widehat{\mathbb{E}}\left[\mathbf{p}_\omega\right] - \beta - \nu b) - \min(0, \widehat{\mathbb{E}}\left[\mathbf{p}_\omega\right] + \beta - \nu a)\Big),
\end{aligned}$$

where the $\max(\cdot)$ and $\min(\cdot)$ functions act componentwise. Equation (28) leads to the saddle point system

$$\begin{pmatrix}
M & & & & A_1^\top & & & \\
& \ddots & & & & \ddots & & \\
& & M & & & & A_N^\top & \\
& & & M & -\zeta_1 M H^k & \ldots & -\zeta_N M H^k & \\
A_1 & & & -M & & & & \\
& \ddots & & \vdots & & & & \\
& & A_N & -M & & & &
\end{pmatrix}
\begin{pmatrix}
\mathbf{dy}_1^k \\ \vdots \\ \mathbf{dy}_N^k \\ \mathbf{du}^k \\ \mathbf{dp}_1^k \\ \vdots \\ \mathbf{dp}_N^k
\end{pmatrix}
=
\begin{pmatrix}
\mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{F}(\mathbf{u}^k) \\ \mathbf{0} \\ \vdots \\ \mathbf{0}
\end{pmatrix}, \tag{29}$$

where $H^k \in \mathbb{R}^{N_h \times N_h}$ is a diagonal matrix representing the charateristic function $\chi_{I_k^+ \cup I_k^-}$, namely

$$(H^k)_{i,i} = \frac{1}{\nu} \text{ if } i \in I_k^+ \cup I_k^- \quad \text{and} \quad (H^k)_{i,i} = 0 \text{ if } i \notin I_k^+ \cup I_k^-,$$

with

$$I_k^+ = \left\{ i : 0 \leq \widehat{\mathbb{E}}\left[\mathbf{p}^k\right] - \beta \leq \nu b \right\} \text{ and } I_k^- = \left\{ i : \nu a \leq \widehat{\mathbb{E}}\left[\mathbf{p}^k\right] + \beta \leq 0 \right\}. \tag{30}$$

To derive the expression of $H$, we assumed that a Lagrangian basis is used for the finite element space. Notice that (29) fits into the general form (2), and thus we use the collective multigrid algorithm to solve it. Further, with the notation of (2), it holds

$$(G)_{i,i} + d_i^\top \text{diag}(a_i)^{-1}\text{diag}(c_i)\text{diag}(a_i)^{-1} e_i = (M)_{i,i} + (M)_{i,i}^3 \sum_{j=1}^{N} \zeta_j (A_j)_{i,i}^{-2} > 0$$

if $i \in I^+ \cup I^-$, and

$$(G)_{i,i} + d_i^\top \text{diag}(a_i)^{-1}\text{diag}(c_i)\text{diag}(a_i)^{-1} e_i = (M)_{i,i} > 0,$$

if $i \notin I^+ \cup I^-$. The collective multigrid iteration is then well-defined.

The overall semismooth Newton Algorithm is summarized in Algorithm 2. At each iteration we solve (29) using the collective multigrid algorithm (line 4) and update the active sets given the new iteration (line 10). Notice that in order to globalize the convergence, we consider a line-search step (lines 6-8) performed on the merit function $\phi(\mathbf{u}) = \sqrt{\mathbf{F}(\mathbf{u})^\top M \mathbf{F}(\mathbf{u})}$ [42].

**Algorithm 2** Globalized semismooth Newton Algorithm to solve $\mathbf{F}(\mathbf{u}) = 0$

**Require:** $\mathbf{u}^0$, Tol $\in \mathbb{R}^+$, $\sigma, \rho \in (0, 1)$.
1: $\mathbf{y}_j^0 = A_j^{-1}(M(\mathbf{f} + \mathbf{u}^0))$, $\mathbf{p}_j^0 = \left(A_j^\top\right)^{-1}(M(\mathbf{y}_d - \mathbf{y}_j^0))$, $j = 1, \ldots, N$.
2: Set $k = 0$ and define $I_0^+$ and $I_0^-$ using (30).
3: **while** $\phi(\mathbf{u}^k) >$ Tol **do**
4: Solve (29) calling Alg. 1 until convergence.
5: Set $\gamma = 1$
6: **while** $\phi(\mathbf{u}^k + \gamma \mathbf{du}^k) - \phi(\mathbf{u}^k) > -\sigma\phi(\mathbf{u}^k)$ **do**
7: $\gamma = \rho\gamma$.
8: **end while**
9: Update $\mathbf{u}^{k+1} = \mathbf{u}^k + \gamma \mathbf{du}^k$, $\mathbf{y}_j^{k+1} = \mathbf{y}_j^k + \gamma \mathbf{dy}_j^k$, $\mathbf{p}_j^{k+1} = \mathbf{p}_j^k + \gamma \mathbf{dp}_j^k$, $j = 1, \ldots, N$.
10: Update $I_k^+$ and $I_k^-$ using (30).
11: Set $k = k + 1$.
12: **end while**
13: **return** $\mathbf{u}^k$, $\mathbf{y}_j^k$ and $\mathbf{p}_j^k$, $j = 1, \ldots, N$.

### 4.1 Numerical Experiments

In this section we test the semismooth Newton algorithm for the solution of (26) and the collective multigrid algorithm to solve the related optimality system (29). We consider the random PDE-constraint (22) with the random diffusion coefficient (23) set on the L-squared

**Table 4** Number of semismooth Newton iterations (left), and average number of V-cycle (center) and preconditioned GMRES (right) iterations (in brackets)

| $\sigma^2$ | 0.1 | 0.5 | 1 | 1.5 |
|---|---|---|---|---|
| It. | 4 \| 22.5 \| 14 | 5 \| 22.6 \| 14.2 | 8 \| 23.0 \| 11.8 | 14.9 \| 22.9 \| 15.0 |
| $N_h = 705, \nu = 10^{-4}\ \beta = 10^{-2}, N = 125, N_L = 3,, L^2 = 0.5, b = 50, a = -50.$ | | | | |
| $N_h(N_L)$ | 161 (2) | 705 (3) | 2945 (4) | |
| It. | 5 \| 22.0 \| 15.2 | 5 \| 22.6 \| 14.2 | 5 \| 22.2 \| 14.0 | |
| $\nu = 10^{-4}, \beta = 10^{-2}, N = 125, \sigma^2 = 0.5, L^2 = 0.5, b = 50, a = -50.$ | | | | |
| $N$ | 8 | 27 | 64 | 125 |
| It. | 5 \| 21.0 \| 13.0 | 5 \| 21.6 \| 14.0 | 5 \| 22.0 \| 14.0 | 5 \| 22.6 \| 14.2 |
| $N_h = 705, \nu = 10^{-4}, \beta = 10^{-2}, \sigma^2 = 0.5, L^2 = 0.5, b = 50, a = -50.$ | | | | |
| $\beta$ | 0 | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ |
| It. | 4 \| 22.5 \| 14.8 | 4 \| 22.5 \| 14.5 | 5 \| 22.4 \| 14.8 | 5 \| 22.6 \| 14.2 |
| $N_h = 705, \nu = 10^{-4}, N = 125, \sigma^2 = 0.5, L^2 = 0.5, b = 50, a = -50.$ | | | | |

**Table 5** Number of semismooth Newton iterations, of V-cycle iterations and of preconditioned GMRES iterations (in brackets). In the second row, the semismooth Newton method starts from a warm-up initial guess obtained through continuation

| $\nu$ | $10^{-2}$ | $10^{-4}$ | $10^{-6}$ | $10^{-8}$ |
|---|---|---|---|---|
| It. | 2 \| 23.0 \| 14.5 | 5 \| 22.7 \| 14.2 | 17 \| 25.6 \| 15.0 | 50 \| 41.4 \| 17.2 |
| It. | 2 \| 23.0 \| 14.5 | 4 \| 22.7 \| 14.2 | 5 \| 22.25 \| 15.4 | 8 \| 58.8 \| 20.9 |
| $N_h = 705, N = 125, N_L = 3, \sigma^2 = 0.5, L^2 = 0.5, \beta = 10^{-2}, b = 50, a = -50.$ | | | | |

domain. The semismooth iteration is stopped when $\phi(\mathbf{u}^k) < 10^{-9}$. The inner linear solvers are stopped when the relative (unpreconditioned) residual is smaller than $10^{-11}$.

Table 4 reports the number of semismooth Newton iterations and in brackets the averaged number of iterations of the V-cycle algorithm used as a solver (left) or as preconditioner for GMRES (right). Table 4 confirms the effectiveness of the multigrid algorithm, which requires essentially the same computational effort as in the linear-quadratic case.

More challenging is the limit $\nu \to 0$ reported in Table 5. The performance of both the (globalized) semismooth Newton iteration and the inner multigrid solver deteriorates. The convergence of the outer nonlinear algorithm can be improved by performing a continuation method, namely we consider a sequence of $\nu = 10^{-j}$, $j = 2, \dots, 8$ and we start the $j$-th problem using as initial condition the optimal solution computed for $\nu = 10^{-j+1}$. Concerning the inner solver, the stand-alone multigrid algorithm struggles since for small values of $\nu$ the optimal control is of bang-bang type, that is satisfies $u = a$, $u = b$ or $u = 0$ for almost every point of the mesh (for $\nu = 10^{-8}$ only five nodes are nonactive at the optimum). The matrices $H^k$ are then close to zero, and the multigrid hierarchy struggles to capture changes at such small scale. Nevertheless, the multigrid algorithm remains a very efficient preconditioner for GMRES even in this challenging limit.

Figure 2 shows a sequence of optimal controls for different values of $\beta$ with and without box-constraints. The optimal control for $\beta = 0$ and without box-constraints corresponds to the minimizer of the linear-quadratic OCP (5). We observe that $L^1$ penalization indeed

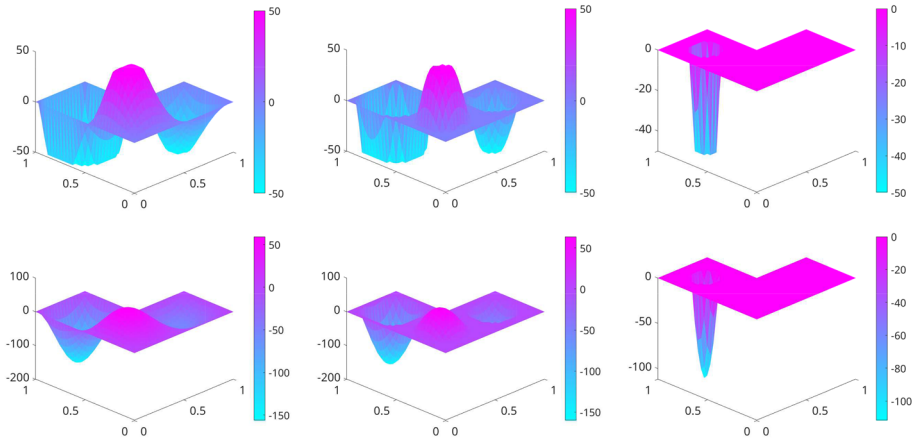

**Fig. 2** From left to right: optimal control computed for $\beta \in \left\{0, 5 \cdot 10^{-3}, 5 \cdot 10^{-2}\right\}$ with (top row) and without (bottom row) box constraints: $a = -50$, $b = 50$

induces sparsity, since the optimal controls are more and more localized as $\beta$ increases. Numerically we have verified that for sufficiently large $\beta$, the optimal control is identically equal to zero, a property shown in [36].

## 5 A Risk-Averse Optimal Control Problem Under Uncertainty

In this section we consider an instance of risk-averse OCPUU. This class of problems has recently drawn lot of attention since in engineering applications it is important to compute a control that minimizes the quantity of interest even in rare, but often troublesome, scenarios [2, 6, 43, 44]. As a risk-measure [45], we use the Conditional Value-At-Risk (CVaR) of confidence level $\lambda \in (0, 1)$,

$$\mathrm{CVaR}_\lambda(X) := \mathbb{E}\left[X | X \geq \mathrm{VaR}_\lambda(X)\right], \quad \forall X \in L^1(\Omega; \mathbb{R}),$$

that is, the expected value of a quantity of interest $X$ given that the latter is greater than or equal to its $\lambda$-quantile, here denoted by $\mathrm{VaR}_\lambda(X)$. Rockafellar and Uryasev [46] proved that $\mathrm{CVaR}_\lambda(X)$ admits the equivalent formulation

$$\mathrm{CVaR}_\lambda(X) = \inf_{t \in \mathbb{R}} \left\{ t + \frac{1}{1-\lambda} \mathbb{E}\left[(X-t)^+\right] \right\},$$

where $(\cdot)^+ := \max(0, \cdot)$, if the distribution of $X$ does not have an atom at $\mathrm{VaR}_\lambda(X)$. In order to use tools from smooth optimization, we rely on a smoothing approach proposed in [2], which consists in replacing $(\cdot)^+$ with a smooth function $g_\varepsilon$, $\varepsilon \in \mathbb{R}^+$, such that $g_\varepsilon \to (\cdot)^+$ in some functional norm as $\varepsilon \to 0$. Specifically, we choose the $C^2$-differentiable approximation

$$g_\varepsilon(x) = \begin{cases} 0 & \text{if } x \leq -\frac{\varepsilon}{2}, \\ \frac{(x-\frac{3}{2})^3}{\varepsilon^2} - \frac{(x-\frac{\varepsilon}{2})^4}{2\varepsilon^3} & \text{if } x \in (-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}), \\ x & \text{if } x \geq \frac{\varepsilon}{2}. \end{cases}$$

Then, the smoothed risk-averse OCPUU is

$$\begin{aligned} &\min_{u\in L^2(\mathcal{D}),t\in\mathbb{R}} \; t+\frac{1}{1-\lambda}\mathbb{E}\left[g_\varepsilon\left(\frac{1}{2}\|y_\omega-y_d\|^2_{L^2(\mathcal{D})}-t\right)\right]+\frac{\nu}{2}\|u\|^2_{L^2(\mathcal{D})},\\ &\quad\text{subject to}\\ &a_\omega(y_\omega,v)=(u+f,v)\quad \forall v\in V,\ \mathbb{P}\text{-a.e. } \omega\in\Omega, \end{aligned} \tag{31}$$

where $\nu\in\mathbb{R}^+$ and $\lambda\in[0,1)$. The well-posedness of (31), the differentiability of its objective functional, as well as bounds for the error introduced by replacing $(\cdot)^+$ with $g_\varepsilon(\cdot)$, have been analyzed in [2]. Further, defining $Q_\omega=\frac{1}{2}\|y_\omega-y_d\|^2_{L^2(\mathcal{D})}-t$, the optimality conditions form the nonlinear system,

$$\begin{aligned} &a_\omega(v,p_\omega)-\tfrac{g'_\varepsilon(Q_\omega)}{1-\lambda}(y_d-y_\omega,v)=0, && \forall v\in V,\ \mathbb{P}\text{-a.e. } \omega\in\Omega,\\ &(\nu\, u-\mathbb{E}[p_\omega],v)=0, && \forall v\in L^2(\mathcal{D}),\\ &a_\omega(y_\omega,v)-(u+f,v)=0, && \forall v\in V,\quad \mathbb{P}\text{-a.e. } \omega\in\Omega,\\ &1-\tfrac{1}{1-\lambda}\mathbb{E}\left[g'_\varepsilon(Q_\omega)\right)]=0. \end{aligned} \tag{32}$$

Approximating $V$ and $\mathbb{E}$ with $V_h$ and $\widehat{\mathbb{E}}$, and letting $\widetilde{\mathbf{x}}=(\mathbf{y},\mathbf{u},\mathbf{p},t)$, the finite-dimensional discretization of (32) correponds to the nonlinear system $\widetilde{\mathbf{F}}(\widetilde{\mathbf{x}})=\mathbf{0}$, where $\widetilde{\mathbf{F}}:\mathbb{R}^{(2N+1)N_h+1}\to\mathbb{R}^{(2N+1)N_h+1}$,

$$\widetilde{\mathbf{F}}(\widetilde{\mathbf{x}})=\begin{pmatrix}\widetilde{\mathbf{F}}_1(\widetilde{\mathbf{x}})\\ \widetilde{\mathbf{F}}_2(\widetilde{\mathbf{x}})\\ \widetilde{\mathbf{F}}_3(\widetilde{\mathbf{x}})\\ \widetilde{F}_4(\widetilde{\mathbf{x}})\end{pmatrix}=\begin{pmatrix}\widetilde{M}(\mathbf{y}-I\mathbf{y}_d)+A^\top\mathbf{p}\\ \nu M\mathbf{u}-M\widehat{\mathbb{E}}[\mathbf{p}]\\ A\mathbf{y}-M(I\mathbf{u}+\mathbf{f})\\ 1-\frac{1}{1-\lambda}\widehat{\mathbb{E}}\left[g'_\varepsilon(Q_\omega)\right]\end{pmatrix}, \tag{33}$$

with $A=\mathrm{diag}(A_1,\dots,A_N)$, $I=[I_{Nh},\dots,I_{Nh}]\in\mathbb{R}^{N_h\times N_hN}$, $I_h$ being the identity matrix, $\mathbf{y}_d$ is the discretization of $y_d$, and

$$\widetilde{M}=\mathrm{diag}\left(\frac{g'_\varepsilon(Q_{\omega_1})}{1-\lambda}M,\dots,\frac{g'_\varepsilon(Q_{\omega_N})}{1-\lambda}M\right),\ \text{with } Q_{\omega_j}:=\frac{1}{2}(\mathbf{y}_j-\mathbf{y}_d)^\top M(\mathbf{y}_j-\mathbf{y}_d)-t,$$

for $j=1,\dots,N$.

A possible approach to solve (33) is to use a Newton method, which given $\mathbf{x}^k=(\mathbf{y}^k,\mathbf{u}^k,\mathbf{p}^k,t^k)$ computes the corrections $\widetilde{\mathbf{dx}}^k=(\mathbf{dy}^k,\mathbf{du}^k,\mathbf{dp}^k,dt^k)$ solution of $\widetilde{\mathbf{J}}^k\widetilde{\mathbf{dx}}^k=-\widetilde{\mathbf{F}}(\widetilde{\mathbf{x}}^k)$, where

$$\widetilde{\mathbf{J}}^k:=\begin{pmatrix} C_1(\mathbf{y}_1^k,t^k) & & & & A_1^\top & & & -\mathbf{v}_1^k\\ & \ddots & & & & \ddots & & \vdots\\ & & C_N(\mathbf{y}_N^k,t^k) & & & & A_N^\top & -\mathbf{v}_N^k\\ & & & \nu M & -\zeta_1 M & \dots & -\zeta_N M & \\ A_1 & & & -M & & & & \\ & \ddots & & \vdots & & & & \\ & & A_N & -M & & & & \\ -\zeta_1\left(\mathbf{v}_1^k\right)^\top & \ddots & -\zeta_N\left(\mathbf{v}_N^k\right)^\top & & & & & \frac{\widehat{\mathbb{E}}[g''_\varepsilon(Q_\omega^k)]}{1-\lambda} \end{pmatrix},$$

with

$$Q_{\omega_i}^k:=\frac{1}{2}(\mathbf{y}_i^k-\mathbf{y}_d)^\top M(\mathbf{y}_i^k-\mathbf{y}_d)-t^k,$$

$$C_i(\mathbf{y}_i^k, t^k) := \frac{1}{1-\lambda}\left(g_\varepsilon'(Q_{\omega_i}^k)M + g_\varepsilon''(Q_{\omega_i}^k)M(\mathbf{y}_i^k - \mathbf{y}_d)(\mathbf{y}_i^k - \mathbf{y}_d)^\top M\right), \tag{34}$$
$$\mathbf{v}_i^k := \frac{1}{1-\lambda} g_\varepsilon''(Q_{\omega_i}^k)M(\mathbf{y}_i^k - \mathbf{y}_d),$$

for $i = 1, \ldots, N$. Unfortunately, $\widetilde{\mathbf{J}}^k$ can be singular away from the optimum, in particular whenever $\widehat{\mathbb{E}}\left[g_\varepsilon''(Q_\omega^k)\right] = 0$ which implies

$$g_\varepsilon''\left(\frac{1}{2}(\mathbf{y}_j^k - \mathbf{y}_d)^\top M(\mathbf{y}_j^k - \mathbf{y}_d) - t^k\right) = 0, \ \forall j = 1, \ldots, N, \tag{35}$$

which is not unlikely for small $\varepsilon$ since $\mathrm{supp}(g_\varepsilon'') = (-\frac{\varepsilon}{2}, \frac{\varepsilon}{2})$. Splitting strategies have been proposed (e.g. [47] in a reduced approach), in which whenever (35) is satisfied, an intermediate value of $t$ is computed by solving $\widetilde{F}_4(t; \mathbf{y}, \mathbf{u}, \mathbf{p}) = 0$ so to violate (35). In the next section, we discuss a similar splitting approach. To speed up the convergence of the outer nonlinear algorithm, we use a preconditioned Newton method based on nonlinear elimination [48]. At each iteration we will need to invert saddle-point matrices like (2), possibly several times. To do so, we rely on the collective multigrid algorithm.

### 5.1 Nonlinear Preconditioned Newton Method

Nonlinear elimination is a nonlinear preconditioning technique based on the identification of variables and equations of $\mathbf{F}$ (e.g. strong nonlinearities) that slow down the convergence of Newton method. These components are then eliminated through the solution of a local nonlinear problem at every step of an outer Newton. This elimination step provides a better initial guess for the outer iteration, so that a faster convergence is achieved [48, 49].

In light of the possible singularity of $\widetilde{\mathbf{J}}$, we split the discretized variables $\widetilde{\mathbf{x}}$ into $\widetilde{\mathbf{x}} = (\mathbf{x}, t)$, and we aim to eliminate the variables $\mathbf{x}$ to obtain a scalar nonlinear equation only for $t$. To do so, we partition (32) as

$$\widetilde{\mathbf{F}}\begin{pmatrix}\mathbf{x}\\ t\end{pmatrix} = \begin{pmatrix}\mathbf{F}_1(\mathbf{x}, t)\\ F_2(\mathbf{x}, t)\end{pmatrix} = \begin{pmatrix}\mathbf{0}\\ 0\end{pmatrix}, \tag{36}$$

where $\mathbf{F}_1 = (\widetilde{\mathbf{F}}_1(\mathbf{x}, t), \widetilde{\mathbf{F}}_2(\mathbf{x}, t), \widetilde{\mathbf{F}}_3(\mathbf{x}, t))$ and $F_2(\mathbf{x}, t) = \widetilde{F}_4(\mathbf{x}, t)$. Similarly, $\widetilde{\mathbf{J}}$ is partitioned into

$$\widetilde{\mathbf{J}} = \begin{pmatrix}\mathbf{J}_{1,1} & \mathbf{J}_{1,2}\\ \mathbf{J}_{2,1} & J_{2,2}\end{pmatrix}$$

whose blocks have dimensions $\mathbf{J}_{1,1} \in \mathbb{R}^{(2N+1)N_h \times (2N+1)N_h}$, $\mathbf{J}_{1,2} \in \mathbb{R}^{(2N+1)N_h \times 1}$, $\mathbf{J}_{2,1} \in \mathbb{R}^{1 \times (2N+1)N_h}$, and $J_{2,2} \in \mathbb{R}$. Notice that $\mathbf{J}_{1,1}$ is always nonsingular, while $\mathbf{J}_{2,1}$, $\mathbf{J}_{1,2}$ and $J_{2,2}$ are identically zero if (35) is verified.

Thus $\mathbf{F}_1$ allows us to define an implicit map $h : \mathbb{R} \to \mathbb{R}^{(2N+1)N_h}$, such that $\mathbf{F}_1(h(t), t) = 0$, so that the first set of nonlinear equations in (36) are satisfied. We are then left to solve the nonlinear scalar equation

$$F(t) = 0, \quad \text{where} \quad F(t) := F_2(h(t), t). \tag{37}$$

To do so using the Newton method, we need the derivative of $F(t)$ evaluated at $t = t^k$ which, using implicit differentiation, can be computed as

$$F'(t^k) = J_{2,2}(h(t^k), t^k) - \mathbf{J}_{2,1}(h(t^k), t^k)\left(\mathbf{J}_{1,1}(h(t^k), t^k)\right)^{-1}\mathbf{J}_{1,2}(h(t^k), t^k).$$

The nonlinear preconditioned Newton method is described in Algorithm 3, and consists in solving (37) with Newton method. However, to overcome the possible singularity of $J^k_{2,2}$, $\mathbf{J}^k_{1,2}$ and $\mathbf{J}^k_{2,1}$, we check at each iteration $k$ if (35) is satisfied, and in the affirmative case we update $\mathbf{x}^k$ by solving $\mathbf{F}_1(\mathbf{x}^{k+1}, t^k) = 0$ using Newton method, and update $t^k$ by solving $F_2(\mathbf{x}^k, t^{k+1}) = 0$. Notice further, that each iteration of the backtracking line-search requires to solve $F_1(h(t), t) = 0$ using Newton method, thus additional linear systems with matrix $\mathbf{J}_{1,1}$ must be solved.

We report that we also tried to eliminate $t$ by computing the map $l$ such that $F_2(\mathbf{x}, l(\mathbf{x})) = 0$, while iterating on the variable $\mathbf{x}$. This has the advantage that $l$ can be evaluted very cheaply, being a scalar equation. However, we needed many more iterations both of the outer Newton method, and consequently of the inner linear solver. Thus, according to our experience, this second approach was less efficient and appealing.

**Algorithm 3** Nonlinear preconditioned Newton method to solve $\widetilde{\mathbf{F}}(\widetilde{\mathbf{x}}) = 0$.

**Require:** $t^0$, Tol $\in \mathbb{R}^+$, $\sigma, \rho \in (0, 1)$.
Compute $\mathbf{x}^0 = h(t^0)$ solving $\mathbf{F}_1(\mathbf{x}^0; t^0) = 0$ using the Newton method.
Set $k = 0$.
**while** $|F(t^k)| >$ Tol **do**
  **if** (35) is satisfied **then**
    Compute $\mathbf{x}^{k+1}$ and $t^{k+1}$ solving $\mathbf{F}_1(\mathbf{x}^{k+1}; t^k) = 0$ and $F_2(\mathbf{x}^{k+1}; t^{k+1}) = 0$.
  **else**
    Compute Newton's direction $d = -(F'(t^k))^{-1} F(t^k)$.
      Set $\gamma = 1$ and compute $\mathbf{x} = h(t^k + \gamma d)$ solving $\mathbf{F}_1(\mathbf{x}; t^k + d) = 0$.
    **while** $|F(t^k + \gamma d)| - |F(t^k)| > -\sigma |F(t^k)|$ **do**
      Set $\gamma = \rho\gamma$.
      Compute $\mathbf{x} = h(t^k + \gamma d)$ solving $\mathbf{F}_1(\mathbf{x}; t^k + \gamma d) = 0$.
    **end while**
    Set $t^{k+1} = t^k + \gamma d$, $\mathbf{x}^{k+1} = \mathbf{x}$, $k = k + 1$.
  **end if**
**end while**
**return** $t^{k+1}$ and $\mathbf{x}^{k+1}$.

### 5.2 Numerical Experiments

In this section we report numerical tests to asses the performance of the preconditioned Newton algorithm to solve (37), and of the collective multigrid algorithm to invert the matrix $\mathbf{J}_{1,1}$. We consider the random PDE-constraint (22) with the random diffusion coefficient (23). Table 6 reports the number of outer and inner Newton iterations, and the average number of V-cycle iterations and of preconditioned GMRES iterations to solve the linear systems at each (inner/outer) Newton iterations. The outer Newton iteration is stopped when $|F(t^k)| \leq 10^{-6}$, the inner Newton method to compute $h(\cdot)$ is stopped when $\max\left(\|\mathbf{F}_{1,1}(\mathbf{x}^k; t)\|_2 / \|\mathbf{F}_{1,1}(\mathbf{x}^0; t)\|_2, \|\mathbf{F}_{1,1}(\mathbf{x}^k; t)\|_2\right) \leq 10^{-8}$, and the linear solvers are stopped when the relative (unpreconditioned) residual is smaller than $10^{-9}$.

In Table 6, the number of outer Newton iterations is stable, while the number of inner Newton iterations varies between five and fifteen iterations per outer iteration. This is essentially due to how difficult it is to compute the nonlinear map $h(t)$ by solving $\mathbf{F}_1(\mathbf{x}; t) = 0$ in

**Table 6** For each numerical experiment, we report from the left to the right: the number of outer preconditioned Newton iterations, the total number of inner Newton iterations, the averaged number of V-cycle iterations and the averaged number of preconditioned GMRES iterations

| | | | | |
|---|---|---|---|---|
| $N_h(N_L)$ | 161 (2) | 705 (3) | 2945 (4) | |
| It. | 5 \| 62 \| 23.0 \| 13.9 | 6 \| 79 \| 28.0 \| 15.5 | 6 \| 79 \| 26.2 \| 14.8 | |
| $\nu = 10^{-4}$, $N = 500$, $\lambda = 0.9$, $\varepsilon = 10^{-2}$, $\sigma^2 = 1$, $L^2 = 0.1$. | | | | |
| $N$ | 500 | 1000 | 2000 | |
| It. | 6 \| 63 \| 55.4 \| 17.5 | 5 \| 66 \| 24.4 \| 14.0 | 4 \| 51 \| 24.4 \| 14.0 | |
| $N_h = 705$, $\nu = 10^{-4}$, $\lambda = 0.95$, $\varepsilon = 10^{-2}$, $\sigma^2 = 1$, $L^2 = 0.1$. | | | | |
| $\lambda$ | 0 | 0.5 | 0.95 | 0.99 |
| It. | 0 \| 1 \| 21.0 \| 14.0 | 5 \| 21 \| 19.4 \| 13.6 | 5 \| 64 \| 23.2 \| 13.8 | 8 \| 129 \| 33.4 \| 17.5 |
| $N_h = 705$, $N = 2000$, $\nu = 10^{-4}$, $\varepsilon = 10^{-2}$, $\sigma^2 = 1$, $L^2 = 0.1$. | | | | |
| $\varepsilon$ | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ |
| It. | 7 \| 67 \| 22.5 \| 17.0 | 3 \| 42 \| 29.1 \| 14.8 | 2 \| 20 \| $>$ 80 \| 27.9 | 1 \| 15 \| 58.0 \| 55.6 |
| $N_h = 705$, $N = 1000$, $\nu = 10^{-4}$, $\beta = 0.95$, $\sigma^2 = 1$, $L^2 = 0.1$. | | | | |

line (5), (8) and (11) of Algorithm 3. The average number of inner linear solver iterations is quite stable across all experiments. The most challenging case is the limit $\varepsilon \to 0$ in which we used the solution to the optimization problem as a warmed-up initial guess for the next smaller value of $\varepsilon$. Further, we emphasize that the top left blocks of $\mathbf{J}_{1,1}$ involve the matrices $C_i(\mathbf{y}_i^k, t^k)$ (see (34)) which contain a dense low-rank term if $g_\varepsilon''(Q_{\omega_i}^k) \neq 0$. As $\varepsilon \to 0$, $g_\varepsilon''(\cdot)$ tends to a Dirac delta, so the dense term become dominant. Multigrid methods based on pointwise relaxations are expected to be not very efficient for these matrices which may not be diagonally dominant. The standard V-cycle algorithm indeed suffers, however the Krylov acceleration performs better as it handles these low-rank perturbation with smaller effort. For $\varepsilon = 10^{-4}$, we sometimes noticed that the GMRES residual stagnates after 20/30 iterations around $10^{-7}/10^{-8}$, due to a loss of orthogonality in the Krylov subspace, and thus resulting in higher number of iterations. We allowed a maximum number of 80 iterations per linear system.

Figure 3 compares the two optimal controls obtained minimizing either $\mathbb{E}[Q(y_\omega)]$ or $\mathrm{CVaR}_{0.99}[Q(y_\omega)]$, and the cumulative distribution functions of $Q(y_{\omega_j})$ computed on 8000 out-of-sample realizations. The risk-averse control indeed minimizes the *risk* of having large values of $Q(y_\omega)$. The CVaR of level $\lambda = 0.99$ is respectively $\mathrm{CVaR}_{0.99}(Q(y_\omega)) = 2.79$ for the risk-neutral control and $\mathrm{CVaR}_{0.99}(Q(y_\omega)) = 0.90$ for the risk-averse control.

## 6 Conclusion

We have presented a multigrid method to solve the large saddle point linear systems that typically arise in full-space approaches to solve OCPUU. We further derived a detailed convergence analysis that fully characterizes the spectrum of the two-level iteration matrix. The algorithm has been tested as an iterative solver and as a preconditioner on three test cases: a linear-quadratic OCPUU, a nonsmooth OCPUU, and a risk-averse nonlinear OCPUU.

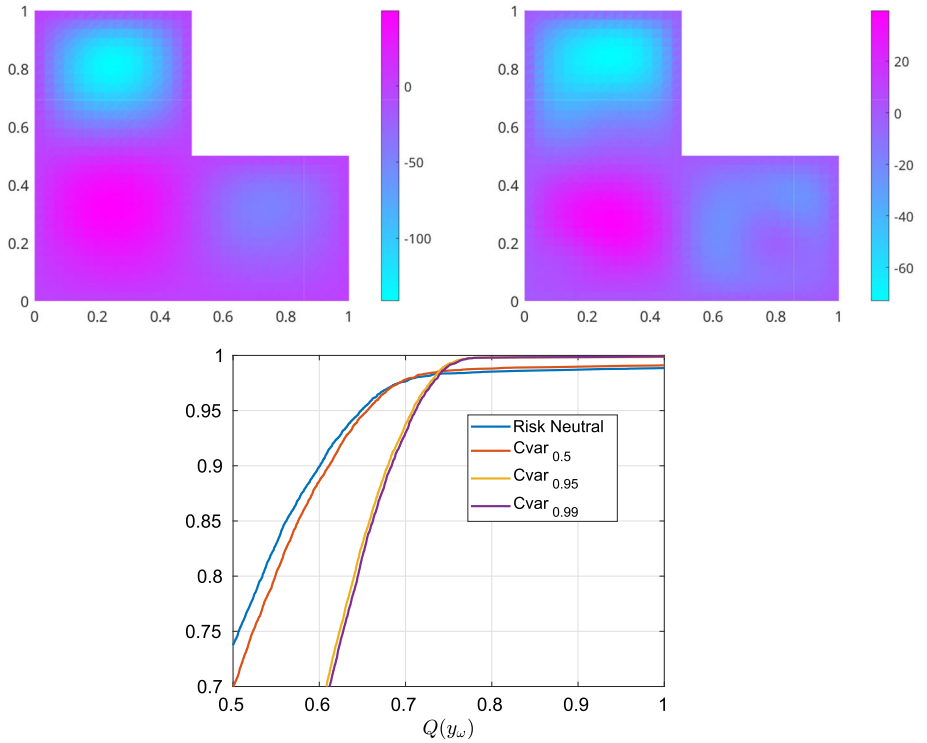


**Fig. 3** Solution of the linear-quadratic OCP (top-left), solution of the smoothed risk-averse OCP with $\lambda = 0.99$ (top-right), and cumulative distribution function of the quantity of interest for the controls computed with $\lambda \in \{0, 0.5, 0.95, 0.99\}$

Overall, the multigrid method shows very good performances and robustness with respect to the several parameters of the problems considered.

**Acknowledgements** The authors wish to thank an anonymous reviewer for the recommendation to develop a convergence analysis of the multigrid algorithm. G. C. and T. V. are members of GNCS (Gruppo Nazionale per il Calcolo Scientifico) of INdAM. The present research is part of the activities of "Dipartimento di Eccellenza 2023-2027".

**Author Contributions** All authors contributed equally to the manuscript.

**Funding** Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

**Data Availability** The codes used in this study are available from the corresponding author.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical Approval** Not applicable.

## References

1. Kouri, D.P., Shapiro, A.: Optimization of PDEs with uncertain inputs. In: Frontiers in PDE-constrained optimization, pp. 41–81. Springer, New York (2018)
2. Kouri, D.P., Surowiec, T.M.: Risk-averse PDE-constrained optimization using the conditional value-at-risk. SIAM J. Optim. **26**(1), 365–396 (2016)
3. Martínez-Frutos, J., Esparza, F.: Optimal control of PDEs under uncertainty: an introduction with application to optimal shape design of structures. Springer, Heidelberg (2018)
4. Guth, P.A., Kaarnioja, V., Kuo, F., Schillings, C., Sloan, I.H.: A Quasi-Monte Carlo method for optimal control under uncertainty. SIAM/ASA J. Uncertain. Quantif. **9**(2), 354–383 (2021)
5. Geiersbach, C., Wollner, W.: A stochastic gradient method with mesh refinement for PDE-constrained optimization under uncertainty. SIAM J. Sci. Comput. **42**(5), 2750–2772 (2020)
6. Antil, H., Dolgov, S., Onwunta, A.: Ttrisk: Tensor train decomposition algorithm for risk averse optimization. Numer. Linear Algebra Appl. **30**(3), 2481 (2023)
7. Nobile, F., Vanzan, T.: A combination technique for optimal control problems constrained by random PDEs. SIAM/ASA J. Uncertain. Quantif. **12**(2), 693–721 (2024)
8. Eigel, M., Neumann, J., Schneider, R., Wolf, S.: Risk averse stochastic structural topology optimization. Comput. Methods Appl. Mech. Eng. **334**, 470–482 (2018)
9. Asadpoure, A., Tootkaboni, M., Guest, J.K.: Robust topology optimization of structures with uncertainties in stiffness - application to truss structures. Comput. Struct. 89(11), 1131–1141 (2011). Computational Fluid and Solid Mechanics 2011
10. Kouri, D.P., Heinkenschloss, M., Ridzal, D., van Bloemen Waanders, B.G.: A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty. SIAM J. Sci. Comput. **35**(4), 1847–1879 (2013)
11. Kouri, D.P., Ridzal, D.: Inexact trust-region methods for PDE-constrained optimization, pp. 83–121. Springer, New York (2018)
12. Nobile, F., Vanzan, T.: Preconditioners for robust optimal control problems under uncertainty. Numer. Linear Algebra Appl. **30**(2), 2472 (2023)
13. Borzì, A., Kunisch, K.: A multigrid scheme for elliptic constrained optimal control problems. Comput. Optim. Appl. **31**(3), 309–333 (2005)
14. Borzì, A., Schulz, V.: Multigrid methods for PDE optimization. SIAM Rev. **51**(2), 361–395 (2009)
15. Takacs, S., Zulehner, W.: Convergence analysis of multigrid methods with collective point smoothers for optimal control problems. Comput. Vis. Sci. **14**(3), 131–141 (2011)
16. Borzì, A., von Winckel, G.: Multigrid methods and sparse-grid collocation techniques for parabolic optimal control problems with random coefficients. SIAM J. Sci. Comput. **31**(3), 2172–2192 (2009)
17. Borzì, A.: Multigrid and sparse-grid schemes for elliptic control problems with random coefficients. Comput. Vis. Sci. **13**(4), 153–160 (2010)
18. Rosseel, E., Wells, G.N.: Optimal control with stochastic PDE constraints and uncertain controls. Comput. Methods Appl. Mech. Eng. **213**, 152–167 (2012)
19. Kouri, D.P.: A multilevel stochastic collocation algorithm for optimization of PDEs with uncertain coefficients. SIAM/ASA J. Uncertain. Quantif. **2**(1), 55–81 (2014)
20. Lord, G.J., Powell, C.E., Shardlow, T.: An introduction to computational stochastic PDEs. Cambridge texts in applied mathematics. Cambridge University Press, Cambridge (2014)
21. Charrier, J., Scheichl, R., Teckentrup, A.L.: Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods. SIAM J. Numer. Anal. **51**(1), 322–352 (2013)
22. Cohn, D.L.: Measure theory, 2nd edn. Birkhäuser, New York (2013)
23. Lions, J.L.: Optimal control of systems governed by partial differential equations. Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen. Springer, Heidelberg (1971)
24. Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: Optimization with PDE constraints, vol. 23. Springer, Heidelberg (2008)

25. Tröltzsch, F.: Optimal control of partial differential equations: theory, methods, and applications. Graduate studies in mathematics. American Mathematical Society, New York (2010)
26. Borzì, A.: Multigrid methods for optimality systems, Habilitation thesis, University of Graz, (2003)
27. Van Barel, A., Vandewalle, S.: Robust optimization of PDEs with random coefficients using a multilevel Monte Carlo method. SIAM/ASA J. Uncertain. Quantif. **7**(1), 174–202 (2019)
28. Noschese, S., Pasquini, L., Reichel, L.: Tridiagonal toeplitz matrices: properties and novel applications. Numer. Linear Algebra Appl. **20**(2), 302–326 (2013)
29. Ciaramella, G., Gander, M.J.: Iterative methods and preconditioners for systems of linear equations. SIAM, Philadelphia (2022)
30. Ciaramella, G., Vanzan, T.: Substructured two-grid and multi-grid domain decomposition methods. Numer. Algorithms **91**(1), 413–448 (2022)
31. Ciaramella, G., Vanzan, T.: Spectral coarse spaces for the substructured parallel schwarz method. J. Sci. Comput. **91**(3), 69 (2022)
32. Charrier, J.: Strong and weak error estimates for elliptic partial differential equations with random coefficients. SIAM J. Numer. Anal. **50**(1), 216–246 (2012)
33. Babuška, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. SIAM Rev. **52**(2), 317–355 (2010)
34. Rees, T., Dollar, H.S., Wathen, A.: Optimal solvers for PDE-constrained optimization. SIAM J. Sci. Comput. **32**(1), 271–298 (2010)
35. Pearson, J.W., Wathen, A.: A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. Numer. Linear Algebra Appl. **19**(5), 816–829 (2012)
36. Stadler, G.: Elliptic optimal control problems with $l^1$ -control cost and applications for the placement of control devices. Comput. Optim. Appl. **44**(2), 159–181 (2009)
37. Casas, E.: A review on sparse solutions in optimal control of partial differential equations. SeMA J. **74**(3), 319–344 (2017)
38. Li, C., Stadler, G.: Sparse solutions in optimal control of PDEs with uncertain parameters: the linear case. SIAM J. Control. Optim. **57**(1), 633–658 (2019)
39. Ekeland, I., Temam, R.: Convex analysis and variational problems. SIAM, Philadelphia (1999)
40. Ulbrich, M.: Semismooth Newton methods for variational inequalities and constrained optimization problems in function spaces. SIAM, Philadelphia (2011)
41. Stadler, G.: Errata: Elliptic optimal control problems with L1 -control cost and applications for the placement of control devices, available at math.nyu.edu/ stadler/papers/correction.pdf
42. Martínez, J., Qi, L.: Inexact newton methods for solving nonsmooth equations. J. Comput. Appl. Math. **60**(1–2), 127–145 (1995)
43. Kouri, D.P., Surowiec, T.M.: Existence and optimality conditions for risk-averse PDE-constrained optimization. SIAM/ASA J. Uncertain. Quantif. **6**(2), 787–815 (2018)
44. Kouri, D.P., Surowiec, T.M.: A primal-dual algorithm for risk minimization. Math. Program. **193**(1), 337–363 (2022)
45. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on stochastic programming: modeling and theory, 2nd edn. SIAM, Philadelphia (2014)
46. Rockafellar, R.T., Uryasev, S., et al.: Optimization of conditional value-at-risk. J. Risk **2**, 21–42 (2000)
47. Markowski, M.: Efficient solution of smoothed risk-adverse PDE-constrained optimization problems. PhD thesis, Rice University (2022)
48. Lanzkron, P.J., Rose, D.J., Wilkes, J.T.: An analysis of approximate nonlinear elimination. SIAM J. Sci. Comput. **17**(2), 538–559 (1996)
49. Yang, H., Hwang, F.-N., Cai, X.-C.: Nonlinear preconditioning techniques for full-space lagrange-newton solution of PDE-constrained optimization problems. SIAM J. Sci. Comput. **38**(5), 2756–2778 (2016)