A VARIANCE REDUCTION METHOD FOR PARAMETRIZED STOCHASTIC DIFFERENTIAL EQUATIONS USING THE REDUCED BASIS PARADIGM*

SÉBASTIEN BOYAVAL† AND TONY LELIÈVRE‡

Abstract. In this work, we develop a reduced-basis approach for the efficient computation of parametrized expected values, for *a large number* of parameter values, using the control variate method to reduce the variance. Two algorithms are proposed to compute *online*, through a cheap reduced-basis approximation, the control variates for the computation of a large number of expectations of a functional of a parametrized Itô stochastic process (solution to a parametrized stochastic differential equation). For each algorithm, a reduced basis of control variates is pre-computed *offline*, following a so-called *greedy* procedure, which minimizes the variance among a trial sample of the output parametrized expectations. Numerical results in situations relevant to practical applications (calibration of volatility in option pricing, and parameter-driven evolution of a vector field following a Langevin equation from kinetic theory) illustrate the efficiency of the method.

Key words. Variance reduction, stochastic differential equations, reduced-basis methods.

AMS subject classifications. 60H10, 65C05.

1. Introduction

This article develops a general variance reduction method for the **many-query** context where a large number of **Monte-Carlo** estimations of the expectation $\mathbb{E}\left(Z^\lambda\right)$ of a functional

$$Z^\lambda = g^\lambda(X_T^\lambda) - \int_0^T f^\lambda(s, X_s^\lambda)\,ds \tag{1.1}$$

of the solutions $\left(X_t^\lambda, t \in [0,T]\right)$ to the **stochastic differential equations** (SDEs):

$$X_t^\lambda = x + \int_0^t b^\lambda(s, X_s^\lambda)\,ds + \int_0^t \sigma^\lambda(s, X_s^\lambda)\,dB_s \tag{1.2}$$

parametrized by $\lambda \in \Lambda$ have to be computed for many values of the parameter λ.

Such many-query contexts are encountered in finance for instance, where pricing options often necessitate computing the price $\mathbb{E}\left(Z^\lambda\right)$ of an option with spot price X_t^λ at time t in order to *calibrate* the local volatility σ^λ as a function of a (multi-dimensional) parameter λ (that is minimize over λ, after many iterations of some optimization algorithm, the difference between observed statistical data with the model prediction). Another context for application is molecular simulation, for instance micro-macro models in rheology, where the mechanical properties of a flowing viscoelastic fluid are determined from the coupled evolution of a non-Newtonian stress tensor field $\mathbb{E}\left(Z^\lambda\right)$ due to the presence of many polymers with configuration X_t^λ in

the fluid with instantaneous velocity gradient field λ. Typically, segregated numerical schemes are used: compute X_t^λ for a fixed field λ, and then compute λ for a fixed field $\mathbb{E}\left(Z^\lambda\right)$. Such tasks are known to be computationally demanding and the use of different **variance reduction** techniques to alleviate the cost of Monte-Carlo computations in those fields is very common (see [2, 19, 22, 3] for instance).

In the following, we focus on one particular variance reduction strategy termed the **control variate** method [10, 21, 20]. More precisely, we propose new approaches to a practical use of control variates in the context of the computation of $\mathbb{E}\left(Z^\lambda\right)$ for a large number of parameter values. In these approaches, the control variates are computed through a **reduced-basis** method which is related to the reduced-basis method [17, 18, 23, 4, 5] previously developed to efficiently solve parametrized partial differential equations (PDEs). Following the reduced-basis paradigm, a low-dimensional vector basis is first built *offline* to span a good linear approximation space for a large trial sample of the λ-parametrized control variates, and then used *online* to compute control variates at any parameter value. The offline computations are typically expensive, but are only done once. Consequently, it is expected that the online computations (namely, approximations of $\mathbb{E}\left(Z^\lambda\right)$ for many values of λ) are very cheap, using the low-dimensional vector basis built offline for *efficiently* computing control variates online. Of course, such reduced-basis approaches can only be *efficient* insofar as:

1. online computations (of one output $\mathbb{E}\left(Z^\lambda\right)$ for one parameter value λ) are significantly cheaper using the reduced-basis approach than without, and

2. the amount of outputs $\mathbb{E}\left(Z^\lambda\right)$ to be computed online (for many different parameter values λ) is sufficient to compensate for the (expensive) offline computations (needed to build the reduced basis).

In this work, we will numerically study how the variance is reduced in two examples using control variates built with two different approaches.

The usual reduced-basis approach for parametrized PDEs also traditionally focuses on the certification of the reduction (in the parametrized solution manifold) by estimating *a posteriori* the error between approximations obtained before/after reduction for some *output* which is a functional of the PDE solution. Our reduced-basis approach for the parametrized control variate method can also be cast into a goal-oriented framework similar to the traditional reduced basis method. One can take the expectation $\mathbb{E}\left(Z^\lambda\right)$ as the reduced-basis output, while the empirically estimated variance $\mathrm{Var}_\mathrm{M}\left(Z^\lambda\right)$ serves as a computable (statistical) error indicator for the Monte-Carlo approximations $\mathrm{E}_\mathrm{M}\left(Z^\lambda\right)$ of $\mathbb{E}\left(Z^\lambda\right)$ *in the limit of large M* through the Central Limit Theorem (see error bound (2.4) in section 2.1).

In section 2, the variance reduction issue and the control variate method are introduced, as well as the principles of our reduced-basis approaches for the computation of parametrized control variates. Section 3 exposes details about the algorithms which are numerically applied to test problems in section 4.

The numerical simulations show good performance of the method for the two test problems corresponding to the applications mentionned above: a scalar SDE with (multi-dimensional) parametrized diffusion (corresponding to the calibration of a local volatility in option pricing), and a vector SDE with (multi-dimensional) parametrized drift (for the parameter-driven evolution of a vector field following a Langevin equation from kinetic theory). Using the control variate method with a 20-dimensional reduced basis of (precomputed) control variates, the variance is approximatively divided by a factor of 10^4 in the mean for large test samples of parameter in the applications

we experiment here. As a consequence, our reduced-basis approaches allow us to approximately divide the online computation time by a factor of 10^2, while maintaining the confidence intervals for the output expectation at the same value as computation without a reduced basis.

This work intends to present a new numerical method and to demonstrate its interest on some relevant test cases. We do not have, for the moment, a theoretical understanding of the method. This is the subject of future work.

2. The variance reduction issue and the control variate method

2.1. Mathematical preliminaries and the variance reduction issue.

Let $\left(B_t \in \mathbb{R}^d, t \in [0,T]\right)$ be a d-dimensional standard Brownian motion (where d is a positive integer) on a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$, endowed with a filtration $(\mathcal{F}_t, t \in [0,T])$. For any square-integrable random variables X, Y on that probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we respectively denote $\mathbb{E}(X)$ and $\mathbb{Var}(X)$ to be the expected value and the variance of X with respect to the probability measure $\mathbb{P}$, and $\mathbb{Cov}(X; Y)$ to be the covariance between X and Y.

For every $\lambda \in \Lambda$ (Λ being the set of parameter values), the Itô processes $\left(X_t^\lambda \in \mathbb{R}^d, t \in [0,T]\right)$ with deterministic initial condition $x \in \mathbb{R}^d$ are well defined as the solutions to the SDEs (1.2) under suitable assumptions on b^λ and σ^λ, for instance provided b^λ and σ^λ satisfy Lipschitz and growth conditions [13]. Let (X_t^λ) be solutions to the SDEs and f^λ, g^λ be measurable functions such that Z^λ is a well-defined integrable random variable $(Z^\lambda \in L_\mathbb{P}^1(\Omega))$. Then, Kolmogorov's strong law of large numbers holds and, denoting Z_m^λ $(m = 1, \ldots, M)$ to be M independent copies of the random variables Z^λ (for all positive integer M), the output expectation $\mathbb{E}\left(Z^\lambda\right) = \int_\Omega Z^\lambda d\mathbb{P}$ can be approximated (almost surely) by Monte-Carlo estimations of the form

$$\mathrm{E}_M\left(Z^\lambda\right) := \frac{1}{M} \sum_{m=1}^M Z_m^\lambda \xrightarrow[M \to \infty]{\mathbb{P}-a.s.} \mathbb{E}\left(Z^\lambda\right). \tag{2.1}$$

Furthermore, assume that the random variable Z^λ is square integrable $(Z^\lambda \in L_\mathbb{P}^2(\Omega))$ with variance $\mathbb{Var}\left(Z^\lambda\right)$. Then an asymptotic error bound for the convergence occuring in (2.1) is given in probabilistic terms by the Central Limit Theorem as confidence intervals: for all $a > 0$,

$$\mathbb{P}\left(\left|\mathrm{E}_M\left(Z^\lambda\right) - \mathbb{E}\left(Z^\lambda\right)\right| \leq a\sqrt{\frac{\mathbb{Var}\left(Z^\lambda\right)}{M}}\right) \xrightarrow[M \to \infty]{} \int_{-a}^a \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx. \tag{2.2}$$

In terms of the error bound (2.2), an approximation $\mathrm{E}_M\left(Z^\lambda\right)$ of the output $\mathbb{E}\left(Z^\lambda\right)$ is thus all the better, for a given M, as the variance $\mathbb{Var}\left(Z^\lambda\right)$ is small. In a many-query framework, the computation of approximations (2.1) for many outputs $\mathbb{E}\left(Z^\lambda\right)$ (corresponding to many queried values of the parameter $\lambda \in \Lambda$) would then be all the faster as the variance $\mathbb{Var}\left(Z^\lambda\right)$ for some $\lambda \in \Lambda$ could be decreased from some knowledge acquired from the $\lambda \in \Lambda$ computed beforehand. This typically defines a many-query setting with parametrized output suitable for a reduced-basis approach similar to the reduced-basis method developed in a deterministic setting for parametrized PDEs.

In addition, the convergence (2.1) controlled by the confidence intervals (2.2) can be easily observed using computable *a posteriori* estimators. Indeed, remember that since the random variable Z^λ has a finite second moment, then the strong law of large numbers also implies the following convergence:

$$\mathrm{Var}_M\left(Z^\lambda\right) := \mathrm{E}_M\left(\left(Z^\lambda - \mathrm{E}_M(Z^\lambda)\right)^2\right) \xrightarrow[M \to \infty]{\mathbb{P}-a.s.} \mathbb{Var}\left(Z^\lambda\right). \tag{2.3}$$

Combining the Central Limit Theorem with Slutsky theorem for the couple of Monte-Carlo estimators $\left(\mathrm{E}_M\left(Z^\lambda\right), \mathrm{Var}_M\left(Z^\lambda\right)\right)$ (see for instance [9], exercise 7.2.(26)), we obtain a fully computable probabilistic (asymptotic) error bound for the Monte-Carlo approximation (2.1) of the output expectation: for all $a > 0$,

$$\mathbb{P}\left(\left|\mathbb{E}\left(Z^\lambda\right) - \mathrm{E}_M\left(Z^\lambda\right)\right| \leq a\sqrt{\frac{\mathrm{Var}_M\left(Z^\lambda\right)}{M}}\right) \overset{M \to \infty}{\longrightarrow} \int_{-a}^a \frac{e^{-x^2/2}}{\sqrt{2\pi}}\, dx. \tag{2.4}$$

It is exactly the purpose of variance reduction techniques to reduce the so-called **statistical** error appearing in the Monte-Carlo estimation of the output expectation $\mathbb{E}\left(Z^\lambda\right)$ through the error bound (2.2). This is usually achieved in practice by using the (*a posteriori*) estimation (2.4).

REMARK 2.1 (SDE discretization and bias error in the output expectation). In practice, there is of course another source of error coming from the time-discretizations of the SDE (1.2) and of the integral involved in the expression for Z^λ.

In the following (for the numerical applications), we use the Euler-Maruyama numerical scheme with discretizations $0 = t_0 < t_1 < \cdots < t_N = T$ $(N \in \mathbb{N})$ of the time interval $[0, T]$ to approximate the Itô process (X_t^λ):

$$\begin{cases} \overline{X}_n^\lambda = \overline{X}_{n-1}^\lambda + |t_n - t_{n-1}|\ b^\lambda(t_{n-1}, \overline{X}_{n-1}^\lambda) + \sqrt{|t_n - t_{n-1}|}\sigma^\lambda(t_{n-1}, \overline{X}_{n-1}^\lambda)G_{n-1}, \\ \overline{X}_0^\lambda = x, \end{cases}$$

where $\{G_n,\ n = 0, \ldots, N-1\}$ is a collection of N independent d-dimensional normal centered Gaussian vectors. It is well-known that such a scheme is of weak order one, so that we have a bound for the **bias** due to the approximation of the output expectation $\mathbb{E}\left(Z^\lambda\right)$ by $\mathbb{E}\left(\overline{Z}^\lambda\right)$ (where $\overline{Z}^\lambda$ is a time-discrete approximation for Z^λ computed from $(\overline{X}_n^\lambda)$ with an appropriate discretization of the integral $\int_0^T f^\lambda(s, X_s^\lambda)ds$):

$$\left|\mathbb{E}\left(\overline{Z}^\lambda\right) - \mathbb{E}\left(Z^\lambda\right)\right| \underset{N \to \infty}{=} O\left(\max_{1 \leq n \leq N}(|t_n - t_{n-1}|)\right).$$

The approximation of the output $\mathbb{E}\left(Z^\lambda\right)$ by $\mathrm{E}_\mathrm{M}(\overline{Z}^\lambda)$ thus contains two types of errors:

- first, a bias $\mathbb{E}\left(Z^\lambda - \overline{Z}^\lambda\right)$ due to discretization errors in the numerical integration of the SDE (1.2) and of the integral involved in Z^λ,

- second, a statistical error of order $\sqrt{\mathbb{V}\mathrm{ar}\left(\overline{Z}^\lambda\right)/M}$ in the empirical Monte-Carlo estimation $\mathrm{E}_\mathrm{M}(\overline{Z}^\lambda)$ of the expectation $\mathbb{E}\left(\overline{Z}^\lambda\right)$.

We focus here on the statistical error.

2.2. Variance reduction with the control variate method. The idea of control variate methods for the Monte-Carlo evaluation of $\mathbb{E}\left(Z^\lambda\right)$ is to find a so-called *control variate* Y^λ (with $Y^\lambda \in L_\mathbb{P}^2(\Omega)$), and then to write:

$$\mathbb{E}\left(Z^\lambda\right) = \mathbb{E}\left(Z^\lambda - Y^\lambda\right) + \mathbb{E}\left(Y^\lambda\right),$$

where $\mathbb{E}\left(Y^\lambda\right)$ can be easily evaluated, while the expectation $\mathbb{E}\left(Z^\lambda - Y^\lambda\right)$ is approximated by Monte-Carlo estimations that have a smaller statistical error than direct Monte-Carlo estimations of $\mathbb{E}\left(Z^\lambda\right)$. In the following, we will consider control variates Y^λ such that $\mathbb{E}\left(Z^\lambda\right) = \mathbb{E}\left(Z^\lambda - Y^\lambda\right)$, equivalently,

$$\mathbb{E}\left(Y^\lambda\right) = 0.$$

The control variate method will indeed be interesting if the statistical error of the Monte-Carlo estimations $E_M(Z^\lambda - Y^\lambda)$ is significantly smaller than the statistical error of the Monte-Carlo estimations $E_M(Z^\lambda)$. That is, considering the following error bound given by the Central Limit Theorem: for all $a > 0$,

$$\mathbb{P}\left(\left|E_M\left(Z^\lambda - Y^\lambda\right) - \mathbb{E}\left(Z^\lambda\right)\right| \leq a\sqrt{\frac{\mathbb{V}\mathrm{ar}\left(Z^\lambda - Y^\lambda\right)}{M}}\right) \overset{M\to\infty}{\longrightarrow} \int_{-a}^{a} \frac{e^{-x^2/2}}{\sqrt{2\pi}}dx, \qquad (2.5)$$

the Monte-Carlo estimations $E_M(Z^\lambda - Y^\lambda)$ will indeed be more accurate approximations of the expectations $\mathbb{E}\left(Z^\lambda\right)$ than the Monte-Carlo estimations $E_M(Z^\lambda)$ provided

$$\mathbb{V}\mathrm{ar}\left(Z^\lambda\right) \geq \mathbb{V}\mathrm{ar}\left(Z^\lambda - Y^\lambda\right).$$

Clearly, the best possible control variate (in the sense of minimal variance) for a fixed parameter $\lambda \in \Lambda$ is

$$Y^\lambda = Z^\lambda - \mathbb{E}\left(Z^\lambda\right), \qquad (2.6)$$

since we then have $\mathbb{V}\mathrm{ar}\left(Z^\lambda - Y^\lambda\right) = 0$. Unfortunately, the result $\mathbb{E}\left(Z^\lambda\right)$ itself is necessary to compute Y^λ as $Z^\lambda - \mathbb{E}\left(Z^\lambda\right)$.

In the following, we will need another representation of the best possible control variate $Z^\lambda - \mathbb{E}\left(Z^\lambda\right)$. Under suitable assumptions on the coefficients b^λ and σ^λ (for well-posedness of the SDE), plus continuity and polynomial growth conditions on f^λ and g^λ, let us define $u^\lambda(t,y)$, for $(t,y) \in [0,T] \times \mathbb{R}^d$, as the unique solution $u^\lambda(t,y) \in C^1\left([0,T], C^2(\mathbb{R}^d)\right)$ to the backward Kolmogorov equation (2.7) satisfying the same polynomial growth assumptions at infinity than f^λ and g^λ (for instance, see Theorem 5.3 in [7]):

$$\begin{cases} \partial_t u^\lambda + b^\lambda(t,y) \cdot \nabla u^\lambda + \dfrac{1}{2}\sigma^\lambda(t,y)\sigma^\lambda(t,y)^T : \nabla^2 u^\lambda = f^\lambda(t,y)\ , \\ u^\lambda(T,y) = g^\lambda(y), \end{cases} \qquad (2.7)$$

where the notation ∇u^λ means $\nabla_y u^\lambda(t,y)$ and $\sigma^\lambda(t,y)\sigma^\lambda(t,y)^T : \nabla^2 u^\lambda$ means $\sum_{i,j,k=1}^{d} \sigma_{ik}^\lambda(t,y)\sigma_{jk}^\lambda(t,y)\partial_{y_i,y_j}^2 u^\lambda(t,y)$. Using the Itô formula for $\left(u^\lambda(t,X_t^\lambda), t \in [0,T]\right)$ with u^λ solution to (2.7), we get the following integral representation of Z^λ (see also Appendix A for another link between the SDE (1.2) and the PDE (2.7), potentially useful to numerics):

$$g^\lambda(X_T^\lambda) - \int_0^T f^\lambda(s,X_s^\lambda)ds = u^\lambda(0,x) + \int_0^T \nabla u^\lambda(s,X_s^\lambda) \cdot \sigma^\lambda(s,X_s^\lambda)dB_s. \qquad (2.8)$$

Note that the left-hand side of (2.8) is Z^λ, and the right-hand side is the sum of a stochatic integral (with zero mean) plus a scalar $u^\lambda(0,x)$ (thus equal to the expected value $\mathbb{E}\left(Z^\lambda\right)$ of the left-hand side). Hence, the optimal control variate also becomes

$$Y^\lambda = Z^\lambda - \mathbb{E}\left(Z^\lambda\right) = \int_0^T \nabla u^\lambda(s,X_s^\lambda) \cdot \sigma^\lambda(s,X_s^\lambda)dB_s. \qquad (2.9)$$

Of course, the formula (2.9) is again idealistic because, generally, numerically solving the PDE (2.7) is a very difficult task (especially in large dimensions $d \geq 4$).

2.3. Outline of the algorithms.   Considering either (2.6) or (2.9), we propose two algorithms for the efficient online computation of the family of parametrized outputs $\{\mathbb{E}(Z^\lambda), \lambda \in \Lambda\}$ when the parameter λ can take any value in a given range Λ, using (for each $\lambda \in \Lambda$) a control variate built as a linear combination of objects precomputed offline.

More precisely, in **Algorithm 1**, we do the following:
- Compute *offline* an accurate approximation $\tilde{Y}^\lambda$ of Y^λ using (2.6) for a small set of selected parameters $\lambda \in \{\lambda_1, \ldots, \lambda_I\} \subset \Lambda$ (where $I \in \mathbb{N}_{>0}$).
- For any $\lambda \in \Lambda$, compute *online* a control variate for the Monte-Carlo estimation of $\mathbb{E}(Z^\lambda)$ as a linear combination of $\{\tilde{Y}^{\lambda_i}, i=1, \ldots, I\}$:

$$\tilde{Y}_I^\lambda = \sum_{i=1}^I \mu_i^\lambda \tilde{Y}^{\lambda_i}.$$

And in **Algorithm 2**, we do the following:
- Compute *offline* an accurate approximation $\tilde{u}^\lambda$ of the solution u^λ to the Kolmogorov backward equation (2.7) for a small set of selected parameters $\lambda \in \{\lambda_1, \ldots, \lambda_I\} \subset \Lambda$.
- For any $\lambda \in \Lambda$, compute *online* a control variate for the Monte-Carlo computation of $\mathbb{E}(Z^\lambda)$, in view of (2.9), as a linear combination of $\int_0^T \nabla \tilde{u}^{\lambda_i}(s, X_s^\lambda) \cdot \sigma^\lambda(s, X_s^\lambda) dB_s$ (where $i = 1, \ldots, I$):

$$\tilde{Y}_I^\lambda = \sum_{i=1}^I \mu_i^\lambda \int_0^T \nabla \tilde{u}^{\lambda_i}(s, X_s^\lambda) \cdot \sigma^\lambda(s, X_s^\lambda) dB_s. \tag{2.10}$$

For a fixed size I of the reduced-basis, being given a parameter λ, both algorithms compute the coefficients μ_i^λ, $i = 1, \ldots, I$, with a view to minimizing the variance of the random variable $Z^\lambda - \tilde{Y}_I^\lambda$ (in practice, the empirical variance $\mathrm{Var}_M(Z^\lambda - \tilde{Y}_I^\lambda)$).

For the moment, we do not make more precise how we choose the set of parameters $\{\lambda_1, \ldots, \lambda_I\}$ offline. This will be done by the same *greedy procedure* for both algorithms, and will be the subject of the next section. Nevertheless, we would now like to make more precise how we build offline:

- in Algorithm 1, approximations $\{\tilde{Y}^{\lambda_i}, i=1, \ldots, I\}$ for $\{Y^{\lambda_i}, i=1, \ldots, I\}$, and
- in Algorithm 2, approximations $\{\nabla \tilde{u}^{\lambda_i}, i=1, \ldots, I\}$ for $\{\nabla u^{\lambda_i}, i=1, \ldots, I\}$,

assuming the parameters $\{\lambda_i, i=1, \ldots, I\}$ have been selected.

For Algorithm 1, $\tilde{Y}^{\lambda_i}$ is built using the fact that it is possible to compute offline *accurate* Monte-Carlo approximations $\mathrm{E}_M(Z^{\lambda_i})$ of $\mathbb{E}(Z^{\lambda_i})$ using a very large number $M = M_{\mathrm{large}}$ of copies of Z^{λ_i}, mutually independent and also independant of the copies of Z^λ used for the online Monte-Carlo estimation of $\mathbb{E}(Z^\lambda)$, $\lambda \neq \lambda_i$ (remember that the amount of offline computations is not meaningful in the case of a very large number of outputs to be computed online). The quantities $\mathrm{E}_{M_{\mathrm{large}}}(Z^{\lambda_i})$ are just real numbers that can be easily stored in memory at the end of the offline stage for re-use online to approximate the control variate $Y^{\lambda_i} = Z^{\lambda_i} - \mathbb{E}(Z^{\lambda_i})$ through

$$\tilde{Y}^{\lambda_i} = Z^{\lambda_i} - \mathrm{E}_{M_{\mathrm{large}}}(Z^{\lambda_i}). \tag{2.11}$$

For Algorithm 2, we compute approximations $\tilde{u}^{\lambda_i}$ as numerical solutions to the Kolmogorov backward equation (2.7). For example, in the numerical results of section 4, the PDE (2.7) is solved numerically with classical deterministic discretization methods (like finite differences in the calibration problem for instance).

REMARK 2.2 (Algorithm 2 for stochastic processes with large dimension d). Most deterministic methods to solve a PDE (such as finite difference or finite element methods) remain suitable only for $d \leq 3$. Beyond this, one can for example resort to probabilistic discretizations: namely, a Feynman-Kac representation of the PDE solution, whose efficiency at effectively reducing the variance has already been shown in [21]. We present this alternative probabilistic approximation in Appendix A, but we will not use it in the present numerical investigation.

One crucial remark is that for both algorithms, in the online Monte-Carlo computations the Brownian motions which are used to build the control variate (namely Z^{λ_i} in (2.11) for Algorithm 1, and the Brownian motion entering $\tilde{Y}_I^\lambda$ in (2.10) for Algorithm 2) are the same as those used for Z^λ.

Note last that, neglecting the approximation errors $\tilde{Y}^{\lambda_i} - Y^{\lambda_i}$ and $\tilde{u}^{\lambda_i} - u^{\lambda_i}$ in the reduced-basis elements computed offline, a comparison between Algorithms 1 and 2 is possible. Indeed, remembering the integral representation:

$$Y^{\lambda_i} = \int_0^T \nabla u^{\lambda_i}(s, X_s^{\lambda_i}) \cdot \sigma^{\lambda_i}(s, X_s^{\lambda_i}) dB_s,$$

we see that the reduced-basis approximation of Algorithm 1 has the form

$$Y_I^\lambda = \sum_{i=1}^I \mu_i^\lambda \int_0^T \nabla u^{\lambda_i}(s, X_s^{\lambda_i}) \cdot \sigma^{\lambda_i}(s, X_s^{\lambda_i}) dB_s,$$

while the reduced-basis approximation of Algorithm 2 has the form

$$Y_I^\lambda = \sum_{i=1}^I \mu_i^\lambda \int_0^T \nabla u^{\lambda_i}(s, X_s^\lambda) \cdot \sigma^\lambda(s, X_s^\lambda) dB_s.$$

The residual variances $\mathbb{Var}\left(Y^\lambda - Y_I^\lambda\right)$ for Algorithms 1 and 2 then respectively read as

$$\int_0^T \mathbb{E}\left(\left|\nabla u^\lambda \cdot \sigma^\lambda(s, X_s^\lambda) - \sum_{i=1}^I \mu_i^\lambda \nabla u^{\lambda_i} \cdot \sigma^{\lambda_i}(s, X_s^{\lambda_i})\right|^2\right) ds, \qquad (2.12)$$

and

$$\int_0^T \mathbb{E}\left(\left|\left(\nabla u^\lambda - \sum_{i=1}^I \mu_i^\lambda \nabla u^{\lambda_i}\right) \cdot \sigma^\lambda(s, X_s^\lambda)\right|^2\right) ds. \qquad (2.13)$$

The formulas (2.12) and (2.13) suggest that Algorithm 2 might be more robust than Algorithm 1 with respect to variations of λ. This will be illustrated by some numerical results in section 4.

3. Practical variance reduction with approximate control variates

Let us now detail how to select parameters $\{\lambda_i \in \Lambda, i = 1, \ldots, I\}$ offline inside a large *a priori* chosen trial sample $\Lambda_{\text{trial}} \subset \Lambda$ of finite size, and how to effectively compute the coefficients $(\mu_i^\lambda)_{i=1,\ldots,I}$ in the linear combinations $\tilde{Y}_I^\lambda$ (see section 3.3.2 for details about practical choices of $\Lambda_{\text{trial}} \subset \Lambda$).

Offline: select parameters $\{\lambda_i \in \Lambda_{\text{trial}}, i = 1, \ldots, I\}$ in $\Lambda_{\text{trial}} \subset \Lambda$, a large finite sample.

Selection under stopping criterion: maximal residual variance $\leq \varepsilon$.

> Let $\lambda_1 \in \Lambda$ be already chosen,
>
> Compute accurate approximation $E_{M_{\text{large}}}(Z^{\lambda_1})$ of $\mathbb{E}(Z^{\lambda_1})$.

Greedy procedure:

> For step $i = 1, \ldots, I - 1$ $(I > 1)$:
>
>> For all $\lambda \in \Lambda_{\text{trial}}$, compute $\tilde{Y}_i^\lambda$ as (3.2) and (cheap) estimations:
>>
>> $\epsilon_i(\lambda) := \text{Var}_{M_{\text{small}}}\left(Z^\lambda - \tilde{Y}_i^\lambda\right)$ for $\mathbb{V}\text{ar}\left(Z^\lambda - \tilde{Y}_i^\lambda\right)$.
>>
>> Select $\lambda_{i+1} \in \underset{\lambda \in \Lambda_{\text{trial}} \setminus \{\lambda_j, j=1,\ldots,i\}}{\arg\max} \{\epsilon_i(\lambda)\}$.
>>
>> If stopping criterion $\epsilon_i(\lambda_{i+1}) \leq \varepsilon$, Then Exit **Offline**.
>>
>> Compute accurate approximation $E_{M_{\text{large}}}(Z^{\lambda_{i+1}})$ of $\mathbb{E}(Z^{\lambda_{i+1}})$.

FIG. 3.1. *Offline stage for Algorithm 1: greedy procedure in metalanguage*

3.1. Algorithm 1.   Recall that some control variates Y^λ are approximated offline with a computationally expensive Monte-Carlo estimator using $M_{\text{large}} \gg 1$ independent copies of Z^λ:

$$\tilde{Y}^\lambda = Z^\lambda - E_{M_{\text{large}}}(Z^\lambda) \approx Y^\lambda, \tag{3.1}$$

for only a few parameters $\{\lambda_i, i = 1, \ldots, I\} \subset \Lambda_{\text{trial}}$ to be selected. The approximations $\tilde{Y}^{\lambda_i}$ are then used online to span a linear approximation space for the set of all control variates $\{Y^\lambda, \lambda \in \Lambda\}$, next linearly combined as $\tilde{Y}_I^\lambda$. For any $i = 1, \ldots, I$, we denote $\tilde{Y}_i^\lambda$ (for any $\lambda \in \Lambda$) to be the reduced-basis approximation of Y^λ built as a linear combination of the first i selected random variables $\{\tilde{Y}^{\lambda_j}, j = 1, \ldots, i\}$:

$$\tilde{Y}_i^\lambda = \sum_{j=1}^i \mu_j^\lambda \tilde{Y}^{\lambda_j} \approx Y^\lambda , \tag{3.2}$$

where $(\mu_j^\lambda)_{j=1,\ldots,i} \in \mathbb{R}^i$ is a vector of coefficients to be computed for each λ (and each step i, but we omit this to explicitly denote the dependence of each entry μ_j^λ, $j = 1, \ldots, i$, on i). The computation of the coefficients $(\mu_j^\lambda)_{j=1,\ldots,i}$ follows the same procedure offline (for each step $i = 1, \ldots, I - 1$) during the reduced-basis construction as online (when $i = I$): it is based on a variance minimization principle (see details in section 3.1.2).

With a view to computing $\mathbb{E}(Z^\lambda)$ online through computationally *cheap* Monte-Carlo estimations $E_{M_{\text{small}}}(Z^\lambda - \tilde{Y}_I^\lambda)$ using only a few M_{small} realizations for all $\lambda \in \Lambda$, we now explain how to select offline a subset $\{\lambda_i, i = 1, \ldots, I\} \subset \Lambda_{\text{trial}}$ in order to minimize $\mathbb{V}\text{ar}\left(Z^\lambda - \tilde{Y}_I^\lambda\right)$ (or at least estimators for the corresponding statistical error).

3.1.1. Offline stage : parameter selection.   The parameters $\{\lambda_i, i = 1, \ldots, I\}$ are selected *incrementally* inside the trial sample Λ_{trial} following a *greedy procedure* (see figure 3.1). The incremental search between steps i and $i + 1$ reads as follows. Assume that control variates $\{\tilde{Y}^{\lambda_j}, j = 1, \ldots, i\}$ have already been selected at

step i of the reduced basis construction (see Remark 3.4 for the choice of $\tilde{Y}^{\lambda_1}$). Then, $\tilde{Y}^{\lambda_{i+1}}$ is chosen following the principle of controlling the maximal residual variance inside the trial sample after the variance reduction using the first i selected random variables

$$\lambda_{i+1} \in \operatorname*{argmax}_{\lambda \in \Lambda_{\mathrm{trial}} \setminus \{\lambda_j, j=1,\ldots,i\}} \mathbb{V}\mathrm{ar}\left(Z^\lambda - \tilde{Y}_i^\lambda\right), \tag{3.3}$$

where the coefficients $(\mu_j^\lambda)_{j=1,\ldots,i}$ entering the linear combinations $\tilde{Y}_i^\lambda$ in (3.2) are computed, at each step i, like for $\tilde{Y}_I^\lambda$ in the online stage (see section 3.1.2).

In practice, the variance in (3.3) is estimated by an empirical variance

$$\mathbb{V}\mathrm{ar}\left(Z^\lambda - \tilde{Y}_i^\lambda\right) \simeq \mathrm{Var}_{\mathrm{M_{small}}}(Z^\lambda - \tilde{Y}_i^\lambda).$$

In our numerical experiments, we use the same number M_{small} of realizations for the offline computations (for all $\lambda \in \Lambda_{\mathrm{trial}}$) as for the online computations, even though this is not necessary. Note that choosing a small number M_{small} of realizations for the offline computations is advantageous because the computational cost of the Monte-Carlo estimations in the greedy procedure is then cheap. This is useful since Λ_{trial} is very large, and at each step i, $\mathrm{Var}_{\mathrm{M_{small}}}(Z^\lambda - \tilde{Y}_i^\lambda)$ has to be computed for all $\lambda \in \Lambda_{\mathrm{trial}}$.

Remarkably, after each (offline) step i of the greedy procedure and for the next online stage when $i = I$, only a few real numbers should be stored in memory, namely the collection $\{E_{M_{\mathrm{large}}}(Z^{\lambda_j}), j = 1,\ldots,i\}$ along with the corresponding parameters $\{\lambda_j, j = 1,\ldots,i\}$ for the computation of the approximations (3.1).

REMARK 3.1. Another natural criterion for the parameter selection in the greedy procedure could be the maximal residual variance **relatively** to the output expectation

$$\max_{\lambda \in \Lambda_{\mathrm{trial}}} \frac{\mathbb{V}\mathrm{ar}\left(Z^\lambda - \tilde{Y}_i^\lambda\right)}{|\mathbb{E}(Z^\lambda)|^2} \simeq \max_{\lambda \in \Lambda_{\mathrm{trial}}} \frac{\mathrm{Var}_{\mathrm{M_{small}}}(Z^\lambda - \tilde{Y}_i^\lambda)}{|\mathrm{E}_{\mathrm{M_{small}}}(Z^\lambda)|^2}. \tag{3.4}$$

This is particularly relevant if the magnitude of the output $\mathbb{E}\left(Z^\lambda\right)$ is much more sensitive than that of $\mathbb{V}\mathrm{ar}\left(Z^\lambda\right)$ to the variations on λ. It has also proved to be useful for comparison and discrimination between Algorithms 1 and 2 in the calibration of a local parametrized volatility for the Black-Scholes equation (see figure 4.5).

3.1.2. Online stage: reduced-basis approximation. To compute the coefficients $(\mu_j^\lambda)_{j=1,\ldots,i}$ in the linear combinations (3.2), both *online* for any $\lambda \in \Lambda$ when $i = I$ and *offline* for each $\lambda \in \Lambda_{\mathrm{trial}}$ and each step i (see greedy procedure above), we solve a small-dimensional least squares problem corresponding to the minimization of (estimators for) the variance of the random variable $Z^\lambda - \tilde{Y}_i^\lambda$.

More precisely, in the case $i = I$ (online stage) for instance, the I-dimensional vector $\mu^\lambda = (\mu_i^\lambda)_{1 \le i \le I}$ is defined, for any $\lambda \in \Lambda$, as the unique global minimizer of the following strictly convex problem of variance minimization:

$$\mu^\lambda = \operatorname*{argmin}_{\mu=(\mu_i)_{1 \le i \le I} \in \mathbb{R}^I} \mathbb{V}\mathrm{ar}\left(Z^\lambda - \sum_{i=1}^I \mu_i \tilde{Y}^{\lambda_i}\right), \tag{3.5}$$

or equivalently as the unique solution to the following linear system:

$$\sum_{j=1}^I \mathbb{C}\mathrm{ov}\left(\tilde{Y}^{\lambda_i}; \tilde{Y}^{\lambda_j}\right) \mu_j^\lambda = \mathbb{C}\mathrm{ov}\left(\tilde{Y}^{\lambda_i}; Z^\lambda\right), \forall i = 1,\ldots,I. \tag{3.6}$$

Of course, in practice, we use the estimator (for $X, Y \in L^2_{\mathbb{P}}(\Omega)$ and $M \in \mathbb{N}_{>0}$)

$$\text{Cov}_M(X;Y) := \frac{1}{M} \sum_{m=1}^{M} X_m Y_m - \left(\frac{1}{M} \sum_{m=1}^{M} X_m \right) \left(\frac{1}{M} \sum_{m=1}^{M} Y_m \right)$$

to evaluate the statistical quantities above. That is, defining a matrix $\mathbf{C}^{M_{small}}$ with entries the following empirical Monte-Carlo estimators ($i, j \in \{1, \dots, I\}$)

$$\mathbf{C}_{i,j}^{M_{small}} = \text{Cov}_{M_{small}} \left(\tilde{Y}^{\lambda_i}; \tilde{Y}^{\lambda_j} \right),$$

and a vector $\mathbf{b}^{M_{small}}$ with entries ($i \in \{1, \dots, I\}$) $\mathbf{b}_i^{M_{small}} = \text{Cov}_{M_{small}} \left(\tilde{Y}^{\lambda_i}; Z^{\lambda} \right)$, the linear combinations (3.2) are computed using as coefficients the Monte-Carlo estimators which are entries of the following vector of $\mathbb{R}^I$:

$$\mu^{M_{small}} = \left[\mathbf{C}^{M_{small}} \right]^{-1} \mathbf{b}^{M_{small}}. \tag{3.7}$$

The cost of one online computation for one parameter λ ranges as the computation of M_{small} (independent) realizations of the random variables $(Z^{\lambda}, Y^{\lambda_1}, \dots, Y^{\lambda_I})$, plus the Monte-Carlo estimators $E_{M_{small}}, \text{Cov}_{M_{small}}, \text{Var}_{M_{small}}$ and the computation of the solution $\mu^{M_{small}}$ to the (small I-dimensional, but full) linear system (3.7).

In practice, one should be careful when computing (3.7), because the likely quasi-colinearity of some reduced-basis elements often induces ill-conditioning of the matrix $\mathbf{C}^{M_{small}}$. Thus the QR or SVD algorithms [8] should be preferred to a direct inversion of (3.6) with Gaussian elimination or the Cholevsky decomposition. One important remark is that, once the reduced basis is built, the *same* (small I-dimensional) covariance matrix $\mathbf{C}^{M_{small}}$ has to be inverted for *all* $\lambda \in \Lambda$, as soon as the same Brownian paths are used for each online evaluation. The latter condition is easily satisfied in practice, simply by resetting the seed of the random number generator to the same value for each new online evaluation (that is for each new $\lambda \in \Lambda$).

REMARK 3.2 (Final output approximations and bounds).   It is a classical result that, taking first the limit $M_{large} \to \infty$ then $M_{small} \to \infty$, $\mu^{M_{small}} \xrightarrow[M_{small}, M_{large} \to \infty]{\mathbb{P}-a.s.} \mu^{\lambda}$. So, the variance is indeed (asymptotically) reduced to the minimum $\mathbb{V}\text{ar}\left(Z^{\lambda} - Y_I^{\lambda}\right)$ in (3.5), obtained with the optimal linear combination Y_I^{λ} of selected control variates Y^{λ_i} (without approximation). In addition, using the Slutsky theorem twice successively for Monte-Carlo estimators of the coefficient vector μ^{λ} and of the variance $\mathbb{V}\text{ar}\left(Z^{\lambda} - Y_I^{\lambda}\right)$, a computable version of the Central Limit Theorem also holds, which is similar to (2.4) except that it uses Monte-Carlo estimations of $Z^{\lambda} - \tilde{Y}_I^{\lambda}$ instead of Z^{λ} to compute the confidence intervals (and with successive limits $M_{large} \to \infty$, $M_{small} \to \infty$). So our output approximations now read for all $\lambda \in \Lambda$:

$$\mathbb{E}\left(Z^{\lambda}\right) \simeq E_{M_{small}} \left(Z^{\lambda} - \sum_{i=1}^{I} \mu_i^{M_{small}} \tilde{Y}^{\lambda_i} \right),$$

and asymptotic probabilistic error bounds are given by the confidence intervals (2.4).

3.2. Algorithm 2.   In Algorithm 2, approximations $\nabla \tilde{u}^{\lambda_i}$ of the gradients ∇u^{λ_i} of the solutions u^{λ_i} to the backward Kolmogorov equation (2.7) are computed offline for only a few parameters $\{\lambda_i, i = 1, \dots, I\} \subset \Lambda_{trial}$ to be selected. In comparison

> **Offline**: select parameters $\{\lambda_i \in \Lambda_{\text{trial}}, i=1,\ldots,I\}$ in $\Lambda_{\text{trial}} \subset \Lambda$ a large finite sample.
> Selection under stopping criterion: maximal residual variance $\leq \varepsilon$.
>
>   Let $\lambda_1 \in \Lambda$ be already chosen,
>
>   Compute approximation $\nabla \tilde{u}^{\lambda_1}$ of ∇u^{λ_1}.
>
>   Greedy procedure:
>
>   For step $i = 1,\ldots,I-1$ $(I > 1)$:
>
>   For all $\lambda \in \Lambda_{\text{trial}}$, compute $\tilde{Y}_i^\lambda$ as (3.8) and estimations:
>
> $\epsilon_i(\lambda) := \text{Var}_{M_{\text{small}}}\left(Z^\lambda - \tilde{Y}_i^\lambda\right)$ for $\mathbb{V}\text{ar}\left(Z^\lambda - \tilde{Y}_i^\lambda\right)$.
>
>   Select $\lambda_{i+1} \in \underset{\lambda \in \Lambda_{\text{trial}} \setminus \{\lambda_j, j=1,\ldots,i\}}{\arg\max} \{\epsilon_i(\lambda)\}$.
>
>   If stopping criterion $\epsilon_i(\lambda_{i+1}) \leq \varepsilon$, Then Exit **Offline**.
>
>   Compute approximation $\nabla \tilde{u}^{\lambda_{i+1}}$ of $\nabla u^{\lambda_{i+1}}$.

FIG. 3.2. *Offline stage for Algorithm 2: greedy procedure in metalanguage*

with Algorithm 1, approximations $(\nabla \tilde{u}^{\lambda_i})_{i=1,\ldots,I}$ are now used online to span a linear approximation space for $\{\nabla u^\lambda, \lambda \in \Lambda\}$. At step i of the greedy procedure $(i=1,\ldots,I)$, the reduced-basis approximations $\tilde{Y}_i^\lambda$ for the control variates Y^λ are (for all $\lambda \in \Lambda$)

$$\tilde{Y}_i^\lambda = \sum_{j=1}^{i} \mu_j^\lambda \tilde{Y}_\lambda^{\lambda_j} \approx Y^\lambda, \tag{3.8}$$

$$\tilde{Y}_\lambda^{\lambda_j} = \int_0^T \nabla \tilde{u}^{\lambda_j}(s, X_s^\lambda) \cdot \sigma^\lambda(s, X_s^\lambda) dB_s, \tag{3.9}$$

where $(\mu_j^\lambda)_{j=1,\ldots,i}$ are coefficients to be computed for each λ (again, the dependence of μ_j^λ on the step i is implicit). Again, the point is to explain, first, how to select parameters $\{\lambda_i, i=1,\ldots,I\} \subset \Lambda_{\text{trial}}$ in the offline stage, and second, how to compute the coefficients $(\mu_j^\lambda)_{j=1,\ldots,i}$ in each of the i-dimensional linear combinations $\tilde{Y}_i^\lambda$. Similar to Algorithm 1, the parameters $\{\lambda_i, i=1,\ldots,I\} \subset \Lambda_{\text{trial}}$ are selected offline following the greedy procedure, and, for any $i=1,\ldots,I$, the coefficients $(\mu_j^\lambda)_{j=1,\ldots,i}$ in the linear combinations offline and online are computed, following the same principle of minimizing the variance, by solving a least squares problem.

3.2.1. Offline stage: parameter selection. The selection of parameters $\{\lambda_j, j=1,\ldots,i\}$ from a trial sample Λ_{trial} follows a greedy procedure like in Algorithm 1 (see figure 3.2). In comparison with Algorithm 1, after i (offline) steps of the greedy procedure $(1 \leq i \leq I-1)$ and online $(i=I)$, note that discretizations of functions $(t,y) \to \nabla \tilde{u}^{\lambda_j}(t,y)$, $j=1,\ldots,i+1$, are stored in memory to compute the stochastic integrals (3.8), which is possibly a huge amount of data.

3.2.2. Online stage: reduced-basis approximation. Like in Algorithm 1, the coefficients $(\mu_j^\lambda)_{j=1,\ldots,i}$ in the linear combination (3.8) are computed similarly *online* (and then $i=I$) for any $\lambda \in \Lambda$ and *offline* (when $1 \leq i \leq I-1$) for each $\lambda \in \Lambda_{\text{trial}}$ as minimizers of (a Monte Carlo discretization of) the least squares problem:

$$\min_{\mu \in \mathbb{R}^I} \mathbb{V}\text{ar}\left(Z^\lambda - \sum_{i=1}^{I} \mu_i \tilde{Y}_\lambda^{\lambda_i}\right), \tag{3.10}$$

where we recall that $\tilde{Y}_\lambda^{\lambda_i}$ are defined by (3.9). Note that contrary to the reduced-basis elements $\tilde{Y}^{\lambda_i}$ in Algorithm 1, the elements $\tilde{Y}_\lambda^{\lambda_i}$ in Algorithm 2 have to be recomputed for *each* queried parameter value $\lambda \in \Lambda$.

Again, in practice, the unique solution $(\mu_j^\lambda)_{j=1,\dots,i}$ to the variational problem (3.10) is equivalently the unique solution to the following linear system:

$$\sum_{j=1}^{I} \mathbb{C}\mathrm{ov}\left(\tilde{Y}_\lambda^{\lambda_i}; \tilde{Y}_\lambda^{\lambda_j}\right) \mu_j^\lambda = \mathbb{C}\mathrm{ov}\left(\tilde{Y}_\lambda^{\lambda_i}; Z^\lambda\right), \forall i = 1,\dots,I, \tag{3.11}$$

and is in fact computed as the unique solution to the discrete minimization problem:

$$\mu^{\mathrm{M_{small}}} = \left[\mathbf{C}^{\mathrm{M_{small}}}\right]^{-1} \mathbf{b}^{\mathrm{M_{small}}}, \tag{3.12}$$

with $\mathbf{C}_{i,j}^{\mathrm{M_{small}}} = \mathrm{Cov}_{\mathrm{M_{small}}}\left(\tilde{Y}_\lambda^{\lambda_i}; \tilde{Y}_\lambda^{\lambda_j}\right)$ and $\mathbf{b}_i^{\mathrm{M_{small}}} = \mathrm{Cov}_{\mathrm{M_{small}}}\left(\tilde{Y}_\lambda^{\lambda_i}; Z^\lambda\right)$.

The cost of one computation online for one parameter λ is more expensive than that in Algorithm 1, and ranges as the computation of M_{small} independent realizations of Z^λ, **plus** the computation of I (discrete approximations of) the stochastic integrals (3.9), plus the Monte-Carlo estimators and the solution $\mu^{\mathrm{M_{small}}}$ to the (small I-dimensional, but full) linear system (3.12). In comparison to Algorithm 1, notice that the (discrete) covariance matrix $\mathbf{C}^{\mathrm{M_{small}}}$ to be inverted depends on λ, and thus cannot be treated offline once for all; it has to be recomputed for each $\lambda \in \Lambda$.

3.3. General remarks about reduced-basis approaches.   The success of our two reduced-basis approaches clearly depends on the variations of Z^λ with $\lambda \in \Lambda$. Unfortunately, we do not yet have a precise understanding of this, similarly to the PDE case [23]. Our reduced-basis approaches have only been investigated numerically in relevant cases for application (see section 4). So we now provide some theoretical ground only for the *a priori* existence of a reduced basis, like in the PDE case [18], with tips for a practical use of the greedy selection procedure based on our numerical experience. Of course, it remains to show that the greedy procedure actually selects a good reduced basis.

3.3.1. *A priori* **existence of a reduced basis.**   Following the analyses [18, 23] for parametrized PDEs, we can prove the *a priori* existence of a reduced basis for some particular collections of parametrized control variates, under very restrictive assumptions on the structure of the parametrization.

PROPOSITION 3.3.   *Assume there exist collections of uncorrelated (parameter-independent) random variables with zero mean $Y_j \in L_\mathbb{P}^2(\Omega)$, $1 \le j \le J$, and of positive $C^\infty(\mathbb{R})$ functions g_j, $1 \le j \le J$, such that*

$$Y^\lambda = \sum_{j=1}^{J} g_j(\lambda) Y_j, \ \forall \lambda \in \Lambda, \tag{3.13}$$

and there exists a constant $C > 0$ such that, for all parameter ranges $\Lambda = [\lambda_{\min}, \lambda_{\max}] \subset \mathbb{R}$, there exists a C^∞ diffeomorphism τ_Λ defined on Λ satisfying

$$\sup_{1 \le j \le J} \sup_{\tilde{\lambda} \in \tau_\Lambda(\Lambda)} (g_j \circ \tau_\Lambda^{-1})^{(M)}(\tilde{\lambda}) \le M! C^M, \text{ for all } M\text{-derivatives of } g_j \circ \tau_\Lambda^{-1}. \tag{3.14}$$

Then, for all parameter ranges $\Lambda = [\lambda_{\min}, \lambda_{\max}] \subset \mathbb{R}$, there exist constants $c_1, c_2 > 0$ independent of Λ and J such that, for all $N \in \mathbb{N}, N \ge N_0 := 1 + c_1(\tau_\Lambda(\lambda_{\max}) - \tau_\Lambda(\lambda_{\min}))$,

there exist N distinct parameter values $\lambda_n^N \in \Lambda$, $n=1,\ldots,N$, (with $\lambda_n^N \leq \lambda_{n+1}^N$ when $1 \leq n \leq N-1$), sastisfying, with $\mathcal{Y}_N = \mathbf{Span}\left(Y^{\lambda_n^N}, n=1,\ldots,N\right)$,

$$\inf_{Y_N^\lambda \in \mathcal{Y}_N} \mathbb{V}\mathrm{ar}\left(Z^\lambda - Y_N^\lambda\right) \leq e^{-\frac{c_2}{N_0-1}(N-1)} \mathbb{V}\mathrm{ar}\left(Z^\lambda\right), \ \forall \lambda \in \Lambda. \tag{3.15}$$

One can always write Y^λ like (3.13) with uncorrelated random variables (using a Gram-Schmidt procedure) and with positive coefficients (at least on a range Λ where they do not vanish). But the assumption (3.14) is much more restrictive. The mapping τ_Λ for the parameter, which depends on the functions g_j, $j=1,\ldots,J$, indeed tells us how the convergence depends on variations in the size of the parameter range Λ. See [18, 23] for an example of such functions g_j and τ_Λ, and Appendix B for a short proof inspired from [18, 23].

Proposition 3.3 may cover a few interesting cases of application for the *a priori* existence theory. One example where the assumption (3.13) holds is the following. Consider an output $Z^\lambda = g(X_T^\lambda)$ with g a polynomial function, and

$$X_t^\lambda = x + \int_0^t b^\lambda(s) X_s^\lambda \, ds + \int_0^t \sigma^\lambda(s) dB_s. \tag{3.16}$$

The optimal control variate Y^λ in such a case is written in the form (3.13) (to see this, one can first explicitly compute the reiterated (or multiple) Itô integrals in the polynomial expression of $g(X_T^\lambda)$ with Hermite polynomials [12]). Then, (3.14) may hold provided b^λ and σ^λ are smooth functions of $\lambda \in \Lambda$ (again, see [18, 23] for functions g_j satisfying (3.14)). But quite often, the reduced bases selected in practice by the *greedy* procedure are much better than $\mathcal{Y}_N$ (see [23] for comparisons when λ is scalar).

3.3.2. Requirements for efficient practical greedy selections. A comprehensive study would clearly need hypotheses about the regularity of Y^λ as a function of λ and about the discretization Λ_{trial} of Λ to show that the greedy procedure actually selects good reduced bases. We do not have precise results yet, but we would nevertheless like to provide the reader with conjectured requirements for the greedy procedure to work and help him as a potential user of our method.

Ideally, one would use the greedy selection procedure directly on $\{Y^\lambda, \lambda \in \Lambda\}$ for Algorithm 1 and on $\{\nabla u^\lambda, \lambda \in \Lambda\}$ for Algorithm 2. But in pratice, one has to resort to approximations only, $\{\tilde{Y}^\lambda, \lambda \in \Lambda\}$ for Algorithm 1 and $\{\nabla \tilde{u}^\lambda, \lambda \in \Lambda\}$ for Algorithm 2. So, following requirements on discretizations of parametrized PDEs in the classical reduced-basis method [23], the stability of the reduced basis selected by the greedy procedure for parametrized control variates intuitively requires

(H1) *For any required accuracy $\varepsilon > 0$, we assume the existence of approximations, $\tilde{Y}^\lambda$ for Y^λ in Algorithm 1 (resp. $\tilde{u}^\lambda$ for u^λ in Algorithm 2), such that the L^2-approximation error is uniformly bounded on Λ:*

$$\forall \lambda \in \Lambda, \mathbb{E}\left(|\tilde{Y}^\lambda - Y^\lambda|^2\right) \leq \varepsilon,$$

$$\left(resp. \ \int_0^T \mathbb{E}\left(|\nabla \tilde{u}^\lambda - \nabla u^\lambda|^2(X_t^\lambda)\right) dt \leq \varepsilon \ \text{or} \ \|\nabla \tilde{u}^\lambda - \nabla u^\lambda\|_{L^2}^2 \leq \varepsilon\right).$$

Moreover, in practice, one can only manipulate finite nested samples of parameter Λ_{trial} instead of the full range Λ. So some representativity assumption about Λ_{trial} is also intuitively required for the greedy selection procedure to work on Λ:

(H2)   *For any required accuracy $\varepsilon > 0$, we assume the existence of a sufficiently representative finite discrete subset $\Lambda_{\text{trial}} \subset \Lambda$ of parameters such that reduced bases built from Λ_{trial} are still good enough for Λ.*

Referring to section 3.3.1, *good enough* reduced bases should satisfy exponential convergence like (3.15), with slowly deteriorating capabilities in terms of approximation when the size of the parameter range grows. Now, in absence of a more precise result, intuition has been necessary so far to choose good discretizations. The numerical results of section 4 have been obtained with $M_{\text{large}} = 100\,M_{\text{small}}$ in Algorithm 1, and with a trial sample Λ_{trial} of 100 parameter values randomly chosen (with uniform distribution) in Λ.

In absence of theory for the greedy procedure, one could also think of using another parameter selection procedure in the offline stage. The interest of the greedy procedure is that it is cheap while effective in practice. In comparison, another natural reduced basis would be defined by the first I leading eigenvectors from the Principal Components Analysis (PCA) of the very large covariance matrix with entries $\mathbb{C}\text{ov}\left(Y^{\lambda_i}; Y^{\lambda_j}\right)_{(\lambda_i, \lambda_j) \in \Lambda_{\text{trial}} \times \Lambda_{\text{trial}}}$. The latter (known as the Proper Orthogonal Decomposition method) may yield similar variance reduction for most parameter values $\lambda \in \Lambda$ [23], but would certainly require more computations during the offline stage.

REMARK 3.4.   The choice of the first selected parameter λ_1 has not been made precise yet. It is observed that generally this choice does not impact the quality of the variance reduction. But to be more precise, we choose $\lambda_1 \in \Lambda_{\text{small trial}}$ such that Z^{λ_1} has maximal variance in a small initial sample $\Lambda_{\text{small trial}} \subset \Lambda$, for instance.

4. Worked examples and numerical tests

The efficiency of our reduced-basis strategies for parametrized problems is now investigated numerically for two problems relevant to some applications.

REMARK 4.1 (High-dimensional parameter).   Although the maximal dimension in the parameter treated here is two, one can reasonably hope for our reduced-basis approach to remain feasible with moderately high-dimensions in the parameter range Λ, say twenty. Indeed, a careful mapping of a multi-dimensional parameter range may allow for an efficient sampling Λ_{trial} that makes a greedy procedure tractable and next yields a good reduced basis for Λ, as was shown for the classical reduced-basis method with parametrized PDEs [25, 5].

4.1. Scalar process with constant drift and parametrized diffusion.

4.1.1. Calibration of the Black–Scholes model with local volatility.

One typical computational problem in finance is the valuation of an option depending on a risky asset with value S_t at time $t \in [0, T]$. In the following we consider Vanilla European Call options with payoff $\phi(S_T; K) = \max(S_T - K, 0)$, K being the exercise price (or strike) of the option at time $t = T$. By the no arbitrage principle for a portfolio mixing the risky asset of value S_t with a riskless asset of interest rate $r(t)$, the price (as a function of time) is a martingale given by a conditional expectation:

$$e^{-\int_t^T r(s)ds} \mathbb{E}\left(\phi(S_T) | \mathcal{F}_t\right) \tag{4.1}$$

where, in the Black-Scholes model with local volatility, $S_t = S_t^\lambda$ is a stochastic process solving the Black-Scholes equation:

$$dS_t^\lambda = S_t^\lambda \left(r(t)\,dt + \sigma^\lambda(t, S_t^\lambda)\,dB_t\right) \qquad S_{t=0}^\lambda = S_0, \tag{4.2}$$

and $(\mathcal{F}_t)$ is the natural filtration for the standard Brownian motion (B_t). For this model to be predictive the parameter λ in the (local) volatility σ^λ needs to be calibrated against observed data.

Calibration, like many numerical optimization procedures, defines a typical many-query context where one has to compute the price (4.1) of the option many times for a large number of parameter values until, for some optimal parameter value λ, a test of adequation with statistical data $P_{(K,\bar{t}_l)}$ observed on the market at times $\bar{t}_l \in [0,T]$, $l = 0, \dots, \bar{L}$ is satisfied. For instance, a common way to proceed is to minimize in λ the quadratic quantity

$$\mathcal{J}(\lambda) = \sum_{l=0}^{\bar{L}} \left| e^{-\int_{\bar{t}_l}^T r(s)\,ds} \mathbb{E}\left(\phi(S_T^\lambda; K)|\mathcal{F}_{\bar{t}_l}\right) - P_{(K,\bar{t}_l)} \right|^2,$$

most often regularized with some Tychonoff functional, using optimization algorithms like descent methods which indeed require many evaluations of the functional $\mathcal{J}(\lambda)$ for various λ. One could even consider the couple (K,T) as additional parameters to optimize the contract, but we do not consider such an extension here.

Note that the reduced-basis method for parameterized PDEs [17, 18, 23] has recently proved very efficient at treating a similar calibration problem [24]. Our approach is different since we consider a probabilistic pricing numerical method.

In the following numerical results, we solve (4.1) for many parameter values assuming that the interest rate r is a fixed given constant and the local volatility σ^λ has "hyperbolic" parametrization (4.3) (used by practitionners in finance)

$$\sigma^\lambda(t,S) = (\Gamma+1) \left(\frac{1}{C(0,S_0)} + \frac{\Gamma}{C(t,S)} \right)^{-1}, \tag{4.3}$$

where $C(t,S) = \frac{1}{2}\left(\sqrt{C_A(t,S)^2 + C_{\min}^2} + C_A(t,S) \right)$ with

$$C_A(t,S) = a + \frac{1}{2}\sqrt{(b-c)^2 \log^2\left(\frac{S}{\alpha S_0 e^{rt}}\right) + 4a^2 d^2} + \frac{1}{2}(b+c)\log\left(\frac{S}{\alpha S_0 e^{rt}}\right).$$

The local volatility σ^λ is thus parametrized with a 7-dimensional parameter $\lambda = (a,b,c,d,\alpha,\Gamma,C_{\min})$.

Our reduced-basis approach aims at building a vector space in order to approximate the family of random variables

$$\left\{ Y^\lambda := e^{-rT}\max(S_T^\lambda - K, 0) - e^{-rT}\mathbb{E}\left(\max(S_T^\lambda - K, 0)\right), \lambda \in \Lambda \right\},$$

which are optimal control variates for the computation of the expectation of $e^{-rT}\max(S_T^\lambda - K, 0)$. In Algorithm 2, we also use the fact that

$$Y^\lambda = \int_0^T \partial_S u^\lambda(t, S_t^\lambda)\sigma^\lambda(t, S_t^\lambda)S_t^\lambda\,dB_t, \tag{4.4}$$

where the function $u^\lambda(t,S)$ solves, for $(t,S) \in [0,T] \times (0,\infty)$

$$\partial_t u^\lambda(t,S) + rS\partial_S u^\lambda(t,S) + \frac{\sigma^\lambda(t,S)^2 S^2}{2}\partial_{SS}u^\lambda(t,S) = 0, \tag{4.5}$$

with final condition $u^\lambda(T,S) = e^{-rT}\max(S-K,0)$. Note the absence of boundary condition at $S=0$ because the advection and diffusion terms are zero at $S=0$. The backward Kolmogorov equation (4.5) is *numerically* solved using finite differences [1]. More precisely, after a change of variable $u^\lambda(t,S) = e^{-rt}C^\lambda(t,S)$, equation (4.4) is rewritten as

$$Y^\lambda = \int_0^T e^{-rt}\partial_S C^\lambda(t,S_t^\lambda)\sigma^\lambda(t,S_t^\lambda)S_t^\lambda dB_t, \tag{4.6}$$

where $C^\lambda(t,S)$ solves the classical Black-Scholes PDE

$$\partial_t C^\lambda(t,S) - rC^\lambda(t,S) + rS\partial_S C^\lambda(t,S) + \frac{\sigma^\lambda(t,S)^2 S^2}{2}\partial_{SS}C^\lambda(t,S) = 0, \tag{4.7}$$

with the final condition $C^\lambda(T,S) = \max(S-K,0)$. In the case of a low-dimensional variable S_t (like one-dimensional here), one can use a finite differences method of order 2 (with Crank-Nicholson discretization in time) to compute approximations $\tilde{C}_{l,j}^\lambda \simeq C^\lambda(t_l,x_j)$, $l=0,\dots,L$, $j=0,\dots,J$ on a grid for the truncated domain $[0,T]\times[0,3K]\subset[0,T]\times[0,\infty)$, with $L=100$ steps in time and $J=300$ steps in space of constant sizes (and with Dirichlet boundary condition $\tilde{C}_{l,J+1}^\lambda = (3-e^{-r(T-t_l)})K, \forall l = 0,\dots,N$ at the truncated boundary). An approximation $\tilde{C}^\lambda(t,S)$ of $C^\lambda(t,S)$ at any $(t,S)\in[0,T]\times[0,3K]$ is readily reconstructed as a linear interpolation on tiles $(t,S)\in[t_l,t_{l+1}]\times[S_j,S_{j+1}]$.

4.1.2. Numerical results.   The Euler-Maruyama scheme with $N=10^2$ time steps of constant size $\Delta t = \frac{T}{N} = 10^{-2}$ is used to compute one realization of a pay-off $\max(\tilde{S}_N^\lambda - K,0)$, for a strike $K=100$ at final time $t_N = T = 1$ when the initial price is $\tilde{S}_0^\lambda = 90$ and the interest rate $r=0.04$. Then, (a large number of) expectations $\mathbb{E}\left(\max(\tilde{S}_N^\lambda - K,0)\right)$ are approximated through Monte-Carlo evaluations $E_{\mathrm{M_{small}}}\left(\max(\tilde{S}_N^\lambda - K,0)\right)$ with $\mathrm{M_{small}} = 10^3$ realizations, when the local volatility parameter $\lambda = (a,b,c,d,\alpha,\Gamma,C_{\min})$ assumes many values in the two-dimensional range $\Lambda = [-.05,.15]\times\{b=c\in[.5,1.5]\}\times\{1.\}\times\{1.1\}\times\{5\}\times\{.05\}$ (variations of the function $\sigma^\lambda(t,S)$ with λ are shown in figure 4.1). We build reduced bases of different sizes $I=1,\dots,20$ from the same sample Λ_{trial} of size $|\Lambda_{\mathrm{trial}}| = 100$, either with Algorithm 1 (figures 4.2 and 4.4) using approximate control variates computed with $\mathrm{M_{large}} = 100\,\mathrm{M_{small}}$ evaluations:

$$\tilde{Y}_I^\lambda = \sum_{i=1}^I \mu_i^{\mathrm{M_{small}}}\tilde{Y}^{\lambda_i} = \sum_{i=1}^I \mu_i^{\mathrm{M_{small}}}\left(\max(\tilde{S}_N^{\lambda_i} - K,0) - E_{\mathrm{M_{large}}}\left(\max(\tilde{S}_N^{\lambda_i} - K,0)\right)\right),$$

or with Algorithm 2 (figure 4.3 and 4.4) using approximate control variates

$$\tilde{Y}_I^\lambda = \sum_{i=1}^I \mu_i^{\mathrm{M_{small}}}\left(\sum_{n=0}^{N-1} e^{-rt_n}\partial_S\tilde{C}^{\lambda_i}(t_n,\tilde{S}_n^\lambda)\sigma^\lambda(t_n,\tilde{S}_n^\lambda)\sqrt{|t_{n+1}-t_n|}G_n\right)$$

computed as first-order discretizations of the Itô stochastic integral (4.6) using the finite-difference approximation of the solution to the backward Kolmogorov equation. We always start the greedy selection procedure by choosing λ_1 such that $\tilde{Y}^{\lambda_1}$ has the maximal correlation with other members in $\Lambda_{\mathrm{small\ trial}}$, a small prior sample of 10 parameter values chosen randomly with uniform law in Λ; see Remark 3.4.
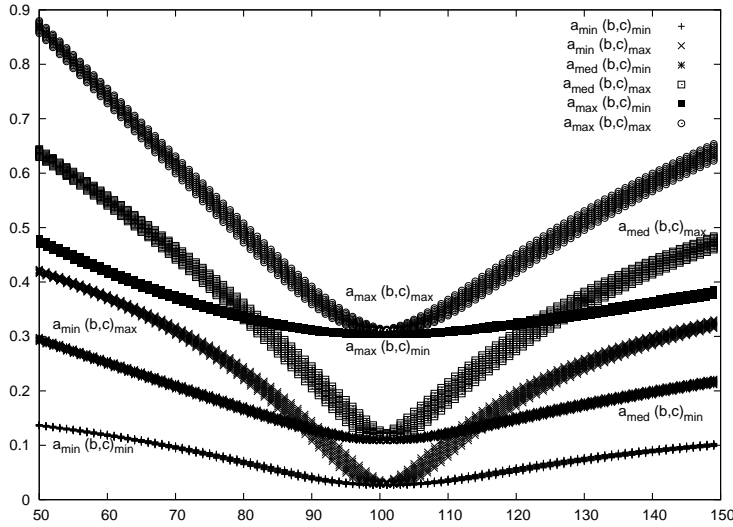
Fig. 4.1.   *Variations of the "hyperbolic" local volatility function* $\sigma^\lambda(t,S)$ *with respect to* $S \in [50,150]$. *Six families of curves are shown (as time t evolves in* $[0,1]$*) for extremal and mid- values of the parameter* $(a,b=c)$ *in* $[-.05,.15] \times \{b=c \in [.5,1.5]\}$*:* $(\min(a),\min(b=c))$, $(\min(a),\max(b=c))$, $(\text{med}(a):=.5\min(a)+.5\max(a),\min(b=c))$, $(\text{med}(a):= .5\min(a)+.5\max(a),\max(b=c))$, $(\max(a),\min(b=c))$, $(\max(a),\max(b=c))$. *Each family of curves shows the time variations of* $S \to \sigma^\lambda(t,S)$ *for* $t \in \{.1 \times k | k=0,\ldots,10\}$*).*

We show in figures 4.2 and 4.3 the absolute variance after variance reduction

$$\text{Var}_{M_{\text{small}}} \left(\max(\tilde{S}_N^\lambda - K, 0) - \tilde{Y}_I^\lambda \right), \tag{4.8}$$

and in figure 4.4 the relative variance after variance reduction

$$\frac{\text{Var}_{M_{\text{small}}} \left(\max(\tilde{S}_N^\lambda - K, 0) - \tilde{Y}_I^\lambda \right)}{\text{E}_{M_{\text{small}}} \left(\max(\tilde{S}_N^\lambda - K, 0) - \tilde{Y}_I^\lambda \right)^2}. \tag{4.9}$$

In each figure, the maximum, the minimum and the mean of one of the two residual variances above are shown, either within the offline sample deprived of the selected parameter values $\Lambda_{\text{trial}} \setminus \{\lambda_i, i=1,\ldots,I\}$, or within an online uniformly distributed sample test $\Lambda_{\text{test}} \subset \Lambda$ of size $|\Lambda_{\text{test}}| = 10|\Lambda_{\text{trial}}|$.

It seems that Algorithm 1 slightly outperfoms Algorithm 2 with a sufficiently large reduced basis, comparing the (online) decrease rates for either the relative variance or the absolute variance. Yet, one should also notice that, with very small-dimensional reduced basis, Algorithm 2 yields good variance reduction very rapidly. Comparing the decrease rates of the variance in offline and online samples tells us how good the (randomly uniformly distributed here) choice of Λ_{trial} was. Algorithm 2 seems more robust than Algorithm 1 for reproducing ("extrapolating") offline results from a sample Λ_{trial} in the whole range Λ. So, comparing the first results for Algorithms 1 and 2, it is not clear which algorithm performs the best variance reduction for a given size of the reduced basis.

Now, in figures 4.5 and 4.6, we show the online (absolute and relative) variance for a new sample test of parameters $\Lambda_{\text{testwide}}$ uniformly distributed in
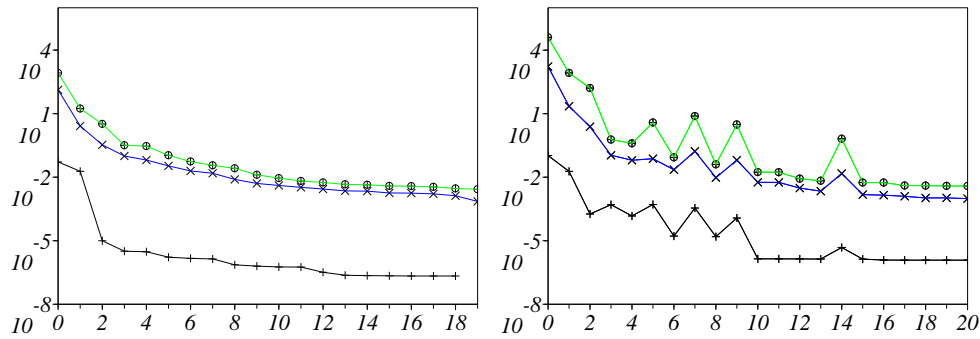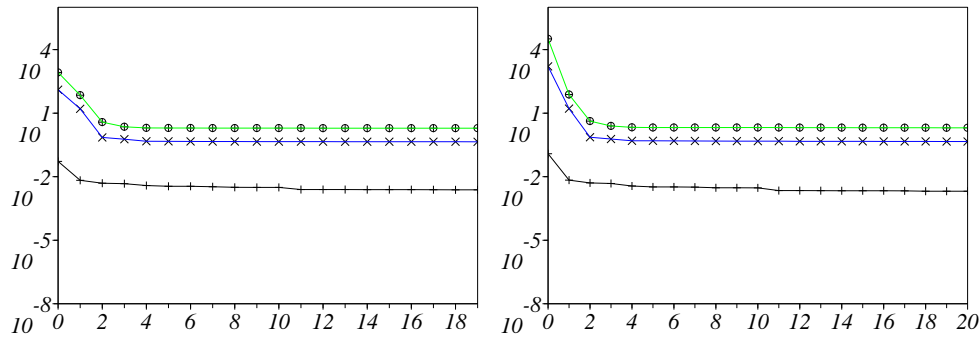
FIG. 4.2. *Algorithm 1 for Black–Scholes model with local "hyperbolic" volatility: Minimum +, mean × and maximum ∘ of the absolute variance (4.8) in samples of parameters (left: offline sample $\Lambda_{\text{trial}} \setminus \{\lambda_i, i=1,\dots,I\}$; right: online sample Λ_{test}) with respect to the size I of the reduced basis.*
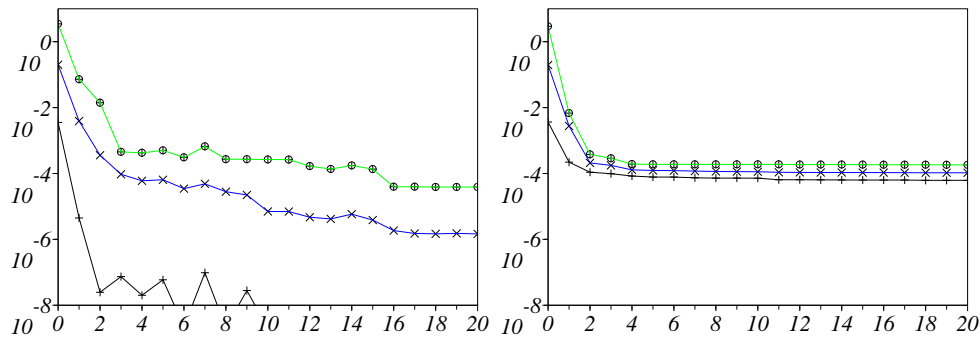

FIG. 4.3. *Algorithm 2 for Black–Scholes model with local "hyperbolic" volatility: Minimum +, mean × and maximum ∘ of the absolute variance (4.8) in samples of parameters (left: offline sample $\Lambda_{\text{trial}} \setminus \{\lambda_i, i=1,\dots,I\}$; right: online sample Λ_{test}) with respect to the size I of the reduced basis.*

FIG. 4.4. *Algorithm 1 (left) and 2 (right) for Black–Scholes model with local "hyperbolic" volatility: Minimum +, mean × and maximum ∘ of the relative variance (4.9) in a sample test (online) Λ_{test} of parameters with respect to the size I of the reduced basis.*

$\Lambda_{\text{wide}} = [-.15, .25] \times \{b = c \in]0, 2[\} \times \{1.\} \times \{1.1\} \times \{5\} \times \{.05\}$, which is twice as large as $\Lambda = [-.05, .15] \times \{b = c \in [.5, 1.5]\} \times \{1.\} \times \{1.1\} \times \{5\} \times \{.05\}$, where the training sample Λ_{trial} of the offline stage is nested. The quality of the variance reduction compared to that for a narrower sample test Λ_{test} seems to decrease faster for Algorithm 1 than for Algorithm 2. Algorithm 2 definitely seems more robust with respect to the variations in λ than Algorithm 1. This observation is even further increased if we use the relative variance (4.9) instead of the absolute variance (4.8), as shown by the results in figures 4.5 and 4.6.
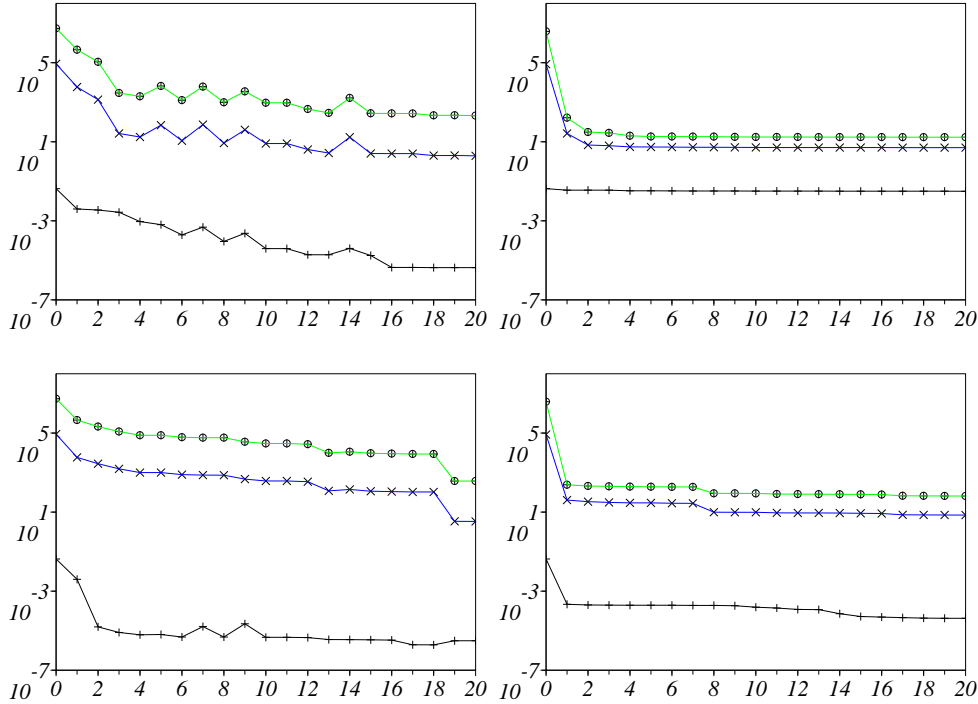

Fig. 4.5. *Algorithm 1 (left) and 2 (right) for Black–Scholes model with local "hyperbolic" volatility: Minimum +, mean × and maximum ∘ of the (online) absolute variance (4.8) in a sample test $\Lambda_{\text{testwide}}$ of parameters with respect to the size I of the reduced basis. Greedy selection with absolute variance (4.8) (top) and relative variance (4.9) (bottom).*

4.2. Vector processes with constant diffusion and parametrized drift.

4.2.1. Molecular simulation of dumbbells in polymeric fluids.
In rheology of polymeric viscoelastic fluids, the long polymer molecules responsible for the viscoelastic behaviour can be modelled through kinetic theories of statistical physics as Rouse chains, that is as chains of Brownian beads connected by springs. We concentrate on the most simple of those models, namely "dumbbells" (two beads connected by one spring) diluted in a Newtonian fluid.

Kinetic models consist in adding to the usual velocity and pressure fields $(\boldsymbol{u}, p)$ describing the (macroscopic) state of the Newtonian solvent, a field of dumbbells represented by their end-to-end vector $\boldsymbol{X}_t(\underline{x})$ at time t and position $\underline{x}$ in the fluid.
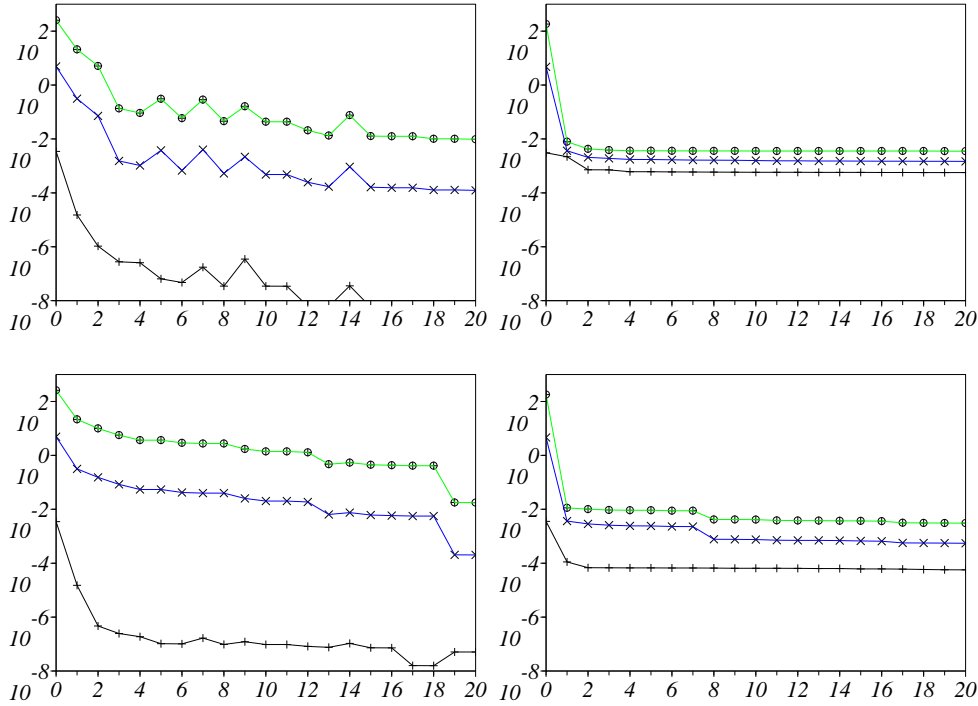
FIG. 4.6. *Algorithm 1 (left) and 2 (right) for Black–Scholes model with local "hyperbolic" volatility: Minimum +, mean × and maximum ○ of the (online) relative variance (4.9) in a sample test $\Lambda_{\text{testwide}}$ of parameters with respect to the size I of the reduced basis. Greedy selection with absolute variance (4.8) (top) and relative variance (4.9) (bottom).*

Vector stochastic processes $(\boldsymbol{X}_t(\underline{x}))$ encode the time evolution of the orientation and the stretch of the dumbbells (the idealized configuration of a polymer molecule) for each position $\underline{x} \in \mathcal{D}$ in a macroscopic domain $\mathcal{D}$ where the fluid flows. To compute the flow of a viscoelastic fluid with such multiscale dumbbell models [15], segregated algorithms are used that iteratively, on successive time steps with duration T:

- first evolve the velocity and pressure fields $(\boldsymbol{u}, p)$ of the Newtonian solvent under a fixed extra (polymeric) stress tensor field $\boldsymbol{\tau}$ (typically following Navier-Stokes equations), and

- then evolve the (probability distribution of the) polymer configurations vector field $(\boldsymbol{X}_t(\underline{x}))$ surrounded by the newly computed fixed velocity field $\boldsymbol{u}$.

The physics of kinetic models is based on a scale separation between the polymer molecules and the surrounding Newtonian fluid solvent. On the one side, the polymer configurations are directly influenced by the (local) velocity and pressure of the Newtonian solvent in which they are diluted. Reciprocally, on the other side, one needs to compute at every $\underline{x} \in \mathcal{D}$ the extra (polymeric) stress, given the Kramers formula:

$$\boldsymbol{\tau}(T, \underline{x}) = \mathbb{E}(\boldsymbol{X}_T(\underline{x}) \otimes \boldsymbol{F}(\boldsymbol{X}_T(\underline{x}))),$$

after one evolution step $t \in [0, T]$ over which the polymer configurations have evolved (remember that here $[0, T]$ should be understood as a timestep). The vector valued

process $\boldsymbol{X}_t(\underline{x})$ in $\mathbb{R}^d$ ($d=2$ or 3) solves a Langevin equation at every physical point $\underline{x} \in \mathcal{D}$ (Eulerian description):

$$d\boldsymbol{X}_t + \boldsymbol{u} \cdot \nabla_{\underline{x}} \boldsymbol{X}_t dt = \left((\nabla_{\underline{x}} \boldsymbol{u}) \boldsymbol{X}_t - \boldsymbol{F}(\boldsymbol{X}_t) \right) dt + d\boldsymbol{B}_t.$$

This Langevin equation describes the evolution of polymers at each $\underline{x} \in \mathcal{D}$, under an advection $\boldsymbol{u} \cdot \nabla_{\underline{x}} \boldsymbol{X}_t$, a hydrodynamic force $(\nabla_{\underline{x}} \boldsymbol{u}) \boldsymbol{X}_t$, Brownian collisions ($\boldsymbol{B}_t$) with the solvent molecules, and an entropic force $\boldsymbol{F}(\boldsymbol{X}_t)$ specific to the polymer molecules. Typically, this entropic force reads either $\boldsymbol{F}(\boldsymbol{X}) = \boldsymbol{X}$ (for Hookean dumbbells), or $\boldsymbol{F}(\boldsymbol{X}) = \frac{\boldsymbol{X}}{1-|\boldsymbol{X}|^2/b}$ (for Finitely-Extensible Nonlinear Elastic or FENE dumbbells, to model the finite extensibility of polymers: $|\boldsymbol{X}| < \sqrt{b}$).

In the following, we do not consider the advection term $\boldsymbol{u} \cdot \nabla_{\underline{x}} \boldsymbol{X}_t$ (which can be handled through integration of the characteristics in a semi-Lagrangian framework, for instance), and we concentrate on solving the parametrized SDE

$$d\boldsymbol{X}_t = \left(\underline{\underline{\lambda}} \boldsymbol{X}_t - \boldsymbol{F}(\boldsymbol{X}_t) \right) dt + d\boldsymbol{B}_t, \tag{4.10}$$

on a time slab $[0,T]$, with a fixed matrix $\underline{\underline{\lambda}}(\underline{x}) = \nabla_x \boldsymbol{u}(\underline{x})$. We also assume, as usual for viscoelastic fluids, that the velocity field is incompressible (that is $\text{tr}(\underline{\underline{\lambda}}) = 0$), hence the parameter $\underline{\underline{\lambda}}$ is only $(d^2 - 1)$-dimensional.

This is a typical many-query context where the Langevin equation (4.10) has to be computed many times at each (discretized) position $\underline{x} \in \mathcal{D}$, for each value of the $d \times d$-dimensional parameter $\underline{\underline{\lambda}}$ (since $\nabla_{\underline{x}} \boldsymbol{u}(\underline{x})$ depends on the position $\underline{x}$). Furthermore, the computation of the time-evolution of the flow defines a *very demanding* many-query context where the latter has to be done iteratively over numerous time steps of duration T between which the tensor field $\underline{\underline{\lambda}}(\underline{x})$ is evolved through a macroscopic equation for the velocity field $\boldsymbol{u}$.

REMARK 4.2 (Initial Condition of the SDE as additional parameter). Let $T_0 = 0$ and $T_{n+1} = (n+1)T$. Segregated numerical schemes for kinetic models of polymeric fluids as described above simulate (4.10) on successive time slabs $[T_n, T_{n+1}]$, for $n \in \mathbb{N}$. More precisely, on each time slab $[T_n, T_{n+1}]$, one has to compute

$$\begin{aligned} \boldsymbol{\tau}(T_{n+1}) &= \mathbb{E} \left(\boldsymbol{X}_{T_{n+1}} \otimes \boldsymbol{F}(\boldsymbol{X}_{T_{n+1}}) \right) \\ &= \mathbb{E} \left(\mathbb{E} \left(\boldsymbol{X}_{T_{n+1}} \otimes \boldsymbol{F}(\boldsymbol{X}_{T_{n+1}}) | \boldsymbol{X}_{T_n} \right) \right) \end{aligned} \tag{4.11}$$

at a fixed position $\underline{x} \in \mathcal{D}$. In practice, (4.11) can be approximated through

$$\boldsymbol{\tau}(T_{n+1}) \simeq \frac{1}{R} \sum_{r=1}^{R} \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{X}_{T_{n+1}}^{r,m} \otimes \boldsymbol{F}(\boldsymbol{X}_{T_{n+1}}^{r,m}), \tag{4.12}$$

after simulating MR processes $(\boldsymbol{X}_t^{r,m})_{t \in [T_n, T_{n+1}]}$ driven by MR independent Brownian motions for a given set of R *different* initial conditions, typically:

$$\boldsymbol{X}_{T_n^+}^{r,m} = \boldsymbol{X}_{T_n^-}^{r,1}, \ r = 1, \ldots, R, \ m = 1, \ldots, M,$$

or any $\boldsymbol{X}_{T_n^-}^{r,m_0}$ (with $1 \leq m_0 \leq M$) given by the computation at final time of the previous time slab $[T_{n-1}, T_n]$. In view of (4.12), for a fixed r, the computation of

$\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{X}_{T_{n+1}}^{r,m}\otimes\boldsymbol{F}(\boldsymbol{X}_{T_{n+1}}^{r,m})$ using the algorithms presented above requires a modification of the methods to the case when the initial condition of the SDE assumes many values.

To adapt Algorithm 1 to the context of SDEs with many different initial conditions, one should consider reduced bases for control variates which depend on the joint-parameter (λ, x), where x is the initial condition of the SDE. And variations on the joint-parameter (λ, x) can be simply recast into the framework of SDEs with fixed initial condition used for presentation of Algorithm 1 after the change of variable $\hat{X}_t^{\lambda,x} = X_t^\lambda - x$, that is using the family of SDEs with fixed initial condition $\hat{X}_0^{\lambda,x} = 0$

$$d\hat{X}_t^{\lambda,x} = \hat{b}^{\lambda,x}(t,\hat{X}_t^{\lambda,x})dt + \hat{\sigma}^{\lambda,x}(t,\hat{X}_t^{\lambda,x})dB_t, \qquad (4.13)$$

where $\hat{b}^{\lambda,x}(t,X) = b^\lambda(t,X+x)$, $\hat{\sigma}^{\lambda,x}(t,X) = \sigma^\lambda(t,X+x)$, for all t, X, and x. Then, with $\hat{g}^{\lambda,x}(X) = g^\lambda(X+x)$ and $\hat{f}^{\lambda,x}(t,X) = f^\lambda(t,X+x)$, the output is the expectation of

$$\hat{Z}^{\lambda,x} = \hat{g}^{\lambda,x}(\hat{X}_T^{\lambda,x}) - \int_0^T \hat{f}^{\lambda,x}(s,\hat{X}_s^{\lambda,x})ds.$$

The corresponding "ideal" control variate reads $\hat{Y}^{\lambda,x} = \hat{Z}^{\lambda,x} - \mathbb{E}\left(\hat{Z}^{\lambda,x}\right)$.

In Algorithm 2, note that u^λ, the solution to (2.7), does not depend on the initial condition used for the SDE. So, once parameters λ_i ($i=1,\ldots,I$) have been selected offline, Algorithm 2 applies similarly for SDEs with one fixed, or many different, initial conditions. Though, the offline selection of parameters λ_i using SDEs with many different initial conditions should consider a larger trial sample than for one fixed initial condition. Indeed, the selection criterion in the greedy algorithm does depend on the initial condition of the SDE. So, defining a trial sample of initial conditions Λ_{IC}, the following selection should be performed at step i in figure 3.2:

$$\text{Select} \quad \lambda_{i+1} \in \underset{\lambda \in \Lambda_{\text{trial}}\backslash\{\lambda_j, j=1,\ldots,i\}}{\arg\max} \ \underset{x \in \Lambda_{\text{IC}}}{\max} \text{Var}_{\text{M}_{\text{small}}}(Z^{\lambda,x} - \tilde{Y}_i^{\lambda,x}),$$

where $Z^{\lambda,x}$ and $\tilde{Y}_i^{\lambda,x}$, defined like Z^λ and $\tilde{Y}_i^\lambda$, depend on x because the stochastic process (X_t^λ) depends on $X_0^\lambda = x$.

It might be useful to build different reduced bases, one for each cell of a partition of the set of the initial condition. In summary, both algorithms can be extended to SDEs with variable initial condition, at the price of increasing the dimension of the parameter (see also Remark 4.1).

REMARK 4.3 (Multi-dimensional output). Clearly, the full output $\boldsymbol{\tau}$ in the problem described above is three-dimensional (it is a symmetric matrix). So our reduced-basis approach such as presented so far would need three different reduced bases, one for each scalar output. Though, one could alternatively consider the construction of only one reduced basis for the three outputs, which may be advantageous, see [4] for one example of such a construction.

Note that it is difficult to compute accurate approximations of the solution to the backward Kolmogorov equation (2.7) in the FENE case, because of the nonlinear explosive term. It is tractable in some situations, see [16, 6] for instance, though at the price of computational difficulties we did not want to deal with in this first work on our new variance reduction approach. On the contrary, the backward Kolmogorov

equation (2.7) can be solved exactly in the case of Hookean dumbells. Hence we have approximated here u^λ in Algorithm 2 by the numerical solution $\tilde{u}^\lambda$ to the backward Kolmogorov equation (2.7) for Hookean dumbells, whatever the type of dumbbells used for the molecular simulation (Hookean or FENE).

We would like to mention the recent work [14] where the classical reduced-basis method for parameterized PDEs has been used in the FENE case (solving the FENE Fokker-Planck by dedicated deterministic methods). Our approach is different since we consider a stochastic discretization.

4.2.2. Numerical results. The SDE (1.2) for FENE dumbbells (when $d=2$) is discretized with the Euler-Maruyama scheme using $N=100$ iterations with a constant time step of $\Delta t = 10^{-2}$ starting from a (deterministic) initial condition $\boldsymbol{X}_0 = (1,1)$, with reflecting boundary conditions at the boundary of the ball with radius $\sqrt{b}$.

The number of realizations used for the Monte-Carlo evaluations, and the sizes of the (offline) trial sample Λ_{trial} and (online) test sample Λ_{test} for the three-dimensional matrix parameter $\underline{\lambda}$ with entries ($\lambda_{11} = -\lambda_{22}, \lambda_{12}, \lambda_{21}$), are kept similar to the previous section 4.1. Samples Λ_{trial} and Λ_{test} for the parameter $\underline{\lambda}$ are uniformly distributed in a cubic range $\Lambda = [-1,1]^3$. We will also make use of an enlarged (online) test sample $\Lambda_{\text{testwide}}$, uniformly distributed in the range $[-2,2]^3$.

When $b=9$, the variance reduction online with Algorithm 1 is again very interesting, of about 4 orders of magnitude with $I=20$ basis functions, whatever the criterion used for the selection (we only show the absolute variance, in figure 4.7). But when $b=4$, the reflecting boundary conditions are more often active, and the maximum online variance reduction slightly degrades (see figure 4.8).

We first tested our variance reduction with Algorithm 2 for Hookean dumbells and it appeared to work well; but such a model is considered too simple generally. Then using the solution to the Kolmogorov backward equation for Hookean dumbells as $\tilde{u}^\lambda$ in Algorithm 2 for FENE dumbbells still yields good variance reduction while the boundary is not touched (see figure 4.9); when $b=4$ and many reflections at the boundary occur, the variance is hardly reduced. Again Algorithm 2 seems to be slightly more robust than Algorithm 1 in terms of extrapolation, that is when the (online) test sample is "enlarged" (see figure 4.10 with $b=16$ and a sample test (online) $\Lambda_{\text{testwide}}$).

5. Conclusion and perspectives

We have demonstrated the feasibility of a reduced-basis approach to compute control variates for the expectation of functionals of a parameterized Itô stochastic process. We have also tested the efficiency of such an approach with two possible algorithms, in two simple test cases where either the drift or the diffusion of scalar ($d=1$), and vector ($d=2$), Itô processes are parametrized, using 2- or 3-dimensional parameters.

Algorithm 2 is less generic than Algorithm 1; it is basically restricted to low-dimensional stochastic processes (X_t) since:

- one needs to solve (possibly high-dimensional) PDEs (offline), and
- discrete approximations of the PDEs solutions on a grid have to be kept in memory (which is possibly a huge amount of data).

Yet, Algorithm 2 seems more robust to variations in the parameter.

From a theoretical viewpoint, it remains to better understand the convergence of reduced-basis approximations for parametrized control variates depending on the
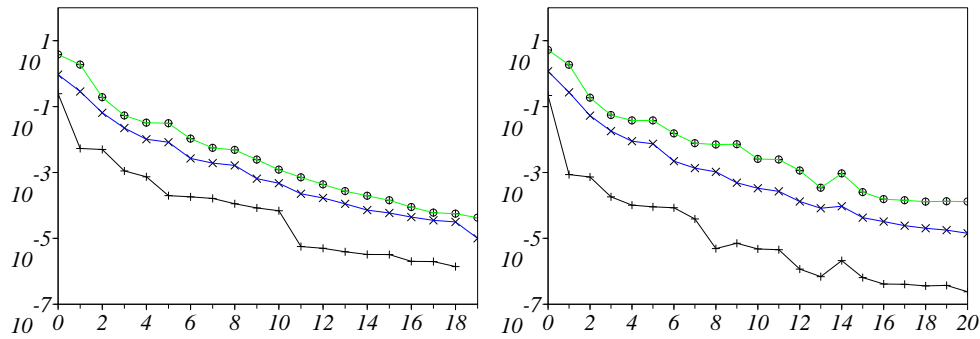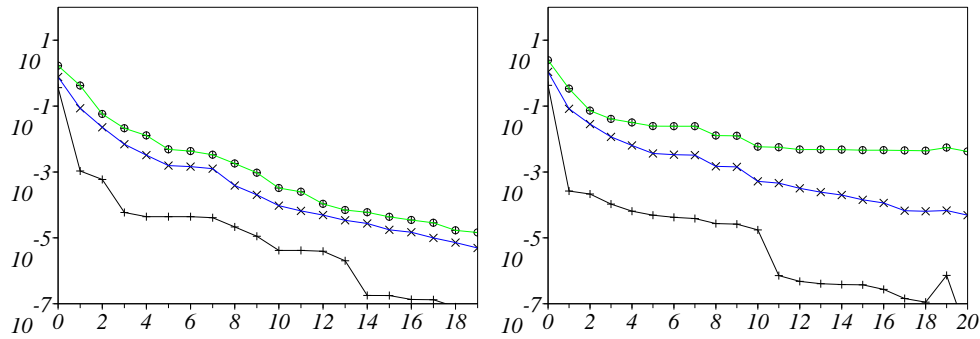
FIG. 4.7. *Algorithm 1 for FENE model with $b=9$: Minimum $+$, mean $\times$ and maximum $\circ$ of the absolute variance (4.8) in samples of parameters (left: offline sample $\Lambda_{\text{trial}} \setminus \{\lambda_i, i=1,\ldots,I\}$; right: online sample Λ_{test}) with respect to the size I of the reduced basis.*
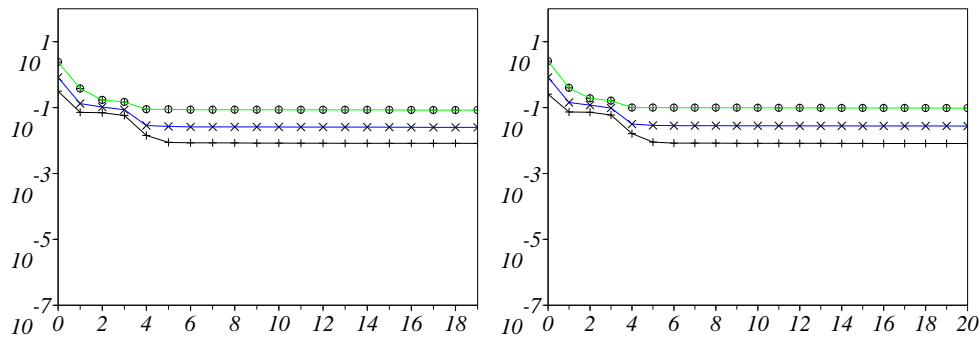

FIG. 4.8. *Algorithm 1 for FENE model with $b=4$: Minimum $+$, mean $\times$ and maximum $\circ$ of the absolute variance (4.8) in samples of parameters (left: offline sample $\Lambda_{\text{trial}} \setminus \{\lambda_i, i=1,\ldots,I\}$; right: online sample Λ_{test}) with respect to the size I of the reduced basis.*

FIG. 4.9. *Algorithm 2 for FENE model with $b=9$: Minimum $+$, mean $\times$ and maximum $\circ$ of the absolute variance (4.8) in samples of parameters (left: offline sample $\Lambda_{\text{trial}} \setminus \{\lambda_i, i=1,\ldots,I\}$; right: online sample Λ_{test}) with respect to the size I of the reduced basis.*
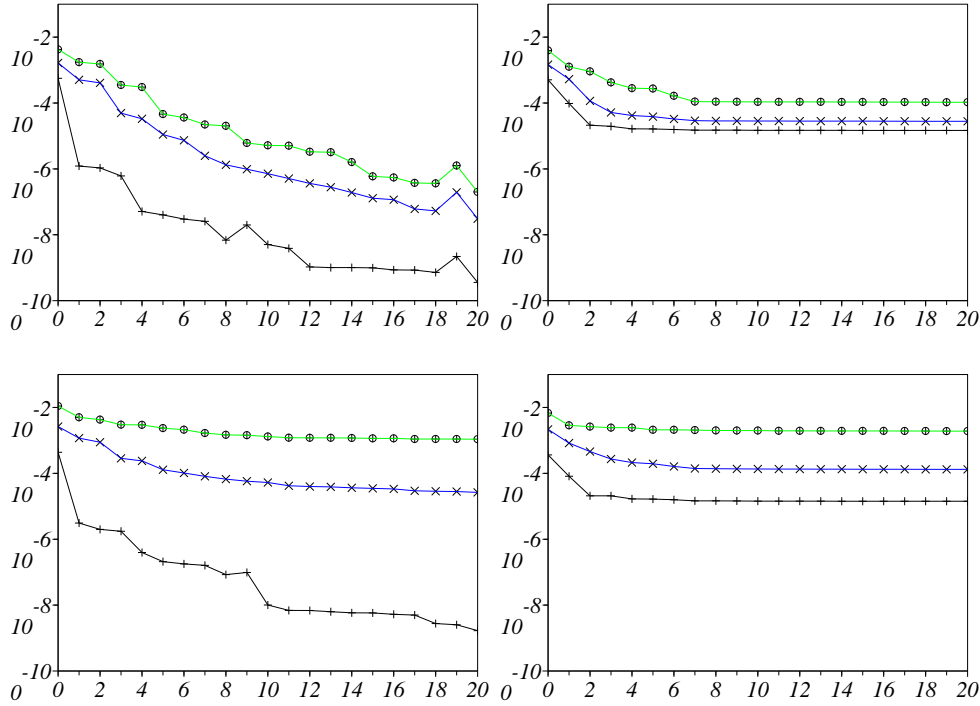
FIG. 4.10. *Algorithm 1 (left) and 2 (right) for FENE model with $b=16$: Minimum $+$, mean $\times$ and maximum $\circ$ of the relative variance* (4.9) *in online test for samples Λ_{test} (top) and $\Lambda_{\text{testwide}}$ (bottom) of parameters, with respect to the size I of the reduced basis.*

parametrization (and on the dimension of the parameter in particular), on the reduced-basis construction (following a greedy procedure) and on an adequate discretization choice (including the computation of approximate control variates and the choice of a trial sample Λ_{trial}).

Appendix A. Algorithm 2 in a higher-dimensional setting ($d\geq 4$). The solution $u^\lambda(t,y)$ to (2.7) can be computed at any $(t,y)\in[0,T]\times\mathbb{R}^d$ by the martingale representation theorem [12]

$$g^\lambda(X_T^\lambda)-\int_t^T f^\lambda(s,X_s^\lambda)ds = u^\lambda(t,X_t^\lambda)+\int_t^T \nabla u^\lambda(s,X_s^\lambda)\cdot\sigma^\lambda(s,X_s^\lambda)dB_s, \qquad (\text{A.1})$$

obtained by an Itô formula similar to (2.8). This gives the following Feynman-Kac formula for $u^\lambda(t,x)$, which can consequently be computed at any $(t,y)\in[0,T]\times\mathbb{R}^d$ through Monte-Carlo evaluations

$$u^\lambda(t,y)=\mathbb{E}\left(g^\lambda(X_T^{\lambda,t,y})-\int_t^T f^\lambda(s,X_s^{\lambda,t,y})ds\right), \qquad (\text{A.2})$$

where $(X_t^{\lambda,t_0,y})_{t_0 \le t \le T}$ is the solution to (1.2) with initial condition $X_{t_0}^{\lambda,t_0,y} = y$. Differentiating (A.2) (provided f^λ and g^λ are differentiable), we even directly get a Feynman-Kac formula for $\nabla u^\lambda(t,y)$,

$$\nabla u^\lambda(t,y) = \mathbb{E}\left(\Phi_T^{\lambda,t,y} \cdot \nabla g^\lambda(X_T^{\lambda,t,y}) - \int_t^T \Phi_s^{\lambda,t,y} \cdot \nabla f^\lambda(s, X_s^{\lambda,t,y}) ds\right), \qquad \text{(A.3)}$$

where the stochastic processes $\left(\Phi_s^{\lambda,t,y}, s \in [t,T]\right)$ in $\mathbb{R}^{d \times d}$ satisfy the first-order variation of the SDE (1.2) with respect to the initial condition, that is $\Phi_s^{\lambda,t,y} = \nabla_y X_s^{\lambda,t,y}$ for any $s \in [t,T]$:

$$\Phi_s^{\lambda,t,y} = \mathrm{Id}_d + \int_t^s \Phi_{s'}^{\lambda,t,y} \cdot \nabla b^\lambda(s', X_{s'}^{\lambda,t,y}) ds' + \int_t^s \Phi_{s'}^{\lambda,t,y} \cdot \nabla \sigma^\lambda(s', X_{s'}^{\lambda,t,y}) dB_{s'}, \quad \text{(A.4)}$$

where Id_d denotes the $d \times d$ identity matrix (see [21] for a more general and rigorous presentation of this Feynman-Kac formula in terms of the Malliavin gradient). The stochastic integral (2.9) can then be computed for each realization of (B_t), after discretizing $\left(\Phi_s^{\lambda,t,y}, s \in [t,T]\right)$.

Discrete approximations of the Feynman-Kac formula (A.3) have already been used succesfully in the context of computing control variates for the reduction of variance, in [21] for instance. Note that this numerical strategy to compute ∇u^λ from a Feynman-Kac formula requires a lot of computations. Yet, most often, the computation time of the functions $(t,y) \to \nabla u^\lambda(t,y)$ would not be a major issue in a reduced-basis approach, since this would be done *offline* (that is, in a pre-computation step, once for all) for only a few selected values of the parameter λ. What is nevertheless necessary for the reduced-basis approach to work is the possibility to store the big amount of data corresponding to a discretization of $\nabla u^\lambda(t,y)$ on a grid for the variable $(t,y) \in [0,T] \times \mathbb{R}^d$ (the parameter λ then assuming only a few values in Λ — of order 10 in our numerical experiments), and to have rapid access to those data in the *online* stage (where control variates are computed for any $\lambda \in \Lambda$ using those precomputed data).

Appendix B. Proof of Proposition 3.3.

Proof. First note that, since $\mathbb{E}(Y^\lambda) = 0$, then $\mathrm{Var}(Z^\lambda) = \mathrm{Var}(Y^\lambda)$.

So, for all $\lambda \in \Lambda$ and for every linear combination of $Y^{\lambda_n^N}$, $n = 1, \ldots, N$

$$Y_N^\lambda = \sum_{n=1}^N a_n(\lambda) \sum_{j=1}^J g_j(\lambda_n^N) Y_j$$

(with any choice $a_n(\lambda) \in \mathbb{R}$, $\lambda_n^N \in \Lambda$, $n = 1, \ldots, N$), there holds (recall that the Y_j, $j = 1, \ldots, J$, are uncorrelated)

$$\mathrm{Var}(Z^\lambda - Y_N^\lambda) = \mathrm{Var}(Y^\lambda - Y_N^\lambda) = \int_\Omega \left|\sum_{j=1}^J \left(g_j(\lambda) - \sum_{n=1}^N a_n(\lambda) g_j(\lambda_n^N)\right) Y_j\right|^2 d\mathbb{P}$$

$$\le \left(\sum_{j=1}^J |g_j(\lambda)|^2 \mathrm{Var}(Y_j)\right) \sup_{1 \le j \le J} \frac{\left|g_j(\lambda) - \sum_{n=1}^N a_n(\lambda) g_j(\lambda_n^N)\right|^2}{|g_j(\lambda)|^2}. \quad \text{(B.1)}$$

To get (3.15), we now explain how to choose the N coefficients $a_n(\lambda)$, $1 \le n \le N$, for each $\lambda \in \Lambda$ when $\lambda_n^N \in \Lambda$, $n = 1, \ldots, N$ is given, and then how to choose those N parameter values $\lambda_n^N \in \Lambda$, $n = 1, \ldots, N$.

Assume the N parameter values $\lambda_n^N \in \Lambda$, $n = 1, \ldots, N$, are given, with $\lambda_1^N = \lambda_{\min}$, $\lambda_N^N = \lambda_{\max}$ and $\lambda_n^N \le \lambda_{n+1}^N$, $n = 1, \ldots, N-1$. Then, for a given $M \in \{2, \ldots, N\}$ (to be determined later on) and for all $\lambda \in \Lambda$, it is possible to choose $1 \le M_0(\lambda) \le N+1-M$ such that $\lambda_{M_0(\lambda)}^N \le \lambda \le \lambda_{M_0(\lambda)+M-1}^N$. Only the M coefficients corresponding to the M contiguous parameter values above are taken non zero, such that $\forall \lambda \in \Lambda$:

$$a_m(\lambda) \ne 0 \Leftrightarrow M_0(\lambda) \le m \le M_0(\lambda) + M - 1,$$

and are more specifically chosen as $a_m(\lambda) = P_m^\lambda(\tau_\Lambda(\lambda))$ where P_m^λ are polynomials of degree $M-1$, such that, for all $M_0(\lambda) \le m, k \le M_0(\lambda) + M - 1$, $P_m^\lambda(\tau_\Lambda(\lambda_k)) = \delta_{mk}$. The polynomial function P_m^λ is the Lagrange interpolant defined on $[\tau_\Lambda(\lambda_{M_0(\lambda)}), \tau_\Lambda(\lambda_{M_0(\lambda)+M-1})]$, taking value 1 at $\tau_\Lambda(\lambda_m)$ and 0 at $\tau_\Lambda(\lambda_k)$, $k \ne m$. We will also need a function $d(\lambda) = |\tau(\lambda_{M_0(\lambda)}) - \tau(\lambda_{M_0(\lambda)+M-1})|$. Using a Taylor-Lagrange formula for $g_j \circ \tau^{-1}$ we have (for some $0 \le \eta \le 1$):

$$g_j(\lambda) - \sum_{n=1}^N a_n(\lambda) g_j(\lambda_n) = \frac{d(\lambda)^M}{M!} (g_j \circ \tau^{-1})^{(M)} \left(\eta \tau(\lambda_{M_0(\lambda)}^N) + (1-\eta) \tau(\lambda_{M_0(\lambda)+M-1}^N) \right).$$

Then, using (3.14) and the fact that $\mathbb{V}\mathrm{ar}\left(Z^\lambda\right) = \sum_{j=1}^J |g_j(\lambda)|^2 \mathbb{V}\mathrm{ar}(Y_j)$, there exists a constant $C > 0$ (independent of Λ and J) such that

$$\mathbb{V}\mathrm{ar}\left(Z^\lambda - Y_N^\lambda\right) \le \mathbb{V}\mathrm{ar}\left(Z^\lambda\right) (C\,d(\lambda))^{2M}, \ \forall \lambda \in \Lambda. \tag{B.2}$$

Finally, to get the result, we now choose a τ_Λ-equidistributed parameter sample:

$$\tau_\Lambda(\lambda_n^N) = \tau_\Lambda(\lambda_{\min}) + \frac{n-1}{N-1} \left(\tau_\Lambda(\lambda_{\max}) - \tau_\Lambda(\lambda_{\min}) \right), \ n = 1, \ldots, N.$$

Then, $d(\lambda) = \frac{M-1}{N-1} \left(\tau_\Lambda(\lambda_{\max}) - \tau_\Lambda(\lambda_{\min}) \right)$ does not depend on λ. Minimizing $(C\,d)^d$ as a function of $d \in (0, \frac{1}{C})$, we choose $d(\lambda) = \frac{1}{eC}$, and the choice $M = 1 + \left\lfloor \frac{1}{eC} \frac{N-1}{\tau_\Lambda(\lambda_{\max}) - \tau_\Lambda(\lambda_{\min})} \right\rfloor$ (where $\lfloor x \rfloor$ denotes the integer part of a real number $x \in \mathbb{R}$) finishes the proof provided $N \ge N_0 \equiv 1 + \lfloor C e \left(\tau_\Lambda(\lambda_{\max}) - \tau_\Lambda(\lambda_{\min}) \right) \rfloor$. $\square$

REFERENCES

[1] Y. Achdou and O. Pironneau, *Computational Methods for Option Pricing*, Frontiers in Applied Mathematics 30, SIAM, 2005.

[2] B. Arouna, *Robbins-Monroe algorithms and variance reduction in finance*, J. Comput. Finance, 7(2), 35–62, 2004.

[3] J. Bonvin and M. Picasso, *Variance reduction methods for CONNFFESSIT-like simulations*, J. Non-Newtonian Fluid Mech., 84, 191–215, 1999.

[4] S. Boyaval, *Reduced-basis approach for homogenization beyond the periodic setting*, SIAM Multiscale Modeling and Simulation, 7(1), 466–494, 2008.

[5] S. Boyaval, C. Le Bris, Y. Maday, N.C. Nguyen, and A.T. Patera, *A reduced basis approach for variational problems with stochastic parameters: application to heat conduction with variable Robin coefficient*, Computer Methods in Applied Mechanics and Engineering, 198(41–44), 3187–3206, 2009.

[6] C. Chauvière and A. Lozinski, *Simulation of dilute polymer solutions using a Fokker-Planck equation*, Computers and Fluids, 33, 687–696, 2004.

[7] A. Friedman, *Stochastic Differential Equations and Applications*, Academic Press (New York; London; Toronto), 1, 1975.

[8]   G. Golub and C. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, London, 1996.

[9]   G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 2nd ed., Oxford, 1992.

[10] J. Hammersley and D. Handscomb, eds., *Monte Carlo Methods*, Chapman and Hall Ltd, London, 1964.

[11] B. Jourdain, *Adaptive variance reduction techniques in finance*, Radon Series Comput. Appl. Math., De Gruyter, to appear, 8, 2009.

[12] I. Karatzas and S.E. Shreve, *Brownian Motion and Stochastic Calculus*, Springer-Verlag, 1991.

[13] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer, 2000.

[14] DJ Knezevic and AT Patera, *A certified reduced basis method for the Fokker-Planck equation of dilute polymeric fluids: FENE dumbbells in extensional flow*, SIAM J. Sci. Comput., in press.

[15] C. Le Bris and T. Lelièvre, *Multiscale modelling of complex fluids: a mathematical initiation*, Multiscale Modeling and Simulation in Science Series, B. Engquist, P. Löstedt, O. Runborg, eds., Lecture Notes in Computational Science and Engineering, Springer, 66, 49–138, 2009.

[16] A. Lozinski and C. Chauvière, *A fast solver for Fokker-Planck equations applied to viscoelastic flows calculations: 2D FENE model*, J. Comput. Phys., 189, 607–625, 2003.

[17] L. Machiels, Y. Maday and A.T. Patera, *Output bounds for reduced-order approximations of elliptic partial differential equations*, Comput. Methods Appl. Mech. Engrg., 190(26-27), 3413–3426, 2001.

[18] Y. Maday, A.T. Patera and G. Turinici, A Priori, *Convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations*, J. Sci. Comput., 17(1-4), 437–446, 2002.

[19] M. Melchior and H.C. Öttinger, *Variance reduced simulations of stochastic differential equations*, J. Chem. Phys., 103, 9506–9509, 1995.

[20] G.N. Milstein and M.V. Tretyakov, *Practical variance reduction via regression for simulating diffusions*, School of Mathematics and Computer Science, University of Leicester,Technical Report MA-06-19, 2006.

[21] N.J. Newton, *Variance reduction for simulated diffusions*, SIAM J. Appl. Math., 54(6), 1780–1805, 1994.

[22] H.C. Öttinger, B.H.A.A. van den Brule and M. Hulsen, *Brownian configuration fields and variance reduced CONNFFESSIT*, J. Non-Newtonian Fluid Mech., 70, 25–261, 1997.

[23] A.T. Patera and G. Rozza, *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*, Version 1.0, Copyright MIT 2006 - 2007, tentative rubric, MIT Pappalardo Graduate Monographs in Mechanical Engineering, to appear.

[24] O. Pironneau, *Calibration of options on a reduced basis*, J. Comp. Appl. Math., Special Issue: Honor of Professor Hideo Kawarada on the Occasion of his 70th Birthday, 232(1), 139–147, 2009.

[25] S. Sen, *Reduced-basis approximation and a posteriori error estimation for many-parameter heat conduction problems*, Numerical Heat Transfer, Part B: Fundamentals 54(5), 69–389, 2008.