

Adaptive selection of sampling points for Uncertainty Quantification

E. Camporeale, A. Agnihotri, C. Rutjes

Received: date / Accepted: date

Abstract We present a simple and robust strategy for the selection of sampling points in Uncertainty Quantification. The goal is to achieve the fastest possible convergence in the cumulative distribution function of a stochastic output of interest. We assume that the output of interest is the outcome of a computationally expensive nonlinear mapping of an input random variable, whose probability density function is known. We use a radial function basis to construct an accurate interpolant of the mapping. This strategy enables adding new sampling points one at a time, *adaptively*. This takes into full account the previous evaluations of the target nonlinear function. We present comparisons with a stochastic collocation method based on the Clenshaw-Curtis quadrature rule, and with an adaptive method based on hierarchical surplus, showing that the new method often results in a large computational saving.

1 Introduction

We address one of the fundamental problems in Uncertainty Quantification (UQ): the mapping of the probability distribution of a random variable through a nonlinear function. Let us assume that we are concerned with a specific physical or engineering model which is computationally expensive. The model is defined by the map $g : \mathbb{R} \rightarrow \mathbb{R}$. It takes a parameter X as input, and produces an output Y , $Y = g(X)$. In this paper we restrict ourselves to a proof-of-principle one-dimensional case. Let us assume that X is a random variable distributed with probability density function (pdf) P_X . The Uncertainty Quantification problem is the estimation of the pdf P_Y of the output variable Y , given P_X . Formally, the

Multiscale Dynamics
Center for Mathematics and Computer Science (CWI)
1090 GB Amsterdam
Netherlands
E-mail: e.camporeale@cwi.nl, a.agnihotri@cwi.nl, c.rutjes@cwi.nl

problem can be simply cast as a coordinate transformation and one easily obtains

$$P_Y(y) = \sum_{x \in \{x|g(x)=y\}} \frac{P_X(x)}{|\det J(x)|}, \quad (1)$$

where $J(x)$ is the Jacobian of $g(x)$. The sum over all x such that $g(x) = y$ takes in account the possibility that g may not be injective. If the function g is known exactly and invertible, Eq.(1) can be used straightforwardly to construct the pdf $P_Y(y)$, but this is of course not the case when the mapping g is computed via numerical simulations.

Several techniques have been studied in the last couple of decades to tackle this problem. Generally, the techniques can be divided in two categories: intrusive and non-intrusive [1–3]. Intrusive methods modify the original, *deterministic*, set of equations to account for the stochastic nature of the input (random) variables, hence eventually dealing with stochastic differential equations, and employing specific numerical techniques to solve them. Classical examples of intrusive methods are represented by Polynomial Chaos expansion [4–7], and stochastic Galerkin methods [8–11].

On the other hand, the philosophy behind non-intrusive methods is to make use of the deterministic version of the model (and the computer code that solves it) as a black-box, which returns one deterministic output for any given input. An arbitrary large number of solutions, obtained by sampling the input parameter space, can then be collected and analyzed in order to reconstruct the pdf $P_Y(y)$.

The paradigm of non-intrusive methods is perhaps best represented by Monte Carlo (MC) methods [12,13]: one can construct an ensemble of input parameters $\{X_n | n = 1, \dots, N\}$ (N typically large) distributed according to the pdf $P_X(x)$, run the corresponding ensemble of simulations $g : X \rightarrow Y$, and process the outputs $\{Y_n | n = 1, \dots, N\}$. MC methods are probably the most robust of all the non-intrusive methods. Their main shortcoming is the slow convergence of the method, with a typical convergence rate proportional to \sqrt{N} . For many applications quasi-Monte Carlo (QMC) methods [12,14] are now preferred to MC methods, for their faster convergence rate. In QMC the pseudo-random generator of samples is replaced by more uniform distributions, obtained through so-called quasi-random generators [15,16].

It is often said that MC and QMC do not suffer the ‘curse of dimensionality’ [17–19], in the sense that the convergence rate (but not the actual error!) is not affected by the dimension D of the input parameter space. Therefore, they represent the standard choice for large dimensional problems. On the other hand, when the dimension D is not very large, collocation methods [20–22] are usually more efficient.

Collocation methods recast an UQ problem as an interpolation problem. In collocation methods, the function $g(x)$ is sampled in a small (compared to the MC approach) number of points (‘collocation points’), and an interpolant is constructed to obtain an approximation of g over the whole input parameter space, from which the pdf $P_Y(y)$ can be estimated.

The question then arises on how to effectively choose the collocation points. Recalling that every evaluation of the function g amounts to performing an expensive simulation, the challenge resides in obtaining an accurate approximation of P_Y with the least number of collocation points.

As the name suggests, collocation methods are usually derived from classical quadrature rules [23–25].

The type of pdf P_X can guide the choice of the optimal quadrature rule to be used (i.e., Gauss-Hermite for a Gaussian probability, Gauss-Legendre for a uniform probability, etc. [20]). Furthermore, because quadratures are associated with polynomial interpolation, it becomes natural to define a global interpolant in terms of a Lagrange polynomial [26]. Also, choosing the collocation points as the abscissas of a given quadrature rule makes sense particularly if one is only interested in the evaluation of the statistical moments of the pdf (i.e., mean, variance, etc.) [27].

On the other hand, there are several applications where one is interested in the approximation of the full pdf P_Y . For instance, when g is narrowly peaked around two or more distinct values, its mean does not have any statistical meaning. In such cases one can wonder whether a standard collocation method based on quadrature rules still represents the optimal choice, in the sense of the computational cost to obtain a given accuracy.

From this perspective, a downside of collocation methods is that the collocation points are chosen a priori, without making use of the knowledge of $g(x)$ acquired at previous interpolation levels. For instance, the Clenshaw-Curtis (CC) method uses a set of points that contains 'nested' subset, in order to re-use all the previous computations, when the number of collocation points is increased. However, since the abscissas are unevenly spaced and concentrated towards the edge of the domain (this is typical of all quadrature rules, in order to overcome the Runge phenomenon [26,28]), it is likely that the majority of the performed simulations will not contribute significantly in achieving a better approximation of P_Y . Stated differently, one would like to employ a method where each new sampling point is chosen in such a way to result in the fastest convergence rate for the approximated P_Y , in contrast to a set of points defined a priori.

As a matter of fact, because the function g is unknown, a certain number of simulations will always be redundant, in the sense that they will contribute very little to the convergence of P_Y . The rationale for this work is to devise a method to minimize such a redundancy in the choice of sampling points while achieving fastest possible convergence of P_Y .

Clearly, this suggests to devise a strategy that chooses collocation points *adaptively*, making use of the knowledge of the interpolant of $g(x)$, which becomes more and more accurate as more points are added.

A well known adaptive sampling algorithm is based on the calculation of the so-called hierarchical surplus [29–32, see e.g.]. This is defined as the difference, between two levels of refinement, in the solution obtained by the interpolant. Although this algorithm is quite robust, and it is especially efficient in detecting discontinuities, it has the obvious drawback that it can be prematurely terminated, whenever the interpolant happens to exactly pass through the true solution on a point where the hierarchical surplus is calculated, no matter how inaccurate the interpolant is in close-by regions (see Figure 1 for an example).

The goal of this paper is to describe an alternative strategy for the adaptive selection of sampling points. The objective in devising such strategy is to have a simple and robust set of rules for choosing the next sampling point. The paper is concerned with a proof-of-principle demonstration of our new strategy, and we will focus here on one dimensional cases and on the case of uniform P_X only, postponing the generalization to multiple dimensions to future work. It is important

to appreciate that the stated goal of this work is different from the traditional approach followed in the overwhelming majority of works that have presented sampling methods for UQ in the literature. Indeed, it is standard to focus on the convergence of the nonlinear unknown function $g(x)$, trying to minimize the interpolation error on $g(x)$, for a given number of sampling points. On the other hand, we will show that the convergence rates of $g(x)$ and of its cumulative distribution function can be quite different. Our new strategy is designed to achieve the fastest convergence on the latter quantity, which is ultimately the observable quantity of an experiment.

The paper is organized as follows. In Section 2 we define the mathematical methods used for the construction of the interpolant and show our adaptive strategy to choose a new collocation points. In Section 3 we present some numerical examples and comparisons with the Clenshaw-Curtis collocation method, and the adaptive method based on hierarchical surplus. Finally, we draw our conclusions in Section 4.

2 Mathematical methods

2.1 Clenshaw-Curtis (CC) quadrature rule

In Section 3, we compare our method with the CC method, which is the standard appropriate collocation method for a uniform P_X . Here, we recall the basic properties of CC, for completeness. The Clenshaw-Curtis (CC) quadrature rule uses the extrema of a Chebyshev polynomial (the so-called ‘extrema plus end-points’ collocation points in [33]) as abscissas. They are particularly appealing to be used as collocation points in UQ, because a certain subset of them are nested. Specifically, they are defined, in the interval $[-1, 1]$ as:

$$x_i = -\cos\left(\frac{\pi(i-1)}{N-1}\right) \quad i = 1, \dots, N. \quad (2)$$

One can notice that the the set of $N = 2^w + 1$ points is fully contained in the set of $N = 2^{w+1} + 1$ points (with w an arbitrary integer, referred to as the level of the set). In practice this means that one can construct a nested sequence of collocation points with $N = 3, 5, 9, 17, 33, 65, 129, \dots$, re-using all the previous evaluations of g .

Collocation points based on quadratures are optimal to calculate moments ¹:

$$\mu_Y^p = \int y^p P_Y(y) dy = \int g(x)^p P_X(x) dx, \quad (3)$$

where we used the identity relation,

$$P_Y(y) dy = P_X(x) dx. \quad (4)$$

It is known that integration by quadrature is very accurate (for smooth enough integrand), and the moments can be readily evaluated, without the need to construct

¹ Here p on the left-hand side is a label, such that μ^1 is the mean, μ^2 is the variance, and so on. On the right-hand side it is an exponent.

an interpolant:

$$\mu_Y^p \simeq \sum_i w_i (g(x_i))^p, \quad (5)$$

where the weights w_i can be computed with standard techniques (see, e.g. [27]). The interpolant for the CC method is the Lagrange polynomial.

2.2 Selection of collocation points based on hierarchical surplus

The hierarchical surplus algorithm is widely used for interpolation on sparse grids. It is generally defined as the difference between the value of an interpolant at the current and previous interpolation levels [29]. The simplest algorithm prescribes a certain tolerance and looks for all the point at the new level where the hierarchical surplus is larger than the tolerance. The new sampling points (at the next level) will be the neighbors of the points where this condition is met. In one-dimension, the algorithm is extremely simple because the neighbors are only two points, that one can define in such a way that cells are always halved. In this work, we compare our new method with a slightly improved version of the hierarchical surplus algorithm. The reason is because we do not want our comparisons to be dependent on the choice of an arbitrary tolerance level, and we want to be able to add new points two at the time. Hence, we define a new interpolation level by adding only the two neighbors of the point with the largest hierarchical surplus. All the previous hierarchical surpluses that have been calculated, but for which new points have not been added yet are kept. The pseudo-code of the algorithm follows. The interpolant is understood to be piece-wise linear interpolation, and the grid is $x \in [-1, 1]$.

Algorithm 1 Hierarchical surplus algorithm

-
- 1: Calculate the interpolant on the grid $x = \{-1, 0, 1\}$.
 - 2: Define $x_h = \{-1/2, 1/2\}$ and add them on the grid
 - 3: **while** Not converged **do**
 - 4: Calculate the interpolant on the new grid
 - 5: Calculate the hierarchical surplus on the last two entries of x_h and store them in the vector h_s
 - 6: Find the largest hierarchical surplus in h_s , remove it from h_s and remove the corresponding x from x_h
 - 7: Append the two neighbors to x_h and add them to the grid
 - 8: **end while**
-

2.3 Multiquadric biharmonic radial basis

We use a multiquadric biharmonic radial basis function (RBF) with respect to a set of points $\{x_i\}$, with $i = 1, \dots, N$, defined as:

$$\Phi_i(x, c) = \sqrt{(x - x_i)^2 + c_i^2}, \quad (6)$$

where c_i are free parameters (referred to as shape parameters). The function $g(x)$ is approximated by the interpolant $\tilde{g}(x)$ defined as

$$\tilde{g}(x) = \sum_{i=1}^N \lambda_i \Phi_i(x, c). \quad (7)$$

The weights λ_i are obtained by imposing that $g(x_i) = \tilde{g}(x_i)$ for each sampling point in the set, namely the interpolation error is null at the sampling points. This results in solving a linear system for $\lambda = (\lambda_1, \dots, \lambda_N)$ of the form $A\lambda^T = g(\mathbf{x})^T$, with A a real symmetric $N \times N$ matrix. We note that, by construction, the linear system will become more and more ill-conditioned with increasing N , for fixed values of c . This can be easily understood because when two points become closer and closer the corresponding two rows in the matrix A become less and less linearly independent. To overcome this problem one needs to decrease the corresponding values of c . In turns, this means that the interpolant $\tilde{g}(x)$ will tend to a piece-wise linear interpolant for increasingly large N .

2.4 New adaptive selection of collocation points

We focus, as the main diagnostic of our method, on the cumulative distribution function (cdf) $C(y)$, which is defined as

$$C(y) = \int_{y_{\min}}^y P_Y(y) dy, \quad (8)$$

where $y_{\min} = \min g(x)$. As it is well known, the interpretation of the cumulative distribution function is that, for a given value y^* , $C(y^*)$ is the probability that $g(x) \leq y^*$. Of course, the cdf $C(y)$ contains all the statistical information needed

to calculate any moment of the distribution, and can return the probability density function $P_Y(y)$, upon differentiation. Moreover, the cdf is always well defined between 0 and 1. The following two straightforward considerations will guide the design of our *adaptive selection strategy*. A first crucial point, already evident from Eq. (1), is whether or not $g(x)$ is bijective. When $g(x)$ is bijective this translates to the cdf $C(y)$ being continuous, while a non-bijective function $g(x)$ produces a cdf $C(y)$ which is discontinuous. It follows that intervals in x where $g(x)$ is constant (or nearly constant) will map into a single value $y = g(x)$ (or a very small interval in y) where the cdf will be discontinuous (or ‘nearly’ discontinuous). Secondly, an interval in x with a large first derivative of $g(x)$ will produce a nearly flat cdf $C(y)$. This is again clear by noticing that the Jacobian J in Eq. (1) ($dg(x)/dx$ in one dimension) is in the denominator, and therefore the corresponding $P_Y(y)$ will be very small, resulting in a flat cdf $C(y)$.

Loosely speaking one can then state that regions where $g(x)$ is flat will produce large jumps in the cdf $C(y)$ and, conversely, regions where the $g(x)$ has large jumps will map in to a nearly flat cdf $C(y)$. From this simple considerations one can appreciate how important it is to have an interpolant that accurately capture both regions with *very large* and *very small* first derivative of $g(x)$. Moreover, since the cdf $C(y)$ is an integrated quantity, interpolation errors committed around a given y will propagate in the cdf for all larger y values. For this reason, it is important to achieve a global convergence with interpolation errors that are of the same order of magnitude along the whole domain.

The adaptive section algorithm works as follows. We work in the interval $x \in [-1, 1]$ (every other interval where the support of $g(x)$ is defined can be rescaled to this interval). We denote with $\{x_i\}$ the sampling set which we assume is always sorted, such that $x_i < x_{i+1}$. We start with 3 points: $x_1 = -1$, $x_2 = 0$, $x_3 = 1$. For the robustness and the simplicity of the implementation we choose to select a new sampling point always at equal distance between two existing points. One can decide to limit the ratio between the largest and smallest distance between adjacent points: $r = \max\{d_i\}/\min\{d_i\}$ (with $i = 1, \dots, N - 1$), where d_i is the distance between the points x_{i+1} and x_i . This avoids to keep refining small intervals when large intervals might still be under-resolved, thus aiming for the above mentioned global convergence over the whole support. At each iteration we create a list of possible new points, by halving every interval, excluding the points that would increase the value of r above the maximum desired (note that r will always be a power of 2). We calculate the first derivative of $\tilde{g}(x)$ at these points, and alternatively choose the point with largest/smallest derivative as the next sampling point. Notice that, by the definition of the interpolant, Eq. (7), its first derivative can be calculated exactly as:

$$\frac{d\tilde{g}(x)}{dx} = \sum_{i=1}^N \lambda_i \frac{d\Phi_i(x, c)}{dx} \quad (9)$$

without having to recompute the weights λ_i . At each iteration the shape parameters c_i are defined at each points, as $c_i = 0.85 \cdot \min(d_{i-1}, d_i)$, i.e. they are linearly rescaled with the smallest distance between the point x_i and its neighbors. The pseudo-code of the algorithm follows.

Algorithm 2 Adaptive selection of sampling points

```

1: while Not converged do
2:    $x_{\text{guess}} \leftarrow 0.5 \cdot (x_i + x_{i+1})$ 
3:   Exclude points in  $x_{\text{guess}}$  such that  $r = \max\{d_i\} / \min\{d_i\} > R$ 
4:   Calculate  $\tilde{g}^n(x)'$  through (9) at  $\{x_{\text{guess}}\}$ 
5:   Alternatively choose  $x_{\text{guess}}$  with largest/smallest values of  $|\tilde{g}^n(x)'|$  as new collocation
   point
6:   Calculate new weights  $\lambda_i$ 
7: end while

```

3 Numerical examples

In this section we present and discuss four numerical examples where we apply our adaptive selection strategy. In this work we focus on a single input parameter and the case of constant probability $P_X = 1/2$ in the interval $x \in [-1, 1]$, and we compare our results against the Clenshaw-Curtis, and the hierarchical surplus methods. We denote with $\tilde{g}^n(x)$ the interpolant obtained with a set of n points (hence the iterative procedure starts with $\tilde{g}^3(x)$). A possible way to construct the cdf $C(y)$ from a given interpolant $\tilde{g}^n(x)$ would be to generate a sample of points in the domain $[-1, 1]$, randomly distributed according to the pdf $P_X(x)$, collecting the corresponding values calculated through Eq. (7), and constructing their cdf. Because here we work with a constant $P_X(x)$, it is more efficient to simply define a uniform grid in the domain $[-1, 1]$ where to compute $\tilde{g}^n(x)$. In the following we will use, in the evaluation of the cdf $C(y)$, a grid in y with $N_y = 10001$ points equally spaced in the interval $[\min \tilde{g}^n(x), \max \tilde{g}^n(x)]$, and a grid in x with $N_x = 1001$ points equally spaced in the interval $[-1, 1]$. We define the following errors:

$$\varepsilon_C = \frac{\|C(\tilde{g}^n(x)) - C(g(x))\|_2}{\sqrt{N_y}} \quad (10)$$

$$\varepsilon_g = \frac{\|\tilde{g}^n(x) - g(x)\|_2}{\sqrt{N_x}} \quad (11)$$

where $\|\cdot\|_2$ denotes the L_2 norm. It is important to realize that the accuracy of the numerically evaluated cdf $C(y)$ will always depend on the binning of y , i.e. the points at which the cdf is evaluated. As we will see in the following examples, the error ε_C saturates for large N , which thus is an artifact of the finite bin size. We emphasize that, differently from most of the previous literature, our strategy focuses on converging rapidly in ε_C , rather than in ε_g . Of course, a more accurate interpolant will always result in a more accurate cdf, however the relationship between a reduction in ε_g and a corresponding reduction in ε_C is not at all trivial. This is because the relation between $P_X(x)$ and $P_Y(y)$ is mediated by the Jacobian of $g(x)$, and it also involves the bijectivity of g .

Finally, we study the convergence of the mean μ_Y , see equation 3, and the variance σ_Y^2 , which is defined as

$$\sigma_Y^2 = \int_{-1}^1 (\tilde{g}(x) - \mu_Y)^2 P_X(x) dx. \quad (12)$$

These will be calculated by quadrature for the CC methods, and with an integration via trapezoidal method for the adaptive methods.

We study two analytical test cases:

- Case 1: $g(x) = \arctan(10^3 x^3)$;
- Case 2: $g(x) = \frac{1}{(2 + \sin(3\pi x))^2}$;

and two test cases where an analytical solution is not available, and the reference $g(x)$ will be calculated as an accurate numerical solution of a set of ordinary differential equations:

- Case 3: Lotka-Volterra model (predator-prey);
- Case 4: Van der Pol oscillator.

While Case 1 and 2 are more favorable to the CC method, because the functions are smooth and analytical, hence a polynomial interpolation is expected to produce accurate results, the latter two cases mimic applications of real interest, where the model does not produce analytical results, although $g(x)$ might still be smooth (at least piece-wise, in Case 4).

3.1 Case 1: $g(x) = \arctan(10^3 x^3)$

In this case $g(x)$ is a bijective function, with one point ($x = 0$) where the first derivative vanishes. Figure 2 shows the function $g(x)$ (top panel) and the corresponding cdf $C(y)$ (bottom panel), which in this case can be derived analytically. Hence, we use the analytical expression of cdf $C(y)$ to evaluate the error ε_C . The convergence of ε_C and ε_g is shown in Figure 3 (top and bottom panels, respectively). Here and in all the following figures blue squares denote the new adaptive selection method, red dots are for the CC methods, and black line is for the hierarchical surplus method. We have run the CC method only for $N = 3, 5, 9, 17, 33, 65, 129$ (i.e. the points at which the collocation points are nested), but for a better graphical visualization the red dots are connected with straight lines. One can notice that the error for the new adaptive method is consistently smaller than for the CC method. From the top panel, one can appreciate the saving in computer power that can be achieved with our new method. Although the difference with CC is not very large until $N = 17$, at $N = 33$ there is an order of magnitude difference between the two. It effectively means that in order to achieve the same error $\varepsilon_C \sim 10^{-5}$, the CC method would run at least twice the number of simulations. The importance of focusing on the convergence of the cdf, rather than on the interpolant, is clear in comparing our method with the hierarchical surplus method. For instance, for $N = 80$, the two methods have a comparable error ε_g , but our method has achieved almost an order of magnitude more accurate solution in $C(y)$. Effectively, this means that our method has sampled the new points less redundantly. In this case $g(x)$ is an anti-symmetric function with zero mean. Hence, any method that chooses sampling points symmetrically distributed around zero would produce the correct first moment μ_Y . We show in figure 4 the convergence of σ_Y^2 , as the absolute value of the different with the exact value σ_{an} , in logarithmic scale. Blue, red, and black lines represent the new adaptive method, the CC, and the hierarchical surplus methods, respectively (where again for the CC, simulations are only performed where the red dots are shown). The exact value is $\sigma_{an}^2 = 2.102$. As we mentioned, the CC method is optimal to calculate moments, since it uses quadrature. Although in our method the error does not decrease monotonically, it is comparable with the result for CC.

3.2 Case 2: $g(x) = \frac{1}{(2+\sin(3\pi x))^2}$

In this case the function $g(x)$ is periodic, and it presents, in the domain $x \in [-1, 1]$ three local minima ($y = 1/9$) and three local maxima ($y = 1$). The function and the cdf $C(y)$ are shown in Figure 5 (top and bottom panel, respectively). Figure 6 shows the error for this case (from now on the same format of Figure 3 will be used). The first consideration is that the hierarchical surplus method is the less accurate of the three. Second, ε_g is essentially the same for the CC and the new method, up to $N = 65$. For $N = 129$ the CC methods achieve a much accurate solution as compared to the new adaptive method, whose error has a much slower convergence. However, looking at the error in the cdf in top panel of Figure 6, the two methods are essentially equivalent. This example demonstrates that, in an UQ framework, the primary goal in constructing a good interpolant should not be to minimize the error of the interpolant with respect to the 'true' $g(x)$, but rather to achieve the fastest possible convergence on the cdf C_Y . Although, the two effects are intuitively correlated, they are not into a linear relationship. In other words, not all sample points in x count equally in minimizing ε_C . The convergence of μ_Y (exact value $\mu_{an} = 0.385$) and σ_Y^2 (exact value $\sigma_{an} = 0.087$) is shown in Figures 7 and 8, respectively. It is interesting to notice that our method presents errors that are always smaller than the CC method, although the errors degrade considerably in the regions between two CC points, where the two adaptive methods yield comparable results.

3.3 Case 3: Lotka-Volterra model (predator-prey)

The Lotka-Volterra model [34–36] is a well-studied model that exemplifies the interaction between two populations (predators and preys). This case is more realistic than Cases 1 and 2, as the solution of the model cannot be written in analytical form. As such, both the $g(x)$ and the cdf $C(y)$ used to compute the errors are calculated numerically. We use the following simple model:

$$\frac{dh(t)}{dt} = h(t) - (5x + 6)h(t)l(t) \quad (13)$$

$$\frac{dl(t)}{dt} = h(t)l(t) - l(t) \quad (14)$$

where $h(t)$ and $l(t)$ denote the population size for each species (say, horses and lions) as function of time. The ODE is easily solved in MATLAB, with the `ode45` routine, with an absolute tolerance set equal to 10^{-8} . We use, as initial conditions, $h(t = 0) = l(t = 0) = 1$, and we solve the equations for $t \in [0, 10]$. Clearly, the solution of the model depends on the input parameter x . We define our test function $g(x)$ to be the result of the model for the l population at time $t = 10$:

$$g(x) = l(t = 10, x). \quad (15)$$

The resulting function $g(x)$, and the computed cdf $C(y)$ are shown in Figure 9 (top and bottom panel, respectively). We note that, although $g(x)$ cannot be expressed as an analytical function, it is still smooth, and hence it does not present particular difficulties in being approximated through a polynomial interpolant. Indeed the

error ε_g undergoes a fast convergence both for the adaptive methods and for the CC method (Figure 10). Once again, the new adaptive method is much more powerful than the CC method in achieving a better convergence rate, and thus saving computational power, while the hierarchical surplus method is the worst of the three. Convergence of μ_Y and σ_Y^2 are shown in Figures 11 and 12, respectively. Similar to previous cases, the CC presents a monotonic convergence, while this is not the case for the adaptive methods. Only for $N = 129$, the CC method yields much better results than the new method.

3.4 Case 4: Van der Pol oscillator

Our last example is the celebrated Van der Pol oscillator[37, 7, 38, 39], which has been extensively studied as a textbook case of a nonlinear dynamical system. In this respect this test case is very relevant to Uncertainty Quantification, since real systems often exhibit a high degree of nonlinearity. Similar to Case 3, we define our test function $g(x)$ as the output of a set of two ODEs, which we solve numerically with MATLAB. The model for the Van der Pol oscillator is:

$$\frac{dQ(t)}{dt} = V(t) \quad (16)$$

$$\frac{dV(t)}{dt} = (-50 + 100(x + 2))(1 - Q(t)^2)V(t) - Q(t). \quad (17)$$

The initial conditions are $Q(t = 0) = 2$, $V(t = 0) = 0$. The model is solved for time $t \in [0, 300]$, and the function $g(x)$ is defined as

$$g(x) = V(t = 300, x). \quad (18)$$

The so-called nonlinear damping parameter is rescaled such that for $x \in [-1, 1]$, it ranges between 50 and 250. The function $g(x)$ and the corresponding cdf $C(y)$ are shown in Figure 13. This function is clearly much more challenging than the previous ones. It is divided in two branches, where it takes values $-2 \leq y \leq -1$ and $1 \leq y \leq 2$, and it presents discontinuities where it jumps from one branch to the other. Correspondingly, cdf $C(y)$ presents a flat plateau for $-1 \leq y \leq 1$, which is the major challenge for both methods. In figure 14 we show the errors ε_g and ε_C . The overall convergence rate of the CC and the new method is similar. For this case, the hierarchical surplus method yields a better convergence, but only for $N > 80$. As we commented before, the mean μ_Y has no statistical meaning in this case, because the output is divided into two separate regions. The convergence for σ_Y^2 is presented in Figure 15.

4 Conclusions and future work

We have presented a new adaptive algorithm for the selection of sampling points for non-intrusive stochastic collocation in Uncertainty Quantification (UQ). The main idea is to use a radial basis function as interpolant, and to refine the grid on points where the interpolant presents large and small first derivative.

In this work we have focused on 1D and uniform probability $P_X(x)$, and we have

shown four test cases, encompassing analytical and non-analytical smooth functions, which are prototype of a very wide class of functions. In all cases the new adaptive method improved the efficiency of both the (non-adaptive) Clenshaw-Curtis collocation method, and of the adaptive algorithm based on the calculation of the hierarchical surplus (note that the method used in this paper is a slight improvement of the classical algorithm). The strength of our method is the ability to select a new sampling point making full use of the interpolant resulting from all the previous evaluation of the function $g(x)$, thus seeking the most optimal convergence rate for the cdf $C(y)$. We have shown that there is no one-to-one correspondence between a reduction in the interpolation error ε_g and a reduction in the cdf error ε_C . For this reason, collocation methods that choose the distribution of sampling points a priori can perform poorly in attaining a fast convergence rate in ε_C , which is the main goal of UQ. Moreover, in order to maintain the nestedness of the collocation points the CC method requires larger and larger number of simulations (2^w moving from level w to level $w + 1$), which is in contrast with our new method where one can add one point at the time.

We envision many possible research directions to further investigate our method. The most obvious is to study multi-dimensional problems. We emphasize that the radial basis function is a mesh-free method and as such we anticipate that this will largely alleviate the curse of dimensionality that afflicts other collocation methods based on quadrature points (however, see [40] for methods related to the construction of sparse grids, which have the same aim). Moreover, it will be interesting to explore the versatility of RBF in what concerns the possibility of choosing an optimal shape parameter c [41]. Recent work [42,43] investigated the role of the shape parameter c in interpolating discontinuous functions, which might be very relevant in the context of UQ, when the continuity of $g(x)$ cannot be assumed a priori. Finally, a very appealing research direction, would be to simultaneously exploit quasi-Monte Carlo and adaptive selection methods for extremely large dimension problems.

5 Acknowledgments

A. A. and C. R. are supported by FOM Project No. 12CSER058 and 12PR304, respectively. We would like to remember dr.ir. J.A.S. Witteveen († 2015) for the useful discussions we had about uncertainty quantification.

References

1. D. Xiu, Communications in computational physics **5**(2-4), 242 (2009)
2. M. Eldred, J. Burkardt, AIAA paper **976**(2009), 1 (2009)
3. G. Onorato, G. Loeven, G. Ghorbaniasl, H. Bijl, C. Lacor, in *V European Conference on Computational Fluid Dynamics ECCOMAS, Lisbon, Portugal* (2010), pp. 14–17
4. D. Xiu, G.E. Karniadakis, SIAM journal on scientific computing **24**(2), 619 (2002)
5. T. Crestaux, O. Le Maître, J.M. Martinez, Reliability Engineering & System Safety **94**(7), 1161 (2009)
6. K. Togawa, A. Benigni, A. Monti, in *Proceedings of the 2011 Grand Challenges on Modeling and Simulation Conference* (Society for Modeling & Simulation International, Vista, CA, 2011), GCMS '11, pp. 30–35. URL <http://dl.acm.org/citation.cfm?id=2348229.2348235>

7. M.S. Eldred, AIAA Paper **2274**, 2009 (2009)
8. I.B.I.R. Tempone, G.Zouraris, SIAM Journal of Numerical Analysis **42(2)**, 800 (2004)
9. M. Grigoriu, *Stochastic systems: uncertainty quantification and propagation* (Springer Science & Business Media, 2012)
10. D. Xiu, *Numerical methods for stochastic computations: a spectral method approach* (Princeton University Press, 2010)
11. O. Le Maître, O.M. Knio, *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics* (Springer Science & Business Media, 2010)
12. R.E. Caflisch, Acta Numerica **7**, 1 (1998)
13. M.H. Kalos, P.A. Whitlock, *Monte carlo methods* (John Wiley & Sons, 2008)
14. P. L'Ecuyer, A.B. Owen, *Monte Carlo and Quasi-Monte Carlo Methods 2008* (Springer, 2009)
15. H.G. Niederreiter, Bull. Amer. Math. Soc. **84(6)**, 957 (1978)
16. M.H. Kalos, P.A. Whitlock, Monte Carlo Methods, Second Edition pp. 179–197 (2008)
17. R. Bellman, Courier Dover Publications (2003)
18. P. Indyk, R. Motwani, in *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (ACM, 1998), pp. 604–613
19. F.Y. Kuo, I.H. Sloan, Notices of the AMS **52(11)**, 1320 (2005)
20. D. Xiu, J.S. Hesthaven, SIAM J. Sci. Comput. **28**, 1167 (2006)
21. J. Foo, G.E. Karniadakis, J. Comput. Phys. **229(5)**, 1536 (2010). DOI 10.1016/j.jcp.2009.10.043. URL <http://dx.doi.org/10.1016/j.jcp.2009.10.043>
22. I. Babuška, F. Nobile, R. Tempone, SIAM Journal on Numerical Analysis **45(3)**, 1005 (2007)
23. P.D. Laurie, J. Comp. Appl. Math. **127(1-2)**, 201 (2001)
24. W. Gautschi, Mathematics of Computation **24(110)**, 245 (1970)
25. J. Waldvogel, BIT Numerical Mathematics **46(1)**, 195 (2006)
26. J.P. Berrut, L.N. Trefethen, SIAM Rev. **46(3)**, 501 (2004)
27. L.N. Trefethen, SIAM: Society for Industrial and Applied Mathematics (2000)
28. J. Epperson, Amer. Math. Monthly **94**, 329 (1987)
29. X. Ma, N. Zabarar, Journal of Computational Physics **228(8)**, 3084 (2009)
30. X. Ma, N. Zabarar, Inverse Problems **25(3)**, 035013 (2009)
31. J.D. Jakeman, S.G. Roberts, in *Sparse grids and applications* (Springer, 2012), pp. 181–203
32. J.A. Witteveen, G. Iaccarino, SIAM Journal on Scientific Computing **34(3)**, A1522 (2012)
33. J.P. Boyd, Dover, New York (2001)
34. F. Baruer, C. Castillo-Chavez, Springer-Verlag (2000)
35. M.H.H. Stevens, in *A Primer of Ecology with R* (Springer, 2009), pp. 135–159
36. P.J. Wangersky, Annual Review of Ecology and Systematics pp. 189–218 (1978)
37. M.B.D.J. Geer, C.M. Andersen, SIAM Journal on Applied Mathematics **44(5)**, 881 (1984)
38. R. Yuan, X. Wang, Y. Ma, B. Yuan, P. Ao, Phys. Rev. E **87**, 062109 (2013). DOI 10.1103/PhysRevE.87.062109. URL <http://link.aps.org/doi/10.1103/PhysRevE.87.062109>
39. L. Liu, E.H. Dowell, K.C. Hall, International Journal of Non-Linear Mechanics **42(1)**, 2 (2007)
40. F.N.R. Tempone, C.G. Webster, SIAM J. Numer. Anal. **46(5)**, 2309 (2008)
41. G.E. Fasshauer, J.G. Zhang, Numer. Algor. **45**, 345 (2007)
42. J.H. Jung, V.R. Durante, Applied Numerical Mathematics **59(7)**, 1449 (2009)
43. J. Wang, G. Liu, Computer methods in applied mechanics and engineering **191(23)**, 2611 (2002)

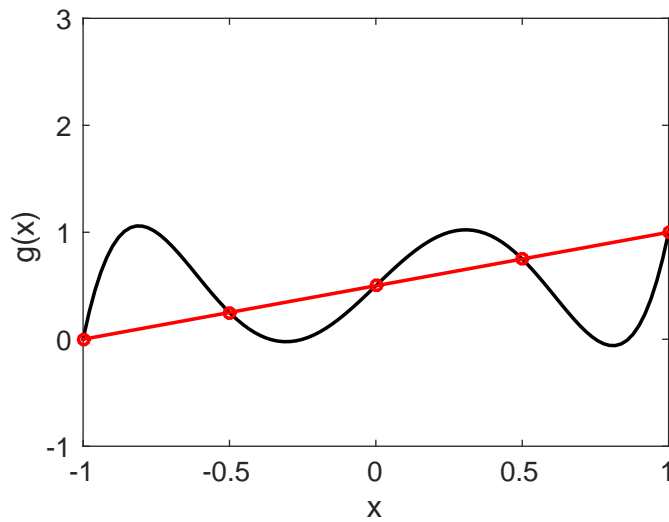


Fig. 1 Example for which the algorithm based on hierarchical surplus fails. The function $g(x) = \frac{256}{30}x^5 - \frac{32}{3}x^3 + \frac{79}{30}x + \frac{1}{2}$ (in black) goes exactly through the red straight line at the points $x = -1, -0.5, 0, 0.5, 1$. Calculating the piece-wise linear interpolant between two ($x = -1, 1$), three ($x = -1, 0, 1$), and five ($x = -1, -\frac{1}{2}, 0, \frac{1}{2}, 1$) points would result in a null hierarchical surplus on these points.

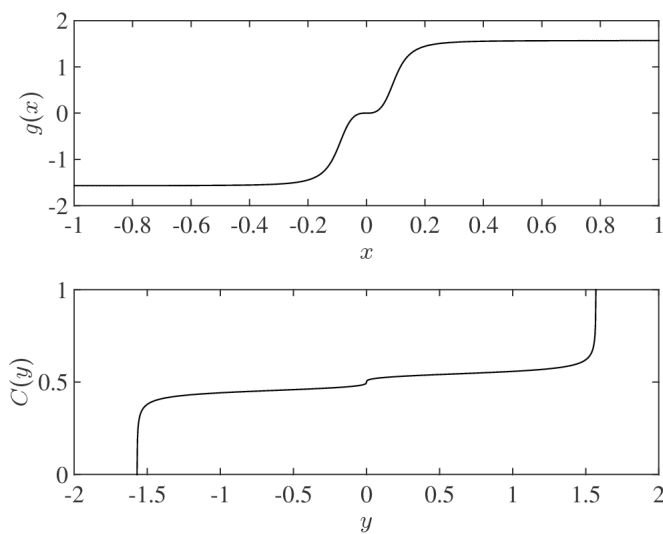


Fig. 2 Case 1: $g(x) = \arctan(10^3 x^3)$. Top panel: $g(x)$; bottom panel: cdf $C(y)$.

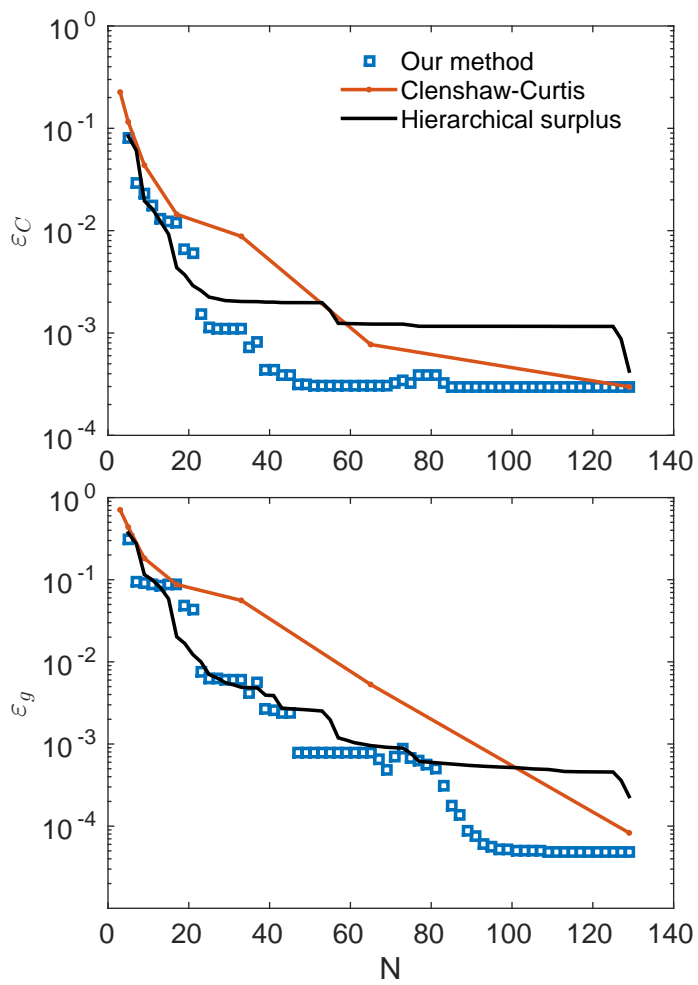


Fig. 3 Case 1. Error ε_C (top) ε_g (bottom) as function of number of sampling points N . Blue squares: new adaptive selection method. Red dots: Clenshaw-Curtis. Black curve: adaptive method based on hierarchical surplus.

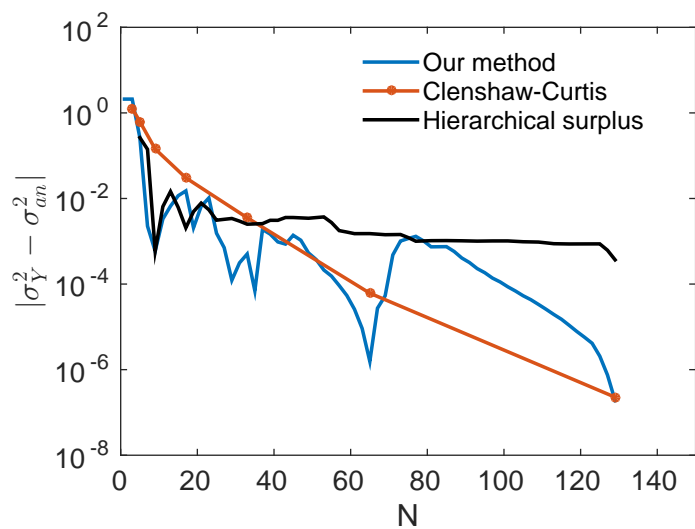


Fig. 4 Case 1. Absolute error in the variance σ_Y^2 versus number of sampling points N . Blue: new adaptive selection method. Red: Clenshaw-Curtis. Black: adaptive method based on hierarchical surplus.

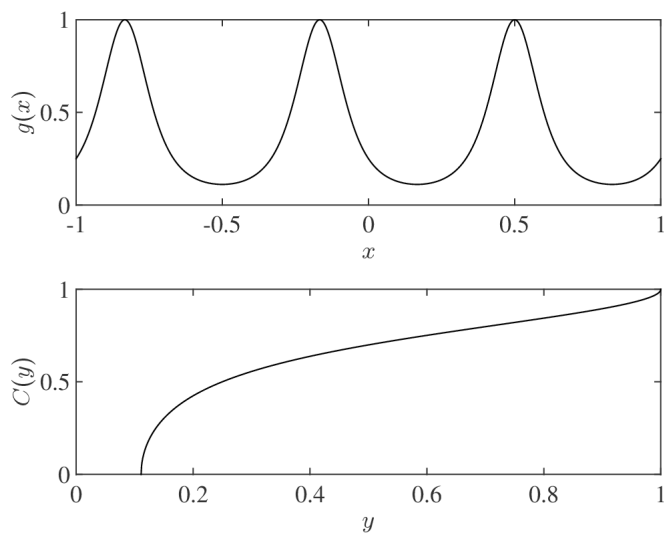


Fig. 5 Case 2: $g(x) = \frac{1}{(2+\sin(3\pi x))^2}$. Top panel: $g(x)$; bottom panel: $C(y)$.

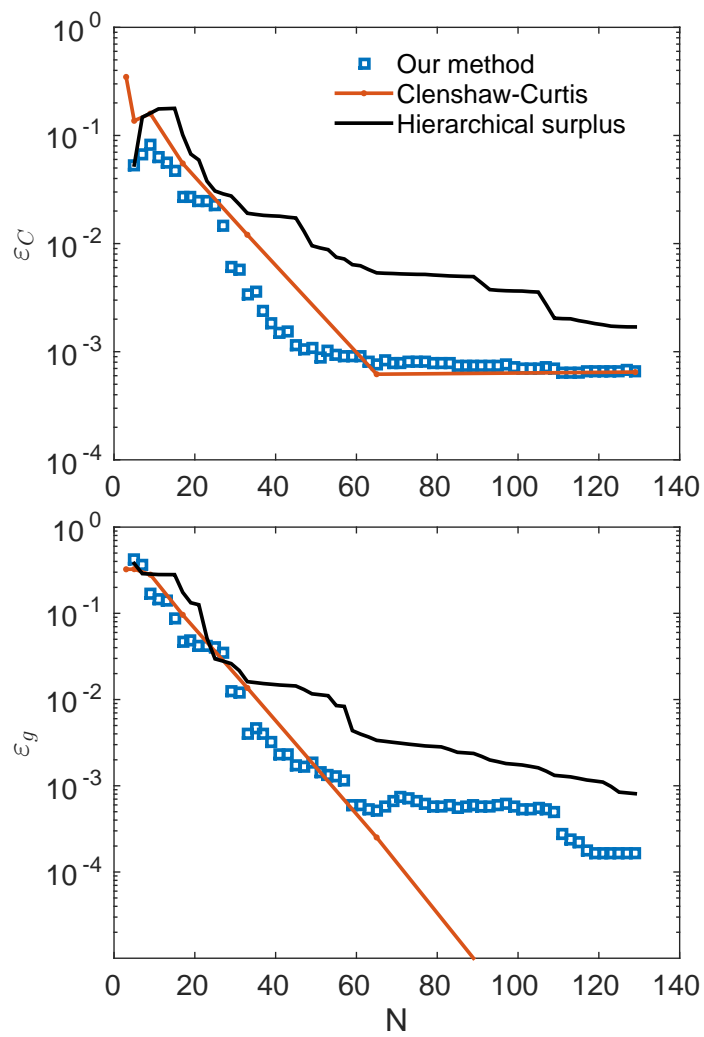


Fig. 6 Case 2. Error ε_C (top) ε_g (bottom) as function of number of sampling points N . Blue squares: new adaptive selection method. Red dots: Clenshaw-Curtis. Black curve: adaptive method based on hierarchical surplus.

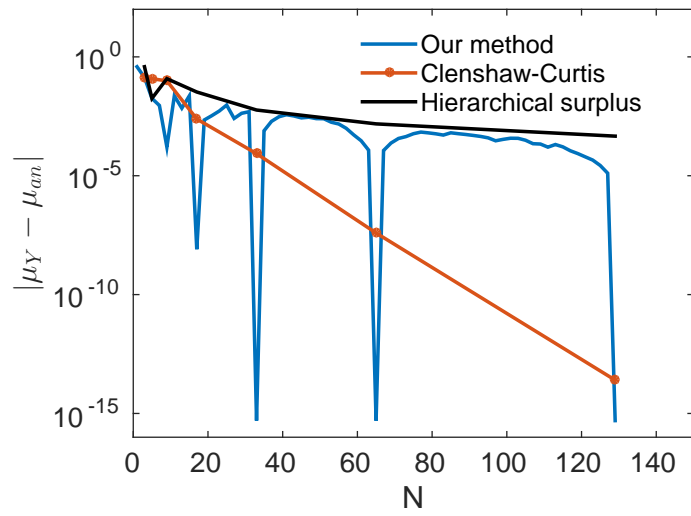


Fig. 7 Case 2. Absolute error in the mean μ_Y versus number of sampling points N . Blue: new adaptive selection method. Red: Clenshaw-Curtis. Black: adaptive method based on hierarchical surplus.

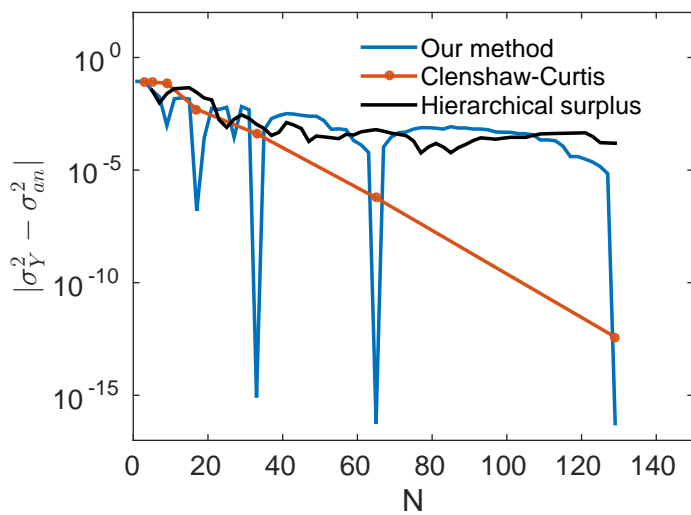


Fig. 8 Case 2. Absolute error in the variance σ_Y^2 versus number of sampling points N . Blue: new adaptive selection method. Red: Clenshaw-Curtis. Black: adaptive method based on hierarchical surplus.

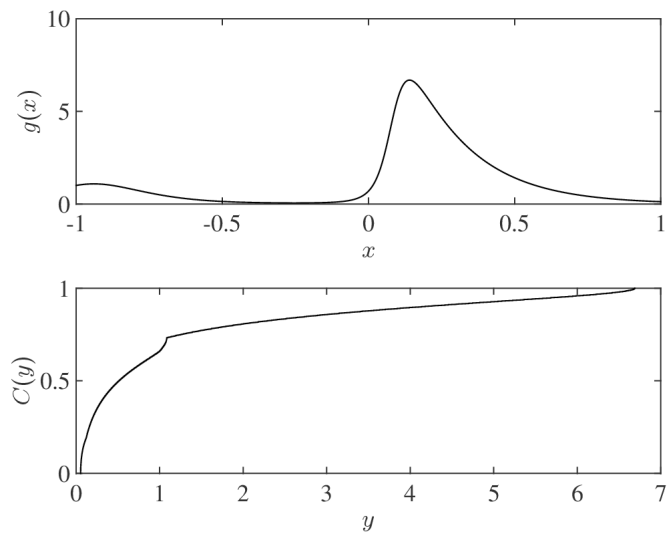


Fig. 9 Case 3: Lotka-Volterra model. Top panel: $g(x)$; bottom panel: $C(y)$.

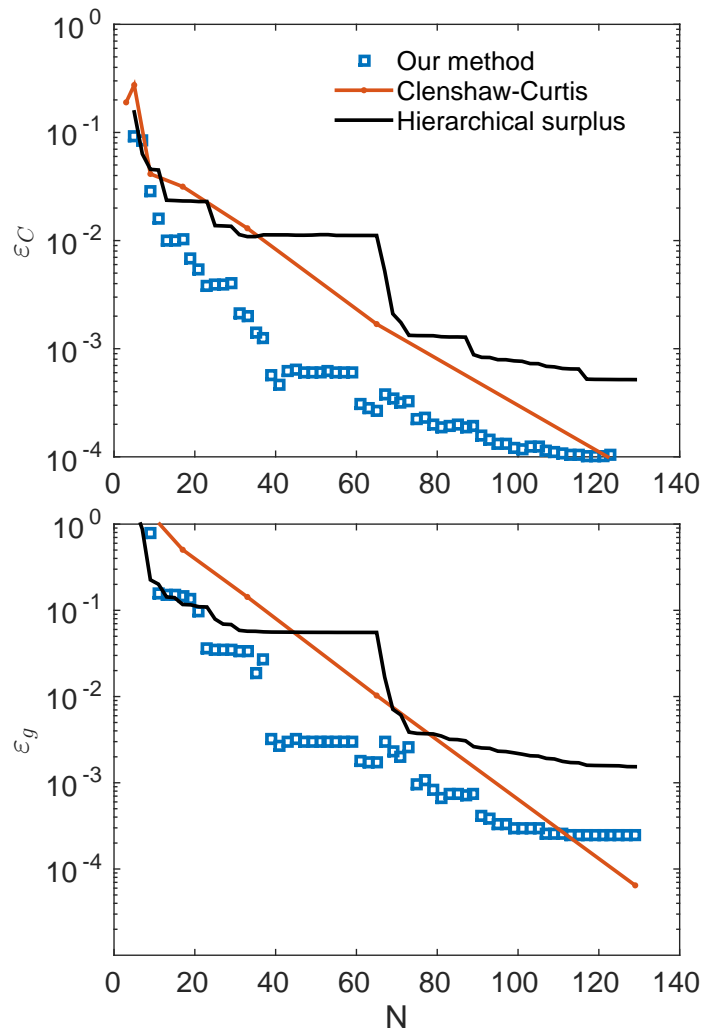


Fig. 10 Case 3. Error ε_C (top) ε_g (bottom) as function of number of sampling points N . Blue squares: new adaptive selection method. Red dots: Clenshaw-Curtis. Black curve: adaptive method based on hierarchical surplus.

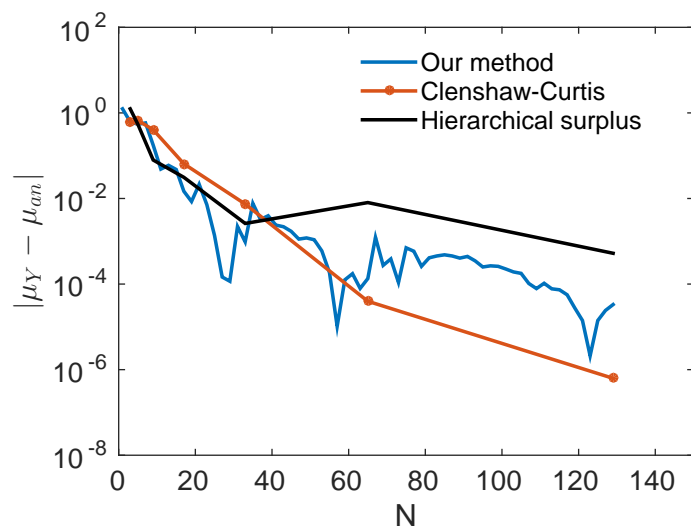


Fig. 11 Case 3. Absolute error in the mean μ_Y versus number of sampling points N . Blue: new adaptive selection method. Red: Clenshaw-Curtis. Black: adaptive method based on hierarchical surplus.

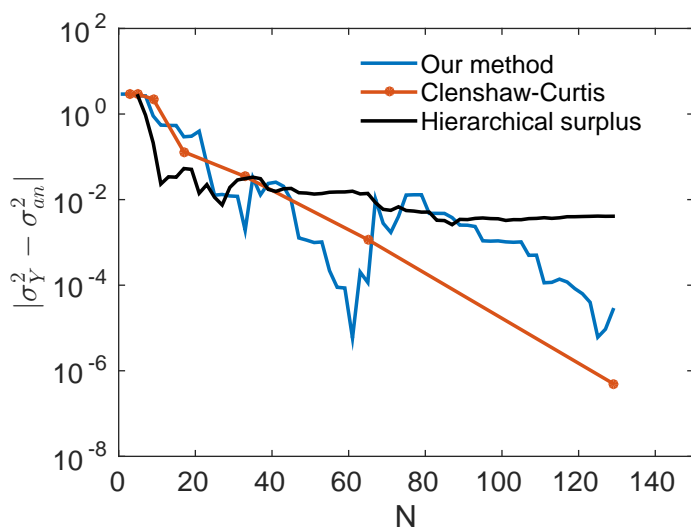


Fig. 12 Case 3. Absolute error in the variance σ_Y^2 versus number of sampling points N . Blue: new adaptive selection method. Red: Clenshaw-Curtis. Black: adaptive method based on hierarchical surplus.

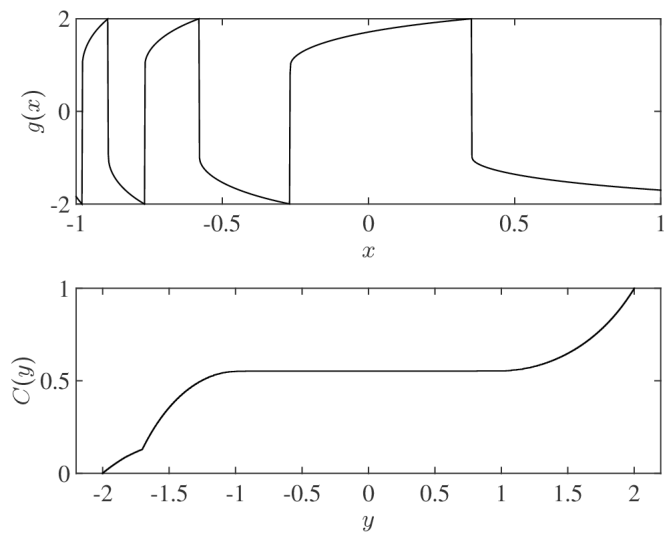


Fig. 13 Case 4: Van der Pol oscillator. Top panel: $g(x)$; bottom panel: $C(y)$.

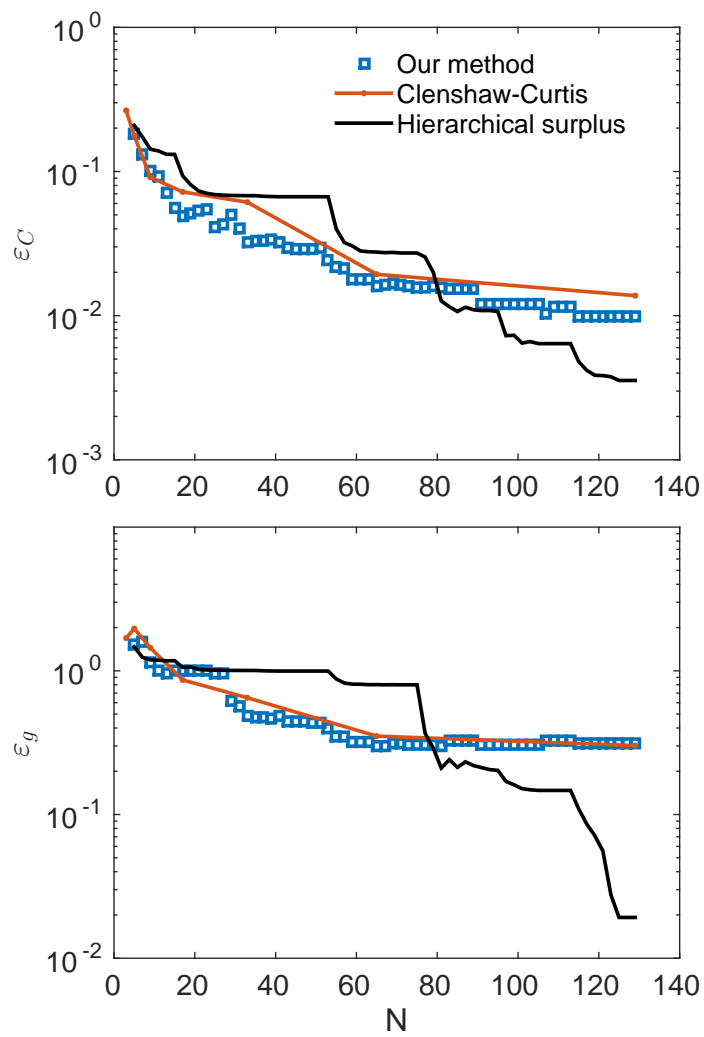


Fig. 14 Case 4. Error ε_C (top) ε_g (bottom) as function of number of sampling points N . Blue squares: new adaptive selection method. Red dots: Clenshaw-Curtis. Black curve: adaptive method based on hierarchical surplus.

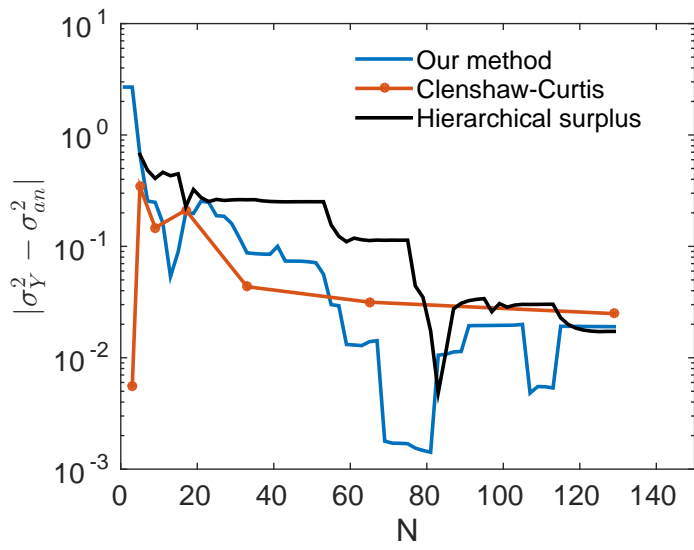


Fig. 15 Case 4. Absolute error in the variance σ_Y^2 versus number of sampling points N . Blue: new adaptive selection method. Red: Clenshaw-Curtis. Black: adaptive method based on hierarchical surplus.