

A COMPARATIVE STUDY BETWEEN KRIGING AND ADAPTIVE SPARSE TENSOR-PRODUCT METHODS FOR MULTI-DIMENSIONAL APPROXIMATION PROBLEMS IN AERODYNAMICS DESIGN *

ABDELLAH CHKIFA¹, ALBERT COHEN¹, PIERRE-YVES PASSAGGIA² AND JACQUES PETER²

Abstract. The performances of two multivariate interpolation procedures are compared using functions that are either synthetic or coming from a shape optimization problem in aerodynamics. The aim is to evaluate the efficiency of adaptive sparse interpolation algorithms [2] and compare them with the kriging approach developed for the design and analysis of computer experiment (DACE) [21]. The accuracy and computational time of the two methods are examined as the number N of samples used in the interpolation increases. It appears in our test cases that both methods perform equivalently, in terms of precision. However, as the dimension d increases, the computational time involved in the enrichment of the kriging sample becomes intractable for large values of N . This problem is circumvented in the case of the sparse interpolation procedure for which the computational time scales linearly with N and d .

Résumé. Nous comparons les performances de deux méthodes d'interpolation en grande dimension, aussi bien sur des fonctions synthétiques que pour celles issues d'un problème d'optimisation de forme en aérodynamique. L'objectif est d'évaluer l'efficacité d'algorithmes adaptatifs d'interpolation parcimonieuse [2], et de les comparer avec l'approche du kriging développée dans le cadre design and analysis of computer experiment (DACE) [21]. La précision et le temps de calcul des deux méthodes sont étudiés lorsque le nombre N d'échantillons utilisés pour l'interpolation augmente. Les cas tests montrent que les deux méthodes sont comparables en terme de précision. Cependant, lorsque la dimension d augmente, le temps de calcul associé à l'enrichissement de l'échantillon pour le kriging devient prohibitif pour les grandes valeurs de N . Ce problème est contourné dans le cas de l'interpolation parcimonieuse pour lequel le temps de calcul est linéaire en N et d .

1. INTRODUCTION

Surrogate models, such as the surface response method are becoming increasingly popular in performing various optimization or uncertainty quantification studies for parameter dependent complex problems. Such problems are difficult to handle numerically, especially in high dimension, that is, when the number d of parameters is large. Surrogate models are a non intrusive approach that aim at providing a simplified and computable representation of the solution in the parameter space. The principle of a surrogate model relies on an efficient interpolation procedure that estimates a scalar or vector field using a sampled data set. Each sample typically corresponds to an instance of the solution of a complex problem, often computationally expensive.

* *Acknowledgements: the authors thank ONERA, IUF and UPMC for their support during the CEMRACS of 2013.*

¹ LJLL, Université Pierre & Marie Curie, 75005, Paris, France

² Onera, The French Aerospace Lab. 92322 Châtillon, France

It is therefore mandatory that the procedure achieves the required accuracy for a reasonable number N of samples, and that the CPU time required for the construction of the interpolant remains negligible compared to that of the instances computation. Efficient algorithms are particularly needed as the number of variables d increases due to the so-called curse of dimensionality. In particular, one may significantly gain by considering adaptive strategies, for which every new sample is chosen based on the information gained through the previous interpolation procedure. Here, two such strategies are being considered.

- The kriging procedure, initially developed by [11] has become increasingly popular over the past decades, due to its robustness in the sense that the interpolation points may be at arbitrary locations. This approach assumes that the function to be interpolated, is the realization of a stochastic process and computes the interpolant at a given point as a best unbiased linear estimator from the available data set. The relation between the interpolation and the current sample is based on a covariance function, often assumed to be of gaussian form, with internal parameters that have to be optimized in order to provide a reliable estimate. In an adaptive context, the variance of this estimator also provides with a way to select the location for additional data to be included in the data sample, in order to improve the accuracy of the surrogate model.
- Sparse approximation methods, originally developed by [24] in the context of quadrature rules, also became popular over the past decades. These methods aim at constructing high precision polynomial or piece-wise polynomial approximations while retaining a limited number of grid points through the introduction of a specific sparse grid. In recent years, adaptive versions have been introduced and studied, in particular in [2] for the polynomial case based on a greedy strategy which selects the next sparse grid point in order to minimize the interpolation error.

Our present study is motivated by shape design problems in aerodynamics. For instance, minimizing drag at constant lift of an aerofoil, at cruise flight conditions, is of major importance since it results in substantial savings of fuel consumption. The basic idea is to prescribe control points around a two-dimensional aerofoil and alter the geometry using CAD functions such as B-spline functions for example [10, 12]. This topic has been treated extensively over the past decades [4, 7–9, 14, 17, 22]. However, most studies appeared to report up to 6 design parameters and less than 5000 samples for evaluating the surrogate model. Beyond this limit, the commonly used interpolation procedures appear to be far more computationally expensive than the aerodynamics code itself. For instance when considering the kriging procedure [12], the computational time evolves cubically with the number of evaluation points N . In addition, the optimization of the internal parameters of the problem requires minimizing a non-convex cost function defined on a space of dimension d , where d is the number of variables, the evaluation of which requires by itself $\mathcal{O}(N^3)$ operations. Therefore the total computational grows fastly with N and this effect is even more accentuated in high dimension. Note in addition that the value of N needed to obtain a reliable estimate typically grows with d , due to the so-called curse of dimensionality.

An alternative approach which aim to alleviate the curse of dimensionality is provided by the use of *sparse grids*. Sparse grid methods have been introduced by Smolyak in the context of numerical integration [24]. They are based on particular unions of tensor product grids for which fineness in one variable is typically compensated by coarseness in other variables. This approach has been extensively applied to uncertainty quantification [29, 30] and applied to aerodynamics recently, see [13, 28] for a review. Very recently, [19] have applied this method for the case of a subsonic aerofoil using 14 uncertain parameters. However there is a lack in the literature when considering surface response methods using sparse grid interpolation approaches. Moreover, surface response methods appear to be efficient methods on non-intrusive robust optimization problems where one has to optimize a design subject to uncertainty, see [5] for a recent review. Here, we want to investigate the already mentioned adaptive versions of sparse grid interpolation introduced in [2] in the polynomial case. We are also interested in adapting this approach to the piecewise polynomial case which is better adapted to the presence of locally sharp regions or discontinuities in the function to be interpolated.

Therefore, we consider the response surface of a two-dimensional aerofoil using adaptive sparse grids interpolation approaches and compare it in terms of approximation capabilities with the kriging, considered as the reference method.

The rest of the paper is organized as follow. We successively detail in §2 the kriging and sparse interpolation adaptive algorithms. For both methods, we discuss several adaptive strategies for the enrichment of the sampling set. We then present in §3 two test cases, namely a synthetic function and the lift to drag ratio of a two-dimensional aerofoil in a transonic flow as a function of shape parameters. These test cases reveal that both adaptive interpolation algorithms behave similar in terms of precision with respect to the number N of samples. However, the adaptive kriging algorithm becomes computationally intractable for large values of d and N , while the CPU cost of the sparse interpolation algorithm is linear in both N and d . This advocates for the use of this second class of algorithms for high-dimensional problems. Some conclusions and perspectives are drawn in §4.

2. METHODS

In the present investigation, the aim is either to approximate or optimize a function

$$y \mapsto f(y), \quad (1)$$

where the coordinate vector $y = (y_1, \dots, y_d)$ concatenates the various parameters of a given model problem, and $f(y) \in \mathbb{R}$ is a quantity of interest computed from the solution of this problem for the particular value of y . For simplicity we assume that y ranges in the hypercube $U = [-1, 1]^d$. There is no loss of generality in this assumption when every y_j ranges independently in a finite interval I_j since we may then use a standard affine change of variable between I_j and $[-1, 1]$.

The objective is to build an accurate numerical approximation of f or of $M := \max_{y \in U} f(y)$ using the smallest possible number N of evaluations $f(y^i)$ at given points $y^i \in U$ from a sampling set $S_N = \{y^1, \dots, y^N\}$. Of course, if an approximation \tilde{f} to f has been computed, one way of approximating M is by $M := \max_{y \in U} \tilde{f}(y)$. However the specific search for the maximum may benefit from using different sampling sets S_N for a given N than when searching for a global approximation to f .

2.1. The kriging procedure

Kriging is based on the hypothesis [15] that the unknown function f is random can be decomposed into

$$f(y) = \mu(y) + g(y), \quad (2)$$

where $\mu(y)$ and $g(y)$ are sought as a deterministic contribution from a low-dimensional space and a random fluctuation respectively. Typically, we search for μ as a constant or low order polynomial. The random process g is assumed to be centered, second order, with covariance kernel of stationary form

$$\text{Cov}(g(y), g(z)) = \mathbb{E}(g(y)g(z)) = \phi(y, z) = \varphi(y - z). \quad (3)$$

In this framework, the kriging interpolation to $f(y)$ is a linear combination

$$I_N f(y) := \sum_{i=1}^N \lambda_i(y) f(y^i), \quad (4)$$

defined as the best linear unbiased estimator, in the sense that it minimizes the mean square error

$$\mathbb{E}(|f(y) - \sum_{i=1}^N \lambda_i f(y^i)|^2), \quad (5)$$

over all choices of λ^i under the constraint that $\mathbb{E}(\sum_{n=1}^N \lambda_n f(y^n)) = \mathbb{E}(f(y))$. This minimization problem leads to a linear system. Different models have been proposed [16] for $\mu(y)$:

- Simple kriging: μ is constant and known *a priori*.

- Ordinary kriging: μ is constant and unknown.
- Universal kriging; μ is an unknown polynomial of known order.

The ordinary kriging is by far the most popular approach and we use it in the present investigation. In this case, the unbiased constraint implies that $\sum_{i=1}^N \lambda_i(y) = 1$, which also means that the interpolation process is exact for constants. Denoting by χ the Lagrange multiplier, the resulting linear system for finding the $\lambda_i(y)$ is given by

$$\begin{bmatrix} \phi(y^1, y^1) & \cdots & \phi(y^1, y^N) & 1 \\ \vdots & \ddots & \vdots & 1 \\ \phi(y^N, y^1) & \cdots & \phi(y^N, y^N) & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ \chi \end{bmatrix} = \begin{bmatrix} \phi(y^1, y) \\ \vdots \\ \phi(y^N, y) \\ 1 \end{bmatrix}. \tag{6}$$

Standard linear algebra manipulations show that the resulting interpolant also writes

$$I_N f(y) = \sum_{i=1}^N \beta_i \phi(y^i, y) + \beta_{N+1}, \tag{7}$$

where the weights β_i are solution to the system

$$\begin{bmatrix} \phi(y^1, y^1) & \cdots & \phi(y^1, y^N) & 1 \\ \vdots & \ddots & \vdots & 1 \\ \phi(y^N, y^1) & \cdots & \phi(y^N, y^N) & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \\ \beta_{N+1} \end{bmatrix} = \begin{bmatrix} f(y^1) \\ \vdots \\ f(y^N) \\ 0 \end{bmatrix}, \tag{8}$$

The expression of the kriging interpolant by (7) is therefore more convenient since the system to be solved does not depend on y . It reveals that the interpolant is a linear combination of the constant function and of the functions $y \mapsto \phi(y_i, y)$ for $i = 1, \dots, N$.

The covariance kernel $\phi(y, z)$ has to be positive definite. A frequently considered form, that we use in the present investigations, is the gaussian

$$\phi(y, z) = \exp\left\{-\frac{1}{2} \sum_{k=1}^d \frac{(y_k - z_k)^2}{\zeta_k^2}\right\}. \tag{9}$$

The kernel parameters $(\zeta_1, \dots, \zeta_d)$ could be fixed *a-priori*, however in order to improve the accuracy of the interpolation process they should rather be adjusted with the objective of minimizing the error between $f(y)$ and its interpolation $I_N f(y)$. Here, we use a standard cross validation procedure, the so-called leave-one-out method, that consists in removing a point y^i from the sampling set S_N and computing the error between the kriging interpolant $I_{N-1}^i f(y^i)$ computed from $S_N - \{y^i\}$ and the exact value $f(y^i)$. We then select $\zeta = (\zeta_1, \dots, \zeta_d)$ by minimizing the mean square error

$$E(\zeta) = \frac{1}{N} \sum_{i=1}^N (f(y^i) - I_{N-1}^i f(y^i))^2. \tag{10}$$

The computations of the N interpolants $I_{N-1}^i f$ can be done efficiently using Rippa's method [20]. However, the optimization task becomes computationally intensive as d and/or N become moderately large, even when using the algorithm proposed in [26].

The choice of the sampling set S_N is of course critical in the behaviour of kriging algorithm. On the one hand, samples that are at mutual distance much smaller than the correlation lengths $\zeta_i^{1/2}$ result in ill conditioning of the covariance matrix in (8). On the other hand, enough samples have to lie within the correlation length in order to ensure a good approximation. The sampling set may either be predefined or adaptively built.

One typical predefined sampling set, that we consider in the present investigations, is given by the section $S_N = \{y^1, \dots, y^N\}$ from the Halton sequence

$$y^n = \begin{bmatrix} 2\gamma_{q_1}(n) - 1 \\ \vdots \\ 2\gamma_{q_d}(n) - 1 \end{bmatrix}, \quad (11)$$

where q_1, \dots, q_d are the first d prime numbers, and where $\gamma_q(n) = n_0q^{-1} + n_1q^{-2} + \dots + n_mq^{-m-1}$ with n_p such that $n = n_0 + n_1q + n_1q^2 + \dots + n_mq^m$ in the q -adic expansion (see for instance [18]). This sequence aims at building for each N a quasi-uniform sampling set over U .

Adaptive strategies for building the sampling set are based on the mean square error function

$$y \mapsto \sigma^2(y) := \mathbb{E}(|f(y) - I_N f(y)|^2) \quad (12)$$

which can be explicitly computed at each y for the kriging estimator. One first strategy consists in selecting the new point y^{N+1} which maximizes $\sigma^2(y)$ over U . In contrast to the Halton sequence, this strategy may lead to non-uniform distributions in the sampling sets. In particular, more data points are likely to be introduced in the directions where the covariance parameters ζ_k are small.

In our numerical tests, we perform the search for the new point y^{N+1} based on maximizing $\sigma(y)$ using the same differential evolution algorithm as used for the optimization of the correlation vector ζ . The computation cost of both optimization procedures constitute a significant computational bottleneck for the adaptive kriging algorithm as d and N both get large. In addition, the computation of the kriging approximation requires solving a $N \times N$ system with a full matrix, typically amounting in $\mathcal{O}(N^3)$ complexity.

2.2. The sparse interpolation procedure

The sparse interpolation procedure in arbitrary dimension is based on tensorization-sparsification of a hierarchical interpolation procedure in one dimension. We recall in a nutshell the idea for polynomial interpolation as proposed and analyzed in [2], then show how the procedure can be generalised to other types of interpolation.

Given the dimension d , the multivariate polynomial spaces considered are of the general form

$$\mathbb{P}_\Lambda := \text{Span}\{y^\nu = y_1^{\nu_1} \dots y_d^{\nu_d} : \nu = (\nu_1, \dots, \nu_d) \in \Lambda\}, \quad (13)$$

where $\Lambda \in \mathbb{N}^d$ is an index set that is assumed to be downward closed (also called lower set), in the sense that, given $\nu, \mu \in \mathbb{N}^d$,

$$\nu \in \Lambda \text{ and } \mu_i \leq \nu_i, \quad i = 1, \dots, d \Rightarrow \mu \in \Lambda. \quad (14)$$

Consider any infinite sequence $Z := (z_i)_{i \geq 0}$ of pairwise distinct points in $[-1, 1]$, and denote by I_k the univariate Lagrange interpolation operator onto \mathbb{P}_k associated with the section $\{z_0, \dots, z_k\}$. The operators I_k can be computed recursively. Indeed, for any $k \geq 0$, the difference operator $\Delta_k := I_k - I_{k-1}$ can be expressed as

$$\Delta_k f = \left(f(z_k) - I_{k-1} f(z_k) \right) h_k, \quad h_0(z) = 1, \quad h_k(z) = \prod_{j=0}^{k-1} \frac{z - z_j}{z_k - z_j}, \quad (15)$$

with the convention I_{-1} is the null operator. Now, for an arbitrary lower set $\Lambda \subset \mathbb{N}^d$, we introduce the grid of points

$$\Gamma_\Lambda := \left\{ z_\nu : \nu \in \Lambda \right\} \quad \text{where} \quad z_\nu := (z_{\nu_j})_{j=1, \dots, d} \in [-1, 1]^d. \quad (16)$$

It is proved in [2] that the grid Γ_Λ is unisolvent for the polynomial space \mathbb{P}_Λ and that the interpolation operator is given by

$$I_\Lambda := \sum_{\nu \in \Lambda} \Delta_\nu, \quad \Delta_\nu := \otimes_{j=1}^d \Delta_{\nu_j} \quad (17)$$

We observe that this coincides with the telescopic sum $I_k = \sum_{j=0}^k \Delta_k$ for the univariate case $d = 1$. Similarly to the univariate case, it has been proved in [2] that the hierarchical computation of the interpolation operator I_Λ is possible. More precisely, given a lower set Λ and any multi-index $\nu \in \mathbb{N}^d \setminus \Lambda$ such that $\Lambda' := \Lambda \cup \{\nu\}$ is a lower set, the operator $I_{\Lambda'}$ is the sum of I_Λ and the increment operator Δ_ν which can be computed using

$$\Delta_\nu = \left(f(z_\nu) - I_\Lambda f(z_\nu) \right) H_\nu, \quad H_\nu := \otimes_{j=1}^d h_{\nu_j}. \tag{18}$$

The stability of the operators I_Λ is critical for numerical applications, such as the design of surrogate models from computer experiments. It is measured by the Lebesgue constant

$$\mathbb{L}_\Lambda := \max_{f \in C(U) \setminus \{0\}} \frac{\|I_\Lambda f\|_{L^\infty(U)}}{\|f\|_{L^\infty(U)}}. \tag{19}$$

This constant depends on the sequence Z , in particular through the Lebesgue constants \mathbb{L}_k of the univariate interpolation operators I_k defined similarly. It was shown in [2] that algebraic growth of \mathbb{L}_k yields algebraic growth of the Lebesgue constant \mathbb{L}_Λ . More precisely

$$\mathbb{L}_k \leq (1+k)^\theta, \quad \text{for any } k \geq 1 \implies \mathbb{L}_\Lambda \leq (\#\Lambda)^{\theta+1}. \tag{20}$$

Surprisingly, the previous implication is valid whatever the dimension d and the shape of the finite lower set Λ . We note that sequences Z with provable slow algebraic growth of the Lebesgue constant \mathbb{L}_k exist in the literature. These sequences are known as \mathfrak{R} -Leja sequences [1]. They can be computed explicitly and we use them in our numerical tests for sparse polynomial interpolation.

The tensorization-sparsification approach used in the construction of the polynomial interpolation procedure can be generalized to other types of interpolation. We describe the approach in an abstract context. We consider the following sets

$$(\mathcal{T}, \leq), \quad Z_{\mathcal{T}} := \{z_\lambda : \lambda \in \mathcal{T}\}, \quad H_{\mathcal{T}} := \{h_\lambda : \lambda \in \mathcal{T}\}, \tag{21}$$

that stands respectively for a countable partially ordered set of indices, a sequence indexed in \mathcal{T} of pairwise distinct abscissas in $[-1, 1]$ and a hierarchical basis indexed in \mathcal{T} of functions on $C([-1, 1])$ satisfying

$$h_\lambda(z_{\lambda'}) = \delta_{\lambda, \lambda'}, \quad \text{if } \lambda' \leq \lambda. \tag{22}$$

We consider the set of multi-indices $\mathcal{T}^d := \{\nu = (\nu_1, \dots, \nu_d) : \nu_j \in \mathcal{T}\}$ and define lower sets $\Lambda \subset \mathcal{T}^d$ similarly to (14) with \leq being the partial order over \mathcal{T} . For a lower set Λ , we introduce

$$\Gamma_\Lambda := \left\{ z_\nu := (z_{\nu_1}, \dots, z_{\nu_d}) : \nu \in \Lambda \right\}, \quad \mathbb{H}_\Lambda := \text{span} \left\{ H_\nu := \otimes_{j=1}^d h_{\nu_j} : \nu \in \Lambda \right\}, \tag{23}$$

the grid of interpolation points and the space of interpolation. The same arguments as used in [2] for the polynomial setting show that the grid Γ_Λ is unisolvant for the space \mathbb{H}_Λ . In the general context where \mathcal{T} might not be totally ordered, the formula (17) does not make clear sense, yet we may still rely on the recursive computation of the interpolation operators. Namely, if Λ is lower set and $\nu \in \mathcal{T}^d \setminus \Lambda$ such that $\Lambda' = \Lambda \cup \{\nu\}$ is a lower set, then we have

$$I_{\Lambda'} f = I_\Lambda f + \left(f(z_\nu) - I_\Lambda f(z_\nu) \right) H_\nu. \tag{24}$$

For numerical experiment, we consider dyadic hierarchical piecewise linear or piecewise quadratic interpolation. For such interpolation procedures, the set \mathcal{T} is defined by

$$\mathcal{T} = \{\lambda_{-1}, \lambda_1, (0, 0)\} \cup \left\{ (j, k) : -2^{j-1} \leq k \leq 2^{j-1} - 1, j = 1, 2, \dots \right\} \tag{25}$$

induced with the partial order $\lambda_{-1} \leq \lambda_1 \leq (0, 0)$ and

$$(j, k) \leq (j + 1, 2k), \quad (j, k) \leq (j + 1, 2k + 1), \quad (j, k) \in \mathcal{T}. \quad (26)$$

The set \mathcal{T} is a binary tree where λ_{-1} is the root node, $(0, 0)$ is a child of λ_1 which is a child of λ_{-1} , every node (j, k) has two children $(j + 1, 2k)$ and $(j + 1, 2k + 1)$, and the relation $\lambda' \leq \lambda$ means λ' is a parent of λ . We associate with \mathcal{T} the set of abscissas

$$Z_{\mathcal{T}} := \{z_{\lambda_{-1}}, z_{\lambda_1}, z_{(0,0)}\} \cup \left\{ z_{(j,k)} := \frac{2k+1}{2^j} : (j, k) \in \mathcal{T}, j \geq 1 \right\}, \quad (27)$$

where $z_{\lambda_{-1}} = -1$, $z_{\lambda_1} = 1$ and $z_{(0,0)} = 0$. We also associate with \mathcal{T} the hierarchical basis of piecewise linear functions $H_{\mathcal{T}} = \{h_{\lambda} : \lambda \in \mathcal{T}\}$ defined over $[-1, 1]$ by

$$h_{\lambda_{-1}}(s) = 1, \quad h_{\lambda_1}(s) = \frac{1+s}{2}, \quad h_{(j,k)}(x) = \psi(2^j(s - z_{j,k})), \quad \psi(s) := \max\{0, 1 - |s|\}, \quad (j, k) \in \mathcal{T}, \quad (28)$$

It is easy to verify that the function h_{λ} and the abscissas z_{λ} satisfies the condition (22), therefore the hierarchical interpolation can be performed. Let us note that in dimension $d = 1$, the hierarchical interpolation amounts in first approximating f by the constant function of value $f(-1)$, second by the affine function that coincides with f at -1 and 1 , third by the piecewise affine function that coincides with f at -1 , 0 and 1 , and in further steps refine by interpolating at the midpoint of an interval between two adjacent interpolation points. The index j corresponds to the level of refinement, or the depth of the node in the binary tree.

In the case of piecewise quadratic interpolation, the procedure is exactly the same, except that the hierarchical basis is now given by

$$h_{\lambda_{-1}}(s) = 1, \quad h_{\lambda_1}(s) = \frac{(1+s)^2}{4}, \quad h_{(j,k)}(x) = \psi(2^j(s - z_{j,k})), \quad \psi(s) := \max\{0, 1 - s^2\}, \quad (j, k) \in \mathcal{T}. \quad (29)$$

which corresponds to interpolation by piecewise quadratic functions with the same ordering on the interpolation points as in the piecewise affine case.

Having fixed the sparse interpolation procedure (polynomial, piecewise affine or piecewise quadratic), the problem is now to select lower sets Λ giving the best possible interpolation spaces \mathbb{H}_{Λ} (or \mathbb{P}_{Λ} in the polynomial case) for the target function f . In the non-intrusive context that motivated this work, no prior information is known on f , whence the optimal approximation space \mathbb{H}_{Λ} for a given $N = \#\Lambda$ is not accessible. Therefore, we may only rely on greedy type strategies. The recursive formula (24) suggests to couple the interpolation algorithm with an adaptive strategy for the choice of best multi-index ν used to enrich Λ . Given Λ a lower set, we denote $\mathcal{N}(\Lambda)$ the set of adjacent neighbours to Λ , which are the multi-indices $\nu \in \mathcal{T}^d \setminus \Lambda$ such that $\Lambda' := \Lambda \cup \{\nu\}$ is a lower set. Depending on the approximation context, we choose to enrich Λ by $\nu \in \mathcal{N}(\Lambda)$ that either satisfies

- the supremum norm of the increment $\|\Delta_{\nu}f\|_{L^{\infty}(U)}$ is maximal,
- the least square norm of the increment $\|\Delta_{\nu}f\|_{L^2(U)}$ is maximal,
- the value $f(z_{\nu})$ is maximal.

The two first criterions are designed for the approximation of f in the L^{∞} and the L^2 sense, respectively, while the third criterion is designed for optimization and can be seen as a way of exploring the local maxima of f .

Although the set of candidates $\mathcal{N}(\Lambda)$ might be very large, the enrichment step requires at most d new evaluations of f . Indeed, for $\nu \in \Lambda$ and $\Lambda' = \Lambda \cup \{\nu\}$, the set $\mathcal{N}(\Lambda') \setminus \mathcal{N}(\Lambda)$ contains at most d indices. We also note that for the two first strategies, the computation of $\|\Delta_{\nu}f\|$ for the new indices consists merely in computing

$$|f(z_{\nu}) - I_{\Lambda}f(z_{\nu})| \prod_{j=0}^d \|h_{\nu_j}\|, \quad (30)$$

with $\|\cdot\|$ being the L^∞ or the L^2 norm. This is computationally fast since the h_n are known in advance and their norm $\|h_n\|$ can be tabulated. The cost of the computation $\|\Delta_\nu f\|$ is essentially dominated by the evaluation of $f(z_\nu)$ in cases where f is evaluated through a heavy numerical solver.

While the above strategies often give good numerical results in practice, they can be defeated when the target function has oscillations that fail to be captured by the greedy selection procedure. For example, consider the two first criterions for piecewise linear adaptive interpolation of a univariate function f such that $f(\frac{1}{2}) = \frac{1}{2}(f(0) + f(1))$. Then, the increment corresponding to the point $z_{(1,0)} = \frac{1}{2}$ is $\Delta_{(1,1)}f = 0$. Therefore the adaptive algorithm might fail to explore the region $[0, 1]$ on which f could still oscillate away from its linear interpolation. Similarly an algorithm based on the third criterion could be trapped in local maximas.

One way to remedy this defect consists in modifying the algorithm as follows: we produce the nested sequence of lower sets

$$\Lambda_1 \subset \Lambda_2 \subset \dots \subset \Lambda_N \subset \dots$$

with $\#\Lambda_N = -N$, by alternating $p - 1$ adaptive steps where the new index $\nu \in \mathcal{N}(\Lambda_N)$ is picked based on the chosen criterion when $N \notin p\mathbb{N}$, and one “conservative” step where we pick the most “ancient” index $\nu \in \mathcal{N}(\Lambda_N)$ when $N \in p\mathbb{N}$, in the sense that it has been lying in $\mathcal{N}(\Lambda_k)$ for the smallest value of $k \leq N$. This conservative step allows us to explore the whole parameter space U , while retaining the adaptive feature of the algorithm. In our numerical test we have used the value $p = 5$, based on empirical observation that it gives a good balance between adaptivity and exploration.

An important feature of these adaptive algorithm is that the computational cost scales like $d + N$, where $N = \#\Lambda_N$ and d is the dimension. Indeed, the construction of Λ_{N+1} from Λ_N requires inspecting at most d new values of the unknown function. This contrasts with the kriging algorithm and will be reflected in the numerical tests.

3. NUMERICAL TESTS

3.1. Synthetic functions

The performances of the above algorithms have been assessed on two classes of problems. First, two synthetic functions are considered. The first is a smooth one-dimensional sinusoidal type function whereas the second is two-dimensional and exhibits a sharp gradient along one abscissa, depicted in figure 1. The synthetic functions are given by

$$h_1(y_1) := g(10y_1 - 2) \cos(y_1^2); \quad g(s) := \frac{s|s|}{1 + s^2}, \tag{31a}$$

$$h_2(y_1, y_2) := g(10y_1 - 2) \cos(5y_1^2)g(100y_2 - 20) \cos(5y_2^2). \tag{31b}$$

These functions are evaluated in the domain $U = [-1, 1]^d$ with $d = 1$ and 2 respectively, and displayed on Figure 1. The univariate function h_1 is smooth with a kink at $y_1 = 0.2$. The two-dimensional function h_2 has a sharp gradient along $y_2 = 0.2$. The sparse interpolation algorithm is first considered and the performances are compared on Figure 2 between the polynomial, piecewise linear and piecewise quadratic procedures. The L^∞ and L^2 errors in all figures have been computed based on 10^6 randomly chosen samples. For the two test cases, the piecewise quadratic algorithm shows the best convergence. This procedure is higher accurate than its piecewise linear counterpart, while still providing local adaptivity as opposed to its polynomial counterpart.

A comparison with the kriging procedure is shown on Figures 3 where results between the piecewise quadratic sparse interpolation appears to be more precise than the kriging using the Halton sequence.

In the case of the one-dimensional function, the quadratic polynomial approach appears to be also more precise than the adaptive kriging strategy based on the mean square error (12). For the two-dimensional synthetic function, a sample size $\#\Lambda > 128$ appears to be mandatory in order to decrease the error significantly in all cases considered. The adaptive kriging appears to be slightly more precise for $128 < \#\Lambda < 512$, yet comparable with the sparse interpolation algorithm when $\#\Lambda$ reaches 512. In the case of the piecewise sparse interpolation

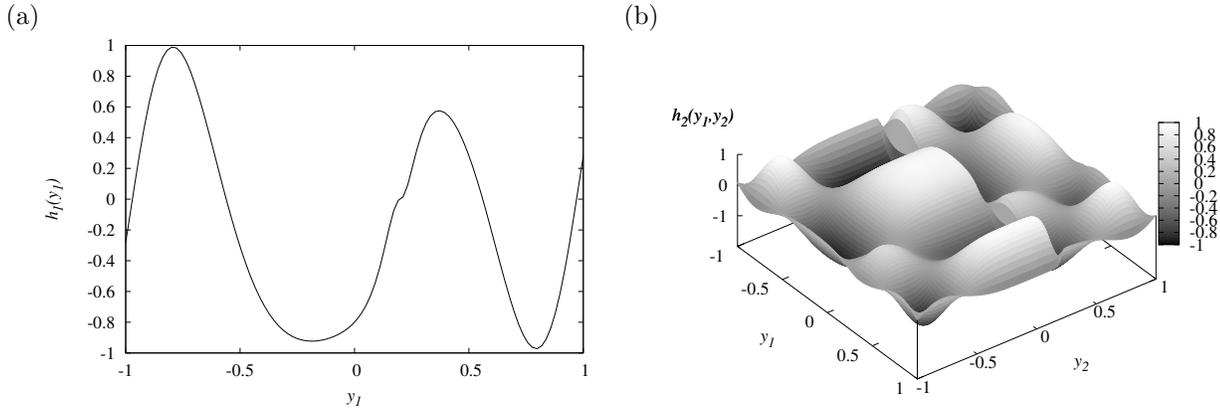


FIGURE 1. Sketch of the one-dimensional (a) and two-dimensional (b) synthetic functions, given in equations (31a) and (31b)

procedure, we have observed that the selected interpolation points are mainly clustered along the sharp gradient region. In the case of the Kriging, the internal parameter ζ_2 appears to be much smaller than ζ_1 , and the points appear to be mostly distributed along the y_2 direction. Note that due to the computational time involved in computing the adaptive kriging, the results for $\#(\Lambda) > 512$ could not be computed as the parameters for the kriging are recomputed every time a new point is included in the data set. We discuss later the comparison between computational costs.

3.2. Surrogate model for a transonic wing profile

The second class of problem studied is a transonic flow over a RAE2822 wing profile. The flow solution is computed using the MISES code [6] which solves the Euler equations together with a boundary layer correction. Results are shown on Figure 4 for the baseline solution, compared with the data from the experiment in [3] for similar inflow conditions. The inflow conditions are given by the Mach number $M = 0.729$, the angle of attack $\alpha = 2.31^\circ$ and the Reynolds number $Re = 10^6$ based on the chord length c and the flow at infinity. The pressure distribution, shown on Figure 4(a) exhibits a discontinuity on the suction side of the aerofoil, denoting the presence of a shock, caused by the expanding geometry of the aerofoil after $c > 0.4$. In the present study, the value of interest is given by the ratio between the lift and the drag

$$f = \frac{C_l}{C_d} \quad (32)$$

where

$$C_l = \int_{\xi} p(s) n_{\alpha}(s) ds, \quad \text{and} \quad C_d = \int_{\xi} p(s) t_{\alpha}(x) ds, \quad (33)$$

where $n_{\alpha} = \mathbf{n} \cdot \begin{pmatrix} -\sin(\alpha) \\ \cos(\alpha) \end{pmatrix}$ and $t_{\alpha} = \mathbf{n} \cdot \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$ with \mathbf{n} the outward normal vector. The shape of the baseline RAE2822 profile has been altered using four B-spline functions. More precisely

- The 4 control points are located at 5, 20, 40 and 60% of the chord c , see Figure 4(a).
- The amplitude of each B-spline is taken in the interval $[-1.7510^{-3}c, 1.7510^{-3}c]$ and is directed towards the normal to the initial profile ξ .
- The leading edge and the trailing edge have been considered as fixed.

In the present study, the parameter range is thus defined by the hypercube $[-1.75 \cdot 10^{-3}c, 1.75 \cdot 10^{-3}c]^4$, which after renormalization can be set to $U := [-1, 1]^4$. The change of the geometry is responsible for triggering the

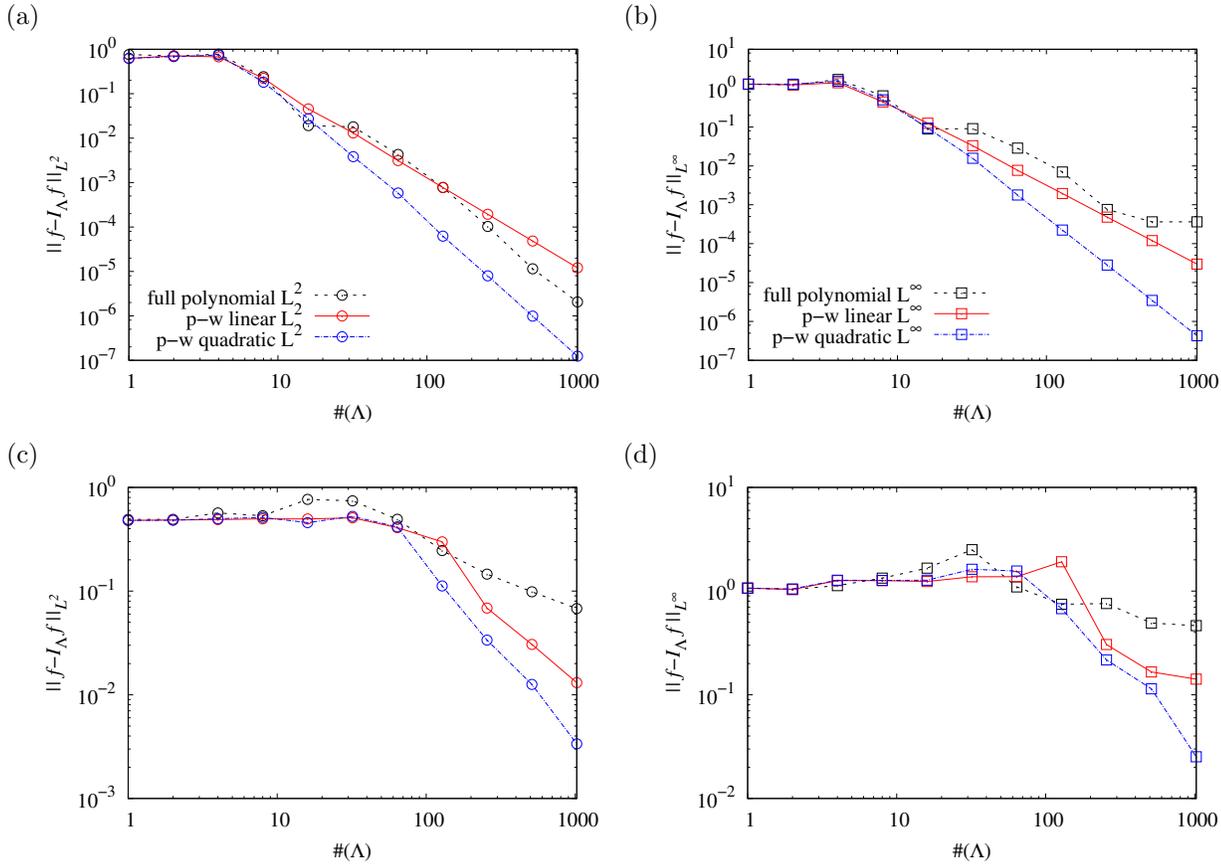


FIGURE 2. Comparison between the polynomial, piecewise linear and piecewise quadratic sparse adaptive interpolation procedures: L^2 error (left), L^∞ error (right), function h_1 (up) and function h_2 (down).

pressure gradient induced by the geometry of the profile. Even for small variations of the shape parameter w , the pressure gradient is altered significantly, which leads to nonlinear variations of the lift and the drag.

The kriging and the sparse adaptive algorithm are compared for different sample size and in several dimensions. The three procedures for sparse interpolation are compared on Figures 5. Similar to the synthetic case, the polynomial procedure is the least accurate whereas the piecewise linear and piecewise quadratic procedures appear to perform equally for both the L^2 and L^∞ error. The two kriging procedure (Halton and adaptive) are then compared on figure 6. The results appear to be comparable for sampling sizes $\#(\Lambda) \leq 128$. Beyond this limit, the adaptive kriging appears to perform slightly better for the L^∞ error. We observed that the valued of the internal parameters ζ are fairly similar and that the points selected by the adaptive procedure tend to be more clustered along the borders. This is due to the fact that the kriging suffers from the lack of data points beyond the limit of the parameter space and thus tends to cluster extra points on the boundaries.

The adaptive kriging is finally compared with the piecewise linear sparse interpolation on Figure 7. The adaptive kriging appears to perform slightly better than the piecewise linear sparse interpolation procedure with an error twice lower for $\#(\Lambda) = 512$ for both the L^2 and L^∞ errors. However, when comparing both methods in terms of computational time, the kriging algorithm appears to strongly suffer from the growth of the sampling size $N = \#(\Lambda)$, in contrast to the sparse interpolation algorithm, as illustrated on Figure 8. For $10^3 < N < 10^4$, the ratio between the CPU times can reach up to 10^5 . The reasons for this have been discussed

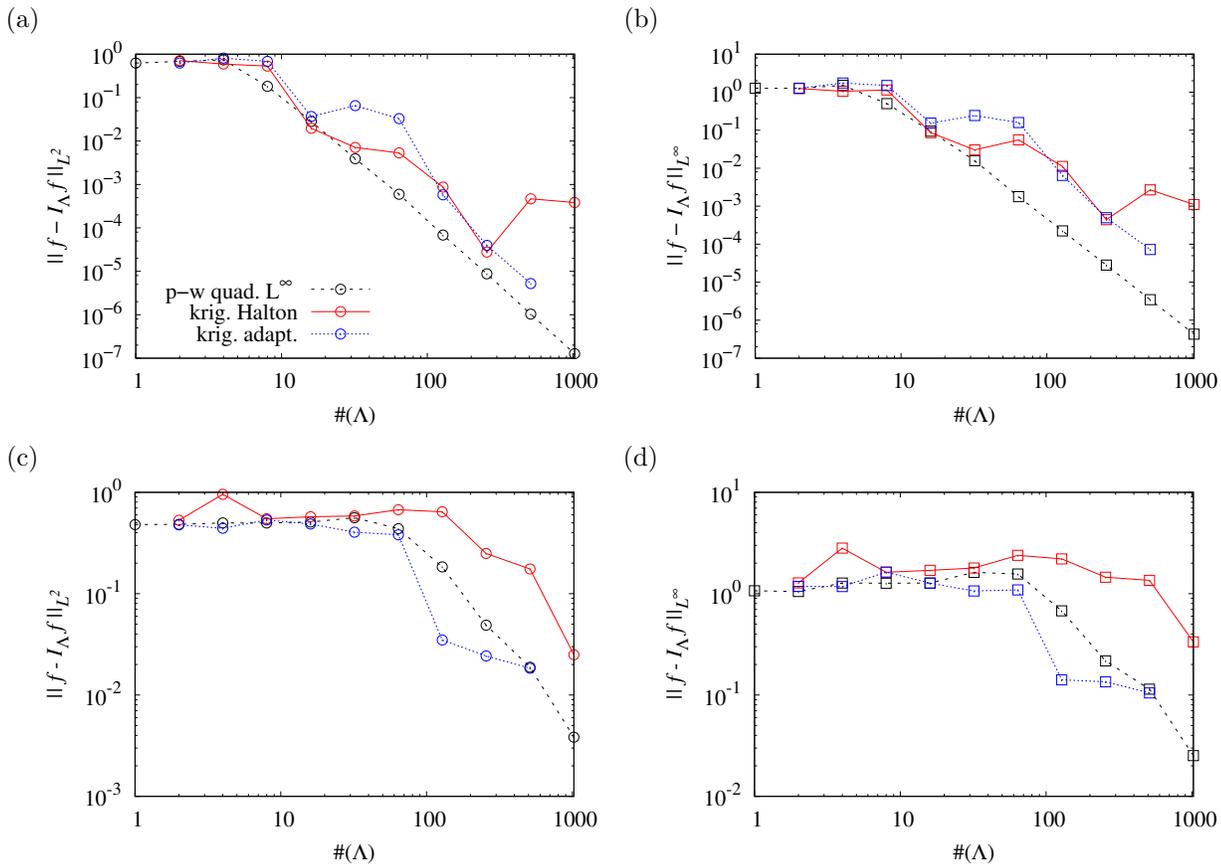


FIGURE 3. Comparison between the Kriging using either the Halton sequence or the adaptively selected sequence, and the piecewise quadratic sparse adaptive interpolation: L^2 error (left), L^∞ error (right), function h_1 (up) and function h_2 (down).

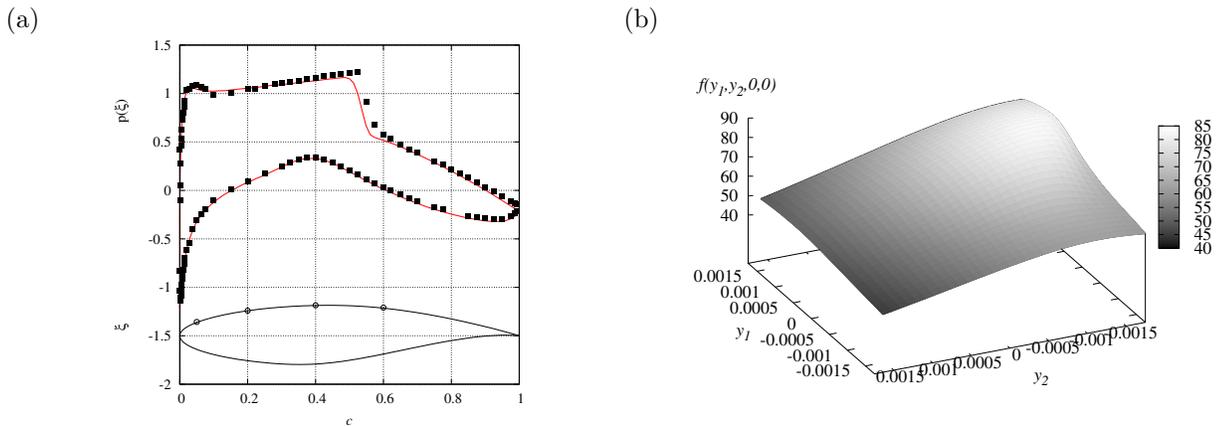


FIGURE 4. (a) Comparison of the pressure distribution on the RAE2822 baseline profile ξ for the MISES code and results from the experiment of [3] (black squares). (b) Surface response of the lift to drag ratio f for the parameter variable (y_1, y_2)

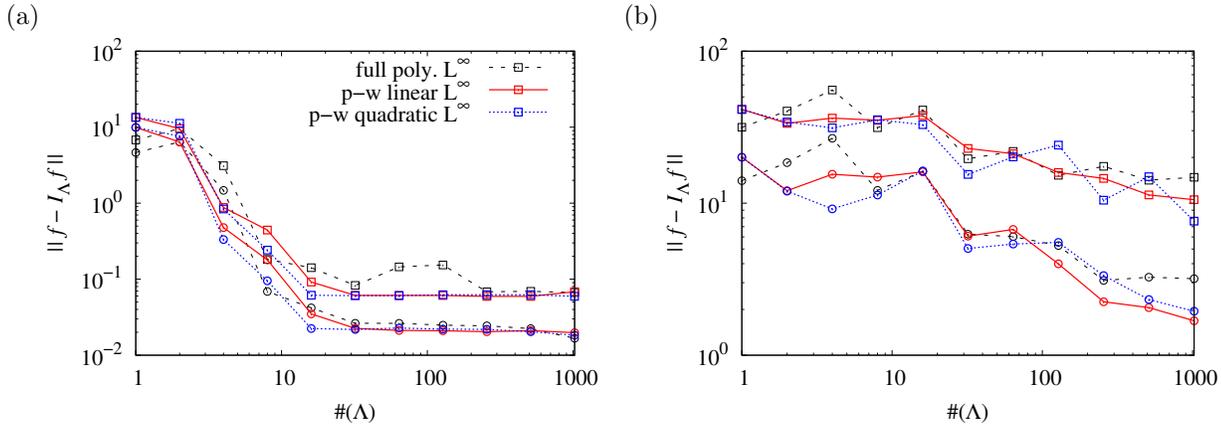


FIGURE 5. Comparison of the L^{∞} (\square) and the L^2 error (\circ) for the sparse interpolation algorithms applied to the functions $y_1 \mapsto f(y_1, 0, 0, 0)$ (a) and $(y_1, y_2, y_3, y_4) \mapsto f(y_1, y_2, y_3, y_4)$ (b).

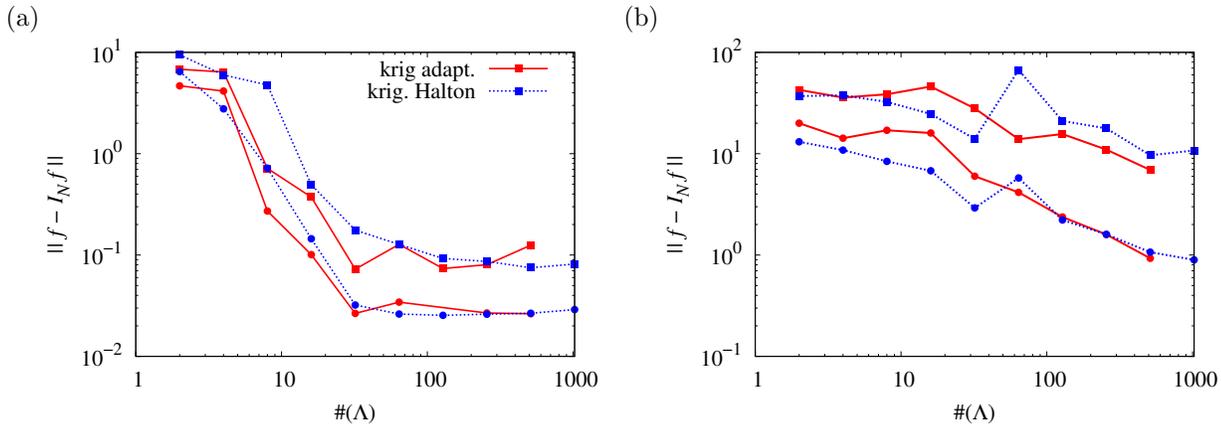


FIGURE 6. Comparison of the L^{∞} (\blacksquare) and the L^2 error (\bullet) for the kriging algorithms applied to the functions $y_1 \mapsto f(y_1, 0, 0, 0)$ (a) and $(y_1, y_2, y_3, y_4) \mapsto f(y_1, y_2, y_3, y_4)$ (b).

in the introduction of this paper, and are related in particular to the $\mathcal{O}(N^3)$ complexity of the inversion process in the kriging procedure.

4. CONCLUSIONS

In the present study, several classes of algorithms are discussed and compared with a special emphasis on adaptivity, with the goal of overcoming as much as possible the so-called *curse of dimensionality*. The first problem considered is the interpolation of a synthetic smooth function, characterized by local regions of sharp gradients. The kriging and the sparse interpolation procedures appear to perform similarly, in terms of error vs. number of evaluation points. Then a test case concerning the response surface of a wing at transsonic speeds in terms of shape parameters is considered. The results are again comparable between the kriging and the sparse interpolation procedure. However the comparison in terms of the computational time for a given amount of evaluation points is strongly in favour of the sparse interpolation algorithm. Although much effort has been

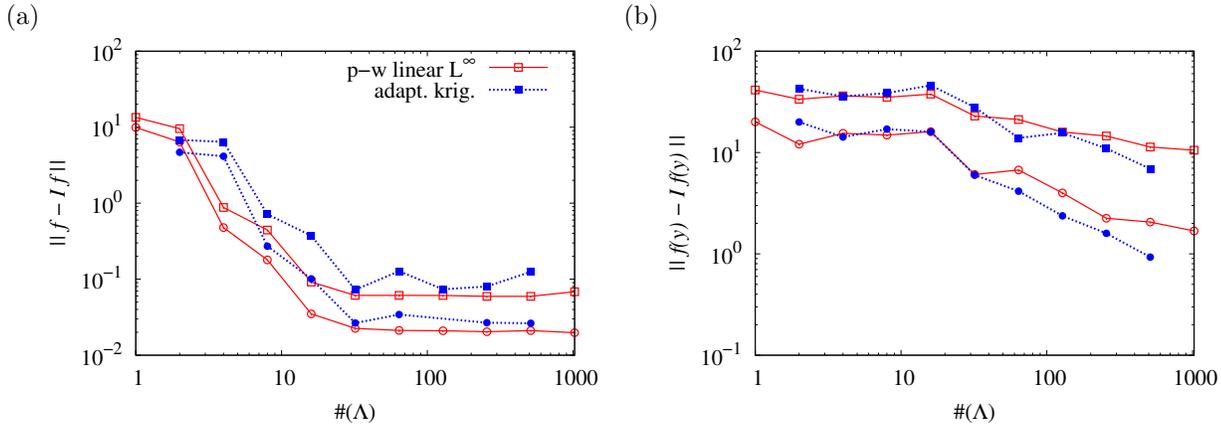


FIGURE 7. Comparison of the L^∞ error (\square piecewise linear sparse, \blacksquare adaptive kriging) and of the L^2 error (\circ piecewise linear sparse, \bullet adaptive kriging), for the functions $y_1 \mapsto f(y_1, 0, 0, 0)$ (a) and $(y_1, y_2, y_3, y_4) \mapsto f(y_1, y_2, y_3, y_4)$ (b).

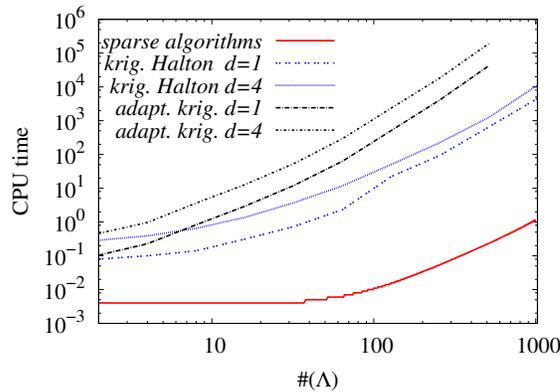


FIGURE 8. Comparison of the CPU time between adaptive and non-adaptive kriging for $d = 1$ and $d = 4$ and adaptive sparse interpolation for $d = 4$.

recently devoted to the kriging procedure, it appears that beyond a sampling $\#(\Lambda)$ of a few hundred points, the surrogate model to compute becomes expensive. Beyond a few thousand samples, the kriging surrogates might become more computationally expensive to evaluate than the CFD solver itself, which questions the relevance of this surrogate model for large scale engineering applications. This comparison between kriging and adaptive sparse interpolation has been carried out for fairly low-dimensional problems (up to $d = 4$) when compared with full size engineering applications. However, these results clearly show that both methods provide comparable accuracy but at significantly different computational cost, which clearly advocates in favour of the sparse adaptive interpolation procedure for computing surrogate models.

ACKNOWLEDGEMENTS

P.-Y. Passaglia acknowledges the support by the French "Agence Nationale pour la Recherche" (ANR <http://www.agence-nationale-recherche.fr/>) as part of the UFO (Uncertainty For Optimisation) project, referenced as ANR-11-MONU-0008. This support was purely financial.

REFERENCES

- [1] Chkifa, A. On the Lebesgue constant of Leja points on the unit disk and their real projections *J. Approx. Theory*, 166 :176-200, 2013 .
- [2] Chkifa, A. and Cohen, A., Schwab, C., High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Research Report ETHZ*, No 2012-22, 2012.
- [3] Cook, P. H., and M. A., McDonald and, Firmin., M.C. P., Aerofoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements, Experimental Data Base for Computer Program Assessment *AGARD Report* , AR 138, 1979.
- [4] Duta, M. C., and Duta, M. C., Multi-objective turbomachinery optimization using a gradient-enhanced multi-layer perceptron *Int. J. Numer. Meth. Fluids* 61, 591-605, 2009.
- [5] Forrester, A. I., and Keane, A. J., Recent advances in surrogate-based optimization *Prog. in Aero. Sci.*, 45, 50–79, 2009.
- [6] Giles, M., Drela, M., and Thompkins, W. T. Jr., Newton Solution of Direct and Inverse Transonic Euler Equations *AIAA paper*, 85-1530, 1985.
- [7] Goel, T., Dorney, D. J., Haftaka, R. T., and Shyy, W., Improving the hydrodynamics performances of diffused vanes via shape optimization *Comput. Fluids*, 37, 705-723, 2008.
- [8] JinWoo, Y., Byung, J. L., and Chongam, K., Exploring multi-stage shape optimization strategy of multi-body geometries using kriging-based metamodel and adjoint methods *Comput. Fluids* 68, 71-87, 2012.
- [9] Jouhaud, J.-C., Sagaut, P., Montagnac, M., and Laurenceau, J., A surrogate-model based multidisciplinary shape optimization method with application to a 2D subsonic airfoil *Comput. Fluids* 36, 520-290, 2007.
- [10] Karakasis M. K., Koubogiannis, D. G. and, Kyriakos, C. G., Hierarchical distributed metamodel-assisted evolutionary algorithms in shape optimization, *Int. J. Num. Meth. in Fluids* 53:455-469, 2007.
- [11] Krige, D. G., A statistical approach to some mine valuations and allied problems at the witwatersrand. Master's thesis, University of Witwatersrand, 1951.
- [12] Laurenceau, J., and Sagaut, P., Efficient response surfaces of aerodynamic functions with Kriging and Cokriging, *AIAA J.*, 46(2), 498-507, 2008.
- [13] Le Maître, O. P., and Knio, O. M., Spectral methods for uncertainty quantification with application to computational fluid dynamics, *Scientific Computation*, Springer Netherlands, 2010.
- [14] Martinelli, M., and Duvigneau, R., On the use of second order derivatives and metamodel-based Monte-Carlo for uncertainty estimation in aerodynamics *Comput. Fluids* 39, 953-964, 2004.
- [15] Matheron. G., La théorie des variables régionalisées, et ses applications. *Centre de Morphologie Mathématique de Fontainebleau*, 1970.
- [16] Matheron. G., *Traité de Géostatistique appliquée. Paris : Editions Technip*, 1962.
- [17] Moon, M., Afzal H., and Kwang-Yong, K., Multi-objective optimization of a rotating cooling channel with staggered pin-fins for heat transfer augmentation *Int. J. Numer. Meth. Fluids* 68(7), 922-938, 2011.
- [18] Rafajlowicz, E., and Schwabe, R., Halton andammersley sequences in multivariate nonparametric regression. *Statistics & Probability Letters* 76(8) :803–812, 2006.
- [19] Resmini, A., Peter, J., and Lucor D., High-dimensional stochastic investigation of 2D RANS flows about an helicopter airfoil *Int. J. Num. Meth. Fluids* (Submitted).
- [20] Rippa, S. An algorithm for selecting a good value for the parameter c in radial basis function interpolation *Advan. in Comp. Math.*, 11(2-3) :193–210, 1995.
- [21] Sacks, J., Welch W. J., Mitchell T. J., and Wynn H. P., Design and Analysis of Computer Experiments *Stat. Science*. 4(4) :409-423, 1989.
- [22] Giles, M., Drela, M., and Thompkins, W. T. Jr., Kriging modls for global approximation in simulation-based multidisciplinary design optimization *AIAA J.*, 39(12), 2233-2241, 2001.
- [23] Schonlau, M., Computer Experiments and Global Optimization *PhD Thesis, University of Waterloo*, 1997
- [24] Smolyak, S., Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions *Dokl. Akad. Nauk SSSR*, 4:240-243, 1963.
- [25] Sóbester, A., and Leary, S. J., and Keane. A. J., Estimation of error rates in discriminant analysis *J. of Global Optim.*, 33(1) :31-59, 2005.
- [26] Storm, R., and Price., K., Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces *Int. Comp. Science Inst.*, TR-95-012, 1995.
- [27] Villemonteix, J., and Vazquez, E., and Walter, E., An informational approach to the global optimization of expensive-to-evaluate functions *J. of Global Optim.*, 44(4)1:509-54, 2009.
- [28] Xiu, D., Numerical Methods for Stochastic Computations: A Spectral Method Approach, *Princeton University Press*, 2010
- [29] Xiu, D., and Karniadakis, G. E., The Wiener–Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, 24(2), 619-644, 2002.
- [30] Xiu, D., and Hesthaven, J. S., High-order collocation methods for differential equations with random inputs *SIAM J. Sci. Comput.*, 27(3), 1118-1139, 2005.