Paper 230

Proceedings of the Eleventh International Conference
on Computational Structures Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Gradient-Enhanced Metamodels and Multiparametric Strategies for Designing Structural Assemblies

L. Laurent, P.A. Boucard and B. Soulier
LMT-Cachan
(ENS Cachan/CNRS/Université Paris 6/PRES UniverSud Paris)
Cachan, France

Abstract

This paper proposes a multilevel model optimization strategy for structural assemblies. The general objective is to reduce computation costs; here, we focus on the costs which are associated with the generation of metamodels. Our goal is achieved through the introduction of two main elements: the multiparametric Strategy based on the LATIN method, which reduces the computation costs when the parameters vary, and the use of gradient-based metamodels. Cokriging and radial basis functions (RBF) metamodels are presented and performance of these approximations is illustrated with analytical and mechanical examples with one to four design variables.

Keywords: multilevel optimization, metamodel, cokriging, radial basis functions, multiparametric strategy, LATIN method, assemblies.

1 Introduction

Optimization techniques have been widely in use in industry for many years. They have been used to design structural elements, geometries, materials ... This kind of optimizations are feasible when the computation cost required to carry out one simulation is the low and acceptable. To achieve them, direct optimizations are mostly used: the optimizer monitors directly the simulator in order to locate the optimum of an objective function and to provide the associated set of parameters with respect of the constraints. Although optimizers are ever-increasing efficient and converges faster and faster to the solution of the optimization process, this kind of global optimization approach is not affordable in the context of assembly design: due to the nonlinearities resulting from contact or friction phenomena between parts, each simulation involves the resolution of a very complex and very costly problem. In order to reduce the computational time one can use a local optimization strategy. In this context the main

remaining difficulty is to define the local optimization problem and more precisely to build the best local optimization problem *i.e.* the problem which allows us to locate accurately the optimum with the fewer calls of the simulator. In order to reduce the computational time and to build the local optimization problem we propose to use two tools: the first is a specific mechanical model based on a dedicated strategy to solve assembly problems and the second is a gradient-enhanced metamodel.

This paper focuses on the cost associated to the building of metamodels using data coming from the mechanical model. The first section of this paper presents the multilevel model optimization strategy, then two tools are introduced: a MultiParametric Strategy to solve assembly problems and metamodels to provide approximation of objective functions. Two kinds of gradient-based metamodels will be presented: a gradient-based Gaussian Process and a gradient-based radial basis function. Finally these two tools are used together and a study based on quality criteria and computational costs is achieved.

2 The optimization process

Although the performance of the mechanical solvers and the optimizer increased since many years, direct optimization strategies are not affordable to deal with certain specific mechanical problems such as complex non-linear problems. Therefore we propose to use a two-level strategy. In opposition to the Surrogate Based Optimization strategy [1] in which the whole optimization process was achieved on the surrogate model, we propose to use the metamodel to feed a local optimizer with a start point. More precisely the whole optimization process is: first, the design space is sampled using a sampling method such as Latin Hypercube Sampling [3], on each sampled points the value and the derivatives of the objective function is computed using the full mechanical model; then the metamodel can be build using this information. A global and very fast optimization is performed on the metamodel using a global optimizer (for instance Particle Swarm Optimizer, Genetic Algorithm...). While an evaluation of the actual function using the full mechanical model is very expensive, each evaluation of the metamodel is very inexpensive. A high number of evaluations of the approximate model is no longer a problem. Finally the full mechanical model is connect directly with a local optimizer initialized with the approximate optimum set of parameters coming from the first optimization. This optimization strategy and the mechanical solver leads to a significant reduction in the computation cost.

Figure 1 shows the different tools used in the proposed two-level strategy. In this paper, we focus on the first part of the first level: the building of the metamodel using responses from the mechanical model. More precisely both quality and computational cost associated to the building of this approximation will be study. To do this, a MultiParametric Strategy [4] and gradient-enhanced versions of radial basis function [5] and kriging [6] metamodels will be applied on analytical and mechanical examples.

In the following sections the MultiParametric Strategy will be introduce; then

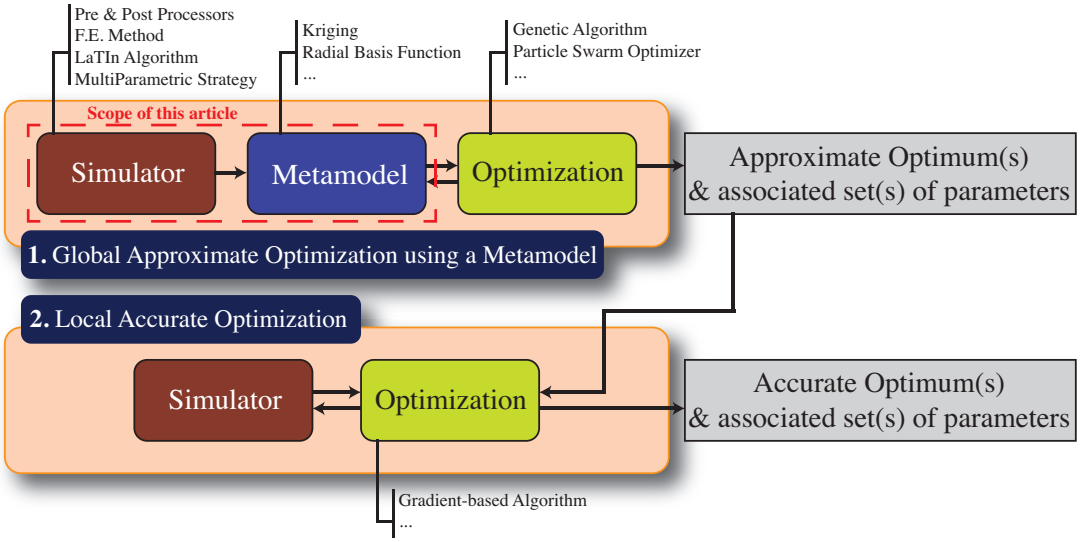


Figure 1: Multilevel Model Optimization

gradient-based metamodels will be detailed and finally these two tools will be used and study together on few mechanical examples.

3 The MultiParametric Strategy

In the context of multiparametric study or optimization process many non-linear similar mechanical simulations are achieve. Actually the mechanical problem is the same in the sense that only few parameters vary. In our case we propose to compute this non-linear problems using a LaTIn algorithm and in order to reduce the computational cost associated to the similar problems we propose to use the MultiParametric Strategy. These both techniques will be briefly described in the following sections.

3.1 The LaTIn method

The structures being considered in this paper are assemblies of linear elastic structures under the assumption of small perturbations. The only nonlinearities occur between parts of the assemblies and are due to contact and/or friction phenomena. In order to solve these problems, we use a dedicated strategy based on the LaTIn algorithm introduced by P. Ladevèze [7].

In the case of structural assemblies we consider a mixed domain decomposition approach based on three main points:

- The structure being studied is divided into substructures and interfaces. The chosen decomposition considers mixed force and displacement unknowns at the interfaces;

- A suitable iterative algorithm is used to solve the mechanical problem;
- The operators of the method remain constant and do not depend on the loading or on the parameters of the interfaces (friction coefficient, gap).

In order to solve non-linear problems, the LaTIn method is based on a partition of the problems on the substructures and on the interfaces in two groups of equations denoted A_d and Γ:

- The group A_d contains the *linear* and possibly global in space variables equations. It corresponds to the problems on the substructures and it is made up of the kinematic and the static admissibilities and the constitutive laws of the substructures;
- The group Γ contains *local* and possibly non-linear equations. This group allows us to treat non-linear problems on interfaces and so, it contains the interfaces specific approach and laws (such as friction and contact problems, [8]) and static equilibrium of the interfaces.

As shown in schematic representation in Figure 2, the LaTIn algorithm consists in a successive resolution of the two groups of equations. The use of two search directions which are parameters of the method ensure the transmission of the quantities between the two groups and the convergence of the method.

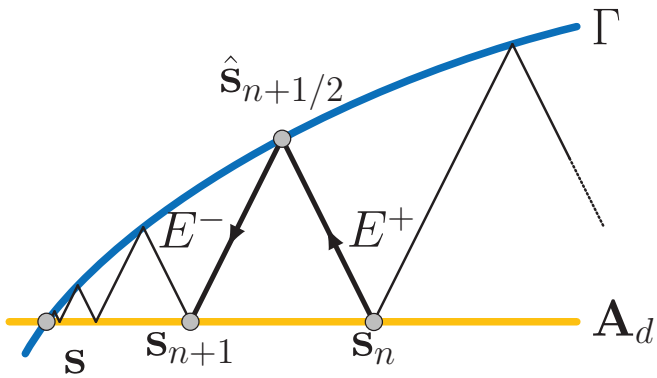


Figure 2: Schematic representation of the LATIN process

In our case we consider linear elastic behavior in the subdomains and the knowledge of the boundary magnitudes (forces and displacements) on each interface between two substructures are sufficient to define the solution: the solution into a substructure can be obtained by solving a classical elastic problem with consideration of the boundary magnitudes coming from the interfaces.

The specific feature of the LaTIn method is that on each iteration, the solver yields an approximate solution not only at all points of structure but also over the whole loading path. Along the following iterations the solution is enriched and a convergence criterion is used to check the convergence rate of the algorithm.

3.2 The MultiParametric Strategy

The MultiParametric Strategy (MPS) takes advantage of the previous feature of the LaTIn algorithm: along the resolutions associated to different set of parameters, a set of converged solutions is build and enriched and if a new calculation associated to a new set of parameters is requested, the LaTIn algorithm is initialized with the "best" previous converged solution. Thanks to this initialization the requested solution converges in fewer iterations of the algorithm and, therefore, in less time. Figure 3 shows that each variation of the parameters on interface leads to the appearance of a new associated group of equations Γ and the associated solution is quickly obtain with the reinitialization. The remained difficulty is the choice of the "best" previous

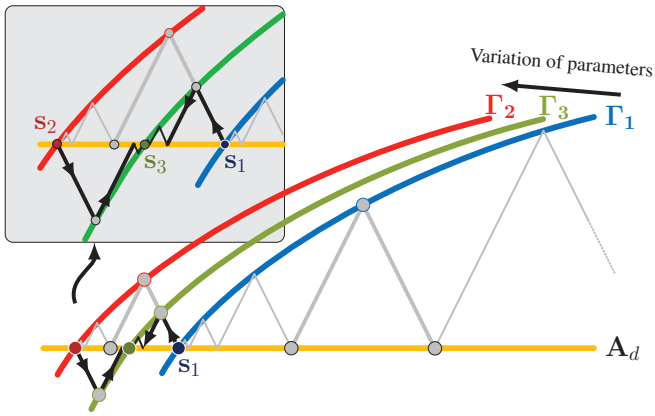


Figure 3: The MultiParametric Strategy to obtain solutions $\mathbf{s_2}$ and $\mathbf{s_3}$ respectively from the solution $\mathbf{s_2}$ and $\mathbf{s_2}$ or $\mathbf{s_3}$

converged solution. Some indications for choosing the best initialization strategy were given in [9]. In our case we will use only a "closest point" strategy: the initial solution of the new problem is chosen to be the converged solution associated with the set of design parameters which is closest to the one being considered.

Some details and performance of the MPS can be found on [4, 20].

4 Metamodels

In order to achieve an inexpensive global optimization using specific algorithm like Particle Swarm Optimizer [10] or Genetic Algorithm [11] one must be able to reduce the computational cost to compute each evaluation of the objective function. This kind of optimizers needs a lot of evaluations for converging and in our case, the computational cost of each mechanical computation increases not only with the complexity of the mechanical problem (*i.e.* the nonlinearities included in the model), but also with the number of degrees of freedom in the mechanical system. One way is to use for example Reduce Order Modeling (see for instance [12]) to reduce the computation cost. Here we propose the use of metamodels and more precisely metamodels which are

able to take into account the derivatives of the actual function. The feature allows us to take advantage of the MultiParametric Strategy to reduce the computation time for computing the derivatives of the objective function using a finite difference method.

4.1 Notations

We will use the following notations:

- $\mathbf{x}^{(i)}$, $i \in [\![0, n_s]\!]$ denotes a point in the design space $\mathscr{D}$. ($\mathbf{x}^{(i)}$, $i \in [\![1, n_s]\!]$ is one of the n_s sample points, and $\mathbf{x}^{(0)}$ is an arbitrary point in the design space, which may or may not be a sample point.)
- $Y(\mathbf{x}^{(i)})$ and $\widetilde{Y}(\mathbf{x}^{(i)})$ denote respectively the response of the analytical function (or the response of the mechanical model) and the approximate response given by the metamodel at point $\mathbf{x}^{(i)}$, $i \in [\![0, n_s]\!]$.
- $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is a correlation function expressing the correlation relation between points $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, $(i, j) \in [\![0, n_s]\!]^2$.
- $K(r)$ denotes a radial basis function.

4.2 Gradient-based metamodels

Several kinds of metamodels has gradient-enhanced build [13]. We propose in this paper to use only gradient-based kriging and gradient-based radial basis functions. This two classes of metamodels will be briefly presented and applied on analytical and mechanical examples.

4.2.1 Cokriging

Cokriging is very similar to kriging metamodel also called Gaussian-process metamodel in context of computer experiments. Introduce in the 50's to achieve geological study, kriging has been developped in the 60's by G. Matheron [14]. Sacks [15] introduces the use of the kriging to deal with optimization problem. Cokriging allows us to build richer metamodel by taking into account primary and auxiliary variables evaluated on several sample points obtained with some technique such as Latin Hypercube Sampling [3].

The main idea of the kriging is to look for the actual function Y (which the deterministic response provided by the mechanical model in our case) as a realization of a Gaussian stochastic process: Y is considered as the sum of a deterministic function μ (which represents the trend of the data) and a stationary Gaussian process Z with known stationary covariance (which represents the fluctuations around the trend). The cokriging keeps this decomposition for the primary variable but introduces the same decomposition for the n_d auxiliary variables W^i (see Equation 1).

$$\forall i \in [\![0, n_d]\!], \forall \mathbf{x}^{(0)} \in \mathscr{D},\ Y(\mathbf{x}^{(0)}) = \mu(\mathbf{x}^{(0)}) + Z(\mathbf{x}^{(0)}) \\ W^i(\mathbf{x}^{(0)}) = \mu_{W^i}(\mathbf{x}^{(0)}) + Q^i(\mathbf{x}^{(0)}) \tag{1}$$

In our case, the primary and the auxiliary variables are respectively the objective function and its derivatives (as deterministic functions) such as

$$\forall i \in [\![0, n_d]\!], \forall \mathbf{x}^{(0)} \in \mathscr{D},\ W^i(\mathbf{x}^{(0)}) = \frac{\partial Y}{\partial x_i}\left(\mathbf{x}^{(0)}\right) \tag{2}$$

So n_d denotes the number of design variables. The following equations presents the stochastic characteristics of the different quantities:

$$\forall i \in [\![0, n_d]\!], \forall \mathbf{x}^{(0)} \in \mathscr{D},\ \mathbb{E}\left[Y(\mathbf{x}^{(0)})\right] = \mu(\mathbf{x}^{(0)}),\ \mathbb{E}\left[Z(\mathbf{x}^{(0)})\right] = 0, \tag{3}$$

$$\mathrm{Var}\left[Y(\mathbf{x}^{(0)})\right] = \mathrm{Var}\left[Z(\mathbf{x}^{(0)})\right] \neq 0, \tag{4}$$

$$\mathbb{E}\left[W^i(\mathbf{x}^{(0)})\right] = \mu_{W^i},\ \mathbb{E}\left[Q^i(\mathbf{x}^{(0)})\right] = 0, \tag{5}$$

$$\mathrm{Var}\left[W^i(\mathbf{x}^{(0)})\right] = \mathrm{Var}\left[Q^i(\mathbf{x}^{(0)})\right] \neq 0. \tag{6}$$

and

$$\forall \left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) \in \mathscr{D}^2, \forall (k,l) \in [\![0, n_d]\!]^2,$$

$$\mathrm{cov}\left[Z\left(\mathbf{x}^{(i)}\right), Z\left(\mathbf{x}^{(j)}\right)\right] = \sigma^2 R\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sigma^2 c_{ij} \tag{7}$$

$$\mathrm{cov}\left[\frac{\partial Z}{\partial x_k}\left(\mathbf{x}^{(i)}\right), Z\left(\mathbf{x}^{(j)}\right)\right] = -\sigma^2 \frac{\partial R}{\partial x_k}\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sigma^2 c_{kij} \tag{8}$$

$$\mathrm{cov}\left[Z\left(\mathbf{x}^{(i)}\right), \frac{\partial Z}{\partial x_k}\left(\mathbf{x}^{(j)}\right)\right] = \sigma^2 \frac{\partial R}{\partial x_k}\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sigma^2 c_{kji} \tag{9}$$

$$\mathrm{cov}\left[\frac{\partial Z}{\partial x_k}\left(\mathbf{x}^{(i)}\right), \frac{\partial Z}{\partial x_l}\left(\mathbf{x}^{(j)}\right)\right] = -\sigma^2 \frac{\partial^2 R}{\partial x_k \partial x_l}\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sigma^2 c_{kilj} \tag{10}$$

$\mathbb{E}$, Var and cov are the classical statistical expected value, variance and covariance.

Moreover the proposed cokriging is called ordinary cokriging: the deterministic function μ is considered as a unknown constant (by contrast with simple and universal cokriging where μ is respectively a known constant and polynomial function). So the deterministic term of the decomposition for the auxiliary variable is null (it stems from the derivative of μ). Equations 7-10 show the covariance relations [16] such as the covariance structure is a function of a generalized distance among the sample responses.

To describe the covariance structure several parametric correlation functions R can be used (linear, gaussian ...). In our case Matérn [17] functions are used.

Due to the fact that we know only the value of the function on several sample points, we can not determinate exactly the function Y. Therefore we are looking for

an estimator $\widetilde{Y}$ of the actual function by considering the following linear predictors of the non-sample point $\mathbf{x}^{(0)}$:

$$\widetilde{Y}\left(\mathbf{x}^{(0)}\right) = \sum_{i=1}^{n_s} \lambda_{0i}\left(\mathbf{x}^{(0)}\right) Y\left(\mathbf{x}^{(i)}\right) + \sum_{j=1}^{n_d}\sum_{i=1}^{n_s} \lambda_{ij}\left(\mathbf{x}^{(0)}\right) \frac{\partial Y}{\partial x_j}\left(\mathbf{x}^{(i)}\right) \tag{11}$$

So the construction of the cokriging metamodel consists in the determination of the coefficients λ_{ij} (with $(i,j) \in [\![0,n_s]\!] \times [\![1,n_s]\!]$) such as the estimator is unbiased (Equation 12) and it minimizes the mean square error (Equation 13).

$$\mathbb{E}\left[\widetilde{Y}\left(\mathbf{x}^{(0)}\right)\right] = \mathbb{E}\left[Y\left(\mathbf{x}^{(0)}\right)\right] \tag{12}$$

$$\mathrm{MSE}\left[\widetilde{Y}(\mathbf{x}^{(0)})\right] = \mathbb{E}\left[\left(\widetilde{Y}\left(\mathbf{x}^{(0)}\right) - Y\left(\mathbf{x}^{(0)}\right)\right)^2\right] \tag{13}$$

Finally the estimator of the ordinary cokriging is obtained on a unsample point $\mathbf{x}^{(0)}$ of the design space.

$$\widetilde{Y}\left(\mathbf{x}^{(0)}\right) = \underbrace{\hat{\beta}_c}_{\mu(\mathbf{x}^{(0)})} + \underbrace{\mathbf{r}_{c0}^T \mathbf{C}_\mathbf{c}^{-1}(\mathbf{Y}_\mathbf{sc} - \mathbf{X}_c\hat{\beta}_c)}_{Z(\mathbf{x}^{(0)})} \tag{14}$$

where $\hat{\beta}_c$ is estimated using the generalized least squares method, $\mathbf{r}_{c0}$ is the correlation vector between the unsample point $\mathbf{x}^{(0)}$ and the sampled point $\mathbf{x}^{(i)}$ ($\forall i \in \mathscr{D}$), $\mathbf{C}_\mathbf{c}$ is the correlation matrix, $\mathbf{Y}_\mathbf{sc}$ contains both the responses and the gradients of the objective function at the sample points and $\mathbf{X}_c$ contains n_s ones et $n_s \times n_d$ zeros. More details of the building process can be found in [18, 19, 20].

As we show previously the model includes several parameters such as the characteristic correlation lengths and the variance σ of the random process Z. These parameters can be determined by maximizing the likelihood function [21].

4.3 Radial Basis Function Network

In order to build an approximation of a function, an other strategy is to use Radial basis Function [5]. The main idea of this approach is to introduce a set of n_s basis functions K_i that we consider associated and centered on each sample point. Firstly introduce to responses only, this approach has been extended to include responses and gradients [13] and thereby the metamodel provides a better approximation than whithout the integration of the derivatives.

The approximation is determinate as the form presented by the Equation 15 for the non-derivative approach and by the Equation 16 for the derivative-based approach.

$$\forall \mathbf{x}^{(0)} \in \mathscr{D},\ \widetilde{Y}(\mathbf{x}^{(0)}) = \sum_{i=1}^{n_s} w_i K_i(\mathbf{x}^{(0)}) = \sum_{i=1}^{n_s} w_i K_{i0} \tag{15}$$

$$\begin{aligned}\forall \mathbf{x}^{(0)} \in \mathscr{D},\ \widetilde{Y}(\mathbf{x}^{(0)}) &= \sum_{i=1}^{n_s} w_i K_i(\mathbf{x}^{(0)}) + \sum_{j=1}^{n_d}\sum_{i=1}^{n_s} w_{ij}\frac{\partial K_i}{\partial x_j}(\mathbf{x}^{(0)}) \\ &= \sum_{j=0}^{n_d}\sum_{i=1}^{n_s} w_{ij} K_{i0,j}\end{aligned} \tag{16}$$

where

$$\forall i \in [\![1,n_s]\!],\ \forall j \in [\![0,n_d]\!],\ \ w_{ij} = \begin{cases} w_{i0} = w_i & \text{if } j=0 \\ w_{ij} & \text{otherwise}\end{cases} \tag{17}$$

and

$$\begin{aligned}&\forall \mathbf{x}^{(0)} \in \mathscr{D},\ \forall (i,j,k) \in [\![1,n_s]\!]\times[\![0,n_s]\!]\times[\![0,n_d]\!], \\ &K_{ij,k} = \begin{cases} K_{ij,0} = K_{ij} = K(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) & \text{if } k=0 \\ K_{ij,k} = \dfrac{\partial K_{ij}}{\partial x_k} = \dfrac{\partial K}{\partial x_k}(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) & \text{otherwise}\end{cases}\end{aligned} \tag{18}$$

The function K is a radial basis function (for instance a linear, cubic, Gaussian radial basis function ...). We also introduce the notation $K_{ij,kl}$ to denoted the secondary derivatives of the radial basis functions.

The building process consists here in the determination of the coefficients w_{ij}. To do this the approximation must respect the interpolation conditions proposed in Equations 19 for the two metamodels and 19-20 for the gradient-based approach.

$$\forall (i,j) \in [\![1,n_s]\!]\times[\![1,n_d]\!],\ \forall \mathbf{x}^{(i)} \in \mathscr{D},,$$

$$\widetilde{Y}(\mathbf{x}^{(i)}) = \widetilde{Y}_i = Y_i = Y(\mathbf{x}^{(i)}) \tag{19}$$

$$\frac{\partial \widetilde{Y}}{\partial x_j}(\mathbf{x}^{(i)}) = \widetilde{Y}_{i,j} = Y_{i,j} = \frac{\partial Y}{\partial x_j}(\mathbf{x}^{(i)}) \tag{20}$$

These conditions lead to the matrix formulations presented on Equation 21 and 22.

$$\mathbf{K}\mathbf{w} = \mathbf{Y} \tag{21}$$

$$\mathbf{K}_g\mathbf{w}_g = \mathbf{Y}_g \tag{22}$$

where

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{21} & \cdots & K_{n_s 1} \\ K_{12} & K_{22} & \cdots & K_{n_s 2} \\ \vdots & & \ddots & \vdots \\ K_{1n_s} & K_{2n_s} & \cdots & K_{n_s n_s}\end{bmatrix}\ \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n_s}\end{bmatrix}\ \mathbf{Y} = \begin{bmatrix} Y(\mathbf{x}^{(1)}) \\ Y(\mathbf{x}^{(2)}) \\ \vdots \\ Y(\mathbf{x}^{(n_s)})\end{bmatrix} \tag{23}$$

and

$$
\begin{aligned}
\mathbf{K}_g &= \begin{bmatrix} \mathbf{K} & \mathbf{K}_d \\ \mathbf{K}_d^T & \mathbf{K}_{dd} \end{bmatrix} \text{ where } \mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n_s} \\ K_{21} & K_{22} & \cdots & K_{2n_s} \\ \vdots & & \ddots & \vdots \\ K_{n_s1} & K_{n_s2} & \cdots & K_{n_sn_s} \end{bmatrix} \\
\mathbf{K}_d &= \begin{bmatrix} K_{11,1} & K_{11,2} & \cdots & K_{11,n_d} & K_{12,1} & K_{12,2} & \cdots & K_{1n_s,n_d} \\ K_{21,1} & K_{21,2} & \cdots & K_{21,n_d} & K_{22,1} & K_{22,2} & \cdots & K_{2n_s,n_d} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ K_{n_s1,1} & K_{n_s1,2} & \cdots & K_{n_s1,n_d} & K_{n_s2,1} & K_{n_s2,2} & \cdots & K_{n_sn_s,n_d} \end{bmatrix} \\
\mathbf{K}_{dd} &= \begin{bmatrix} K_{11,11} & K_{11,12} & \cdots & K_{11,1n_d} & K_{12,11} & K_{12,12} & \cdots & K_{1n_s,1n_d} \\ K_{11,21} & K_{11,22} & \cdots & K_{11,2n_d} & K_{12,21} & K_{12,22} & \cdots & K_{1n_s,2n_d} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ K_{11,n_d1} & K_{11,n_d2} & \cdots & K_{11,n_dn_d} & K_{12,n_d1} & K_{12,n_d2} & \cdots & K_{1n_s,n_dn_d} \\ K_{21,n_d1} & K_{21,n_d2} & \cdots & K_{21,n_dn_d} & K_{22,n_d1} & K_{22,n_d2} & \cdots & K_{2n_s,n_dn_d} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ K_{n_s1,n_d1} & K_{n_s1,n_d2} & \cdots & K_{n_s1,n_dn_d} & K_{n_s2,n_d1} & K_{n_s2,n_d2} & \cdots & K_{n_sn_s,n_dn_d} \end{bmatrix} \\
\mathbf{Y}_g &= \begin{bmatrix} Y_1 & Y_{1,1} & Y_{1,2} & \cdots & Y_{1,n_d} & Y_2 & Y_{2,1} & \cdots & Y_{n_s,n_d} \end{bmatrix}^T
\end{aligned}
\tag{24}
$$

The gradient-based approach leads to take into account a $(n_d+1)n_s \times (n_d+1)n_s$ square matrix $\mathbf{K}_g$ instead of the $n_s \times n_s$ square matrix $\mathbf{K}$ for the classical approach. This feature results in a high dimensional problem when we consider many sample points and many design variables.

The building of the RBF approach is easier to implement and less expensive in term of CPU time than kriging. In our case we consider a Matérn radial basis function and the shape parameters of this functions are determined using a *leave-one-out* strategy [22, 23]: the main idea of this strategy is to build the prediction of the response at the sample point $\mathbf{x}^{(i)}$ using $n_s - 1$ sample points (the original database without the sample point $\mathbf{x}^{(i)}$). The approximate response is denotes $\widetilde{Y}_{-i}(\mathbf{x}^{(i)})$. Then a classical prediction error is used: $e_i = (\widetilde{Y}_{-i}(\mathbf{x}^{(i)}) - Y(\mathbf{x}^{(i)}))^2$. Each sample point is successively remove of the database and the associated error is computed. This approach leads to a Mean Square Error (Equation 25) that we minimize to determine the best shape parameters of the radial basis function.

$$
MSE_{LOO} = \frac{1}{n_s} \sum_{i=1}^{n_s} (\widetilde{Y}_{-i}(\mathbf{x}^{(i)}) - Y(\mathbf{x}^{(i)}))^2 \tag{25}
$$

When we build gradient-based RBF, MSE_{LOO} is evaluated by considering $n_s - 1$ responses **and** gradients.

4.4 Validation

In the context of metamodeling one very important point is to be able to evaluate the quality of the approximation provided by the metamodel. The first idea is to use the actual response for evaluating the performance of the metamodel by computing the objective function on many points on a grid and by computing the distance between the approximate and the actual function. In the following example we propose to use this metric because the computational time remains acceptable. Here a grid with n_c points ($n_c >> n_s$) is considered and different metrics are calculated (see Equations 26-31):

$$R^2 = 1 - \frac{SS_{err}}{SS_{tot}} \text{ where } SS_{err} = \sum_{i=1}^{n_c} \left[Y(\mathbf{x}^{(i)}) - \widetilde{Y}(\mathbf{x}^{(i)}) \right]^2 \tag{26}$$

$$\text{and } SS_{tot} = \sum_{i=0}^{n_c} \left[Y(\mathbf{x}^{(i)} - \overline{Y} \right]^2$$

$$RAAE = \frac{1}{n_c \sigma^2_{\widetilde{Y}}} \sum_{i=1}^{n_c} |Y(\mathbf{x}^{(i)}) - \widetilde{Y}(\mathbf{x}^{(i)})| \tag{27}$$

$$RMAE = \frac{1}{\sigma^2_{\widetilde{Y}}} \max_{i \in [\![0, n_c]\!]} |Y(\mathbf{x}^{(i)}) - \widetilde{Y}(\mathbf{x}^{(i)})| \tag{28}$$

$$\mathtt{Q}_1 = \sup_{i \in \{1,2,\ldots,n_c\}} e_i = \|\mathbf{e}\|_\infty \tag{29}$$

$$\mathtt{Q}_2 = \sum_{i=1}^{n_c} e_i = \|\mathbf{e}\|_1 \tag{30}$$

$$\mathtt{Q}_3 = \frac{\mathtt{Q}_2}{n_c} \text{ (the average of } \mathtt{Q}_2) \tag{31}$$

$$\text{where } \forall i \in \{1,2,\ldots,n_c\},\ e_i = \frac{\left(Y\left(\mathbf{x}^{(i)}\right) - \widetilde{Y}\left(\mathbf{x}^{(i)}\right) \right)^2}{\sup\limits_{j \in \{1,2,\ldots,n_c\}} Y\left(\mathbf{x}^{(j)}\right)^2}$$

$$\text{and } \mathbf{e} = \begin{bmatrix} e_1 & e_2 & \cdots & e_{n_c} \end{bmatrix}$$

with $\|\cdot\|_1$ and $\|\cdot\|_\infty$ being the L^1 norm and infinity norm respectively.

In the classical context of expensive functions, this first approach is not viable. Therefore a classic way is to use k-fold Cross-Validation procedure as presented in the previous section (see Section 4.3) in the case of 1-fold or *Leave-One-Out* approach. In the following examples this Leave-One-Out strategy is achieved and the Mean Square Error (Equation 25), the minimum, maximum and average SCVR (Standardized Cross-Validated Residual, Equation 32 and see [22, 24]) criteria will be evaluated:

$$\forall i \in [\![1, n_s]\!],\ SCVR_i = \frac{Y(\mathbf{x}^{(i)}) - \widetilde{Y}_{-i}(\mathbf{x}^{(i)})}{\sigma^2_{-i}(\mathbf{x}^{(i)})} \tag{32}$$

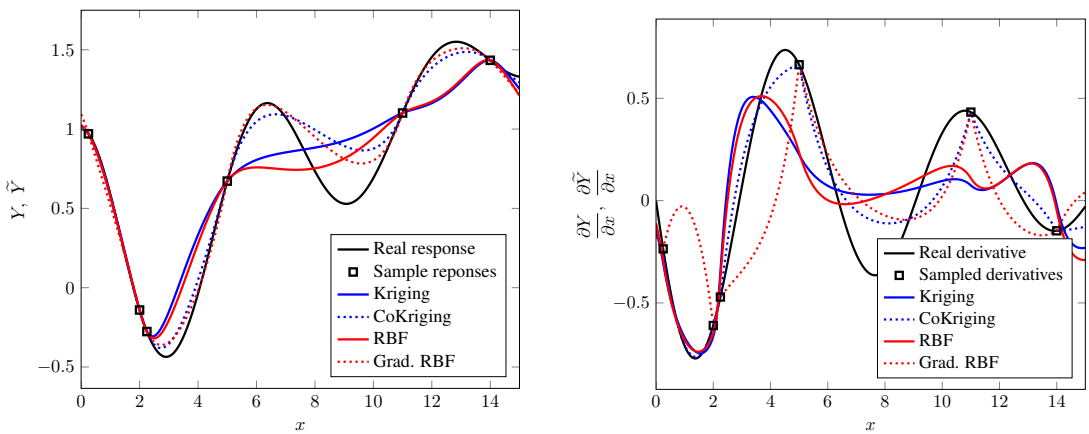
This new criteria will be only evaluated for kriging-based metamodels.

5 Examples

In this Section metamodels will be built on few analytical test-case. Firstly a 1D function is considered and kriging and RBF metamodels will be built to illustrate their behaviors. Then one two-dimensional irregular function will be introduced and for both RBF and kriging metamodels a matérn basis function and a matérn correlation function will be respectively used. Finally the metamodels will be coupled with the MultiParametric Strategy on three- and four-dimensional mechanical examples.

5.1 Analytical test-cases

First, we consider the analytical function $y(x) = \exp(-x/10)\cos(x) + x/10$ and we used 6 sampled responses to build kriging-based and RBF-based metamodels. The 6 sample points are chosen in the design space using Latin Hypercube Sampling (LHS) [3]. Figure 4 shows the capability of the gradient-enhanced metamodels to provided a



(a) Responses of the metamodels

(b) Derivatives of the metamodels

Figure 4: The classic and gradient-based kriging and RBF metamodels and their derivatives

very good approximation of this one-dimensional test-case.

Now, an irregular function will be used: the *six-hump camelback function*[1]. The response surface of the actual function is presented on Figure 5. 16 points are sampled in the design space using Latin Hypercube Sampling. Responses and gradients are analytically evaluated on these sample points. In order to compare the quality the metamodels they have been building using the same information obtained on the same sample points. Figures 6 and 7 show the response surfaces associated to the metamodels.

These two test-cases show that taking into account the derivatives allows us to build more accurate metamodels then the classical approachs. This feature is due to

[1] $\forall (x_1, x_2) \in [-2,2] \times [-1,1]$, $f(x_1,x_2) = (4 - 2.1x_1^2 + x_1^4)x_1^2 + x_1x_2 + 4(x_2^2 - 1)x_2$

	Classic		Gradient-based	
	RBF	Kriging	RBF	Kriging
MSE	$4.48 \cdot 10^{-2}$	$5.24 \cdot 10^{-2}$	$1.07 \cdot 10^{-2}$	$1.75 \cdot 10^{-2}$
R^2	0.865	0.866	0.973	0.957
R^2_{adj}	0.859	0.865	0.973	0.957
RAAE	0.361	0.395	0.119	0.150
RMAE	0.874	0.890	0.518	0.670
Q_1	$6.95 \cdot 10^{-2}$	$7.00 \cdot 10^{-2}$	$3.51 \cdot 10^{-2}$	$5.62 \cdot 10^{-2}$
Q_2	3.73	4.36	0.894	1.45
Q_3	$1.87 \cdot 10^{-2}$	$2.18 \cdot 10^{-2}$	$4.47 \cdot 10^{-3}$	$7.26 \cdot 10^{-4}$
MSE_{LOO}	0.168	0.115	0.108	0.150
Avg. SCVR	–	$5.40 \cdot 10^{-15}$	–	2.07
Min. SCVR	–	-2.99	–	-3.26
Max. SCVR	–	6.87	–	16.45

Table 1: Characteristics of the four previous metamodels associated to the analytical one-dimensional test-case

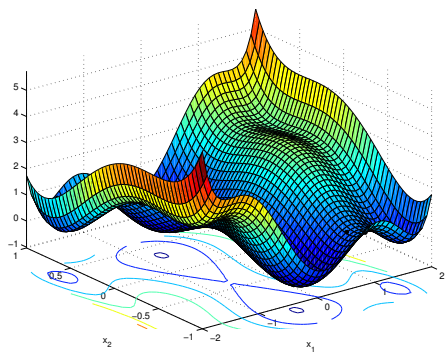
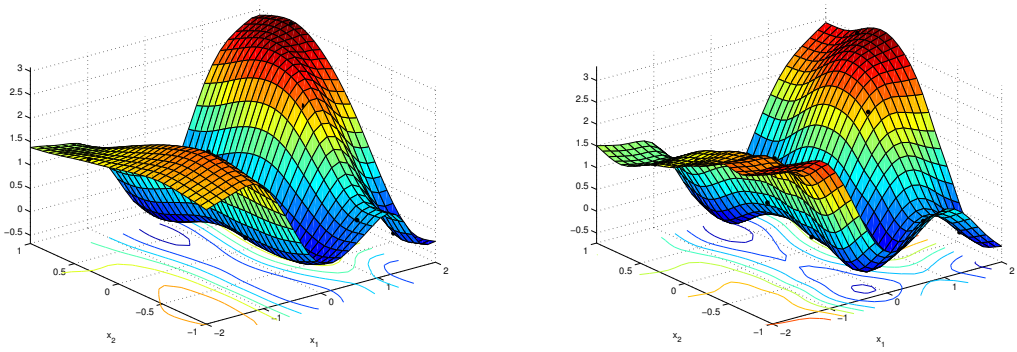


Figure 5: The response surface of the actual function

the fact that these both metamodels interpolate not only the responses but also the gradients. On the one-dimensional test-case, this assessment is very impressive and with only six responses and gradients we are able to build almost the actual function. On the two-dimensional test-case the cokriging metamodel provides a very accurate approximation. This approximation is slightly better than the gradient-based RBF (as shown on Table 2).

5.2 Mechanical test-case

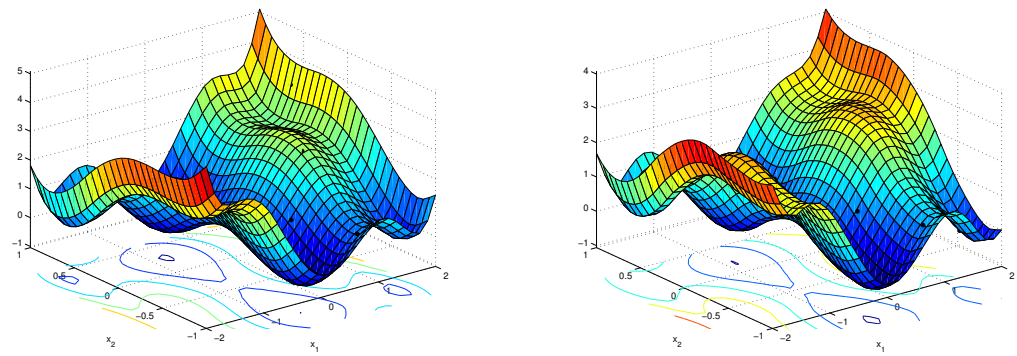
Let us consider the example of a quasi-static academic problem which was presented in [4, 20]. Figure 8 show the new geometry of the problem, which consists of four subdomains ($h = 50mm$, Young's modulus $E = 2 \cdot 10^5 MPa$ and Poisson's coefficient



(a) Kriging metamodel

(b) RBF metamodel

Figure 6: The non-gradient-based metamodels with 16 evaluations



(a) Cokriging metamodel

(b) Gradient-based RBF metamodel

Figure 7: The gradient-based metamodels obtained with 16 evaluations and 16 gradients

$\nu = 0.3$). We consider:

- contact with friction on interfaces $\Omega_1 - \Omega_2$, $\Omega_2 - \Omega_3$ and $\Omega_2 - \Omega_4$ with the respective friction coefficients μ_1, μ_2 and μ_3.
- unilateral contact on interfaces $\Omega_1 - \Omega_4$ and between Ω_2 and the rigid wall associated to two gaps j_1 and j_2.

Each part is represented by a single substructure discretized using bilinear quadrangles. The loading consists of two stages: first, a progressive vertical pressure P_1 up to a maximum of $50MPa$ applied at the top of substructure Ω_3 (the preloading stage), then a progressive horizontal load from 0 to $30MPa$ applied to substructure Ω_2. From this problem we have built two parametric studies: firstly the friction coefficient μ_2 (0.1) and the gap j_2 ($24\mu m$) are fixed and the parameters which vary are the two

	Classic		Gradient-based	
	RBF	Kriging	RBF	Kriging
MSE	0.330	0.450	0.150	$\mathbf{2.33 \cdot 10^{-2}}$
R^2	0.796	0.726	0.925	**0.992**
R^2_{adj}	0.795	0.726	0.925	**0.992**
RAAE	0.365	0.447	0.184	$\mathbf{7.27 \cdot 10^{-2}}$
RMAE	2.71	4.01	2.32	**1.00**
Q_1	0.247	0.479	0.208	$\mathbf{4.30 \cdot 10^{-2}}$
Q_2	9.12	12.3	4.10	**0.639**
Q_3	$1.01 \cdot 10^{-3}$	$1.37 \cdot 10^{-2}$	$4.55 \cdot 10^{-3}$	$\mathbf{7.10 \cdot 10^{-4}}$
MSE_{LOO}	0.306	0.450	0.198	0.151
Avg. SCVR	–	$1.83 \cdot 10^{-15}$	–	-1.10
Min. SCVR	–	-3.53	–	-4.69
Max. SCVR	–	4.82	–	3.33

Table 2: Characteristics of the four metamodels associated with the analytical two-dimensional test-case

friction coefficients μ_1 ($[0, 0.95]$) and μ_3 ($[0, 0.95]$) and the gap j_1 ($[-28\mu m, 48\mu m]$). For the second study only the gap j_2 ($24\mu m$) is fixed and we consider four parameters which are the three friction coefficients μ_1, μ_2 and μ_3 ($[0, 0.95]$) and the gap j_1 ($[-28\mu m, 48\mu m]$). For the two studies the observed quantity of interest (the objective function) will be the reaction force of the subdomain Ω_2 on the rigid wall.

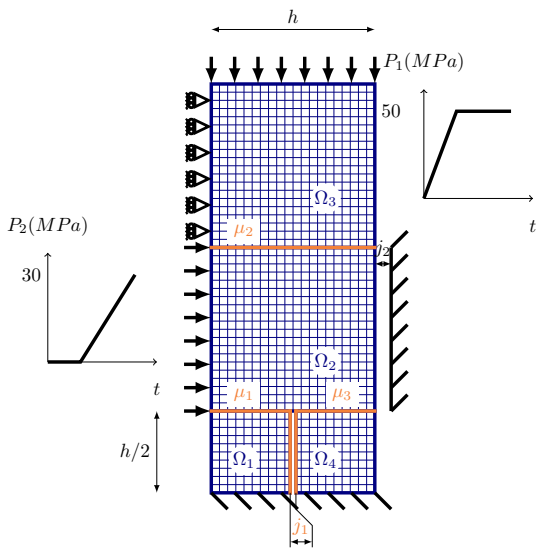


Figure 8: The geometry of the problem

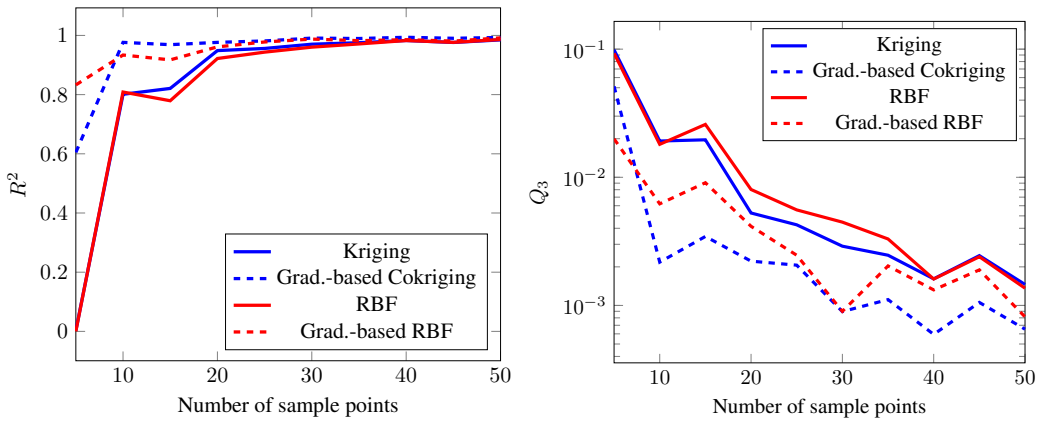
In order to illustrate the performance of the MPS and the gradient-based metamodels we propose to sampled the design space using LHS from 5 to 50 sample points

and compute the responses and the gradients of the objective function on each sample points (we built two sets: one with only the responses and one with the responses and the gradients). The metamodels with and without derivatives will be built and using reference responses obtained on two regular grids (1000 and 1296 sample points for respectively the three-variables and four-variables examples) qualities of the approximations will be compared. We proposed also to study the computational cost associated to the mechanical computations to obtained gradient- and non-gradient-based metamodels.

5.3 Quality of the approximations

Figures 9 and 10 show the quality of the approximations obtained with the four kinds of metamodels. We can observe the non-gradient-based metamodels provide very close quality although the gradient-based metamodels are significantly better. For this mechnical problem we can conclude that

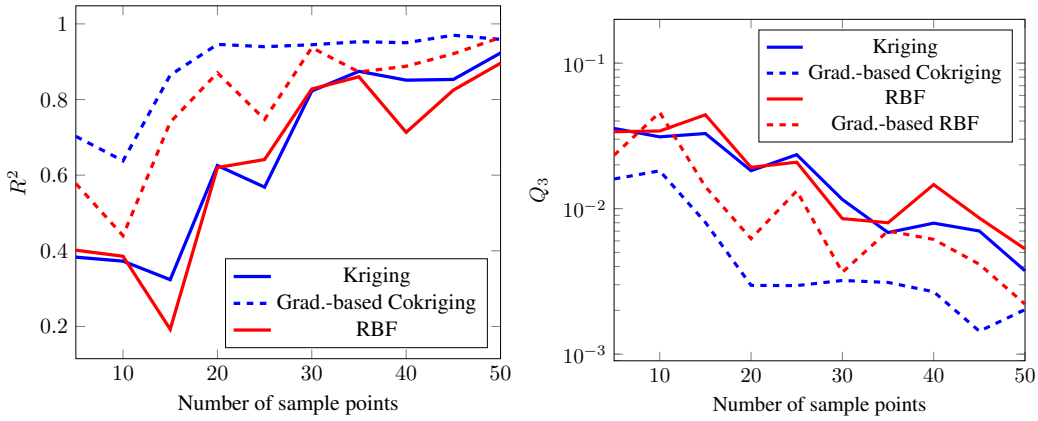
- for large number of sample points all the metamodels are accurate and have very close quality.
- for small number of sample points gradient-based metamodels are significantly better. For example to obtain the quality criteria about $3 \cdot 10^{-3}$ ($\mathtt{Q}_3$) we need to use responses and gradients on 10 sample points for the gradient-based metamodels whereas, for non-gradient-based metamodel, data on about 35 sample points are needed (for the three-dimensional test-case).



(a) R^2 (b) $\mathtt{Q}_3$

Figure 9: Measures of the quality of the metamodels build (three-dimensional test-case) versus the number of sample points

These figures are very bumpy. This assesment is due to the fact that the sample points are obtained using a Latin Hypercube Sampling that is provided one set of sample points but not necessarily the best to build the best approximation. The second fact



(a) R^2 (b) Q_3

Figure 10: Measures of the quality of the metamodels build (four-dimensional test-case) versus the number of sample points

is observed for the gradient-based RBF: the RBF parameters is determined using the minimization of a *leave-one-out* mean-square error that is very multimodal. Thus the previous figures show that cokriging is more stable than the gradient-based RBF metamodel.

5.4 Computational cost

Figures 11 and 12 show the computational cost associated to the building of non-gradient-based and gradient-based metamodels. The gain presented here describes the gain associated with the use of the MultiParametric Strategy and is calculated using the following formula (Eq. 33).

$$\text{Gain} = \frac{\text{Number of Calculations} \times \text{CPU Time of the first calculation}}{\text{CPU Time with the MPS}} \tag{33}$$

In Figure 11 and 12, the gain of responses and gradients are calculated using Eq. 33 but the number of calculations that we take into account is n_s for the responses gains and $n_s \times n_d$ for the gradients gains.

Here we can observe that the MPS allows use to compute the gradients of the non-linear mechanical problems very inexpensively. Although we use a first-order finite difference method to compute the gradients and the fact that n_d responses of the objective function are needed to obtain the gradient on one sample point, the required time to evaluate the gradients is the same than the time require to evaluate the entire responses. This feature is very useful to build gradient-based metamodel: if we reuse the previous example (cf. Item 2 in Section 5.3) the CPU time to build the non-gradient-based metamodels is mostly half as much again as the CPU time to build gradient-based metamodels.

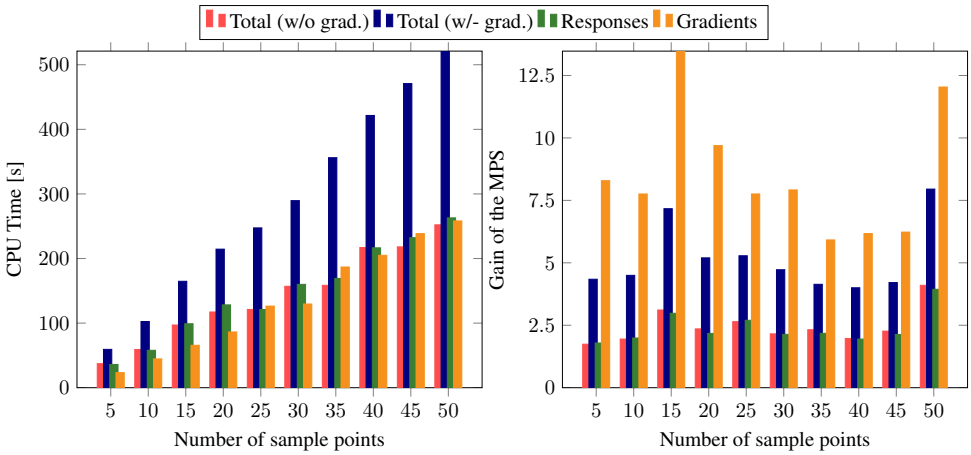


Figure 11: CPU time and gain of the MPS associated to the building of the metamodels for the three-dimensional test-case

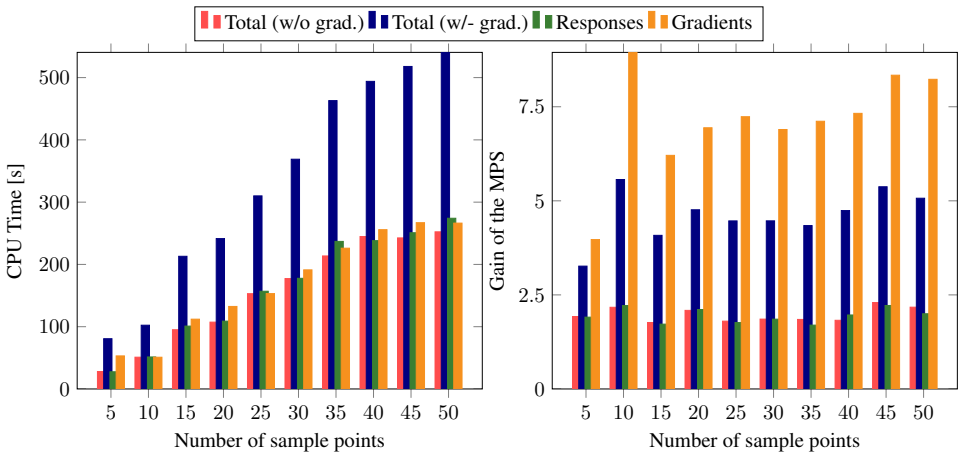


Figure 12: CPU time and gain of the MPS associated to the building of the metamodels for the four-dimensional test-case

6 Conclusion

This paper focuses on the building of gradient-based metamodels and the cost associated to this building process in terms of mechanical CPU time. Gradient-based and non-gradient-based metamodels are compared using analytical and mechanical examples. In order to deal with non-linear mechanical problems, a dedicated strategy called multiparametric strategy based on the LaTIn method is used and allows us to reduce significantly the computational time. This feature makes the use of derivatives information acceptable in terms of CPU time. Finally the proposed strategy enables one to build richer approximation for a fixed computational time.

In the context of the multilevel optimization presented in this paper, the use of gradient-based metamodel coupled with the multiparametric strategy will enable one

to reduce the computation cost associated to the approximate optimization of the first level but also to decrease significantly the computational time the accurate optimization of the second level.

Acknowledgment

This work was supported by the *Agence Nationale de la Recherche* as part as ANR-08-COSI-007-10: Distributed Multi-Disciplinary Optimization.

References

[1] A. Booker, J. Dennis, P. Frank, D. Serafini, V. Torczon, M. Trosset, "A rigorous framework for optimization of expensive functions by surrogates", *Structural and Multidisciplinary Optimization*, 17(1): 1–13, 1999.

[2] N. Queipo, R. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. Tucker, "Surrogate-based analysis and optimization", *Progress in Aerospace Sciences*, 41(1): 1 – 28, 2005.

[3] M. McKay, W. Conover, R. Beckman, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code", *Technometrics*, 21(2): 239–245, 1979.

[4] P. Boucard, L. Champaney, "A suitable computational strategy for the parametric analysis of problems with multiple contact", *International Journal for Numerical Methods in Engineering*, 57(9): 1259–1281, 2003.

[5] R. Hardy, "Multiquadric Equations of Topography and Other Irregular Surfaces", *J. Geophys. Res.*, 76: 1905–1915, 1971.

[6] N. Cressie, *Statistics for spatial data*, John Wiley & Sons, New York, 1993.

[7] P. Ladevèze, *Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation*, Springer Verlag, 1999.

[8] C. Blanzé, L. Champaney, J. Cognard, P. Ladevèze, "A modular approach to structure assembly computations: application to contact problems", *Engineering Computations*, 13(1): 15–32, 1995.

[9] B. Soulier, P.A. Boucard, "Multilevel optimization using interpolation models coupled with a multiparametric strategy", in *4th European Conference on Computational Mechanics*, 2010.

[10] J. Kennedy, R. Eberhart, "Particle swarm optimization", in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Volume 4, pages 1942–1948. IEEE, 1995.

[11] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-wesley, 1989.

[12] C. Heyberger, P. Boucard, D. Néron, "Multiparametric analysis within the proper generalized decomposition framework", *Computational Mechanics*, pages 1–13, 2011.

[13] S. Leary, A. Bhaskar, A. Keane, "Global approximation and optimization using adjoint computational fluid dynamics codes", *AIAA journal*, 42(3): 631–641, 2004.

[14] G. Matheron, "Principles of geostatistics", *Economic geology*, 58(8): 1246, 1963.

[15] J. Sacks, S. Schiller, W. Welch, "Designs for computer experiments", *Technometrics*, 31(1): 41–47, 1989.

[16] M. Morris, T. Mitchell, D. Ylvisaker, "Bayesian design and analysis of computer experiments: use of derivatives in surface prediction", *Technometrics*, 35(3): 243–255, 1993.

[17] B. Matérn, *Spatial Variation (Lecture NotesStatist. 36)*, Springer, Berlin, 1960.

[18] H. Chung, J. Alonso, "Using gradients to construct cokriging approximation models for high-dimensional design optimization problems", in *40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*. Citeseer, 2002.

[19] S. Leary, A. Bhaskar, A. Keane, "A derivative based surrogate model for approximating and optimizing the output of an expensive computer simulation", *Journal of Global Optimization*, 30(1): 39–58, 2004.

[20] L. Laurent, P.A. Boucard, B. Soulier, "Generation of a Cokriging Metamodel using a Multiparametric Strategy", *Journal of Computational Mechanics*, doi:10.1007/s00466-012-0711-0 (to be published).

[21] K. Mardia, R. Marshall, "Maximum likelihood estimation of models for residual covariance in spatial regression", *Biometrika*, 71(1): 135, 1984.

[22] D. Jones, M. Schonlau, W. Welch, "Efficient global optimization of expensive black-box functions", *Journal of Global optimization*, 13(4): 455–492, 1998.

[23] S. Rippa, "An algorithm for selecting a good value for the parameter c in radial basis function interpolation", *Advances in Computational Mathematics*, 11(2): 193–210, Nov. 1999.

[24] A. Keane, P. Nair, "Computational approaches for aerospace design: the pursuit of excellence.", 2005.