Support vectors machines for the estimation of probability of failure: Multifidelity classifiers built from a posteriori discretization error estimators

Ludovic Pierre Jérôme Mell, Valentine Rey, Franck Schoefs

STRUCS 102321
Article reference: STRUCS STRUCS-D-22-00130
Support vectors machines for the estimation of probability of failure: multifidelity classifiers built from a posteriori discretization error estimators

Ludovic Mell[1], Valentine Rey[1,*], Franck Schoefs[1]

[1] *Nantes Université, École Centrale Nantes, CNRS, GeM, UMR 6183, F-44000 Nantes, France*
[*] *corresponding author valentine.rey@univ-nantes.fr*

Abstract

This article proposes two algorithms to compute the probability of failure using Support Vector Machine (SVM) classifiers and Monte Carlo estimator in the context of structural mechanics. The observations used to build the classifiers are obtained from calls to a finite element solver which introduces discretization error. By exploiting guaranteed discretization error estimators, the proposed methodology aim at computing an estimation of the probability of failure not polluted by this discretization error. The first algorithm builds two classifiers in parallel to separate the guaranteed fail population, the guaranteed safe population and the uncertain population. It enables to compute an upper bound and a lower bound of the exact probability of failure. The second algorithm only uses observations whose status (fail or safe) is guaranteed by the error estimators. It results in multi-fidelity SVM-based meta-models as observations computed on different mesh sizes can be used. Those two algorithms are illustrated on two-dimensional mechanical examples for different Monte Carlo populations.

Keywords: Reliability, Probability of failure, Support Vector Machines, Finite Element Method, Discretization Error

1. Introduction

Many industrial structures are used in an uncertain environment. For example, structures may be subjected to wind loads or wave loads that are not deterministic but random. Moreover, as the perfect knowledge of the structure itself is impossible, its geometry or its material properties can also be a source of uncertainty and are sometimes modeled as random variables. In

this context, the design of the structure requires taking into account those uncertainties, which is usually done by reliability analysis.

Reliability analysis consists in studying the ability of a structure to accomplish its function throughout its lifetime. A failure scenario is modeled by a performance function G, also called limit state function, which is the difference between a resistance and a solicitation. A negative limit state function corresponds to failure and a strictly positive state function corresponds to safety. Generally, the uncertainties on the mechanical structure (load, geometry, material properties) are modeled as random variables following identified distributions. As a consequence, reliability analysis consists in computing sensitivity factors, reliability indexes or a probability of failure, which is the probability that the performance function is negative. In this article, we focus on the estimation of the probability of failure.

Usually, for industrial structures, no explicit form of G with respect to the random variables is available. Therefore, even if the joint probability of the random variables is known, it is not possible to directly compute the probability of failure. However, a mechanical model of the structure is postulated and the arising equations can be solved, usually thanks to discretization techniques, such as the finite element method (FEM) [1]. FEM is nowadays largely used in commercial softwares and enables simulating and predicting the response of a structure. As a consequence, the probability of failure can be estimated using Monte Carlo estimators [2] in a non-intrusive way. In spite of their simplicity, these methods suffer a poor convergence rate: a precise estimation of the probability of failure (usually around 10^{-4}) requires the computation of $G(x)$, that is to say a call to the FEM code, for every point x of an extremely large Monte Carlo population. As a consequence, variance reduction techniques, such as importance sampling [3], subset sampling [4], Multi-Level Monte Carlo [5], adjusted control variates techniques [6] have been proposed. Another way to reduce computational costs consists in building a meta-model $\hat{G}$ that would be a satisfying cheap approximation of G. Kriging-based meta-models [7] are very popular in reliability analysis, but physical response surfaces [8], polynomial responses surfaces, radial basis functions or neuronal networks, see [9], can also be considered.

As the definition of the probability of failure only relies on the sign of G, classifiers are interesting candidates to build a meta-model. Among them, support vector machines (SVM) [10] are widely used in the context of reliability analysis, see [11, 12, 13, 14, 15]. The classifier enables separating the Monte Carlo population into two sub-populations: the failure population and the safety population. A learning criterion is available to improve the quality of the classifier from calls to the FEM code. The resulting algorithms are competitive and require comparable

number of observations comparing to standard kriging methods.

It is well-known that the result of FEM computations depends on the discretization [16]. A too coarse mesh might lead to wrong displacements and therefore introduces an error, called discretization error, in the computations of all the mechanical outputs (average stress, ...) leading to a misclassification. It is crucial to take into account this discretization error in the reliability analysis [17, 18]. To this day, only few works address the question of the consequence of the discretization error on the estimation of the probability of failure. It was done with response surface methodology [19], with FORM in [20] or in the context of kirging with co-kriging [21]. Discretization error estimators are available in the literature [16] and enable computing the error on a quantity of interest of the mechanical problem. Those estimators were exploited in the context of relibility within FORM [22] and kriging [23]. However, those approaches are limited to certain type of meta-models and do not include SVM.

In this paper, we propose to exploit a posteriori error estimators to control the influence of the discretization error during the construction of an adaptive SVM-based classifier. We give two algorithms that compute the probability of failure. The first one builds two classifiers: the first classifier separates the guaranteed safe domain from its complementary and the second classifier separates the guaranteed fail domain from its complementary. Therefore, we can compute upper and lower bound of the probability of failure. The second algorithm only uses learning points for which the sign is not polluted by the discretization error and aims at using coarse meshes far from the limit state and fine meshes only close to the limit state. It results in a multi-fidelity meta-model because its construction relies on computations done on two different meshes. These two algorithms are illustrated on two mechanical examples with 2 random variables: a synthetic example for which the exact solution is known and a cracked plate for which the failure scenario is the propagation of the crack. In Section 2, we formulate the mechanical problem, define the limit state function, the probability of failure and derive bounds on values of the limit state function. In Section 3, we give the principles of SVM and briefly describe its use for the estimation of the probability of failure. The two new algorithms are presented in Section 4. Numerical simulations are done in Section 5. The last section concludes the paper.

2. Description of the mechanical reliability problem

In this section, we first define the continuous mechanical problem and its discretization. Then, we show how the use of *a posteriori* error estimator based on the constitutive relation enables to obtain discretization error bounds on the exact value of the limit state function $G_{ex}(x)$. For the

sake of simplicity, these derivations are done in a deterministic framework. In the last subsection, we introduce uncertainties in the form of random variables and explicit the link between the limit state function and the probability of failure.

2.1. Continuous problem

Let us consider a body Ω occupying the physical space $\mathbb{R}^3$ that is subjected to a body force $\underline{f}_{vol}$, to an imposed displacement $\underline{u}_d$ on $\partial_u\Omega$ whose measure is not null and to external forces $\underline{F}$ on $\partial_F\Omega$. Note that $\partial_F\Omega \cup \partial_u\Omega = \partial\Omega$ and that $\text{meas}(\partial_F\Omega \cap \partial_u\Omega) = 0$. We consider that this body undergoes small perturbations and that there is no inertia effect so that the evolution is quasi-static. We also assume that the material behaviour of this solid Ω is linear elastic so that it can be characterized by the Hooke tensor $\mathbb{H}$. We note $\underline{u}$ the displacement of the body and $\underline{\underline{\sigma}}$ the Cauchy stress tensor.

Two affine subspaces and a positive form are introduced:

- The affine subspace of kinematically admissible fields (KA-fields)

$$\text{CA} = \left\{ \underline{u} \in \left(\text{H}^1(\Omega)\right)^d,\ \underline{u} = \underline{u}_d \text{ on } \partial_u\Omega \right\} \tag{1}$$

  and we note CA^0 the associated vectorial space.

- Affine subspace of statically admissible fields (SA-fields)

$$\text{SA} = \left\{ \underline{\underline{\tau}} \in \left(\text{L}^2(\Omega)\right)^{d\times d}_{\text{sym}};\ \forall \underline{v} \in \text{CA}^0, \int_\Omega \underline{\underline{\tau}} : \underline{\underline{\varepsilon}}(\underline{v})\, d\Omega = \int_\Omega \underline{f}_{vol} \cdot \underline{v} d\Omega + \int_{\partial_F\Omega} \underline{F} \cdot \underline{v} dS \right\} \tag{2}$$

- Error in constitutive equation

$$e_{CR_\Omega}(\underline{u}, \underline{\underline{\sigma}}) = \| \underline{\underline{\sigma}} - \mathbb{H} : \underline{\underline{\varepsilon}}(\underline{u}) \|_{\mathbb{H}^{-1},\Omega} \tag{3}$$

  where $\|\underline{\underline{x}}\|_{\mathbb{H}^{-1},\Omega} = \sqrt{\int_\Omega \left(\underline{\underline{x}} : \mathbb{H}^{-1} : \underline{\underline{x}}\right) d\Omega}$

The continuous problems reads:

$$\begin{cases} \text{Find a displacement field } \underline{u} \text{ and a stress field } \underline{\underline{\sigma}} \text{ such that} \\ \underline{u} = \underline{u}_d \text{ on } \partial_u\Omega \text{ and } \underline{\underline{\varepsilon}}(\underline{u}) = \frac{1}{2}(\underline{\underline{grad}}(\underline{u}) + \underline{\underline{grad}}^T(\underline{u})) \text{ on } \Omega \\ \underline{\text{div}}(\underline{\underline{\sigma}}) + \underline{f}_{vol} = \underline{0} \text{ on } \Omega \text{ and } \underline{\underline{\sigma}}\underline{n} = \underline{F} \text{ on } \partial_F\Omega \\ \underline{\underline{\sigma}} = \mathbb{H} : \underline{\underline{\varepsilon}}(\underline{u}) \text{ on } \Omega \end{cases} \tag{4}$$

The following formulation is equivalent to the problem 4.

$$\text{Find } \left(\underline{u}_{ex}, \underline{\underline{\sigma}}_{ex}\right) \in \text{CA} \times \text{SA} \text{ such that } e_{CR_\Omega}(\underline{u}_{ex}, \underline{\underline{\sigma}}_{ex}) = 0$$

Under the hypothesis presented at the beginning of this section, the exact solution $\left(\underline{u}_{ex}, \underline{\underline{\sigma}}_{ex}\right)$ exists and is unique.

In the context of reliability, one defines a limit state function or performance function G that is usually written as a difference between a resistance R and a solicitation S: $G = R - S$. In this paper, we consider that the solicitation is a linear form of the displacement

$$S = \widetilde{L}(\underline{u}) \tag{5}$$

If $G \leqslant 0$, the structure fails. If $G > 0$, the structure is considered safe.

Unfortunately, the analytical form of the solution $\underline{u}_{ex}$ is unavailable so $S(\underline{u}_{ex})$ and $G_{ex} = R - S(\underline{u}_{ex})$ are generally unknown. Therefore, this problem is usually solved thanks to discretization technique. In this paper, we consider that the finite element method (FEM) is used.

2.2. Discrete problem

Let define Ω_H a tessellation of Ω. Continuous shape functions are associated to this tessellation. The FEM relies on the approximation of the subspace of kinematically admissible fields by a subspace of finite dimension:

$$\text{CA}_H = \left\{ \underline{u} \in \left(\text{H}^1(\Omega_H)\right)^d,\ \underline{u} = \underline{u}_d \text{ on } \partial_u \Omega_H \right\} \tag{6}$$

We note CA_H^0 the associated vectorial space.

Therefore, the discretized problem reads:

$$\begin{aligned} &\text{Find a displacement field } \underline{u}_H \in \text{CA}_H \text{ such that } \forall \underline{v}_H \in \text{CA}_H^0 \\ &\int_{\Omega_H} \underline{\underline{\varepsilon}}\left(\underline{u}_H\right) : \mathbb{H} : \underline{\underline{\varepsilon}}\left(\underline{v}_H\right) d\Omega = \int_{\Omega_H} \underline{f}_{vol} \cdot \underline{v}_H d\Omega + \int_{\partial_F \Omega_H} \underline{F} \cdot \underline{v}_H dS \end{aligned} \tag{7}$$

Once $\underline{u}_H$ is obtained, an approximated stress field can be computed thanks to the constitutive relation:

$$\underline{\underline{\sigma}}_H = \mathbb{H} : \underline{\underline{\varepsilon}}\left(\underline{u}_H\right) \tag{8}$$

However, the finite element solution $\underline{u}_H$ almost never coincides with the exact solution $\underline{u}_{ex}$, so $R - S(\underline{u}_{ex}) \neq R - S(\underline{u}_H)$ which is the reason why the FEM introduces a discretization error that is propagated in the reliability analysis. Note that choosing *a priori* an optimal mesh for the

reliability study is impossible as the error on S may non linearly depend on the random variables and that the conception point is usually unknown [17], even more if the resistance R is random.

In the next subsection, we define the discretization error and present a posteriori error estimators.

2.3. Estimation of the discretization error

2.3.1. Generalities

One can introduce the displacement discretization error defined as $\underline{e}_{discr} = \underline{u}_{ex} - \underline{u}_H$ and its energetic norm $|||\underline{e}_{discr}|||_\Omega := \|\underline{\underline{\varepsilon}}(\underline{u}_{ex} - \underline{u}_H)\|_{\mathbb{H}^{-1},\Omega}$. The Pragger-Synge theorem [24] reads:

$$\begin{aligned} &\forall(\hat{\underline{u}}, \hat{\underline{\underline{\sigma}}}) \in \mathrm{CA}(\Omega) \times \mathrm{SA}(\Omega), \\ &\left\|\underline{\underline{\varepsilon}}(\underline{u}_{ex}) - \underline{\underline{\varepsilon}}(\hat{\underline{u}})\right\|^2_{\mathbb{H},\Omega} + \left\|\underline{\underline{\sigma}}_{ex} - \hat{\underline{\underline{\sigma}}}\right\|^2_{\mathbb{H}^{-1},\Omega} = e^2_{CR_\Omega}(\hat{\underline{u}}, \hat{\underline{\underline{\sigma}}}) \end{aligned} \tag{9}$$

This equality is the key ingredient of the discretization error estimator based on the constitutive relation. Indeed, as the finite element solution $\underline{u}_H \in \mathrm{CA}_H$, one can choose $\hat{\underline{u}} = \underline{u}_H$ and then we can write:

$$e_{discr} := |||\underline{e}_{discr}|||_\Omega \leqslant e_{CR_\Omega}(\underline{u}_H, \hat{\underline{\underline{\sigma}}}) \tag{10}$$

This inequality proves that $e_{CR_\Omega}(\underline{u}_H, \hat{\underline{\underline{\sigma}}})$ is a guaranteed *a posteriori* error estimator. To compute this estimator, it is required to build a statically admissible stress field $\hat{\underline{\underline{\sigma}}}$. This task may be complex but several techniques are available in the litterature (see [25], [26], [27] and [28]).

The energetic norm of the discretization error is often difficult to use in an industrial context. Indeed, engineers are often more interested in the error done for a specific quantity of interest such as a stress component in a region of the structure. The next paragraph shows how this error estimator can be used to obtain discretization error bounds on a quantity of interest that is the solicitation S_H.

2.3.2. Definition of the linear quantity of interest and the adjoint problem

We propose to write G as:

$$G = R - \widetilde{L}(\underline{u}) = R - \int_\Omega (\underline{\underline{\sigma}}_\Sigma : \underline{\underline{\varepsilon}}(\underline{u}) + \underline{f}_\Sigma \underline{u}) d\Omega \tag{11}$$

In this equation $\underline{\underline{\sigma}}_\Sigma$ and $\underline{f}_\Sigma$ are called extractors and depend on the nature of the quantity of interest which is the solicitation S in a reliability analysis. For example, if the quantity of

interest is the displacement at the point $\underline{P}$ on the $\underline{e}_1$ direction, then the extractors are $\underline{\underline{\sigma}}_\Sigma = \underline{\underline{0}}$ and $\underline{f}_\Sigma = \delta(\underline{P})\underline{e}_1$.

A second mechanical problem, also called adjoint problem, is defined:

$$\text{Find } \left(\underline{\widetilde{u}}_{ex}, \underline{\underline{\widetilde{\sigma}}}_{ex}\right) \in \mathrm{CA}^0(\Omega) \times \widetilde{\mathrm{SA}}(\Omega) \text{ such that } e_{CR_\Omega}(\underline{\widetilde{u}}_{ex}, \underline{\underline{\widetilde{\sigma}}}_{ex}) = 0 \tag{12}$$

where

$$\widetilde{\mathrm{SA}}(\Omega) = \left\{ \underline{\underline{\tau}} \in \left(\mathrm{L}^2(\Omega)\right)^{d\times d}_{\mathrm{sym}};\ \forall \underline{v} \in \mathrm{CA}^0(\Omega),\ \int_\Omega \underline{\underline{\tau}} : \underline{\underline{\varepsilon}}\left(\underline{v}\right) d\Omega = \tilde{L}(\underline{v}) \right\} \tag{13}$$

The solution to this problem exists and is unique. Since the exact solution is not known, this problem can be solved with the finite element method. By introducing a tesselation $\widetilde{H}$, we then define the vectorial subspace :

$$\mathrm{CA}^0_{\widetilde{H}} = \left\{ \underline{u} \in \left(\mathrm{H}^1(\Omega_{\widetilde{H}})\right)^d,\ \underline{u} = 0 \text{ on } \partial_u\Omega_H \right\} \tag{14}$$

The discretized adjoint problem reads:

$$\begin{aligned} &\text{Find a displacement field } \underline{\widetilde{u}}_{\widetilde{H}} \in \mathrm{CA}^0_{\widetilde{H}} \text{ such that } \forall \underline{v}_{\widetilde{H}} \in \mathrm{CA}^0_{\widetilde{H}} \\ &\int_{\Omega_{\widetilde{H}}} \underline{\underline{\varepsilon}}\left(\underline{\widetilde{u}}_{\widetilde{H}}\right) : \mathbb{H} : \underline{\underline{\varepsilon}}\left(\underline{v}_{\widetilde{H}}\right) d\Omega = \tilde{L}(\underline{v}_{\widetilde{H}}) \end{aligned} \tag{15}$$

One can observe that this problem is similar to the reference finite element problem. The only difference is that the linear form $\tilde{L}$ has replaced the mechanical forces $\underline{f}_{vol}$ and $\underline{F}$. Therefore, if one choses to use the same tessellation for the reference and the adjoint problems ($\widetilde{H} = H$) then both problems can be solved at the same time since they share the same stiffness matrix which enables multiple right-hand side solving. The use of the FEM for the adjoint problem introduces a discretization error that can be estimated in the same way as the reference mechanical problem:

$$\widetilde{e}_{discr} = |||\underline{\widetilde{u}}_{ex} - \underline{\widetilde{u}}_{\widetilde{H}}|||_\Omega \leqslant e_{CR_\Omega}(\underline{\widetilde{u}}_{\widetilde{H}}, \underline{\underline{\hat{\widetilde{\sigma}}}}) \tag{16}$$

where $\underline{\underline{\hat{\widetilde{\sigma}}}}$ is a statically admissible stress field for the adjoint problem.

2.3.3. Error estimation on a quantity of interest

In [29], the following inequality is demonstrated:

$$|\widetilde{L}(\underline{u}_{ex}) - \widetilde{L}(\underline{u}_H) - I_{H\widetilde{H}}| \leqslant \frac{1}{2} e_{CR_\Omega}(\underline{u}_H, \underline{\underline{\hat{\sigma}}}_H) e_{CR_\Omega}(\underline{\widetilde{u}}_{\widetilde{H}}, \underline{\underline{\hat{\widetilde{\sigma}}}}_{\widetilde{H}}) \tag{17}$$

where

$$I_{H\widetilde{H}} = \int_\Omega \frac{1}{2}(\underline{\underline{\hat{\widetilde{\sigma}}}}_{\widetilde{H}} + \mathbb{H} : \underline{\underline{\varepsilon}}\left(\underline{\widetilde{u}}_{\widetilde{H}}\right)) : \mathbb{H}^{-1} : (\underline{\underline{\hat{\sigma}}}_H - \mathbb{H} : \underline{\underline{\varepsilon}}\left(\underline{u}_H\right)) d\Omega \tag{18}$$

and where $\hat{\underline{\underline{\sigma}}}_{\widetilde{H}} \in \widetilde{\mathrm{SA}}_{\widetilde{H}}(\Omega)$. Therefore, by defining

$$G^m = R - \widetilde{L}(\underline{u}_H) + I_{H\widetilde{H}} \tag{19}$$

we can derive bounds on the exact value of $G_{ex} = R - \widetilde{L}(\underline{u}_{ex})$

$$G^- \leqslant G_{ex} \leqslant G^+ \tag{20}$$

where

$$G^- := G^m - \frac{1}{2} e_{CR_\Omega}(\underline{u}_H, \hat{\underline{\underline{\sigma}}}_H) e_{CR_\Omega}(\widetilde{\underline{u}}_{\widetilde{H}}, \hat{\underline{\underline{\sigma}}}_{\widetilde{H}}) \tag{21}$$

and

$$G^+ := G^m + \frac{1}{2} e_{CR_\Omega}(\underline{u}_H, \hat{\underline{\underline{\sigma}}}_H) e_{CR_\Omega}(\widetilde{\underline{u}}_{\widetilde{H}}, \hat{\underline{\underline{\sigma}}}_{\widetilde{H}}) \tag{22}$$

Note that G_H may be outside of $[G^-; G^+]$. As a consequence, G^m is usually a better estimation of G_{ex} than $R - \widetilde{L}(\underline{u}_H)$.

2.3.4. Probability of failure

Let us consider that the uncertainties can be modeled by a vector random variable $X : \zeta \in Z \rightarrow X(\zeta) = \underline{x} \in \mathbb{R}^p$ where Z is the set of possible outcomes and $\underline{x} \in \mathbb{R}^p$ a realization of this random variable. The random variables can be the imposed forces or imposed displacements, a parameter of the geometry, a material property, ... As a consequence, the displacement $\underline{u}$ and the quantity of interest $\widetilde{L}(\underline{u})$ are also random variables.

Let us note p the joint distribution of X. The exact probability of failure is:

$$P_{f,ex} = \int_{D_{ex,f}} p(\underline{x}) dx \tag{23}$$

where $D_{ex,f} = \{\underline{x} \in \mathbb{R}^p | G_{ex}(\underline{x}) \leqslant 0\}$ is the exact failure domain. However, the exact limit state function G_{ex} being usually unknown, the computation of the exact probability of failure is impossible and G_{ex} is approximated by G_H to compute the probability of failure

$$P_{f,H} = \int_{D_{H,f}} p(\underline{x}) dx \tag{24}$$

where $D_{H,f} = \{\underline{x} \in \mathbb{R}^p | G_H(\underline{x}) \leqslant 0\}$ is the finite element failure domain. Of course, due to the discretization error, $G_{ex} \neq G_H$ so $D_{ex,f} \neq D_{H,f}$ and therefore $P_{f,H} \neq P_{f,ex}$. Moreover, G_H is not known explicitly but implicitly, which makes the analytical evaluation of the probability $P_{f,H}$ impossible. Additionnaly, there is no warranty that $P_{f,H}$ exceeds or not $P_{f,ex}$, so that

the approximation may be conservative or not. Finally, this sum is usually estimated using the Monte Carlo method. Even though the Monte Carlo estimation introduces an approximation error, it can be controlled by ensuring that the Monte Carlo population is large enough. The coefficient of variation (COV) of the probability is a widely used indicator of this approximation.

Note that the computation of the integral does not require knowing the value of G_H on $\mathbb{R}^p$ but only the sign of G_H through the definition of the failure domain. This is the reason why classification methods are relevant tools to estimate the probability of failure. Once the classifier is built, the classification of the Monte Carlo population and then the computation of the probability of failure are cheap in terms of computational cost even if the population is large. In the next subsection, the support vector machine classification is briefly presented.

3. Support Vector Machine (SVM) for classification

In this section, we briefly describe the basics on both linear SVM and non linear SVM for the classification of a point $(\mathbf{x}, y) \in \mathbb{R}^p \times \{-1; 1\}$ from n data points, also called observations $(\mathbf{x}_i, y_i)_{i=1..n}$. The objective of this method is to build a classifier $D : \mathbb{R}^p \to \{-1; 1\}$. The interested reader can complete this short section on the basics of SVM classification with the following reference [10].

3.1. Linear SVM

If the data are linearly separable, the classifier D may be built from the linear function f defined as:

$$f(\mathbf{x}) = \mathbf{v}^T\mathbf{x} + a \text{ where } a \in \mathbb{R} \text{ and } \mathbf{v} \in \mathbb{R}^p \tag{25}$$

where $\mathbf{v}^T\mathbf{x}$ is the scalar product between $\mathbf{v}$ and $\mathbf{x}$. Therefore the classifier D reads:

$$D(\mathbf{x}) = \text{sign}(f(\mathbf{x})) \tag{26}$$

where

$$\text{sign}(\square) = \begin{cases} 1 \text{ if } \square > 0 \\ -1 \text{ otherwise} \end{cases} \tag{27}$$

The hyperplane Δ is defined as $\Delta = \{\mathbf{x} \in \mathbb{R}^p \text{ such that } f(\mathbf{x}) = 0\}$ and the margin m is the minimal distance between the data and the hyperplane Δ :

$$m = \min_{i=1..n} \left(\frac{|\mathbf{v}^T\mathbf{x}_i + a|}{||\mathbf{v}||} \right) \tag{28}$$

The parameters $\mathbf{v}$ and a of the function f are sought such that they maximize the margin m. To obtain a unique solution, this optimization problem is rewritten with new variables $\mathbf{w} = \frac{\mathbf{v}}{m||\mathbf{v}||}$ and $b = \frac{a}{m||\mathbf{v}||}$ and reads :

$$\text{Find } \mathbf{w} \text{ and } b \text{ such that } \frac{1}{2}||\mathbf{w}||^2 \text{ is minimum and } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1 \ \forall i = 1..n \tag{29}$$

This problem is called the primal problem as the unknowns $w = \frac{v}{m||v||}$ and $b = \frac{a}{m||v||}$ can be directly connected to the function $\widetilde{f} = \mathbf{w}^T\mathbf{x} + b$ written with the new variables and defining the classifier D. Indeed, $\forall \mathbf{x} \in \mathbb{R}^p,\ \text{sign}(f(\mathbf{x})) = \text{sign}(\widetilde{f}(\mathbf{x}))$.

It is possible to obtain a dual formulation of this problem by writing the Lagrangian $\mathcal{L}$ of problem (29). The Lagrangian reads:

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right) \tag{30}$$

with α_i the Lagrange multipliers associated to the inequality constraint.The saddle point optimality conditions are:

$$\begin{cases} \mathbf{w} - \sum_{i=1}^{n} \alpha_i \left(y_i\mathbf{x}_i\right) = \mathbf{0} \\ \sum_{i=1}^{n} \alpha_i y_i = \mathbf{0} \\ \forall i \in [1;n],\ \alpha_i \geqslant 0 \\ \forall i \in [1;n],\ \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right) \geqslant 0 \\ \forall i \in [1;n],\ \alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right) = 0 \end{cases} \tag{31}$$

Let $\mathcal{A} = \{i \in [1;n] | \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right) = 0\}$ denote the set of active constraints. Then, the stationarity of the Lagrangian reads:

$$\mathbf{w} = \sum_{i \in \mathcal{A}} \alpha_i y_i \mathbf{x}_i \tag{32}$$

The vector $\mathbf{w}$ is a linear combination of the observations associated to the active constraints: those observations are called support vectors. The other data do not affect the definition of the hyperplane. One can replace the primal variables b and $\mathbf{w}$ in the primal formulation (29) to obtain the dual formulation:

$$\begin{aligned} &\text{Find } \alpha_i \text{ for } i \in [1;n] \text{ such that } \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i=1}^{n} \alpha_i \text{ is minimum and} \\ &\sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } \alpha_i \geqslant 0 \ \forall i = 1..n \end{aligned} \tag{33}$$

Both primal and dual problems can be solved with standard optimization solvers such as quadratic programming for instance.

3.2. Non-linear SVM

Unfortunately, in general cases, the optimal classifier is not linear. To overcome this difficulty, non-linear kernels may be introduced. The kernels are used to replace the scalar product $\mathbf{x}_i^T \mathbf{x}_j$ by a measure of the influence of $\mathbf{x}_i$ on $\mathbf{x}_j$. This influence is noted $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ and can be interpreted as a correlation. Therefore, only positive kernels are allowed. One can cite polynomial kernel, Gaussian kernel, exponential kernel. In this article, we use the Gaussian kernel:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{||\mathbf{x}_i - \mathbf{x}_j||_2}{2\sigma^2}\right) \tag{34}$$

where σ is an hyperparameter. In this paper, we do not study the influence of the choice of the kernel. However, methods to build an optimal kernel from the observations exist [30]. The choice of the hyperparameter σ is of prime importance. In this article, we perform cross-validation as available in Mathworks. The dual formulation of the non-linear SVM reads:

$$\begin{aligned} &\text{Find } \alpha_i \text{ for } i \in [1; n] \text{ such that } \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i^T \mathbf{x}_j) - \sum_{i=1}^{n} \alpha_i \text{ is minimum and} \\ &\sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } 0 \leqslant \alpha_i \leqslant C \ \forall i = 1..n \end{aligned} \tag{35}$$

One can observe the introduction of the parameter C which is a penalty parameter. Choosing a finite value of C authorizes misclassification. In this paper, we choose to increase C to infinity which means that the values α_i are not bounded and therefore that misclassification is not allowed. Once again, this optimization problem can be solved with standard solvers.

3.3. SVM classification for estimation of the probability of failure

Classification with SVM has already been used to estimate the probability of failure [31, 12, 13] in the context of reliability analysis. In [12], the authors build a SVM classifier to separate the failure points from the safety points in the Monte Carlo Population U of size n_{MC}. From observations on a design of experiment DOE, a SVM classifier is built. Then, a learning criterion and two learning functions are proposed to improve the classifier adding new observations. Let ξ be the ratio between the number of points of U located inside the margin of the classifier over the size of U. Let k be the number of the learning iteration. In order to have a more stable criterion, the authors of [12] propose to fit the points of ξ using a exponential curve and then define

$\hat{\xi}(k) = A\exp(Bk)$. If $\xi > \eta_1$ or $\hat{\xi} > \eta_1$ or $\left|\frac{d\hat{\xi}}{dk}\right| > \eta_2$ where η_1 and η_2 are two positive scalars defining the stopping criterion for the learning process, the learning criterion is not satisfied. Therefore, the enrichment of the meta-model is done by computing $y_{new} = G(\mathbf{x_{new}})$ for a new point $\mathbf{x_{new}}$. This $\mathbf{x_{new}}$ is chosen close to the separator and far from other observations so that its observation will bring valuable information. Therefore, it reads:

$$\mathbf{x_{new}} = \underset{\mathbf{x}\in U}{\operatorname{argmin}} \frac{s(\mathbf{x})\max(d)}{d(\mathbf{x})\max(s)} \tag{36}$$

where $s(\mathbf{x})$is the distance between $\mathbf{x}$ and the separator. $d(\mathbf{x})$ is the distance between $\mathbf{x}$ and the closest point in the DOE. Note that η_1 and η_2 are defined by the user. In this paper, we choose $\eta_1 = 10^{-4}$ and $\eta_2 = 10^{-5}$, that is to say the same order of magnitude as in [12].

It results in the adaptive construction of the SVM classifier. Once the learning criterion is reached, the Monte Carlo estimator is used to compute the probability of failure by classifying the Monte Carlo population U. A second criterion on the COV of the probability is defined to ensure the quality of the Monte Carlo estimator. If the criterion is not fulfilled, the Monte Carlo population is enlarged to reduce the sampling error. Because of the poor convergence, a very large Monte Carlo population can be required. In order to reduce the computational burden, it is possible to use important sampling (used with Kriging in [32]), subset simulations [4] or directional sampling [33, 34].

The resulting algorithm is called ASVM-MCS (Adaptive Support Vector Machine and Monte Carlo Simulation).

4. Controlling the discretization error into SVM classifier meta-modeling

As explained in subsection 2.3, the use of finite element simulations introduces a discretization error which can lead to a large error on the probability of failure. In this section, we propose two strategies to benefit from the discretization error bounds on G_{ex} developped in subsection 2.3.2 to improve the estimation of the probability of failure *via* SVM classification. Two new algorithms based on ASVM-MCS are detailed respectively in the next two subsections.

4.1. Computation of bounds on the probability of failure

In this subsection, we propose a methodology to compute error bounds on the probability of failure using a unique mesh size. Indeed, we suggest to build two SVM classifiers. The first classifier D^+ is built from the upper bounds observations $y = \text{sign}(G^+(\mathbf{x}))$. Once this classifier D^+ is built, it can be used to separate the Monte Carlo population into two sub-populations:

the guaranteed failure population $D_{pop,gf} = \{(\mathbf{x}_i)_{i=1..n_{MC}} | D^+(\mathbf{x}_i) = -1\}$ and its complementary. Indeed, evaluating $D^+(\mathbf{x}_i)$ for $i = 1..n_{MC}$ is cheap once the classifier D^+ is defined. The second classifier D^- is built from the lower bound observations $y = \text{sign}(G^-(\mathbf{x}))$. Once this classifier D^- is built, it can be used to separate the Monte Carlo population into two sub-populations: the guaranteed safe population $D_{pop,gs} = \{(\mathbf{x}_i)_{i=1..n_{MC}} | D^-(\mathbf{x}_i) = +1\}$ and its complementary. We propose to use these two classifiers to compute two probabilities $P^- = \frac{\text{card}(D_{pop,gf})}{n_{MC}}$ and $P^+ = 1 - \frac{\text{card}(D_{pop,gs})}{n_{MC}}$ as illustrated in Algorithm 1. The learning function ξ^+ is the ratio between the number of points of U located inside the margin of the classifier D^+ over the size of U. The learning function ξ^- is the ratio between the number of points of U located inside the margin of the classifier D^- over the size of U.

Assuming that the two classifiers are correct, which means they are well-trained and do not misclassify points in the Monte Carlo population, we have :

$$D_{pop,gf} \subset \{\mathbf{x} \in U | \, \text{sign}(G_{ex}(\mathbf{x})) = -1\} \subset (U - D_{pop,gs}) \tag{37}$$

Finally, by assuming that the Monte Carlo is large enough so that it does not introduce an approximation error, we can obtain bounds on the exact probability of failure:

$$P^- \leqslant P_{f,ex} \leqslant P^+ \tag{38}$$

The authors are aware that those bounds are not guaranteed because of the meta-modeling error (due to the construction of the SVM classifier) and of the approximation error (due to the finite size of the Monte Carlo population which is classified by the metamodel). However, those two errors are controlled : the learning criterion identifies new observations to update the meta-model and the coefficient of variation indicates if a larger Monte Carlo population is required. Directional sampling or important sampling may help to reduce the computational time if it is necessary. The bounds P^+ and P^- enables to know the influence of the discretization error on the probability of failure. Moreover, the two classifiers D^+ and D^- can be used to exhibit the population for which the mesh size is not fine enough to obtain a classification not polluted by the discretization error. Indeed, the set $D_{pop,uc} = \{\mathbf{x}_i \in U | D^-(\mathbf{x}_i) < 0 \text{ and } D^+(\mathbf{x}_i) > 0\}$, defined as the uncertain population, can be easily obtained from the two classifiers. If the bounding on the exact probability of failure is too large, the user can perform a classification only on $D_{pop,uc}$ with a finer mesh.

4.2. Construction of multi-fidelity SVM-based meta-model

In this subsection, we propose a methodology to train a SVM classifier from observations computed on different mesh sizes. Let us introduce a familiy of mesh sizes $(h_j)_j$ for $j = 0..M$ where h_0 corresponds to the coarsest mesh and h_M to the finest mesh. The key ingredient of this method is the following : only observations G^m (defined in Equation (19)) for which the sign is guaranteed by the bounds G^+ (defined in Equation (22)) and G^- (defined in Equation (21)) are used to train the classifier. Note that using $G^m = G^m(\mathbf{x})$ is a better choice than using the solution of the finite element problem $G_H(\mathbf{x})$ because $G_H(\mathbf{x})$ may not be in the interval $[G^-; G^+]$. Therefore, the classifiers is built only from observations that are not polluted by the discretization error. If an observation at the point $\mathbf{x}$ is such that $G^+G^- < 0$, then the simulation is done at the same point but with a finer mesh as long as the state of the point (fail or safe) is not guaranteed. The construction of the SVM classifier and the estimation of the probability of failure is unchanged. As a consequence, the SVM classifier is built from computations done on different mesh sizes. This is the reason why this approach builds a multi-fidelity SVM-based meta-model. The algorithm of this method is given in algorithm 2. Note that a finest mesh is defined so that the algorithm does not exhibit an infinite while loop. In that only case, it is allowed to use the sign of G^m computed on the finest mesh to enrich the meta-model even if $G^+G^- < 0$.

Note that during the execution of this algorithm, the upper and lower bounds are computed. Therefore, it is possible, for the user, at the end of the algorithm, to build the SVM classifiers from the collection of upper bounds and the collection of lower bounds, in a similar way as it is done in the first subsection and therefore obtain bounds on the probability of failure.

5. Numerical experiments

In this section, we illustrate the two algorithms described in Section 4 on two numerical examples. Both examples are two-dimensional mechanical problem with 2 random variables. The first one is a square plate for wich the exact solution is known. Therefore, for this problem, the exact probability of failure is available. The second example is a cracked plate for which the failure occurs when crack grows, according to Griffith criterion. For both problem the criterion to stop the enrichment of the Monte Carlo population is $\eta_3 = 0.02$ and the criteria to stop the learning of the SVM meta-model are $\eta_1 = 10^{-4}$ and $\eta_2 = 10^{-5}$. The initial size of the DOE is set to $n_{DOE} = 12$. The computations were done on Dell mobile Precision 3530 CTO-Base with Intel Core i7 8750H (6C, 2.2 4.1 GHz, 9 Mo, 45W).

5.1. Square plate

Let us consider a rectangular plate of dimension $3mm \times 3mm$ with the hypothesis of small perturbations and plane strain. All displacements are imposed to be null on the perimeter of the plate. The only external action is the force $\underline{f}$ defined such that the exact solution is:

$$\underline{u}_{ex} = x(x-3)y(y-3)\left((y-3)^2\underline{e}_1 + y\underline{e}_2\right) \tag{39}$$

The solicitation S is the average of the component σ_{xy} of the stress on a square of size $0.5mm$ located on the top left-hand corner of the structure, as illustrated in Figure 1 left.

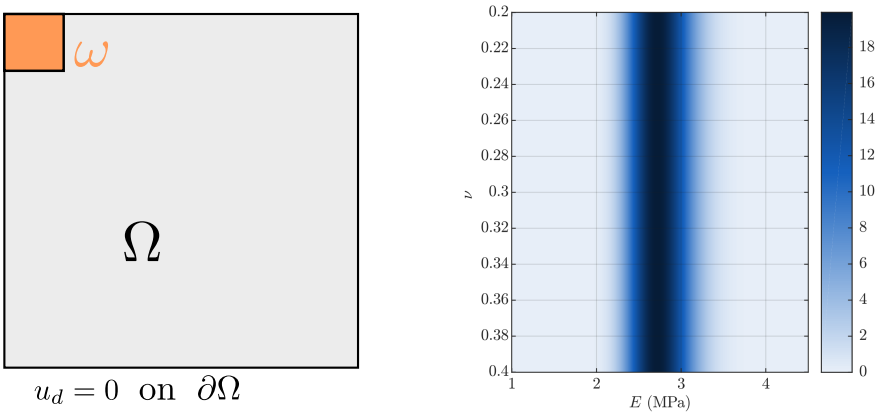

Figure 1: Left figure: square plate, Right figure: joint probability distribution of E and ν

The loading of the adjoint problem is therefore defined through the following linear form:

$$\tilde{L}(\underline{u}) = \frac{1}{mes(\omega)} \int_\omega (\underline{e}_1 \otimes \underline{e}_2) : \underline{\underline{\varepsilon}}(\underline{u})\, d\omega \tag{40}$$

E is the Young modulus and follows a log-normal law of parameters $\mu = 1$ and $\sigma = 0.01$. ν is the Poisson ratio and follows a uniform law between $\nu_{min} = 0.2$ and $\nu_{max} = 0.4$. A map of density for the joint probability density function is plotted on Figure 1 right. Those two random variables are considered independent. The resistance R is deterministic and $R = 6.3$MPa. Since the exact displacement field is available, the exact probability of failure can be computed from (23): we obtain $P_{f,ex} = 1.1118\ 10^{-4}$. Moreover, it is also possible to plot the exact limit state ($G_{ex} = 0$). Note that for this problem the exact limit state is linear. However, since the form of the limit state is usually unkown we used the kernel trick to build the SVM classifier which explains the form of the computed limit states. The finite element problem is solved with second-order elements composing the uniform mesh of size h.

340 *5.1.1. Strategy 1 : Computation of bounds thanks to 2 classifiers*

341 The strategy described in Algorithm 1 is applied for this example on a first mesh of size
342 $h_1 = 0.25$mm. In Table 1, we give the number of calls to the FE code as well as the bounds on
343 the probability of failure obtained for 5 different Monte Carlo populations. We observe that $P_{f,ex}$
344 lies between the computed bounds for all the Monte Carlo populations. For each simulation, the
345 two classifiers enable to exhibit the uncertain population. We also observe that the number of
346 calls is highly dependant on the Monte Carlo population. Then, a second mesh size $h_2 = 0.1$mm
347 is defined and the algorithm is run again to classify the uncertain population into certain safe
348 population and certain failure population. It results in additional calls to the finite element
349 solver but also in an improvement of the bounds on the exact probability of failure that can
350 be observed in Table 1. One can notice that for the first Monte Carlo population, the upper
351 $P_+ = 1.1115\ 10^{-4}$ is smaller than $P_{f,ex} = 1.1118\ 10^{-4}$. This is probably due to the finite size
352 of the Monte Carlo population as the requirement on the coefficient of variation is that it has
353 to be smaller than 0.02. In Figure 2, we plot for each mesh size, the two classifiers as well as
354 the certain failure domains and the certain safety domains for the first Monte Carlo population.
355 The exact limit state $G_{ex} = 0$ is also plotted in blue line on these plot. In addition, the points of
356 the Monte Carlo population that were selected for a finite element call to enrich the meta-model
357 are specified. We observe that the calls on the fine mesh h_2 are close to the exact limit state.
358 This is not surprising as the second mesh was used to classify the uncertain population. We also
359 notice in the right-hand side figure (zoom) that the exact limit state lies between the two limit
360 classifiers for both h_1 and h_2.

361 *5.1.2. Strategy 2 : multi-fidelity classifier*

362 The strategy described in Algorithm 2 is applied on this example with a family of two meshes
363 of size $h_{max} = h_1$ and $h_{min} = h_2$. In Table 2, we give the number of calls to the FE code as
364 well as the probabilities of failure obtained for 5 different Monte Carlo populations. Note that
365 these populations are the same as the ones used to illustrate the first strategy. We observe that
366 the 5 probabilities of failure are very close to the exact probability of failure $P_{f,ex}$. In Figure
367 3, we plot the limit state as well as the failure and safety domains for the first Monte Carlo
368 population. The exact limit state $G_{ex} = 0$ is also plotted in blue line. In addition, the points of
369 the Monte Carlo population that were selected for a finite element call to enrich the meta-model
370 are specified. We observe that the calls on the fine mesh h_{min} are done close to the exact limit
371 state. The computational cost is focused on the frontier between safe and failure domains which

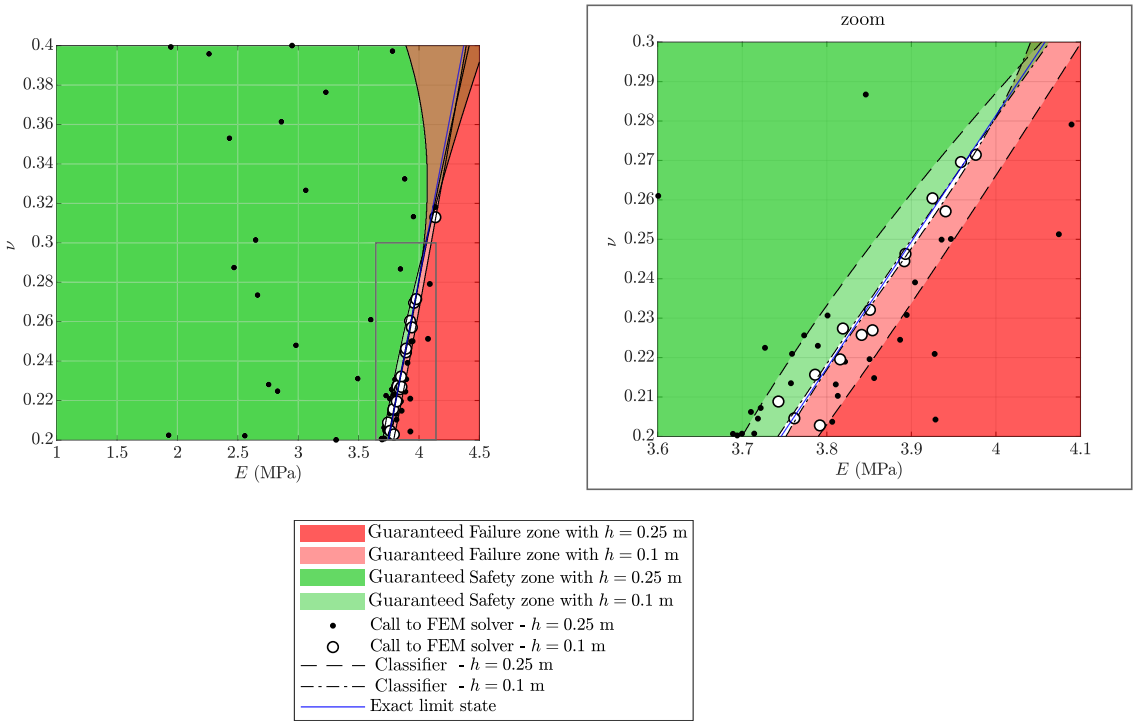

Figure 2: Left: Classifiers obtained for the strategy 1 on the square plate problem; Right: zoom

enables to obtain a good estimation of the probability of failure.

5.2. Cracked plate

Let us consider a cracked plate of length $L = 16mm$ and width $w = 7mm$. The length of the crack is denoted by a and is random. The material is linear elastic isotropic and the Poisson ratio is $\nu = 0.3$. The traction force $\overrightarrow{F}$ of norm $F = 1MPa$ is applied on two sides of the plate with an angle θ, which is the second random variable of the problem. The solicitation S is the stress intensity factor on mode I, which we noted K_I, that can be computed thanks to integrals on a crown around the crack tip, see [35], as illustrated on Figure 4 left. The outer radius of the crown is 1.5mm and the inner radius is 1mm. The failure is defined from the Griffith criterion, therefore:

$$G = K_{lim} - K_I \tag{41}$$

where $K_{lim} = 22MPa\sqrt{mm}$ is the deterministic resistance. K_I is a linear quantity of interest with respect to the displacement. As a consequence, it is possible to define an adjoint problem, see [36], to obtain error bounds on G_{ex}, as explained in Section 2.3. Both random variables follow

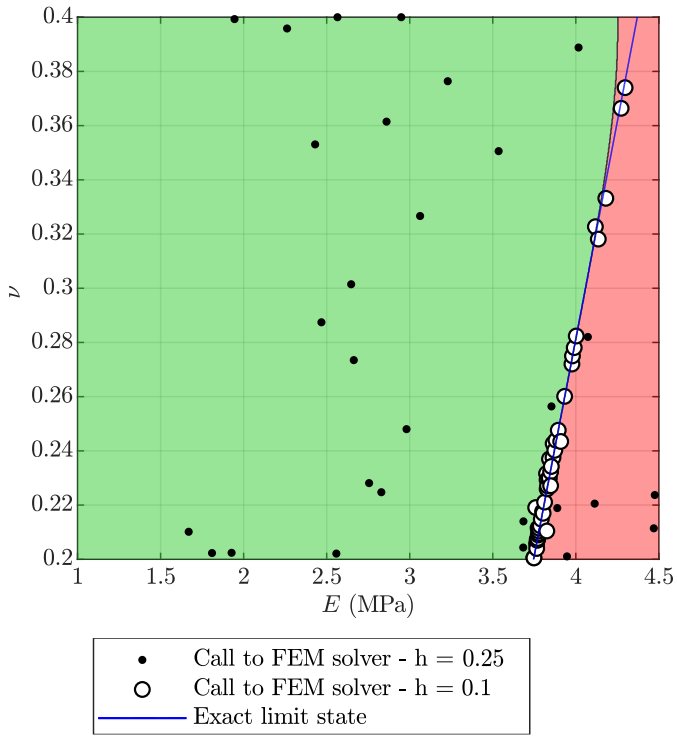

Figure 3: Limit state obtained for the strategy 2 on the square plate problem

the Beta distribution and are independant. The parameters given in Table 3 are the distribution of these two random variables which is illustrated in Figure 4 right.

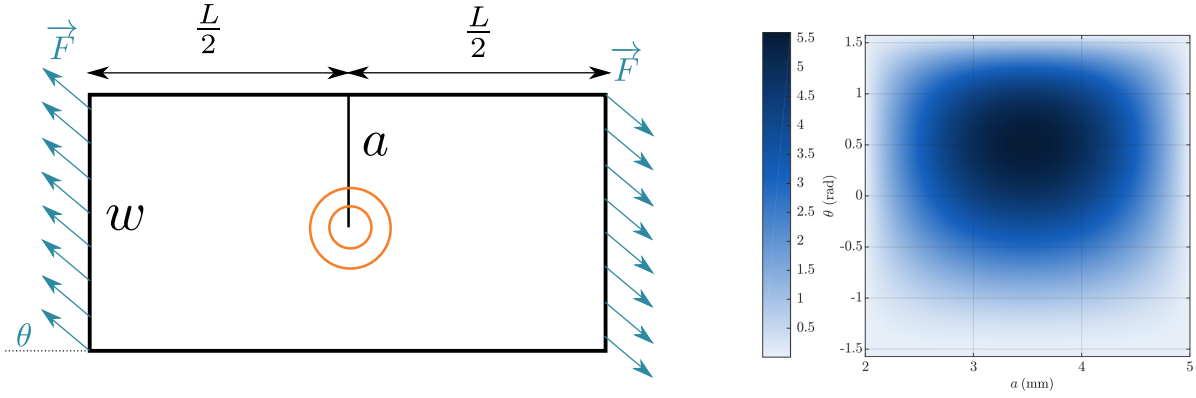

Figure 4: Left figure: cracked plate, Right figure: joint distribution distribution of a and θ.

For this problem, the exact value of the stress intensity factor for the first mode (crack opening) is not available for $\theta \neq 0$. A reference probability of failure P_{ref} is computed on a

overkill uniform mesh of size $h_{overkill} = 0.02mm$ with the reference method ASVMMCS [12] using the first Monte Carlo population. We obtained $P_{ref} = 5.9\ 10^{-3}$. This simulation required 69 calls to the finite element solver that last 16 184 seconds. The overkill classifier obtained during this process is plotted in blue on Figures 5 and 6.

5.2.1. Strategy 1

We applied the Algorithm 1 first with the mesh size $h_1 = 0.28mm$ and then with the mesh size $h_2 = 0.1mm$. In Table 4, we give the number of calls to the FE code as well as the bounds on the probability of failure obtained for 5 different Monte Carlo populations. The values of P^+ and P^- are comparable for all the Monte Carlo populations. Similarly to the previous example, the number of calls highly depends on the Monte Carlo population. As for the first example, we observe that the bounds on the probability of failure are improved when the mesh size h_2 is used. In Table 5, we give the CPU time spent for the FE solving and the error estimation for the two mesh size h_1 and h_2 for 5 different Monte Carlo populations. We observe that the discretization error estimation is very expensive. However, this procedure enables to obtain bounds on the probability of failure, which is more informative than a unique value. In Figure 5, we give the classifiers as well as the guaranteed fail and safe domains for the two meshes for the first Monte Carlo population. The calls carried out to enrich the SVM-meta-models are also depicted in black or white, depending on the mesh size. We observe that the overkill classifier lies between the two classifiers in the uncertain white zone. Calls on the fine mesh are done close to the limit state. This figure clearly illustrates that applying again the Algorithm on a finer mesh enables to reduce the size of the uncertain zone.

5.2.2. Strategy 2

We applied the second algorithm on the cracked plate case for two mesh sizes $h_{max} = 0.28mm$ and $h_{min} = 0.1mm$. In Figure 6, we represent the limit state obtained during the algorithm that separates the failure domain from the safety domain for the first Monte Carlo population. In this figure, the initial design of experiment and the learning points are presented. Points for which the coarse mesh was precise enough are in black. Points for which the sign of the performance function was not guaranteed and that required an additional call on the fine mesh are in white. We observe that far from the classifier, the coarse mesh is satisfying. However, as expected, when precision is required close to the limit state, calls on the fine mesh are done. We also observe that the multifidelity classifier (in black) is close to the overkill classifier (in blue) which illustrates the precision of the meta-model.

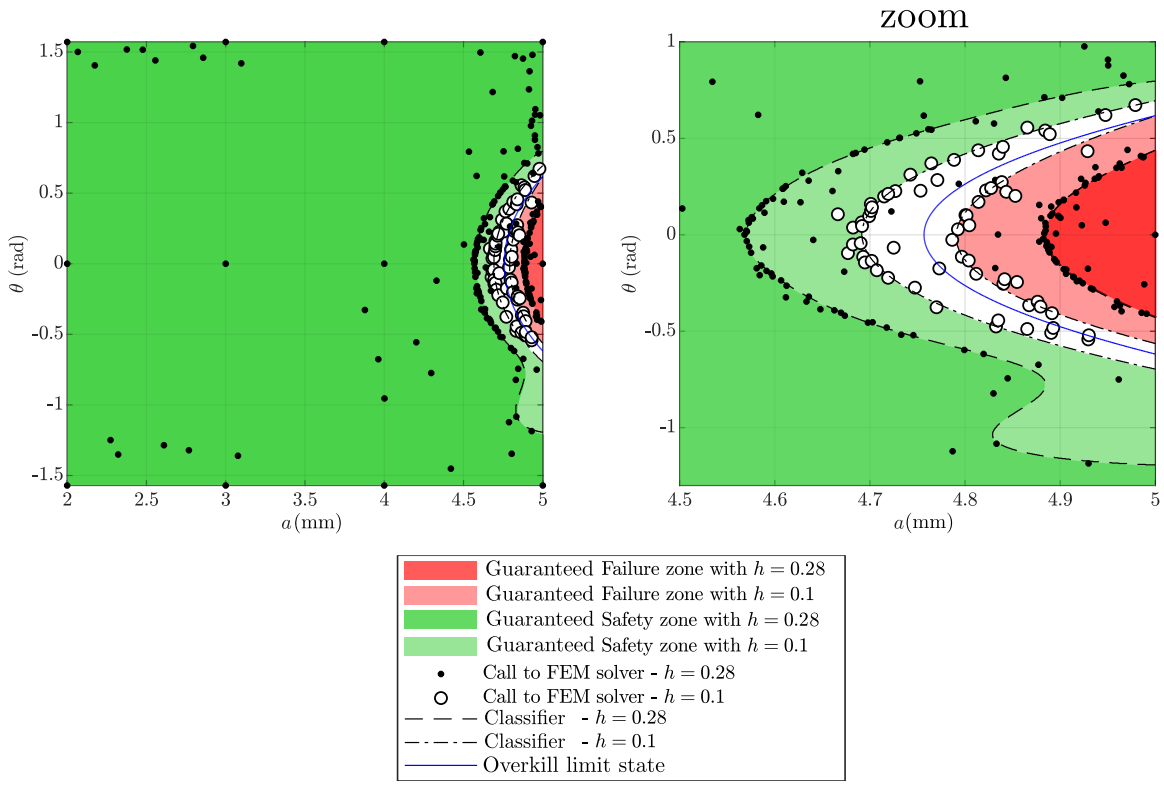

Figure 5: Limit states obtained for the strategy 1 on the cracked plate

In Table 6, we give the number of calls on each mesh, as well as the CPU time and the probabilities of failure obtained for 5 Monte Carlo populations. We observe that for all the Monte Carlo populations, the computed probability of failure has the same order of magnitude as P_{ref}. The probabilities of failure are always larger than P_{ref}. This difference can be explained by the fact that the computations were done on different meshes since $h_{min} \neq h_{overkill}$. Moreover, for this second strategy, G^m is used whereas for the overkill monofidelity strategy, the FE output G_H is used. From this table, we can observe that the computational cost associated to error estimation is very large. This is due to the construction of the statically admissible stress field. For the first Monte Carlo population, we observe that the CPU time for FE solving and error estimation is equivalent to the CPU time for FE solving on the overkill mesh.

On this example, the second strategy requires less FE calls and therefore less CPU time than the first strategy.

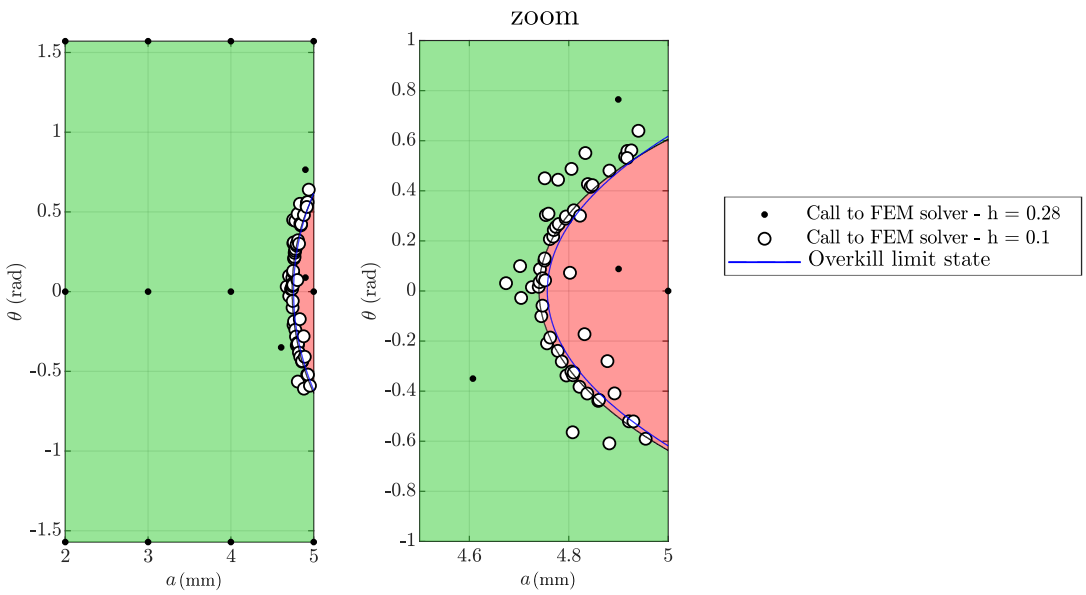

Figure 6: Left: limit state obtained for the strategy 2 on the cracked plate; Right: zoom

6. Conclusion

In this paper, we propose to use discretization error estimators to build an improved SVM-based meta-model for the estimation of the probability of failure for quasi-static mechanical problems. This first method considers a defined mesh size and enables to compute upper and lower bounds of the exact probability of failure. Two classifiers are built during the execution of the algorithm. It allows to identify the points in the Monte Carlo population for which the status (fail or safe) is polluted by the finite element discretization error. If this subpopulation is too large, that is to say if the bounds on the probability of failure are not precise enough, the user can run again this algorithm on this subpopulation with a finer mesh. The second method consists in using observations to construct the SVM classifier only if the discretization error bounds on the performance function enable to guarantee its sign. The SVM meta-model is therefore built from observations on different meshes. It results in a multi-fidelity meta-model. Those two strategies were illustrated on two mechanical examples and both strategies exhibit their ability to precisely estimate the probability of failure. Both strategies lead to an additional cost as they rely on discretization error estimations which can be expensive. However, the first strategy enables to obtain bounds on the probability of failure which is a more valuable information than a unique value. Moreover, the second strategy usually gives better results than monofidelity approach with

no error estimation as G^m is usually closer to G_{ex} than G_H. Finally, using purely monofidelity meta-model would require to do a convergence study on the probability of failure with respect to the mesh size. Therefore, the computational cost of our proposed methods should be compared to the total cost of the whole convergence study. The two proposed approaches were illustrated on problems with two random variables but the algorithms can be used straightforwardly for problems with more random variables.

Further work will consist in exploiting the error map provided by the discretization error estimators to build an optimal non uniform mesh. Another interesting topic would be to adapt the stopping criterion of learning process and the stopping criterion on the size of the Monte Carlo population to the discretization error. For example, in the first method, reaching a very small COV for the probability of failure may not be relevant if the discretization error is too large. This would require separating the different sources of error in the estimation of the probability of failure.

7. Acknowledgment

This work was carried out within the project MUSCAS (MUlti-SCAle Stochastic computation for MRE) granted by WEAMEC, West atlantic Marine Energy Community with the support of Région Pays de la Loire and in partnership with Chantiers de l'Atlantique.

8. Bibliography

References

[1] P. Ciarlet, The Finite Element Method for Elliptic Problems, Studies in Mathematics and its Applications, Elsevier Science, 1978.
URL `https://books.google.fr/books?id=TpHfoXnpKvAC`

[2] N. Metropolis, S. Ulam, The monte carlo method, Journal of the American statistical association 44 (247) (1949) 335–341.

[3] M. Lemaire, A. Chateauneuf, J.-C. Mitteau, Fiabilité des structures.

[4] S.-K. Au, J. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probabilistic engineering mechanics 16 (4) (2001) 263–277.

[5] M. Giles, Multilevel monte carlo path simulation, Operations Research 56 (3) (2008) 607–617.

[6] M. Rashki, A. Ghavidel, H. Arab, S. Mousavi, Low-cost finite element method-based reliability analysis using adjusted control variate technique, Structural Safety 75 (2018) 133–142.

[7] J. N. Fuhg, A. Fau, U. Nackenhorst, State-of-the-art and comparative review of adaptive sampling methods for kriging, Archives of Computational Methods in Engineering 28 (4) (2021) 2689–2747.

[8] F. Schoefs, Sensitivity approach for modelling the environmental loading of marine structures through a matrix response surface, Reliability Engineering & System Safety 93 (7) (2008) 1004–1017.

[9] R. Teixeira, M. Nogal, A. O'Connor, Adaptive approaches in metamodel-based reliability analysis: A review, Structural Safety 89 (2021) 102019.

[10] V. Vapnik, The nature of statistical learning theory, Springer science & business media, 2013.

[11] T. Most, An adaptive response surface approach for structural reliability analyses based on support vector machines, in: Proceedings of the eleventh international conference on civil, structural and environmental engineering computing, BHV Topping, 2007.

[12] Q. Pan, D. Dias, An efficient reliability method combining adaptive support vector machine and monte carlo simulation, Structural Safety 67 (2017) 85–95.

[13] H. Song, K. Choi, I. Lee, L. Zhao, D. Lamb, Adaptive virtual support vector machine for reliability analysis of high-dimensional problems, Structural and Multidisciplinary Optimization 47 (4) (2013) 479–491.

[14] J.-M. Bourinet, F. Deheeger, M. Lemaire, Assessing small failure probabilities by combined subset simulation and support vector machines, Structural Safety 33 (6) (2011) 343–353.

[15] A. Basudhar, S. Missoum, An improved adaptive sampling scheme for the construction of explicit boundaries, Structural and Multidisciplinary Optimization 42 (4) (2010) 517–529.

[16] I. Babuvška, W. Rheinboldt, Error estimates for adaptive finite element computations, SIAM Journal on Numerical Analysis 15 (4) (1978) 736–754.

[17] L. Mell, V. Rey, F. Schoefs, Multifidelity adaptive kriging metamodel based on discretization error bounds, International Journal for Numerical Methods in Engineering 121 (20) (2020) 4566–4583. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6451, doi:10.1002/nme.6451.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6451

[18] A. Ghavidel, M. Rashki, H. Arab, M. Moghaddam, Reliability mesh convergence analysis by introducing expanded control variates, Frontiers of Structural and Civil Engineering 14 (4) (2020) 1012–1023.

[19] K. Alvin, Method for treating discretization error in nondeterministic analysis, AIAA journal 38 (5) (2000) 910–916.

[20] L. Morse, Z. Khodaei, M. Aliabadi, A multi-fidelity boundary element method for structural reliability analysis with higher-order sensitivities, Engineering Analysis with Boundary Elements 104 (2019) 183–196.

[21] J. Yi, F. Wu, Q. Zhou, Y. Cheng, H. Ling, J. Liu, An active-learning method based on multi-fidelity kriging model for structural reliability analysis, Structural and Multidisciplinary Optimization (2020) 1–23.

[22] L. Gallimard, Error bounds for the reliability index in finite element reliability analysis, International journal for numerical methods in engineering 87 (8) (2011) 781–794.

[23] L. Mell, V. Rey, F. Schoefs, Two multifidelity kriging-based strategies to control discretization error in reliability analysis exploiting a priori and a posteriori error estimators, Computers & Structures (2022) 106897doi:https://doi.org/10.1016/j.compstruc.2022.106897.

[24] W. Prager, J. Synge, Approximations in elasticity based on the concept of function space, Quarterly of Applied Mathematics 5 (3) (1947) 241–269.

[25] P. Ladeveze, D. Leguillon, Error estimate procedure in the finite element method and applications, SIAM Journal on Numerical Analysis 20 (3) (1983) 485–509.

[26] N. Parés, P. Díez, A. Huerta, Subdomain-based flux-free a posteriori error estimators, Computer Methods in Applied Mechanics and Engineering 195 (4-6) (2006) 297–323.

[27] F. Pled, L. Chamoin, P. Ladevèze, On the techniques for constructing admissible stress fields in model verification: Performances on engineering examples, International Journal for Numerical Methods in Engineering 88 (5) (2011) 409–441.

[28] V. Rey, P. Gosselet, C. Rey, Study of the strong prolongation equation for the construction of statically admissible stress fields: implementation and optimization, Computer Methods in Applied Mechanics and Engineering 268 (2014) 82–104.

[29] P. Ladevèze, Strict upper error bounds on computed outputs of interest in computational structural mechanics, Computational Mechanics 42 (2) (2008) 271–286.

[30] L. Dioşan, A. Rogozan, J.-P. Pecuchet, Improving classification performance of support vector machine by genetically optimising kernel shape and hyper-parameters, Applied Intelligence 36 (2) (2012) 280–294.

[31] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, Journal of machine learning research 2 (Nov) (2001) 45–66.

[32] B. Echard, N. Gayton, M. Lemaire, N. Relun, A combined importance sampling and kriging reliability method for small failure probabilities with time-demanding numerical models, Reliability Engineering & System Safety 111 (2013) 232–240.

[33] P. Bjerager, Probability integration by directional simulation, Journal of Engineering Mechanics 114 (8) (1988) 1285–1302.

[34] J. Nie, B. Ellingwood, Directional methods for structural reliability analysis, Structural Safety 22 (3) (2000) 233–249.

[35] M. Stern, E. B. Becker, R. S. Dunham, A contour integral computation of mixed-mode stress intensity factors, International Journal of Fracture 12 (3) (1976) 359–368.

[36] L. Gallimard, J. Panetier, Error estimation of stress intensity factors for mixed-mode cracks, International Journal for Numerical Methods in Engineering 68 (3) (2006) 299–316.

Algorithm 1:

Generate the the Monte Carlo population U of size n_{MC} ;
Generate the Design Of Experiment (DOE) of size n_{DOE};
for $i = 1..n_{DOE}$ **do**
    Evaluate $G^m(\mathbf{x}_i)$, $G^+(\mathbf{x}_i)$ and $G^-(\mathbf{x}_i)$;
    Complete the training samples with $y_i = \text{sign}(G^+(\mathbf{x}_i))$ and $z_i = \text{sign}(G^-(\mathbf{x}_i))$;
end
Train the SVM classifier D^+ from the training sample $(y_i)_{i=1..n_{DOE}}$;
Train the SVM classifier D^- from the training sample $(z_i)_{i=1..n_{DOE}}$;
Classify the Monte Carlo population thanks to the SVM classifier D^+ into two populations: guaranteed failure population $D_{pop,gf} = \{(\mathbf{x}_i)_{i=1..n_{MC}} | D^+(\mathbf{x}_i) = -1\}$ and its complementary;
Classify the Monte Carlo population thanks to the SVM classifier D^- into two populations: guaranteed safe population $D_{pop,gs} = \{(\mathbf{x}_i)_{i=1..n_{MC}} | D^-(\mathbf{x}_i) = +1\}$ and its complementary;
Estimate the probability $P^- = \frac{\text{card}(D_{pop,gf})}{n_{MC}}$;
Compute the coefficient of variation $\text{COV}^- = \sqrt{\frac{1-P^-}{P^- n_{MC}}}$;
Compute the learning functions ξ^- and $\hat{\xi}^-$;
Estimate the probability $P^+ = 1 - \frac{\text{card}(D_{pop,gs})}{n_{MC}}$;
Compute the coefficient of variation $\text{COV}^+ = \sqrt{\frac{1-P^+}{P^+ n_{MC}}}$;
Compute the learning functions ξ^+ and $\hat{\xi}^+$;
Initialize enrichment iteration number k=0;
while $\xi^+ > \eta_1$ *or* $\hat{\xi}^+ > \eta_1$ *or* $\left|\frac{d\hat{\xi}^+}{dk}\right| > \eta_2$ *or* $\xi^- > \eta_1$ *or* $\hat{\xi}^- > \eta_1$ *or* $\left|\frac{d\hat{\xi}^-}{dk}\right| > \eta_2$ *or* $\text{COV}^+ > \eta_3$ *or* $\text{COV}^- > \eta_3$ **do**
    if $\xi^+ > \eta_1$ *or* $\hat{\xi}^+ > \eta_1$ *or* $\left|\frac{d\hat{\xi}^+}{dk}\right| > \eta_2$ *or* $\xi^- > \eta_1$ *or* $\hat{\xi}^- > \eta_1$ *or* $\left|\frac{d\hat{\xi}^-}{dk}\right| > \eta_2$ **then**
        if $\xi^+ > \eta_1$ *or* $\hat{\xi}^+ > \eta_1$ *or* $\left|\frac{d\hat{\xi}^+}{dk}\right| > \eta_2$ **then**
            Set k=k+1 ;
            Select the optimal next sample point $\mathbf{x_{new}}$ using equation (36) with classifier D^+;
            Evaluate $G^m(\mathbf{x_{new}})$, $G^+(\mathbf{x_{new}})$ and $G^-(\mathbf{x_{new}})$ at this point;
            Complete the training sample with $y_{new} = \text{sign}(G^+(\mathbf{x_{new}}))$;
            Complete the training sample with $z_{new} = \text{sign}(G^-(\mathbf{x_{new}}))$;
        end
        if $\xi^- > \eta_1$ *or* $\hat{\xi}^- > \eta_1$ *or* $\left|\frac{d\hat{\xi}^-}{dk}\right| > \eta_2$ **then**
            Set k=k+1 ;
            Select the optimal next sample point $\mathbf{x_{new}}$ using equation (36) with classifier D^-;
            Evaluate $G^m(\mathbf{x_{new}})$, $G^+(\mathbf{x_{new}})$ and $G^-(\mathbf{x_{new}})$ at this point;
            Complete the training sample with $y_{new} = \text{sign}(G^+(\mathbf{x_{new}}))$;
            Complete the training sample with $z_{new} = \text{sign}(G^-(\mathbf{x_{new}}))$;
        end
        Train the SVM classifier D^+ from the training sample (y_i);
        Train the SVM classifier D^- from the training sample (z_i);
        Classify the Monte Carlo population thanks to the SVM classifier D^+ ;
        Classify the Monte Carlo population thanks to the SVM classifier D^- ;
        Estimate the probabilities P^- and P^+ ;
        Compute the coefficients of variation COV^- and COV^+;
        Compute the learning functions ξ^- , $\hat{\xi}^-$, ξ^+ and $\hat{\xi}^+$;
    end
    else
        Enlarge the Monte Carlo population;
        Classify the Monte Carlo population thanks to the SVM classifier D^+ ;
        Classify the Monte Carlo population thanks to the SVM classifier D^- ;
        Estimate the probabilities P^- and P^+ ;
        Compute the coefficients of variation COV^- and COV^+;
        Compute the learning functions ξ^- , $\hat{\xi}^-$, ξ^+ and $\hat{\xi}^+$;
    end
end

Algorithm 2:

Define a list of mesh sizes $(h_j)_{j=0..N}$;
Generate the the Monte Carlo population U of size n_{MC} ;
Generate the Design Of Experiment (DOE) of size n_{DOE};
Set $j = 0$;
for $i = 1..n_{DOE}$ **do**
    Evaluate $G^m(\mathbf{x}_i)$, $G^+(\mathbf{x}_i)$ and $G^-(\mathbf{x}_i)$ with the mesh size h_j;
    while $G^+(\mathbf{x}_i)G^-(\mathbf{x}_i) < 0$ *and* $j < M$ **do**
        $j = j + 1$;
        Evaluate $G^m(\mathbf{x}_i)$, $G^+(\mathbf{x}_i)$ and $G^-(\mathbf{x}_i)$ with the mesh size h_j;
    end
    Complete the training sample with $y_i = \text{sign}(G^m(\mathbf{x}_i))$;
end
Set $j = 0$;
Train the SVM classifier D from the training sample;
Classify the Monte Carlo population thanks to the SVM classifier D into two populations: failure population $D_{pop,f} = \{(\mathbf{x}_i)_{i=1..n_{MC}} | D(\mathbf{x}_i) = -1\}$ and safety population $D_{pop,s} = \{(\mathbf{x}_i)_{i=1..n_{MC}} | D(\mathbf{x}_i) = +1\}$;
Estimate the probability $P_f = \frac{\text{card}(D_{pop,f})}{n_{MC}}$;
Compute the coefficient of variation $\text{COV} = \sqrt{\frac{1-P_f}{P_f n_{MC}}}$;
Compute the learning functions ξ and $\hat{\xi}$;
while $\xi > \eta_1$ *or* $\hat{\xi} > \eta_1$ *or* $\left|\frac{d\hat{\xi}}{dk}\right| > \eta_2$ *or* $\text{COV} > \eta_3$ **do**
    if $\xi > \eta_1$ *or* $\hat{\xi} > \eta_1$ *or* $\left|\frac{d\hat{\xi}}{dk}\right| > \eta_2$ **then**
        Select the optimal next sample point $\mathbf{x_{new}}$ using equation (36);
        Evaluate $G^m(\mathbf{x_{new}})$, $G^+(\mathbf{x_{new}})$ and $G^-(\mathbf{x_{new}})$ with the mesh size h_j;
        while $G^+(\mathbf{x_{new}})G^-(\mathbf{x_{new}}) < 0$ *and* $j < M$ **do**
            $j = j + 1$;
            Evaluate $G^m(\mathbf{x_{new}})$, $G^+(\mathbf{x_{new}})$ and $G^-(\mathbf{x_{new}})$ with the mesh size h_j;
        end
        Complete the training sample with $y_{new} = \text{sign}(G_m(\mathbf{x_{new}}))$;
    end
    Train the SVM from the training sample;
    Classify the Monte Carlo population thanks to the SVM classifier D into two populations: failure domain $D_{pop,f}$ and safety domain $D_{pop,s}$;
    Estimate the probability $P_f = \frac{\text{card}(D_{pop,f})}{n_{MC}}$;
    Compute the coefficient of variation $\text{COV} = \sqrt{\frac{1-P_f}{P_f n_{MC}}}$;
    Compute the learning functions ξ and $\hat{\xi}$;
    else
        Enlarge the Monte Carlo population;
        Classify the Monte Carlo population thanks to the SVM ;
        Estimate the probability $P_f = \frac{\text{card}(D_{pop,f})}{n_{MC}}$;
        Compute the coefficient of variation $\text{COV} = \sqrt{\frac{1-P_f}{P_f n_{MC}}}$;
        Compute the learning functions ξ and $\hat{\xi}$;
    end
end

$h_1 = 0.25$mm			$h_2 = 0.1$mm		
Nb calls	P_- ($\times 10^{-4}$)	P_+ ($\times 10^{-4}$)	Nb calls	P_- ($\times 10^{-4}$)	P_+ ($\times 10^{-4}$)
54	0.7031	1.6822	16	1.0602	1.1115
113	0.6771	1.7963	18	1.0659	1.1610
153	0.6756	1.7931	13	1.0524	1.12469
114	0.6620	1.8171	13	1.0812	1.1561
92	0.6925	1.8453	10	1.0162	1.3097

Table 1: Bounds of the probability of failure for the strategy 1 on the square plate problem

P_f ($\times 10^{-4}$)	Nb calls h_{max}	Nb calls h_{min}
1.0979	74	45
1.1194	70	42
1.1109	75	41
1.0964	85	47
1.1121	74	44

Table 2: Estimated probabilities of failure with the strategy 2 on the square plate problem

random variable	distribution	lower bound	upper bound	first shape parameter	second shape parameter
a	Beta	2	5	2	2
θ	Beta	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	3	2

Table 3: Distribution of the random variables a and θ

h_1			h_2		
Nb calls	P_- ($\times 10^{-3}$)	P_+ ($\times 10^{-3}$)	Nb calls	P_- ($\times 10^{-3}$)	P_+ ($\times 10^{-3}$)
194	0.938	23.00	65	4.022	10.36
171	0.923	22.44	52	4.274	10.45
110	0.944	22.68	52	3.868	10.46
164	0.961	22.53	61	4.043	10.42
156	0.919	22.32	69	3.932	11.56

Table 4: Bounds of the probability of failure for the strategy 1 on the cracked plate

h_1		h_2	
$t_{EF}(s)$	t_{err} (s)	$t_{EF}(s)$	t_{err} (s)
223	5293	202	11069
244	6499	188	11580
151	3938	206	13017
289	7553	277	19177
303	7878	519	27717

Table 5: Numerical cost for strategy 1 for 5 Monte Carlo populations

P_f ($\times 10^{-3}$)	Nb calls $h_{max} = 0.28$	Nb calls $h_{min} = 0.2$	$t_{EF}(s)$	t_{err} (s)
6.765	73	58	326	16253
6.849	71	56	389	20447
7.062	66	53	326	15190
6.640	90	74	414	17129
6.995	93	73	362	14922

Table 6: Strategy 2 on the cracked plate for 5 Monte Carlo populations