Reliability-based design optimization using adaptive kriging surrogates

Vincent Dubourg

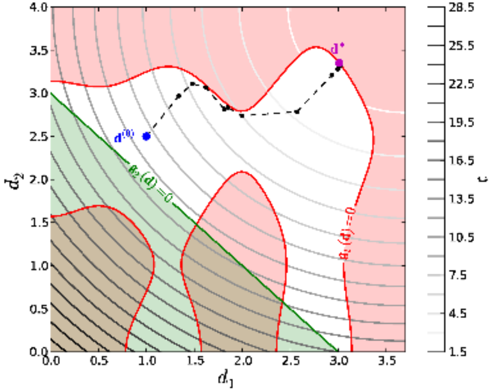
Phimeca Engineering SA
Laboratoire de Mécanique & Ingénieries

GdR MascotNUM – Research topics day – IHP, Paris – May 11, 2012

Design problem formulation

Deterministic design optimization

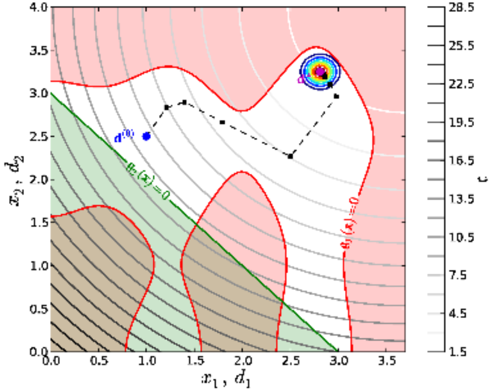
$$d^* = \arg\min_{d \in \mathbb{D}} c(d) \; : \begin{cases} f_i(d) \leq 0, i = 1, \ldots, n_c \\ g_l(x, d) \geq 0, l = 1, \ldots, n_p \end{cases}$$



Design problem formulation

Reliability-based design optimization

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}\in\mathbb{D}} c(\boldsymbol{d}) \ : \begin{cases} f_i(\boldsymbol{d}) \leq 0, i = 1, \dots, n_c \\ \mathbb{P}\left[g_l(\boldsymbol{X}) \leq 0 \mid \boldsymbol{d}\right] \leq p^0_{f\,l}, l = 1, \dots, n_p \end{cases}$$



Premise & objectives

- Performance models are *computationally expensive* (mostly finite-element based).

⇒ Replace the original expensive model with a *cheaper meta-model*.

- *Reliability approximation techniques* (such as FORM) cannot guarantee the safety level of their designs.

⇒ Develop a strategy that is able to *guarantee the design's safety*.

- Stakeholders target *highly reliable designs*.

⇒ The overall strategy should be scalable to *low failure probabilities*.

A particular interest has been given to *quantifying*, *reducing* and *eliminating* the error induced by the use of a surrogate.

Premise & objectives

- Performance models are *computationally expensive* (mostly finite-element based).

⇒ Replace the original expensive model with a *cheaper meta-model*.

- *Reliability approximation techniques* (such as FORM) cannot guarantee the safety level of their designs.

⇒ Develop a strategy that is able to *guarantee the design's safety*.

- Stakeholders target *highly reliable designs*.

⇒ The overall strategy should be scalable to *low failure probabilities*.

A particular interest has been given to *quantifying*, *reducing* and *eliminating* the error induced by the use of a surrogate.

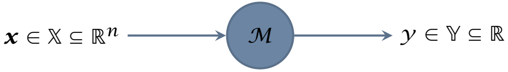
Outline

1. Gaussian process meta-modelling
2. Adaptive designs of experiments
3. Reliability analysis
4. Reliability-based design optimization

Meta-modelling

Meta-modelling techniques

- aim at constructing a *predictor* $\widetilde{\mathcal{M}}$
- that *mimics* the behaviour of an existing model $\mathcal{M}$



- from a collection of observations gathered in a *dataset*:

$$\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y_i\right), i = 1, \ldots, m\right\}, \quad y_i = \mathcal{M}\left(\boldsymbol{x}^{(i)}\right), i = 1, \ldots, m$$

- and *statistical considerations*.

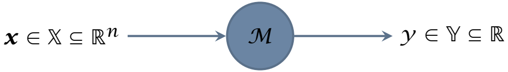
Interest for reliability-based design

Such predictors are *much faster to evaluate* than the original model $\mathcal{M}$, and come with a sort of *confidence measure*.

Meta-modelling

Meta-modelling techniques

- aim at constructing a *predictor* $\widetilde{\mathcal{M}}$
- that *mimics* the behaviour of an existing model $\mathcal{M}$



- from a collection of observations gathered in a *dataset*:

$$\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y_i\right), i = 1, \ldots, m\right\}, \quad y_i = \mathcal{M}\left(\boldsymbol{x}^{(i)}\right), i = 1, \ldots, m$$

- and *statistical considerations*.

Interest for reliability-based design

Such predictors are *much faster to evaluate* than the original model $\mathcal{M}$, and come with a sort of *confidence measure*.

Gaussian process meta-modelling

The Gaussian process prior model

(Santner et al., *2003)*

The function $\mathcal{M}$ is a sample path of a Gaussian process (GP) Y:

$$Y(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\beta} + Z(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{X}$$

where:

- $\boldsymbol{f}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\beta}$ is a linear regression model;
- $Z(\boldsymbol{x})$ is a zero-mean, stationary GP with covariance:

$$\mathrm{Cov}\left[Y(\boldsymbol{x}), Y(\boldsymbol{x}')\right] = \sigma^2 R\left(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\theta}\right), \quad (\boldsymbol{x}, \boldsymbol{x}') \in \mathbb{X} \times \mathbb{X}$$

Hence, given a vector of *observations* $\mathbf{Y} = \left(Y_i = Y\left(\boldsymbol{x}^{(i)}\right), i = 1, \ldots, m\right)$ and an unobserved value $Y(\boldsymbol{x})$, we have:

$$\begin{Bmatrix} Y(\boldsymbol{x}) \\ \mathbf{Y} \end{Bmatrix} \sim \mathcal{N}_{1+m}\left(\begin{Bmatrix} \boldsymbol{f}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\beta} \\ \mathbf{F}\,\boldsymbol{\beta} \end{Bmatrix}, \sigma^2 \begin{bmatrix} 1 & \boldsymbol{r}(\boldsymbol{x})^{\mathsf{T}} \\ \boldsymbol{r}(\boldsymbol{x}) & \mathbf{R} \end{bmatrix}\right)$$

whose parameters $\mathbf{F}$, $\boldsymbol{r}(\boldsymbol{x})$, $\mathbf{R}$ are inherited from the GP's statistics ($\boldsymbol{f}$ and R).

Gaussian process meta-modelling
The Gaussian process prior model
(Santner et al.*, 2003)*

The function $\mathcal{M}$ is a sample path of a Gaussian process (GP) Y:

$$Y(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^{\mathsf{T}}\,\boldsymbol{\beta} + Z(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{X}$$

where:

- $\boldsymbol{f}(\boldsymbol{x})^{\mathsf{T}}\,\boldsymbol{\beta}$ is a linear regression model;
- $Z(\boldsymbol{x})$ is a zero-mean, stationary GP with covariance:

$$\mathrm{Cov}\left[Y(\boldsymbol{x}),\, Y(\boldsymbol{x}')\right] = \sigma^2\, R\left(\boldsymbol{x} - \boldsymbol{x}',\, \boldsymbol{\theta}\right), \quad (\boldsymbol{x},\, \boldsymbol{x}') \in \mathbb{X} \times \mathbb{X}$$

Hence, given a vector of *observations* $\boldsymbol{Y} = \left(Y_i = Y\left(\boldsymbol{x}^{(i)}\right),\, i = 1, \ldots, m\right)$ and an unobserved value $Y(\boldsymbol{x})$, we have:

$$\begin{Bmatrix} Y(\boldsymbol{x}) \\ \boldsymbol{Y} \end{Bmatrix} \sim \mathcal{N}_{1+m}\left(\begin{Bmatrix} \boldsymbol{f}(\boldsymbol{x})^{\mathsf{T}}\,\boldsymbol{\beta} \\ \mathbf{F}\,\boldsymbol{\beta} \end{Bmatrix}, \sigma^2 \begin{bmatrix} 1 & \boldsymbol{r}(\boldsymbol{x})^{\mathsf{T}} \\ \boldsymbol{r}(\boldsymbol{x}) & \mathbf{R} \end{bmatrix}\right)$$

whose parameters $\mathbf{F}$, $\boldsymbol{r}(\boldsymbol{x})$, $\mathbf{R}$ are inherited from the GP's statistics ($\boldsymbol{f}$ and R).

Gaussian process meta-modelling

Posterior ≡ The best linear unbiased predictor (BLUP) *(Santner* et al., *2003)*

- Here, we are interested in the *posterior* distribution of the unobserved value given $\boldsymbol{y} = \left(y_i = \mathcal{M}\left(\boldsymbol{x}^{(i)}\right),\, i = 1, \ldots, m\right)$:

$$\widehat{Y}(\boldsymbol{x}) = [Y(\boldsymbol{x}) \,|\, \boldsymbol{Y} = \boldsymbol{y}]$$

- Given σ^2 and $\boldsymbol{\theta}$, the *universal Kriging* predictor is also Gaussian:

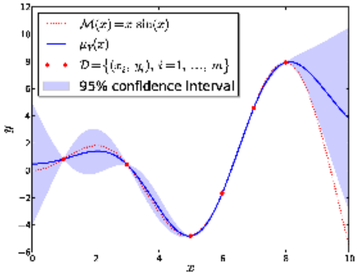
$$\widehat{Y}(\boldsymbol{x}) = \left[Y(\boldsymbol{x}) \,\middle|\, \boldsymbol{Y} = \boldsymbol{y}, \sigma^2, \boldsymbol{\theta}\right] \sim \mathcal{N}_1\left(\mu_{\widehat{Y}}(\boldsymbol{x}),\, \sigma^2_{\widehat{Y}}(\boldsymbol{x})\right)$$

  where the *mean prediction* $\mu_{\widehat{Y}}(\boldsymbol{x})$ and the *prediction variance* $\sigma^2_{\widehat{Y}}(\boldsymbol{x})$ have analytical expressions.

- In practice, σ^2 and $\boldsymbol{\theta}$ are not known so that they must be estimated from the observations $\boldsymbol{y}$ using *e.g.* *maximum likelihood estimation*.

(Welch et al., *1992; Marrel* et al., *2008)*

Illustration on a one-dimensional regression exercise



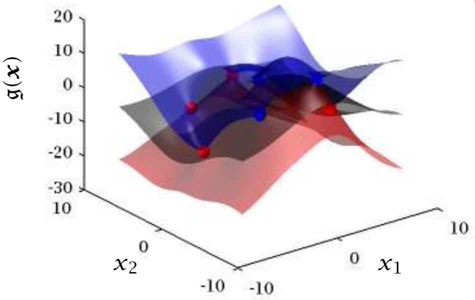
$$\hat{Y}(\boldsymbol{x}) \sim \mathcal{N}_1\left(\mu_{\hat{Y}}(\boldsymbol{x}),\, \sigma^2_{\hat{Y}}(\boldsymbol{x})\right)$$

Interesting properties

- *interpolating*;
- *asymptotically consistent* (provided the correlation R is "compatible" with the data $\mathcal{Y}$ and the model $\mathcal{M}$); (Vazquez, 2005)
- *Gaussian* (consequence of the prior).

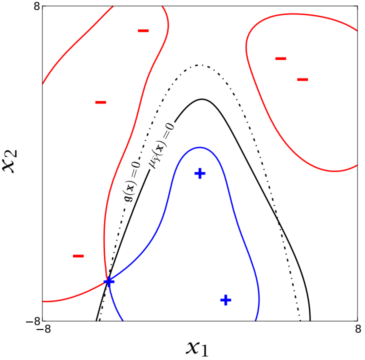
From regression to probabilistic classification

Ex: Let $\mathfrak{g}$ denote a quadratic limit-state function.

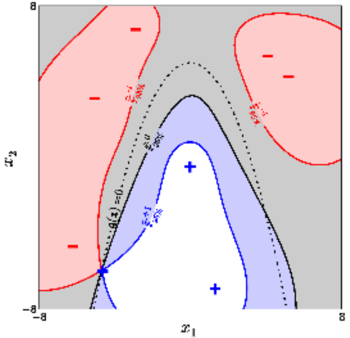


Regression

Classification ($\mathfrak{g} \leq 0$ *vs.* $\mathfrak{g} > 0$)



From regression to probabilistic classification



- Let $\hat{\mathbb{F}}^{-1}_{95\%}$, $\hat{\mathbb{F}}^{0}_{95\%}$, $\hat{\mathbb{F}}^{+1}_{95\%}$ denote the three following *approximate failure subsets*:

$$\hat{\mathbb{F}}^{i}_{95\%} \quad \left\{ \boldsymbol{x} \in \mathbb{X} : \mu_{\hat{Y}}(\boldsymbol{x}) \leq i\, 1.96\, \sigma_{\hat{Y}}(\boldsymbol{x}) \right\},$$

$i = -1,\ 0,\ +1.$

- In turns, this enables the definition of the *margin of uncertainty*:

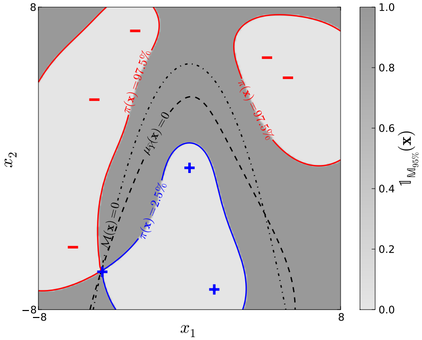
$$\mathbb{M}_{95\%} = \hat{\mathbb{F}}^{-1}_{95\%} \setminus \hat{\mathbb{F}}^{-1}_{95\%}$$

- Let π denote the *probabilistic classification function*:

$$\pi(\boldsymbol{x}) \quad \mathcal{P}\left[\hat{Y}(\boldsymbol{x}) \leq 0\right] \quad \Phi\left(\frac{0 - \mu_{\hat{Y}}(\boldsymbol{x})}{\sigma_{\hat{Y}}(\boldsymbol{x})}\right)$$

$\mathcal{P}(\cdot + \mathbb{P})$ *denotes the probability measure w.r.t. the Kriging* epistemic *uncertainty.*

From regression to probabilistic classification



- Let $\hat{\mathbb{F}}^{-1}_{95\%}$, $\hat{\mathbb{F}}^{0}_{95\%}$, $\hat{\mathbb{F}}^{+1}_{95\%}$ denote the three following *approximate failure subsets*:

$$\hat{\mathbb{F}}^{i}_{95\%} = \left\{ \boldsymbol{x} \in \mathbb{X} : \mu_{\hat{Y}}(\boldsymbol{x}) \leq i\, 1.96\, \sigma_{\hat{Y}}(\boldsymbol{x}) \right\},$$

$i = -1,\ 0,\ +1.$

- In turns, this enables the definition of the *margin of uncertainty*:

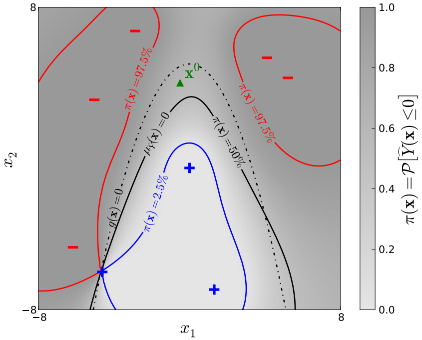
$$\mathbb{M}_{95\%} = \hat{\mathbb{F}}^{+1}_{95\%} \setminus \hat{\mathbb{F}}^{-1}_{95\%}$$

- Let π denote the *probabilistic classification function*:

$$\pi(\boldsymbol{x}) = \mathcal{P}\left[\hat{Y}(\boldsymbol{x}) \leq 0\right] = \Phi\left(\frac{0 - \mu_{\hat{Y}}(\boldsymbol{x})}{\sigma_{\hat{Y}}(\boldsymbol{x})}\right)$$

$\mathcal{P}(\neq \mathbb{P})$ *denotes the probability measure w.r.t. the Kriging* epistemic *uncertainty.*

From regression to probabilistic classification



- Let $\widehat{\mathbb{F}}^{-1}_{95\%}$, $\widehat{\mathbb{F}}^{0}_{95\%}$, $\widehat{\mathbb{F}}^{+1}_{95\%}$ denote the three following *approximate failure subsets*:

$$\widehat{\mathbb{F}}^{i}_{95\%} = \left\{\boldsymbol{x} \in \mathbb{X} : \mu_{\hat{Y}}(\boldsymbol{x}) \leq i\,1.96\,\sigma_{\hat{Y}}(\boldsymbol{x})\right\},$$

$i = -1,\ 0,\ +1.$

- In turns, this enables the definition of the *margin of uncertainty*:

$$\mathbb{M}_{95\%} = \widehat{\mathbb{F}}^{+1}_{95\%} \setminus \widehat{\mathbb{F}}^{-1}_{95\%}$$

- Let π denote the *probabilistic classification function*:

$$\pi(\boldsymbol{x}) = \mathcal{P}\left[\hat{Y}(\boldsymbol{x}) \leq 0\right] = \Phi\left(\frac{0 - \mu_{\hat{Y}}(\boldsymbol{x})}{\sigma_{\hat{Y}}(\boldsymbol{x})}\right)$$

$\mathcal{P}\,(\neq \mathbb{P})$ *denotes the probability measure w.r.t. the Kriging* epistemic *uncertainty.*

Designs of experiments

Designs of experiments

- A DOE is the input part of a dataset:

$$\mathcal{X} = \left\{\boldsymbol{x}^{(i)},\, i = 1, \ldots, m\right\}$$

- Its *size m* must be minimized for the sake of *efficiency*.
- Experiments must be *selected carefully* for the sake of *accuracy* *(space-filling DOEs, Franco, 2008)*.

Adaptive designs of experiments

- are built in an *iterative* manner;
- on purpose to *refine the predictor locally* (*e.g.* in the vicinity of a contour);

Sequential adaptive DOEs for GP predictors rely on the *maximization* of a so-called *refinement criterion*.

Sequential adaptive DOEs

Refinement criteria for contour approximation

- Simple criteria mostly apply the *margin shrinking concept* for support vector machines

(Hurtado, 2004b; Deheeger, 2008)

- Here, we propose the "*margin probability*":

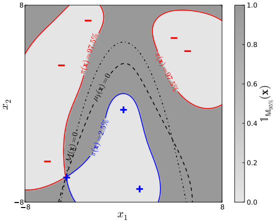
$$\mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] = \Phi\left(\frac{1.96\,\sigma_{\widehat{Y}}(\boldsymbol{x}) - \mu_{\widehat{Y}}(\boldsymbol{x})}{\sigma_{\widehat{Y}}(\boldsymbol{x})}\right) - \Phi\left(\frac{-1.96\,\sigma_{\widehat{Y}}(\boldsymbol{x}) - \mu_{\widehat{Y}}(\boldsymbol{x})}{\sigma_{\widehat{Y}}(\boldsymbol{x})}\right)$$

(Dubourg et al., 2010a)

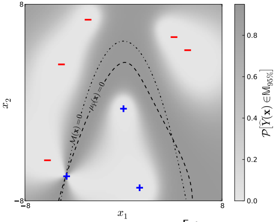
Limitation of sequential strategies

- The multiple modes of these criteria make their *maximization difficult*;
- There does not exist *a single best point*;
- Availability of *distributed computing* platforms for $\mathcal{M}$.

(Ginsbourger et al., 2010)



The margin of uncertainty $\mathbb{M}_{95\%}$



The margin probability $\mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right]$

Sequential adaptive DOEs

Refinement criteria for contour approximation

- Simple criteria mostly apply the *margin shrinking concept* for support vector machines

(Hurtado, 2004b; Deheeger, 2008)

- Here, we propose the "*margin probability*":

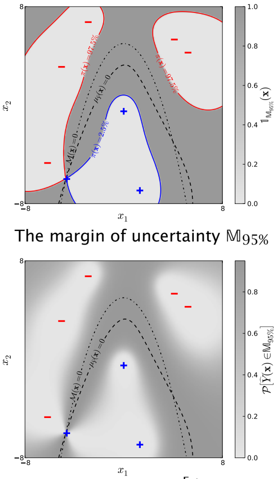
$$\mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] = \Phi\left(\frac{1.96\,\sigma_{\widehat{Y}}(\boldsymbol{x}) - \mu_{\widehat{Y}}(\boldsymbol{x})}{\sigma_{\widehat{Y}}(\boldsymbol{x})}\right) - \Phi\left(\frac{-1.96\,\sigma_{\widehat{Y}}(\boldsymbol{x}) - \mu_{\widehat{Y}}(\boldsymbol{x})}{\sigma_{\widehat{Y}}(\boldsymbol{x})}\right)$$

(Dubourg et al., 2010a)

Limitation of sequential strategies

- The multiple modes of these criteria make their *maximization difficult*;
- There does not exist *a single best point*;
- Availability of *distributed computing* platforms for $\mathcal{M}$.

(Ginsbourger et al., 2010)



The margin of uncertainty $\mathbb{M}_{95\%}$

The margin probability $\mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right]$

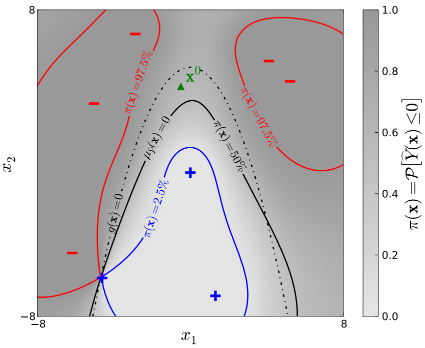
Sampling-based adaptive DOEs

Given an initial dataset $\mathcal{D}$
and a pseudo-PDF w:

1. Fit a Kriging predictor $\widehat{Y}(\boldsymbol{x})$
2. Define a *weighted refinement criterion*

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] w(\boldsymbol{x})$$

3. Sample N candidates from C
   (MCMC slice sampler, Neal, 2003)
4. Reduce the N candidates to K points
   (K-means clustering, Lloyd, 1982)
5. Enrich the dataset $\mathcal{D}$ with $\left\{\left(\boldsymbol{x}^{(m+k)}, \mathcal{M}\left(\boldsymbol{x}^{(m+k)}\right)\right), k = 1, \ldots, K\right\}$
6. Loop back to step 1



Sampling-based adaptive DOEs

Given an initial dataset $\mathcal{D}$
and a pseudo-PDF w:

1. Fit a Kriging predictor $\widehat{Y}(\boldsymbol{x})$
2. Define a *weighted refinement criterion*

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] w(\boldsymbol{x})$$

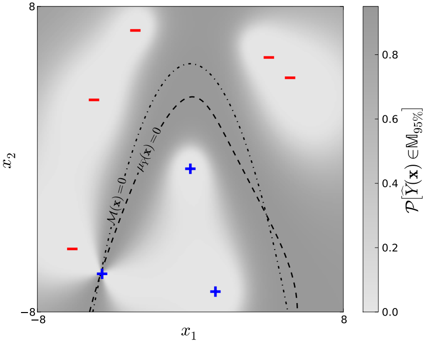
3. Sample N candidates from C

   (MCMC slice sampler, Neal, 2003)

4. Reduce the N candidates to K points

   (K-means clustering, Lloyd, 1982)

5. Enrich the dataset $\mathcal{D}$ with $\left\{\left(\boldsymbol{x}^{(m+k)}, \mathcal{M}\left(\boldsymbol{x}^{(m+k)}\right)\right), k = 1, \ldots, K\right\}$
6. Loop back to step 1



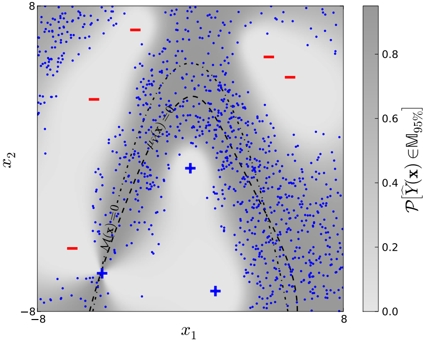
Sampling-based adaptive DOEs

Given an initial dataset $\mathcal{D}$
and a pseudo-PDF w:

1. Fit a Kriging predictor $\widehat{Y}(\boldsymbol{x})$
2. Define a *weighted refinement criterion*

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] w(\boldsymbol{x})$$

3. Sample N candidates from C
   (MCMC slice sampler, Neal, 2003)
4. Reduce the N candidates to K points
   (K-means clustering, Lloyd, 1982)
5. Enrich the dataset $\mathcal{D}$ with
   $\left\{\left(\boldsymbol{x}^{(m+k)}, \mathcal{M}\left(\boldsymbol{x}^{(m+k)}\right)\right), k = 1, \ldots, K\right\}$
6. Loop back to step 1



N is given (say $10,000$)

Sampling-based adaptive DOEs

Given an initial dataset $\mathcal{D}$
and a pseudo-PDF w:

1. Fit a Kriging predictor $\widehat{Y}(\boldsymbol{x})$
2. Define a *weighted refinement criterion*

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] w(\boldsymbol{x})$$

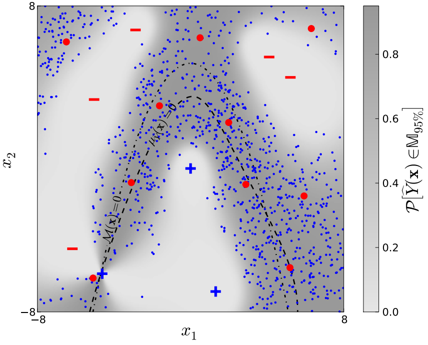
3. Sample N candidates from C

   (MCMC slice sampler, Neal, 2003)

4. Reduce the N candidates to K points

   (K-means clustering, Lloyd, 1982)

5. Enrich the dataset $\mathcal{D}$ with $\left\{\left(\boldsymbol{x}^{(m+k)}, \mathcal{M}\left(\boldsymbol{x}^{(m+k)}\right)\right), k = 1, \ldots, K\right\}$
6. Loop back to step 1



K is given *(say the number of CPUs)*

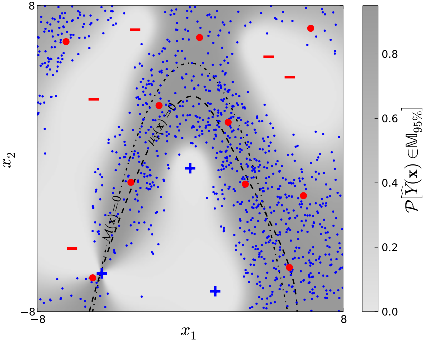
Sampling-based adaptive DOEs

Given an initial dataset $\mathcal{D}$
and a pseudo-PDF w:

1. Fit a Kriging predictor $\widehat{Y}(\boldsymbol{x})$
2. Define a *weighted refinement criterion*

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] w(\boldsymbol{x})$$

3. Sample N candidates from C
   (MCMC slice sampler, Neal, 2003)
4. Reduce the N candidates to K points
   (K-means clustering, Lloyd, 1982)
5. Enrich the dataset $\mathcal{D}$ with
   $\left\{\left(\boldsymbol{x}^{(m+k)}, \mathcal{M}\left(\boldsymbol{x}^{(m+k)}\right)\right), k = 1, \ldots, K\right\}$
6. Loop back to step 1



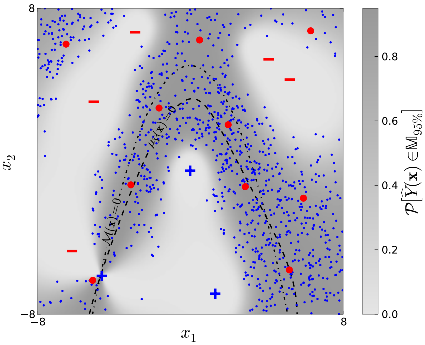
Sampling-based adaptive DOEs

Given an initial dataset $\mathcal{D}$
and a pseudo-PDF w:

1. Fit a Kriging predictor $\widehat{Y}(\boldsymbol{x})$
2. Define a *weighted refinement criterion*

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] w(\boldsymbol{x})$$

3. Sample N candidates from C
   (MCMC slice sampler, Neal, 2003)
4. Reduce the N candidates to K points
   (K-means clustering, Lloyd, 1982)
5. Enrich the dataset $\mathcal{D}$ with $\left\{\left(\boldsymbol{x}^{(m+k)}, \mathcal{M}\left(\boldsymbol{x}^{(m+k)}\right)\right), k = 1, \dots, K\right\}$
6. Loop back to step 1

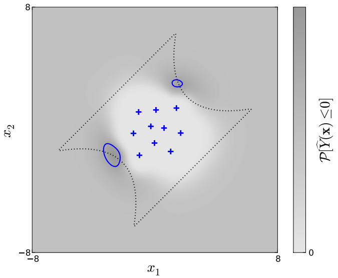


Illustration

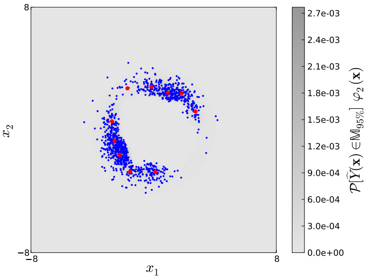
A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\hat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction



Sampled & clustered refinement criterion

Iteration #1

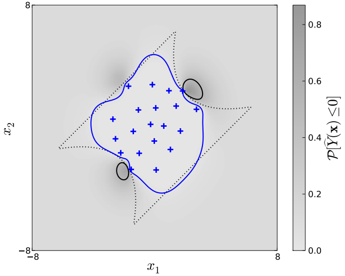
Convergence criteria depend on the *application*...

Illustration

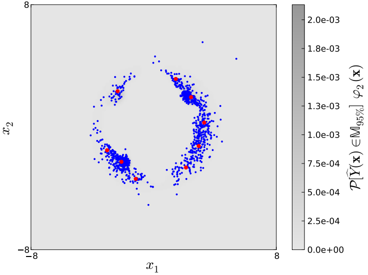
A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction



Sampled & clustered refinement criterion

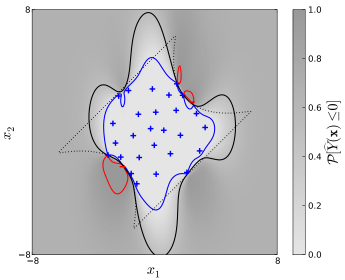
Iteration #2

Convergence criteria depend on the *application*...

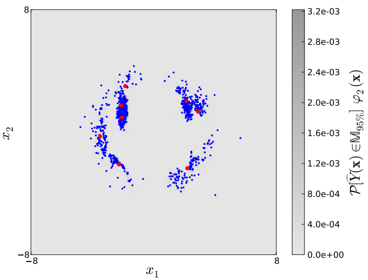
Illustration

A four-branch series system *(Waarts, 2000)*

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction



Sampled & clustered refinement criterion

Iteration #3

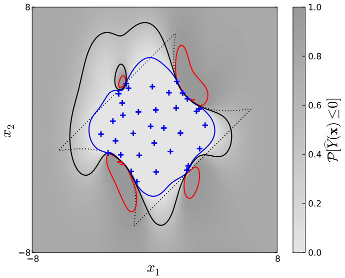
Convergence criteria depend on the *application*...

Illustration

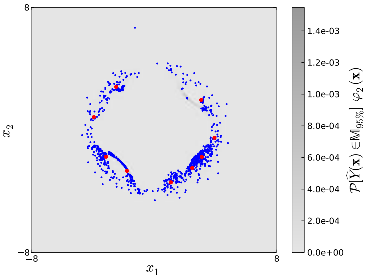
A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction



Sampled & clustered refinement criterion

Iteration #4

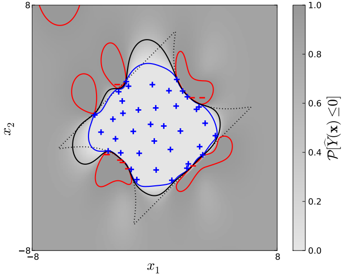
Convergence criteria depend on the *application*...

Illustration

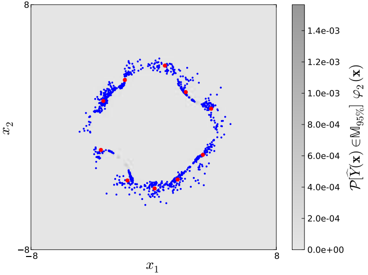
A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction



Sampled & clustered refinement criterion

Iteration #5

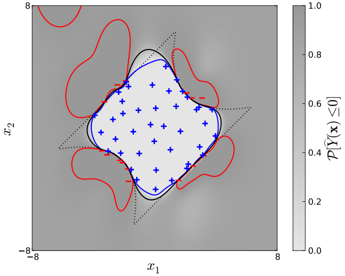
Convergence criteria depend on the *application*...

Illustration

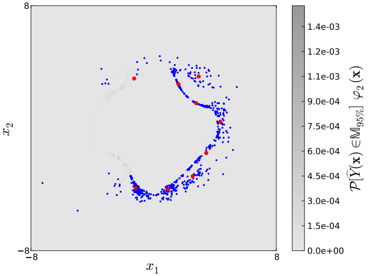
A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction



Sampled & clustered refinement criterion

Iteration #6

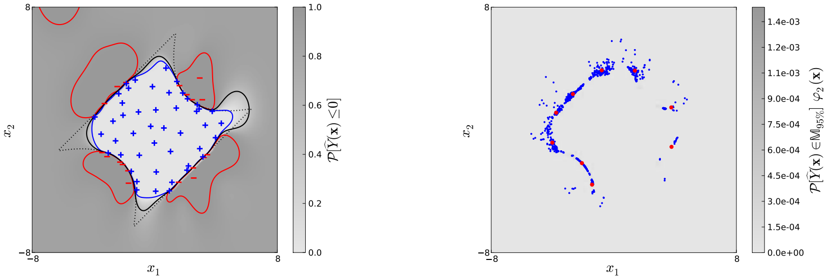
Convergence criteria depend on the *application*...

Illustration

A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\hat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction

Sampled & clustered refinement criterion

Iteration #7

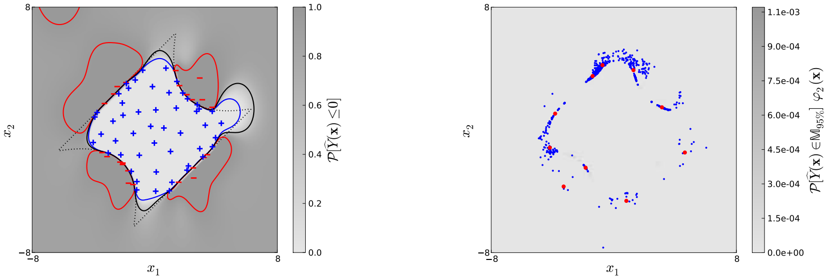
Convergence criteria depend on the *application*...

Illustration

A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction | Sampled & clustered refinement criterion

Iteration #8

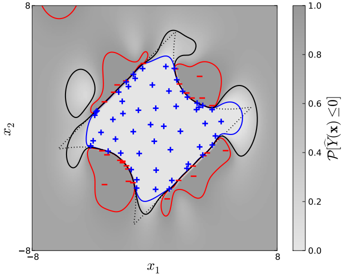
Convergence criteria depend on the *application*...

Illustration

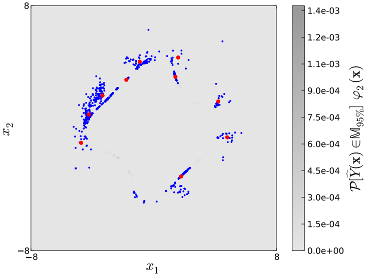
A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\hat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction



Sampled & clustered refinement criterion

Iteration #9

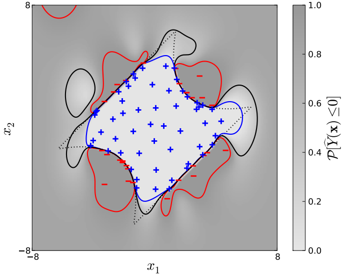
Convergence criteria depend on the *application*...

Illustration

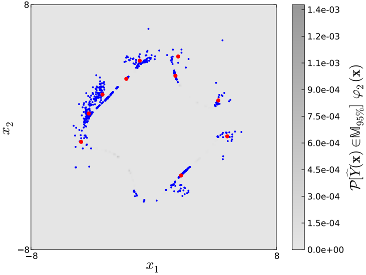
A four-branch series system

(Waarts, 2000)

$$C(\boldsymbol{x}) = \mathcal{P}\left[\widehat{Y}(\boldsymbol{x}) \in \mathbb{M}_{95\%}\right] \varphi_2(\boldsymbol{x}) \qquad K = 10$$



Target contour, design of experiments & prediction



Sampled & clustered refinement criterion

Iteration #9

Convergence criteria depend on the *application*...

Reliability analysis

Introduction

(Ditlevsen & Madsen, 1996; Lemaire, 2009)

Problem formulation

- Given a *failure domain*:

$$\mathbb{F} = \{\boldsymbol{x} \in \mathbb{X} : g(\boldsymbol{x}) \leq 0\}$$

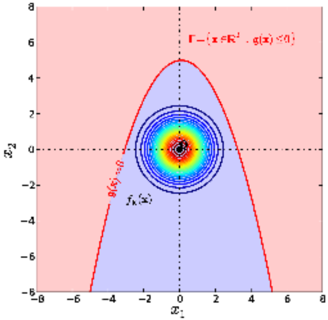
- and a *random vector* $\boldsymbol{X}$ with known distribution:

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = C\left(F_{X_i}(x_i),\, i = 1, \ldots, n\right)$$

(Lebrun & Dutfoy, 2009a,b,c)

- the purpose is to quantify the reliability of a design in the form of a *failure probability*:

$$p_f = \mathbb{P}\left[\boldsymbol{X} \in \mathbb{F}\right] = \int_{\mathbb{F}} f_{\boldsymbol{X}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$$



$\mathbb{P}(\neq \mathcal{P})$ denotes the probability measure w.r.t. the random vector $\boldsymbol{X}$.

Reliability analysis

Monte Carlo sampling as a motivation for the *structural reliability methods*

Monte Carlo sampling

- The failure probability rewrites:

$$p_f = \int_{\mathbb{X}} \mathbb{1}_{\mathbb{F}}(\boldsymbol{x}) \, f_{\boldsymbol{X}}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \mathbb{E}\left[\mathbb{1}_{\mathbb{F}}(\boldsymbol{X})\right]$$

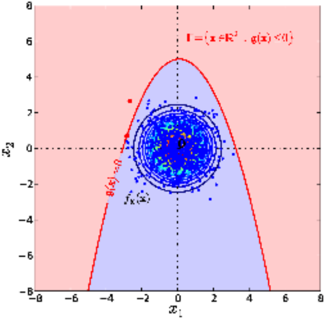
- Hence, the central limit theorem ensures that:

$$\widehat{p}_f = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\mathbb{F}}\left(\boldsymbol{X}^{(i)}\right) \hookrightarrow \mathcal{N}_1\left(p_f, \frac{p_f\,(1-p_f)}{N}\right)$$

- *provided N is sufficiently large!*
- In order to involve $\widehat{p}_f$ in an optimization loop:

$$p_f \approx 10^{-k} \Rightarrow N \geq 10^{k+2}$$

- *Structural reliability methods* aim at reducing N



Surrogate-based reliability analysis

Principle

- A *surrogate-based* estimator:

$$\tilde{p}_f = \int_{\tilde{\mathbb{F}}} f_X(x)\,\mathrm{d}x$$

- where:

$$\tilde{\mathbb{F}} = \{x \in \mathbb{X} : \tilde{g}(x) \leq 0\} \approx \mathbb{F}$$

and $\tilde{g}$ is a *meta-model* of g.

- $\tilde{g}$ is built from $m \ll N$ runs of g.

Error (bias) quantification?

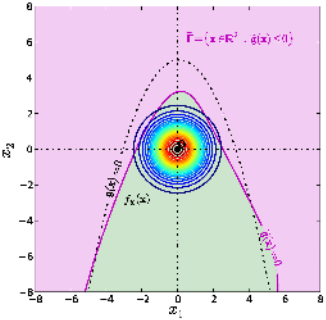
- Provided $\tilde{g}$ is a *Kriging predictor*:

$$\hat{\mathbb{F}}^{-1}_{95\%} \subseteq \hat{\mathbb{F}}^{0}_{95\%} \subseteq \hat{\mathbb{F}}^{+1}_{95\%} \Rightarrow p^{-1}_{f\,95\%} \leq p^{0}_{f\,95\%} \leq p^{+1}_{f\,95\%}$$

- Hence the following *empirical error*:

$$\Delta p_{f\,95\%} = \log_{10}\left(\frac{p^{+1}_{f\,95\%}}{p^{-1}_{f\,95\%}}\right)$$

For the sake of *efficiency* *low probabilities* can be handled by *subset sampling* (Au & Beck, 2001)



$$\Delta p_{f\,95\%} \leq \Delta_0$$

Surrogate-based reliability analysis

Principle

- A *surrogate-based* estimator:

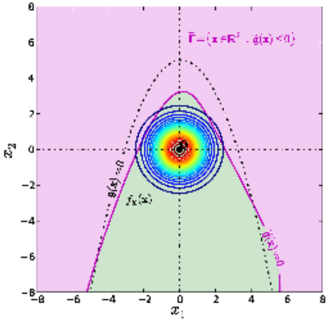
$$\widetilde{p}_f = \int_{\widetilde{\mathbb{F}}} f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

- where:

$$\widetilde{\mathbb{F}} = \{\boldsymbol{x} \in \mathbb{X} : \widetilde{\mathfrak{g}}(\boldsymbol{x}) \leq 0\} \approx \mathbb{F}$$

  and $\widetilde{\mathfrak{g}}$ is a *meta-model* of $\mathfrak{g}$.
- $\widetilde{\mathfrak{g}}$ is built from $m \ll N$ runs of $\mathfrak{g}$.

For the sake of *efficiency* *low probabilities* can be handled by *subset sampling* (Au & Beck, 2001)



Error (bias) quantification?

- Provided $\widetilde{\mathfrak{g}}$ is a *Kriging predictor*:

$$\widehat{\mathbb{F}}^{-1}_{95\%} \subseteq \widehat{\mathbb{F}}^{0}_{95\%} \subseteq \widehat{\mathbb{F}}^{+1}_{95\%} \Rightarrow p^{-1}_{f\,95\%} \leq p^{0}_{f\,95\%} \leq p^{+1}_{f\,95\%}$$

- Hence the following *empirical error*:

$$\Delta p_{f\,95\%} = \log_{10}\left(\frac{p^{+1}_{f\,95\%}}{p^{-1}_{f\,95\%}}\right)$$

$$\Delta p_{f\,95\%} \leq \Delta_0$$

Surrogate-based reliability analysis

Principle

- A *surrogate-based* estimator:

$$\tilde{p}_f = \int_{\tilde{\mathbb{F}}} f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

- where:

$$\tilde{\mathbb{F}} = \{\boldsymbol{x} \in \mathbb{X} : \tilde{\mathfrak{g}}(\boldsymbol{x}) \leq 0\} \approx \mathbb{F}$$

  and $\tilde{\mathfrak{g}}$ is a *meta-model* of $\mathfrak{g}$.
- $\tilde{\mathfrak{g}}$ is built from $m \ll N$ runs of $\mathfrak{g}$.

Error (bias) quantification?

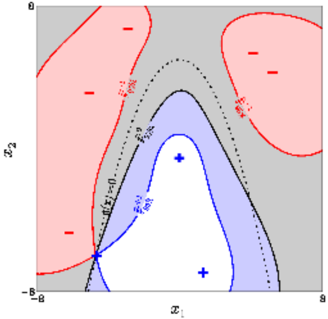
- Provided $\tilde{\mathfrak{g}}$ is a *Kriging predictor*:

$$\hat{\mathbb{F}}^{-1}_{95\%} \subseteq \hat{\mathbb{F}}^{0}_{95\%} \subseteq \hat{\mathbb{F}}^{+1}_{95\%} \Rightarrow p^{-1}_{f\,95\%} \leq p^{0}_{f\,95\%} \leq p^{+1}_{f\,95\%}$$

- Hence the following *empirical error*:

$$\Delta p_{f\,95\%} = \log_{10}\left(\frac{p^{-1}_{f\,95\%}}{p^{-1}_{f\,95\%}}\right)$$

For the sake of *efficiency*
low probabilities can be handled by
subset sampling (Au & Beck, 2001)



$\Delta p_{f\,95\%} \leq \Delta_0$

Surrogate-based reliability analysis

Principle

- A *surrogate-based* estimator:

$$\tilde{p}_f = \int_{\tilde{\mathbb{F}}} f_X(x)\,\mathrm{d}x$$

- where:

$$\tilde{\mathbb{F}} = \{x \in \mathbb{X} : \tilde{g}(x) \le 0\} \approx \mathbb{F}$$

  and $\tilde{g}$ is a *meta-model* of g.
- $\tilde{g}$ is built from $m \ll N$ runs of g.

Error (bias) quantification?

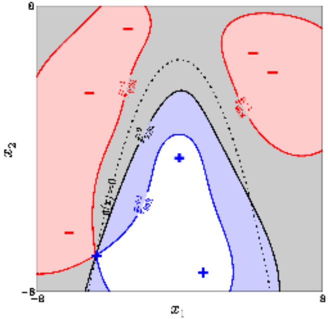
- Provided $\tilde{g}$ is a *Kriging predictor*:

$$\hat{\mathbb{F}}^{-1}_{95\%} \subseteq \hat{\mathbb{F}}^{0}_{95\%} \subseteq \hat{\mathbb{F}}^{+1}_{95\%} \Rightarrow p^{-1}_{f\,95\%} \le p^{0}_{f\,95\%} \le p^{+1}_{f\,95\%}$$

- Hence the following *empirical error*:

$$\Delta p_{f\,95\%} = \log_{10}\left(\frac{p^{-1}_{f\,95\%}}{p^{-1}_{f\,95\%}}\right)$$

For the sake of *efficiency* *low probabilities* can be handled by *subset sampling* (Au & Beck, 2001)



$$\Delta p_{f\,95\%} \le \Delta_0$$

Reliability-based design optimization

Introduction

(Tsompanakis et al.*, 2008)*

Problem formulation

$$d^* = \arg\min_{d \in \mathbb{D}} \; \mathfrak{c}(d) : \begin{cases} \mathfrak{f}_i(d) \leq 0, & i = 1, \ldots, n_c \\ p_{f\,l}(d) \leq p^0_{f\,l}, & l = 1, \ldots, n_p \end{cases}$$

where d is exclusively involved in the definition of the random vector X (*e.g. mean values*).

Bottlenecks

- The *repeated* reliability estimations are *computationally expensive*;
- Most NLP constrained optimization algorithms require *the gradients of the failure probabilities*.

Solutions

- Nested approaches *(Enevoldsen & Sørensen, 1994)*
- Sequential approaches *(Du & Chen, 2004)*
- Surrogate-based approaches *(Eldred* et al.*, 2002)*

Surrogate-based RBDO
The augmented reliability space

(*Taflanidis & Beck, 2008, 2009a,b*)

Motivation

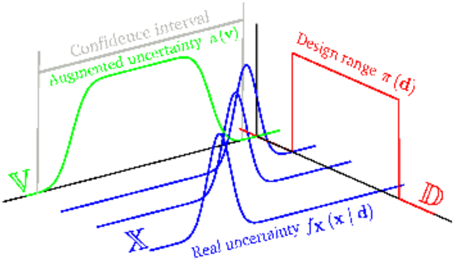
Building the Kriging surrogates *from scratch* for each nested reliability analysis would be particularly inefficient.

Definition

- The admissible range $\mathbb{D}$ simply *augments the spread* of f_X:

$$h(x) = \int_{\mathbb{D}} f_X(x \mid d)\, \pi(d)\, \mathrm{d}d$$

  where π is the uniform distribution over $\mathbb{D}$.
- The idea is to work on a *sufficiently large confidence region* of h.



Surrogate-based RBDO
Reliability sensitivity analysis

Motivations

- NLP optimization algorithms require *the gradient of the failure probabilities*;
- How to compute these derivatives with *Monte Carlo* techniques?

The score function approach *(Rubinstein, 1976, 1986)*

Given a random vector $\boldsymbol{X}$ with parameter d, *provided its support $\mathbb{X}$ does not depend on d*:

$$\frac{\partial p_f(d)}{\partial d} = \mathbb{E}_{\boldsymbol{X}}\left[\mathbb{1}_{\mathbb{F}}(\boldsymbol{X})\,\frac{\partial \log f_{\boldsymbol{X}}(\boldsymbol{X} \mid d)}{\partial d}\right]$$

Interesting properties

- *A simple post-processing* of a reliability analysis!
- The **score function** comes *analytically* when the copula formalism is used.
  (Lee et al., 2011a,b)
- The approach extends to *reduction variance techniques* such as:
  - subset sampling (Song et al., 2009)
  - (meta-model-based) importance sampling (Dubourg, 2011)

Surrogate-based RBDO
Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

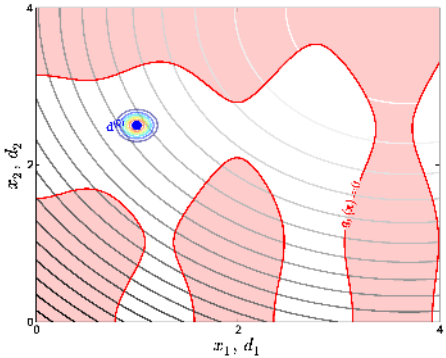
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO
Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

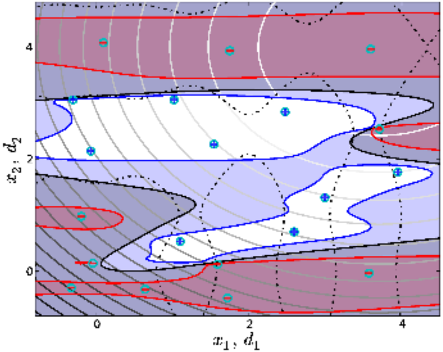
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO
Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

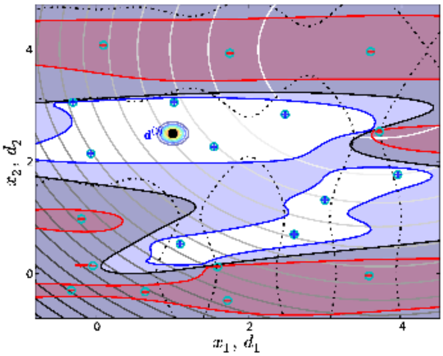
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

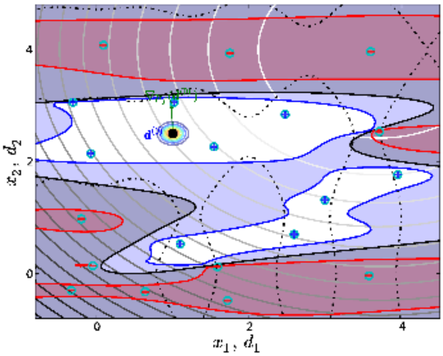
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

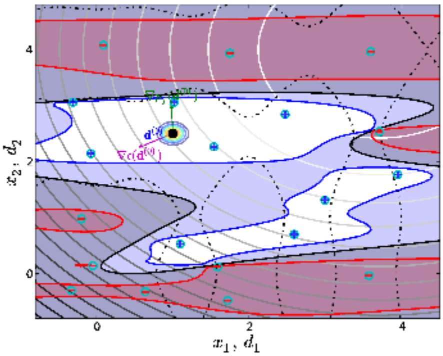
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $d^{(0)} \in \mathbb{D}$ (*bounded*):

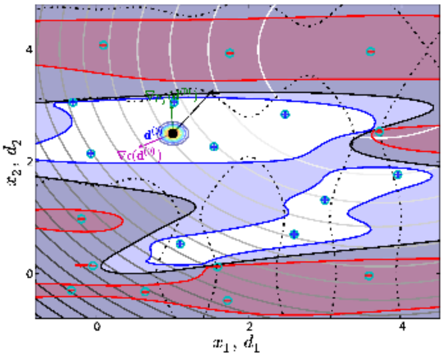
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(d^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(d^{(i)})$ and *its gradient*;
4. Compute *search direction* $h^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$d^{(i+1)} = d^{(i)} + s^{(i)}\, h^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

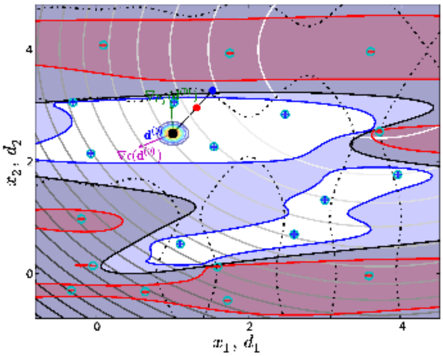
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

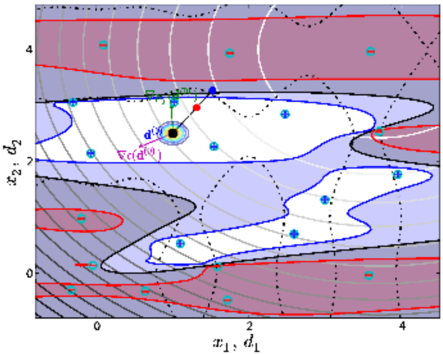
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

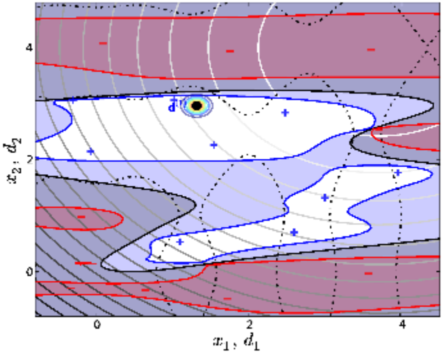
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO
Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

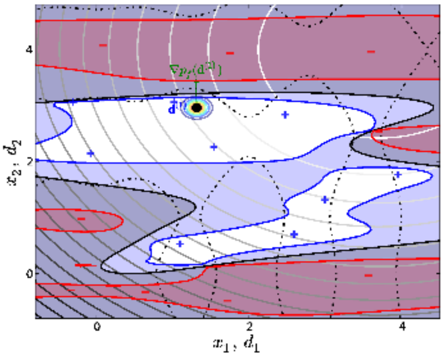
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

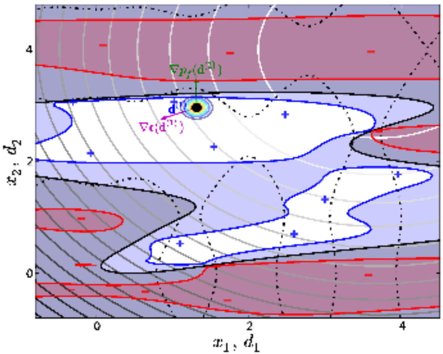
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO
Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

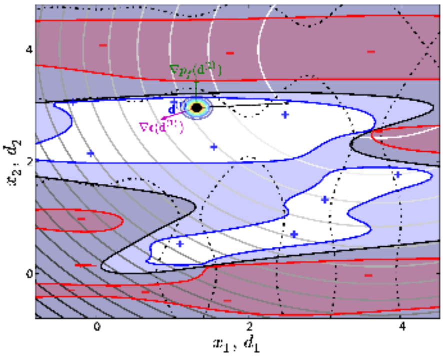
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

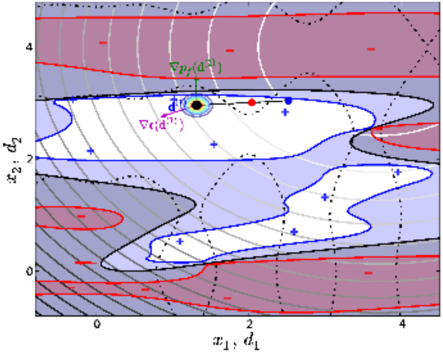
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

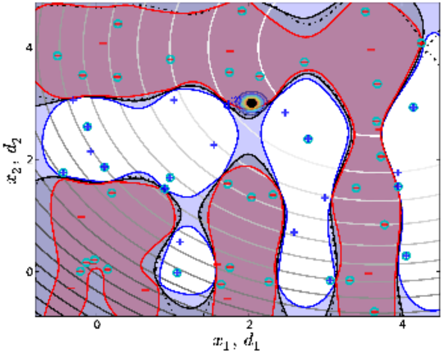
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

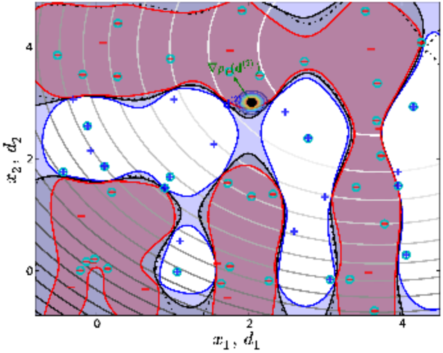
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

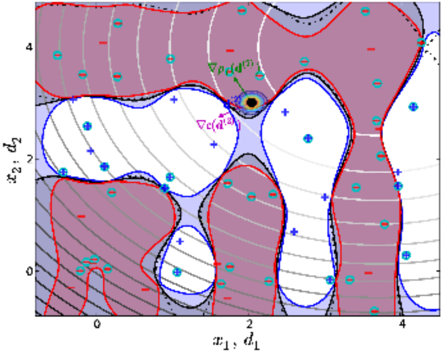
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

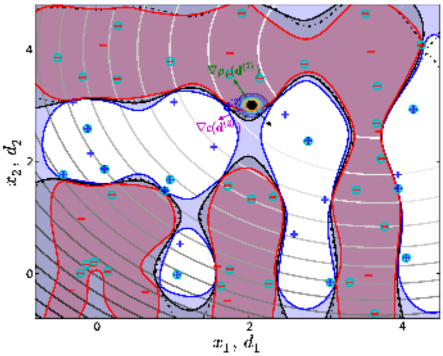
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

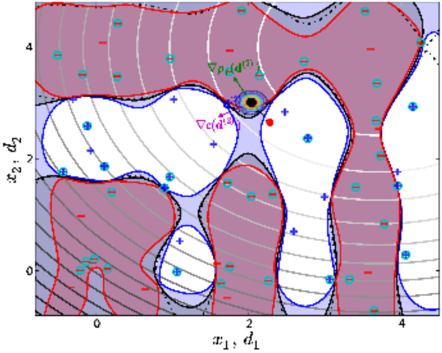
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

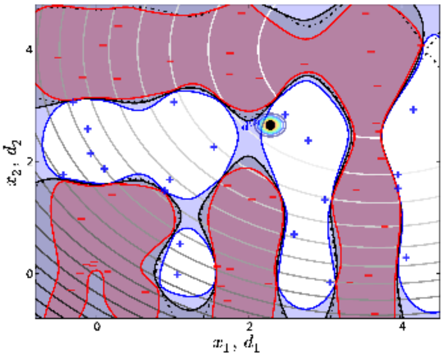
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

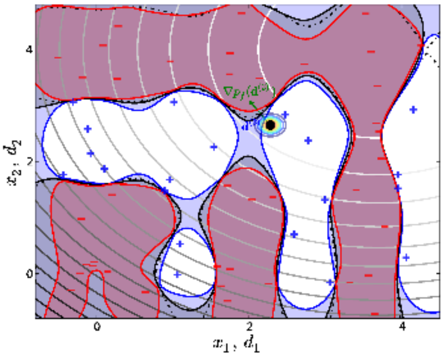
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

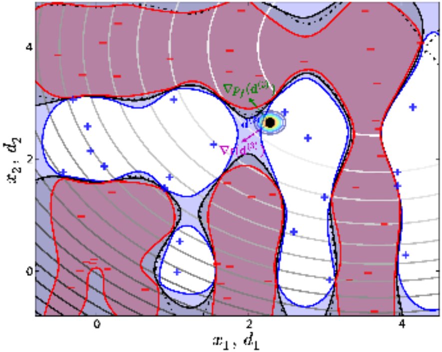
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

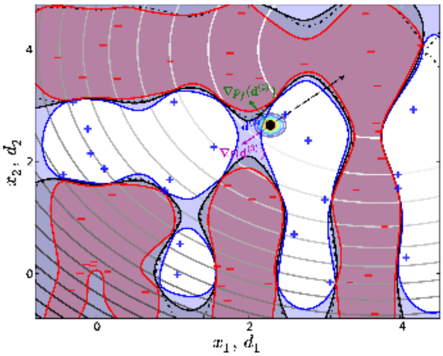
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO
Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

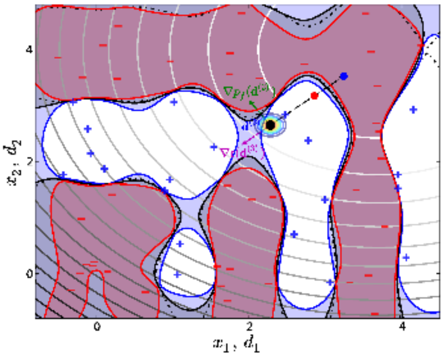
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

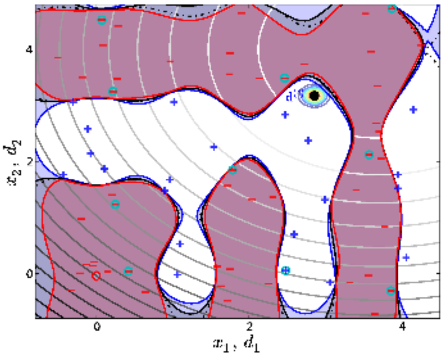
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

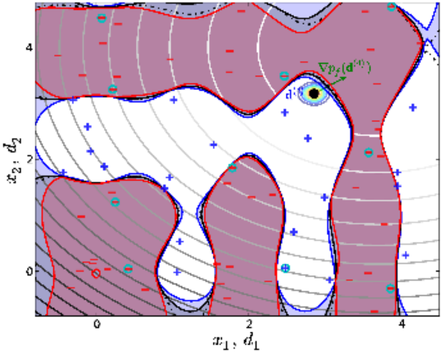
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

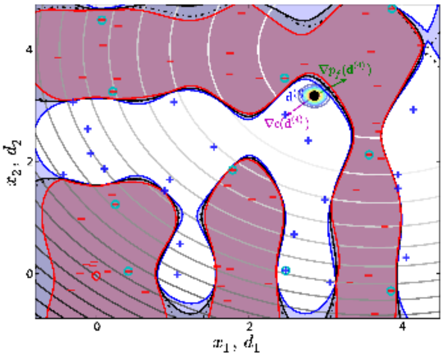
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

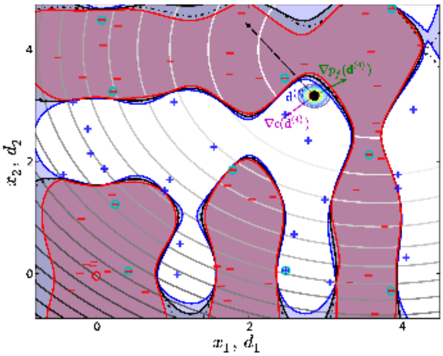
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.



Surrogate-based RBDO
Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

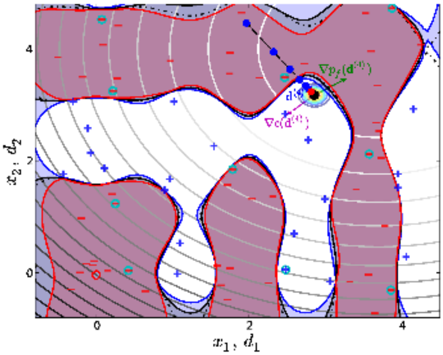
$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.

Optimizer converged!

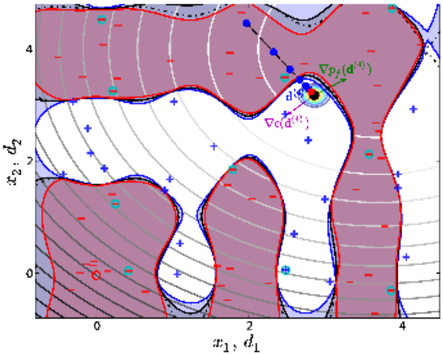


Surrogate-based RBDO
Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:
   $$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$
3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;
   $$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$
6. Loop back to step 2 until the *optimizer converges*.

Optimizer converged!



Shrink $\mathbb{D}$ and decrease Δ_0

Surrogate-based RBDO

Overview of the proposed algorithm

Given an initial design $\boldsymbol{d}^{(0)} \in \mathbb{D}$ (*bounded*):

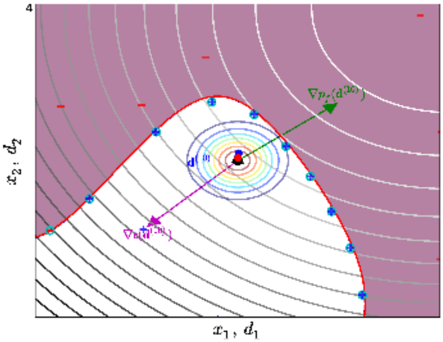
1. Determine the *augmented reliability space*;
2. Fit an *adaptive Kriging surrogate* with target local accuracy:

$$\Delta p_{f\,95\%}\left(\boldsymbol{d}^{(i)}\right) \leq \Delta_0$$

3. Compute $\hat{p}^{\,0}_{f\,95\%}(\boldsymbol{d}^{(i)})$ and *its gradient*;
4. Compute *search direction* $\boldsymbol{h}^{(i)}$ using min-max formulation;
5. Perform *approximate line-search* using Goldstein-Armijo step size rule;

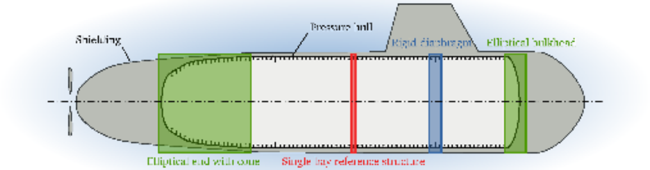
$$\boldsymbol{d}^{(i+1)} = \boldsymbol{d}^{(i)} + s^{(i)}\,\boldsymbol{h}^{(i)}$$

6. Loop back to step 2 until the *optimizer converges*.

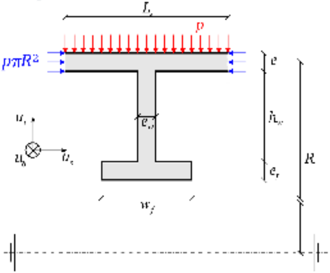


Converged!

Application to the design of an imperfect submarine hull
Problem formulation



Single bay reference structure



Design objectives

- The design should minimize the following *weight ratio*:

$$c(\boldsymbol{d}) = \frac{\rho_{\text{sea water}}\,\mathcal{V}_{\text{sea water}}(\boldsymbol{d})}{\rho_{\text{steel}}\,\mathcal{V}_{\text{steel}}(\boldsymbol{d})}$$

- while ensuring structural integrity for an *accidental depth charge*:

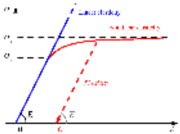
$$g(\boldsymbol{x}) = p_{\text{collapse}}(\boldsymbol{x}) - p_{\text{acc}}$$

Application to the design of an imperfect submarine hull

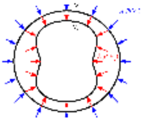
Mechanical and probabilistic modelling

Mechanical modelling *(Noirfalise, 2009)*

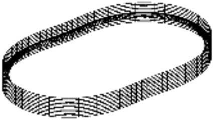
- Nonlinear finite element model (geometry, material and load);
- Shape imperfections distributed according to *two critical buckling patterns*



Nonlinear elasticity



Follower forces



Mode 2 (A_2)

Mode 14 (A_{14})

Probabilistic model

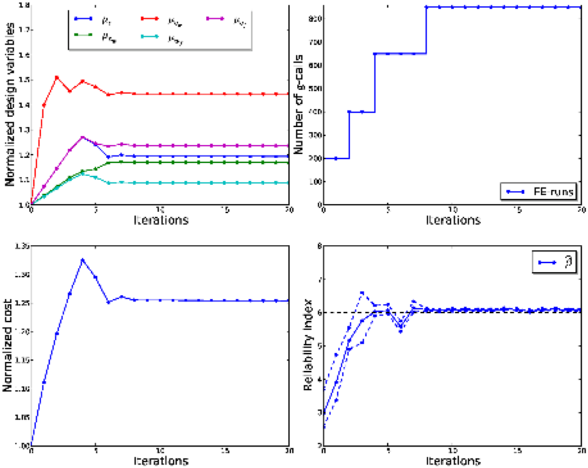
Variable		Distribution	Mean	C.o.V.
E	(MPa)	Lognormal	200,000	5%
σ_y	(MPa)	Lognormal	390	5%
σ_u	(MPa)	Lognormal	370	3%
e	(mm)	Lognormal	μ_e	3%
h_w	(mm)	Lognormal	μ_{h_w}	3%
e_w	(mm)	Lognormal	μ_{e_w}	3%
w_f	(mm)	Lognormal	μ_{w_f}	3%
e_f	(mm)	Lognormal	μ_{e_f}	3%
A_2	(mm)	Lognormal	$\frac{1}{3}\,\frac{5\,R}{1,000}$	50%
A_{14}	(mm)	Lognormal	$\frac{1}{3}\,\frac{L_s}{100}$	50%

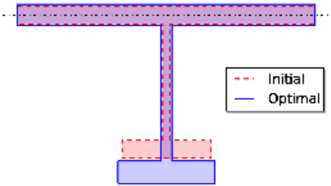
Probabilistic constraint

$$\mathbb{P}\left[p_{\text{collapse}}(\boldsymbol{X}) \leq p_{\text{acc}} \mid \boldsymbol{d}\right] \leq 10^{-9}$$

Application to the design of an imperfect submarine hull

Results





- +25% on the weight ratio
- but the failure probability was drastically reduced ($10^{-3} \to 10^{-9}$):

$$\hat{p}_{f\,\mathrm{metaIS}}(\boldsymbol{d}^*) = 10^{-9}$$

with a C.o.V. $\delta_{\mathrm{metaIS}} < 5\%$

- The whole procedure required about *1,000 FE runs*.

Conclusions & Open questions

Conclusions

- Universal Kriging enable an objective *quantification of the substitution error*;
- Sampling-based adaptive DOEs enabled a *reduction of this error*, while making the *use of distributed computing platforms* possible.
- The *augmented reliability space* ensures the coupling "optimization–reliability–surrogates" is efficient;
- *Subset sampling* still reveals *unavoidable* to deal with the *possibly low* failure probabilities encountered during the optimization;
- The score function approach revealed efficient for *reliability sensitivity analysis*;

Open questions

- Real engineering problems feature a *large number of parameters* (> 10): how to extend the use of kriging surrogates to such cases?
- Search for the *optimum optimorum*: use of global optimization techniques and/or better initialization of gradient-based optimizers?
- Some manufactured products benefits from *100% quality control*: how to deal with *truncated distributions* (zero probability, reliability sensitivity)?

Conclusions & Open questions

Conclusions

- Universal Kriging enable an objective *quantification of the substitution error*;
- Sampling-based adaptive DOEs enabled a *reduction of this error*, while making the *use of distributed computing platforms* possible.
- The *augmented reliability space* ensures the coupling "optimization–reliability–surrogates" is efficient;
- *Subset sampling* still reveals *unavoidable* to deal with the *possibly low* failure probabilities encountered during the optimization;
- The score function approach revealed efficient for *reliability sensitivity analysis*;

Open questions

- Real engineering problems feature a *large number of parameters* (> 10): how to extend the use of kriging surrogates to such cases?
- Search for the *optimum optimorum*: use of global optimization techniques and/or better initialization of gradient-based optimizers?
- Some manufactured products benefits from *100% quality control*: how to deal with *truncated distributions* (zero probability, reliability sensitivity)?

Thank you for your attention!

Au, S.-K. and J. Beck (2001).
Estimation of small failure probabilities in high dimensions by subset simulation.
Prob. Eng. Mech. 16(4), 263–277.

Dubourg, V. (2011).
Adaptive surrogate models for reliability analysis and reliability-based design optimization.
Ph. D. thesis, Université Blaise Pascal - Clermont II.

Lemaire, M. (2009).
Structural Reliability.
Wiley.

Polak, E. (1997).
Optimization algorithms and consistent approximations.
Springer.

Rubinstein, R. and D. Kroese (2008).
Simulation and the Monte Carlo method.
Wiley Series in Probability and Statistics. Wiley.

Santner, T., B. Williams, and W. Notz (2003).
The design and analysis of computer experiments.
Springer series in Statistics. Springer.

Song, S., Z. Lu, and H. Qiao (2009).
Subset simulation for structural reliability sensitivity analysis.
Reliab. Eng. Sys. Safety 94(2), 658–665.

Tsompanakis, Y., N. Lagaros, and M. Papadrakis (Eds.) (2008).
Structural design optimization considering uncertainties.
Taylor & Francis.

?