Parameter control in the presence of uncertainties

Victor Trappler
Supervisors: Elise Arnaud, Laurent Debreu, Arthur Vidard

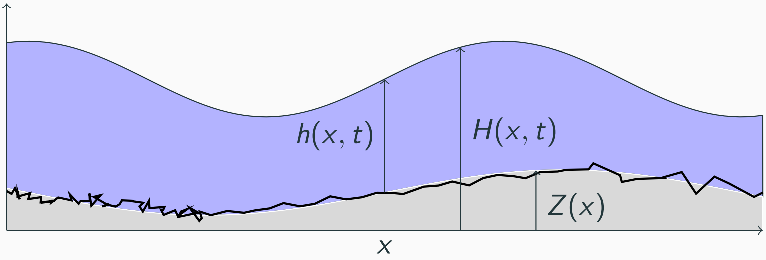
November 17, 2017

AIRSEA (Inria)– LJK

Introduction

Bottom friction

- The friction of the ocean bed has an influence on the water circulation
- Depends on the type and/or characteristic length of the asperities $\longrightarrow$ hard to observe
- Subgrid phenomenon $\longrightarrow$ Parametrization



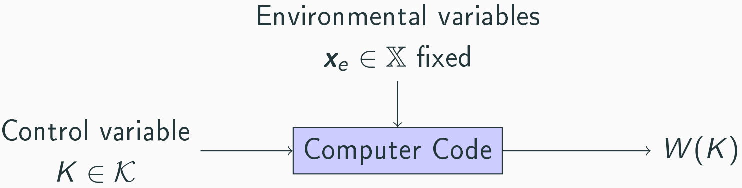
Outline

Deterministic problem

Computer code : the Shallow Water Equations

Input
- K: Bottom friction (spatially distributed)
- $\boldsymbol{x}_e$: Environmental variables (fixed and known)

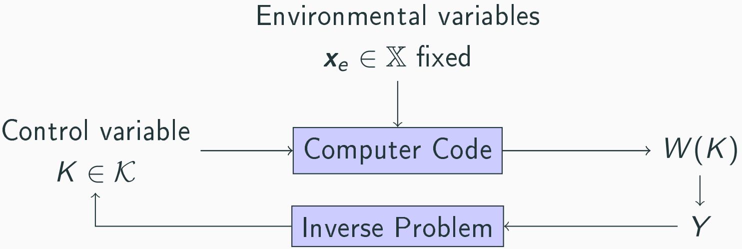
Output
- $W(K) = \{W_i^n(K)\}_{i,n}$, where $W_i^n(K) = [h_i^n(K) \quad q_i^n(K)]^T$ for $0 \leq i \leq N_x$ and $0 \leq n \leq N_t$



Computer code : the Shallow Water Equations

Input
- K: Bottom friction (spatially distributed)
- $\boldsymbol{x}_e$: Environmental variables (fixed and known)

Output
- $W(K) = \{W_i^n(K)\}_{i,n}$, where $W_i^n(K) = [h_i^n(K) \quad q_i^n(K)]^T$ for $0 \leq i \leq N_x$ and $0 \leq n \leq N_t$



Data assimilation framework: Twin experiments

K_{ref} and $\mathcal{H}$ observation operator

We have $Y = \mathcal{H}W(K_{\mathrm{ref}}) = \{h_i^n(K_{\mathrm{ref}})\}_{i,n}$

$$j(K) = \frac{1}{2}\|\mathcal{H}W(K) - Y\|^2$$

Data assimilation framework: Twin experiments

K_{ref} and $\mathcal{H}$ observation operator

We have $Y = \mathcal{H}W(K_{\mathrm{ref}}) = \{h_i^n(K_{\mathrm{ref}})\}_{i,n}$

$$j(K) = \frac{1}{2}\|\mathcal{H}W(K) - Y\|^2$$

$$\operatorname*{arg\,min}_{K\in\mathcal{K}} j(K)?$$

- Gradient-free: Simulated annealing, Nelder-mead,... $\rightarrow$ High number of runs, very expensive

Data assimilation framework: Twin experiments

K_{ref} and $\mathcal{H}$ observation operator

We have $Y = \mathcal{H}W(K_{\mathrm{ref}}) = \{h_i^n(K_{\mathrm{ref}})\}_{i,n}$

$$j(K) = \frac{1}{2}\|\mathcal{H}W(K) - Y\|^2$$

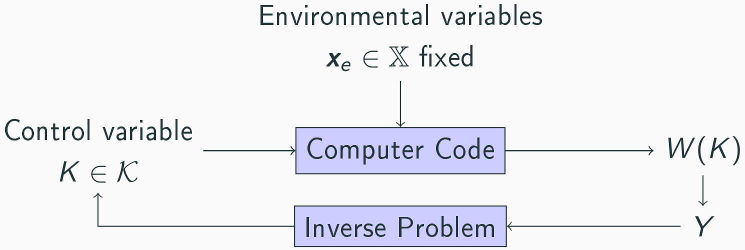
$$\operatorname*{arg\,min}_{K \in \mathcal{K}} j(K)?$$

- Gradient-free: Simulated annealing, Nelder-mead,... $\rightarrow$ High number of runs, very expensive
- Gradient-based: gradient-descent, (quasi-) Newton method $\rightarrow$ Less number of runs, but need the adjoint code

Dealing with uncertainties

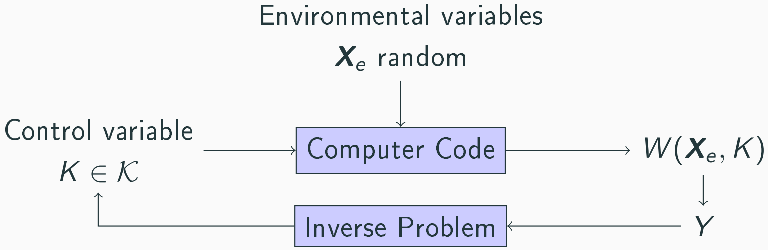
Introducing the uncertainties

Instead of considering $\boldsymbol{x}_e$ fixed, we consider that $\boldsymbol{X}_e$ is a random variable, and the output of the model depends on its realization.



Introducing the uncertainties

Instead of considering $\boldsymbol{x}_e$ fixed, we consider that $\boldsymbol{X}_e$ is a random variable, and the output of the model depends on its realization.



The cost function as a random variable

- Output of the computer code ($\boldsymbol{x}_e$ is an input):

$$W(K) \quad \text{becomes} \quad W(\boldsymbol{x}_e, K)$$

- The (deterministic) quadratic error is now

$$j(\boldsymbol{x}_e, K) = \frac{1}{2}\|\mathcal{H}W(\boldsymbol{x}_e, K) - Y\|^2$$

What to do with $j(\boldsymbol{X}_e, K)$ (r.v.) ?

Variational approach or Bayesian approach ?

- **Variational**: Minimize a function of $j(\boldsymbol{X}_e, K)$,
  e.g. Minimize $\mathbb{E}[j(\boldsymbol{X}_e, K)|K]$.
  $\longrightarrow$ Estimate efficiently $\mathbb{E}$ for a given K?

Variational approach or Bayesian approach ?

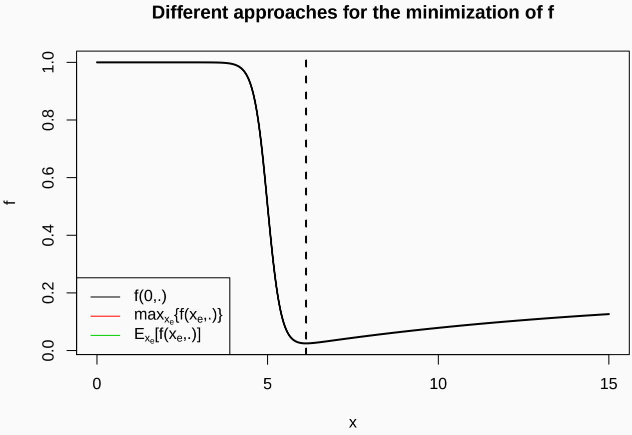
- **Variational**: Minimize a function of $j(\boldsymbol{X}_e, K)$,
  e.g. Minimize $\mathbb{E}[j(\boldsymbol{X}_e, K)|K]$.
  $\longrightarrow$ Estimate efficiently $\mathbb{E}$ for a given K?
- **Bayesian**: $e^{-j(\boldsymbol{x}_e, K)} \propto p(Y|K, \boldsymbol{X}_e)$ under gaussian assumptions.
  Find posterior distribution $p(K|Y)$ using inference and find Bayesian estimator and/or MAP
  $\longrightarrow$ Assumptions on errors ?

Robust minimization

An illustration

$$(x_e, K) \mapsto f(x_e, K) = \tilde{f}(x_e + K)$$

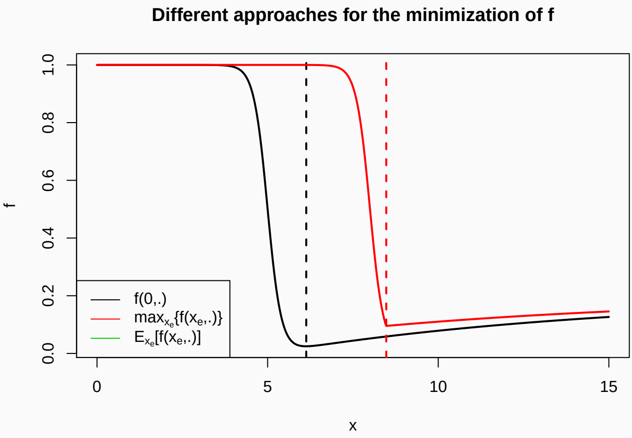
$X_e \sim \mathcal{N}(0, s^2)$ truncated on $[-3; 3]$. Plot of $f(0, \cdot) = \tilde{f}(\cdot)$



An illustration

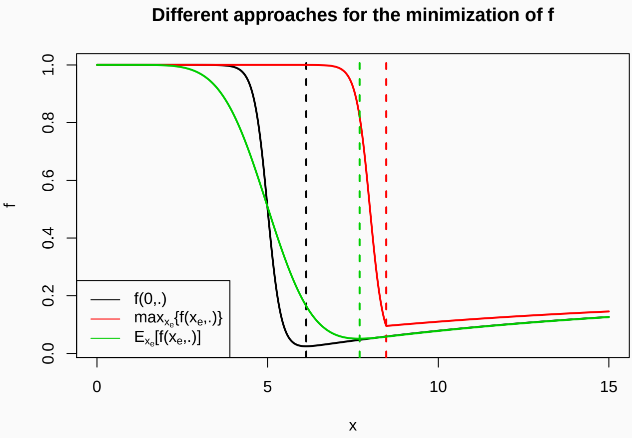
$(x_e, K) \mapsto f(x_e, K) = \tilde{f}(x_e + K)$

$X_e \sim \mathcal{N}(0, s^2)$ truncated on $[-3; 3]$. Plot of $\max_{x_e}\{f(x_e, \cdot)\}$



An illustration

$(x_e, K) \mapsto f(x_e, K) = \tilde{f}(x_e + K)$
$X_e \sim \mathcal{N}(0, s^2)$ truncated on $[-3; 3]$. Plot of $\mathbb{E}_{x_e}[f(x_e, \cdot)]$



Different Notions of robustness

- Global Optimum: $\min j(\boldsymbol{x}_e, K) \longrightarrow$ EGO

Different Notions of robustness

- Global Optimum: $\min j(\boldsymbol{x}_e, K) \longrightarrow$ EGO
- Worst case: $\min_K \max_{\boldsymbol{x}_e} j(\boldsymbol{x}_e, K) \longrightarrow$ Explorative EGO

Different Notions of robustness

- Global Optimum: $\min j(\boldsymbol{x}_e, K) \longrightarrow$ EGO
- Worst case: $\min_K \max_{\boldsymbol{x}_e} j(\boldsymbol{x}_e, K) \longrightarrow$ Explorative EGO
- M-robustness: $\min_K \mathbb{E}\left[J(\boldsymbol{X}_e, K)|K\right] \longrightarrow$ iterated LHS

Different Notions of robustness

- Global Optimum: $\min j(\boldsymbol{x}_e, K) \longrightarrow$ EGO
- Worst case: $\min_K \max_{\boldsymbol{x}_e} j(\boldsymbol{x}_e, K) \longrightarrow$ Explorative EGO
- M-robustness: $\min_K \mathbb{E}\left[J(\boldsymbol{X}_e, K)|K\right] \longrightarrow$ iterated LHS
- V-robustness: $\min_K \mathbb{V}\mathrm{ar}\left[J(\boldsymbol{X}_e, K)|K\right] \longrightarrow$ gradient-descent with PCE

Different Notions of robustness

- Global Optimum: $\min j(\boldsymbol{x}_e, K) \longrightarrow$ EGO
- Worst case: $\min_K \max_{\boldsymbol{x}_e} j(\boldsymbol{x}_e, K) \longrightarrow$ Explorative EGO
- M-robustness: $\min_K \mathbb{E}\left[J(\boldsymbol{X}_e, K)|K\right] \longrightarrow$ iterated LHS
- V-robustness: $\min_K \mathbb{V}\mathrm{ar}\left[J(\boldsymbol{X}_e, K)|K\right] \longrightarrow$ gradient-descent with PCE
- ρ-robustness: $\min \rho(J(\boldsymbol{X}_e, K)) \longrightarrow$ gradient-descent with PCE

Different Notions of robustness

- Global Optimum: $\min j(\boldsymbol{x}_e, K) \longrightarrow$ EGO
- Worst case: $\min_K \max_{\boldsymbol{x}_e} j(\boldsymbol{x}_e, K) \longrightarrow$ Explorative EGO
- M-robustness: $\min_K \mathbb{E}\left[J(\boldsymbol{X}_e, K)|K\right] \longrightarrow$ iterated LHS
- V-robustness: $\min_K \mathbb{V}\mathrm{ar}\left[J(\boldsymbol{X}_e, K)|K\right] \longrightarrow$ gradient-descent with PCE
- ρ-robustness: $\min \rho(J(\boldsymbol{X}_e, K)) \longrightarrow$ gradient-descent with PCE
- Multiobjective: choice within Pareto frontier $\longrightarrow$ 1L/2L kriging

Bayesian inference

Bayesian approach

Having observed Y, joint distribution of $(K, \boldsymbol{X}_e)$?

Bayes' Theorem

$$\begin{aligned} p(K, \boldsymbol{X}_e | Y) &\propto p(Y | K, \boldsymbol{X}_e) \pi(K, \boldsymbol{X}_e) \\ &\propto L(K, \boldsymbol{X}_e; Y) \pi(K) \pi(\boldsymbol{X}_e) \end{aligned}$$

Estimation of the posterior distribution: computationally expensive techniques such as Markov Chain Monte Carlo.

Why surrogates?

- Computer model: expensive to run
- $\dim \mathcal{K}$ can be very large
- Convenient way to introduce uncertainties upon $\boldsymbol{x}_e$ directly in the model

Environmental variables

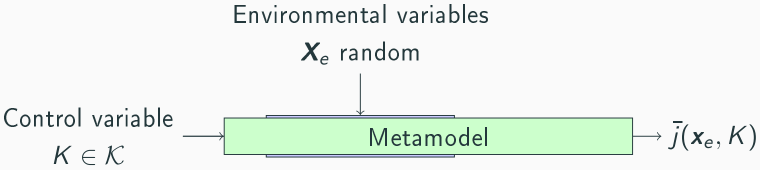
$\boldsymbol{X}_e$ random

$\downarrow$

Control variable $K \in \mathcal{K}$ $\longrightarrow$ $\boxed{\text{Computer Code}}$ $\longrightarrow W(\boldsymbol{x}_e, K) \longrightarrow j(\boldsymbol{x}_e, K)$

Why surrogates?

- Computer model: expensive to run
- $\dim \mathcal{K}$ can be very large
- Convenient way to introduce uncertainties upon $\boldsymbol{x}_e$ directly in the model

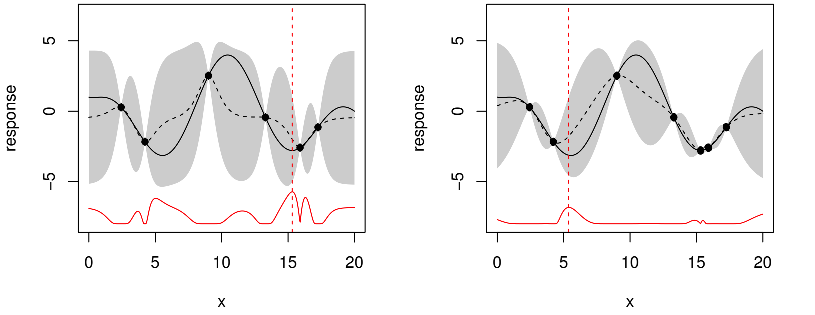


Using surrogates for optimization : adaptative sampling

Based on kriging model $\longrightarrow$ mean and variance

How to choose a new point to evaluate ? Criterion $\kappa(\boldsymbol{x}) \longrightarrow$ "potential" of the point

$$\boldsymbol{x}_{\text{new}} = \arg\max \kappa(\boldsymbol{x})$$



Conclusion

Conclusion

Wrapping up

- Variational and bayesian approaches for this inverse problem results in different methods
- In both case, these strategies rely heavily on surrogate models $\longrightarrow$ Kriging, Polynomial chaos

Perspective and future work

- Cost of computer evaluations $\rightarrow$ limit the total number of runs
- Dimensionality of the input space $\rightarrow$ reduction of the input space ?
- How to deal with uncontrollable errors $\rightarrow$ errors between model and reality ?