

Analyse de sensibilité déterministe pour la simulation numérique du transfert de contaminants

Estelle Marchand

financial support from ANDRA
additional support from GDR MOMAS

12 décembre 2007

- 1 Introduction
- 2 Statistical Analyses
- 3 Deterministic uncertainty and sensitivity analysis
 - First order approximation
 - Singular Value Decomposition
 - SVD and statistical analysis
- 4 Applications
 - Flow problem
 - Transport problem
- 5 Implementation aspects
 - Generic tools
 - C++ implementation of PIO
 - A generic tool for deterministic sensitivity analysis
- 6 Conclusions

- 1 Introduction
- 2 Statistical Analyses
- 3 Deterministic uncertainty and sensitivity analysis
 - First order approximation
 - Singular Value Decomposition
 - SVD and statistical analysis
- 4 Applications
 - Flow problem
 - Transport problem
- 5 Implementation aspects
 - Generic tools
 - C++ implementation of PIO
 - A generic tool for deterministic sensitivity analysis
- 6 Conclusions

Long term management of radionuclear waste

- long life waste : separation / transmutation → CEA
- reversible storage in deep geological layers → ANDRA
- storage in subsurface

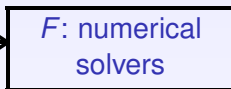
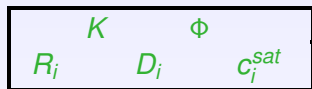
Storage in deep geological layers

- waste packages
- elaborated barrier
- geological barrier

Motivation

Modeling flow and transport around a nuclear waste storage site

subsurface properties,
contaminants properties:



safety indicators
(doses, water flow,
molar flow, concentrations)

Uncertainties about input data:
measurement techniques, spatial variability



uncertainties about
safety indicators

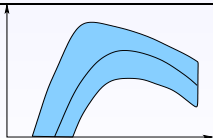
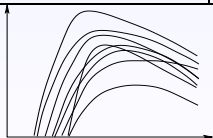
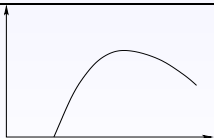
Question: how to evaluate the influence of the uncertainties about the input parameters on the safety indicators?

Sensitivity analysis

Evaluation of the uncertainties about the safety indicators due to the uncertainties about the input parameters.

Hierarchical classification of the input parameters according to their influence on the safety indicators.

Two approaches	Probabilistic (Monte Carlo)	Deterministic
Type of analysis	global	local
Implementation	simple	more complex
Computation time	very important	smaller



The two approaches are complementary.

- 1 Introduction
- 2 **Statistical Analyses**
- 3 Deterministic uncertainty and sensitivity analysis
 - First order approximation
 - Singular Value Decomposition
 - SVD and statistical analysis
- 4 Applications
 - Flow problem
 - Transport problem
- 5 Implementation aspects
 - Generic tools
 - C++ implementation of PIO
 - A generic tool for deterministic sensitivity analysis
- 6 Conclusions

Sampling

- **Generate a sample** of vectors **of input parameters** (LHS...)
- **Apply F** to each vector

Uncertainty analysis

Analyse the sample of output variables

- **scatterplots**
- **estimation** of means, of standard deviations

Sensitivity analysis

- **Analyse correlations** between input and output variables to determine which inputs influence the outputs:
computation of **statistical indicators**

difficulties	solution
Deal with strong nonlinearities of F	use of rank transformation
Separate influences of F and of input correlations	partial correlations, standard regressions

- F must be **monotoneous**

- 1 Introduction
- 2 Statistical Analyses
- 3 Deterministic uncertainty and sensitivity analysis**
 - First order approximation
 - Singular Value Decomposition
 - SVD and statistical analysis
- 4 Applications
 - Flow problem
 - Transport problem
- 5 Implementation aspects
 - Generic tools
 - C++ implementation of PIO
 - A generic tool for deterministic sensitivity analysis
- 6 Conclusions

Uncertainty analysis of the linearized problem

$F : \quad p \quad \mapsto \quad \phi$
vector of input parameters vector of output variables

Local study: analysis of the linearized problem $d\phi = F'(p)dp$

Computation of

- **uncertainties** about output variables of the **linearized** problem:

$$(|dp|, F') \rightarrow \text{upper bound for } |d\phi_i| = \sum_j |F'_{ij}| |dp_j|$$

- **statistical indicators** for the **linearized** problem
computed using formulas **without sampling** from F' and
correlations and standard deviations of the **input** parameters

Singular Value Decomposition

Local study: analysis of the linearized problem $d\Phi = F'(p)dp$

Definition

SVD of $F'(p)$: $F'(p) = USV^T$ where

- S : diagonal matrix

$s_1 \geq s_2 \geq \dots \geq 0$: singular values = $\sqrt{\text{eigenvalues of } F'(p)^T F'(p)}$

- U and V : orthogonal matrices

u_k, v_k columns of U, V : singular vectors

$$\begin{aligned} &F'(p)v_k = s_k u_k \\ \text{or } &v_k \in \mathbf{ker} F'(p) \\ \text{or } &u_k \in (\mathbf{im} F'(p))^\perp \end{aligned}$$

→ Hierarchical classification of inputs according to influence on outputs

SVD and statistical analysis

Proper orthogonal decompositions: Bibliographic study

- Simplify the simulation of a random vector x by decomposing

$$x = \sum s_k u_k$$

- with u orthogonal basis and s_k random variables
- variance concentrated in the first terms of the decomposition
- by **eigenvalue decomposition** of the **covariance** matrix of x
- approximation using a sample of x : **SVD** of the sample

Extension to sensitivity analysis?

- $y = F(x)$, x random vector, F deterministic model
- We assume components of x are independent and $E(xx^T) = \mathbf{I}$.
- We compute **SVD** of $\hat{F} = \Sigma_{y,x} \Sigma_{x,x}^{-1}$
- We obtain changes of variables in input and output spaces s. t.
 - new input components are still **independent**
 - each new output is **correlated with at most one** new input

Computation of the derivatives of F

Approximate differentiation

- Divided differences (DD) are either expensive or inaccurate
→ to be used for validation only

Exact differentiation

Adjoint or reverse mode / direct mode

- Automatic differentiation (AD) by operator overloading with `Ado1C`
- **Manual differentiation** (MD): implementation of analytical formulas

Performance comparison

- AD with `Ado1C` is competitive with MD concerning running time
- Memory management is difficult with AD
- AD is more convenient than DD to validate MD.

- 1 Introduction
- 2 Statistical Analyses
- 3 Deterministic uncertainty and sensitivity analysis
 - First order approximation
 - Singular Value Decomposition
 - SVD and statistical analysis
- 4 Applications**
 - Flow problem
 - Transport problem
- 5 Implementation aspects
 - Generic tools
 - C++ implementation of PIO
 - A generic tool for deterministic sensitivity analysis
- 6 Conclusions

Flow model: stationary Darcy's law

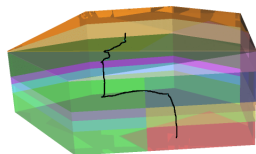
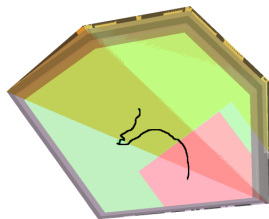
$$\begin{cases} \operatorname{div} \vec{\mathbf{u}} = q & \Omega \\ \vec{\mathbf{u}} = -\mathbf{K} \vec{\nabla} \rho & \Omega \\ \rho = \bar{\rho} & \Gamma_D \\ \vec{\mathbf{u}} \cdot \vec{\nu} = g_N & \Gamma_N \end{cases}$$

F_i associates water flow ϕ_i through outlet channel S_i to input parameters $\log \mathbf{K}$:

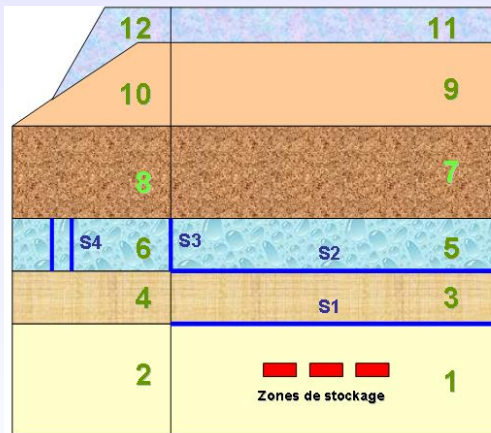
$$F_i : \log \mathbf{K} \rightarrow \phi_i = \int_{S_i} \vec{\mathbf{u}} \cdot \vec{\nu} .$$

- \mathbf{K} is piecewise constant.
- Computed with a mixed hybrid finite element formulation.

Hydraulic test case



Display of a stream line



2D vertical cut

Simplified 3D hydrological model

Spectrum of possible input parameters

Global probabilistic study

- Truncated lognormal laws of bounds *min*, *max*

Variable	K_v (m.s ⁻¹)	K_h/K_v	K_4/K_3	K_5 (m.s ⁻¹)	K_6/K_5
<i>min</i> , <i>max</i>	10^{-14} , 10^{-12}	1, 10^2	10, 10^3	10^{-9} , 10^{-7}	10, 10^3

- $K_3/K_v \rightsquigarrow$ uniform law of bounds *min* = 10, *max* = 1000

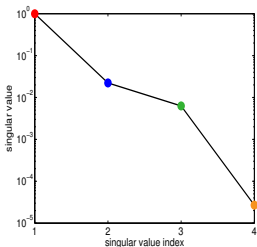
Choice of points for local deterministic studies with 6 parameters

- Most probable set of parameters (1 study)
- One parameter is set to *min* or *max* (2×6 studies)
- All parameters are set to *min* or *max* (2^6 studies)

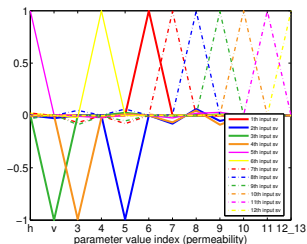
Sensitivity results for most probable set of parameters

$$F'(\log K) v_k = s_k u_k$$

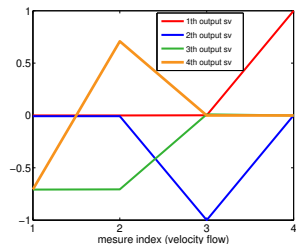
Singular values S_k



input space singular vectors V_k



output space singular vectors U_k



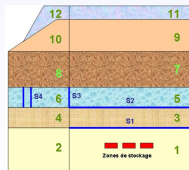
Hierarchical classification of local influences:

$\log(K_6)$ on Φ_4

$\log(K_5)$ on Φ_3

$\log(K_v)$ on $\Phi_1 + \Phi_2$

$\log(K_3)$ on $\Phi_1 - \Phi_2$ (almost zero)



Sensitivity results for a less probable set of parameters

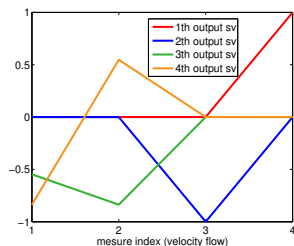
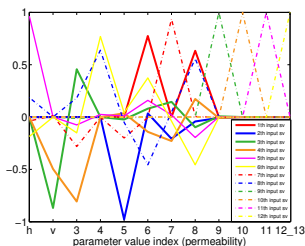
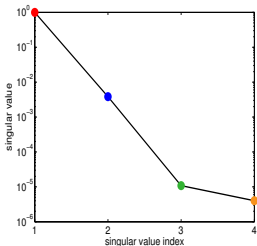
$$F'(\log K) v_k = s_k u_k$$

Chosen parameters: $\max K_6$ and $\max(\log K_3 - \log K_v)$

Singular values S_k

input space singular vectors V_k

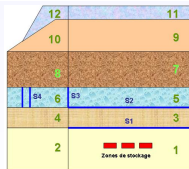
output space singular vectors U_k



Modifications of the local influences:

$\log(K_6) + \log(K_8)$ on Φ_4

v_3 and v_4 slightly modified



Sensitivity results for a less probable set of parameters

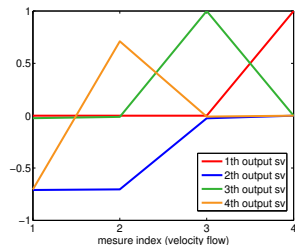
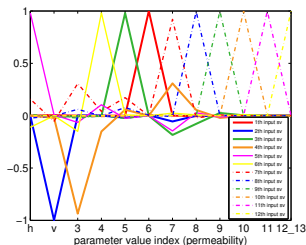
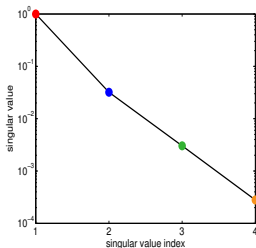
$$F'(\log K) v_k = s_k u_k$$

Chosen parameters: $\max(\log K_5 - \log K_v)$

Singular values S_k

input space singular vectors V_k

output space singular vectors U_k

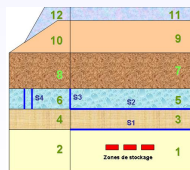


Local influences are reordered:

$\ln(K_6)$ on Φ_4

$\ln(K_v)$ on $\Phi_1 + \Phi_2$

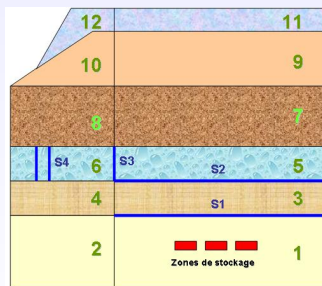
$\ln(K_5)$ on Φ_3



Results of probabilistic study: indicator of correlation between the water flows and the input parameters (*)

(*) Laurent Loth and Guillaume Pépin, ANDRA

	Φ_1	Φ_3	Φ_4
$\log K_h$	0.10	0.03	0.01
$\log K_v$	1.00	0.05	0.01
$\log K_3$	0.08	0.00	0.03
$\log K_4$	0.02	0.01	0.02
$\log K_5$	0.12	0.93	0.14
$\log K_6$	0.08	0.47	1.00



indicator = mean of the coefficients of Spearman, PRCC, SRRC

Transport model

$$\left\{ \begin{array}{l} \omega \frac{\partial c}{\partial t} + \operatorname{div} \left(c \vec{u} - \mathbf{D} \vec{\nabla} c \right) + \lambda c = q \quad \Omega \\ c = c_d \quad \Gamma_D \\ \left(c \vec{u} - \mathbf{D} \vec{\nabla} c \right) \cdot \vec{\nu} = g_F \quad \Gamma_F \\ \vec{\nabla} c \cdot \vec{\nu} = g_N \quad \Gamma_N \end{array} \right.$$

F_i^n associates contaminant flow Ψ_i^n through outlet channel S_i at t_n to weighted (log of) \mathbf{K} , ω , \mathbf{D} and q :

$$F_i^n : \text{input parameters} \rightarrow \Psi_i^n = \int_{S_i} \left(c \vec{u} - \mathbf{D} \vec{\nabla} c \right)^n \cdot \vec{\nu}.$$

- \vec{u} from flow equation
- Parameters are piecewise constant.
- Computed with a splitting method
 - explicit for convection (for precision), with finite volumes
 - implicit for diffusion (for stability), with mixed finite elements

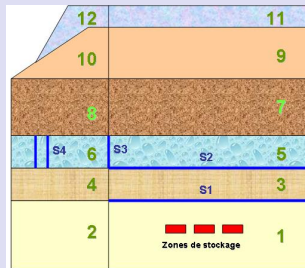
Flow+transport test case

Simplified model

- **Two dimensional** computation over a vertical non planar cut
- Computation over zones 1 to 6

11 uncertain parameters

- q (source term)
- ω over $\{1, 2, 3, 4\}$ and $\{5, 6\}$
- D over $\{1, 2, 3, 4\}$ and $\{5, 6\}$
- $k_h, k_v, k_3, k_4, k_5, k_6$



- Computation for one set of parameters
- Output = fluxes $(\Psi_1^{t_n}, \Psi_2^{t_n}, \Psi_3^{t_n}, \Psi_4^{t_n})$, at a given date t_n
- Output = fluxes $(\Psi_i^{t_1}, \Psi_i^{t_2}, \dots)$, through a given outlet channel S_i

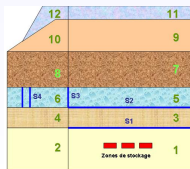
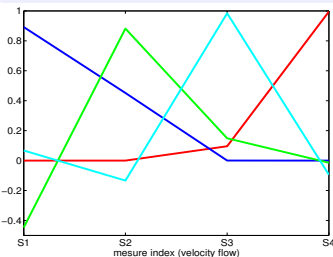
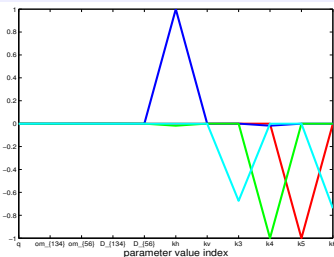
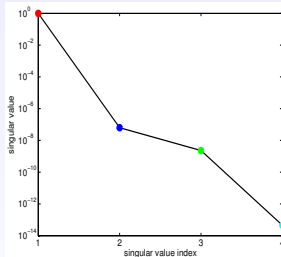
Sensitivity results for most probable set of parameters, $t_n = 10^6$ years

$$F'(\log K) v_k = s_k u_k$$

Singular values S_k

input space singular vectors V_k

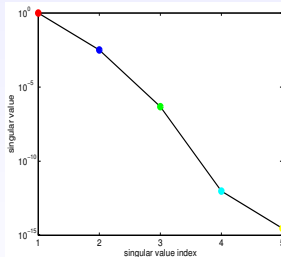
output space singular vectors U_k



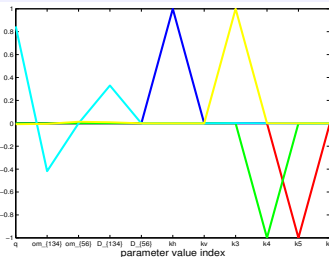
Sensitivity results for most probable set of parameters,
 output = $\Psi_2^{t1}, \dots, \Psi_2^{t5}$

$$F'(\log K) v_k = s_k u_k$$

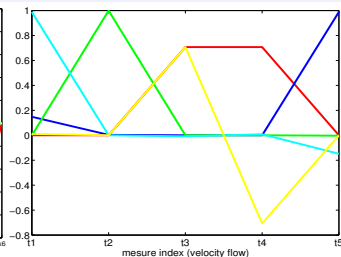
Singular values S_k



input space singular vectors V_k



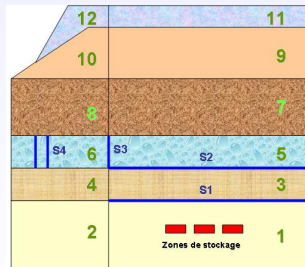
output space singular vectors U_k



Diffusion parameters then convection parameters.

Probabilistic results (ANDRA)

- Most influent parameters: $\Phi_{1,2,3,4}$, $D_{1,2,3,4}$, k_v
- For S_1 et S_2
 - weak influence of param zones 5,6
 - time decreasing influence of diffusion parameters
- For S_3
 - less weak influence of param zones 5,6



Conclusion about comparisons

Flow

- Easy to choose a non linear scaling
- Log parameterization is consistent with probabilistic data

Transport

- Scaling more difficult
- Stronger coupling between parameters → consistency with probabilistic data difficult

- 1 Introduction
- 2 Statistical Analyses
- 3 Deterministic uncertainty and sensitivity analysis
 - First order approximation
 - Singular Value Decomposition
 - SVD and statistical analysis
- 4 Applications
 - Flow problem
 - Transport problem
- 5 **Implementation aspects**
 - **Generic tools**
 - **C++ implementation of PIO**
 - **A generic tool for deterministic sensitivity analysis**
- 6 Conclusions

Possible structure for a program

- **Generic: algorithmic** part, difficult to set up
 - ex. domain decomposition algorithm
- **Specific:** computational part, **heavy**, already existing in libraries
 - ex. solve flow equation

Ideas

- **Avoid reimplementing** the generic part for each specific application
- Safe and expressive **functional language** used for the **generic** part

Paradigm for the implementation of generic tools – 1

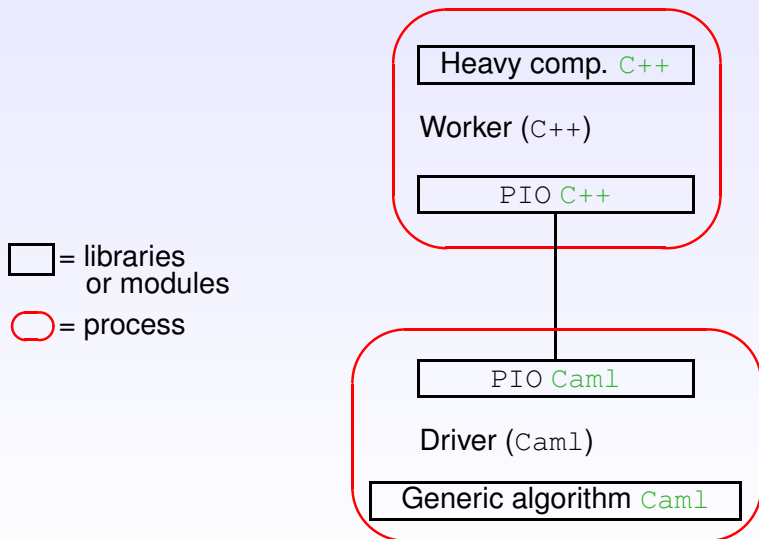
Paradigm

- **Heavy** computations → **external** programs (computational servers)
- Master program in **Cam1**
- **Communications** between external and master programs → **PIO**

To deal with heterogeneity of codes

- **Polyglot** **Input Output**, **agnostic communication library**
 - **Simple, robust** protocols
 - Ascii and binary modes
 - Polyglot: implemented in **Cam1, C++, C, Fortran**
in project: **Matlab, Java**

Paradigm for the implementation of generic tools – 2



- **Channels** = standard i/o channels or file streams
- A **communication** can be
 - a **Task** communication: a function name and its arguments,
 - a **Result** communication: a sequence of typed values,
 - an **Error** communication: an exception value.
- When a **Task** communication is received
 - **Try** to apply the function given by its name
 - If success: **Result** communication sent back
 - Otherwise: **Error** communication sent back

→ a communication is **always** sent back
- **Values**: one string, one integer, one float, one integer couple, one integer triple, one float couple, or one float triple, or vectors of such, or float matrices.

Caml: a strongly typed functional language

- Distributed by INRIA **since 1985** (<http://caml.inria.fr/>)
⇒ Unison, MLdonkey, Coq, FFTW, cdtools, ...
- High-level language based on **clear theoretical foundation** and **formal semantics**
- **Strong typing**: static and automatic
No automatic type conversion ⇒ **safety**
Segmentation fault or bus error **never happens**
A program that **compiles** is close to be **correct**
- **Functional language** ⇒ close to the mathematics

```
# let compose f g = function x -> g (f x) ;;  
compose : ( $\alpha \rightarrow \beta$ )  $\rightarrow$  ( $\beta \rightarrow \gamma$ )  $\rightarrow$   $\alpha \rightarrow \gamma$ 
```
- **Polymorphism, modules, objects** ⇒ complex structures/algorithms

Implementing with Caml requires a **different way of thinking**

- Based on preexisting Caml implementation
- Difficulty: **late** knowledge of concrete data types during communication
- Implementation of a general *UNION* type using standard C++ library

Implementation of the *UNION* type

From Caml and C union

- `class Elem<...>` to simulate **free pointers** ie “Something (of fixed type) or nothing”
 - One vector of length 0 (“empty”) or of length 1 (“full”)
 - Identifier, safe constructors, safe assignment
- A *UNION* class
 - Several `Elem<...>` attributes: exactly one is “full”
 - Identifier, safe constructors, safe assignement

Information over communication channels

- **Most general type:** “Communication”
UNION composed of: Task, Result, Error
- **General type of data:** “Typed_value”
for the arguments of a Task or the components of a Result
UNION composed of: Matrix_type[1,2], Vector_type, Lexem
- Matrix_type[1,2], Vector_type, Lexem

Using a generic tool is easy

- The user writes (several) independent **external** programs in **any programming language**
- Only his main function must follow a particular specification (**infinite loop** receiving orders and returning results **using PIO**). Examples are provided in each language
- It is not necessary to know `Cam1` language to use the generic tool

Generic tool for deterministic sensitivity analysis

Steps for an analysis

- **Initialize**: read mesh, compute values ... Returns the **size** of the Jacobian matrix
- Compute **rows** or **columns** of F'
- **Assemble** the Jacobian matrix
- **Parameterize**
- Compute **SVD**

We can use **several** generic tools together (for example computation of a column of F' using domain decomposition)

Outline

- 1 Introduction
- 2 Statistical Analyses
- 3 Deterministic uncertainty and sensitivity analysis
 - First order approximation
 - Singular Value Decomposition
 - SVD and statistical analysis
- 4 Applications
 - Flow problem
 - Transport problem
- 5 Implementation aspects
 - Generic tools
 - C++ implementation of PIO
 - A generic tool for deterministic sensitivity analysis
- 6 Conclusions

Local deterministic sensitivity analysis for a flow and transport problem

- **Exact computation** of the Jacobian matrix, use `AdolC` for validation
- SVD of Jacobian matrix provides **locally** a **hierarchy** for influences of the inputs on the outputs
- Cheap → **several** local analyses

Conclusions

Comparison with global probabilistic results for the **flow** problem

- For **this example**: **weak variability** of the results of the deterministic study \leftrightarrow weak nonlinearity of the model in the spectrum of possible input parameters
- For **this example** probabilistic and deterministic studies provide very **similar results**
- A strong variability of the deterministic results should lead to be wary of the results of the probabilistic study

Difficulties for the transport problem

- Scaling
- Consistant parameterization

Software development

- Development of a generic tool for sensitivity analysis
- PIO C++

Perspective

- Use it with parallel domain decomposition, with Neumann-Neumann preconditioning and balancing
- Go on connecting probabilistic and deterministic analyses with numerical applications

THANK YOU FOR YOUR ATTENTION