NATIONAL TECHNICAL UNIVERSITY OF ATHENS (NTUA)
SCHOOL OF MECHANICAL ENGINEERING
PARALLEL CFD & OPTIMIZATION UNIT (PCOpt/NTUA)

Uncertainty Quantification (UQ) - Optimisation under Uncertainties / Robust Design Optimisation (RDO)

Dr. Kyriakos C. Giannakoglou, Professor NTUA

Dr. Varvara Asouti, Adjunct Lecturer NTUA

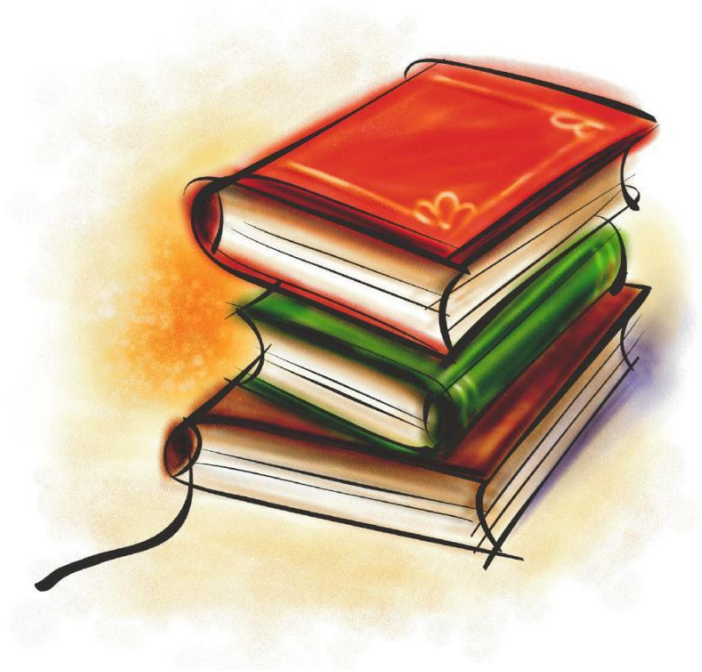
Contributors:

Dr. E. Papoutsis-Kiachagias

Dr. T. Skamagkis

I. Lyras

V. Kachrimani



Introduction

Prof. K.C. Giannakoglou, kgianna@mail.ntua.gr, Dr. V.G. Asouti, vasouti@mail.ntua.gr

Introduction

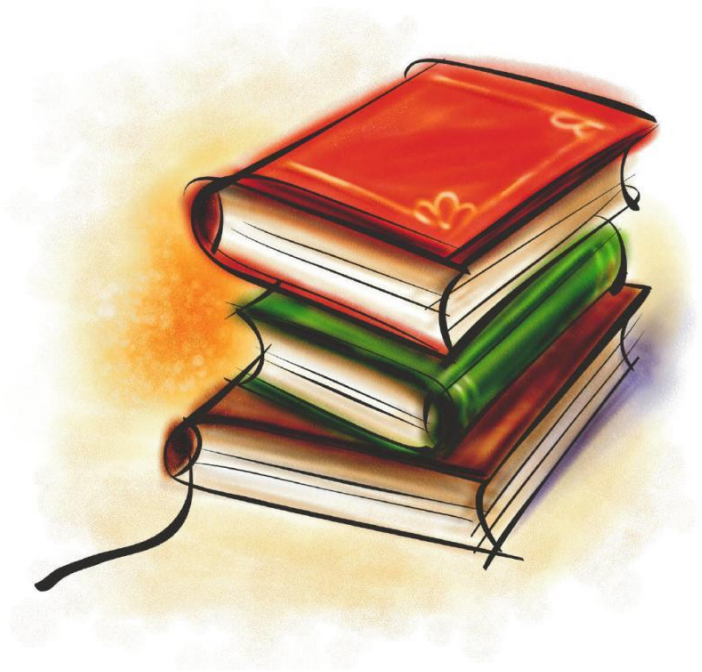
- ✓ **CFD-based aerodynamic shape optimisation (ShpO) problem: minimise/maximise an objective function J (e.g. drag, lift) by modifying the shape of the aerodynamic body (e.g. airfoil, car).**
- ✓ **The CFD analysis and/or ShpO are carried out by assuming that all inputs (boundary conditions, fluid properties, geometry, etc) are known with certainty.**
- ✓ **<u>This is rarely the case in real-world problems.</u>**

- ✓ **Uncertain variables/inputs, $c_i, i \in [1, M]$, exist and these are associated with known (or assumed) mean values μ_i and standard deviations σ_i**

- ✓ **Quantity of Interest (QoI) vs. Objective Function.**
- ✓ **Uncertainty Quantification (UQ): Propagate uncertainties related to the output J (QoI), by "seeing" it as a statistical quantity with mean value μ_J and standard deviation σ_J.**
- ✓ **Robust Design Optimisation (RDO) or Optimisation under Uncertainties (OuU): Minimise or maximise (a combination of) μ_J and σ_J.**

UQ Methods:

- **Monte – Carlo**
- **Polynomial Chaos Expansion (PCE)**
- **Sparse PCE**
- **Adjoint-assisted PCE**
- **Method of Moments (MoM)**
- **Machine Learning (ML) variants of these methods**
- **…**



Polynomial Chaos Expansion (PCE)

Prof. K.C. Giannakoglou, kgianna@mail.ntua.gr, Dr. V.G. Asouti, vasouti@mail.ntua.gr

Polynomial Chaos Expansion (PCE)

- Expands the QoI (J) as a weighted sum of orthogonal polynomials $H_i(\vec{c})$

$$J(\vec{c}) \cong \sum_{i=0}^{Q-1} b_i H_i(\vec{c})$$

- H_i: Hermite polynomials for a normal distribution of $\vec{c}$. Other distributions?
- k: chaos order (largest polynomial degree)
- M: number of uncertain variables
- $Q = \frac{(k+M)!}{k!M!}$: number of polynomial coefficients
- Statistical moments of the QoI:

$$\mu_J = b_0, \qquad \sigma_J \cong \sqrt{\sum_{i=1}^{Q-1} b_i^2}$$

- ✓ b_i can be computed by just evaluating J in a number of $\vec{c}$ instances
- ✓ The evaluation tool is used as a black box
- × Q grows (very) fast with increasing M and k

Polynomial Types for know PDFs

Distribution	PDF	Polynomial	Support range
Normal	$\frac{1}{\sqrt{2\pi}}e^{-\xi^2/2}$	Hermite	$(-\infty,\infty)$
Uniform	$\frac{1}{2}$	Legendre	$[-1,1]$
Beta	$\frac{(1-\xi)^\alpha(1+\xi)^\beta}{2^{\alpha+\beta+1}B(\alpha+1,\beta+1)}$	Jacobi	$[-1,1]$
Exponential	$e^{-\xi}$	Laguerre	$[0,\infty)$
Gamma	$\frac{\xi^\alpha e^{-\xi}}{\Gamma(\alpha+1)}$	Generalized Laguerre	$[0,\infty)$

Polynomial Chaos Expansion (PCE) Variants

- **Non-intrusive PCE:**
  - PCE applied to the QoI, e.g. the lift coefficient, c_L.
  - No software development, the evaluation tool used as 'black box'.
  - b_i can be computed by just evaluating c_L at a number of $\vec{c}$ instances.

$$c_L \cong \sum_{i=0}^{Q-1} b_i H_i(\vec{c})$$

- **Intrusive PCE:**
  - PCE applied to each field variable U, where J=J(U).
  - New set of equations to find U_i through Galerkin projections.
  - Q iPCE set of equations.
  - QoIs are computed as in terms of the computed field variables.

$$U(\vec{x}) \cong \sum_{i=0}^{Q-1} U_i(\vec{x}) H_i(\vec{c})$$

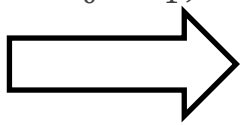
$x \in [0,1]$

$$\frac{df}{dx} + f^2 - h = 0$$
$$\frac{dh}{dx} + fh = 0$$
$$f(0) = 5 + 0.1\xi$$
$$h(0) = 1$$
$$\xi \sim N(0,1)$$

Chaos order k= 1

$f = f_0 + f_1\xi$

$h = h_0 + h_1\xi$



ξ *is the uncertain variable*

$$\frac{df_0}{dx} + f_0^2 + f_1^2 - h_0 = 0$$
$$\frac{df_1}{dx} + 2f_0f_1 - h_1 = 0$$
$$\frac{dh_0}{dx} + f_0h_0 + f_1h_1 = 0$$
$$\frac{dh_1}{dx} + 2f_1h_1 = 0$$
$$f_0(0) = 5\,,\ f_1(0) = 0.1$$
$$h_0(0) = 1\,,\ h_1(0) = 0$$

Two coupled systems of ODEs

Non-Intrusive Polynomial Chaos Expansion (PCE)

1. Gauss Quadrature (gPCE): $b_i = \int J(\vec{c}) H_i(\vec{c}) W(\vec{c}) d\vec{c}$

- Need to evaluate *J* at $(k+1)^M$ Gauss nodes. ***Exponential growth!***
- Use of sparse grids (Smolyak) to reduce the cost if *k>>* or *M>>*

2. Regression (rPCE):

- Formulate $J(\vec{c}) \cong \sum_{i=0}^{Q-1} b_i H_i(\vec{c})$ as a system of *Q* unknowns

$$\begin{bmatrix} H_0(\vec{c_0}) & H_1(\vec{c_0}) & \dots & H_{Q-1}(\vec{c_0}) \\ & \vdots & & \\ H_0(\vec{c}_{L-1}) & H_1(\vec{c}_{L-1}) & \dots & H_{Q-1}(\vec{c}_{L-1}) \end{bmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_{Q-1} \end{bmatrix} = \begin{bmatrix} J(\vec{c_0}) \\ \vdots \\ J(\vec{c}_{L-1}) \end{bmatrix}$$

- Sampling (e.q. using Latin hypercube sampling) of J, at L collocation points; solve a least squares problem.
- L depends on:

  (a) the total number of PCE coefficients: $Q = \frac{(k+M)!}{k!M!}$

  (b) the oversampling ratio r (# of eqs./# of coefs.).

gPCE – Polynomial Function Example

Assume a QoI with two uncertain variables (M=2):

$$J(\vec{c}) = J(c_1, c_2) = 4c_1 + c_2$$

$c_1 \sim N(2,1)$

$c_2 \sim N(1,9)$

Method	μ_f	%error	σ_f	%error
Analytical Solution	9	-	5	-
gPCE (9, k=2)	9	0	5	0
MC (10000)	9.09852	1.09467	4.97571	0.4858
MC (50000)	8.99734	0.02956	5.03779	0.7558
MC (100000)	8.99091	0.10101	4.99224	0.1552
MC (500000)	9.00236	0.02622	5.00125	0.0250

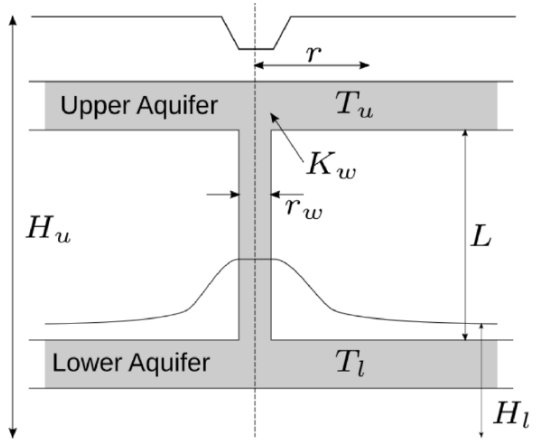
rPCE – Borehole Function

Mathematical model that describes the flow of water through a borehole

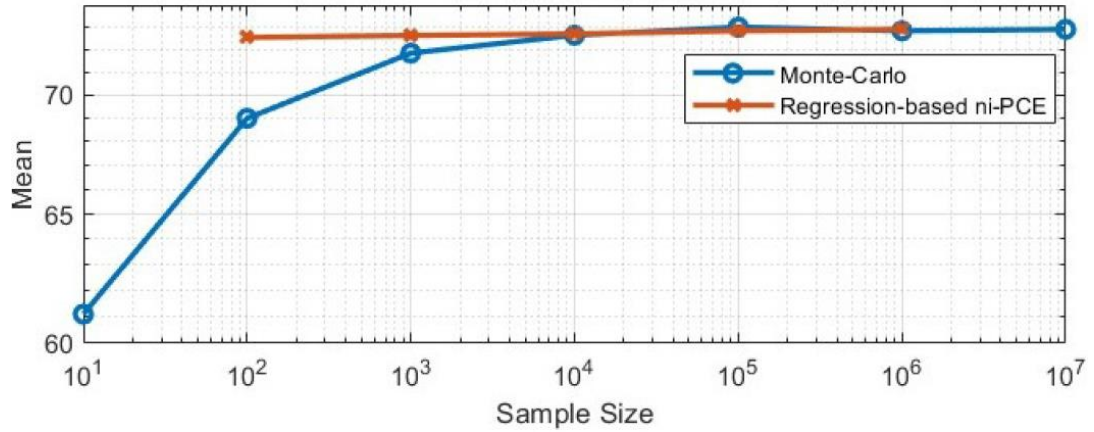
$$f(\vec{c}) = \frac{2\pi T_u (H_u - H_l)}{ln(\frac{r}{r_w})\left[1 + \frac{2LT_u}{ln\left(\frac{r}{r_w}\right) + r_w^2 K_w} + \frac{T_u}{T_l}\right]}$$

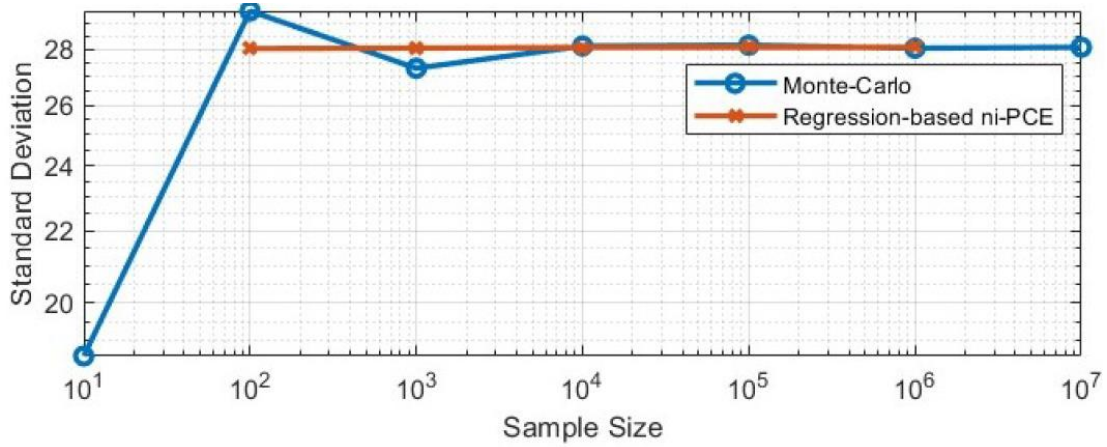
M=8 uncertain variables

$c_1 = r_w \sim N(0.1, 0.0161812)$
$c_2 = r \sim LogNor(7.71, 1.005)$
$c_3 = T_u \sim U[63070, 115600]$
$c_4 = H_u \sim U[990, 1110]$
$c_5 = T_l \sim U[63.1, 116]$
$c_6 = H_l \sim U[700, 820]$
$c_7 = L \sim U[1120, 1680]$
$c_8 = K_w \sim U[9855, 12045]$



Method	μ_f	σ_f	Cost
MC	73.6985	28.3713	10^6
rPCE	73.5399	28.3677	150



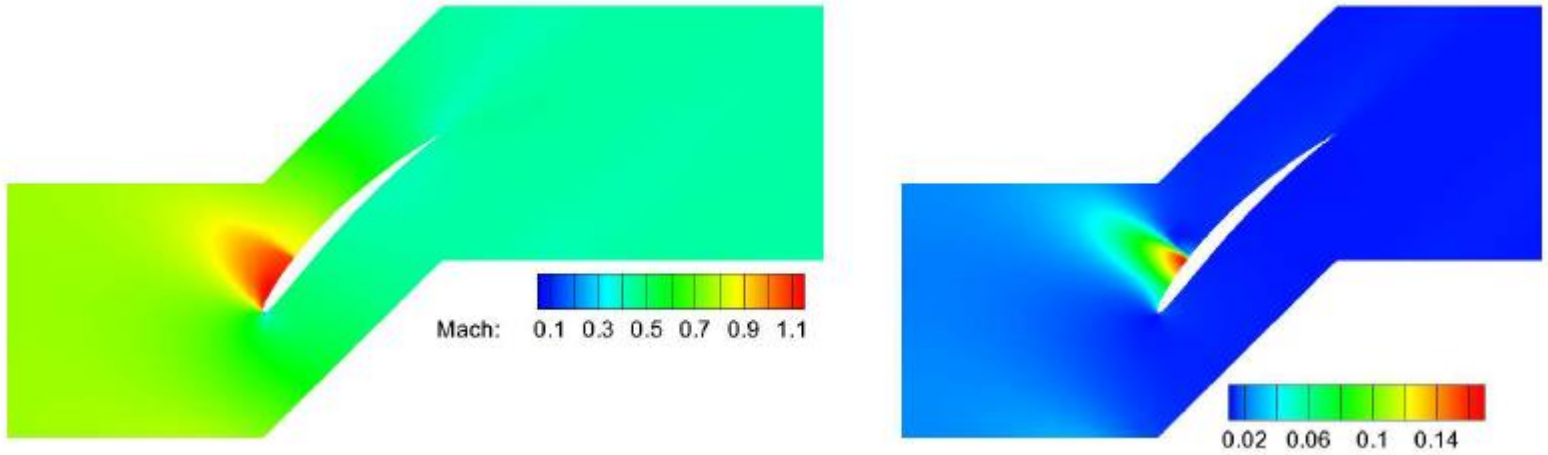


Comparison of intrusive (iPCE) and non-intrusive PCE (niPCE)

- **Inviscid flow in the SC10 compressor cascade ($\mathrm{M}_1 = 0.4425, \alpha_1 = 58^o$)**
- ***QoI J*: Static pressure rise (p_2/p_1)**
- **Uncertainty in flow conditions:**

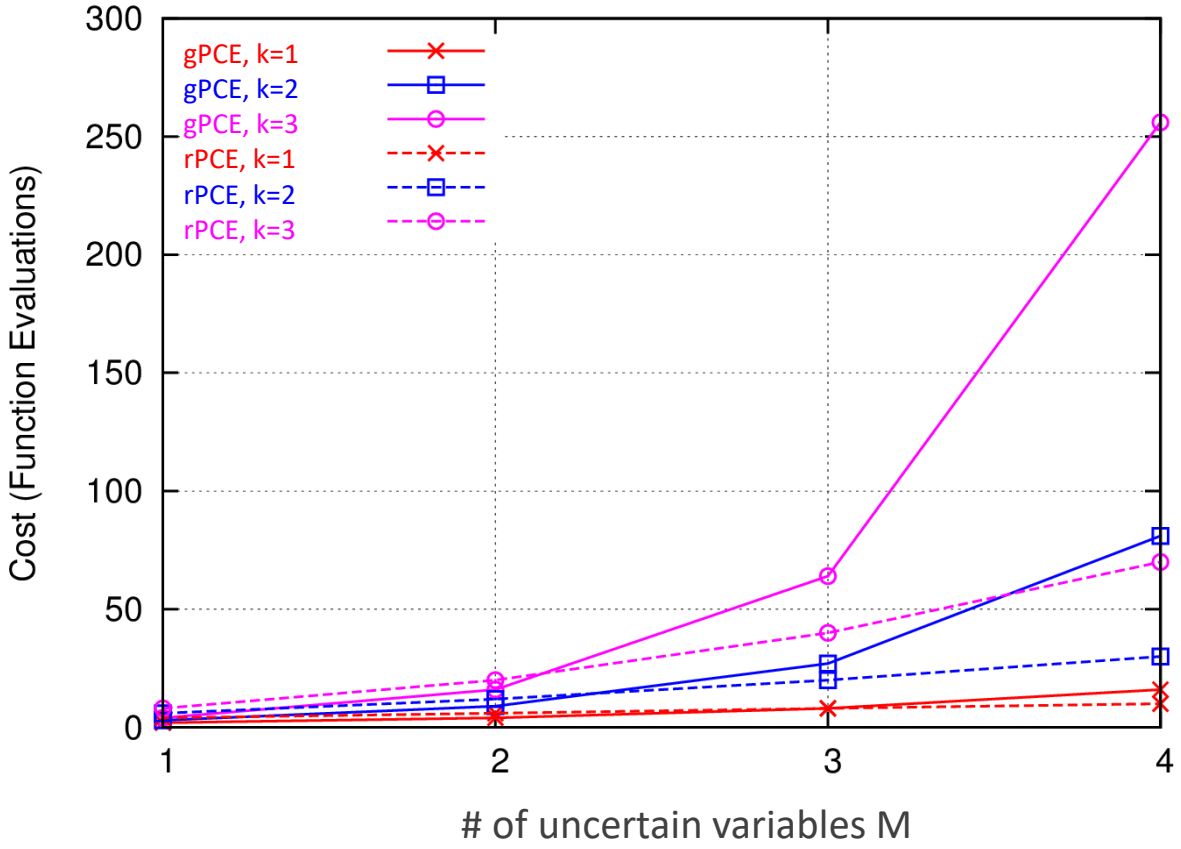
$\mathrm{M}_1 \sim \mathrm{N}(0.4425, 0.005)$
$\alpha_1 \sim U[57^o, 59^o]$

	iPCE Chaos order 1	niPCE	iPCE Chaos order 2	niPCE
μ	1.2933	1.2936	1.2938	1.2937
σ	0.0257	0.0254	0.0251	0.0253
TU	0.8	1	1	1



mean standard deviation

niPCE – Cost comparison between gPCE and rPCE



Regression (rPCE)

k \ M	1	2	3	4
1	4	6	8	10
2	6	12	20	30
3	8	20	40	70

Gauss Quadrature (gPCE)

k \ M	1	2	3	4
1	2	4	8	16
2	3	9	27	81
3	4	16	64	256

For M > 2, regression is cheaper than Gauss Quadrature
For chaos order k = 2*:*

- **gPCE Cost: 3^M**
- **rPCE Cost: $M^2 + 3M + 2$**

A cheaper (affordable) approach is needed for *M>>*

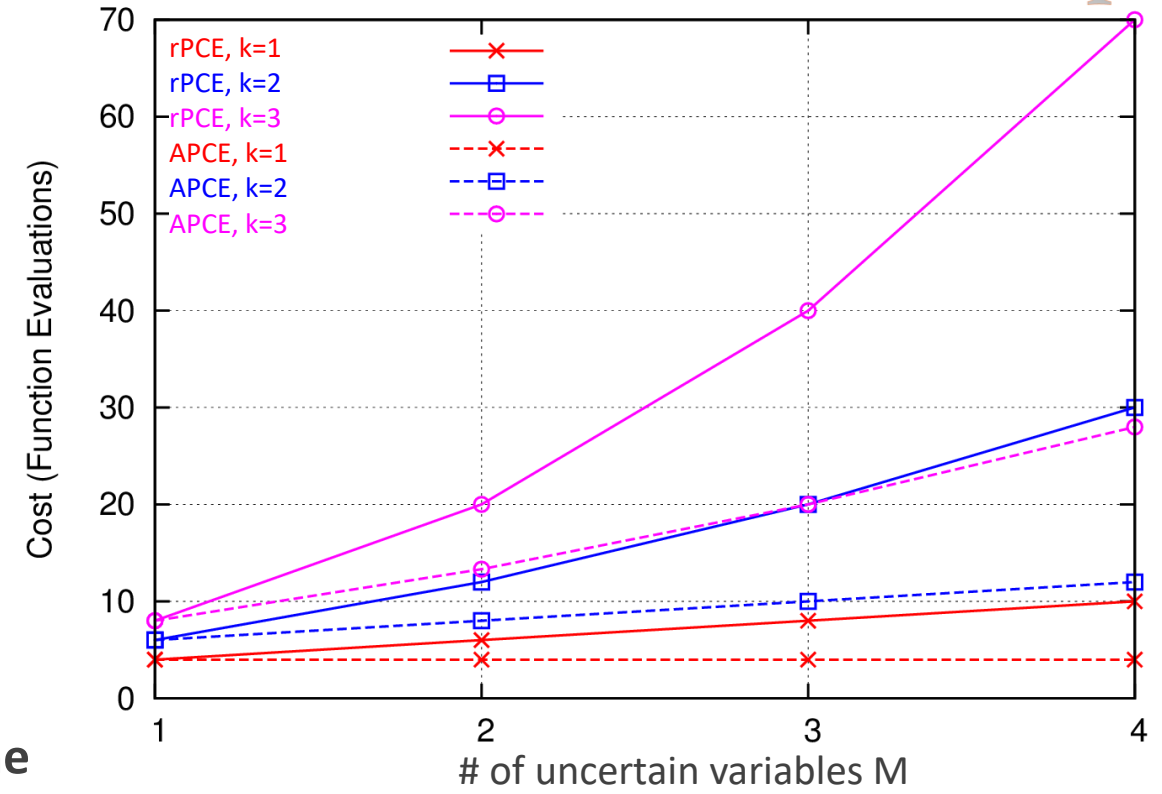
Prof. K.C. Giannakoglou, kgianna@mail.ntua.gr, Dr. V.G. Asouti, vasouti@mail.ntua.gr

Adjoint-assisted PCE (APCE)

Use the adjoint method to accelerate rPCE

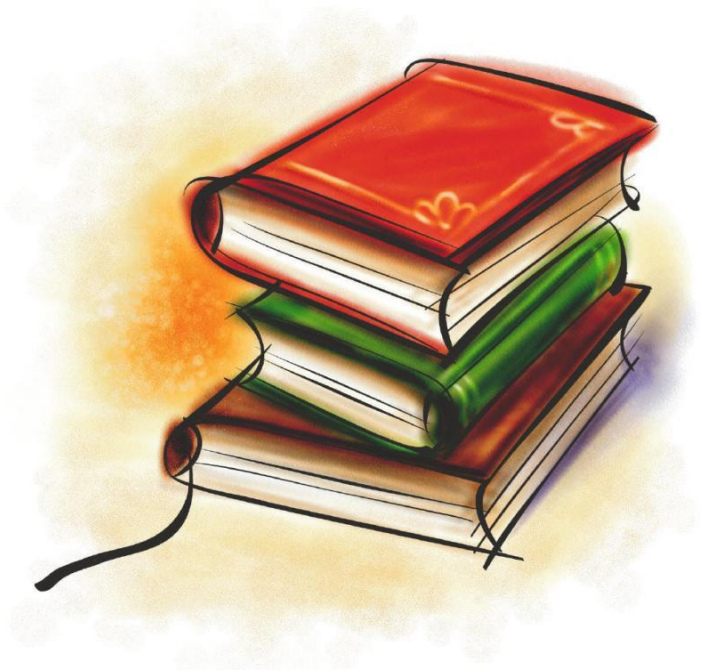
- The *M* components of $\frac{\delta J}{\delta c_m}$ can be computed at the cost of one TU.
- 2 TUs are enough to provide ***1+M*** data (at the same collocation point) for the regression.
- The **APCE** costs **2L** TUs (for L collocation points) to collect **(1+M)L** pieces of information (eqs.), for UQ.
- **Cost reduction by a factor of M, compared to standard rPCE.**

  - One adjoint solution per QoI.
  - Can become counter-productive if the number of the QoIs is greater that the uncertain variables (rarely the case).



For *k = 2*, (a typical case)

- gPCE Cost: 3^M
- rPCE Cost: $M^2 + 3M + 2$
- **APCE Cost: *2M + 4***

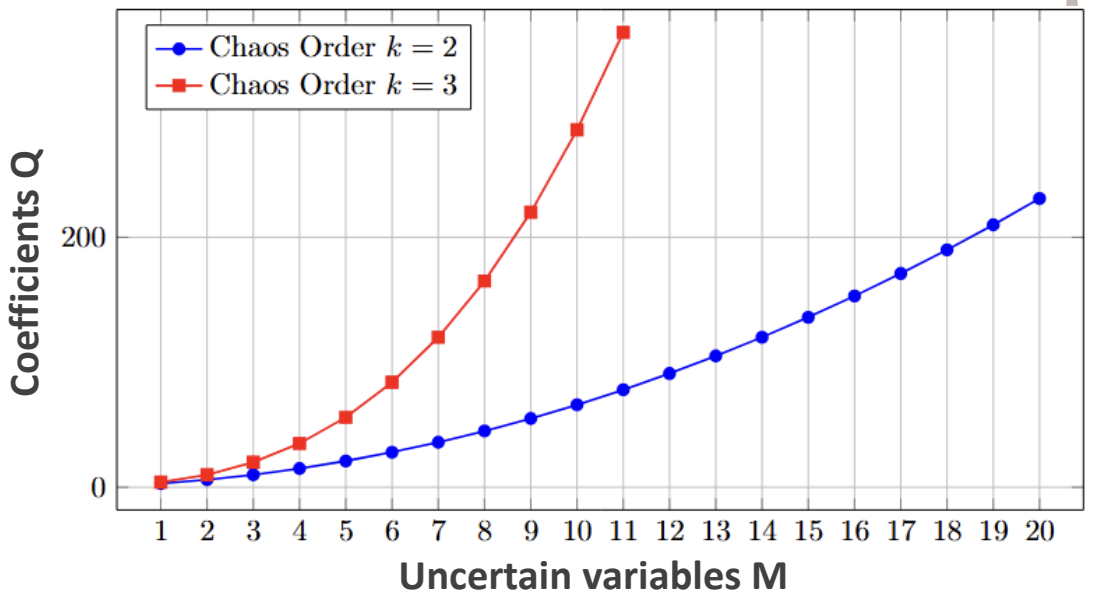


Sparse Regression Methods

Prof. K.C. Giannakoglou, kgianna@mail.ntua.gr, Dr. V.G. Asouti, vasouti@mail.ntua.gr

Why Sparse PCE ?

If M>>, the number Q of coefficients increases a lot.
→ Increased cost due the increased number of samples.
→ Many coefficients b_i are expected to be close to zero; so, only a small subset of basis functions really matters.



Get a sparse form of the OLS (Ordinary Least Squares) system:

$$\begin{bmatrix} H_0(\vec{c_0}) & H_1(\vec{c_0}) & \dots & H_{Q-1}(\vec{c_0}) \\ & \vdots & & \\ H_0(\vec{c}_{L-1}) & H_1(\vec{c}_{L-1}) & \dots & H_{Q-1}(\vec{c}_{L-1}) \end{bmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_{Q-1} \end{bmatrix} = \begin{bmatrix} J(\vec{c}_0) \\ \vdots \\ J(\vec{c}_{L-1}) \end{bmatrix}$$

How?

Orthogonal Matching Pursuit (OMP)

Candidate pool C: All available polynomials (full PCE up to order k)
Active matrix A: Selected polynomials (sparse basis)

$$C = \begin{bmatrix} H_0(\vec{c_0}) & H_1(\vec{c_0}) & \dots & H_{Q-1}(\vec{c_0}) \\ & & \vdots & \\ H_0(\vec{c}_{L-1}) & H_1(\vec{c}_{L-1}) & \dots & H_{Q-1}(\vec{c}_{L-1}) \end{bmatrix}$$

Set L=5M as a criterion to stop the algorithm
Initialize matrix A (use only the constant/zeroth-order polynomial terms, H_0)
<u>For i in 1,Q-1</u>

- **Compute the residual $\vec{r} = \vec{J} - A\,\vec{b}$**
- **Compute the correlation index $|\cos(H_i, r)| = \left|\frac{H_i^T r}{\|H_i\|\|r\|}\right|$**
- **Select the polynomial with maximum correlation and copy it to matrix *A***
- **Recompute *b* coefficients via OLS.**
- **Compute the leave-one-out error, ε_{LOO}.**
- **Terminate if:**

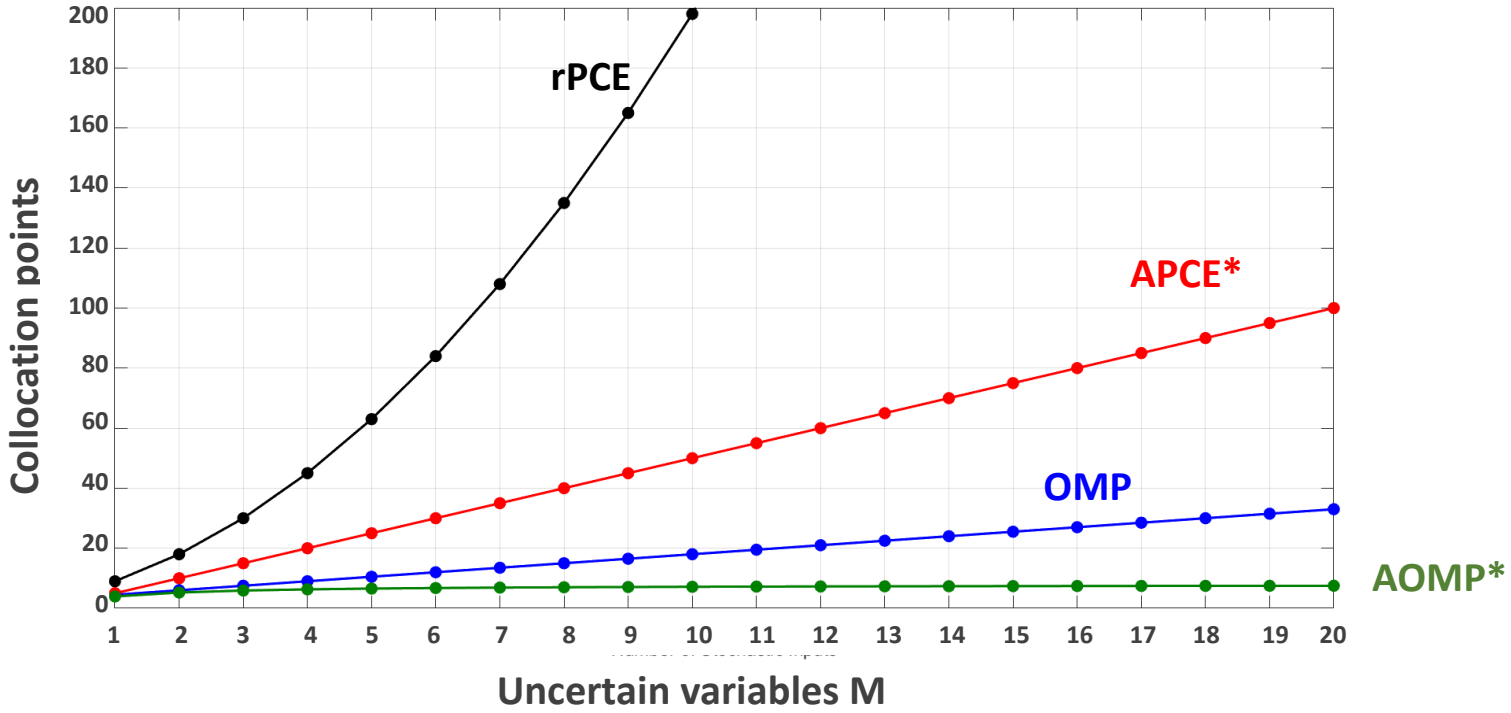
  ε_{LOO} increases more that 10% or max $|\cos(H_i, r)| <$ 1e-3

OMP

$$A = \begin{bmatrix} H_0(\vec{c_0}) & \dots & H_{Q_{sparse}-1}(\vec{c_0}) \\ & \vdots & \\ H_0(\vec{c}_{L-1}) & \dots & H_{Q_{sparse}-1}(\vec{c}_{L-1}) \end{bmatrix}$$

Adjoint – assisted OMP (AOMP)

The adjoint method can be used to accelerate OMP too!



Number of samples required for rPCE, APCE, OMP and AOMP
*** APCE & AOMP include the gradient**

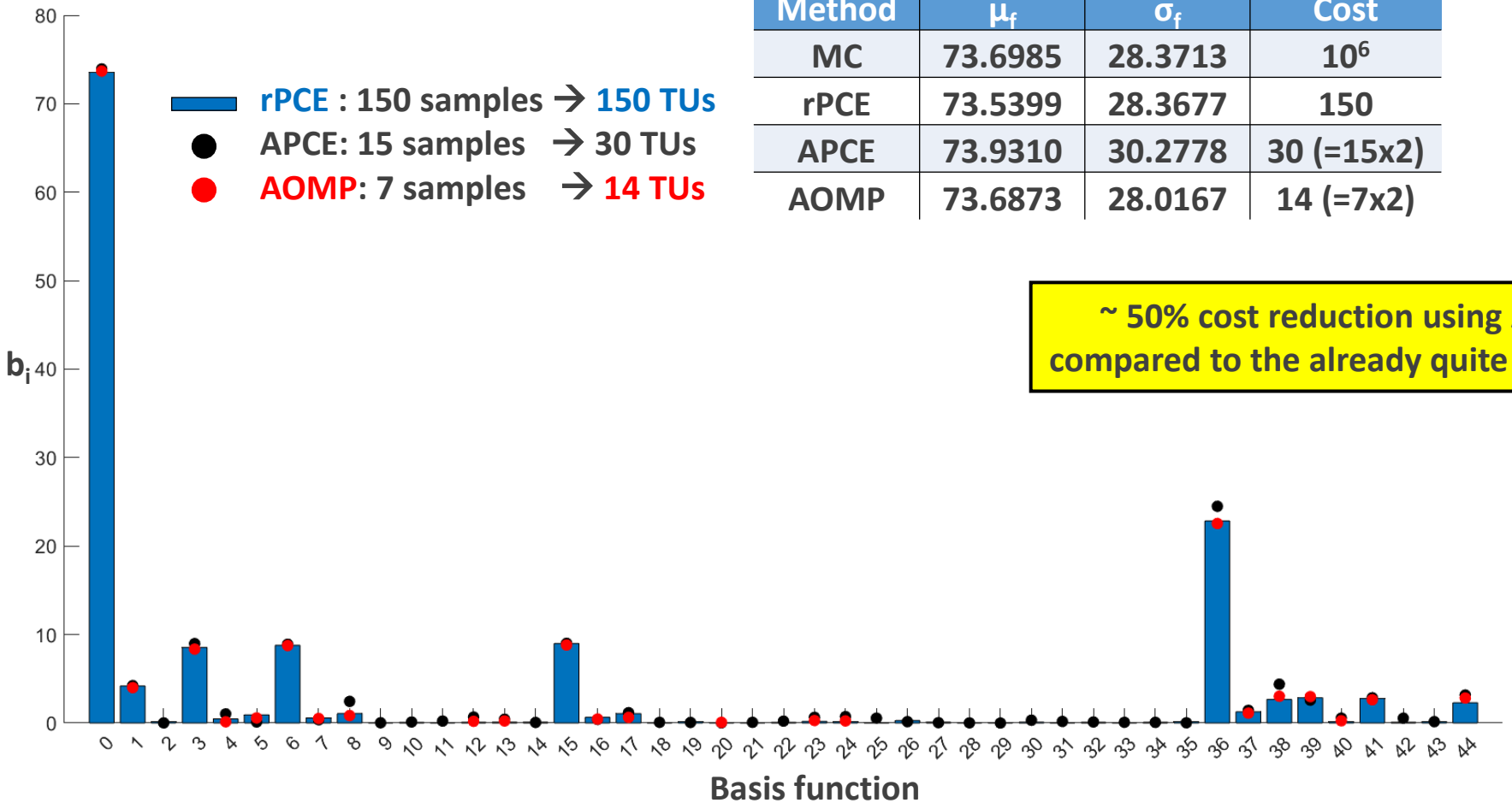
Borehole Function – Revisited (1/2)

M=8 uncertain variables
Chaos order k=2

⇨ **45 coefficients b_i should be computed**

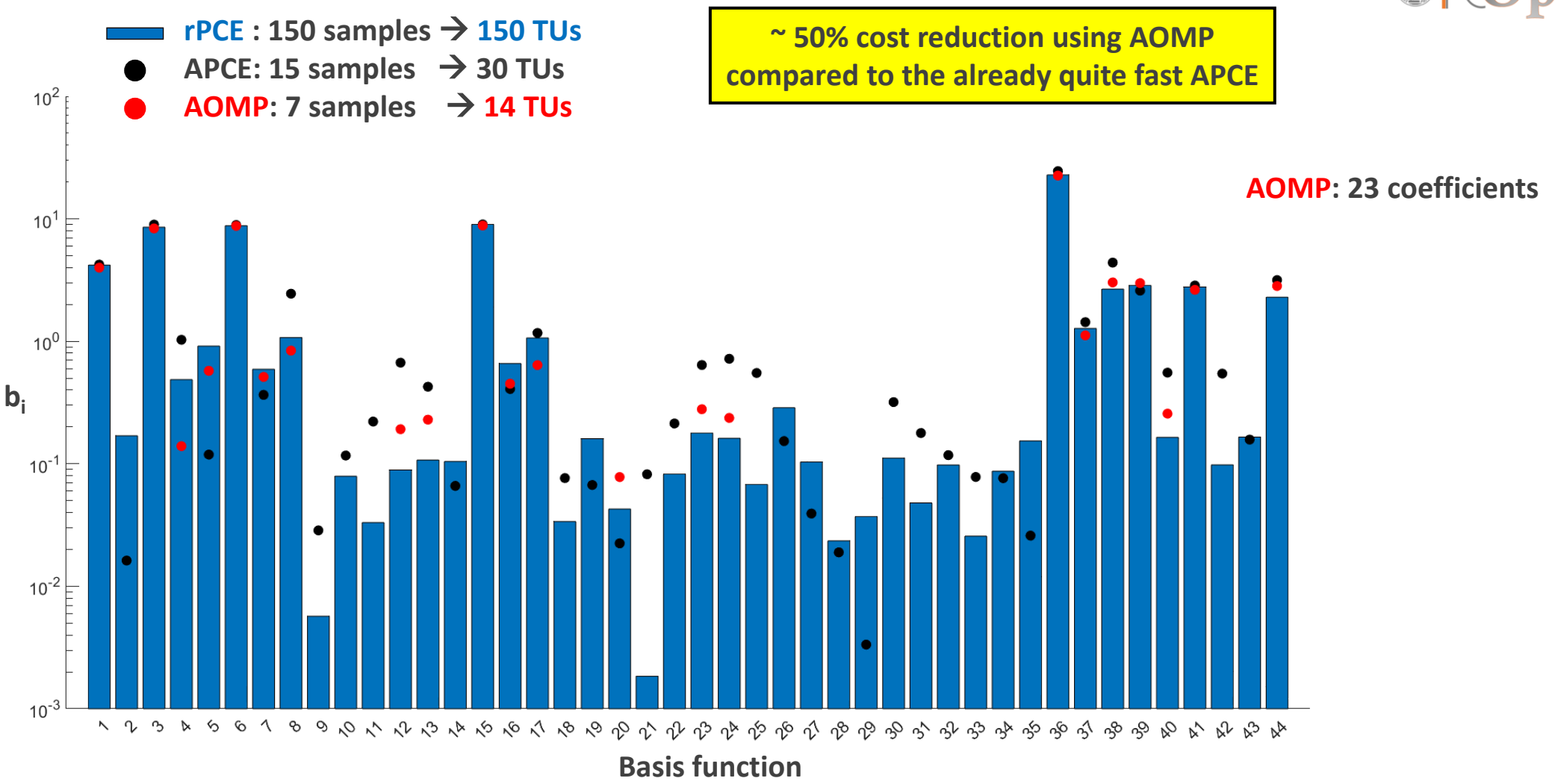
- **rPCE : 150 samples → 150 TUs**
- **APCE: 15 samples → 30 TUs**
- **AOMP: 7 samples → 14 TUs**

Method	μ_f	σ_f	Cost
MC	73.6985	28.3713	10^6
rPCE	73.5399	28.3677	150
APCE	73.9310	30.2778	30 (=15x2)
AOMP	73.6873	28.0167	14 (=7x2)

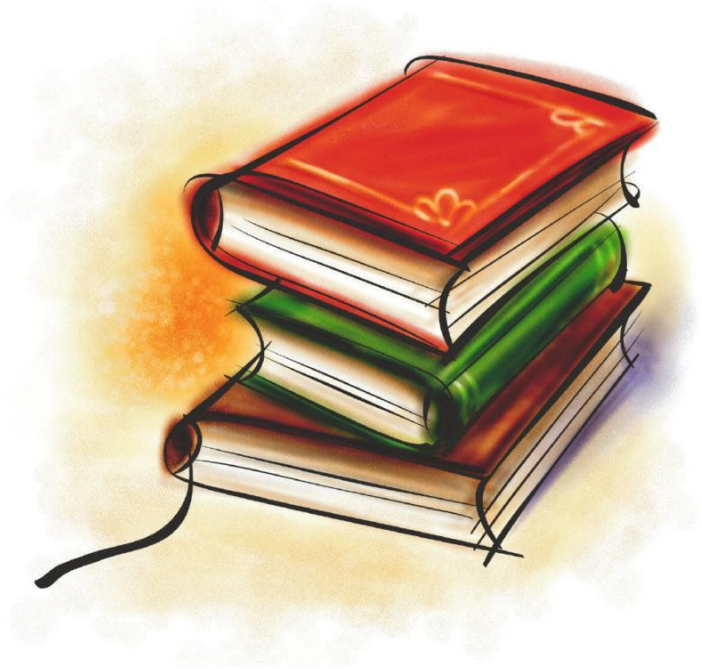


~ 50% cost reduction using AOMP compared to the already quite fast APCE

Borehole Function – Revisited (1/2)



Prof. K.C. Giannakoglou, kgianna@mail.ntua.gr, Dr. V.G. Asouti, vasouti@mail.ntua.gr



Method of Moments (MoM)

Prof. K.C. Giannakoglou, kgianna@mail.ntua.gr, Dr. V.G. Asouti, vasouti@mail.ntua.gr

Method of Moments (1/2)

- **Based on the Taylor expansion of *J* around the mean value of the uncertain variables:**

$$J(\overline{\boldsymbol{c}}+\Delta\boldsymbol{c}) = J|_{\overline{\boldsymbol{c}}} + \left.\frac{\delta J}{\delta c_i}\right|_{\overline{\boldsymbol{c}}} \Delta c_i + \frac{1}{2}\left.\frac{\delta^2 J}{\delta c_i \delta c_j}\right|_{\overline{\boldsymbol{c}}} \Delta c_i \Delta c_j + O(\Delta \boldsymbol{c}^3)$$

- **Order/accuracy of the method is determined by the number of terms in the Taylor expansion.**

- **Statistical moments of *J* obtained by substituting the Taylor expansion into their definitions:**

$$\mu_J(\boldsymbol{c}) = J|_{\overline{\boldsymbol{c}}} + \frac{1}{2}\sum_{i=1}^{M}\left[\frac{\delta^2 J}{\delta c_i^2}\right]_{\overline{\boldsymbol{c}}}\sigma_i^2, \qquad \sigma_J(\boldsymbol{c}) = \sqrt{\sum_{i=1}^{M}\left[\frac{\delta J}{\delta c_i}\right]_{\overline{\boldsymbol{c}}}^2\sigma_i^2 + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}\left[\frac{\delta^2 J}{\delta c_i \delta c_j}\right]_{\overline{\boldsymbol{c}}}^2\sigma_i^2\sigma_j^2}$$

- **Variants:**
  - **First-Order Second-Moment (FOSM)**: **retains only first-order derivatives**
  - **Second-Order Second-Moment (SOSM)**: **retains up to second-order derivatives**

- **Need to efficiently compute up to second-order derivatives** ⟹ ***The DD/AV method***

Method of Moments (1/2)

- For the **first-order derivatives:**
  - Computed using the adjoint method, computing all M components of $\delta J/\delta c_i$ by solving the adjoint PDEs only once, i.e. at the extra cost of one TU.

- For the **second-order derivatives (Hessian matrix):**
  - Computed using a combination of the adjoint method and direct differentiation, computing all M^2 components of $\delta^2 J/\delta c_i \delta c_j$ at the cost of $M+1$ TUs.

UQ Method	Order	Req. $\frac{\delta J}{\delta c_i}$	Req. $\frac{\delta^2 J}{\delta c_i \delta c_j}$	UQ cost (TUs)	
FOSM	1	☑	☒	2	Independent of M.
SOSM	2	☑	☑	$M+2$	Scales linearly with M

Computational Cost Comparison of Methods (1/2)

Method	Order	Sample points	Cost (EFS)
FOSM	1	1	$1 + N_J$
SOSM	2	1	$M + 1 + N_J$
APCE	k	$\left\lfloor \frac{r(M+k)!}{(M+1)!\,k!} \right\rfloor$	$(1+N_J)\left\lfloor \frac{r(M+k)!}{(M+1)!\,k!} \right\rfloor$
rPCE	k	$\frac{r(M+k)!}{M!\,k!}$	$\frac{r(M+k)!}{M!\,k!}$
gPCE	k	$(k+1)^M$	$(k+1)^M$

N_J : number of QoI
k : chaos order
M : number of uncertain variables
r : oversampling ratio

Methods Glossary
gPCE: GQ-based PCE
rPCE: regression-based PCE
APCE: Adjoint-assisted rPCE
FOSM: First-Order Second Moments
SOSM: Second-Order Second Moments

Sparse methods are excluded from this comparison!

Computational Cost Comparison of Methods (2/2)

Specialisation for
N_J = 1 QoI
k = 2
r = 2

Method	Order	Sample points	Cost (EFS)
FOSM	1	1	*2*
SOSM	2	1	*M + 2*
APCE	2	*M + 2*	*2M + 4*
rPCE	2	$M^2 + 3M + 2$	$M^2 + 3M + 2$
gPCE	2	3^M	3^M

M = 1	*M = 2*	*M = 3*	*M = 4*
2	*2*	*2*	*2*
3	*4*	*5*	*6*
6	*8*	*10*	*12*
6	*12*	*20*	*30*
3	*9*	*27*	*81*

Cheapest second-order method

Cheapest PCE variant

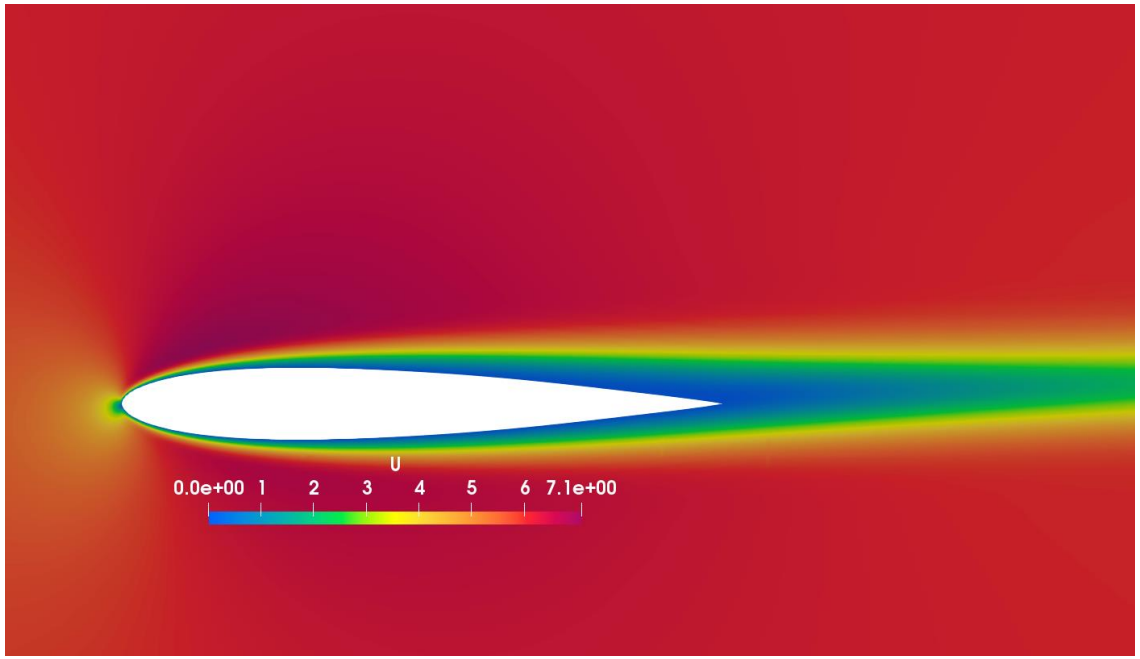
Methods Glossary
gPCE: GQ-based PCE
rPCE: regression-based PCE
APCE: Adjoint-assisted rPCE
FOSM: First-Order Second Moments
SOSM: Second-Order Second Moments

Sparse methods are excluded from this comparison!

Verification of UQ Methods (1/2)

- NACA0012 isolated airfoil at laminar flow; Re = 2000.
- *QoIs J*: drag and lift forces
- Flow, adjoint and direct differentiation equations solved using an in-house variant of OpenFOAM.
- M=3 uncertain variables, as in table

Uncertain Variable	μ	σ
Farfield velocity magnitude (m/s)	6	0.6
Farfield velocity angle (deg)	2	0.2
Kinematic viscosity (m2/s)	$3 \cdot 10^{-3}$	$1 \cdot 10^{-4}$



Verification of UQ Methods (2/2)

Method	Cost	Lift		Drag	
		μ_J (%Diff)	σ_J(%Diff)	μ_J (%Diff)	σ_J(%Diff)
FOSM	2	0.09538 (-0.36)	0.02071 (8.77)	0.08463 (-0.24)	0.01265 (1.2)
SOSM	5	0.09557 (-0.17)	0.02071 (8.77)	0.84723 (-0.13)	0.01265 (1.2)
APCE	10	0.09529 (-0.47)	0.02001 (5.08)	0.08462 (-0.25)	0.01262 (0.9)
rPCE	20	0.09584 (0.11)	0.01915 (0.61)	0.08463 (-0.23)	0.01261 (0.85)
gPCE	27	0.09575 (0.02)	0.01913 (0.5)	0.08462 (-0.25)	0.01263 (0.99)
Monte Carlo	1000	0.09573	0.01904	0.08483	0.01251

Verification with the results of Monte-Carlo

- μ_J error: < 0.5% for all UQ methods
- σ_J error: can be non-negligible, especially for the MoM-based methods
- SOSM improves μ_J over FOSM, but not necessarily σ_J
- All PCE variants: quite accurate for μ_J and σ_J
- gPCE: most expensive but also the least unpredictable (no ambiguity in choosing the sampling points, no derivatives) → can be used as the reference method when Monte-Carlo is too expensive to run

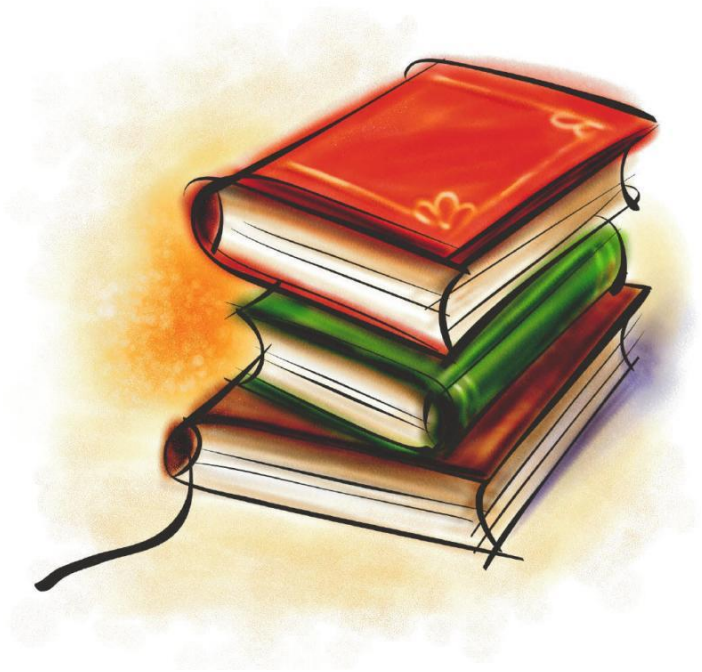
$$\mu_J(\boldsymbol{c}) = J|_{\overline{\boldsymbol{c}}} + \frac{1}{2}\sum_{i=1}^{M}\left[\frac{\delta^2 J}{\delta c_i^2}\right]_{\overline{\boldsymbol{c}}}\sigma_i^2$$

SOSM contribution: can be > 0 or < 0

$$\sigma_J(\boldsymbol{c}) = \sqrt{\sum_{i=1}^{M}\left[\frac{\delta J}{\delta c_i}\right]_{\overline{\boldsymbol{c}}}^2\sigma_i^2 + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}\left[\frac{\delta^2 J}{\delta c_i \delta c_j}\right]_{\overline{\boldsymbol{c}}}^2\sigma_i^2\sigma_j^2}$$

SOSM contribution: **always > 0**, so $\sigma_{SOSM} > \sigma_{FOSM}$

Selection of the most appropriate method should be based on (a) computational cost, (b) development cost and (c) accuracy.

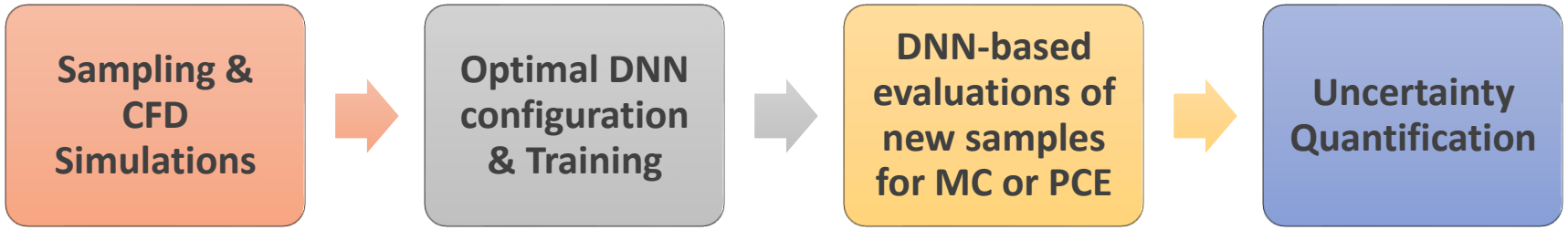


Deep Neural Networks to Support UQ

Prof. K.C. Giannakoglou, kgianna@mail.ntua.gr, Dr. V.G. Asouti, vasouti@mail.ntua.gr

Concept

Use a DNN, trained on a "reasonable" number of samples, to predict whatever is necessary for MC or PCE.



ONERA M6 Wing – Case Definition

- **ONERA M6 wing at transitional flow:**
  - **Re=3.5·10^6, M$_\infty$=0.262, AoA =0^o, Tu=0.20%.**

- **Use of the $\gamma - \widetilde{Re}_{\theta t}$ transition model coupled with the Spalart – Allmaras turbulence model. This is implemented in the PCOpt/NTUA GPU accelerated s/w PUMA.**

Skin friction coefficient over the wing suction side (left) and pressure side (right).

Flow Model: The PCOpt/NTUA GPU-Accelerated Software PUMA

RANS equations for compressible fluid flows with the Spalart-Allmaras turbulence model and the $\gamma - \tilde{Re}_{\theta t}$ transition model

$$\frac{\partial U_n}{\partial \tau} + \frac{\partial f_{nk}^{inv}}{\partial x_k} - \frac{\partial f_{nk}^{vis}}{\partial x_k} = 0$$

$$\frac{\partial(\rho\tilde{\nu})}{\partial \tau} + \frac{\partial(\rho\tilde{\nu}v_k)}{\partial x_k} - \frac{\rho}{\sigma}\left\{\frac{\partial}{\partial x_k}\left[(\nu+\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_k}\right] + c_{b_2}\frac{\partial\tilde{\nu}}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k}\right\} - \tilde{P}_{\tilde{\nu}} + D_{\tilde{\nu}} = 0$$

$$\tilde{P}_{\tilde{\nu}} = \gamma\rho c_{b_1}\tilde{S}\tilde{\nu}$$

$$\frac{\partial(\rho\gamma)}{\partial \tau} + \frac{\partial(\rho v_k\gamma)}{\partial x_k} - \frac{\partial}{\partial x_k}\left[\left(\mu+\frac{\mu_t}{\sigma_f}\right)\frac{\partial\gamma}{\partial x_k}\right] - P_\gamma + D_\gamma = 0$$

$$\frac{\partial(\rho\tilde{Re}_{\theta t})}{\partial \tau} + \frac{\partial(\rho v_k\tilde{Re}_{\theta t})}{\partial x_k} - \frac{\partial}{\partial x_k}\left[\sigma_{\theta,t}(\mu+\mu_t)\frac{\partial\tilde{Re}_{\theta t}}{\partial x_k}\right] - P_{\theta,t} - D_{SCF} = 0$$

$$P_\gamma = \rho c_{\alpha_1} F_{length} F_{onset}\left[\phi_{-300}\left(\Omega, \frac{M\sqrt{MRe}}{20}\right)\right]\sqrt{\gamma}(1 - c_{\epsilon_1}\gamma)$$

$$D_\gamma = \rho c_{\alpha_2} F_{turb}\left[\phi_{-300}\left(\Omega, \frac{M\sqrt{MRe}}{20}\right)\right]\gamma(c_{\epsilon_2}\gamma - 1)$$

$$P_{\theta,t} = \rho\frac{c_{\theta,t}}{T}\left(Re_{\theta,t}^{eq} - \tilde{Re}_{\theta t}\right)(1 - F_{\theta t})$$

$$D_{SCF} = c_{\theta,t}\frac{\rho}{T}c_{crossflow}\min\left(Re_{SCF} - \tilde{Re}_{\theta t}, 0\right)F_{\theta t}$$

h_{rms}

$$c_{\alpha_1} = 2 \qquad \sigma_f = 1$$
$$c_{\alpha_2} = 0.06 \qquad \sigma_{\theta,t} = 2$$
$$c_{\epsilon_1} = 1 \qquad c_{\theta,t} = 0.03$$
$$c_{\epsilon_2} = 50$$
$$c_{crossflow} = 0.6$$

ONERA M6 Wing – Case Definition

- ONERA M6 wing at transitional flow:
  - Re=$3.5 \cdot 10^6$, M_∞=0.262, AoA =0°, Tu=0.20%.
- Transition model constants (without uncertainties):
  - $c_{\alpha 1}$=2, $c_{\alpha 2}$=0.06, $c_{\epsilon 2}$=50, $c_{\theta,t}$=0.03, h_{rms}=5μm.

Skin friction coefficient over the wing suction side (left) and pressure side (right).

UQ problem

- Quantities of Interest (QoI): c_D, c_L.
- Five (M=5) uncertain variables, namely: $c_{\alpha 1}$, $c_{\alpha 2}$, $c_{\epsilon 2}$, $c_{\theta,t}$ and h_{rms}.
- All of them follow normal distributions as:

$$c_{\alpha_1} \sim \mathcal{N}(2.0, 0.2)$$

$$c_{\alpha_2} \sim \mathcal{N}(0.06, 0.006)$$

$$c_{\epsilon_2} \sim \mathcal{N}(50.0, 5.0)$$

$$c_{\theta,t} \sim \mathcal{N}(0.03, 0.003)$$

$$h_{rms} \sim \mathcal{N}(5 \cdot 10^{-6}, 1.6 \cdot 10^{-6})$$

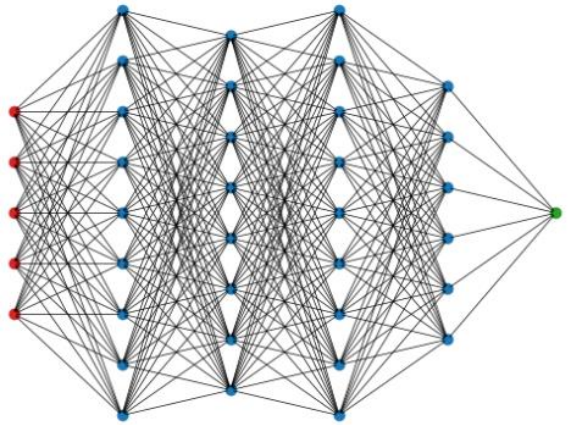
For chaos order k=2 → 21 (=Q+1) b_i coefficients.
243 =$(K+1)^M$=3^5 CFD runs (TUs) for gPCE.
Working assumption: 120 samples can only be afforded.

ONERA M6 Wing – Optimizing the DNN Configuration

- **Train two DNNs (for c_L and c_D):**
  - **120 samples on the uncertain space.**
- **Use a Metamodel-Assisted Evolutionary Algorithm (MAEA) to minimize the prediction error of a fully connected network.**

5 Inputs

($c_{\alpha 1}$,$c_{\alpha 2}$,$c_{\epsilon 2}$,$c_{\theta,t}$,h_{rms})



1 Output

(c_L or c_D)

? Number of layers

? Neurons per layer

? Activation function in each layer

ONERA M6 Wing – DNN Configurations & Error Metrics

Optimized DNN configurations

Quantity	Layers	Neurons	Activation functions
c_L	10	512,512,2048,2048,64,32,4096,1024,256	ReLU/sigmoid
c_D	4	32,1024,32,128	ReLU/tanh

DNN error metrics

Quantity	c_L	c_D
Mean Absolute % Error	0.028347	0.039051
Mean Absolute Error	0.001230	0.011312

ONERA M6 Wing – Case Definition

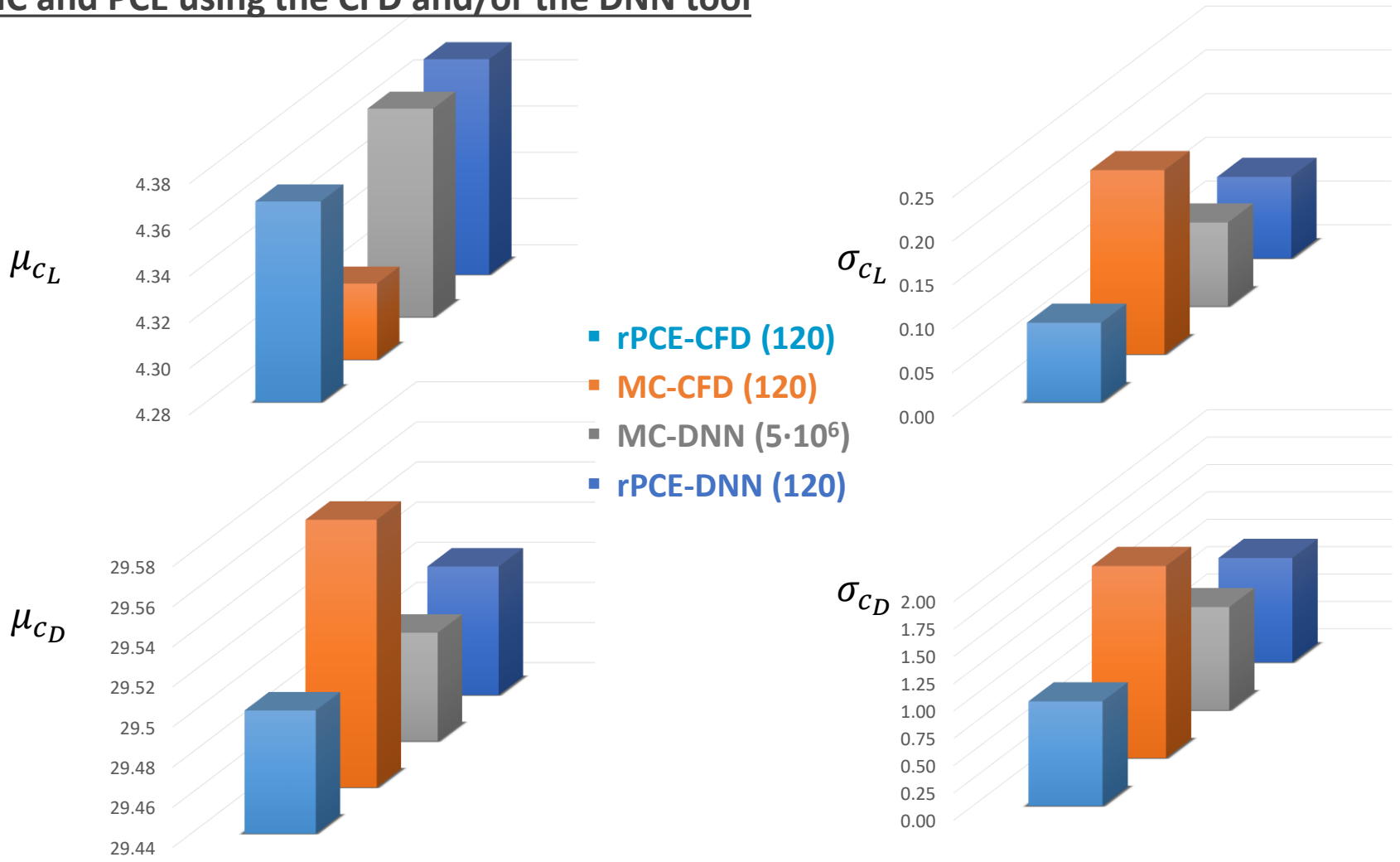
Comparison of MC and PCE using either the CFD and/or the DNN tool

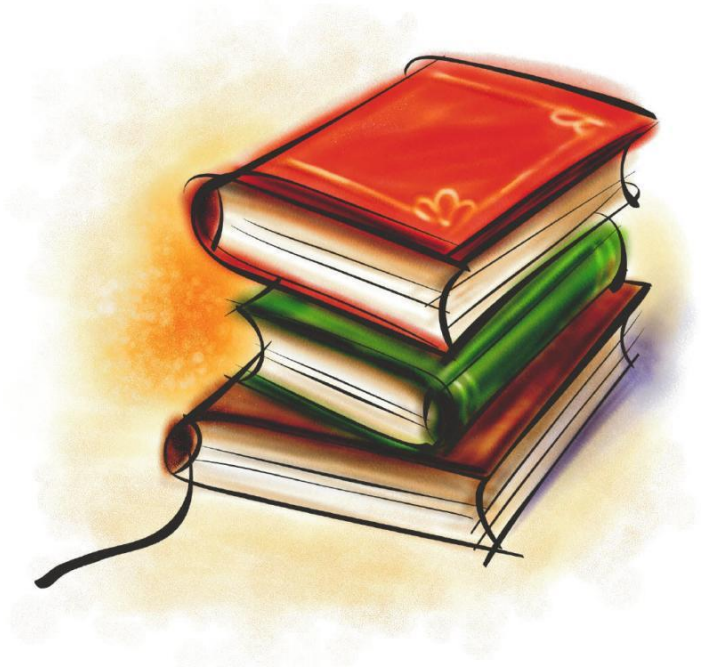
UQ Method	Using	Samples	c_L		c_D	
			μ	σ	μ	σ
rPCE	CFD	120	4.366775	0.090594	29.50131	0.956043
MC	CFD	120	4.312858	0.210206	29.57335	1.755292
MC	DNN	5×10^6	4.370152	0.095526	29.49404	0.941851
rPCE	DNN	120	4.373161	0.092948	29.50401	0.952404

DNN can safely be used to support UQ either with MC or rPCE by thus reducing the computational cost.

ONERA M6 Wing – Case Definition

Comparison of MC and PCE using the CFD and/or the DNN tool

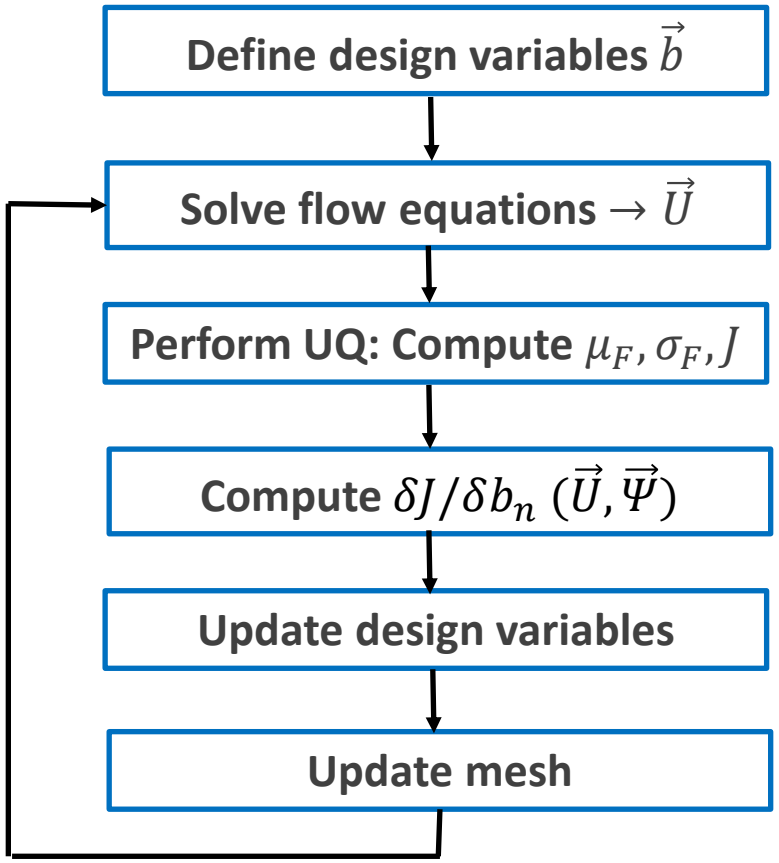




Optimisation under Uncertainties

Prof. K.C. Giannakoglou, kgianna@mail.ntua.gr, Dr. V.G. Asouti, vasouti@mail.ntua.gr

ShpO under Uncertainties using a GBM, in Fluid Mechanics



F = QoI
J = Objective function

⇨ **Depending on the UQ method, potentially several times**

⇨ **Using any of the previously described UQ methods**

⇨ **Using the adjoint method**

⇨ **Steepest descent, CG, BFGS, SQP.**

Projected FOSM (pFOSM) Method

- **Objective function:** $J = \mu_F + w\sigma_F$

F = QoI
J = Objective function

- **Statistical moments in FOSM:**

$$\mu_F^{(FOSM)} = F(\boldsymbol{\mu}) \quad \text{and} \quad \sigma_F^{(FOSM)} = \sqrt{\frac{\delta F}{\delta c_m}\bigg|_{\boldsymbol{\mu}} \frac{\delta F}{\delta c_j}\bigg|_{\boldsymbol{\mu}} \mathcal{K}_{mj}}$$

$\mathcal{K}_{mj}$: Covariance matrix

- **For gradient-based optimisation (N design variables, M stochastic variables):**

$$\frac{\delta J}{\delta b_n} = \frac{\delta F}{\delta b_n} + \frac{w}{\sigma_F^{(FOSM)}} z_m \frac{\delta^2 F}{\delta c_m \delta b_n} \quad , \quad z_m^{(FOSM)} = \mathcal{K}_{mj} \frac{\delta F}{\delta c_j}\bigg|_{\boldsymbol{\mu}}$$

: projection vector

- **The best method to compute these mixed derivatives has a cost that scales with min(N,M).**
- **Main concept in pFOSM: Compute projections of the mixed derivatives matrix onto z_m , instead of the second derivatives themselves!**

Projected FOSM (pFOSM) Method

$$z_m \frac{\delta^2 F}{\delta c_m \delta b_n} = \ldots + \int_\Omega \left(\ldots + z_m \frac{\delta q}{\delta c_m} \frac{\partial v_j}{\partial x_k} + \ldots \right) \frac{\partial}{\partial x_j} \left(\frac{\delta x_k}{\delta b_n} \right) d\Omega + \ldots$$

Let $\widehat{(\ldots)}$ denote any quantity $\widehat{\phi} = z_m \frac{\delta \phi}{\delta c_m}$

$v_i, p, \tilde{\nu}$: **primal variables**

$u_i, q, \tilde{\nu}_\alpha$: **adjoint variables**

$\widehat{v}_i, \widehat{p}, \widehat{\tilde{\nu}}$: **Projected derivatives of primal variables**

$\widehat{u}_i, \widehat{q}, \widehat{\tilde{\nu}}_\alpha$: **Projected derivatives of adjoint variables**

pFOSM requires the solution of PDEs for all $\widehat{(\ldots)}$ primal/adjoint fields

Projected APCE (pAPCE) Method

- **Objective function:**

$$J = \mu_F + w\sigma_F$$

- **Gradient of J:**

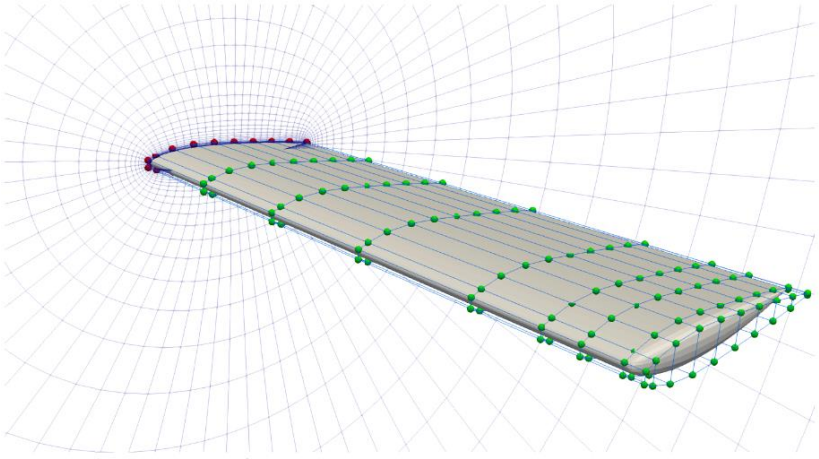
$$\frac{\delta J}{\delta b_n} = \sum_{l=1}^{L} k^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \sum_{l=1}^{L} z_m^{(l)} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)})$$

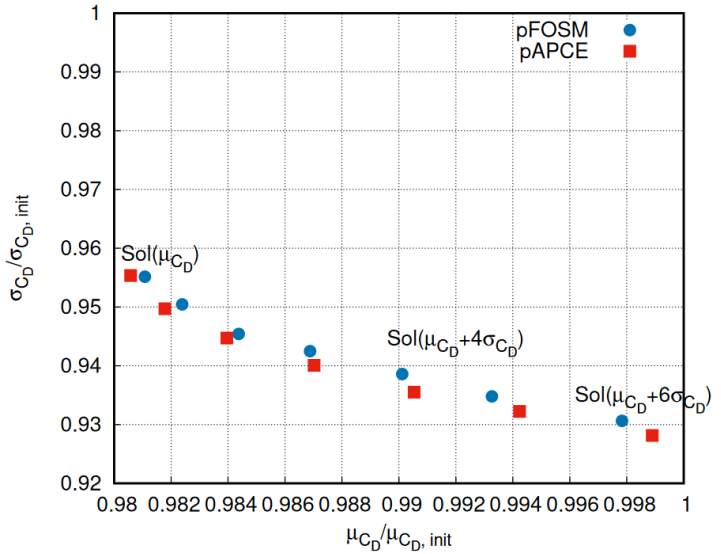
$k^{(l)}$, $z_m^{(l)}$ are known quantities.

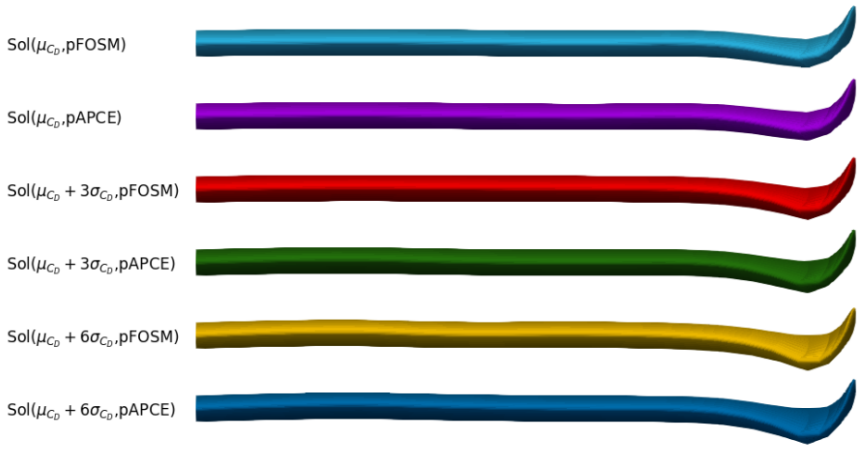
As in the pFOSM, pAPCE computes mixed derivatives at the cost of 4 EFS per collocation point (total cost 4L EFS). Doesn't scale with M as APCE does!

The pAPCE is more costly than the pFOSM but, also, more accurate.

ShpO of a Wing with pFOSM & pAPCE



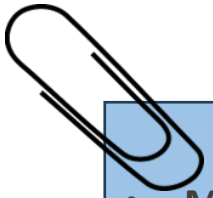




Running each method 7 times for min. $min(\mu_{C_D} + w\sigma_{C_D})$ for w=0,1,2,…,6. Constrained case, aiming at constant μ_{C_L}. ShpO with pAPCE: O_{PCE}=3, L=10. Uncertainties related to the flow conditions:

$$\mu_{v_\infty} = 50m/s,\ \sigma_{v_\infty} = 10m/s \qquad \mu_{\alpha_\infty} = 5^\circ,\ \sigma_{\alpha_\infty} = 1^\circ$$

Read more in:



- M. Chatzimanolakis, K-D. Kantarakias, V.G. Asouti and K.C. Giannakoglou, "A Painless Intrusive Polynomial Chaos Method with RANS-based Applications", Computer Methods in Applied Mechanics and Engineering 2019; 348:207-221.
- K. Fragkos, E.M. Papoutsis-Kiachagias and K.C. Giannakoglou, "pFOSM: An Efficient algorithm for aerodynamic robust design based on continuous adjoint and Matrix-Vector products", Computers & Fluids 2019; 181:57-66.
- Th. Skamagkis, E.M. Papoutsis-Kiachagias, K. Giannakoglou, "CFD-based shape optimization under uncertainties using the Adjoint-assisted Polynomial Chaos Expansion and projected derivatives", Computers & Fluids 2022; 241:105458
- A.K. Papageorgiou, E.M. Papoutsis-Kiachagias, K. Giannakoglou, "Development and Assessment of an iPCE-based Continuous Adjoint Method for Shape Optimization under Uncertainties", International Journal for Numerical Methods in Fluids 2021; 94:59-75