Imperial College London

IMACS seminar on MCM

28/08/2011 – 02/09/2011

Metamodelling with a dependent input vector: comparison of two approaches.

Miguel Munoz Zuniga

Sergei Kucherenko

Nilay Shah

Outline

1. ***Independent Uniform inputs***
   - *ANOVA decomposition*
   - *HDMR*
   - *Polynomial approximation of HDMR*

2. ***Arbitrary independent inputs***
   - *Steps of the method*
   - *An example*
   - *General principle*

3. ***Dependent inputs***
   - *First method: Polynomial (chaos) expansion with transformation step*
   - *Second method: Direct Polynomial (chaos) expansion*

4. ***Numerical results***

ANOVA decomposition

- *For any d-dimensional vector of input variables defined on* $H^d = [0,1]^d$ *the following ANOVA decomposition can be achieved.*

$$y = f(x) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_i \sum_{j>i} f_{ij}(x_i, x_j) + \ldots + f_{1,2,\ldots,d}(x_1, x_2, \ldots, x_d),$$

(Example: a random vector uniformly distributed on the unit (hyper)cube)

- *Property of the ANOVA decomposition functions:*

$$\int_0^1 f_{i_1 \cdots i_s}(x_{i_1}, , \ldots, x_{i_s}) \quad dx_{i_k} = 0, \ \forall k, \ 1 \le k \le s$$

Imperial College London

ANOVA decomposition: a first metamodel (hdmr)

- *For many practical problems only low order terms in the ANOVA decomposition are important.*

- *Proposed metamodel: High Dimetional Model Rrepresentation (Rabitz et al):*

$$f(x) \approx h(x) = f_0 + \sum_{i=1}^{n} f_i\left(x_i\right) + \sum_{i} \sum_{j>i} f_{ij}\left(x_i, x_j\right)$$

Imperial College London

HDMR: polynomial approximation

$$f(x) \approx h(x) = f_0 + \sum_{i=1}^{n} f_i(x_i) + \sum_i \sum_{j>i} f_{ij}(x_i, x_j)$$

- *Orthonormal polynomial expansion:*

$$f_i(x_i) \approx \sum_{r=1}^{k} \alpha_r^i \phi_r(x_i), \qquad f_{ij}(x_i, x_j) \approx \sum_{p=1}^{l} \sum_{q=1}^{l'} \beta_{pq}^{ij} \phi_{pq}(x_i, x_j)$$

- *Orthonormal polynomial coefficients:*

$$\alpha_r^i = \int_{H^d} f(\vec{x}) \phi_r(x_i) d\vec{x} \qquad \beta_{pq}^{ij} = \int_{H^d} f(\vec{x}) \phi_p(x_i) \phi_q(x_j) d\vec{x}$$

- *Orthonormal polynomial properties:*

$$\int_0^1 \phi_r(x_i) dx_i = 0 \quad \forall r, i$$

$$\int_0^1 \phi_r^2(x_i) dx_i = 1 \quad \forall r, i$$

$$\int_0^1 \phi_p(x_i) \phi_q(x_i) dx_i = 0 \quad p \neq q$$

First few Legendre Polynomials:

$$\phi_1(x) = \sqrt{3}(2x - 1)$$

$$\phi_2(x) = 6\sqrt{5}(x^2 - x + \frac{1}{6})$$

...

General Independent Input Case

Two step :

- *Transform each random component of X into a same chosen random variable*
- *Decompose f(X) using suitable orthonormal polynomial basis*

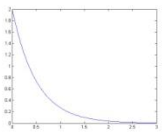
Random variable distribution	Orthogonal polynomial familly	Support
Gaussian	Hermite	$(-\infty, +\infty)$
Gamma	Laguerre	$[0, +\infty)$
Beta	Jacobi	$[a, b]$
Uniform	Legendre	$[a, b]$
Poisson	Charlier	$\{0, 1, 2, \ldots\}$
Binomial	Krawtchouk	$\{0, 1, 2, \ldots, N\}$

Imperial College London

Independent Input Case. (Nataf) transformation step - example

Inputs → *Model* → *Output(s)*

Exponential



f

$$Y = f\left(X_1\right) = X_1$$

$$= -2\ln\left(1 - U_1\right)$$

X_1

$$T^{-1}: \; X_1 = -2\ln\left(1 - U_1\right)$$

$$T: \; U_1 = 1 - \exp\left(\frac{-X_1}{2}\right)$$

U_1

Model

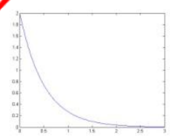
(Normalized)

$f \circ T^{-1}$

Independent Input Case. (Nataf) transformation step - example

Inputs → *Model* → *Output(s)*

Exponential



f

$$Y = f\left(X_1\right) = X_1$$

$$= -2\ln\left(1 - U_1\right)$$

X_1

$T^{-1}: \quad X_1 = -2\ln\left(1 - U_1\right)$

$T: \quad U_1 = 1 - \exp\left(\frac{-X_1}{2}\right)$

U_1

Model

(Normalized)

$f \circ T^{-1}$

Independent Input Case. PC expansion - example

Inputs

$$U_1$$

Model (Normalized)

$$f \circ T^{-1}$$

Output(s)

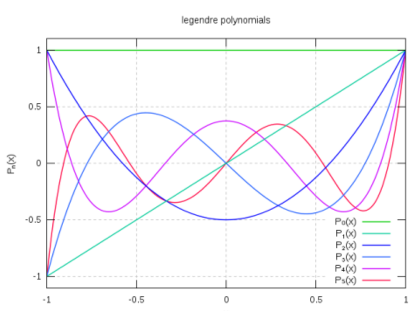
$$Y = f \circ T^{-1}(U_1) = -2\ln\left(1 - U_1\right)$$

$$= \sum_{i=1}^{+\infty} a_i \phi_i(U_1)$$

Weighted superposition of polynomials

$$\approx$$

$$\phi_i(u_1)$$



Legendre polynomials

Metamodel

$$(\hat{a}, \phi)$$

$$\hat{Y} = \sum_{i=1}^{M} \hat{a}_i \phi_i(U_1)$$

$$\hat{a}_i = \frac{1}{N} \sum_{j=1}^{N} f \circ T^{-1}(U_1^j)\phi_i(U_1^j)$$

Weight estimation by MC

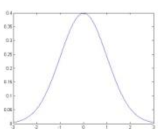
Independent Input Case. (Nataf) transformation step

Inputs → Model → Output(s)

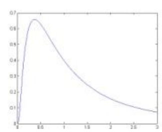
$$\left(X_1,\ldots,X_d\right)$$

$$f$$

$$Y = f\left(X_1,\ldots,X_d\right) = f\left[T^{-1}\left(U_1,\ldots,U_d\right)\right]$$



Any Independent Random Inputs



T^{-1} T

Uniform Independent Random Inputs

$$\left(U_1,\ldots,U_d\right)$$

Model (Normalized)

$$f \circ T^{-1}$$

Imperial College London

Independent Input Case. Polynomial chaos expansion

Inputs

$$(U_1,\ldots,U_d)$$

Uniform
Independent
Random Inputs

Model
(Normalized)

$$f \circ T^{-1}$$

Output(s)

$$Y = f\left[T^{-1}(U_1,\ldots,U_d)\right] = \sum_{i=1}^{+\infty} a_i \phi_i (U_1,\ldots,U_d)$$

$$\approx$$

$$\hat{Y} = \sum_{i=1}^{M} \hat{a}_i \phi_i (U_1,\ldots,U_d)$$

Metamodel

$$(\hat{a},\phi)$$

Legendre
polynomials

$$\hat{a}_i = \frac{1}{N}\sum_{j=1}^{N} f \circ T^{-1}(U_1^j,\ldots,U_d^j)\phi_i(U_1^j,\ldots,U_d^j)$$

Weight estimation by MC

Dependent Input Case.

Problem:

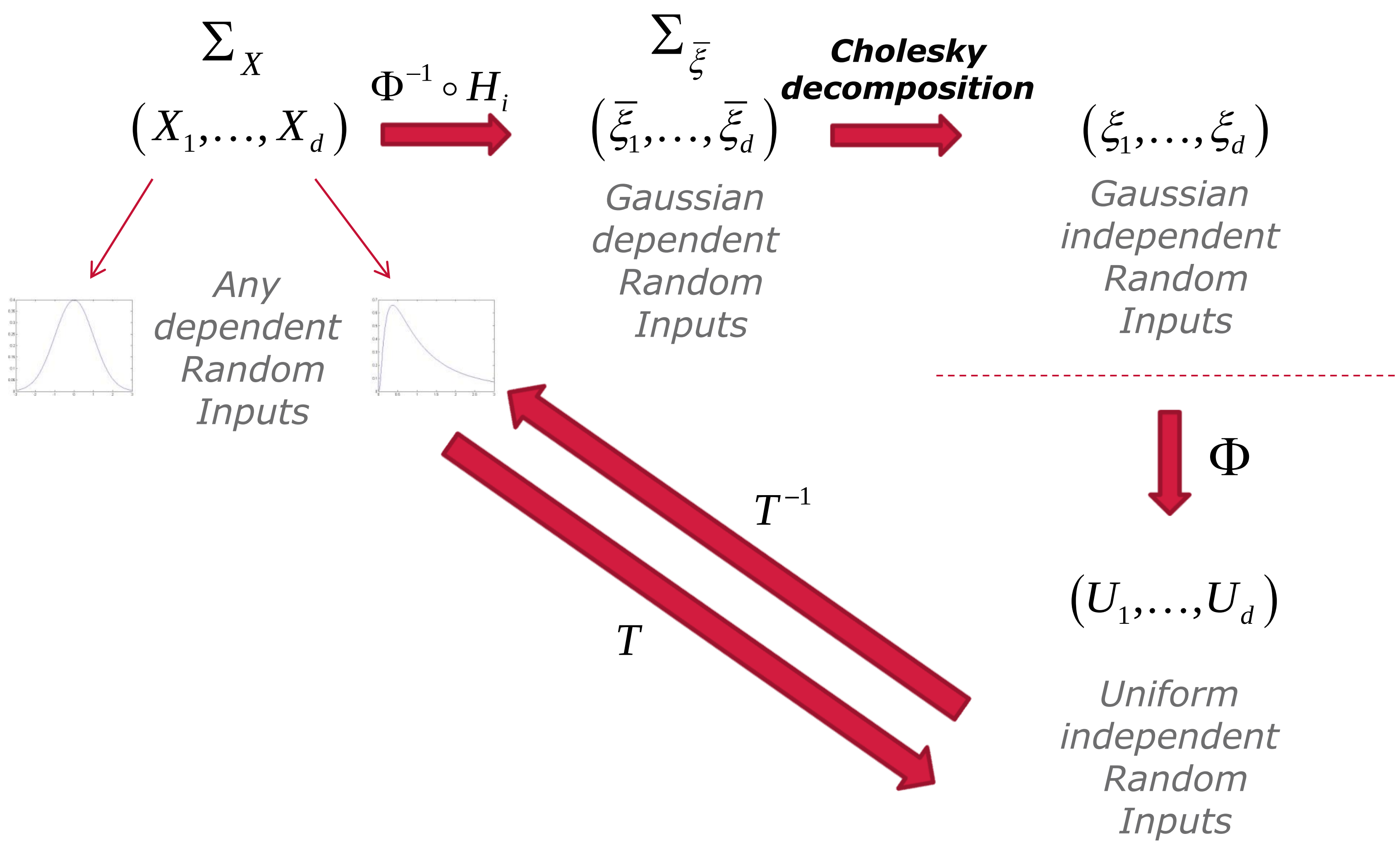
ANOVA decomposition is unique only if the inputs are independent
(The orthogonality of the standard polynomials basis is verified only if the inputs are independent).

Possible Solutions:

A. Make the inputs independent and use the methodology for an independent case.

B. Construct an orthonormal basis directly without the «independent step».

First method for the dependent Input Case: transformation to independent

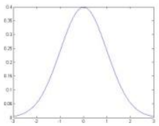


First method for the dependent Input Case: transformation to independent

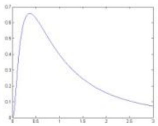
Inputs → Model → Output(s)

$$\left(X_1,\ldots,X_d\right) \qquad f \qquad Y = f\left(X_1,\ldots,X_d\right)$$

$$= f\left[T^{-1}\left(U_1,\ldots,U_d\right)\right]$$



Any dependent Random Inputs



T^{-1} T

Uniform Independent Random Inputs

$$\left(U_1,\ldots,U_d\right)$$

Model (independent)

$$f \circ T^{-1}$$

Imperial College London

First method for the dependent Input Case: Polynomial chaos expansion

Inputs → Model (independent) → Output(s)

$(U_1, \ldots, U_d)$

Uniform Independent Random Inputs

$f \circ T^{-1}$

$$Y = f\left[T^{-1}(U_1, \ldots, U_d)\right] = \sum_{i=1}^{+\infty} a_i \phi_i (U_1, \ldots, U_d)$$

$$\approx$$

Metamodel

$(\hat{a}, \phi)$

$$\hat{Y} = \sum_{i=1}^{M} \hat{a}_i \phi_i (U_1, \ldots, U_d)$$

Legendre polynomials

$$\hat{a}_i = \frac{1}{N} \sum_{j=1}^{N} f \circ T^{-1}(U_1^j, \ldots, U_d^j) \phi_i (U_1^j, \ldots, U_d^j)$$

Weight estimation by QMC

Second method for the dependent Input Case: direct polynomial chaos expansion

Model:

$$Y = f\left(X_1, \ldots, X_d\right)$$

Orthonormal Basis:

$$\Psi_i(X_1,\ldots,X_d) = \left(\frac{\overbrace{p_{X_1}\left(X_1\right)\ldots p_{Xd}\left(X_d\right)}^{\text{Marginal pdf}}}{\underbrace{p_X\left(X\right)}_{\text{joint pdf}}}\right)^{1/2} \underbrace{\Phi_i\left(X_1,\ldots,X_d\right)}_{\text{Standard orthogonal polynomials}}$$

Exact polynomial Expansion:

$$Y = \sum_{i=1}^{+\infty} a_i \Psi_i(X_1,\ldots,X_d)$$

Metamodel:

$$\hat{Y} = \sum_{i=1}^{M} \hat{a}_i \Psi_i(X_1,\ldots,X_d)$$

Weight estimation by MC:

$$\hat{a}_i = \frac{1}{n}\sum_{j=1}^{n} f(X_1^j,\ldots,X_d^j)\Psi_i(X_1^j,\ldots,X_d^j)$$

Dependent Input Case. Test example (Hyperplane)

- *Model*

Inputs are Gaussians

$$Y = f\left(X_1, X_2, X_3\right) = X_1 + X_2 + X_3$$

- *Correlation matrix*

$$\Sigma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \rho \\ 0 & \rho & 1 \end{pmatrix}$$

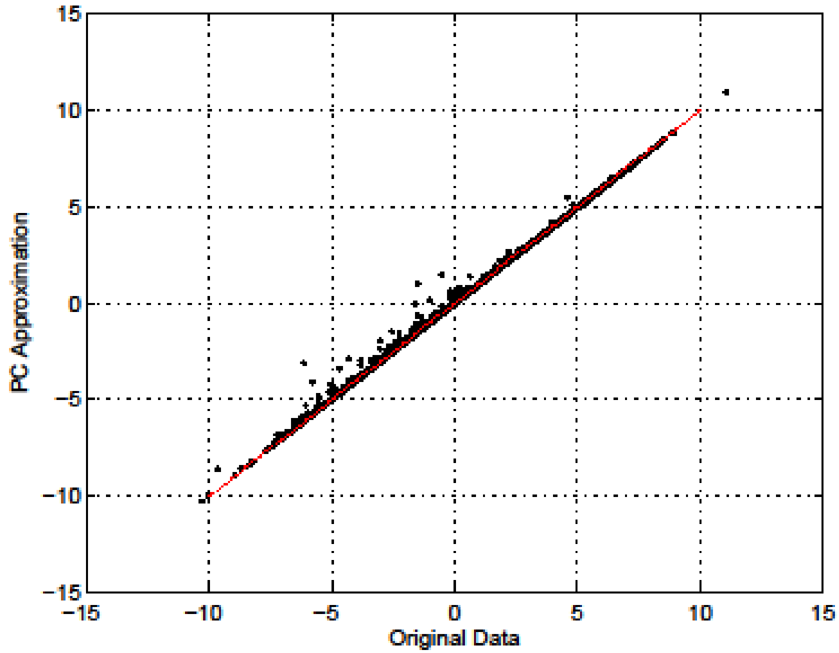
- *Error measurement:*

$$\delta(f, h) = \frac{1}{\sigma} \int \left[f(x) - h(x) \right]^2 dx$$

First Method. Test example (Hyperplane)

Only first order terms in the ANOVA polynomial expansion $\longrightarrow$ *[1; 1; 1]*

Number of QMC sampling N = 1024

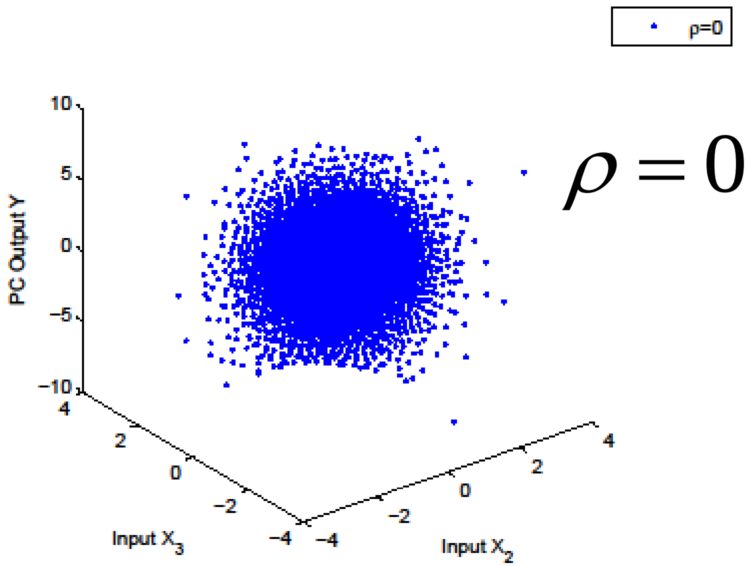


$$\rho = 0, 0.5, 0.9$$

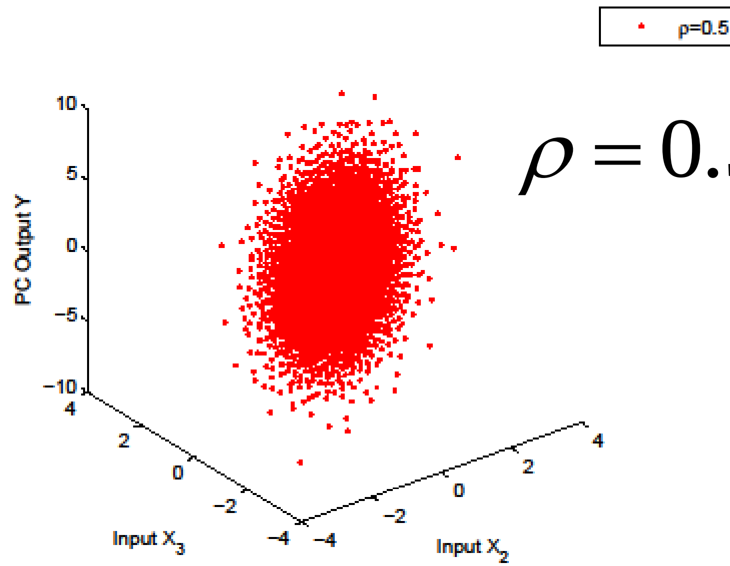
$$\log_2(\delta) \approx -10$$

Imperial College London

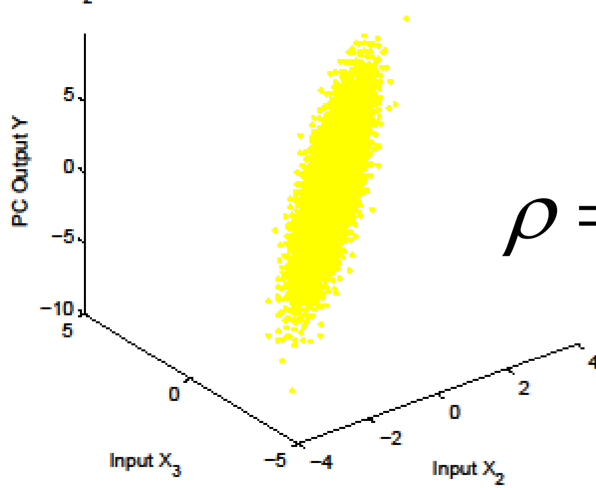
First Method.
Test example (Hyperplane)



$\rho = 0$



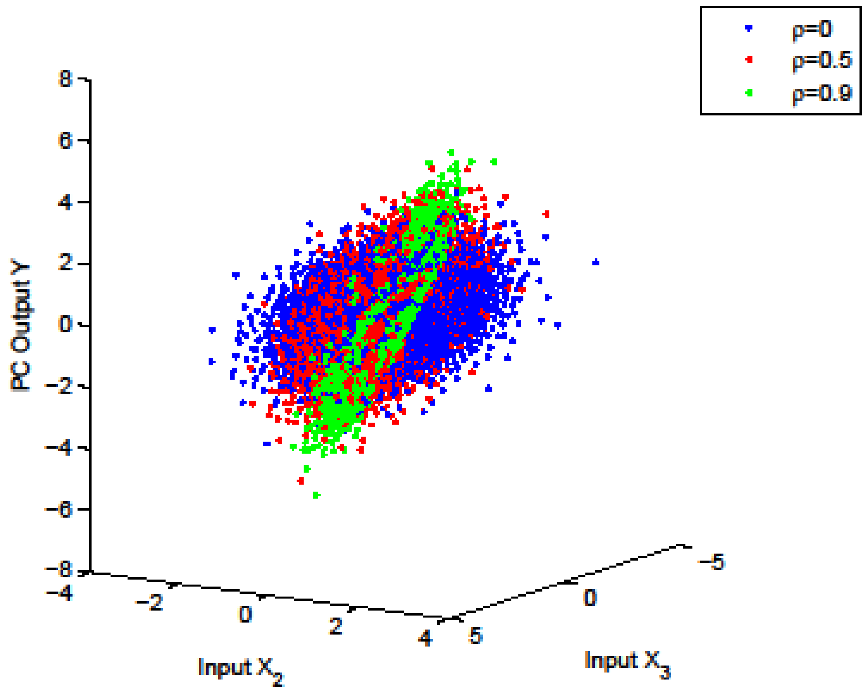
$\rho = 0.5$



$\rho = 0.9$

Imperial College London

First Method.
Test example (Hyperplane)



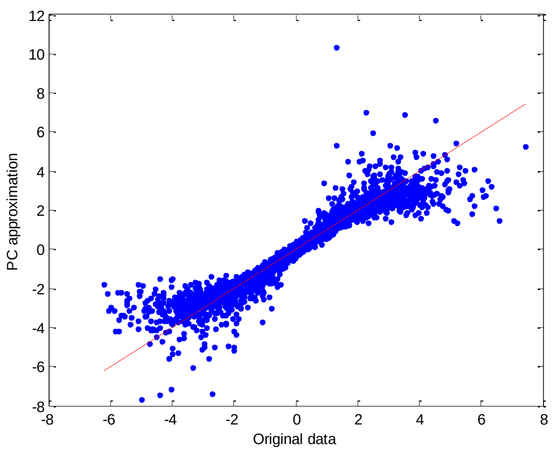
Imperial College London

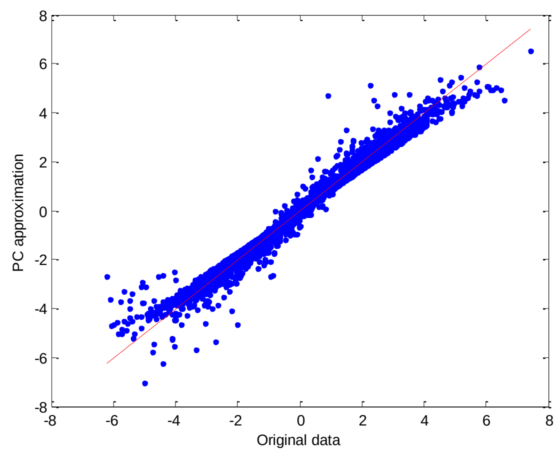
Second Method.
Test example (Hyperplane)

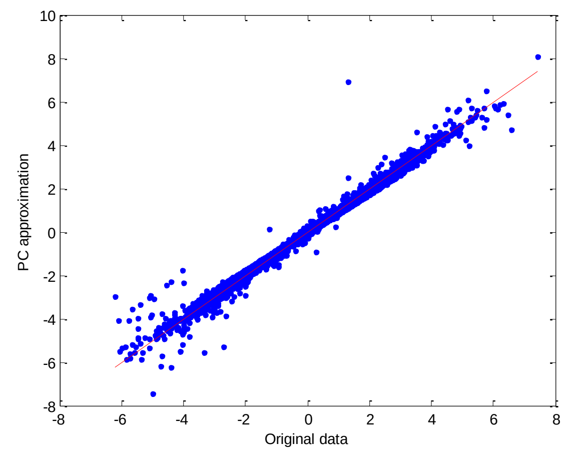
Number of QMC sampling N = 1024

$$\rho = 0.5$$

(2nd orders for the correlated variables)







Only 1st orders
[1; 1; 1]

$$\log_2(\delta) = -3.4$$

Only 1st and 2nd orders
[1; 1; 1; 4; 4]

$$\log_2(\delta) = -5.1$$

1st to 3rd orders
[1; 1; 1; 4; 4; 1; 1; 1]

$$\log_2(\delta) = -6$$

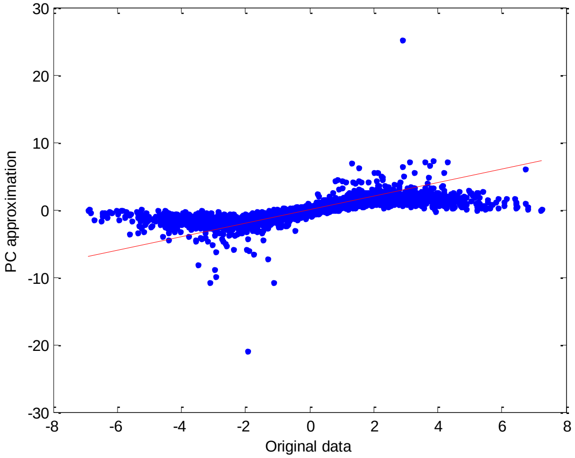
Imperial College London

Second Method. Test example (Hyperplane)

Number of QMC sampling N = 1024

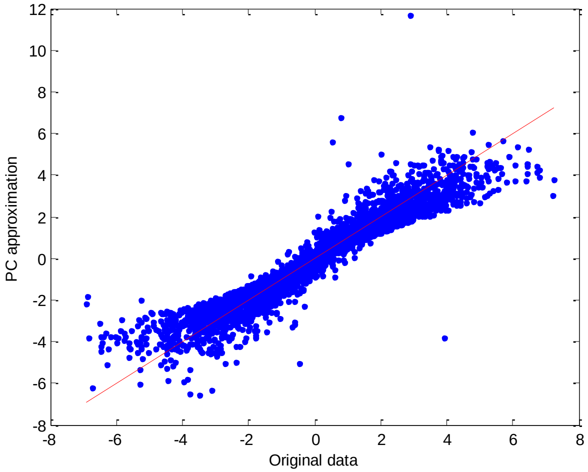
$$\rho = 0.9$$

(2nd orders for all couples of variables)



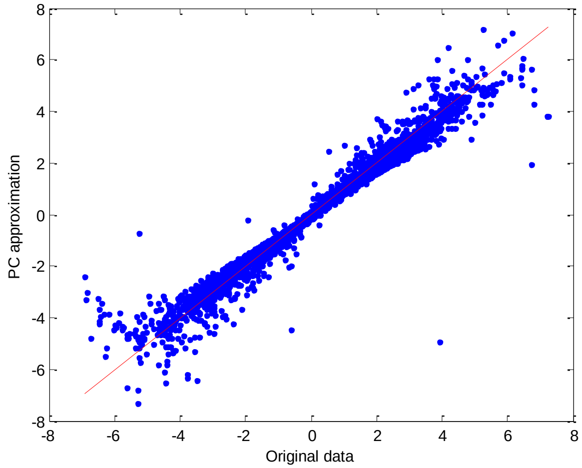
Only 1st orders
[1; 3; 3]

$$\log_2(\delta) = -3.1$$



Only 1st and 2nd orders
[1; 1; 1; 8; 8]

$$\log_2(\delta) = -3.2$$



1st to 3rd orders
[1; 1; 1; 8; 8; 4; 4; 4]

$$\log_2(\delta) = -4.6$$

Conclusions

Limitation of the second method

- The second approach is more straightforward but it can require polynomial orders and degrees much higher than those with the first method.
- High polynomial orders and degrees imply higher complexity.

Advantage of the second method

- The first method implicitly suppose that the dependence between the variables is given by a Gaussian copula.
- The advantage of the second approach is that it can be used when the dependence between the variables is given by any copula

Perspective: compare on more complex examples (Ishigami...)

Appendix: Numerical Results (Ishigami)

- *Model*

Inputs are uniformly distributed between -1 and 1

$$Y = f\left(X_1, X_2, X_3\right) = \sin\left(\pi X_1\right) + 7\sin\left(\pi X_2\right)^2 + 0.1\left(\pi X_3\right)^4 \sin\left(\pi X_1\right)$$

- *Correlation matrix*

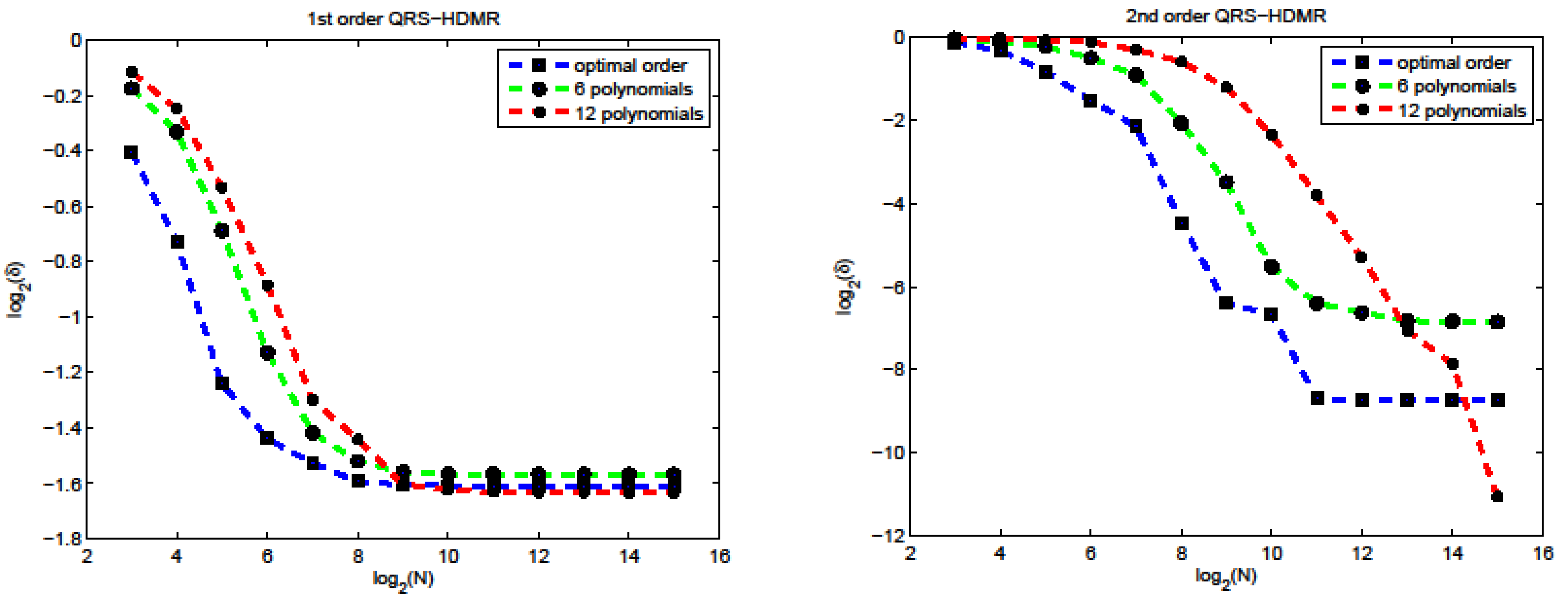
$$\Sigma = \begin{pmatrix} 1 & 0 & \rho \\ 0 & 1 & 0 \\ \rho & 0 & 1 \end{pmatrix}$$

- *Error measurement:*

$$\delta(f,h) = \frac{1}{\sigma}\int\left[f(x) - h(x)\right]^2 dx$$

Appendix: Numerical Results (Ishigami)

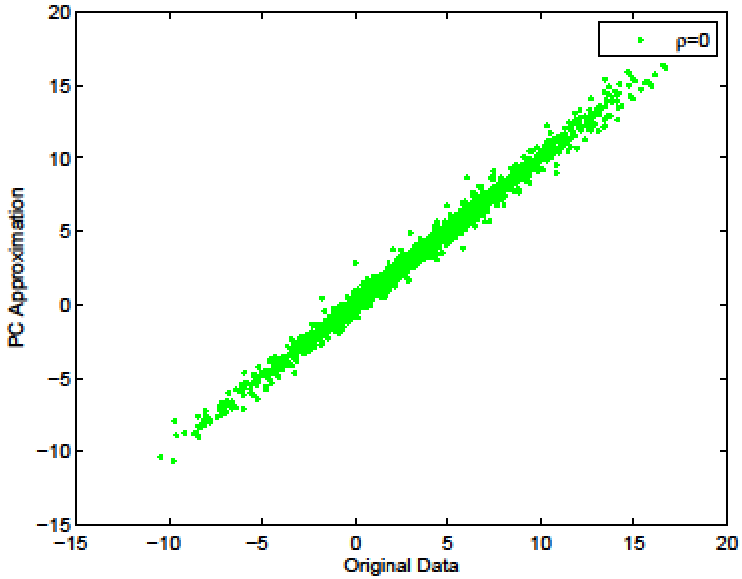
$$\rho = 0$$



Appendix:
Numerical Results (Ishigami)

- *N= 1024*
- *Optimal Orders = [3; 8; 0; 5; 5]*

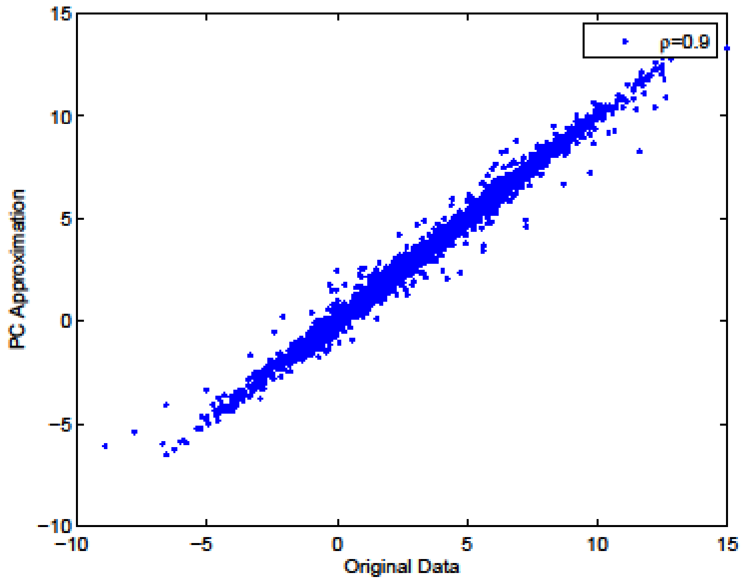
$$\rho = 0$$



$$\log_2(\delta) = -6.92$$

- *N= 1024*
- *Optimal Orders = [5; 8; 3; 5; 5]*

$$\rho = 0.9$$

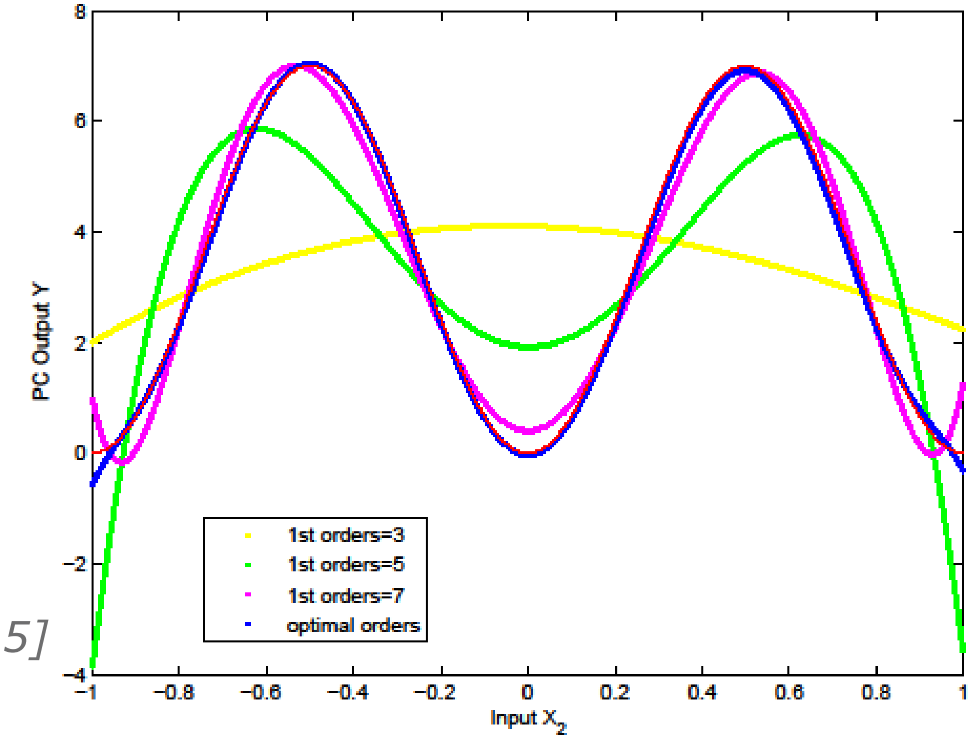


$$\log_2(\delta) = -6.53$$

Appendix:
Numerical Results (Ishigami)

**N=102
4**

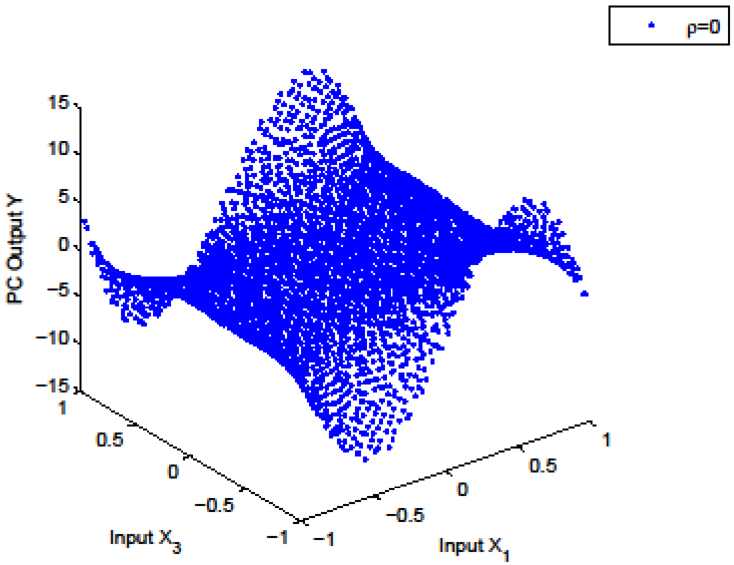
- *Orders = [3; 3; 3; 5; 5]*
- *Orders = [5; 5; 5; 5; 5]*
- *Orders = [7; 7; 7; 5; 5]*
- *Optimal Orders = [5; 8; 3; 5; 5]*



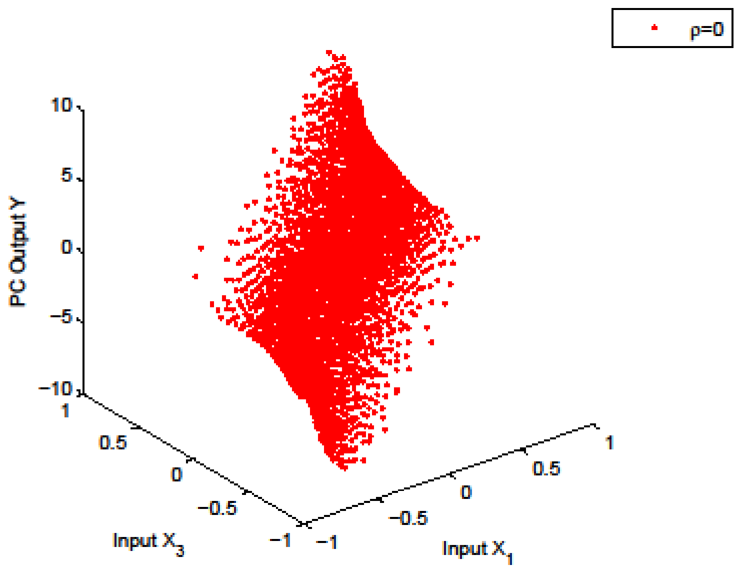
$$\rho = 0, 0.9$$

Appendix:
Numerical Results (Ishigami)

$$\rho = 0$$



$$\rho = 0.9$$



Appendix:
Numerical Results (Ishigami)