Efficient Usage of 2nd Order Sensitivity for Uncertainty Quantification

Marcin Wyrozębski

Supervision: prof. dr hab. inż. Jacek Rokicki
mgr Łukasz Łaniewski-Wołłk

March 24th, 2017

Outline

1. Introduction
2. Uncertainty Quantification
   - Method of Moments
   - Sensitivities computation
3. Numerical results
   - Parametrization
   - Uncertainty Quantification
4. Summary

Outline

1 Introduction

2 Uncertainty Quantification
- Method of Moments
- Sensitivities computation

3 Numerical results
- Parametrization
- Uncertainty Quantification

4 Summary

Introduction

Aim:

- Development of an uncertainty quantification method based on 2nd order sensitivities

Motivation

- The most broadly used approach for modeling structural and flow problems is fully deterministic. Simulations are led for a strictly specified inputs, such as
  - operational conditions
    - loads
    - pressures
    - free-stream parameters
  - geometrical data
    - airfoil shape
    - product dimensions
    - sheet metal thickness
- Assumption: inputs remain the same for every manufactured product
- Result: Value of the objective (lift force, temperature distribution) corresponding to the specified, model conditions and perfectly manufactured product.

Motivation

- Real life scenarios:
  - every product will be slightly different from the designed one and between each other due to
    - manufacturing tolerances
    - element wear-off
  - variability of operational conditions is unavoidable due to
    - existance of random environmental perturbations, e.g. ground vibrations, wind gusts
    - inaccurate in-flight measurements (preserving Mach number, AoA)
- One has to incorporate uncertainty management into the design process.

Motivation

State-of-the-art

- safety factor
- 6σ approach – minimize the chance for a failure
  - 5 uncertain steps
  - 3σ –> p(failure)$=$ 0.995
  - 6σ –> p(failure)$=$ 0.999999995

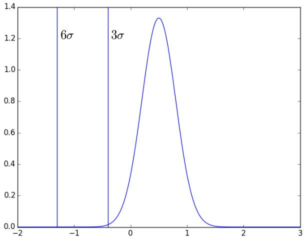


Figure: Gaussian PDF

Motivation

Research:

- Based on statistical parameters of inputs (mean, variance, pdf) compute statistical parameters of outputs (mean lift force/pressure drop)

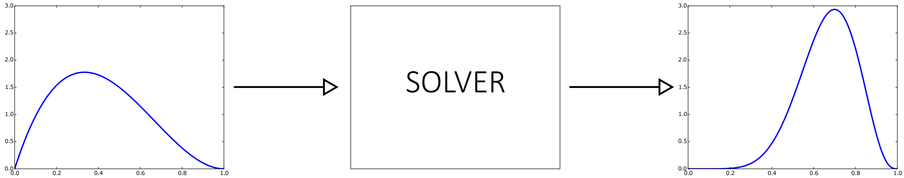


Figure: Input - airfoil thickness PDF, Output - lift force PDF

UMRIDA Project

Uncertainty Management for Robust Industrial Design in Aeronautics

- 7th Frame Programme EU Project
- Consortium of 21 partners from both academia and industry
- Aim:
  Analyze >10 uncertainties in 10 hours on 100 cores

UMRIDA Project

Tasks:

- Uncertainty Quantification (UQ)
  - evaluate statistical parameters (e.g.: mean, variance, kurtosis)
- Robust Design Optimisation
  - optimization under uncertainties (e.g.: minimize variance)
- Inverse Robust Design
  - determine input uncertainties based on defined requirements on the system performance
- ...and everything in a multi-objective framework

UMRIDA Project

Tasks:

- Uncertainty Quantification (UQ)
  - evaluate statistical parameters (e.g.: mean, variance, kurtosis)
- Robust Design Optimisation
  - optimization under uncertainties (e.g.: minimize variance)
- Inverse Robust Design
  - determine input uncertainties based on defined requirements on the system performance
- . . . and everything in a multi-objective framework

UMRIDA Project

Uncertainty Quantification Methods:

- Non-intrusive – CFD solver treated as a black-box
  - Multi-level Monte Carlo
    - run large number of independent, deterministic simulations
    - compute statistical quantities
  - Surrogate Models
    - run numerous, parallel simulations
    - perform polynomial expansion of a solution
- Intrusive – solver code manipulations
  - Method of Moments
    - Taylor series expansion of statistical quantity
    - evaluation of derivatives

UMRIDA Project

Uncertainty Quantification Methods:

- Non-intrusive – CFD solver treated as a black-box
  - Multi-level Monte Carlo
    - run large number of independent, deterministic simulations
    - compute statistical quantities
  - Surrogate Models
    - run numerous, parallel simulations
    - perform polynomial expansion of a solution
- Intrusive – solver code manipulations
  - Method of Moments
    - Taylor series expansion of statistical quantity
    - evaluation of derivatives

UMRIDA Project

Uncertainty Quantification subjects:

- operational
- geometrical

UMRIDA Project

Uncertainty Quantification subjects:

- operational
- geometrical

Typical UQ procedure for geometrical uncertainties

Geometrical uncertainties
↓
Continuous uncertainty field on body surface
↓
Discretization with CFD meshes
↓
Large number of correlated uncertainties
↓
Set of uncertainties too large to be analyzed in reasonable time
↓
Need of reduction

Outline

1. Introduction
2. Uncertainty Quantification
   - Method of Moments
   - Sensitivities computation
3. Numerical results
   - Parametrization
   - Uncertainty Quantification
4. Summary

Outline

1 Introduction

2 Uncertainty Quantification
- Method of Moments
- Sensitivities computation

3 Numerical results
- Parametrization
- Uncertainty Quantification

4 Summary

Method of Moments

Let us assume

- f – objective (lift, drag force)
- x – geometrical parametrization
- ζ – uncertainties, random variables

Mean value – Taylor series expansion:

$$\mathrm{E}[f(x + h\zeta)] =$$

Method of Moments

Let us assume

- f – objective (lift, drag force)
- x – geometrical parametrization
- ζ – uncertainties, random variables

Mean value – Taylor series expansion:

$$\mathrm{E}[f(x+h\zeta)] = \mathrm{E}\left[f(x) + h\zeta_i \frac{\partial f}{\partial x_i} + \frac{1}{2}h^2\zeta_i\zeta_j\frac{\partial^2 f}{\partial x_i \partial x_j} + o(h^3)\right]$$

Method of Moments

Let us assume

- f – objective (lift, drag force)
- x – geometrical parametrization
- ζ – uncertainties, random variables

Mean value – Taylor series expansion:

$$\begin{aligned} \mathrm{E}[f(x+h\zeta)] &= \mathrm{E}\left[f(x) + h\zeta_i \frac{\partial f}{\partial x_i} + \frac{1}{2}h^2\zeta_i\zeta_j\frac{\partial^2 f}{\partial x_i \partial x_j} + o(h^3)\right] \\ &= f(x) + h\frac{\partial f}{\partial x_i}\mathrm{E}\left[\zeta_i\right] + \frac{1}{2}h^2\frac{\partial^2 f}{\partial x_i \partial x_j}\mathrm{E}\left[\zeta_i\zeta_j\right] + o(h^3) \end{aligned}$$

Method of Moments

Let us assume

- f – objective (lift, drag force)
- x – geometrical parametrization
- ζ – uncertainties, random variables

Mean value – Taylor series expansion:

$$\begin{aligned} \mathrm{E}[f(x+h\zeta)] &= \mathrm{E}\left[f(x) + h\zeta_i \frac{\partial f}{\partial x_i} + \frac{1}{2}h^2 \zeta_i \zeta_j \frac{\partial^2 f}{\partial x_i \partial x_j} + o(h^3)\right] \\ &= f(x) + h\frac{\partial f}{\partial x_i}\mathrm{E}\left[\zeta_i\right] + \frac{1}{2}h^2 \frac{\partial^2 f}{\partial x_i \partial x_j}\mathrm{E}\left[\zeta_i \zeta_j\right] + o(h^3) \\ &= f(x) + \frac{1}{2}h^2 \frac{\partial^2 f}{\partial x_i \partial x_j}\mathrm{E}\left[\zeta_i \zeta_j\right] + o(h^3) \end{aligned}$$

Method of Moments

Proposed method

- Cut-off at 3rd order term

$$E[f(x + h\zeta)] = f(x) + \frac{1}{2}h^2 \frac{\partial f}{\partial x_i \partial x_j} C_{ij} + o(h^3)$$

Method of Moments

Proposed method

- Cut-off at 3rd order term

$$E[f(x+h\zeta)] = f(x) + \frac{1}{2}h^2\frac{\partial f}{\partial x_i \partial x_j}C_{ij}$$

Covariance matrix

- measurements
- assumption
- simplified model

Reduction in CPU cost and memory on covariance matrix

- Highly correlated nodal uncertainties
  - Dense covariance matrix
  - Low Rank Approximation
- Uncorrelated nodal uncertainties
  - Sparse covariance matrix
  - Might be need to analyze larger number of modes to preserve accuracy

Method of Moments

Proposed method

- Cut-off at 3rd order term

$$\mathrm{E}[f(x+h\zeta)] = f(x) + \frac{1}{2}h^2 \frac{\partial f}{\partial x_i \partial x_j} C_{ij}$$

Hessian matrix

- Large number of uncertainties
- Expensive construction of a full matrix
- Reduction techniques
- Select several good base vectors to represent the full problem

Method of Moments

Proposed method

$$\mathrm{E}[f(x+h\zeta)] = f(x) + \frac{1}{2}h^2 \frac{\partial f}{\partial x_i \partial x_j} C_{ij}$$

Properties

- Choose representatives w.r.t. largest eigenvalues

$$H_{ij}C_{ij} = H_{ij}C_{ji} = \sum_i A_{ii} = \sum_i \lambda_i$$

$$Hv = \lambda C^{-1} v$$

- No need to construct full Hessian matrix
- Requires only vector-by-hessian multiplication (power method)
- Inexpensive vector-by-hessian multiplication – cost proportional to primal iteration (tangent-on-reverse)
- Accuracy and cost depend on number of analyzed modes

How to efficiently compute sensitivities in CFD?

Outline

1. Introduction
2. Uncertainty Quantification
   - Method of Moments
   - Sensitivities computation
3. Numerical results
   - Parametrization
   - Uncertainty Quantification
4. Summary

Sensitivity computation – gradient

Finite Difference Method – simple approach

- for **each parameter** solve an additional primal problem $J(x+h)$

$$\frac{\partial J}{\partial x} \approx \frac{J(x+h) - J(x)}{h}$$

Sensitivity computation – gradient

Finite Difference Method – simple approach

- for **each parameter** solve an additional primal problem $J(x+h)$

$$\frac{\partial J}{\partial x} \approx \frac{J(x+h) - J(x)}{h}$$

Adjoint method

- developed in '70s for the structural and optimal control problems
- nowadays commonly used also in CFD simulations
- cost of **full gradient** computation proportional to one primal **iteration**

Sensitivity computation – gradient

Let us assume

- u – flow problem solution
- α – set of design parameters
- $R(u, \alpha)$ – flow equations (Euler, RANS)
- $J(u, \alpha)$ – objective function to be optimized (lift/drag force)

Optimization under constraints (functional analysis) – Augmented Lagrangian

$$I(u, \alpha) = J(u, \alpha) - \lambda^T R(u, \alpha)$$

Under some assumptions:

$$dI(u, \alpha) = \frac{\partial J}{\partial u} dU + \frac{\partial J}{\partial \alpha} d\alpha - \lambda^T \left(\frac{\partial R}{\partial u} dU + \frac{\partial R}{\partial \alpha} d\alpha \right)$$

$$dI(u, \alpha) = \left(\frac{\partial J}{\partial u} - \lambda^T \frac{\partial R}{\partial u} \right) dU + \left(\frac{\partial J}{\partial \alpha} - \lambda^T \frac{\partial R}{\partial \alpha} \right) d\alpha$$

Sensitivity computation – gradient

Adjoint method splits the formula into two parts corresponding to flow and parametrization

$$dI(u,\alpha) = \underbrace{\left(\frac{\partial J}{\partial u} - \lambda^T \frac{\partial R}{\partial u}\right)}_{\text{flow variables}} dU + \underbrace{\left(\frac{\partial J}{\partial \alpha} - \lambda^T \frac{\partial R}{\partial \alpha}\right)}_{\text{design parameters}} d\alpha$$

If the adjoint equation is satisfied

$$\left(\frac{\partial R}{\partial u}\right)^T \lambda = \frac{\partial J}{\partial u}$$

then the gradient of the objective w.r.t. parameters is equal to

$$\frac{dI(u,\alpha)}{d\alpha} = \frac{\partial J}{\partial \alpha} - \lambda^T \frac{\partial R}{\partial \alpha}$$

Sensitivity computation – gradient

Adjoint equation: $\left(\frac{\partial R}{\partial u}\right)^T \lambda = \frac{\partial J}{\partial u}$

- does not depend on the parametrization
- its solution λ is a sensitivity of the objective on adding a local, nodal source at given point
- cost is proportional to one iteration of implicit solver $\frac{\partial R}{\partial u}\Delta u = -R$

Gradient equation: $\frac{dI}{d\alpha} = \frac{\partial J}{\partial \alpha} - \lambda^T \frac{\partial R}{\partial \alpha}$

- depends only on the design parameters
- very cheap
- for a shape optimization number of parameters is proportional to number of surface nodes – $o(N^2)$ with a complexity of the flow problem – $o(N^3)$

Sensitivity computation – 2nd order

Hessian matrix computation

- Extension of adjoint method
- Required only multiplication by vector
- Cost of one multiplication proportional to solving one tangent and one adjoint equation
- Total cost proportional to number of analyzed directions, not number of parameters

Sensitivity computation – 2^{nd} order

Procedure for hessian multiplication

1. Solving primal equation (Euler, Navier-Stokes)

$$R_i(u) = 0$$

2. Solving adjoint equation (J - objective)

$$\frac{\partial R_i}{\partial u_j} v_i = -\frac{\partial J}{\partial u_j}$$

3. Gradient computation

$$\frac{d}{d\alpha_k} J = \frac{\partial J}{\partial \alpha_k} + v_i \frac{\partial R_i}{\partial \alpha_k}$$

4. ...

Sensitivity computation – 2nd order

Procedure for hessian multiplication (cont.)

4. For each direction β
   1. Solving tangent equation

$$\frac{\partial R_i}{\partial u_q} b_q = -\beta_p \frac{\partial R_i}{\partial \alpha_p}$$

   2. Solving adjoint equation

$$\frac{\partial R_i}{\partial u_j} a_i = -\left[\left(b_q \frac{\partial}{\partial u_q} + \beta_p \frac{\partial}{\partial \alpha_p}\right) \frac{\partial}{\partial u_j}(J + v_i R_i)\right]$$

   3. Multiplication of Hessian by given β

$$\beta_p \frac{d^2}{d\alpha_k d\alpha_p}(J) = a_i \frac{\partial R_i}{\partial \alpha_k} + \left(b_q \frac{\partial}{\partial u_q} + \beta_p \frac{\partial}{\partial \alpha_p}\right) \frac{\partial}{\partial \alpha_k}(J + v_i R_i)$$

Sensitivity computation – 2nd order

Procedure for hessian multiplication (cont.)

4. For each direction β
   1. Solving tangent equation

$$\frac{\partial R_i}{\partial u_q} b_q = -\beta_p \frac{\partial R_i}{\partial \alpha_p}$$

   2. Solving adjoint equation

$$\frac{\partial R_i}{\partial u_j} a_i = -\left[\left(b_q \frac{\partial}{\partial u_q} + \beta_p \frac{\partial}{\partial \alpha_p}\right) \frac{\partial}{\partial u_j}(J + v_i R_i)\right]$$

   3. Multiplication of Hessian by given β

$$\beta_p \frac{d^2}{d\alpha_k d\alpha_p}(J) = a_i \frac{\partial R_i}{\partial \alpha_k} + \left(b_q \frac{\partial}{\partial u_q} + \beta_p \frac{\partial}{\partial \alpha_p}\right) \frac{\partial}{\partial \alpha_k}(J + v_i R_i)$$

Considering β as versors, one can construct full Hessian matrix

Sensitivity computation – 2^{nd} order

State equations $R(u)$ is nonlinear, thus a numerical differentiation technique is required:

- Finite Difference Method
  - easy implementation
  - very efficient when applied locally
  - no special memory requirements
  - inaccurate

Sensitivity computation – 2^{nd} order

State equations $R(u)$ is nonlinear, thus a numerical differentiation technique is required:

- Finite Difference Method
  - easy implementation
  - very efficient when applied locally
  - no special memory requirements
  - inaccurate
- Automatic Differentiation Tools (AD)
  - exact, even for highly nonlinear cases
  - higher memory requirements (operator overloading)
  - ability to use depends on the solver
  - in most cases difficult to implement in parallel
  - Tapenade (INRIA), DCO (RWTH)

Sensitivity computation – 2^{nd} order

State equations $R(u)$ is nonlinear, thus a numerical differentiation technique is required:

- Finite Difference Method
  - easy implementation
  - very efficient when applied locally
  - no special memory requirements
  - inaccurate
- Automatic Differentiation Tools (AD)
  - exact, even for highly nonlinear cases
  - higher memory requirements (operator overloading)
  - ability to use depends on the solver
  - in most cases difficult to implement in parallel
  - Tapenade (INRIA), DCO (RWTH)

Sensitivity computation – 2nd order

State equations $R(u)$ is nonlinear, thus a numerical differentiation technique is required:

- Finite Difference Method
  - easy implementation
  - very efficient when applied locally
  - no special memory requirements
  - inaccurate
- Automatic Differentiation Tools (AD)
  - exact, even for highly nonlinear cases
  - higher memory requirements (operator overloading)
  - ability to use depends on the solver
  - in most cases difficult to implement in parallel
  - Tapenade (INRIA), DCO (RWTH)

Outline

1 Introduction

2 Uncertainty Quantification
- Method of Moments
- Sensitivities computation

3 Numerical results
- Parametrization
- Uncertainty Quantification

4 Summary

Numerical results

Flow2/RED solver:

- in-house tool developed by Jerzy Majewski
- Residual Distribution Scheme
  - Multidimensional upwind
  - Lower numerical diffusion compared to FVM
  - Residuum computed locally inside cell
- Equations: Compressible Euler, Navier-Stokes, RANS
- Common turbulence models: Spalart-Allmaras, k-ω
- 2D/3D, unstructured meshes
- C++ Object-Oriented
- Parallelization: MPI, PETSc, Domain decomposition
- Good scalability
- Verified accuracy (ADIGMA, IDIHOM)

Numerical results

Flow2/RED extension:

- Mesh deformation
- Optimization (Adjoint method)
- Uncertainty Quantification
- Source transformation (Tapenade)
- Verification and validation

Numerical results

BC-03 UMRIDA Test-case

- Geometry: DLR-F6
- Euler equations
- Transonic conditions: $M = 0.76$, $AoA = 1^\circ$
- Objective: lift force

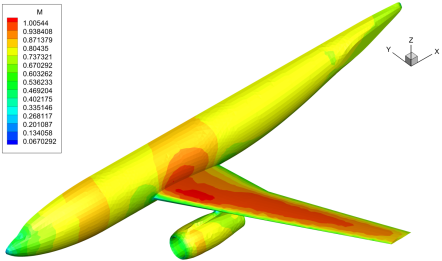


Figure: Solution - distribution of Mach number

Outline

1 Introduction

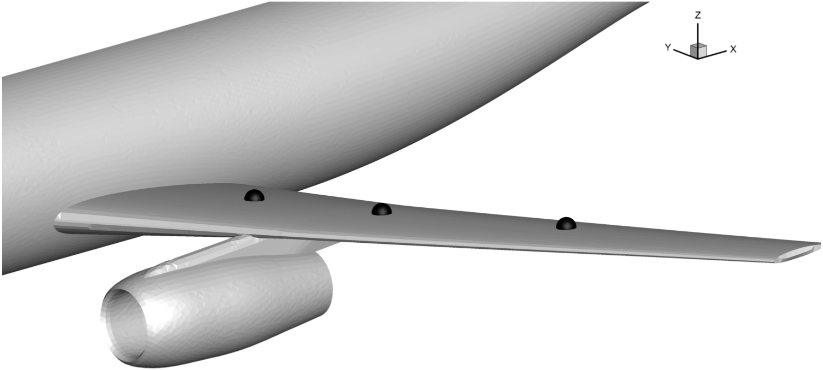
2 Uncertainty Quantification
- Method of Moments
- Sensitivities computation

3 Numerical results
- Parametrization
- Uncertainty Quantification

4 Summary

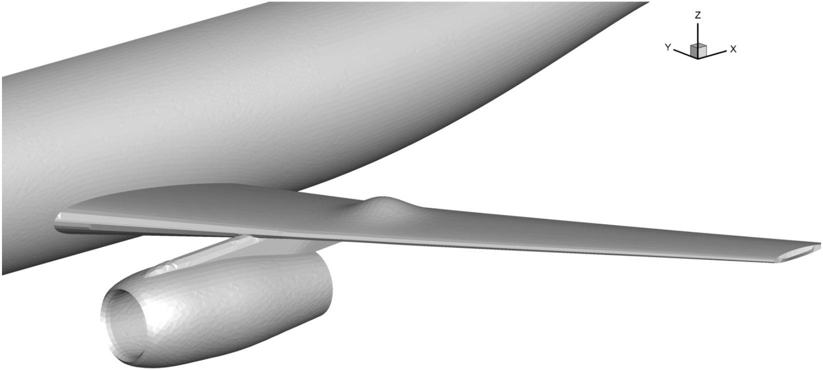
Numerical results – parametrization

Radial Basis Function



Numerical results – parametrization

Radial Basis Function



Numerical results – parametrization

Different distributions available

- leading/trailing edge
- maximum variance

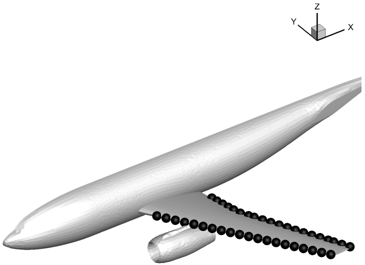


Figure: Leading and trailing edge

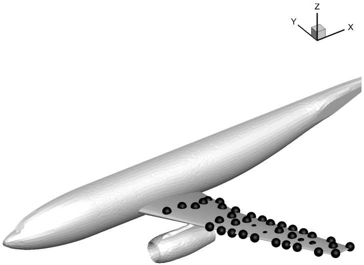


Figure: Max. variance distribution

Numerical results – parametrization

Possible freezing of specific geometry regions

- Example with fixed fuselage and nacelle

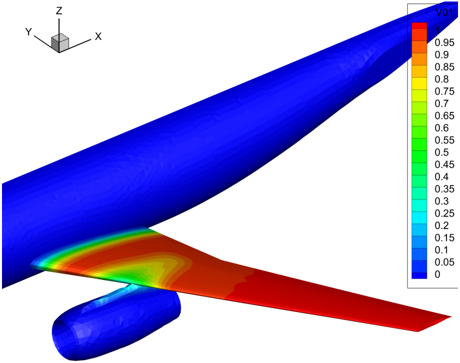
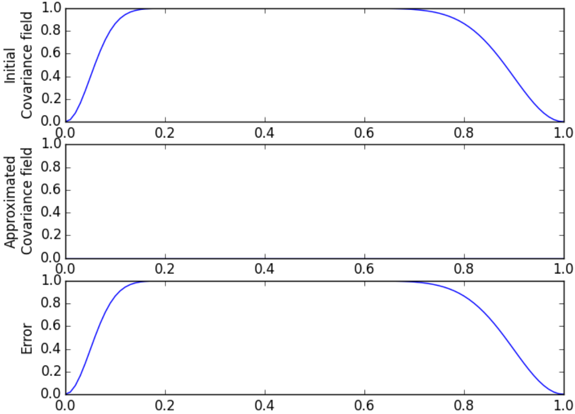


Figure: Variance distribution on surface

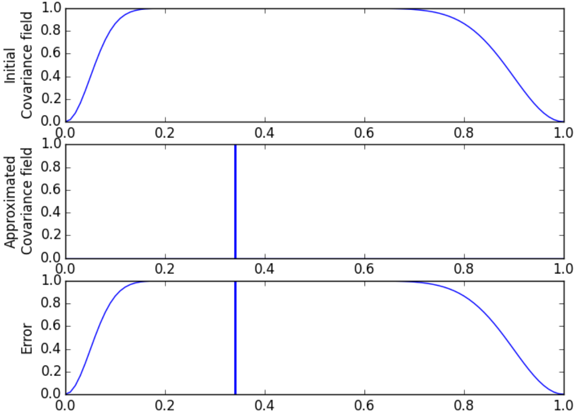
Numerical results

Max. variance distribution – 1D example



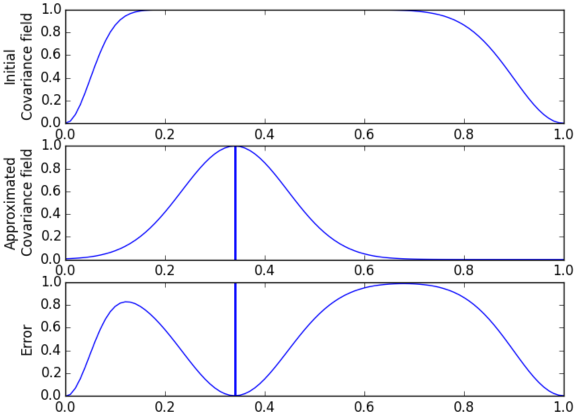
Numerical results

Max. variance distribution – 1D example



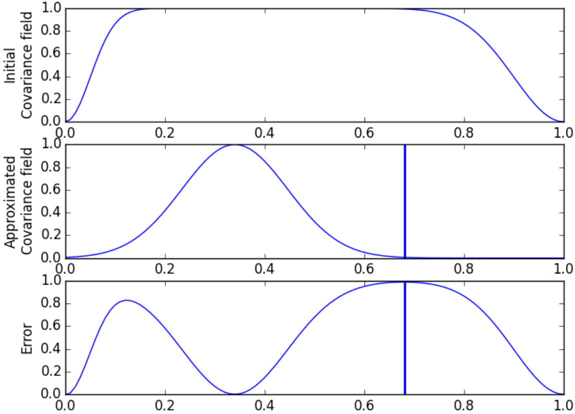
Numerical results

Max. variance distribution – 1D example



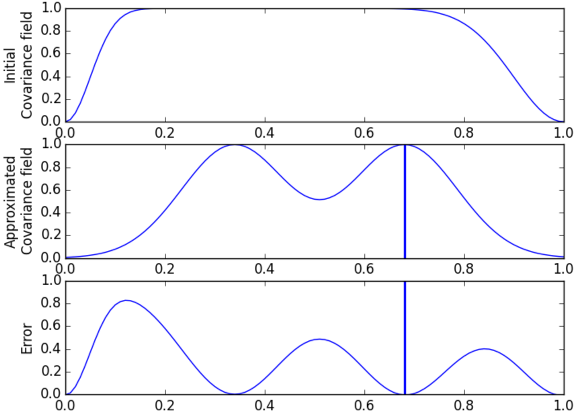
Numerical results

Max. variance distribution – 1D example



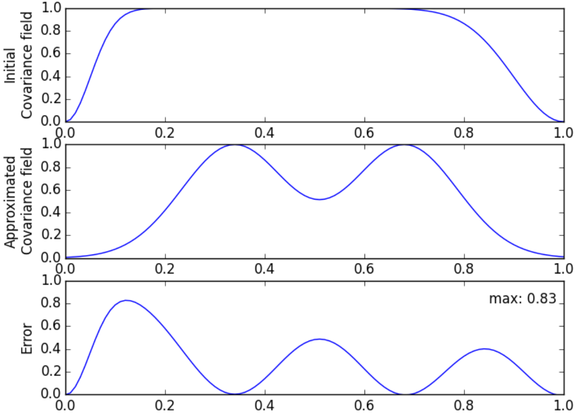
Numerical results

Max. variance distribution – 1D example



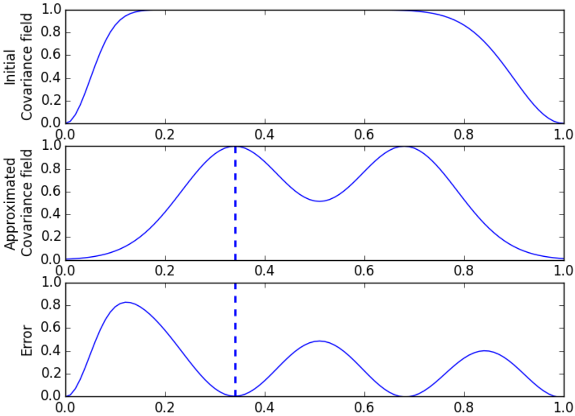
Numerical results

Max. variance distribution – 1D example



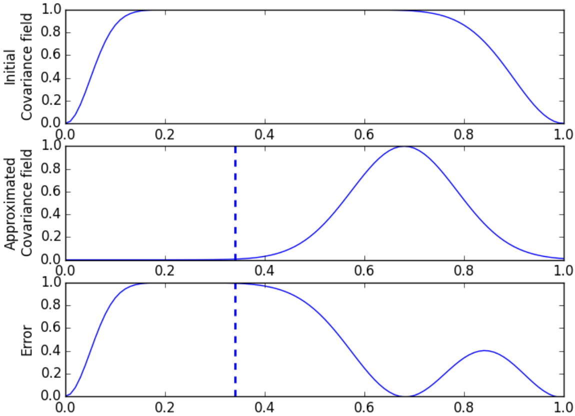
Numerical results

Max. variance distribution – 1D example



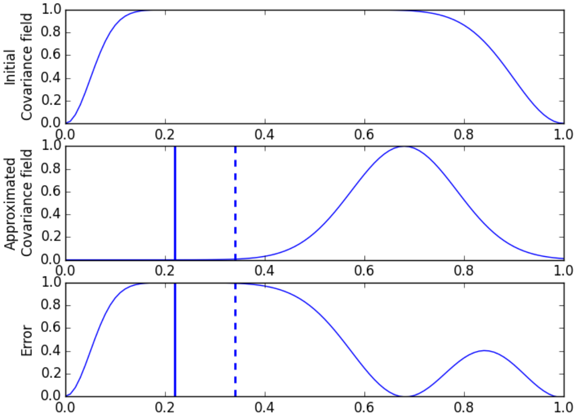
Numerical results

Max. variance distribution – 1D example



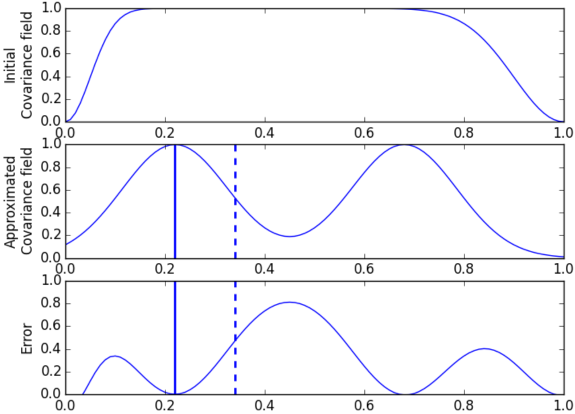
Numerical results

Max. variance distribution – 1D example



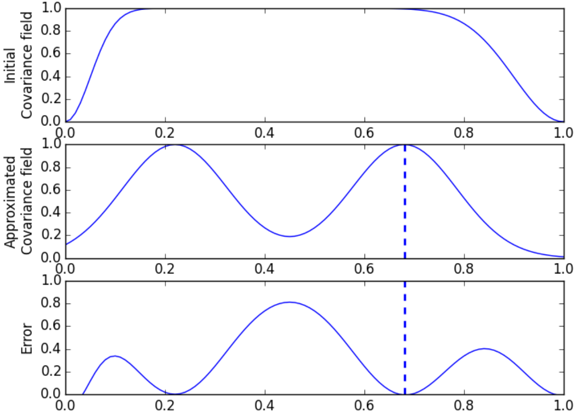
Numerical results

Max. variance distribution – 1D example



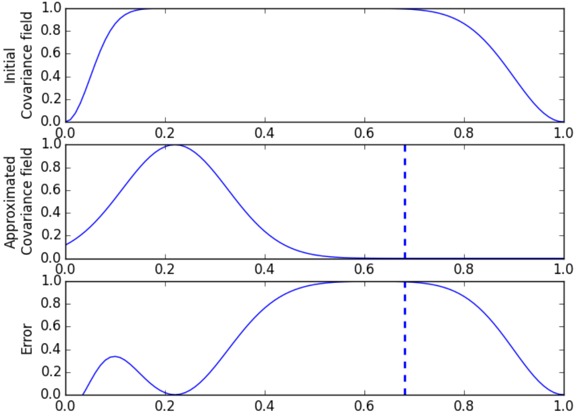
Numerical results

Max. variance distribution – 1D example



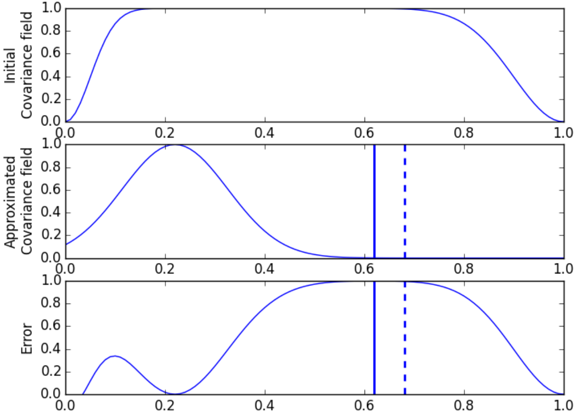
Numerical results

Max. variance distribution – 1D example



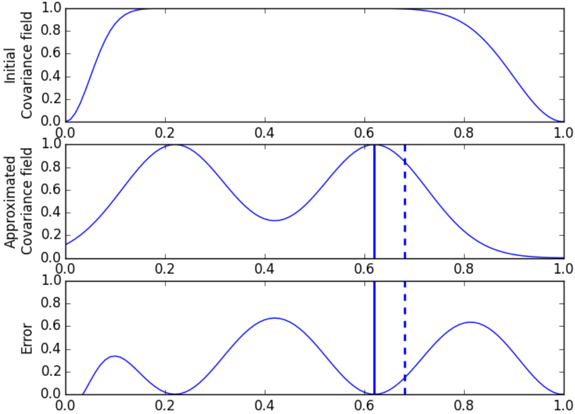
Numerical results

Max. variance distribution – 1D example



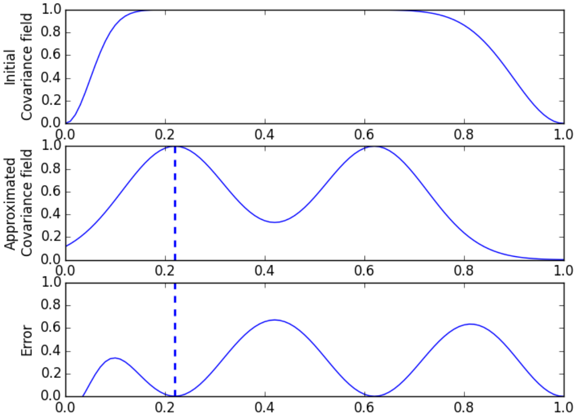
Numerical results

Max. variance distribution – 1D example



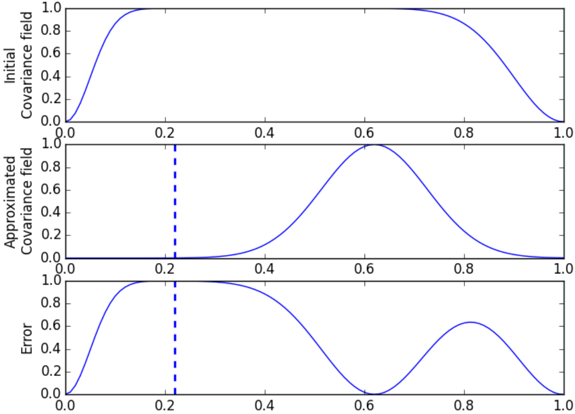
Numerical results

Max. variance distribution – 1D example



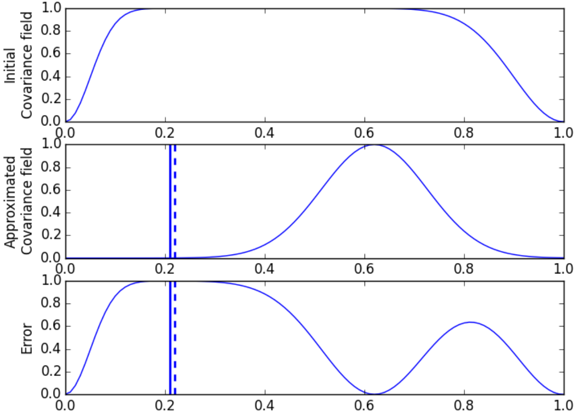
Numerical results

Max. variance distribution – 1D example



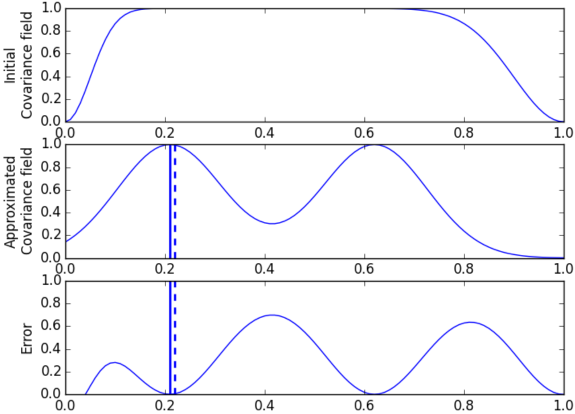
Numerical results

Max. variance distribution – 1D example



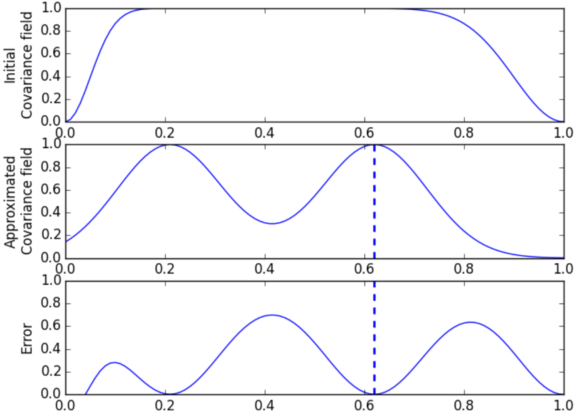
Numerical results

Max. variance distribution – 1D example



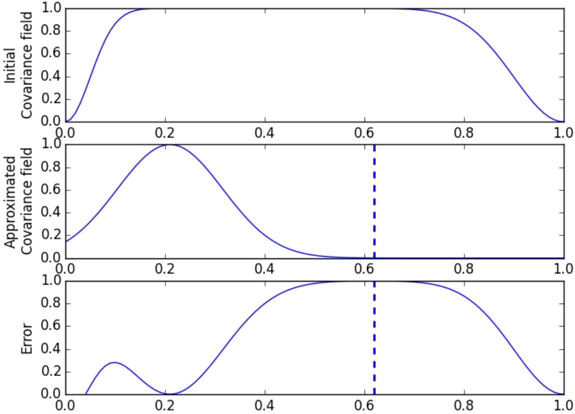
Numerical results

Max. variance distribution – 1D example



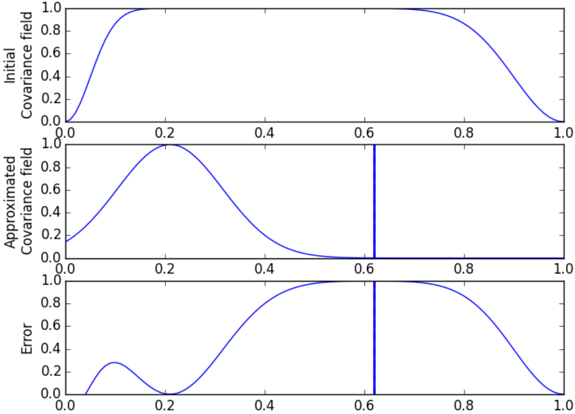
Numerical results

Max. variance distribution – 1D example



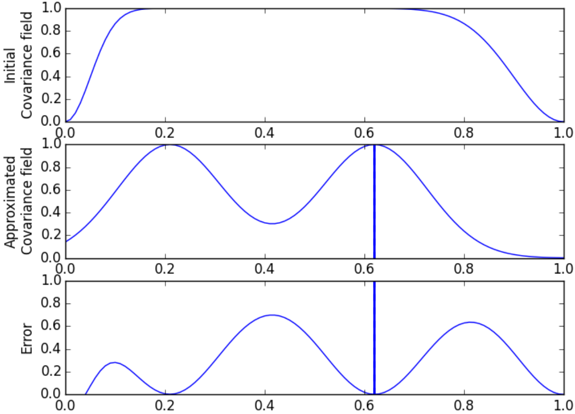
Numerical results

Max. variance distribution – 1D example



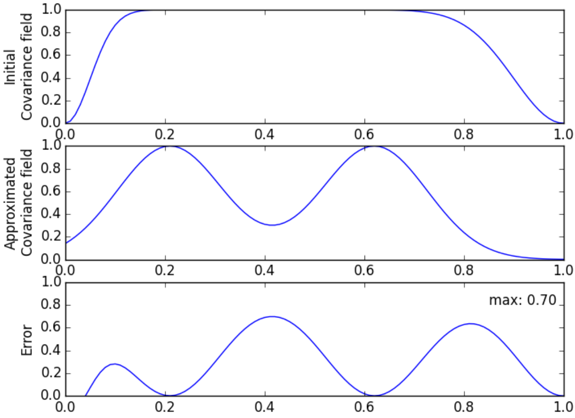
Numerical results

Max. variance distribution – 1D example



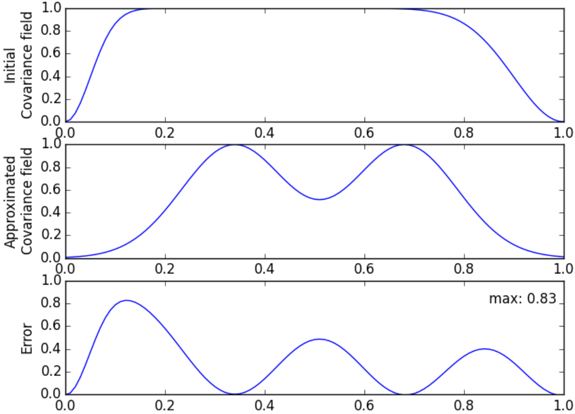
Numerical results

Max. variance distribution – 1D example



Numerical results

Max. variance distribution – 1D example



Numerical results

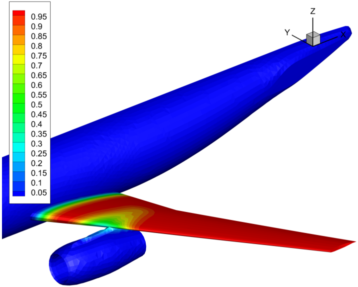


Figure: Variance distribution

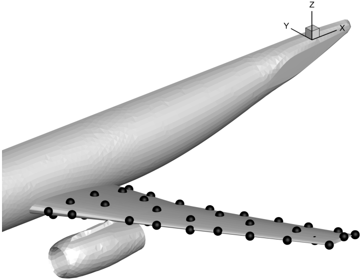


Figure: RBF distribution

Outline

1 Introduction

2 Uncertainty Quantification
- Method of Moments
- Sensitivities computation

3 Numerical results
- Parametrization
- Uncertainty Quantification

4 Summary

Numerical results – Uncertainty Quantification

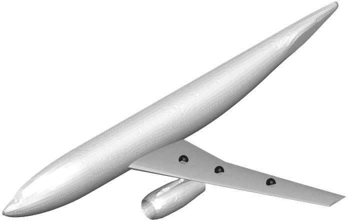


Figure: RBF Distribution

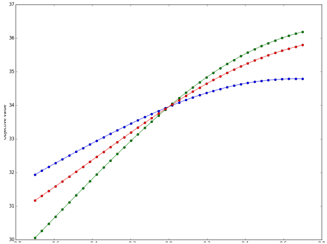


Figure: Objective value

Hessian validation against:

- Kriging
- Polynomial fitting

Small differences – 3%

- Which one is the most accurate?

Numerical results – Uncertainty Quantification

- Objective, gradient and hessian investigation on meshes with different element size

Mesh (# nodes)	60k	200k	300k	400k
Objective rel. error	-0.24976	-0.09120	-0.05415	ref.
Gradient rel. error	0.55138	0.30017	0.16180	ref.
Hessian rel. error	11.09583	7.40420	4.53642	ref.

- Error decreasing on finer meshes
- Relatively high errors – slightly different parameterization across meshes

Numerical results – Uncertainty Quantification

Hessian - Eigenvalues spectrum

- Mesh size: 300k nodes
- Parametrization: 40 RBF (max. variance distribution)

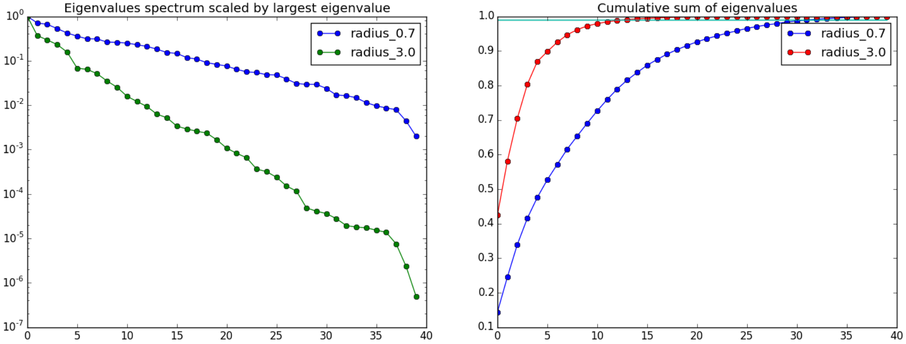


Figure: Generalized eigenvalue solution

Numerical results – Uncertainty Quantification

Hessian - Eigenvalues spectrum

- Mesh size: 300k nodes
- Parametrization: 40 RBF (max. variance distribution)

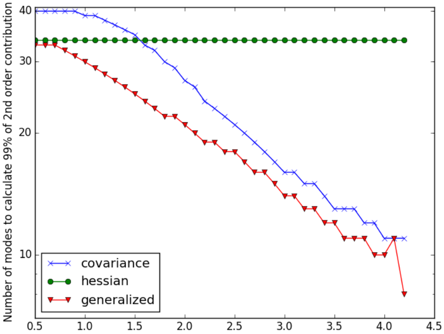


Figure: Number of modes required for 99% representation of 2nd order information as function of parametrization correlation radius

Objective — Mean-value

- Comparison of mean-value estimation

$$\mathrm{E}\left[f(x+h\zeta)\right] \approx f + \underbrace{\frac{1}{2}h^2\frac{\partial f}{\partial x_i \partial x_j}C_{ij}}_{\Delta f}$$

Method	Δf
Monte Carlo	0.4026461
Kriging	0.4025724
Our method	0.3514531
Kriging (2nd order)	0.3489399

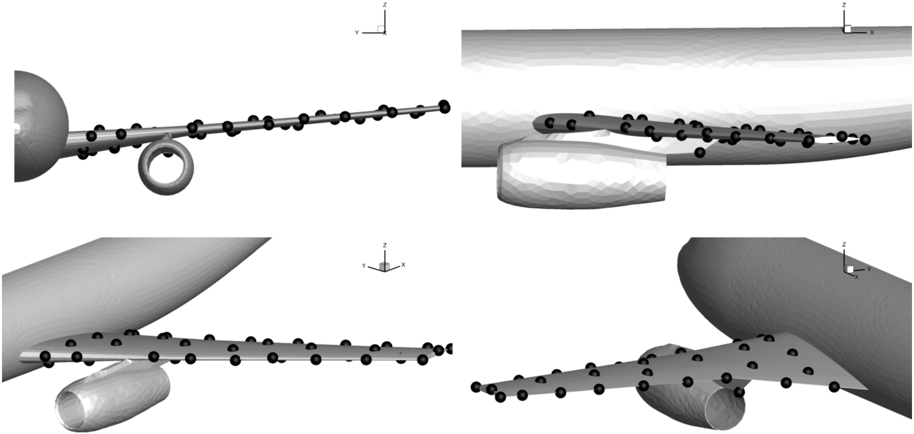
- Relatively high error in objective correction (Δf) caused by Taylor series cut-off
- Good agreement with Kriging (based on hessian)

Resulting eigenvectors

Orthogonal base of eigenvectors

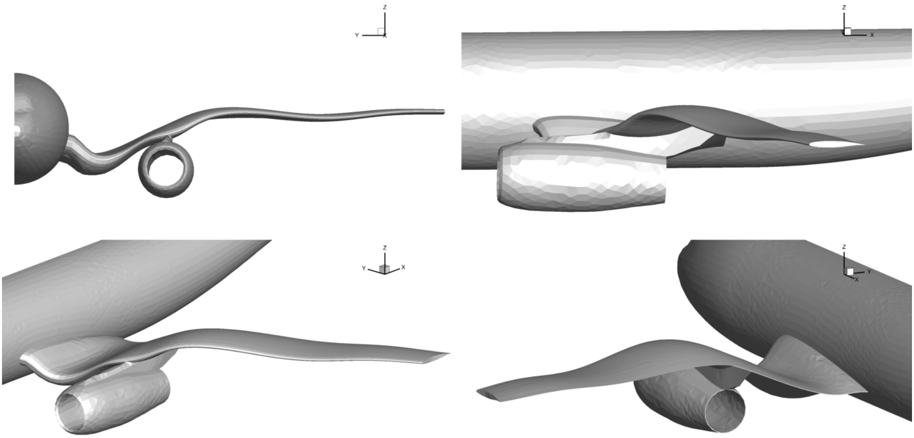
- Our method gives a convienient base for the UQ problem
  - Diagonal covariance matrix — **independent uncertainties**
  - No cross-terms in 2nd order derivatives — **less coefficients in polynomial approximation**
- Eigenvectors – geometry deformations that produces the most mean-value shift caused by uncertain input parameters
- Resulting shape can be an important information in the design process.

Resulting eigenvectors



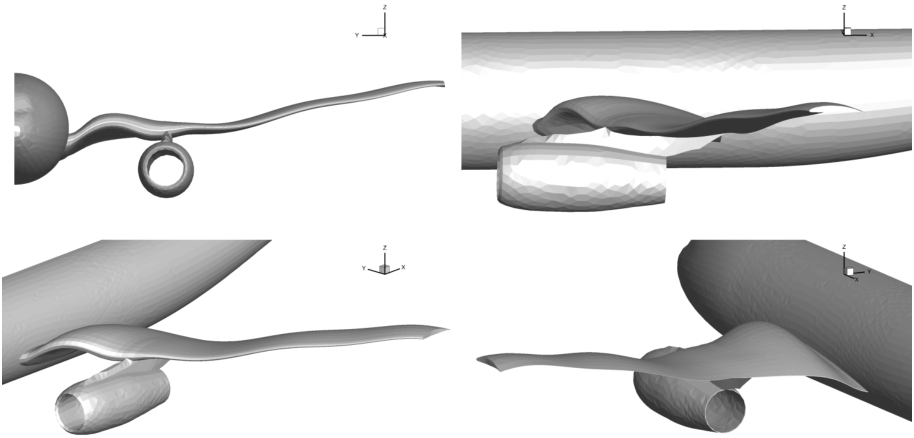
Base shape with parameters location

Resulting eigenvectors



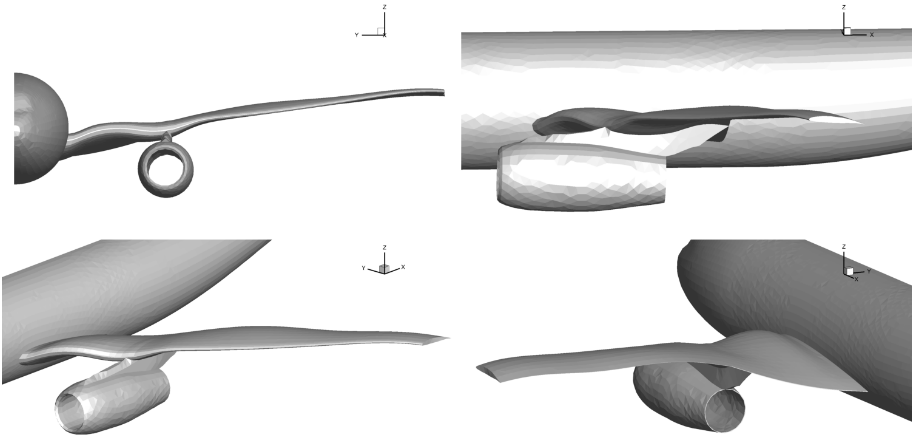
1st Eigenvector

Resulting eigenvectors



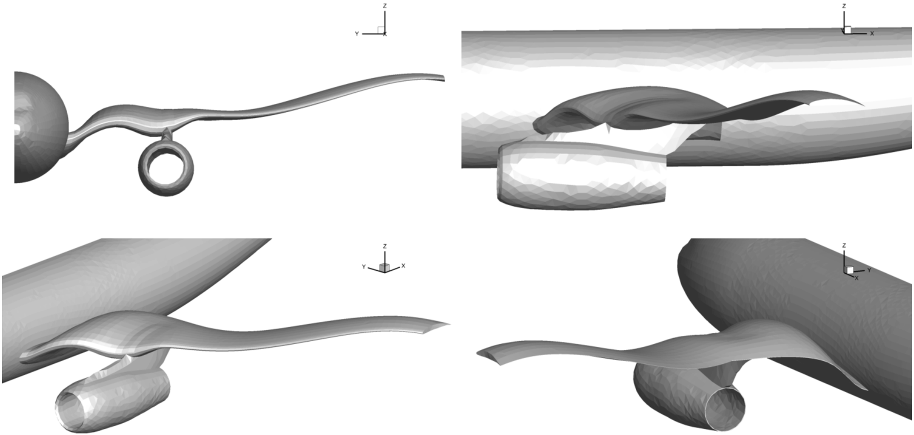
2nd Eigenvector

Resulting eigenvectors



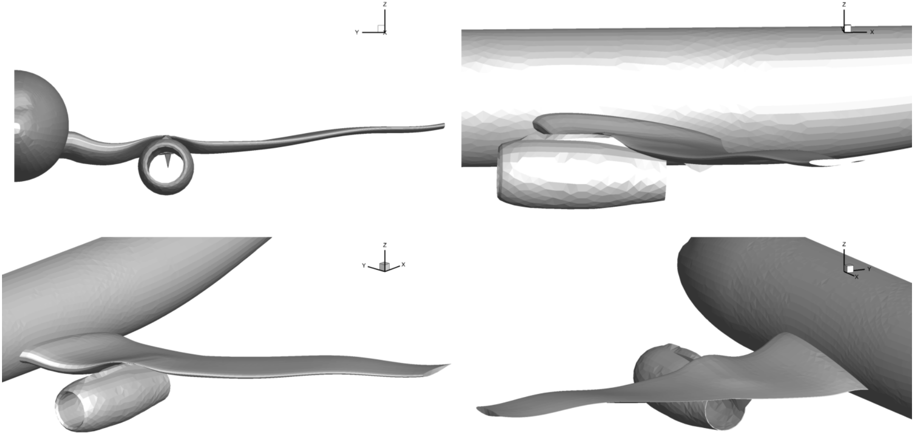
3rd Eigenvector

Resulting eigenvectors



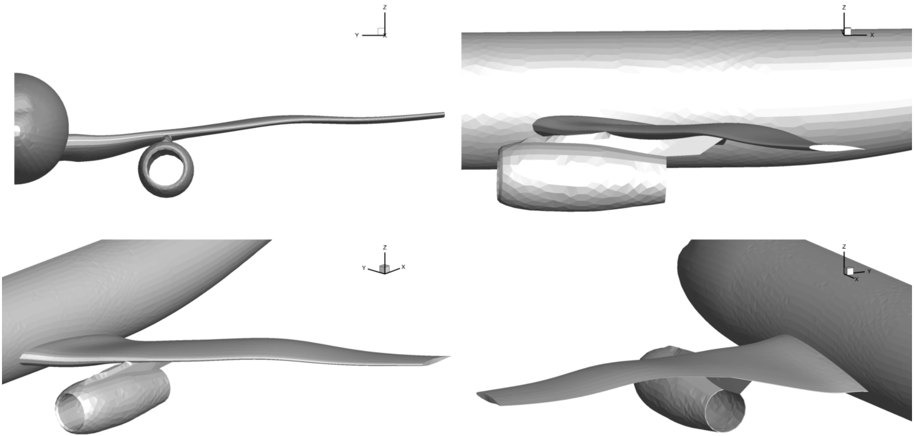
4th Eigenvector

Resulting eigenvectors



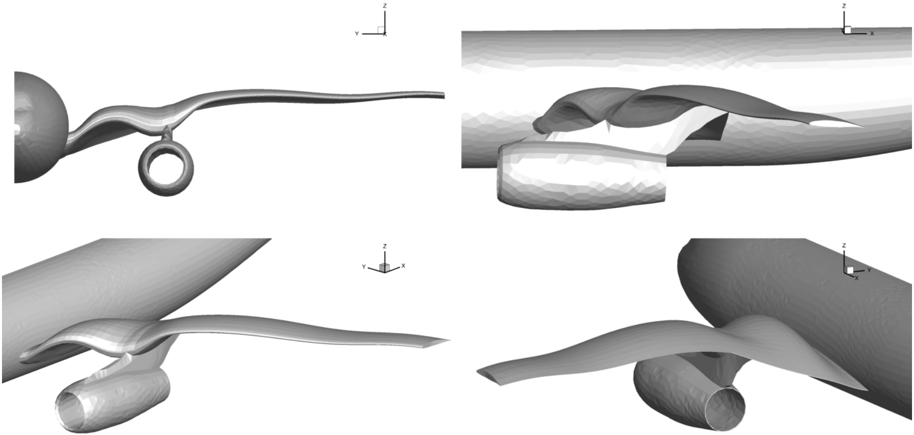
5th Eigenvector

Resulting eigenvectors



6th Eigenvector

Resulting eigenvectors



7th Eigenvector

Outline

1. Introduction
2. Uncertainty Quantification
   - Method of Moments
   - Sensitivities computation
3. Numerical results
   - Parametrization
   - Uncertainty Quantification
4. Summary

Summary

Conclusions

- Uncertainty Quantification
  - Hessian successfully validated
  - Proposed UQ method works well for presented case
  - Computational cost is always less than pure hessian analysis and KLE providing the same accuracy level
  - Good approximation of objective mean-value
  - Method provides valueable by-products for further UQ investigation

Summary

Future work

- Publication
  - Monte Carlo – large number of simulations
  - Compare results with Active Subspace
- PhD Thesis
  - Implement iterative method for generalized eigenvalue problem
  - Compare results for variance
- Other
  - Application to viscid/turbulent cases
  - Implement different parametrizations (e.g. elastic/Laplace)

Acknowledgments

- Majority of this work was done in FP7 project UMRIDA – Uncertainty Management for Robust Industrial Design in Aeronautics

UNCERTAINTY MANAGEMENT FOR ROBUST INDUSTRIAL DESIGN IN AERONAUTICS
UMRIDA

- This work has been supported by the European Union in the framework of European Social Fund through the “Didactic Development Program of the Faculty of Power and Aeronautical Engineering of the Warsaw University of Technology”.

EUROPEAN UNION
EUROPEAN SOCIAL FUND

HUMAN CAPITAL
NATIONAL COHESION STRATEGY

Thank you for your attention!