Surrogate-based multi-objective stator optimization using variation of complex free-form surfaces

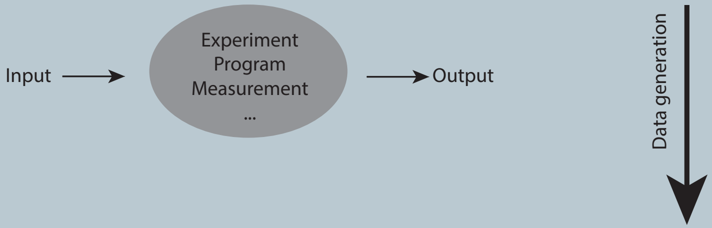
K. Cremanns, A. Graßmann, D. Roos[1], C. Fütterer, J. Palluch[2]

[1]Institute of Modelling and High-Performance Computing, Niederrhein University of Applied Sciences, [2]Friendship Systems AG

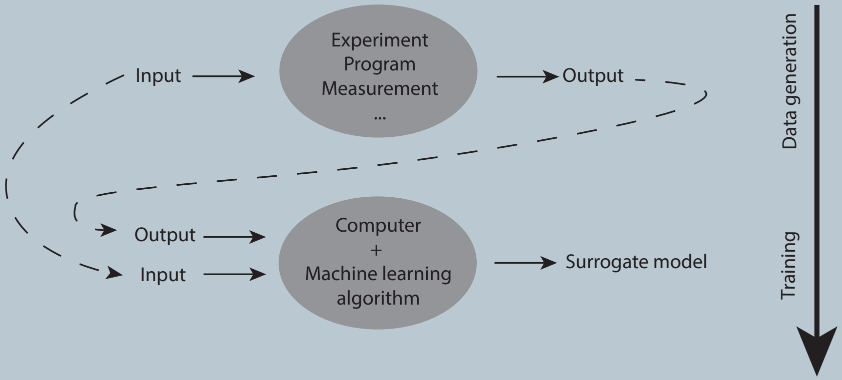
Hochschule Niederrhein
University of Applied Sciences
IMH
Institut für Modellbildung und Hochleistungsrechnen
Institute of Modelling and High-Performance Computing

Outline

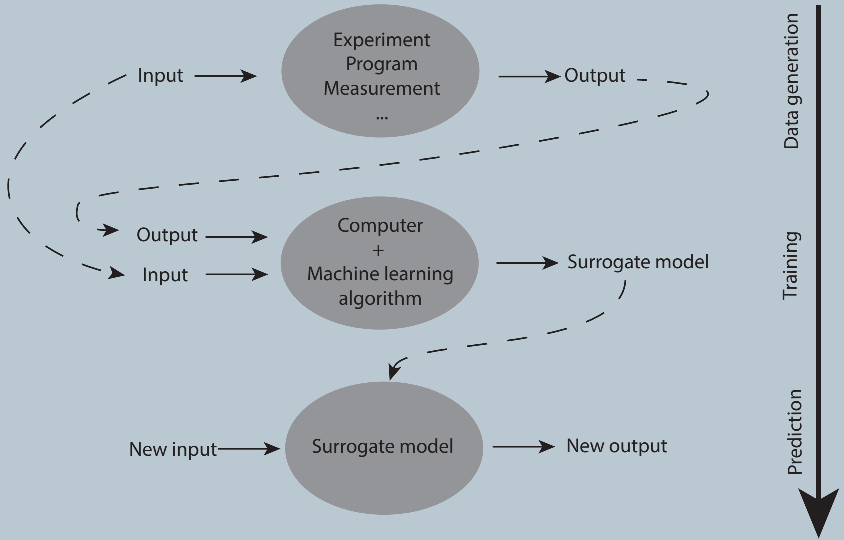
Definition of machine learning (supervised)



Definition of machine learning (supervised)

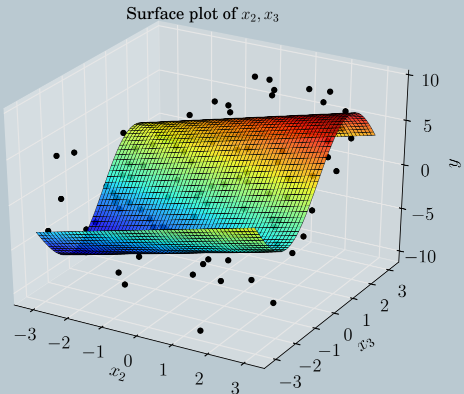


Definition of machine learning (supervised)



Advantages

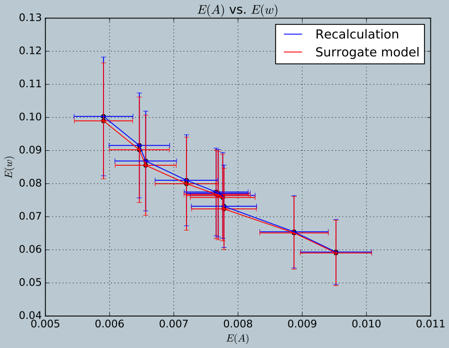
- Fast and cheap model evaluations
- Deeper knowledge of your data (sensitivity analysis)
- Connection of tools without interface problems
- Forecast the future (time dependent problems)
- Expensive design optimization, robustness evaluation, ... becomes possible



Examples of machine learning

In the field of mechanical engineering:

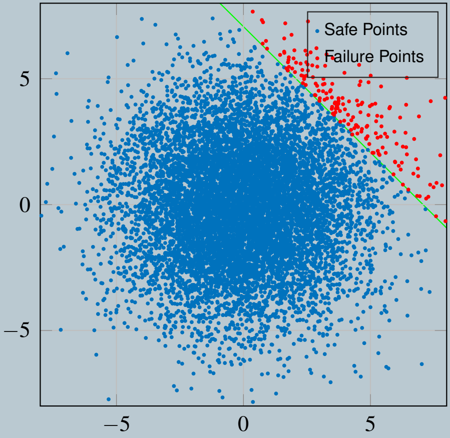
- (Robust) Design optimization



Examples of machine learning

In the field of mechanical engineering:

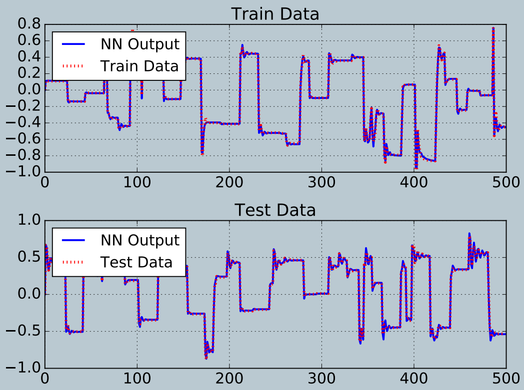
- (Robust) Design optimization
- Robustness / reliability analysis



Examples of machine learning

In the field of mechanical engineering:

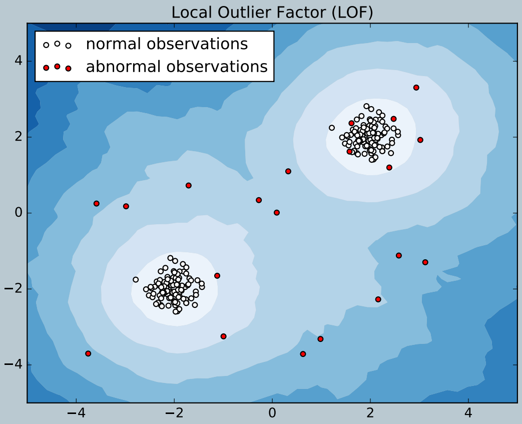
- (Robust) Design optimization
- Robustness / reliability analysis
- Time series forecasting



Examples of machine learning

In the field of mechanical engineering:

- (Robust) Design optimization
- Robustness / reliability analysis
- Time series forecasting
- Classification / anomaly detection



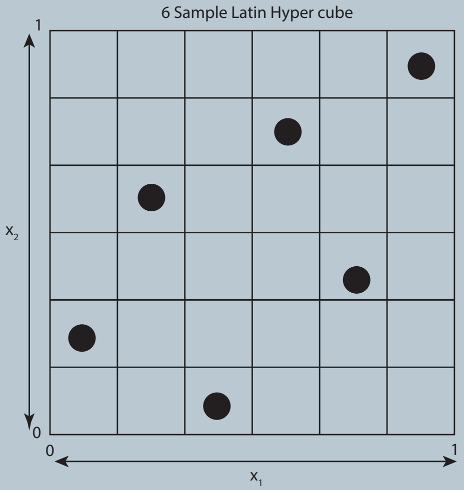
Examples of machine learning

In the field of mechanical engineering:

- (Robust) Design optimization
- Robustness / reliability analysis
- Time series forecasting
- Classification / anomaly detection
- ...

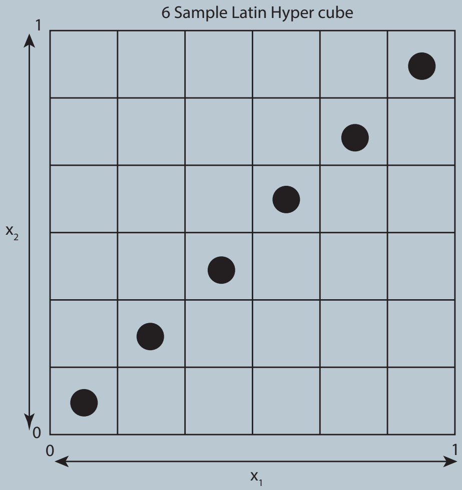
Design of Experiment - Latin hypercube sampling

- Latin Hypercube sampling (LHS) is one of the most used sampling methods for design plans...

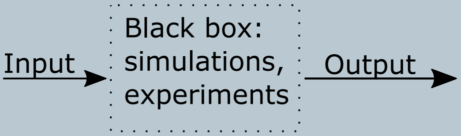


Design of Experiment - Latin hypercube sampling

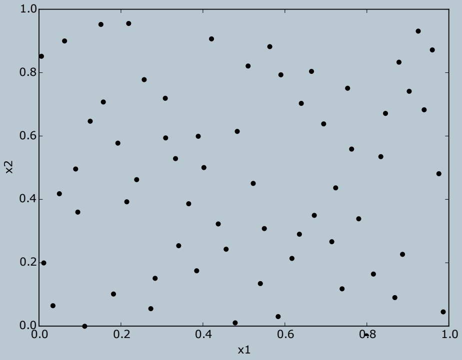
- Latin Hypercube sampling (LHS) is one of the most used sampling methods for design plans...
- but it can also be very useless if it is not optimized



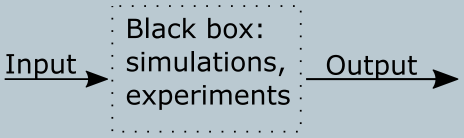
Optimized Latin hypercube sampling



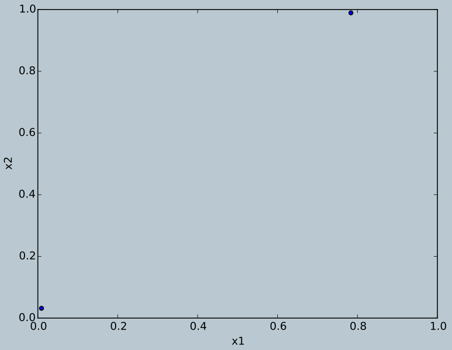
- Space filling
- No unwanted correlation
- One shot / Sequential

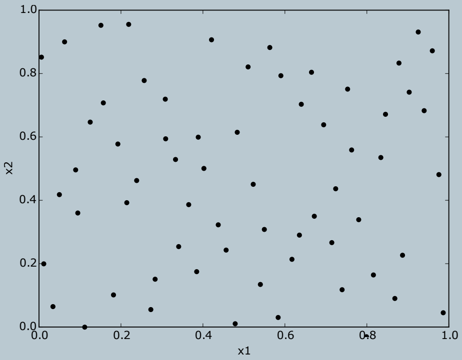


Optimized Latin hypercube sampling

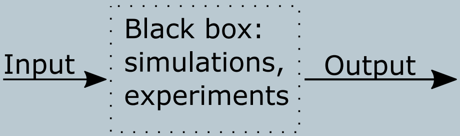


- Space filling
- No unwanted correlation
- One shot / Sequential

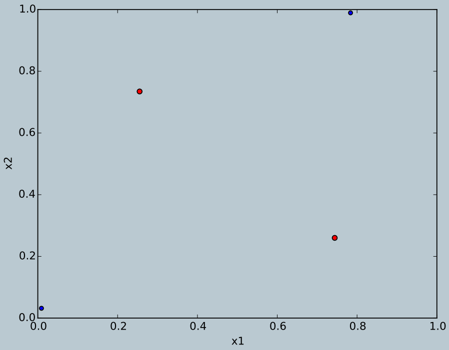


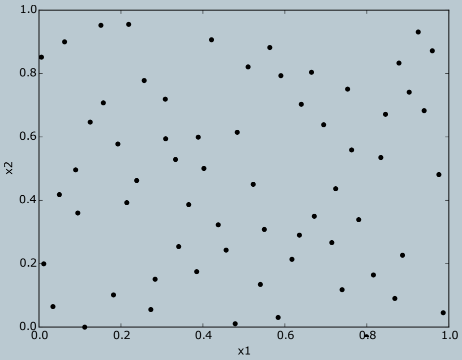


Optimized Latin hypercube sampling



- Space filling
- No unwanted correlation
- One shot / Sequential

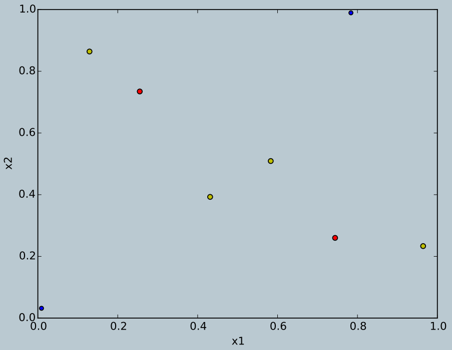


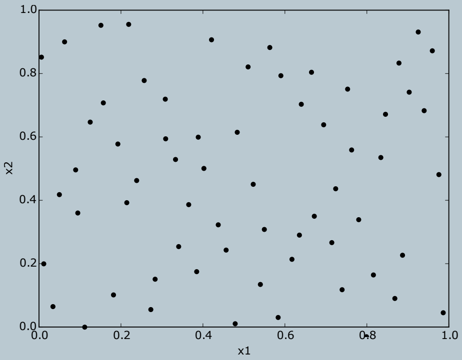


Optimized Latin hypercube sampling

Input → Black box: simulations, experiments → Output

- Space filling
- No unwanted correlation
- One shot / Sequential

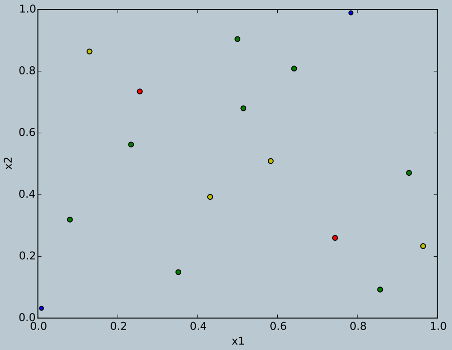


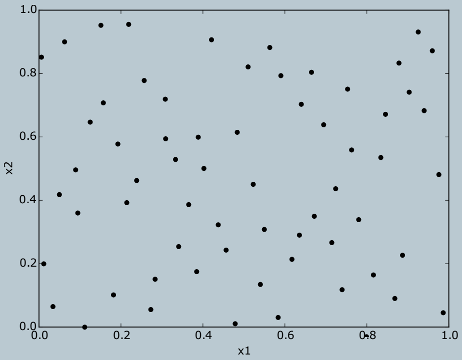


Optimized Latin hypercube sampling

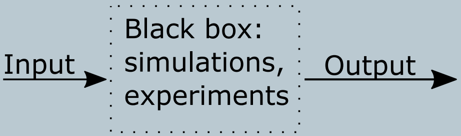
Input → Black box: simulations, experiments → Output

- Space filling
- No unwanted correlation
- One shot / Sequential

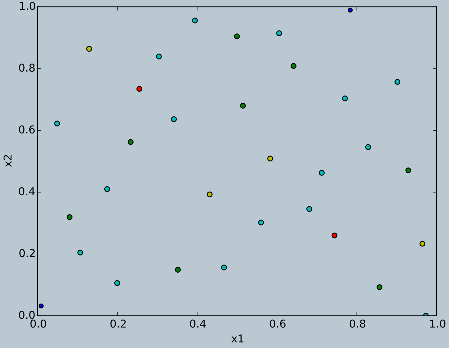


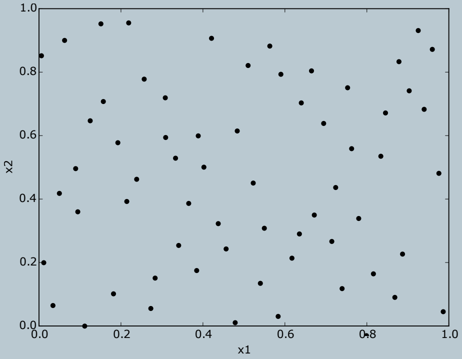


Optimized Latin hypercube sampling



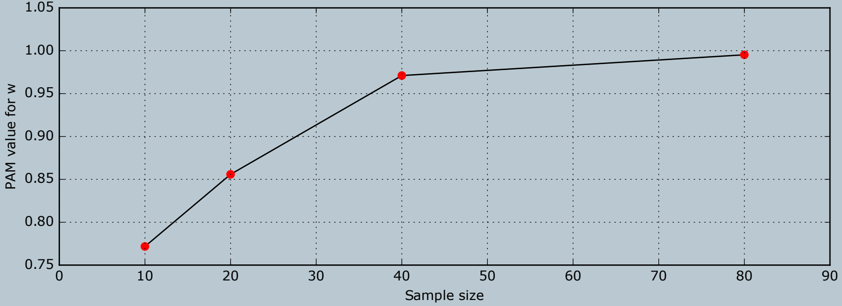
- Space filling
- No unwanted correlation
- One shot / Sequential





Stepwise convergence check

Sequential sampling can be used for convergence check of surrogate model

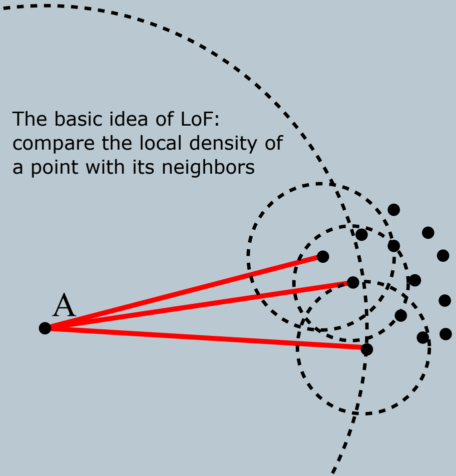


No wasted samples

Automatic outlier detection

- Local outlier factor (LoF). Density based outlier detection.
- It detects outliers in data with areas of different density.
- Just one outlier can have a huge impact on the prognosis quality.
- Searches for each output individually for outliers.
- LoF $\gg 1$ outlier.

The basic idea of LoF: compare the local density of a point with its neighbors



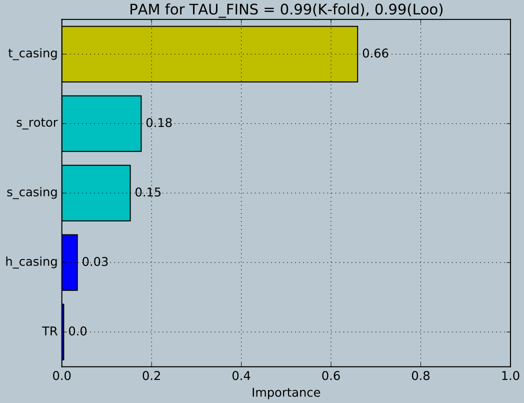
Variance based sensitivity analysis

First order:

- $S_i = \frac{V_i}{Var(y)}$

Total order:

- $S_{T_i} = \frac{E_{x_{\sim i}}(Var_{x_i}(y|x_{\sim i}))}{Var(y)}$
- Sum of $S_{Ti} > 1$ if interaction effects between x, y are present.
- Non-linear, non-monotonic, multivariate sensitivities.
- Useful for root cause analysis



Deep Gaussian covariance networks - Advantages

- Any training samples size is possible even on low hardware resources, since flexible batch training can be used.

Deep Gaussian covariance networks - Advantages

- Any training samples size is possible even on low hardware resources, since flexible batch training can be used.
- Gaussian process based confidence intervals (measurement of uncertainty) of prediction.

Deep Gaussian covariance networks - Advantages

- Any training samples size is possible even on low hardware resources, since flexible batch training can be used.
- Gaussian process based confidence intervals (measurement of uncertainty) of prediction.
- A trained model can be updated with new trainings points without retrain the whole model (online-learning).

Deep Gaussian covariance networks - Advantages

- Any training samples size is possible even on low hardware resources, since flexible batch training can be used.
- Gaussian process based confidence intervals (measurement of uncertainty) of prediction.
- A trained model can be updated with new trainings points without retrain the whole model (online-learning).
- It is very useful for adapted or sparse Input and different noise levels like in multi-fidelity data.

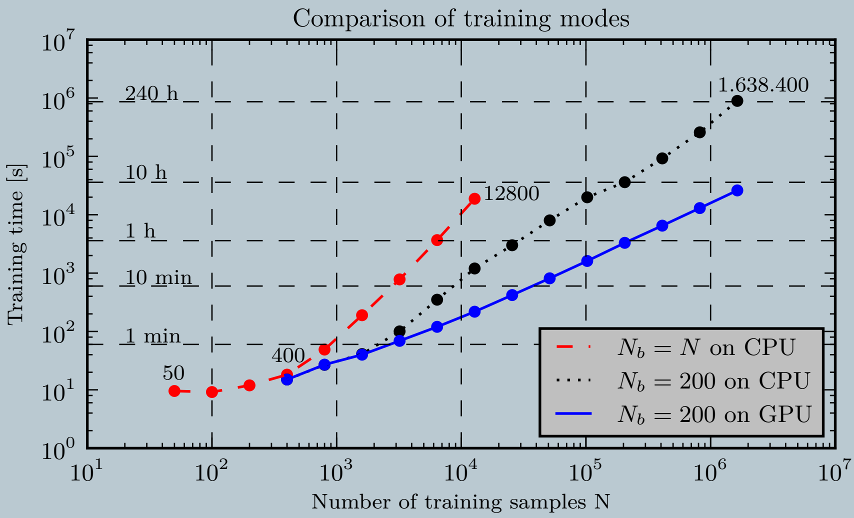
Deep Gaussian covariance networks - Advantages

- Any training samples size is possible even on low hardware resources, since flexible batch training can be used.
- Gaussian process based confidence intervals (measurement of uncertainty) of prediction.
- A trained model can be updated with new trainings points without retrain the whole model (online-learning).
- It is very useful for adapted or sparse Input and different noise levels like in multi-fidelity data.
- Multiple different Gaussian process covariance functions are used and trained simultaneously. So complex problems can be predicted even better.

Training time comparison (1/2)

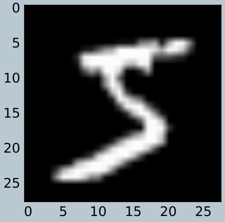
- Test function with 100 input parameters and 1 output.
- Training is based on 100 epochs (reasonable good model in most cases).
- Training samples N range from 50 to 1.638.400 (doubling).
- 13.360 trainable hyperparameters.
- 3 training modes are compared:
  - Batch size = N on CPU
  - Batch size = 200 on CPU
  - Batch size = 200 on GPU
- Training on a HPC (distributed learning) would also be possible.
- Used CPU: Intel Core i7 3770 2 cores at 3.50 GHz.
- Used GPU: Nvidia Quadro 4000 256 Cuda cores (7 years old) modern GPU have 3840 Cuda cores.

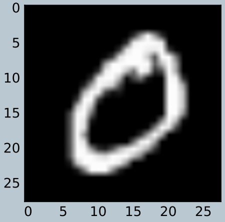
Training time comparison (2/2)

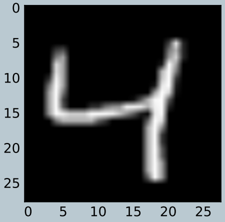


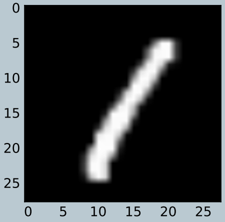
Example - Classification MNIST data set (1/2)

- 60.000 training and 10.000 test points of handwritten digits
- Each data point is defined by an 28X28 pixel image (784 parameters)
- Corresponding number represented by a 10 element vector (0,1,2,3,4,5,6,7,8,9)



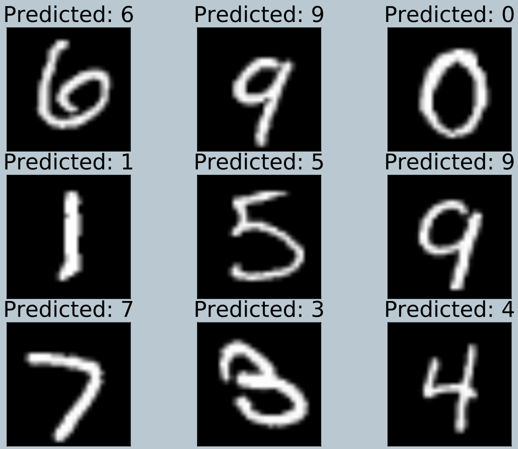






Example - Classification MNIST data set (2/2)

- Training time ~ 10 min
- Error rate $\sim 5\%$



Mode for sequential problems

- The framework also includes a calculation mode for sequential dependent problems.

Mode for sequential problems

- The framework also includes a calculation mode for sequential dependent problems.
- For example time dependent problems, where $x_{t-1}, x_{t-2}, ..., x_{t-n}$ or $y_{t-1}, y_{t-2}, ..., y_{t-n}$ need to be considered.

Mode for sequential problems

- The framework also includes a calculation mode for sequential dependent problems.
- For example time dependent problems, where $x_{t-1}, x_{t-2}, ..., x_{t-n}$ or $y_{t-1}, y_{t-2}, ..., y_{t-n}$ need to be considered.
- In this case the framework even learns the hyper parameters not only dependent on X but also over the time t.

Validation methods

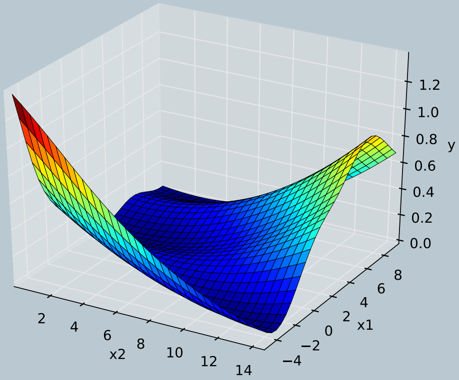
- To estimate the prognosis quality of the surrogate model, R^2_{pred} calculated via cross validation:

$$R^2_{pred} = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - E(y_i))^2}$$

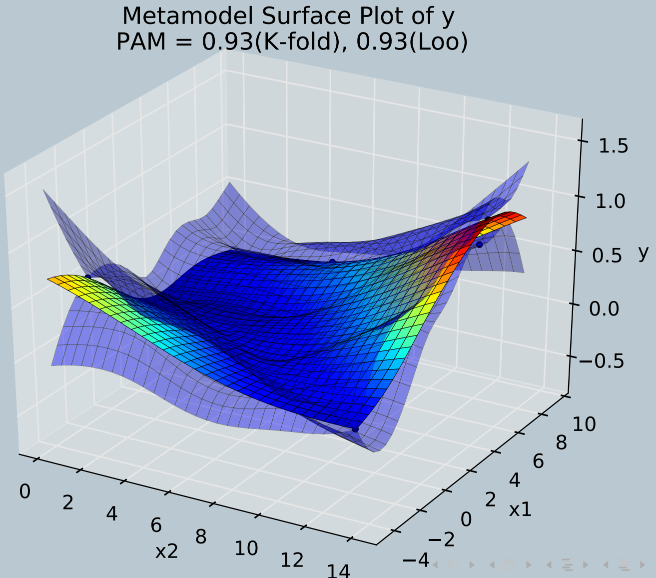
- Predicted vs. observed plot.
- Visual check of the model for example surface plot.
- Prediction of further test data.
- Uncertainty estimation of the prediction.

Confidence intervals

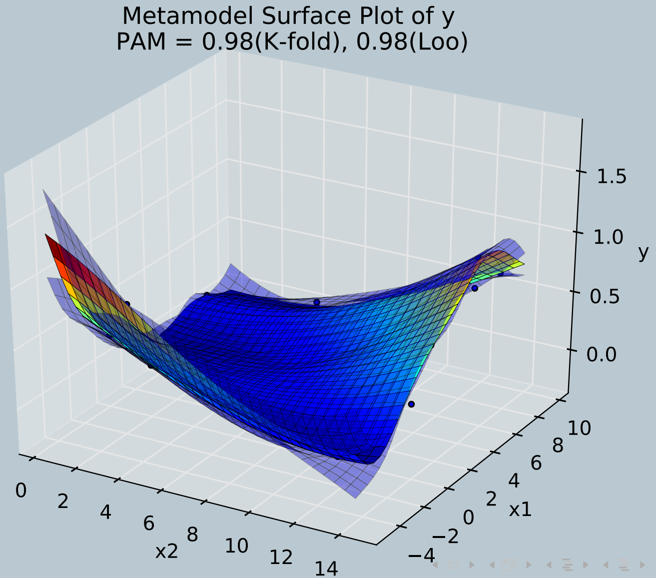
- The framework provides the opportunity to give the confidence interval of its prediction.
- Model uncertainties can be visualized.
- Useful for sampling adaption strategies and surrogate based robust / reliability analysis.
- Example: Branin function



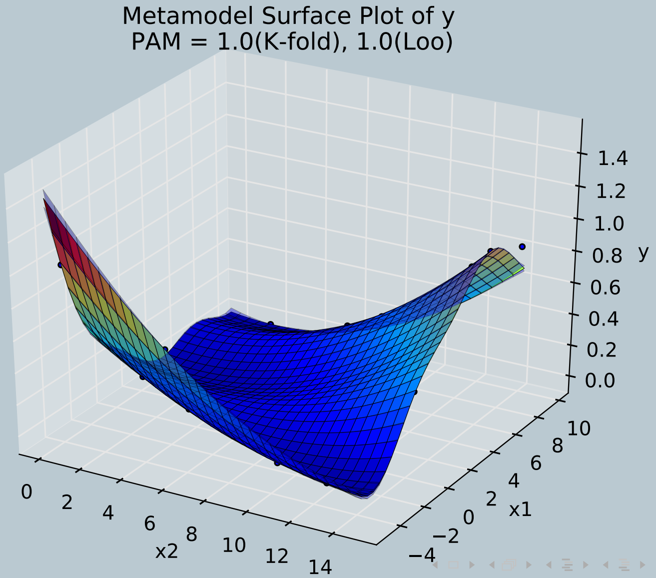
16 samples with 95% confidence intervals (1/3)



32 samples with 95% confidence intervals (2/3)

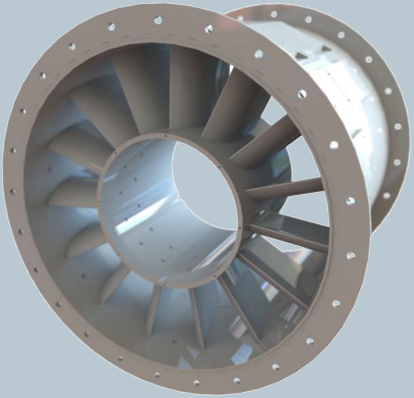


64 samples with 95% confidence intervals (3/3)



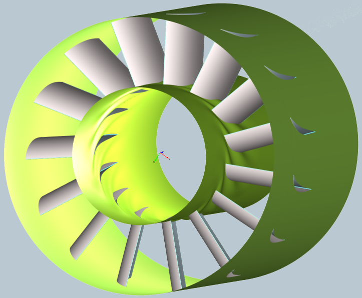
Application

- The "‘TurboLab Stator"’ is a stator in a measurement rig at the TU Berlin.
- An initial stator geometry has been designed based on a representative stator geometry.
- The task is to reduce the total pressure loss and to minimize the flow angle deviation at the outlet over an incidence range.
- 3 operation points (inlet flow angle varies $\pm 5°$).



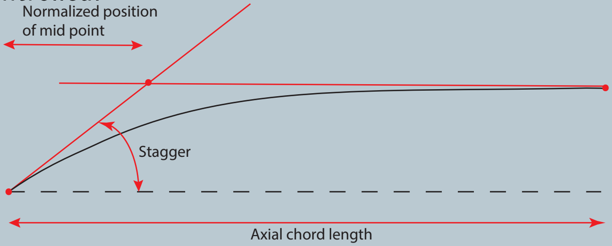
Parametric model

- CAD tool CAESES.
- Overall 73 input parameters:
  - 5 profile sections.
  - 28 main parameters (stagger angle, camberline, thickness).
  - 6 stacking parameters.
  - 31 tuning parameters (thickness tuning).
  - 8 endwall contouring parameters (amplitude, position).



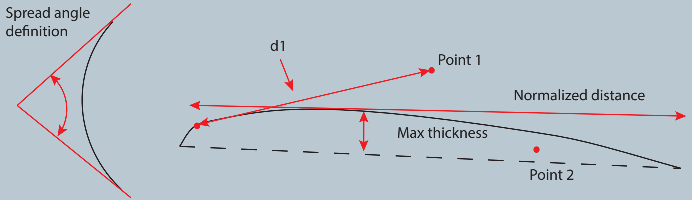
Main parameters - Camberline / Stagger angle

- NURBS curve connects three points.
- Weight of the second point can be controlled.



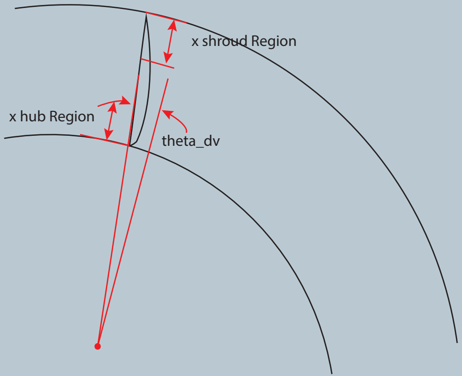
Main parameters - Thickness

- LE, TE is given by radius , ellipse factor and spread angle.
- With LE angle the point 1 is created.
- With TE angle and the parameter TE Shape point 2 is created.
- NURBS curve connects LE with TE including point 1 and point 2.
- Distance d1 to point 1 is optimized internally to get a maximum thickness.



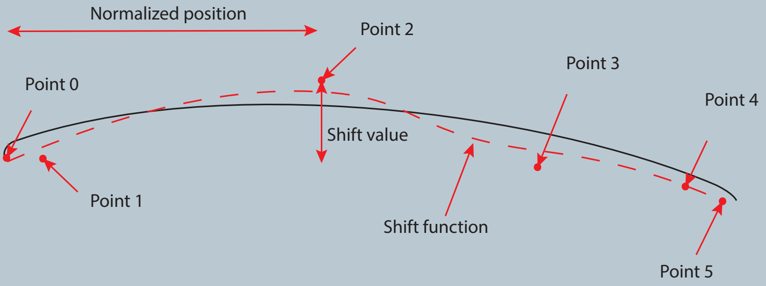
Stacking parameters

- Stacking axis is controlled by 4 points, which can be varied by its:
    - theta angle.
    - the distance of the mid points from inner and outer radii.



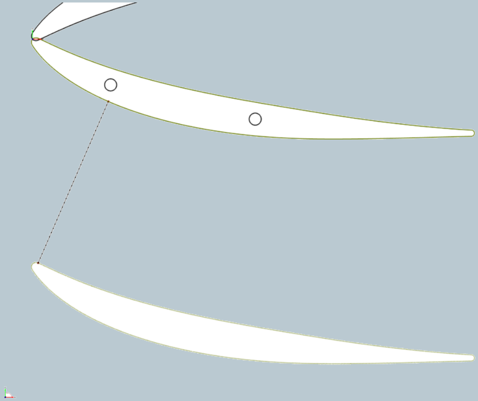
Tuning parameters

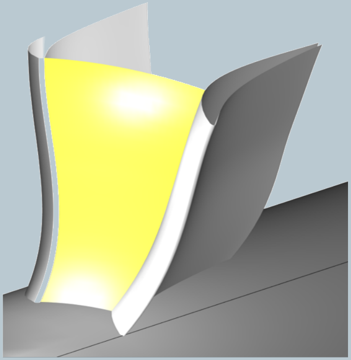
- Shift function can shift the thickness distribution in x and y direction.



Throat optimization

- In order to place the throat at the leading edge of the profile, the spread angle of the leading edge is optimized internally.





Endwall contouring - Trigonometric approach

- Crosssection of endwall for one blade passage is represented by sine function, controlled by:
  - Frequency.
  - Amplitude.
  - Phase shift.
- Functions control how these parameters change in streamwise direction.

CFD setup / Constraints

- CFX solver.
- Meshing in ICEM.
- Spalart-Allmaras turbulence model.
- Scalable wall function.
- 3 operation points with $37°$, $42°$, $47°$ inlet whirl angle.
- Specific heat coefficient of 1.4.
- Inlet total pressure: 102713.0 Pa.
- Inlet total temperature: 294.314 K.
- Inlet pitch angle: $0°$.
- Inlet turbulence intensity: 4%.
- Outlet massflow of 9.0 kg/s (full annulus).

Process chain

CAESES

- Creates geometry
- Export .ttin file for ICEM
- Controls DoE work-flow
- Check manufacturing contraints.

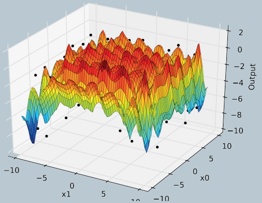
Design →

ANSYS

- ICEM creates mesh
- 3 load cases in CFX
- Calculate objectives

Data for
surrogate model

Input



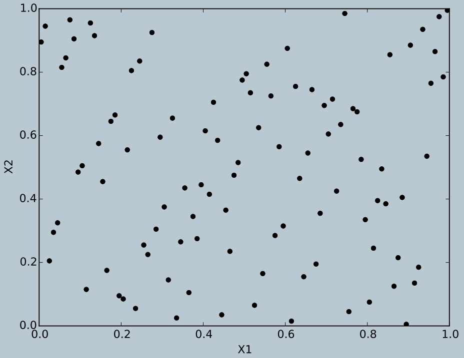
Output

Optimizer

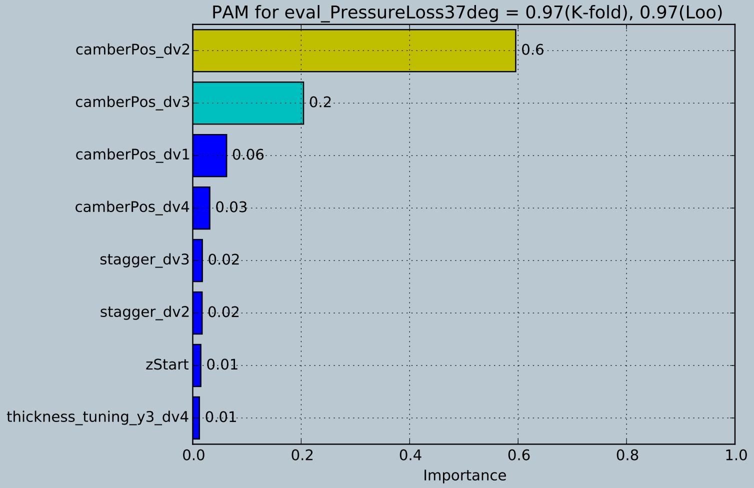
→ Optimal design

Design of experiment

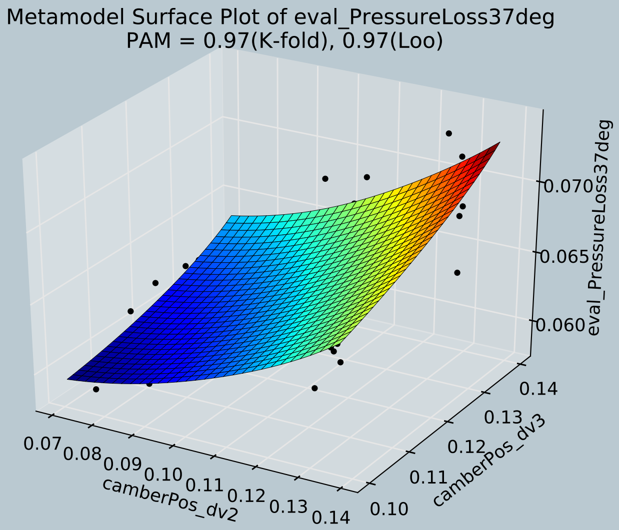
- Pre studies within CAESES to check geometry stability (no calculations on CFX).
- Starting around +-20% of the initial blade geometry for the parameters (if possible).
- Final DoE with overall 80 designs (73 input parameters) calculations with optimized LHS.



Sensitivity - pressure loss $37°$



Sensitivity - pressure loss 37°



Optimization objective

Two optimization criteria, leading to a multi-objective optimization problem. In addition three operating points have to be considered:

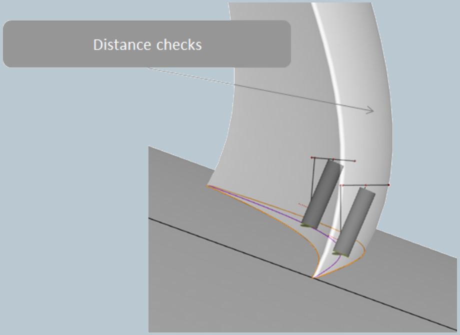
- Minimize the total pressure loss between inlet and outlet under the constraint of keeping the mass flow at 9.0 +/- 0.1 kg/s (full annulus). The total pressure loss is defined as $loss = \frac{p_{total_{in}} - p_{total_{out}}}{p_{total_{in}} - p_{static_{in}}}$.
- Minimize the flow angle deviation at the CFD outlet from the axial direction.
- The inlet whirl angle is allowed to vary by $+-5^{\circ}$. Thus three operating points have to be considered.

Optimization constraints - Manufacturing constraints

- The number of blades is fixed to $n = 15$.
- The axial chord of the blade has to be kept constant.
- The minimum value for leading and trailing edge radius is 1mm.
- The two holes for the fixture in the middle of the blade have a radius of 2.5 mm and a depth of 20 mm. The blade thickness at these positions has to accommodate a cylinder of material with a radius of 5 mm and a depth of 20mm. The two holes have to be at least 60mm apart from each other.
- The blade has to be mountable on a plate of dimensions 200mm x 80 mm.
- The reduction of radius due to the hub contouring has to be $<= 5$mm and the increase of the radius due to the hub contouring has to stay below 10mm.

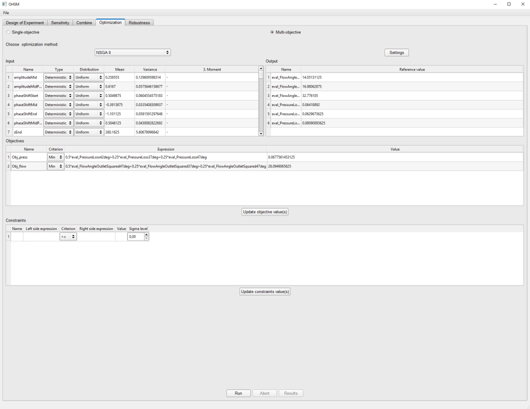
Constraints - Drilling clearances - drilling angle

- After the position is obtained, the drilling angle will be optimized, to avoid an intersection of the drilling with the blade surface.
- Drilling angle is the difference angle to the normal, applied around the z-axis.

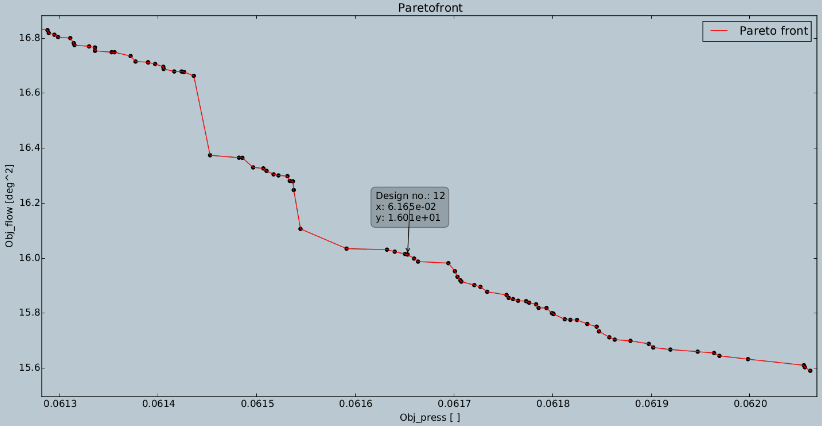


Optimization on surrogate model

- NSGA II optimization algorithm.
- 300 generations, with 50 population size.
- 15000 overall design evaluations on surrogate model.
- 5 minutes on surrogate model against 625 days on CFX (32 cores per operation point).



Multi-objective Pareto-front



Results

Recalculation chosen design

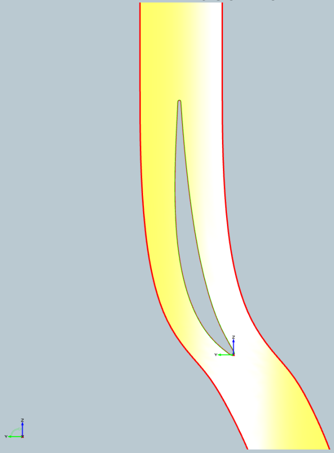
	Design point 42°	Off-design 37°	Off-design 47°
Total pressure loss (%)	6.09/6.16	5.95/6.13	7.05/7.39
Rel. error (%)	1.15	3.02	4.82
Exit whirl angle (deg^2)	14.98/15.44	12.63/12.95	27.28/27.78
Rel. error (%)	3.07	2.53	1.83

Baseline vs. optimum objective function results

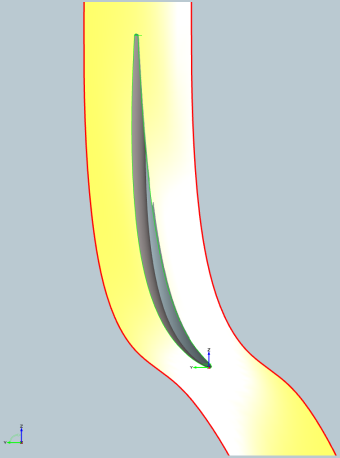
	Design point 42°	Off-design 37°	Off-design 47°
Improvement TPL(%)	5.23	2.85	5.98
Improvement EWA(%)	30.57	19.31	28.27

Comparison: blade geometry - hub

Baseline

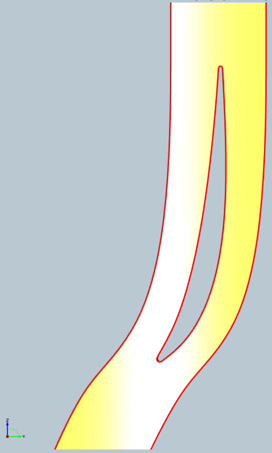


Optimum

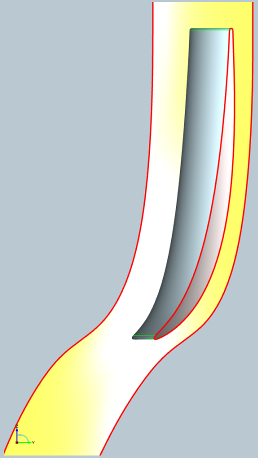


Comparison: blade geometry - shroud

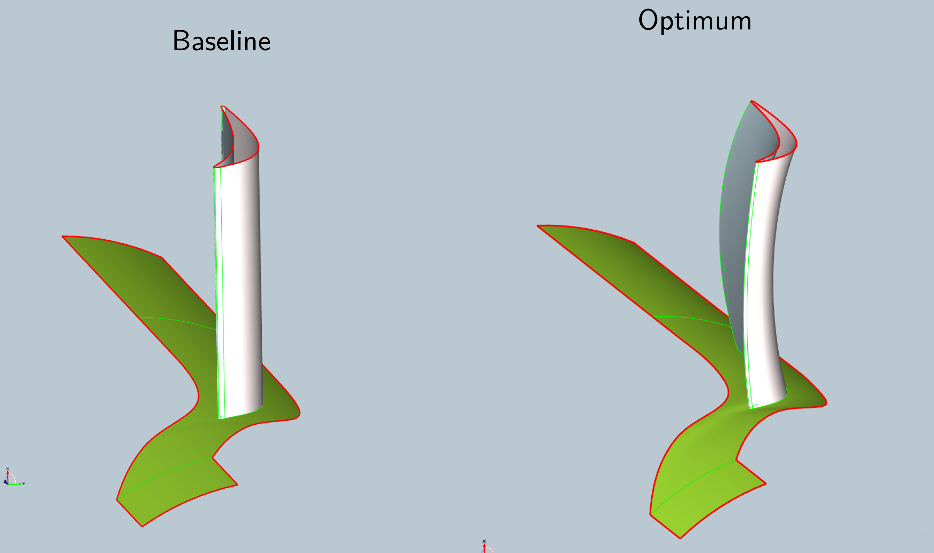
Baseline



Optimum

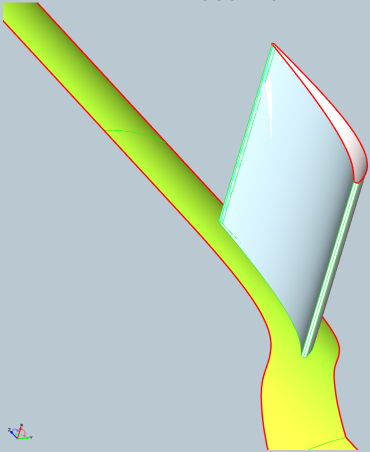


Comparison: blade geometry - front

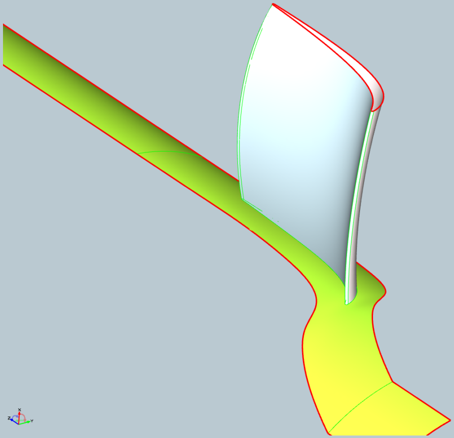


Comparison: blade geometry -side

Baseline

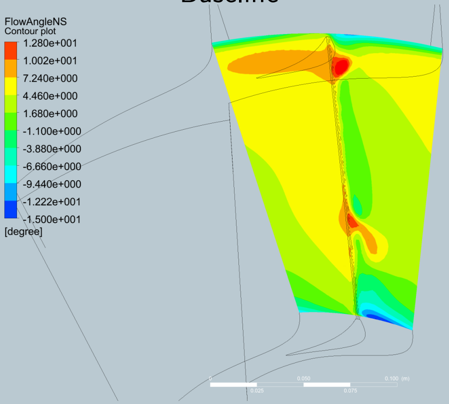


Optimum

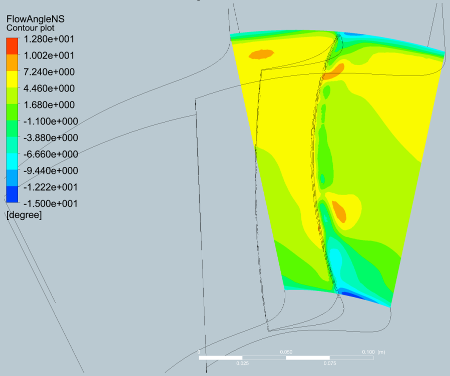


Comparison: 2D field axial flow angle

Baseline



Optimum



Summary

- Deep Gaussian covariance networks are capable to handle different type of problems with arbitrary size of data, with fast training times and high prognosis quality.
- Automatic variable selection and deeper knowledge of your data through sensitivity analysis.
- Multi-objective optimization application TU-Berlin stator with only 80 designs for 73 input parameters and 3 operation points.
- Good accuracy of surrogate model and efficient improvement for both objective functions of 4.86% for the pressure loss and 27.85% for the axial flow deviation.