Workshop on HPC enabling of OpenFOAM for CFD applications

CINECA (Casalecchio di Reno, BO, Italy). March 25th-27th, 2015.

An open-source framework for multi-physics simulations and optimization

wolf dynamics

UNIVERSITÀ DEGLI STUDI DI GENOVA

DLTM
DISTRETTO LIGURE delle TECNOLOGIE MARINE

Presented by:
Joel GUERRERO
joel.guerrero@unige.it

UNIVERSITÀ DEGLI STUDI DI GENOVA

wolf dynamics

In collaboration with:
Giovanni BAILARDI
giovanni.bailardi@dltm.it

DLTM
DISTRETTO LIGURE delle TECNOLOGIE MARINE

Roadmap

1. Brief overview of optimization
2. Surrogate based optimization
3. The optimization driver
4. Practical applications
5. Wrap-up

Roadmap

1. **Brief overview of optimization**
2. Surrogate based optimization
3. The optimization driver
4. Practical applications
5. Wrap-up

Brief overview of optimization

What is optimization?

- In plain English, optimization is the act of obtaining the best result under given circumstances.

- This applies to any field (finance, health, construction, operations, manufacturing, transportation, engineering design, sales, public services, mail, and so on).

- The ultimate goal is either to minimize, maximize or zeroed a quantity of interest (QoI).

Brief overview of optimization

What is optimization?

- Mathematically speaking, an optimization problem can be stated as follows,

Find $\mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$ which minimizes / maximizes / zeroed $f_j(\mathbf{X}), \quad j = 1, 2, \ldots, q$

Subject to the following constraints (linear and/or non-linear)

$$g_j(\mathbf{X}) \leq 0, \quad j = 1, 2, \ldots, m$$

$$l_j(\mathbf{X}) = 0, \quad j = 1, 2, \ldots, p$$

where $\mathbf{X}$ is an *n*-dimensional vector called the design vector, $f_j(\mathbf{X})$ is the objective function or QoI, and $g_j(\mathbf{X})$ and $l_j(\mathbf{X})$ are known as inequality and equality constraints, respectively.

Brief overview of optimization

Optimization methods

- To find the optimal value we can choose between gradient-based methods and derivative-free methods.
- Gradient-based methods look for improvement based on derivative information.
- Derivative-free methods look for the optimal value using sampling with bias/ rules toward improvement or they do broad exploration with selective exploitation (e.g. genetic algorithms).
- Gradient-based methods will converge to local extremes, while derivative-free methods can find local and global extremes.
- We can also do parametrical studies and design of experiments, this is part of design exploration.
- We can also do surrogate based optimization (SBO).

Brief overview of optimization

Optimization methods

- Optimization can be single-objective or multi-objective.
- In multi-objective optimization (MOO) we are interested in optimizing more than one QoI simultaneously.
- The QoI’s can be competitive or opposing.
- The final goal in MOO is to find a representative set of optimal solutions (Pareto front), quantify the trade-offs, and finding a single or set of solutions that satisfy the subjective preferences of a human decision maker.

Choosing an optimization method

- **Unconstrained or bound-constrained problems**
  - Smooth and cheap: any method is suitable, however gradient-based methods will be the fastest.
  - Smooth and expensive: gradient-based methods.
  - Non-smooth and cheap: derivative-free methods or surrogate-based optimization.
  - Nonsmooth and expensive: surrogate-based optimization (SBO).
  - Multi-objective: derivative-free methods or SBO.

- **Nonlinearly-constrained problems**
  - Smooth and cheap: gradient-based methods.
  - Smooth and expensive: gradient-based methods.
  - Nonsmooth and cheap: derivative-free methods or SBO.
  - Nonsmooth and expensive: SBO.
  - Multi-objective: derivative-free methods or SBO.

Roadmap

1. ~~Brief overview of optimization~~
2. **Surrogate based optimization**
3. The optimization driver
4. Practical applications
5. Wrap-up

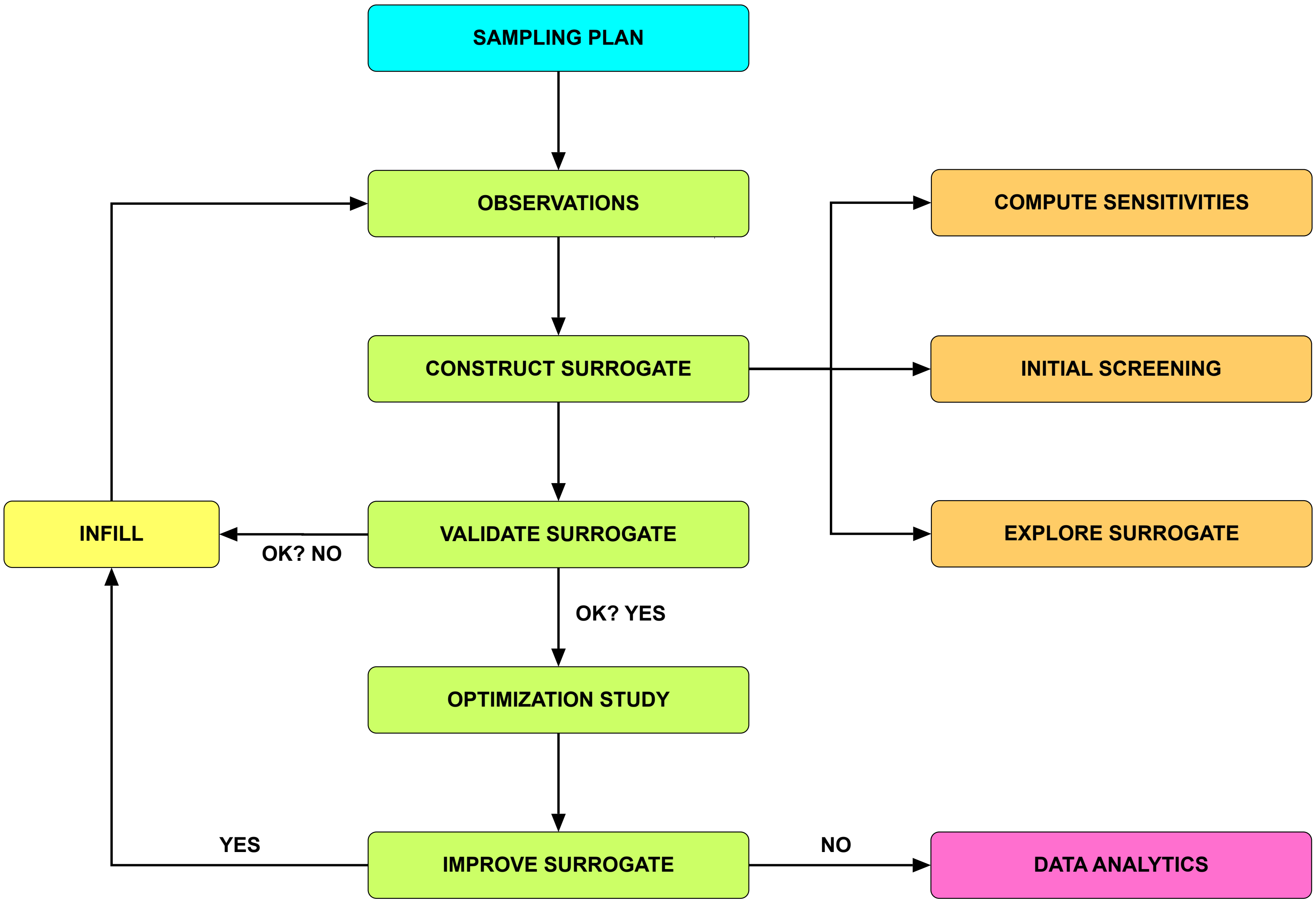
Surrogate based optimization

What is surrogate based optimization (SBO)?

- When we do SBO, we use a surrogate model (also know as meta-model or response surface), to approximate an original high fidelity model (e.g., expensive CFD simulations).
- The surrogate acts as data fit to the observations so that new results can be predicted without recurring to expensive simulations.
- Once the surrogate is built, we can use any kind of optimization or calibration method. Evaluating the QoI at the surrogate level is inexpensive.
- Working at the surrogate level is order of magnitude faster than using high fidelity models.
- Surrogates can be also used with noisy and incomplete data.
- They can be also used for data mining and data analytics.
- In engineering design, surrogates can be used for initial screening and to provide information on the sensitivities of the data.

Surrogate based optimization

SBO workflow



Surrogate based optimization

Let us illustrate the idea behind SBO by using an analytical function: the Branin function

Surrogate based optimization

The Branin function

$$f(x,y) = \left(y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos(x) + 10$$

Subject

$s.t.\ 0 \leq y \leq 15 \qquad s.t.\ -5 \leq x \leq 10$

Global minimum

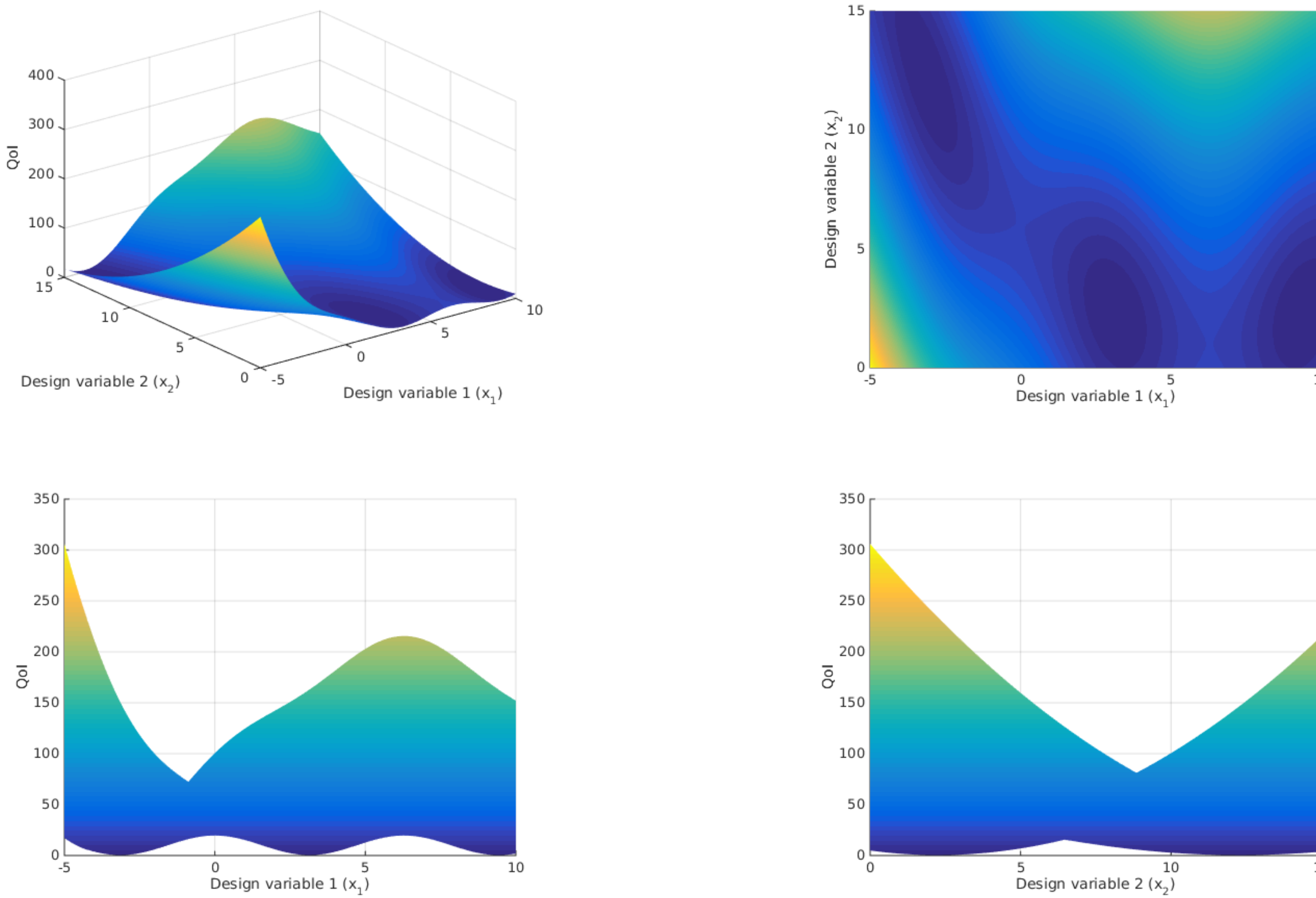
$$f(x,y) = 0.397887$$

$$(x,y) = (-\pi, 12.275), (\pi, 2.275), (9.42478, 2.475)$$

Surrogate based optimization

The Branin function

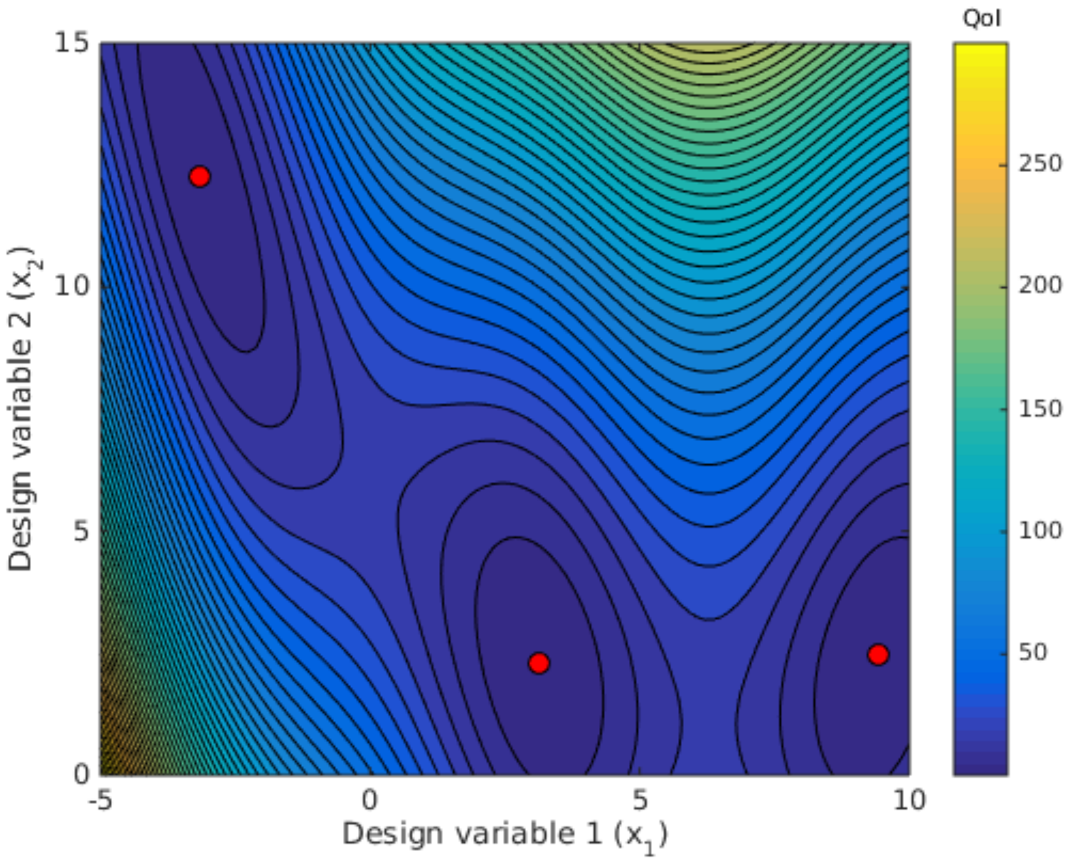
Analytical function – Surface representation



Surrogate based optimization

The Branin function

Analytical function – Contour plot and minimum values



Surrogate based optimization

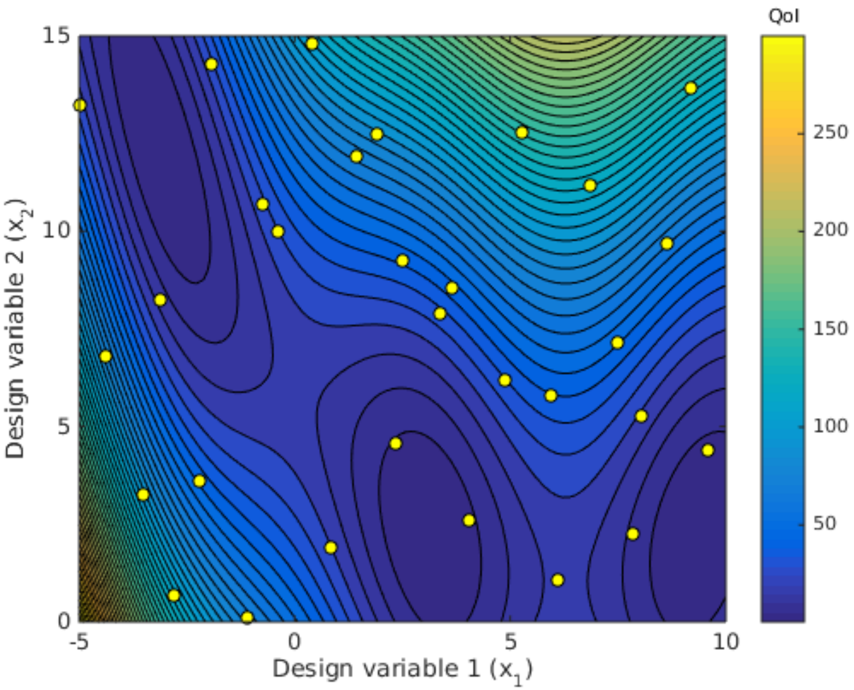
The Branin function

- To perform the SBO, we need to proceed as follows:
    - Design an experiment.
    - Run high fidelity simulations.
    - Built the surrogate. There are many methods, just to name a few: kriging interpolation (Gaussian process), neural networks, radial basis functions, multivariate adaptive regression splines, polynomial functions, least squares and so on.
    - Compute initial sensitivities and do and initial screening.
    - Validate the surrogate.
    - Improve the surrogate. This includes training the surrogate, removing outliers and smoothing the surrogate.
    - Do the optimization at the surrogate level.
    - Visualize to design scenario.

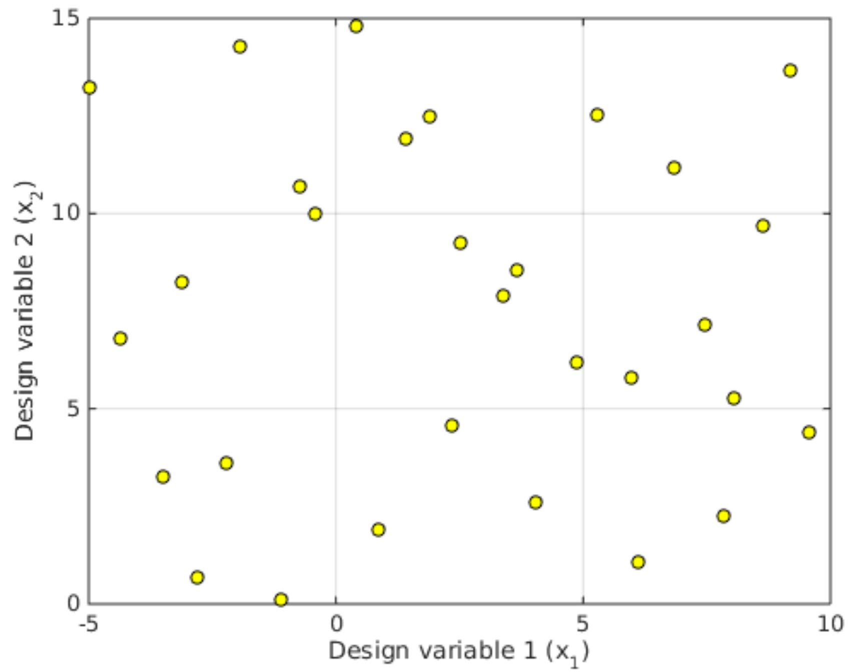
Surrogate based optimization

The Branin function

DACE experiment



Branin function - Analytical

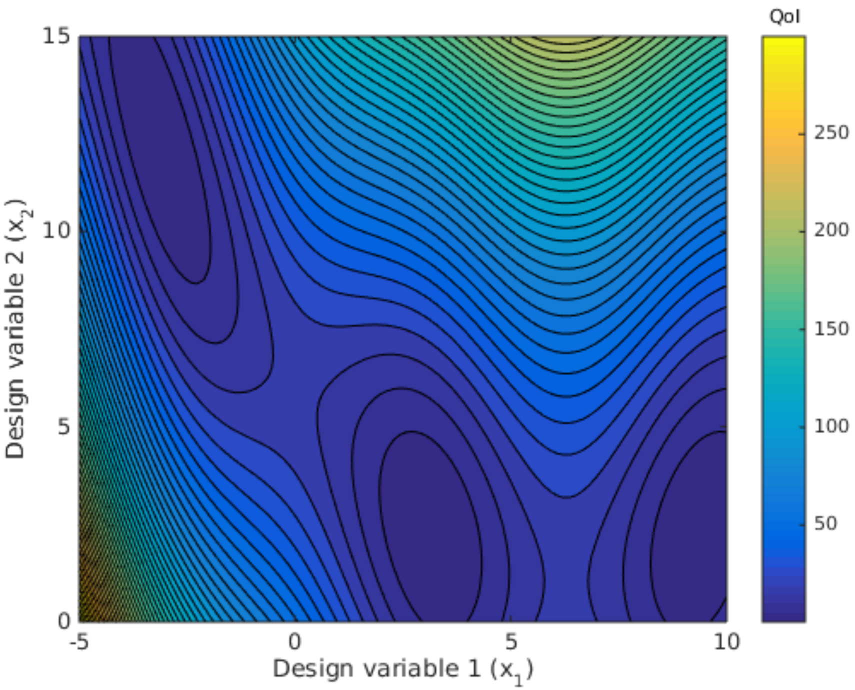


LHS sampling in design space
(30 experiments)

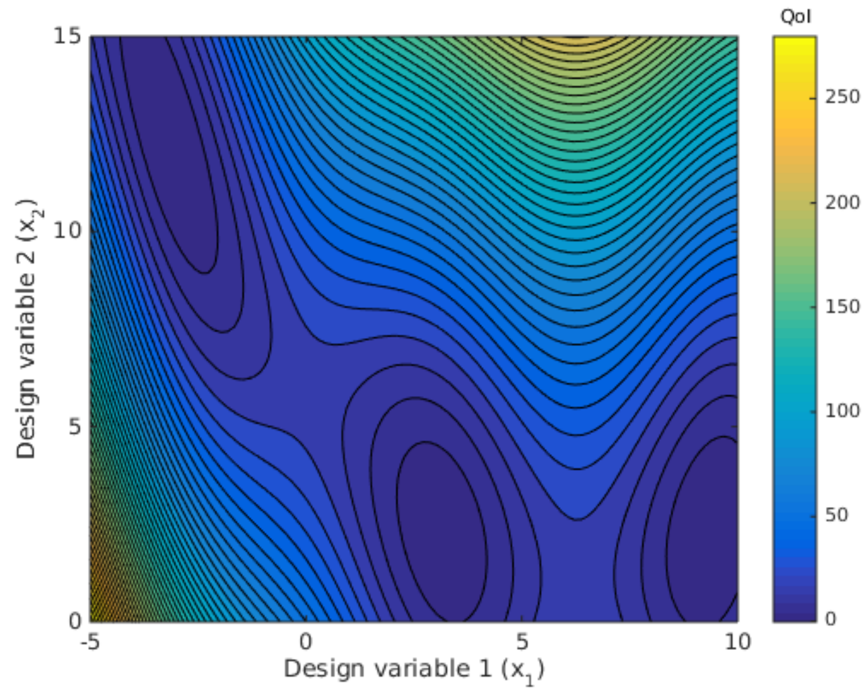
Surrogate based optimization

The Branin function

Surrogate – Kriging interpolation



Branin function - Analytical

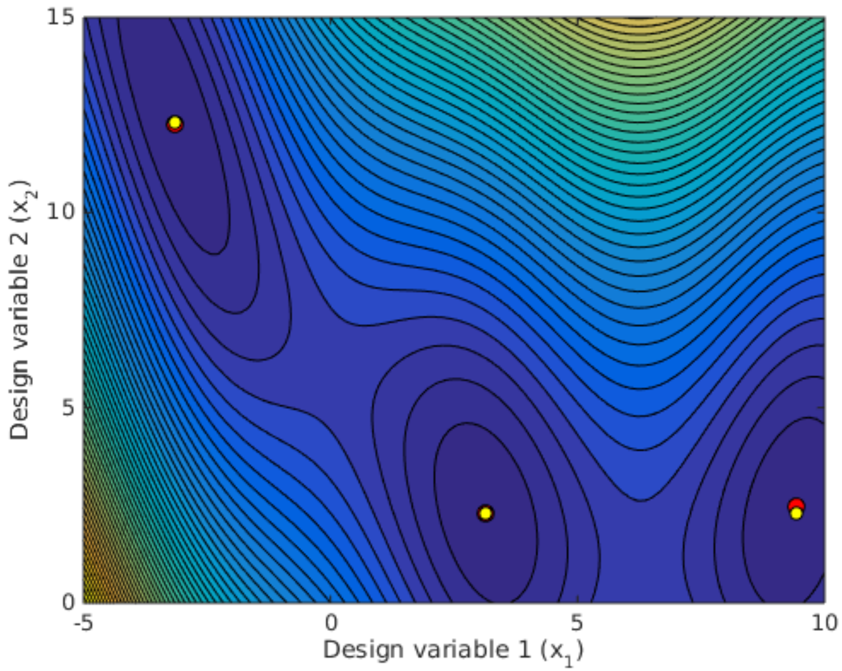


Branin function – Surrogate, meta-model, response surface, you name it.

Surrogate based optimization

The Branin function

Surrogate based optimization at the surrogate level

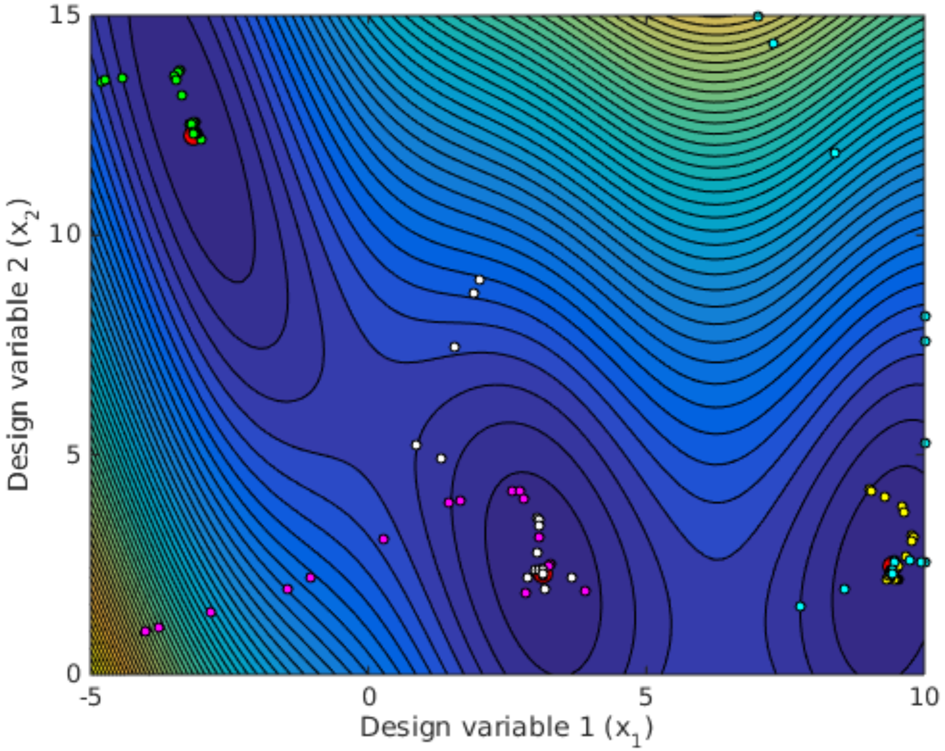


- The red points are global minimum of the analytical Branin function, and the yellow points are the global minimum in the surrogate.
- We conduct constrained gradient based optimization on the surrogate, for this we use the method of feasible directions (MFD), with multiple starting points (multi-start).
- We choose different initial points because we want to increase the possibilities of finding all the minimum.

Surrogate based optimization

The Branin function

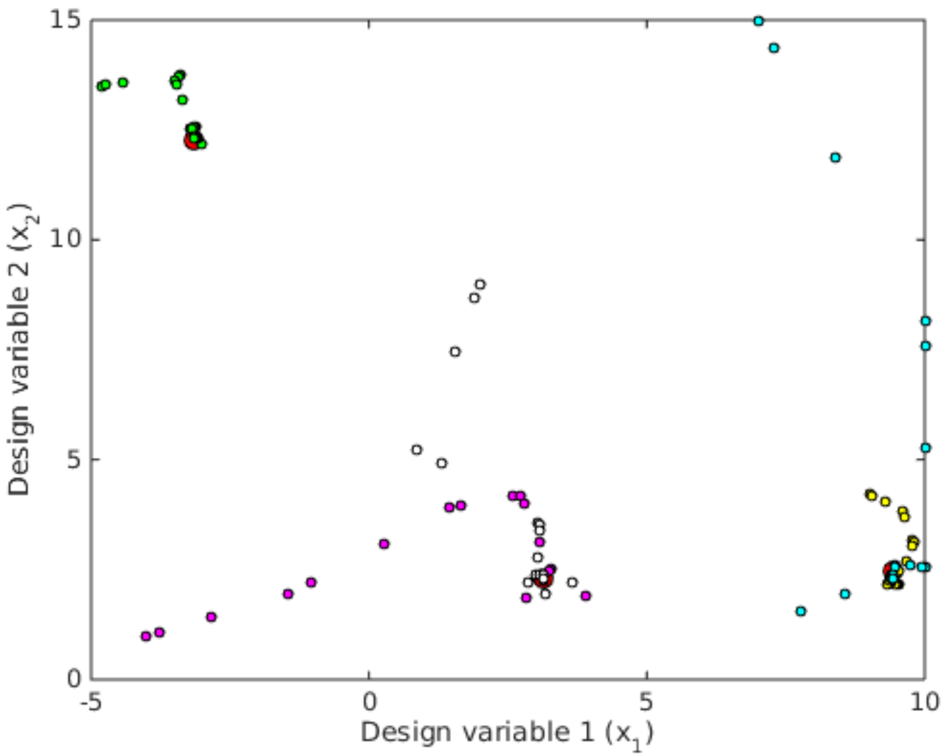
- Surrogate based optimization using the MFD gradient based method
- Surrogate generated using kriging interpolation



Surrogate based optimization

The Branin function

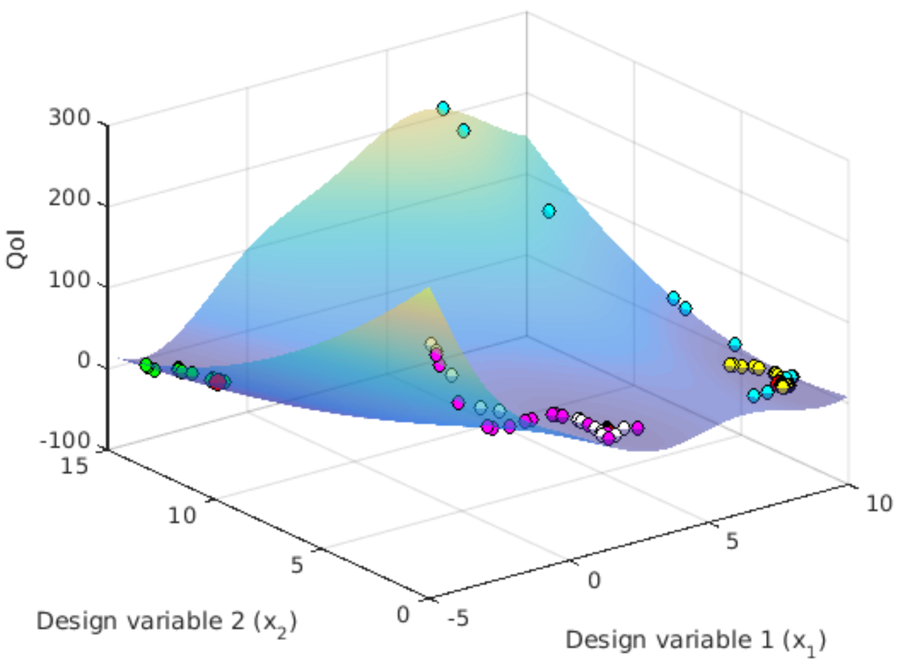
- Surrogate based optimization using the MFD gradient based method
- Surrogate generated using kriging interpolation

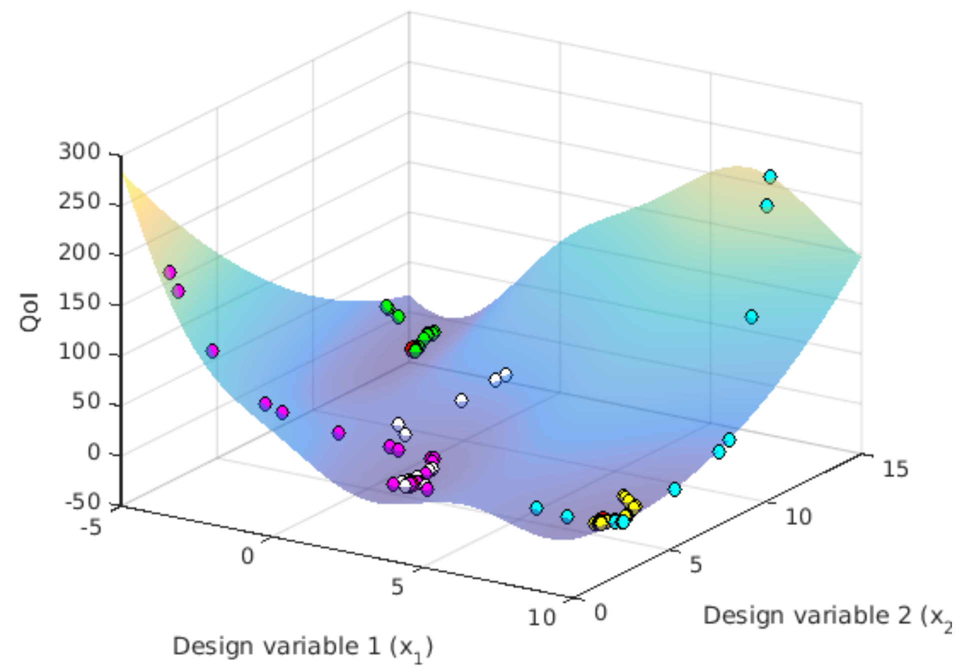


Surrogate based optimization

The Branin function

- Surrogate based optimization using the MFD gradient based method
- Surrogate generated using kriging interpolation



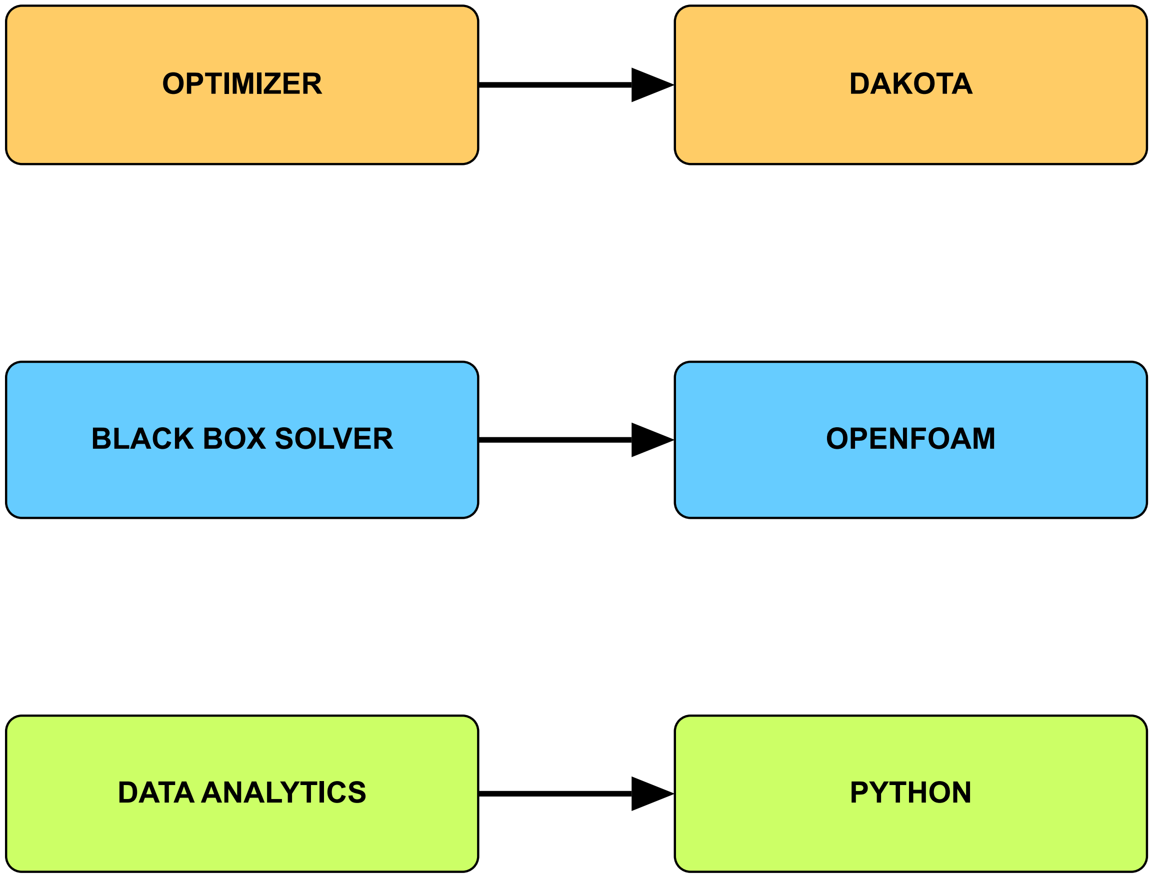


Roadmap

1. ~~Brief overview of optimization~~
2. ~~Surrogate based optimization~~
3. **The optimization driver**
4. Practical applications
5. Wrap-up

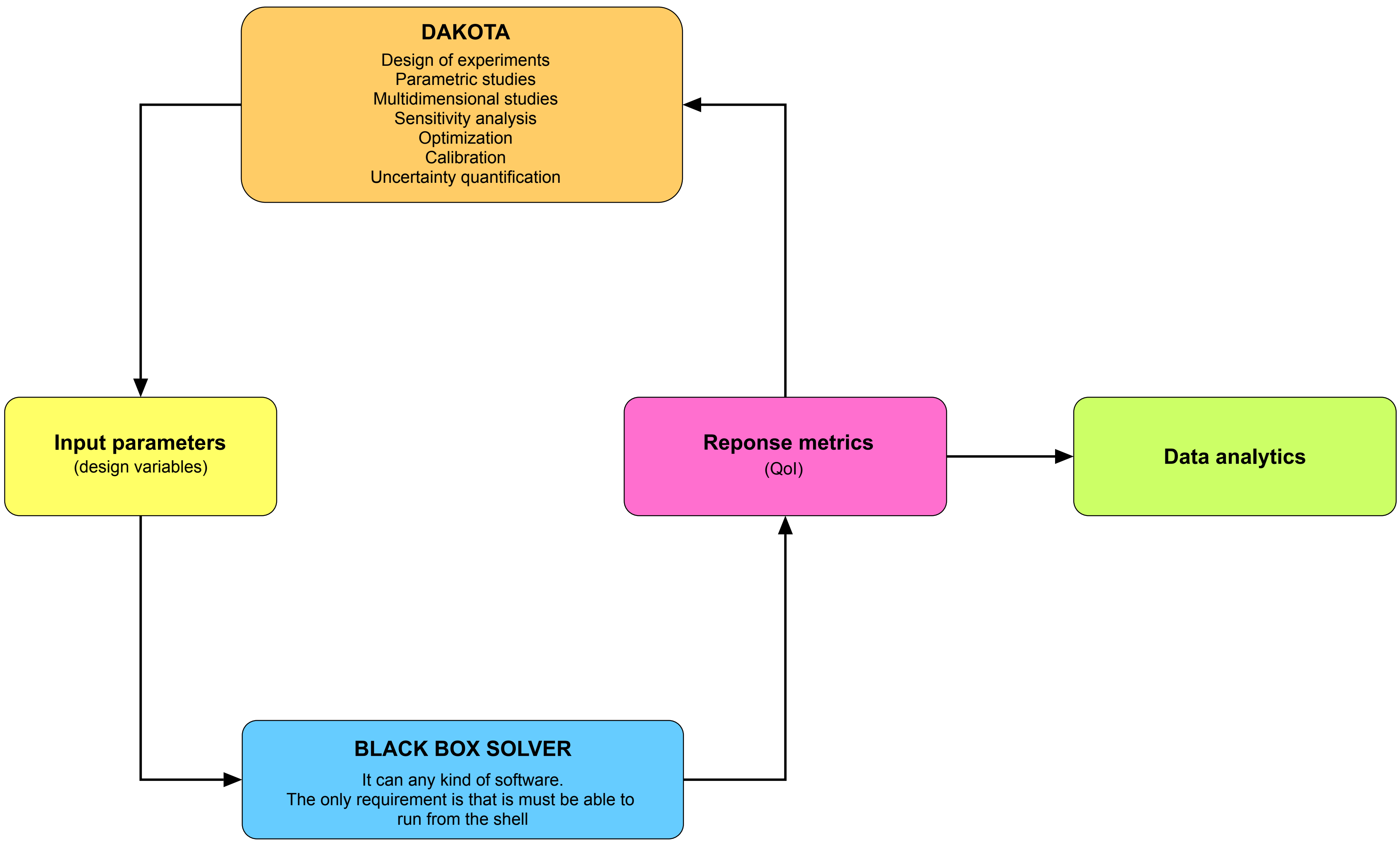
The optimization driver

The big picture – Tools in use



The optimization driver

The big picture – The optimization loop



The optimization driver

DAKOTA in a nutshell

- DAKOTA stands for **D**esign and **A**nalysis tool**K**it for **O**ptimization and **T**erascale **A**pplications.

- DAKOTA is a general-purpose software toolkit for performing optimization, uncertainty quantification, parameter estimation, design of experiments, and sensitivity analysis on high performance computers.

    - DAKOTA is developed and supported by U.S. Sandia National Labs.
    - DAKOTA is well documented and comes with many tutorials.
    - Extensive support via a dedicated mailing list.
    - Distribute under the GNU GPL.

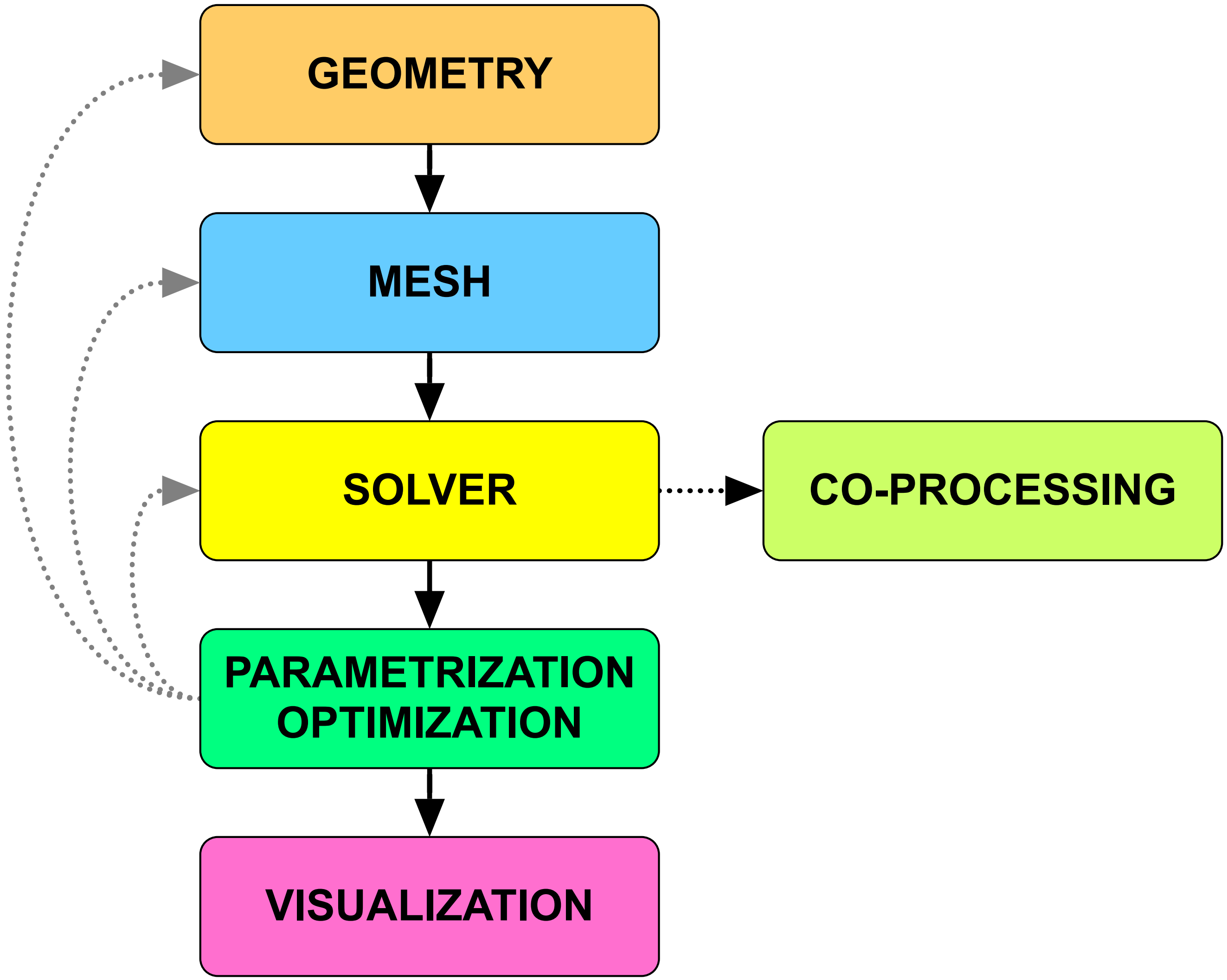
The optimization driver

DAKOTA capabilities

- Parameter Studies (PS).
- Design of Experiments (DOE) – Design and Analysis of Computer Experiments (DACE) .
- Sensitivity Analysis (SA).
- Uncertainty Quantification (UQ).
- Optimization (OPT) via Gradient-based and derivative-free local and global methods.
- Surrogate based optimization (SBO).
- Calibration (CAL) or data fitting – Parameter estimation.
- Generic interface to black box solvers.
- Scalable parallel computations from desktop to clusters.
- Asynchronous evaluations.
- Restart capabilities and Python interface.

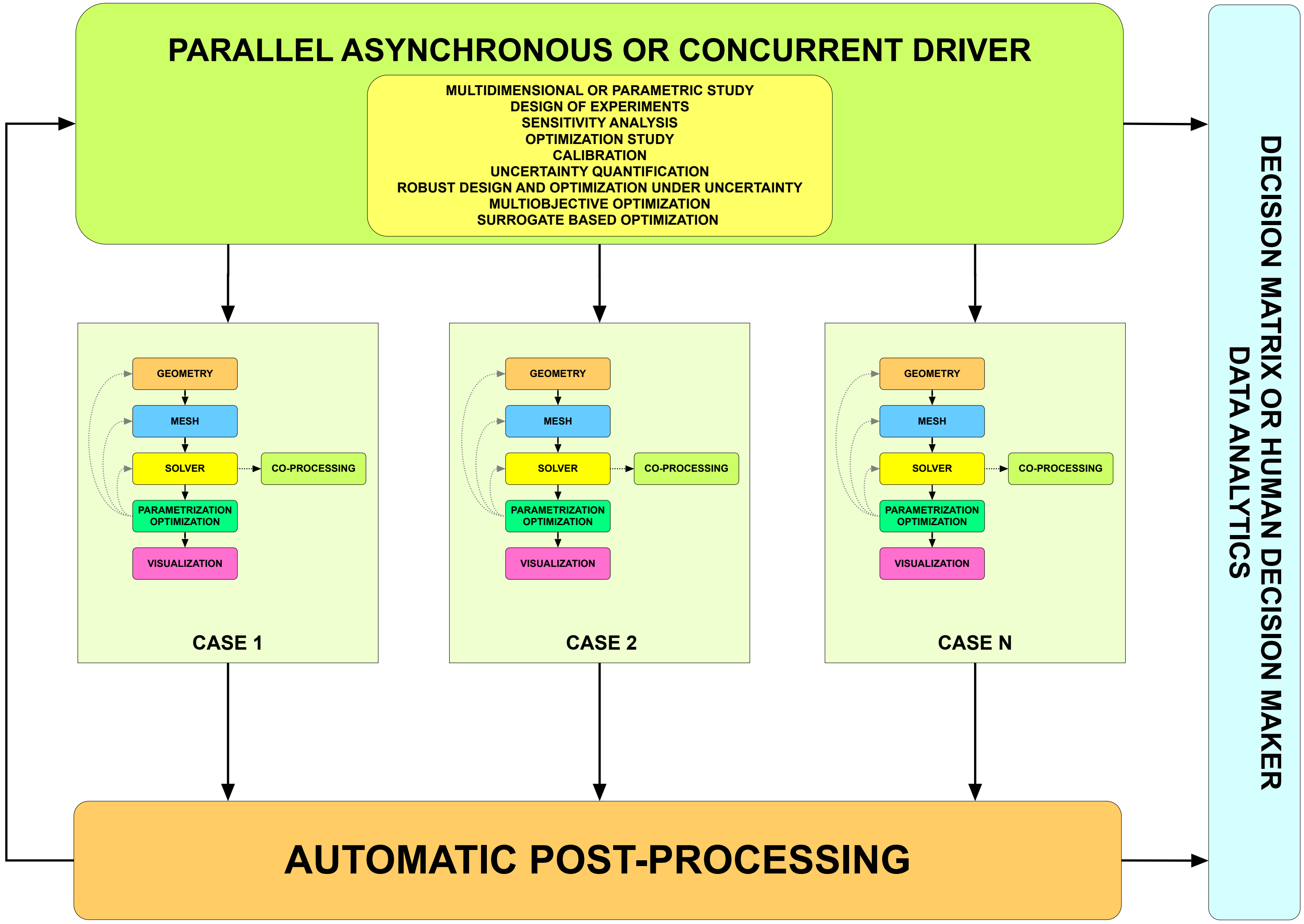
The optimization driver

Typical optimization loop



The optimization driver

Typical optimization loop



The optimization driver

Summary of DAKOTA optimization methods

Gradient-based Optimization:

- DOT: frcg, bfgs, mmfd, slp, sqp **(commercial)**
- **CONMIN:** frcg, mfd
- **NPSOL:** sqp **(commercial)**
- **NLPQLP:** sqp **(commercial)**
- **OPT++:** cg, Newton, quasi-Newton

Derivative-free Optimization:

- **COLINY:** PS, EA, Solis-Wets, COBYLA, DIRECT
- **JEGA:** MOGA, SOGA
- **EGO**: efficient global optimization via Gaussian Process models
- **NCSU:** DIRECT
- **OPT++:** PDS (Parallel Direct Search, simplex based method)

Parameter studies: vector, list, centered, grid, multidimensional

Design of experiments:

- **DDACE:** LHS, MC, grid, OA, OA_LHS, CCD, BB
- **FSUDace:** CVT, Halton, Hammersley
- **PSUADE:** MOAT
- **Sampling:** LHS, MC, Incr. LHS, IS/AIS/MMAIS

Multi-objective optimization, pareto, hybrid, multi-start, surrogate-based optimization (local and global).

Roadmap

1. ~~Brief overview of optimization~~
2. ~~Surrogate based optimization~~
3. ~~The optimization driver~~
4. **Practical applications**
5. Wrap-up

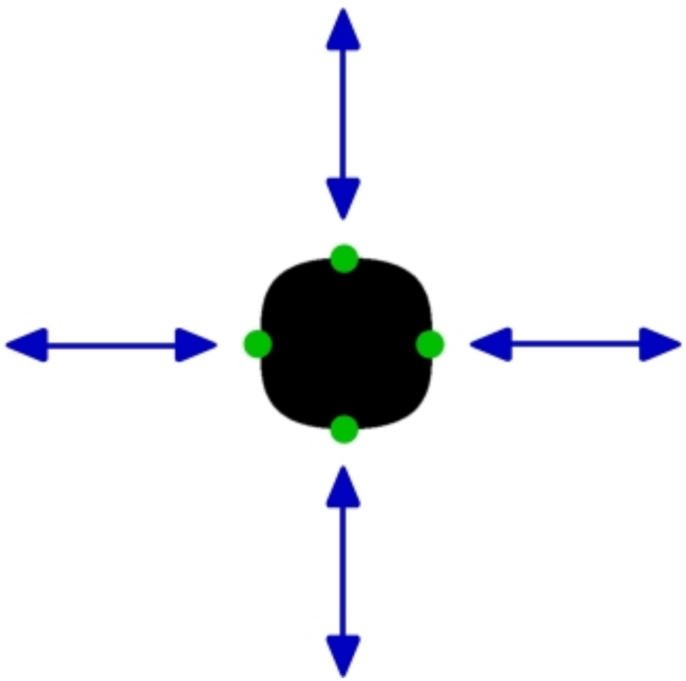
Practical applications

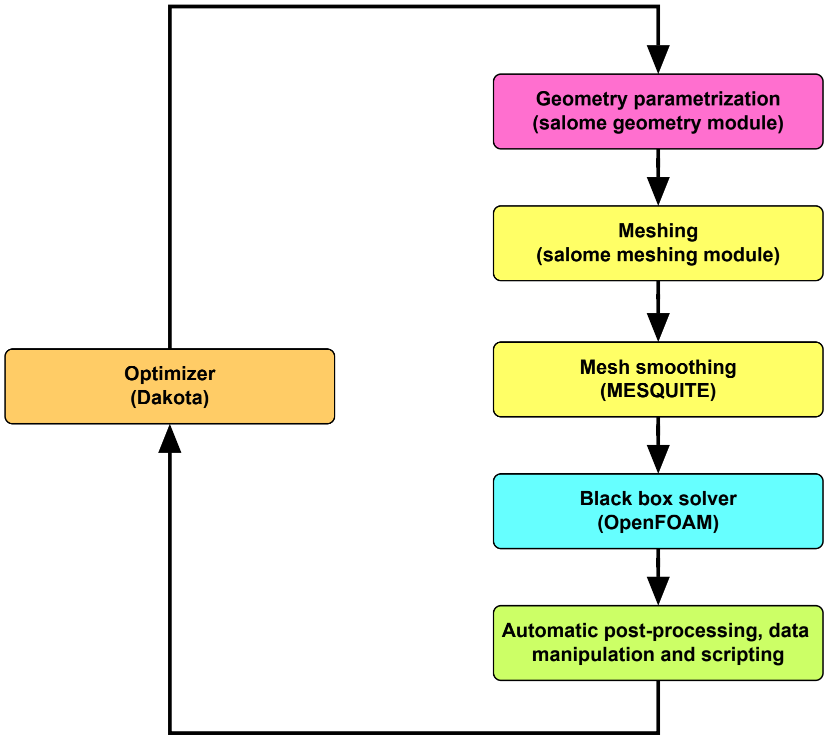
Blunt body shape optimization

Practical applications

Blunt body shape optimization

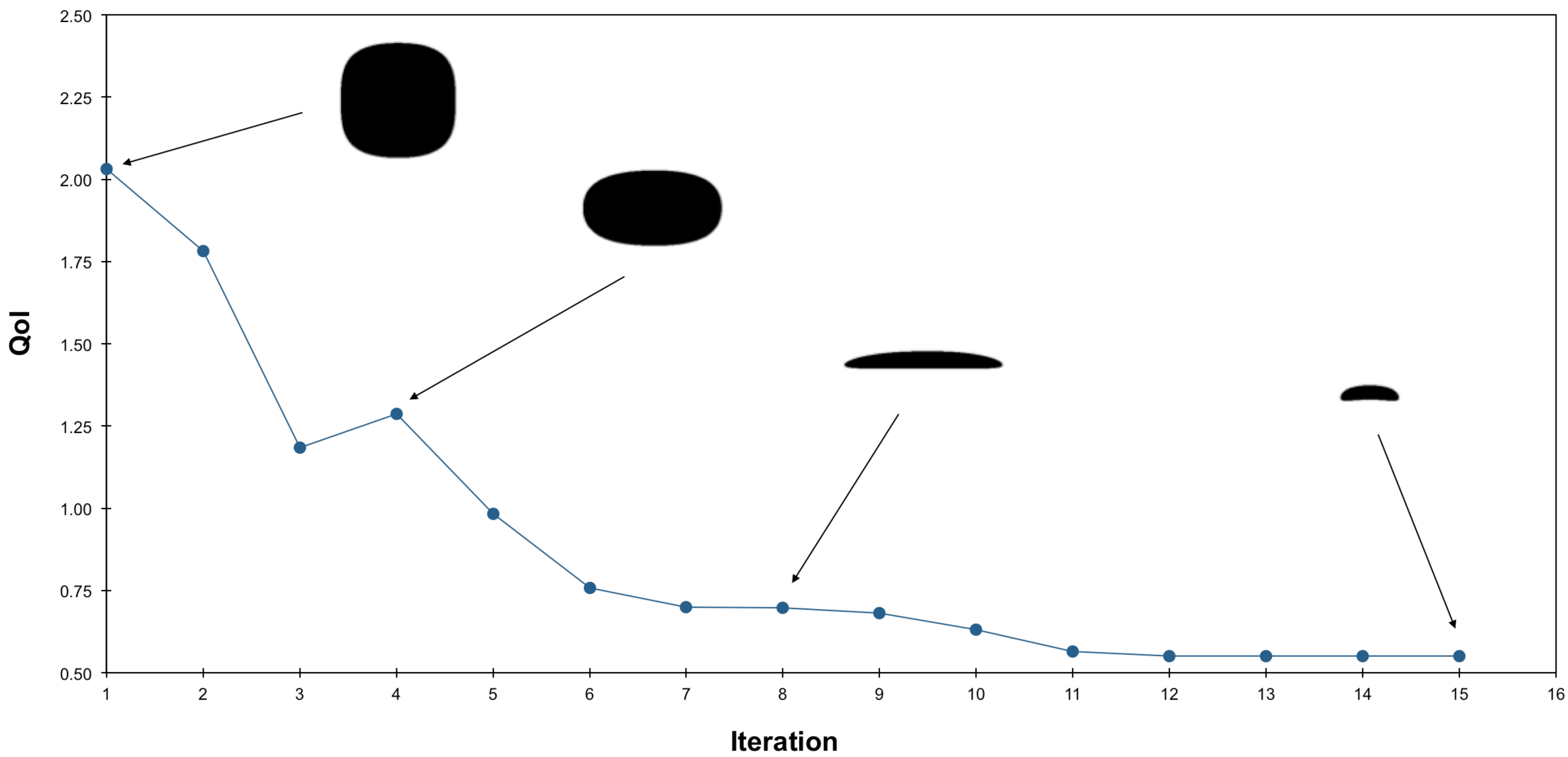
- In this case we aim at optimizing the shape of a blunt body.
- The goal is to minimize the drag coefficient.
- The body is parametrized using Bezier curves with four control points.
- In this case we use gradient based optimization and six linear constraints.





Practical applications

Blunt body shape optimization



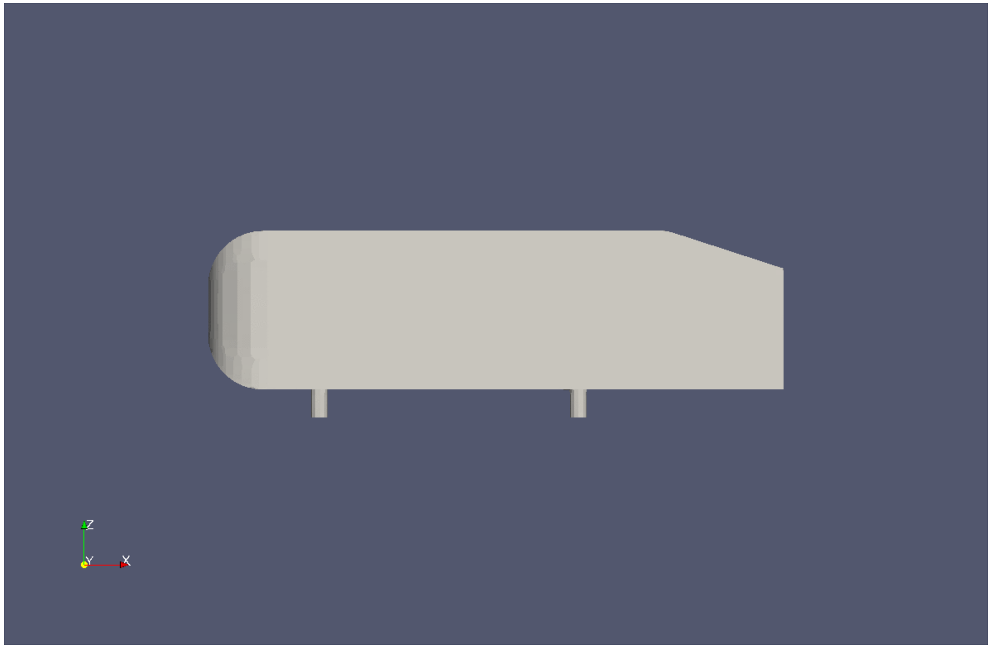
Practical applications

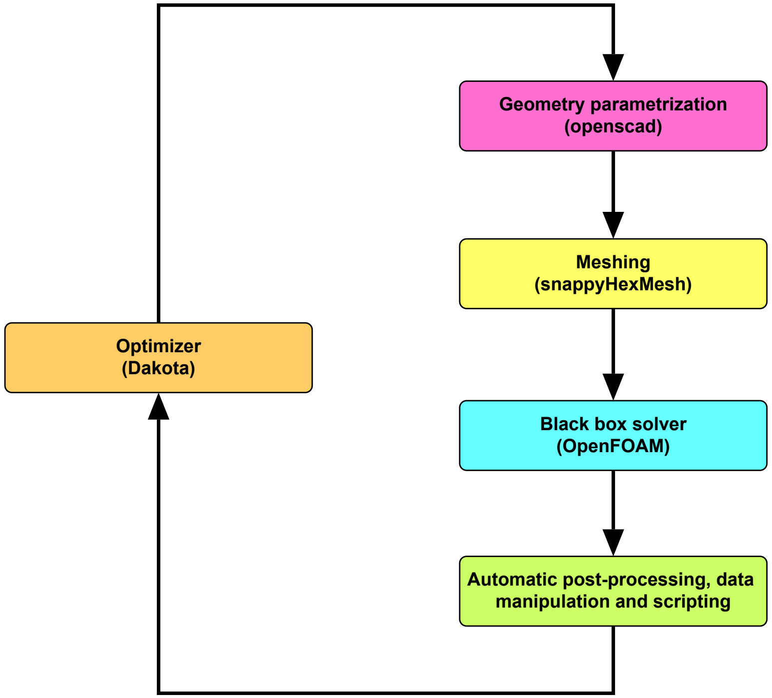
Ahmed body

Practical applications

Ahmed body

- In this case we aim at optimizing the ahmed body.
- The design variable is the slant angle and the objective function is the drag coefficient.
- In this example we conducted a parametric study, a constrained gradient optimization and a surrogate based optimization (SBO).

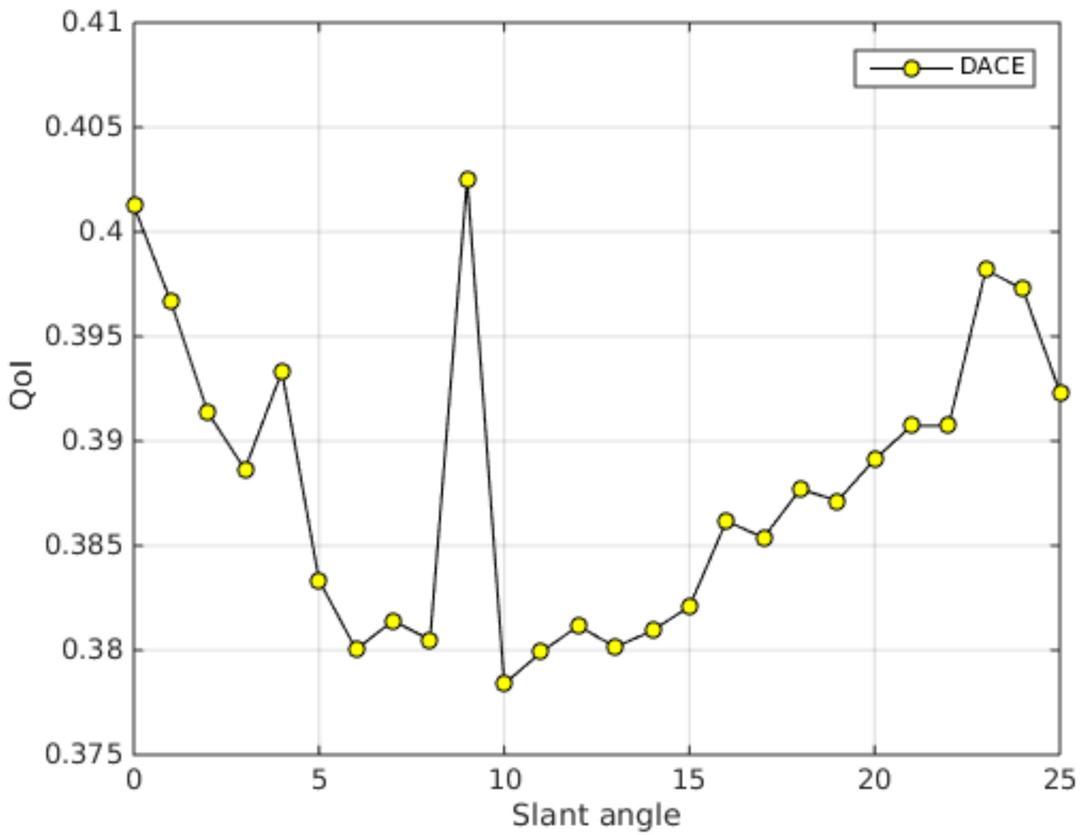




Practical applications

Ahmed body

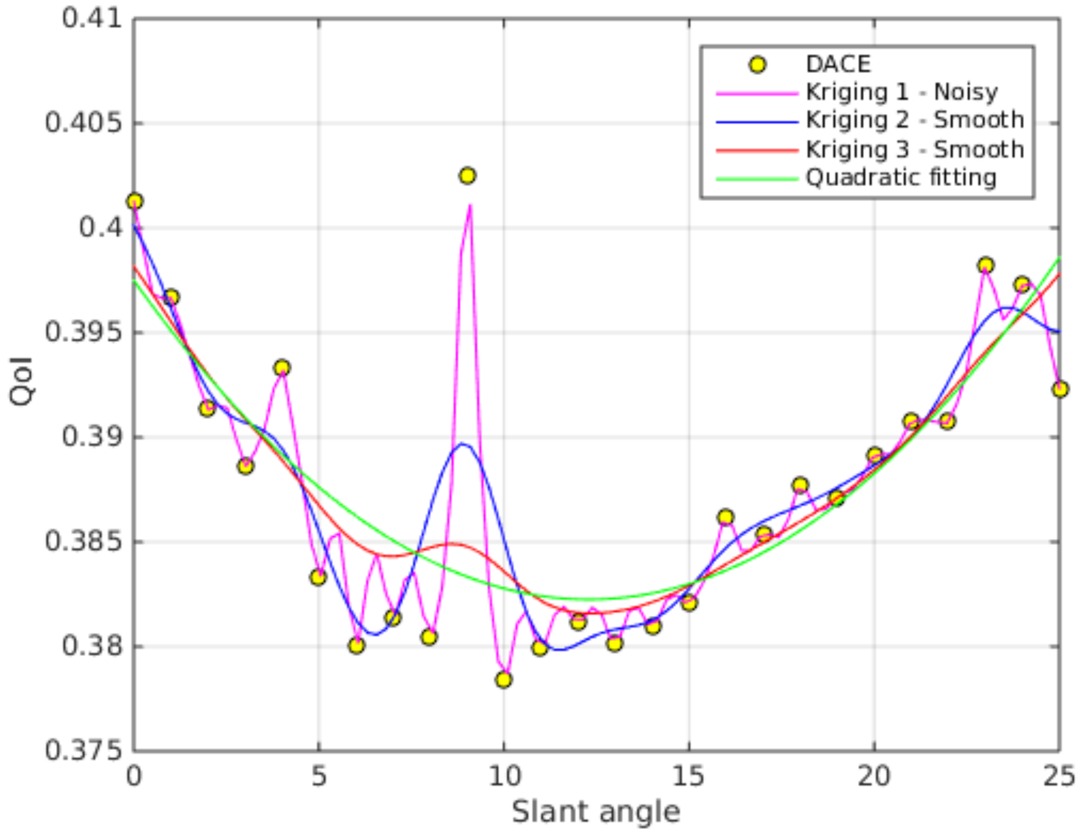
Parametric study



Practical applications

Ahmed body

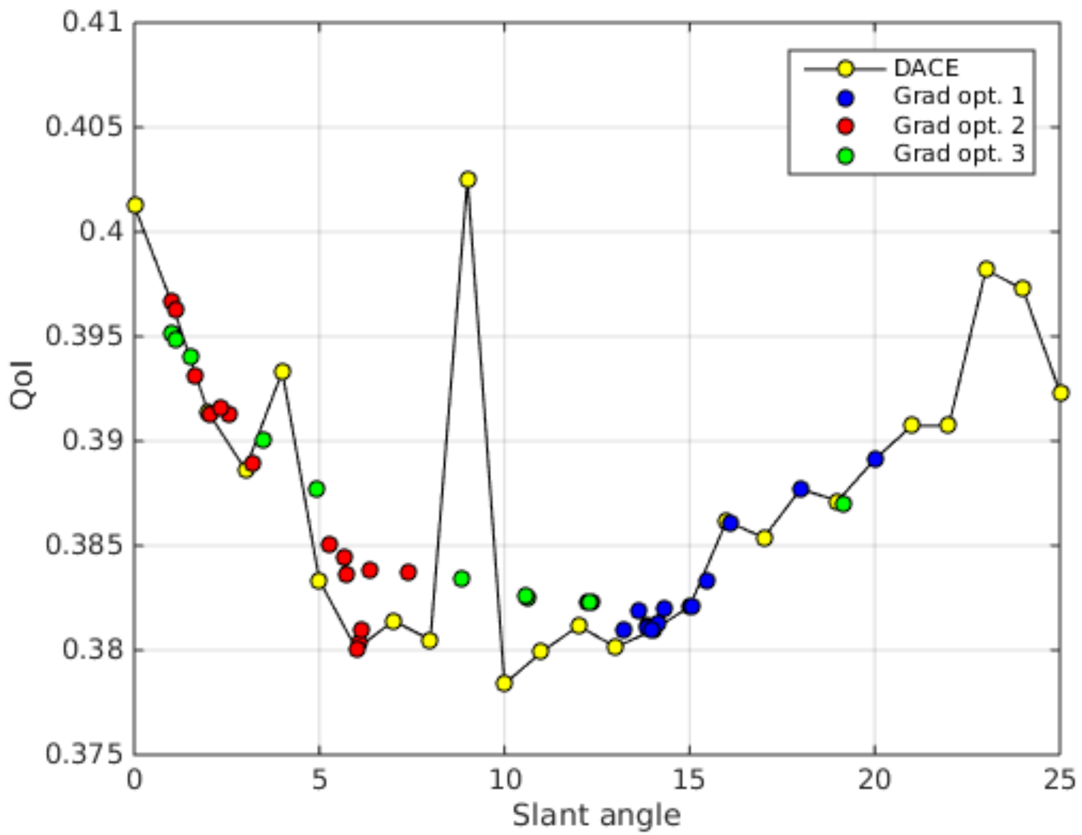
Surrogate, meta-models or response surface.



Practical applications

Ahmed body

Surrogate based optimization on the surrogate built using Kriging interpolation.
Optimization method: Method of feasible direction.



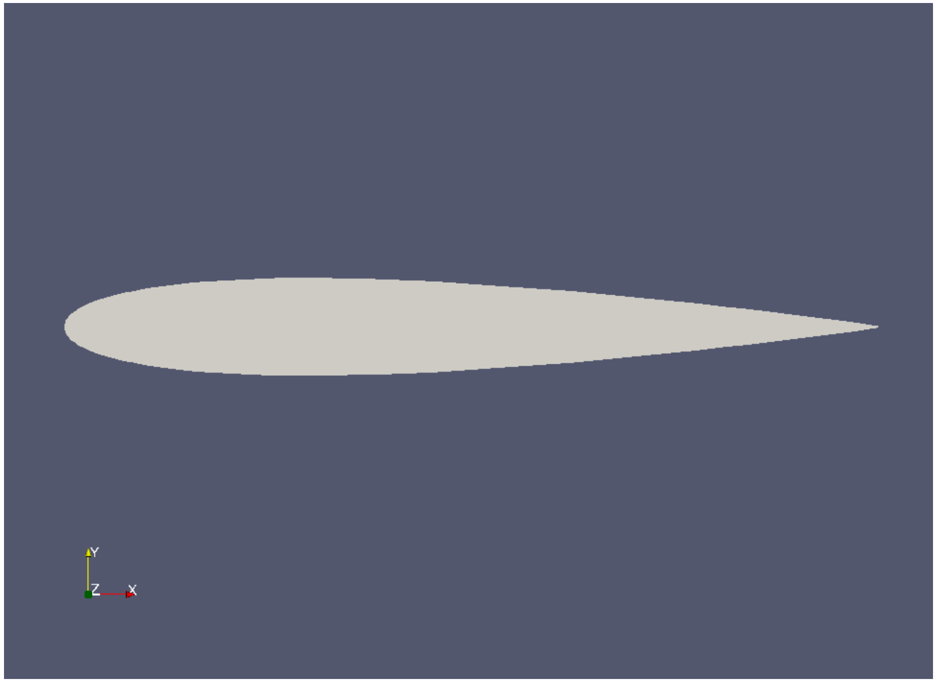
Practical applications

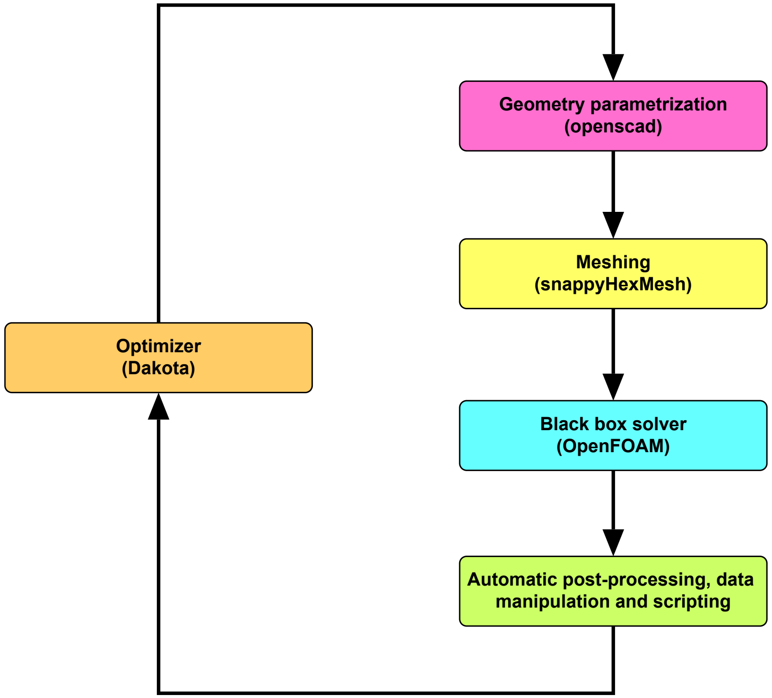
NACA airfoil shape optimization

Practical applications

NACA airfoil shape optimization

- In this case we aim at optimizing the shape of a NACA Series 4 airfoil. The goals are maximize the lift coefficient and minimize the drag coefficient
- The design variables are the curvature and position of the maximum curvature and the objective functions are drag, lift and moment coefficient.
- Hereafter, we show a MOO case using evolutionary algorithms (EA) and surrogate based optimization (SBO).

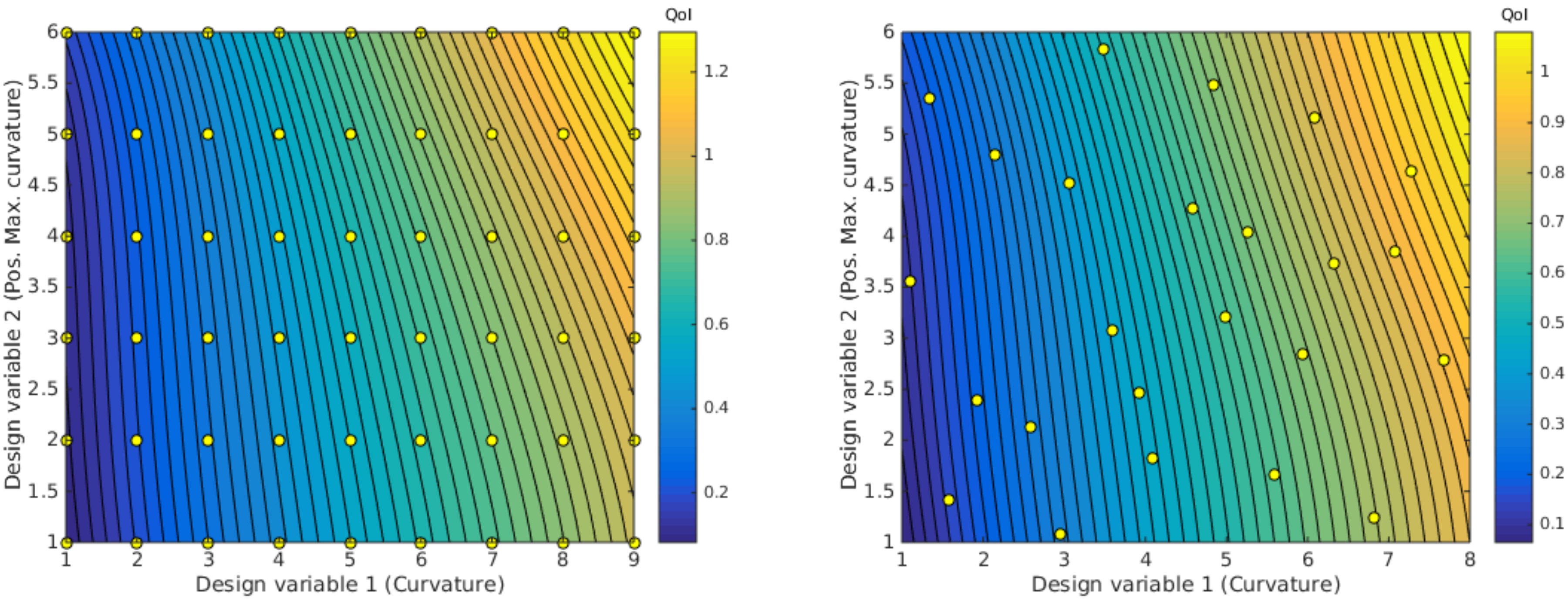




Practical applications

NACA airfoil shape optimization

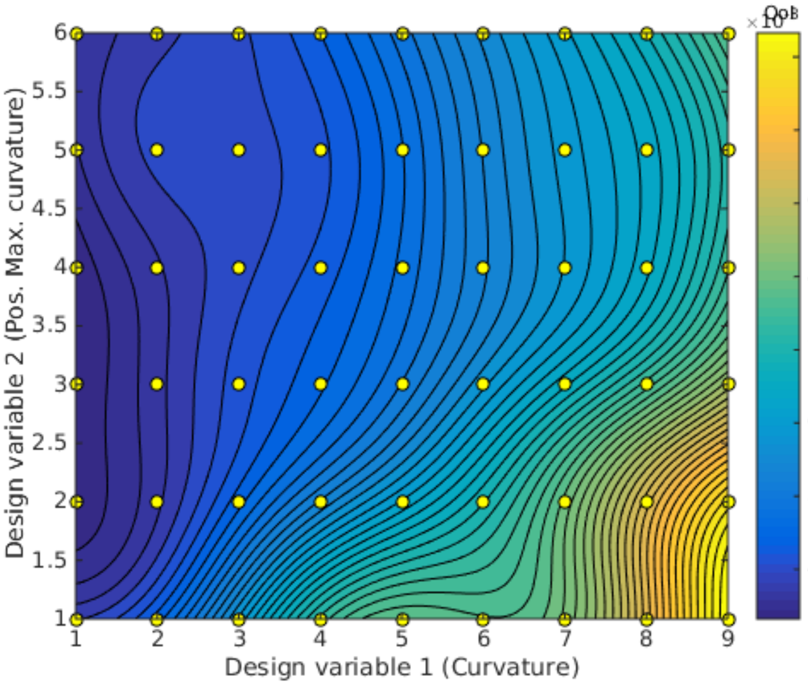
Lift coefficient surrogate (kriging interpolation).

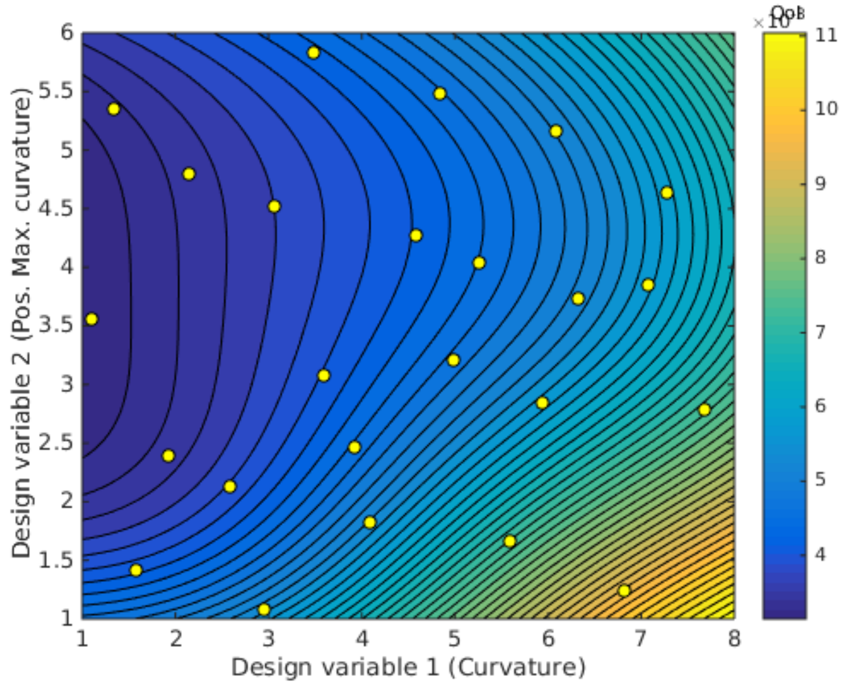


Practical applications

NACA airfoil shape optimization

Drag coefficient surrogate (kriging interpolation).

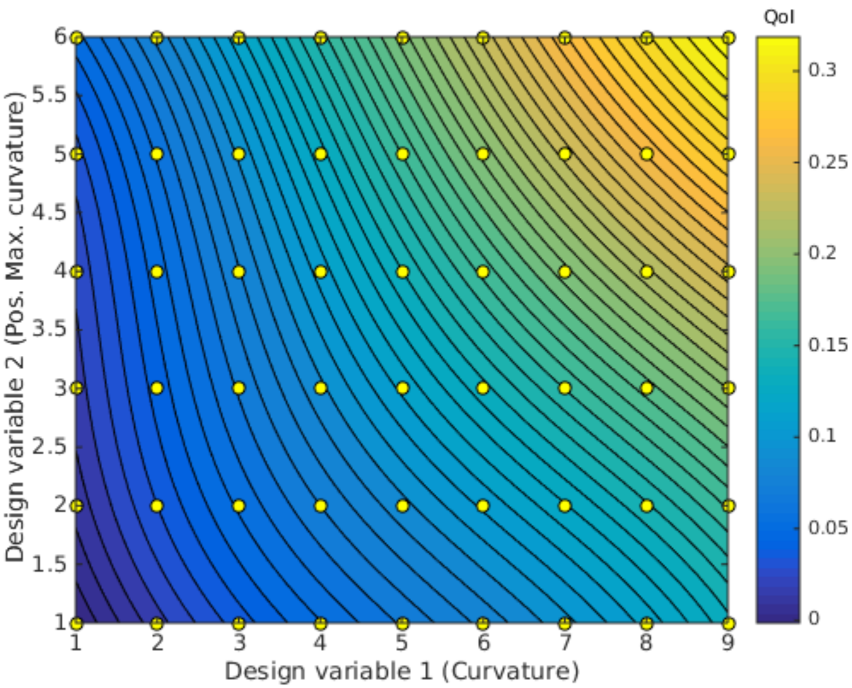


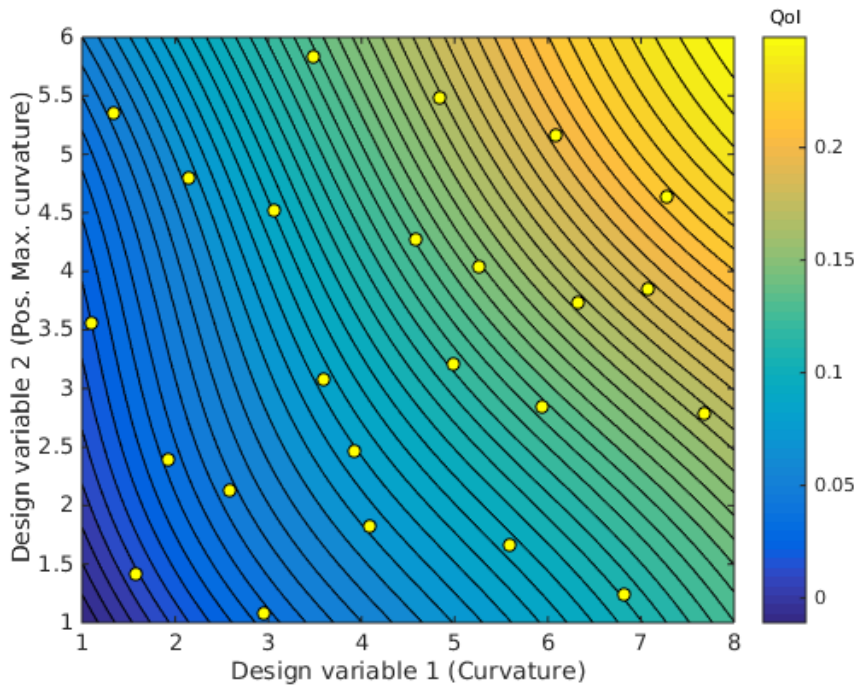


Practical applications

NACA airfoil shape optimization

Moment coefficient surrogate (kriging interpolation).

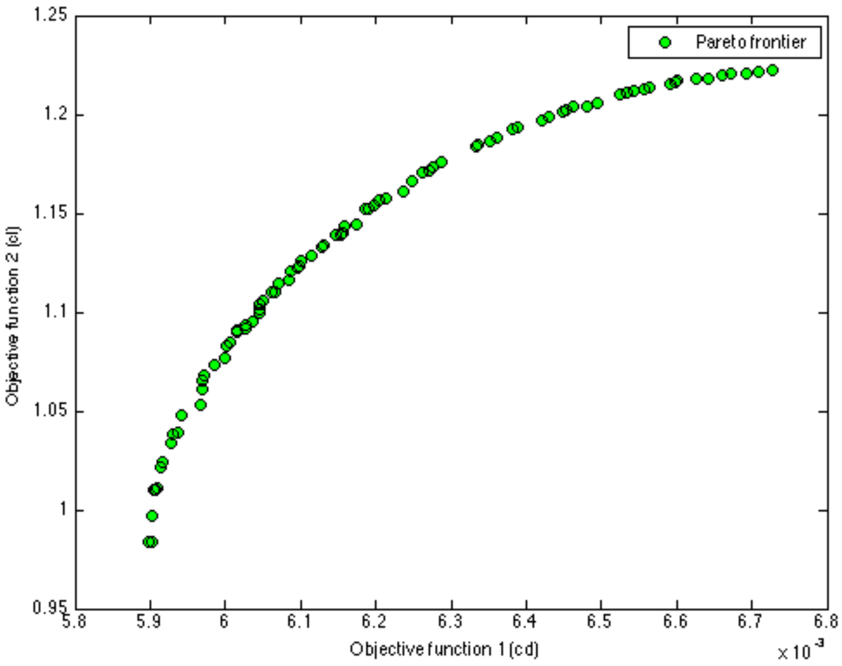


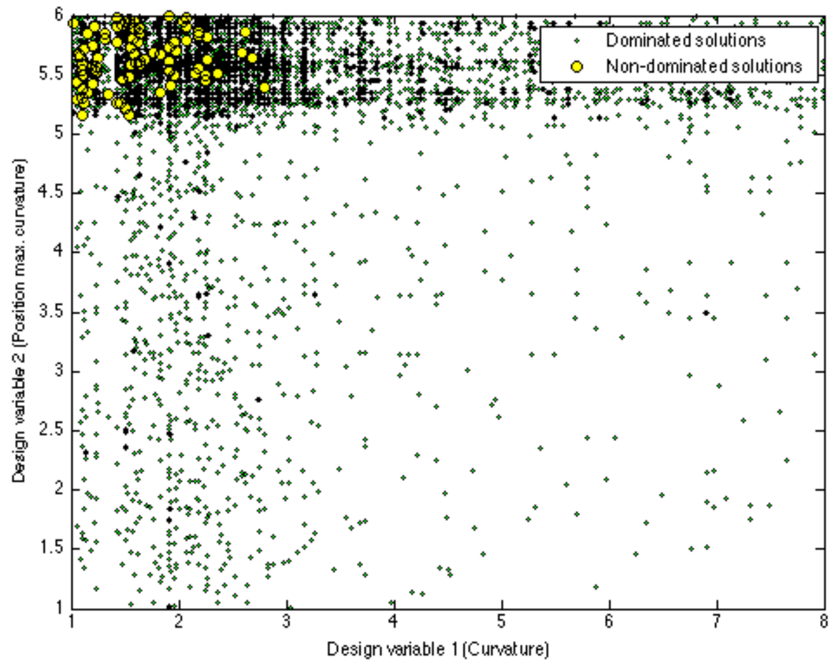


Practical applications

NACA airfoil shape optimization

- After building the surrogates, we can optimize the airfoil shape.
- The goals are to maximize the lift coefficient and minimize the drag coefficient.
- For MOO we use the derivative-free MOGA method.





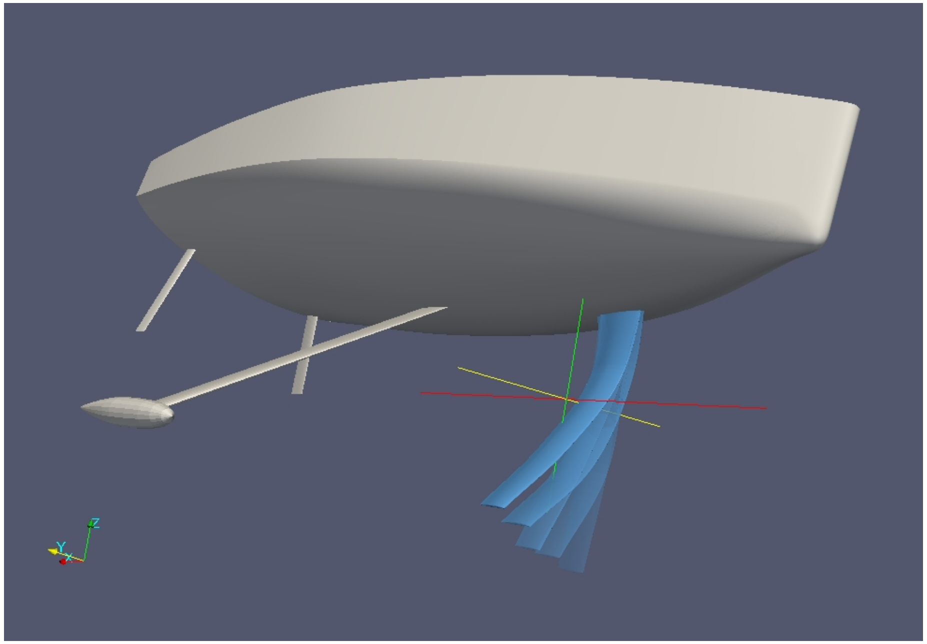
Practical applications

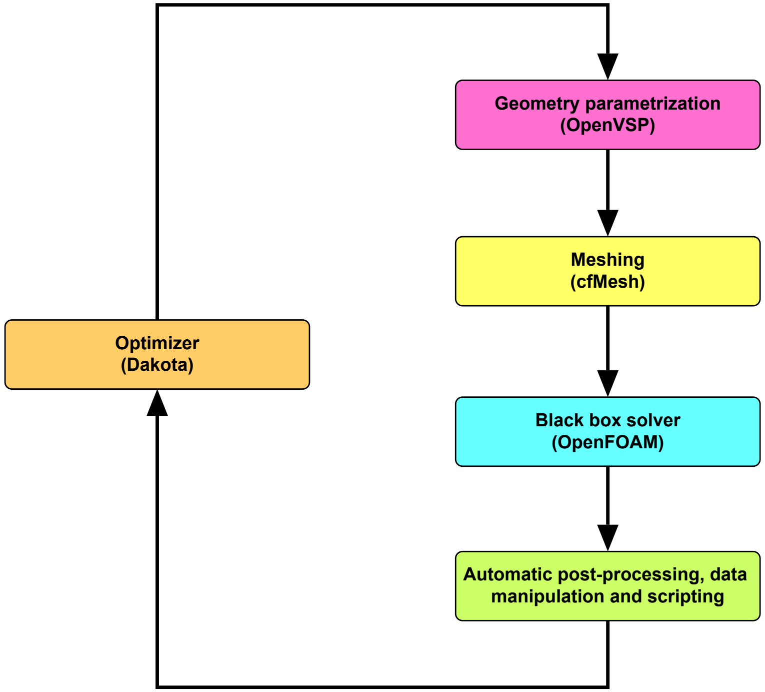
Sailing yacht daggerboard optimization

Practical applications

Sailing yacht daggerboard optimization

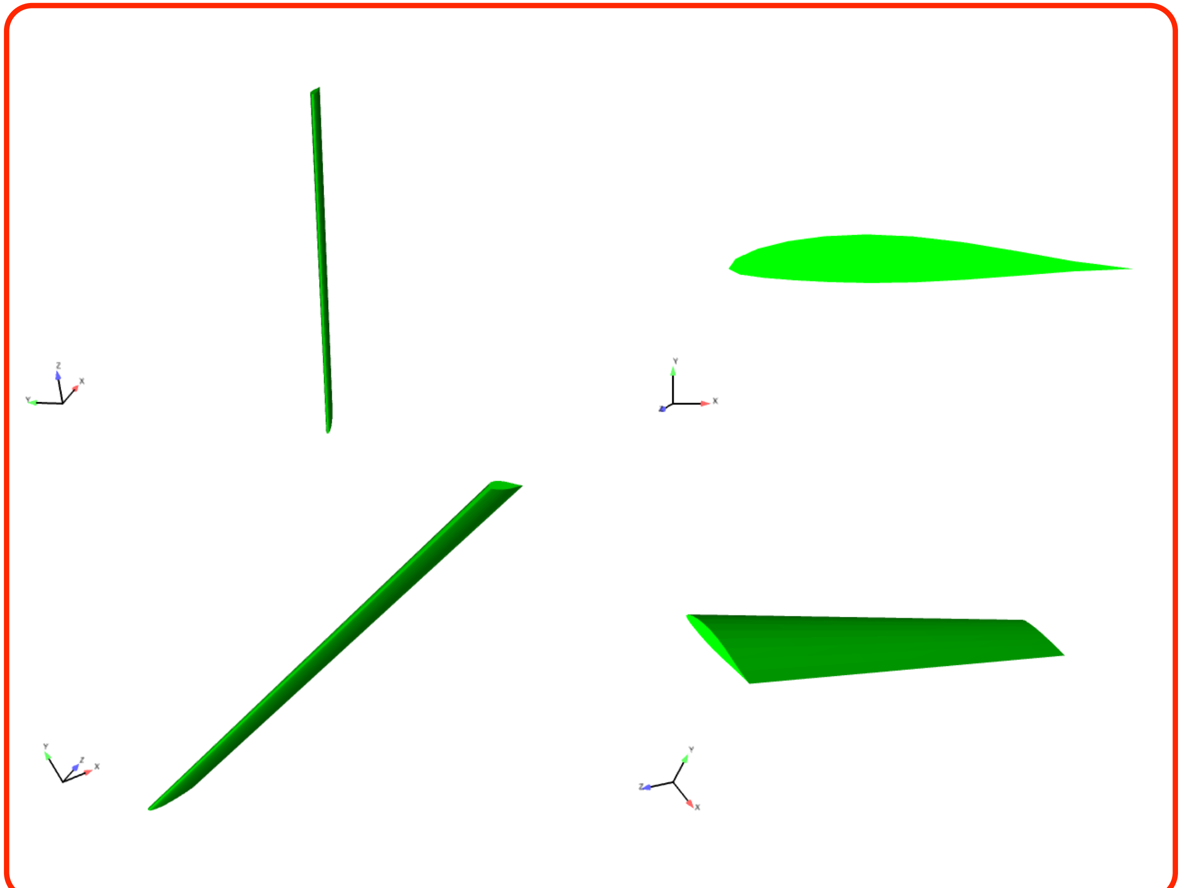
- In this case we aim at optimizing the shape of a daggerboard. The goals are maximize the vertical force and minimize the drag coefficient.
- There are 12 design variables and 1 non-linear constraint (the lateral force on the daggerboard). All design variables are bounded and for the non-linear constraint we use an inequality.
- The design variables control the airfoil shape (NACA 6-Series and NACA 4-Series) and the daggerboard shape and flexion.
- To conduct the MOO we use the MOGA method and SBO.
- We also perform online data analytics using Python.





Practical applications

Sailing yacht daggerboard optimization



12 Design variables (dv):

- 3 airfoil dv – c_l (dv1, dv2, dv3)
- 3 airfoil dv – A (dv4, dv5, dv6)
- 3 wing chord dv (dv7, dv8, dv9)
- 2 wing dihedral dv (dv10, dv11)
- 1 wing sweep dv (dv12)

2 Objective functions (of)

- Drag (of1)
- Vertical force (of2)

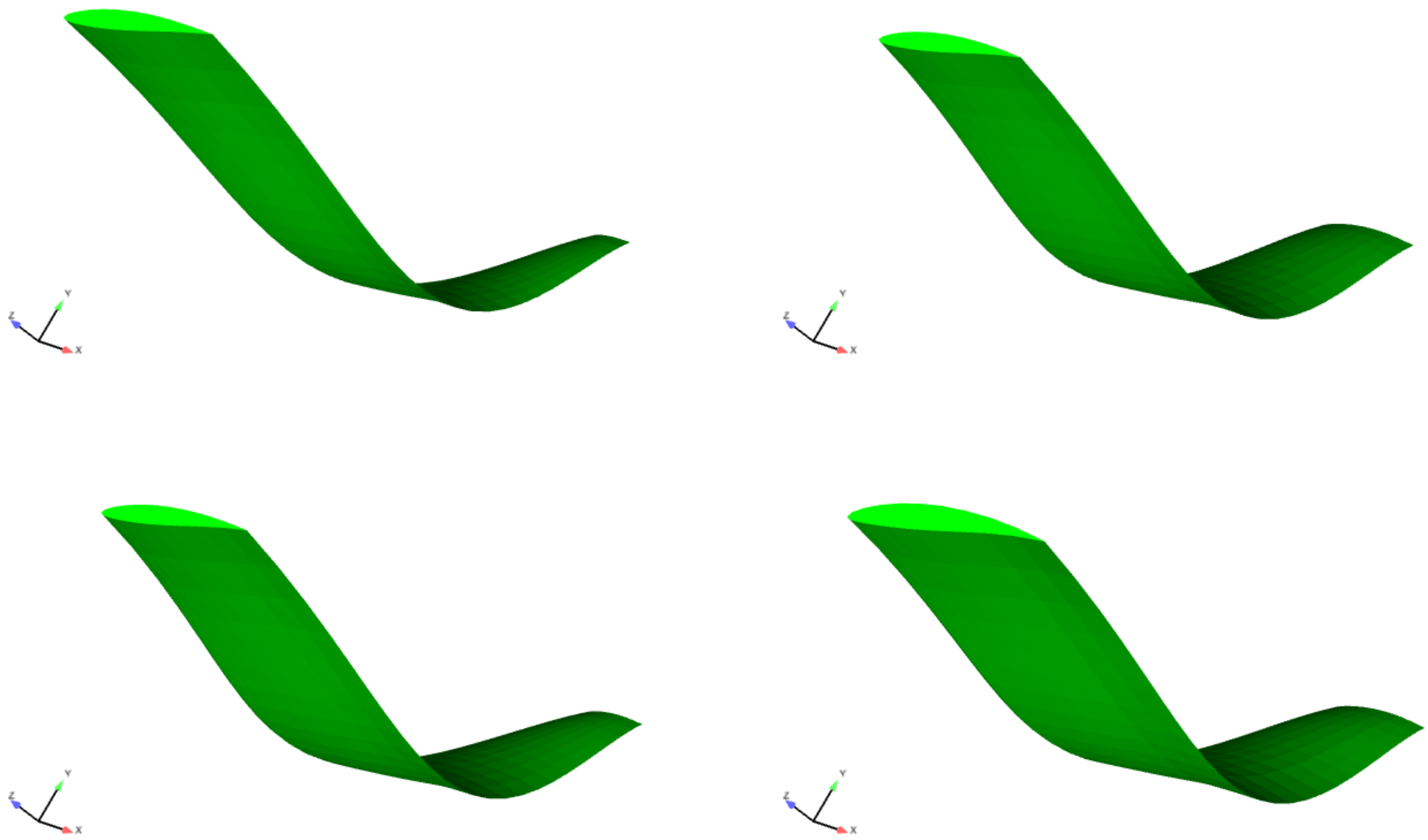
1 non-linear constraint (of)

- Lateral force (of3)

Daggerboard – Initial geometry

Practical applications

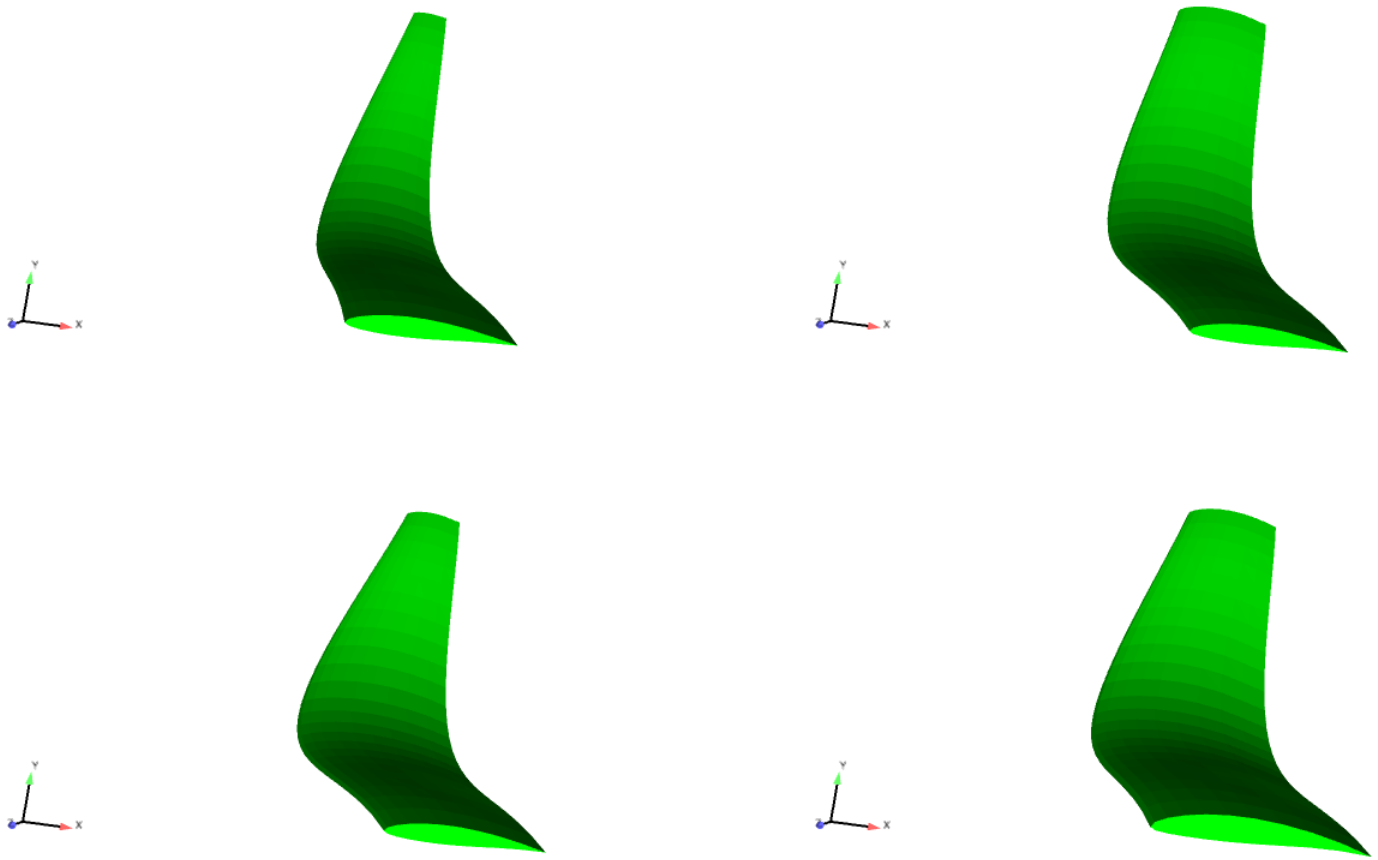
Sailing yacht daggerboard optimization



Daggerboard – Optimized geometry (4 non-dominated solutions)

Practical applications

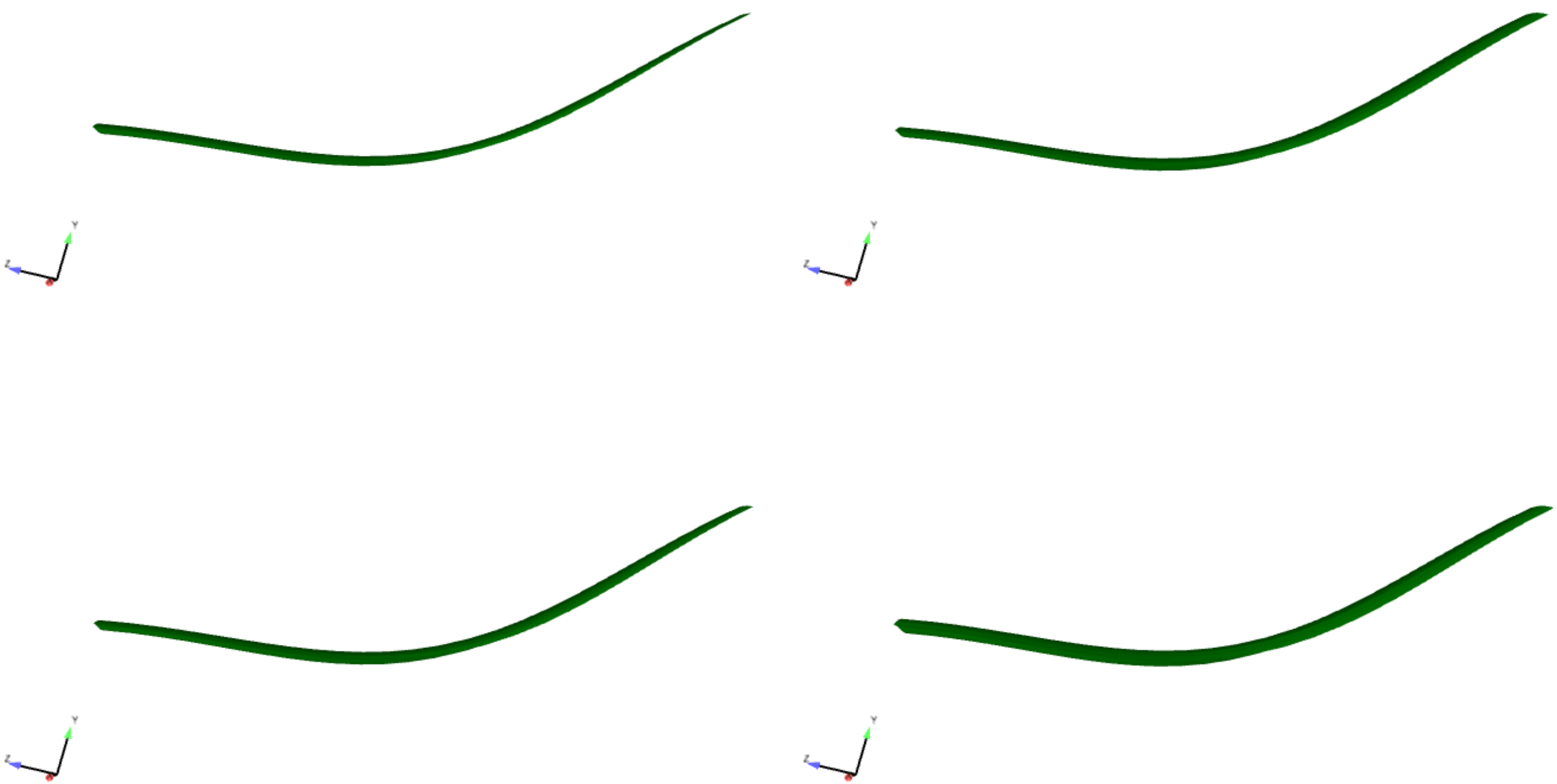
Sailing yacht daggerboard optimization



Daggerboard – Optimized geometry (4 non-dominated solutions)

Practical applications

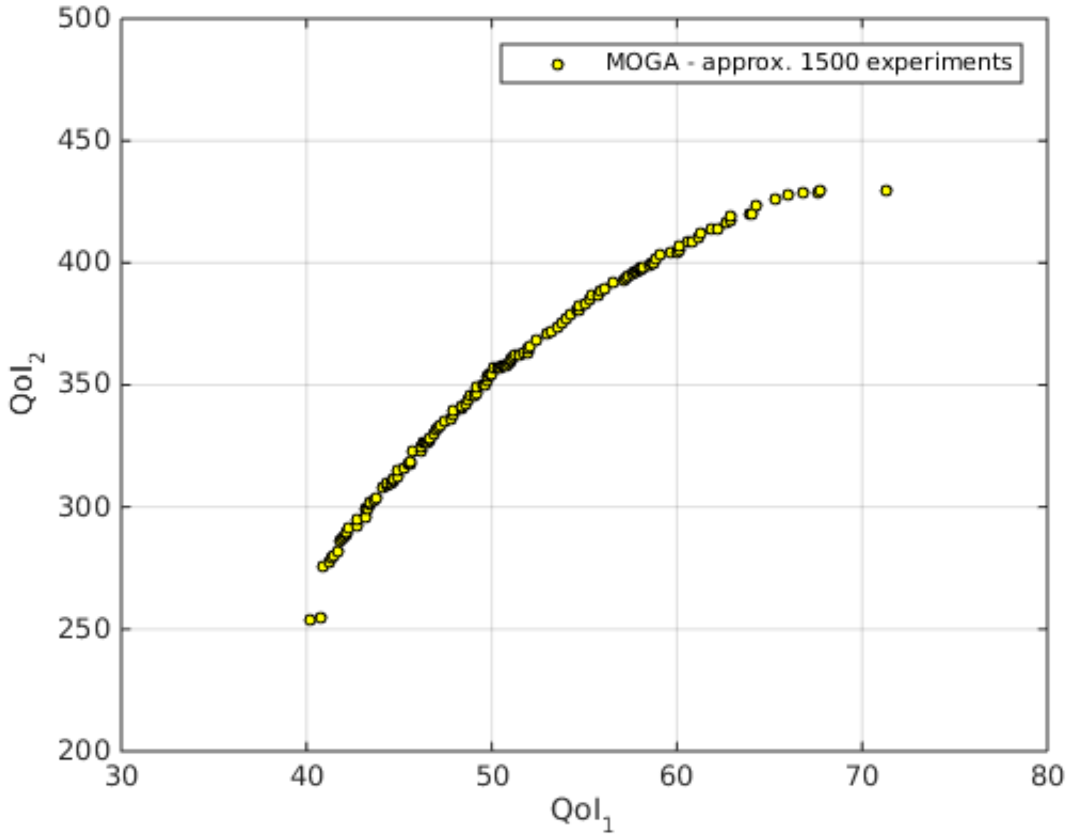
Sailing yacht daggerboard optimization



Daggerboard – Optimized geometry (4 non-dominated solutions)

Practical applications

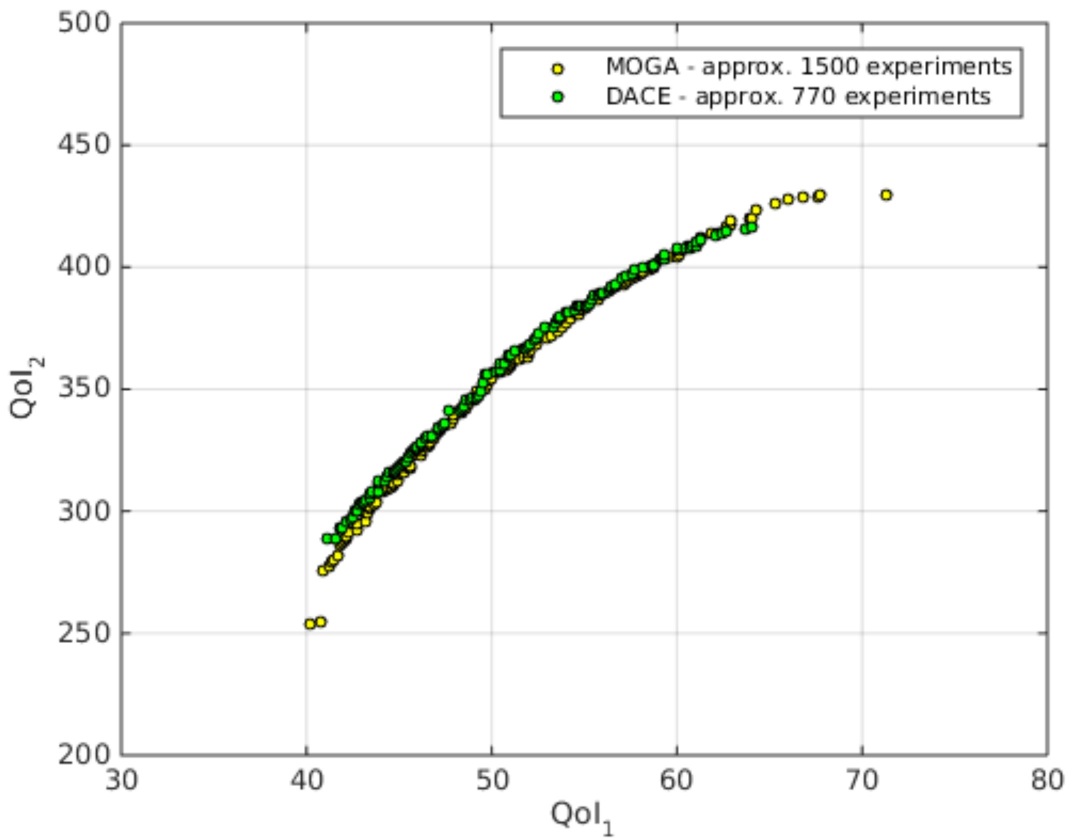
Sailing yacht daggerboard optimization



Pareto front
(QoI 1 = drag, QoI 2 = vertical force)

Practical applications

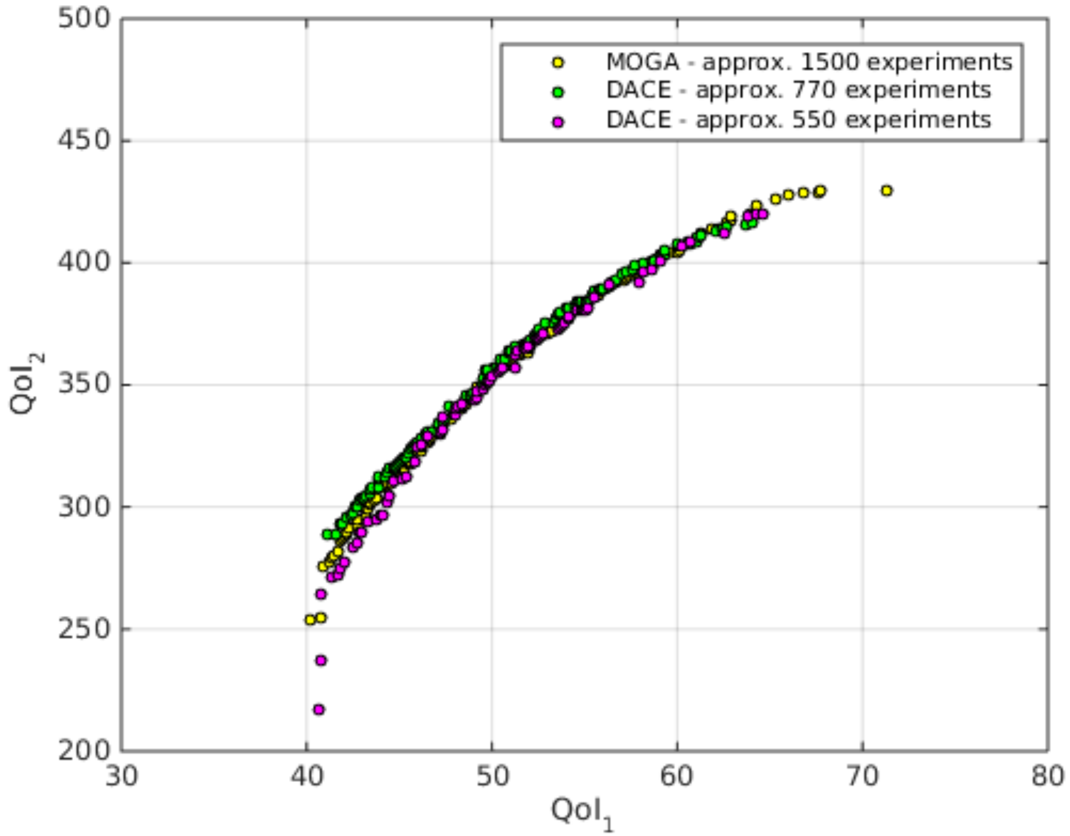
Sailing yacht daggerboard optimization



Pareto front
(QoI 1 = drag, QoI 2 = vertical force)

Practical applications

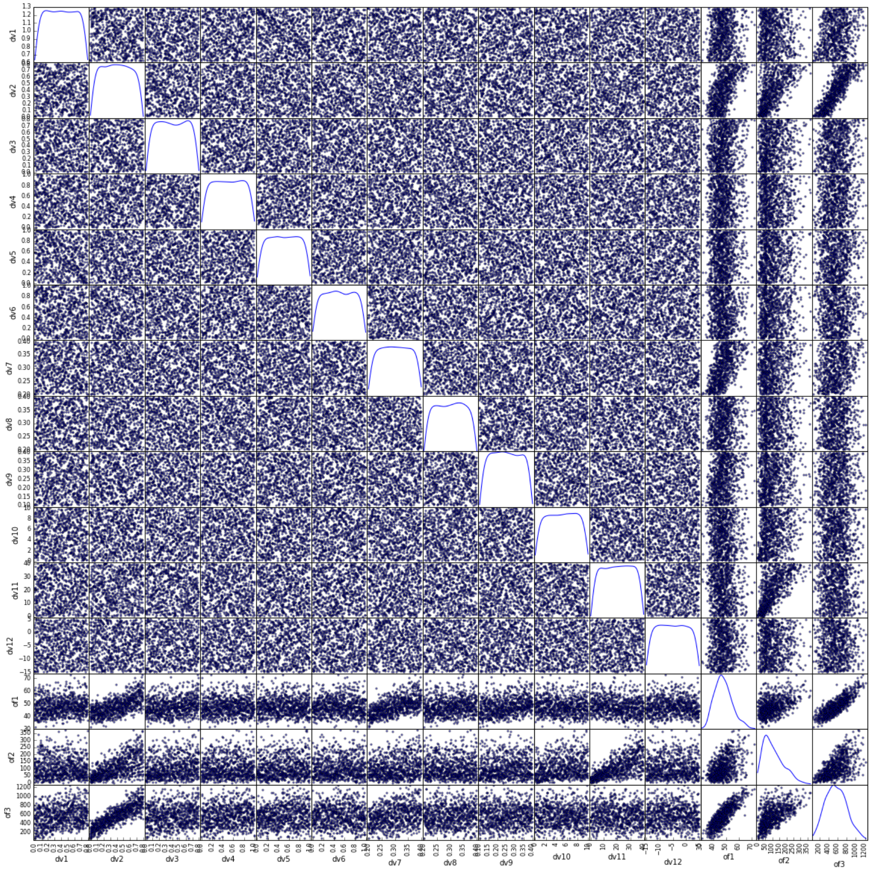
Sailing yacht daggerboard optimization



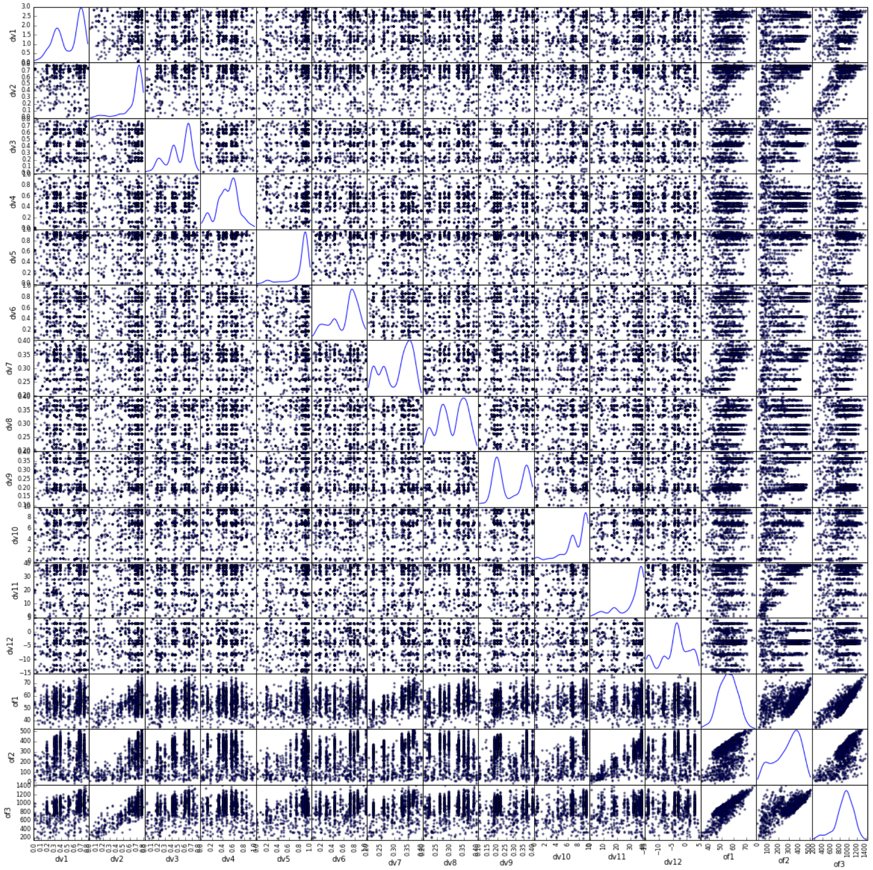
Pareto front
(QoI 1 = drag, QoI 2 = vertical force)

Practical applications

Sailing yacht daggerboard optimization



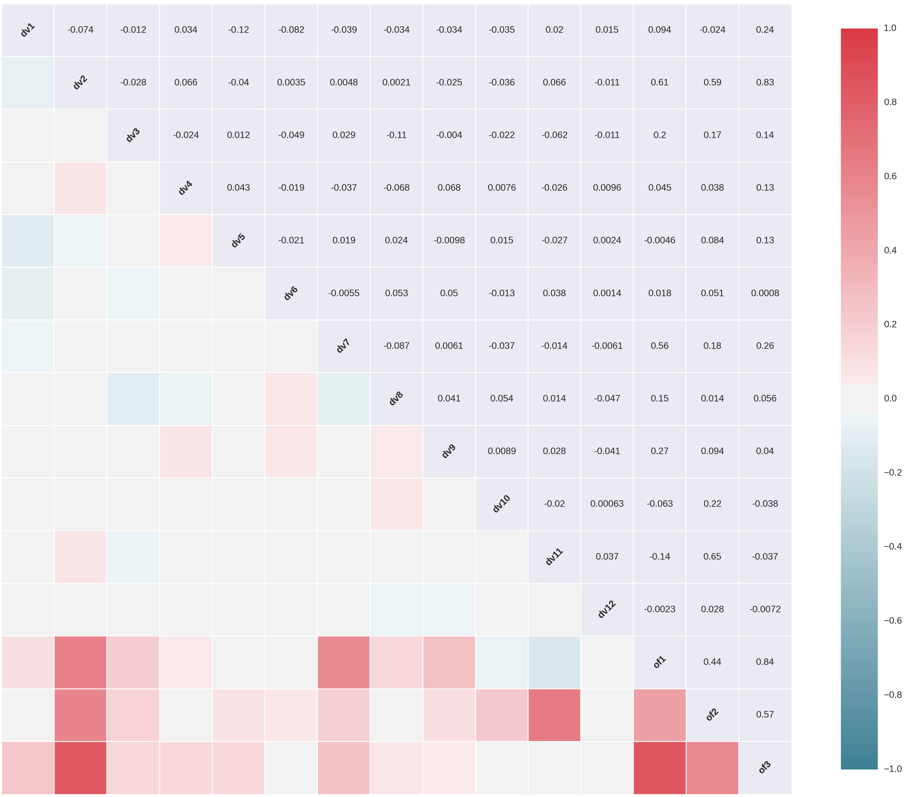
Scatter plot matrix – DACE 700



Scatter plot matrix – MOGA

Practical applications

Sailing yacht daggerboard optimization

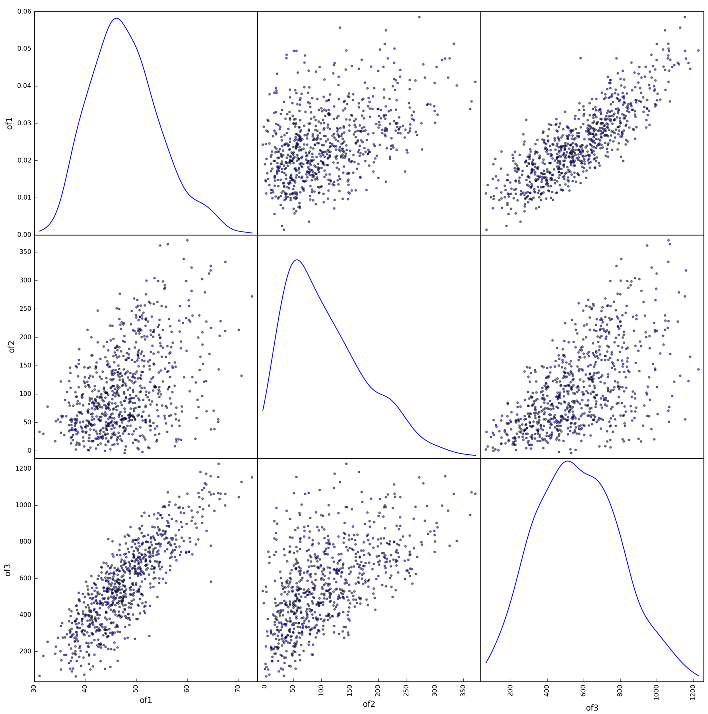


Correlation matrix – DACE 700

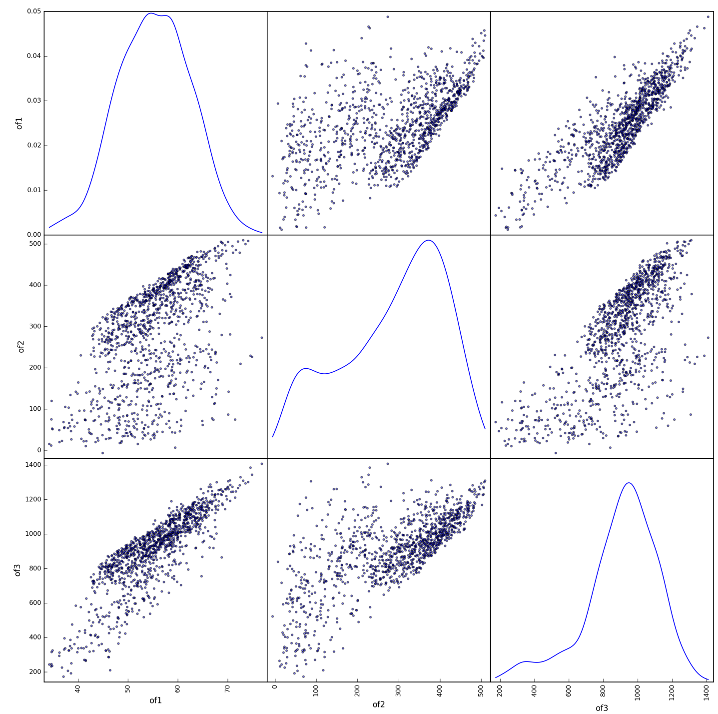
Correlation matrix – MOGA

Practical applications

Sailing yacht daggerboard optimization



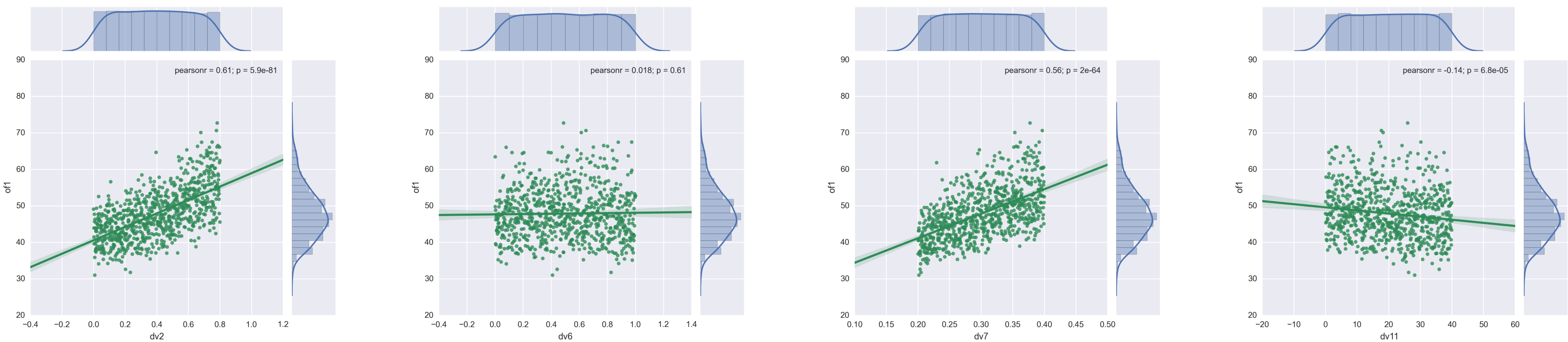
Scatter plot matrix – DACE 700



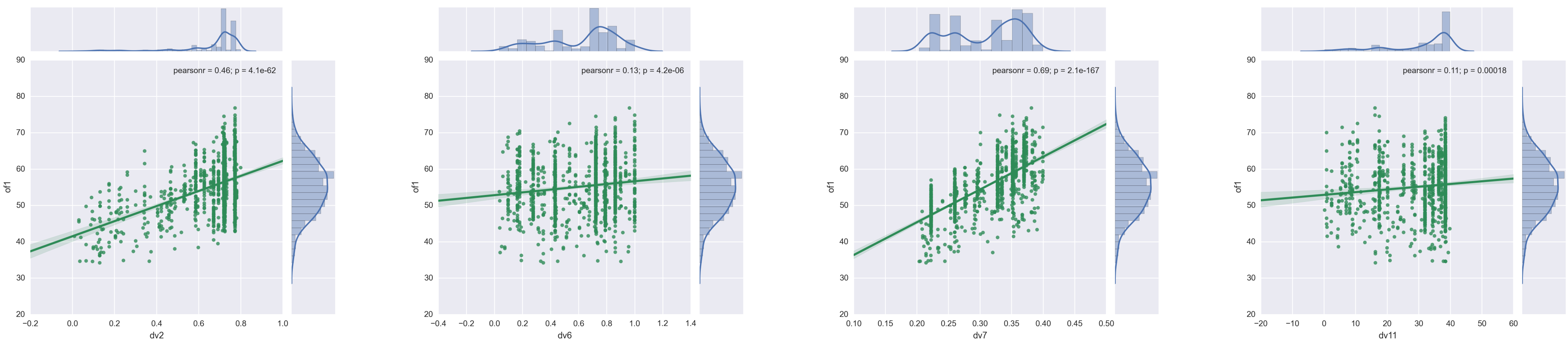
Scatter plot matrix – MOGA

Practical applications

Sailing yacht daggerboard optimization



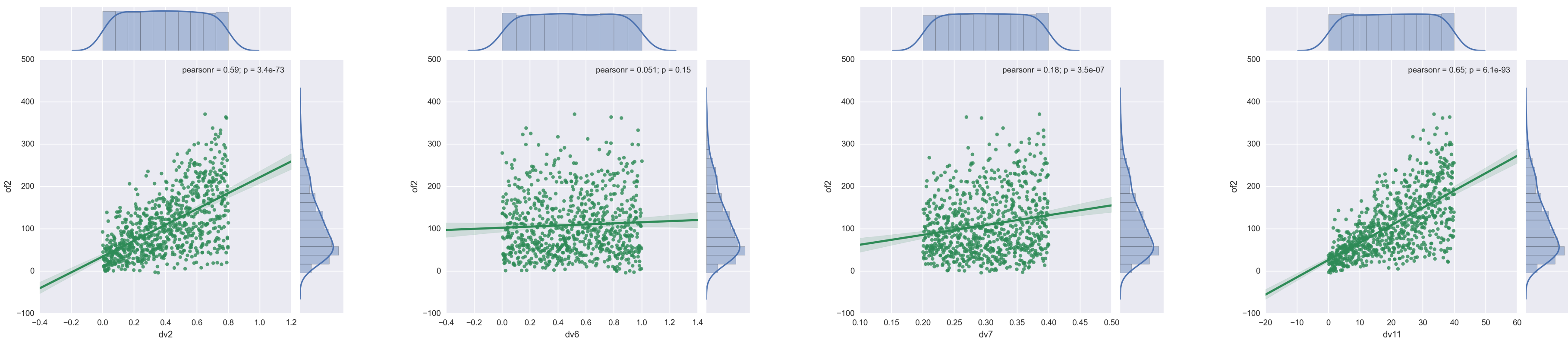
QoI (of1) vs. DV (dv2, dv6, dv7, dv11) – DACE 700



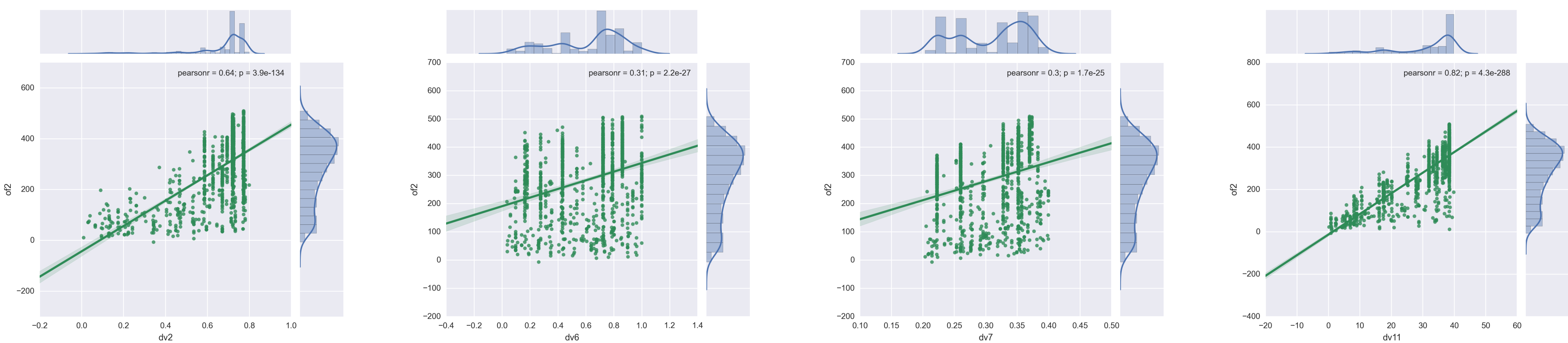
QoI (of1) vs. DV (dv2, dv6, dv7, dv11) – MOGA

Practical applications

Sailing yacht daggerboard optimization



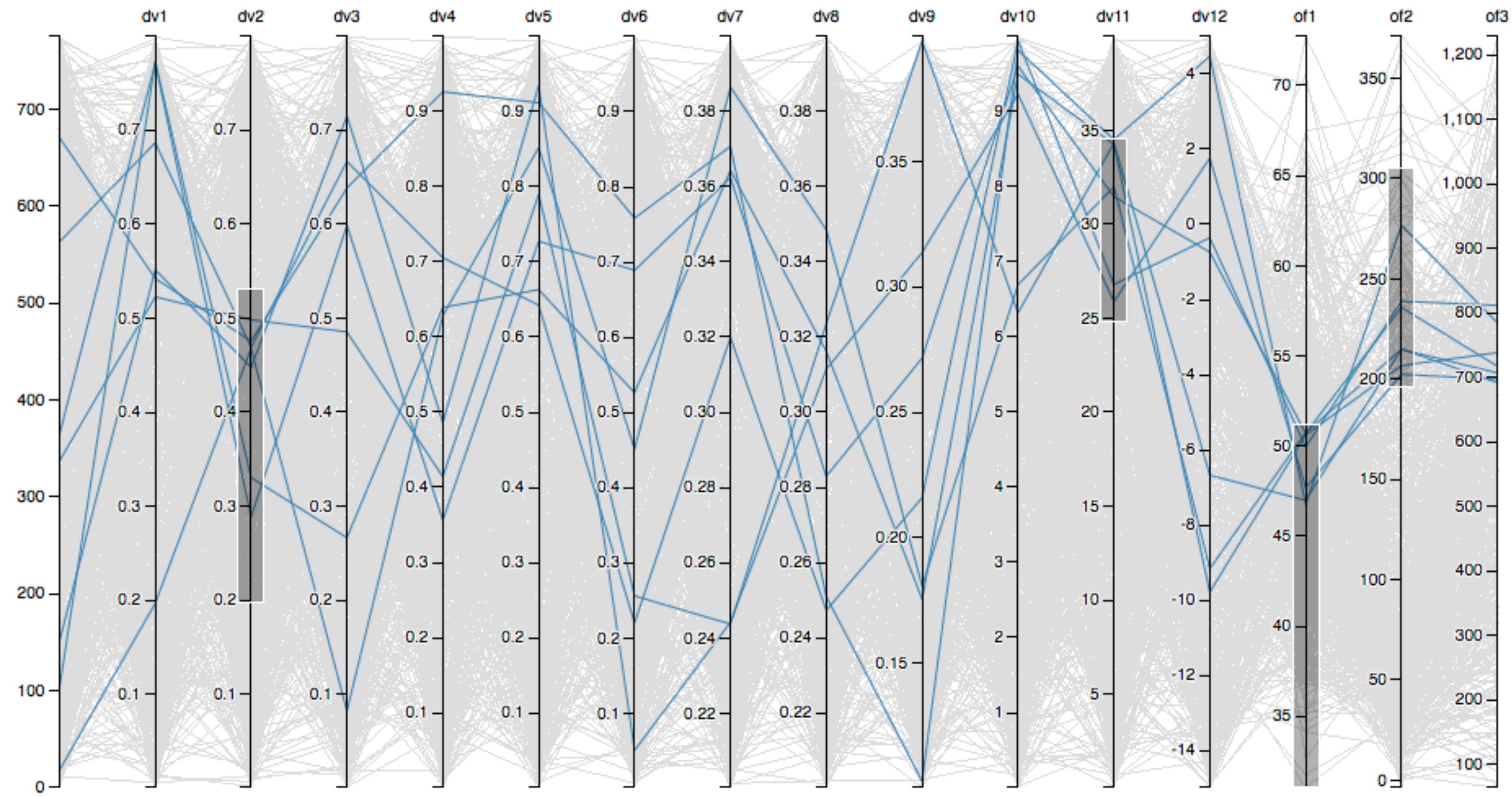
QoI (of2) vs. DV (dv2, dv6, dv7, dv11) – DACE 700



QoI (of2) vs. DV (dv2, dv6, dv7, dv11) – MOGA

Practical applications

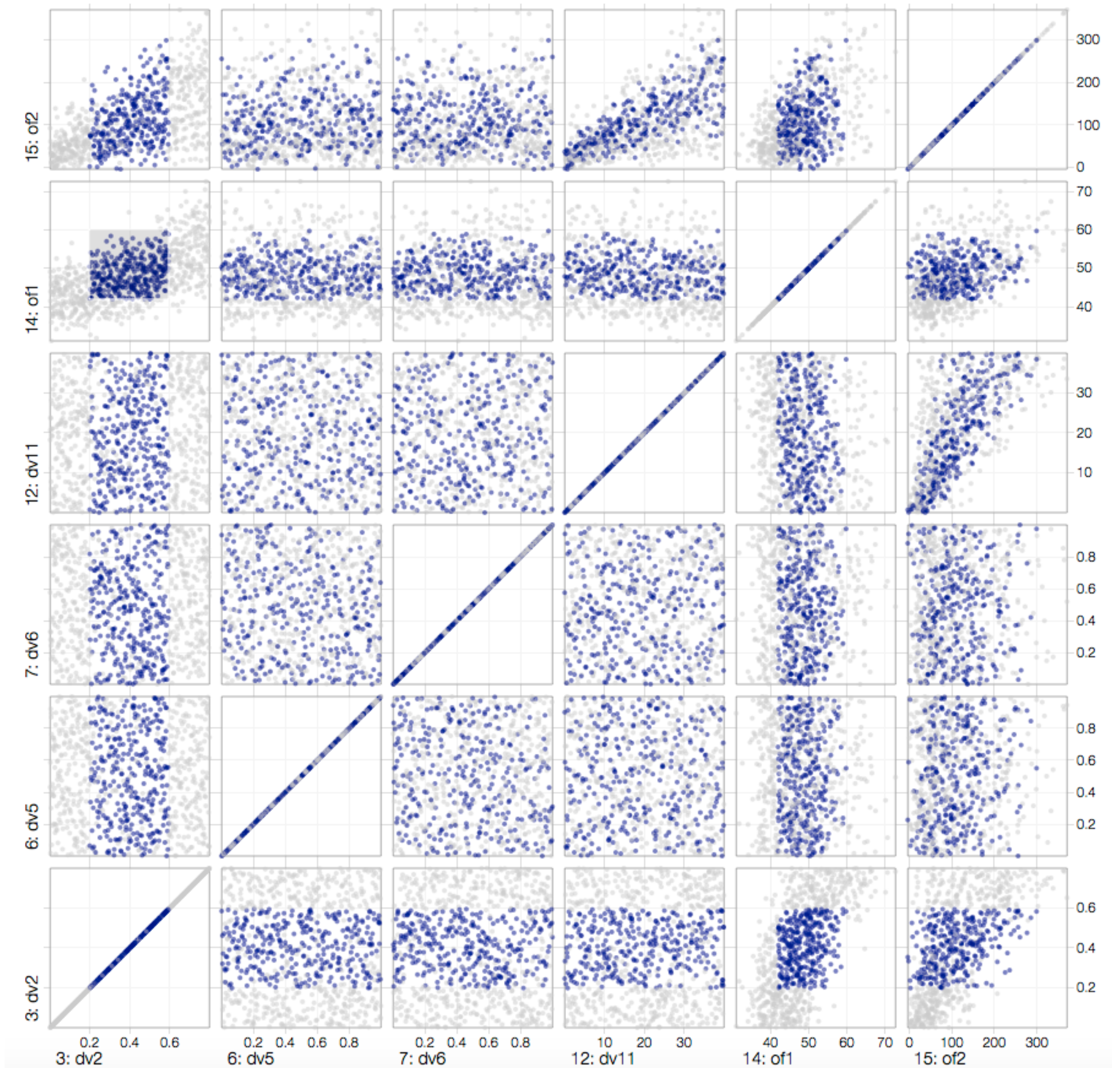
Sailing yacht daggerboard optimization



Dynamic parallel coordinates plot

Practical applications

Sailing yacht daggerboard optimization



Interactive scatterplot matrix

Roadmap

1. ~~Brief overview of optimization~~
2. ~~Surrogate based optimization~~
3. ~~The optimization driver~~
4. ~~Practical applications~~
5. **Wrap-up**

Wrap-up

To conclude

- We have effectively used an optimization framework entirely based on open-source technology.
- The tools used are capable of performing and completing general purpose applications, as well as complex engineering tasks.
- The framework can be easily automated and used in HPC environments.

Wrap-up

To conclude

- Evolutionary algorithm studies can be quite expensive due to the large number of function evaluation needed to arrive to the optimal. By using SBO, we can reduce the number of function evaluations while getting similar results.
- An added benefit of working at the surrogate level, is that by using the output of the sampling plan we can do initial screening, compute sensitivities, and explore the design space.
- The observations (or experiments), can be used to ask and answer questions about the data, this is data analytics.

Wrap-up

Ongoing work

- Currently we are working in adding advanced dataset exploration and machine learning tools to the framework.
- Improvement on the computation of basic statistics.
- Uncertainty quantification.
- Substituting all the scripts with a GUI.
- Dynamic multidimensional detective (parallel coordinates).
- Interactive data visualization and real time rendering.
- Visualization of multidimensional surrogates.
- All the visualization will be implemented using Python and D3.js

Wrap-up

Future developments (or good intentions)

- Efficient shape and mesh morphing tools.
- Mesh smoothing and mesh morphing using MESQUITE.
- An intuitive GUI for code coupling.
- A web interface for data analytics.
- Adjoint optimization.

Thank you for your attention

Questions?