A multi-fidelity method for uncertainty quantification in engineering problems

L. Tamellini

Institute for Applied Mathematics and Information Technologies “E. Magenes” (CNR-IMATI), Pavia, Italy

The Uncertainty Quantification framework

Parameters $\boldsymbol{y}$ ⟹ PDE/ODE ⟹

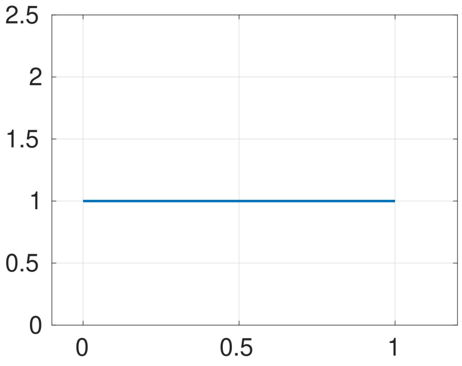
- Solution $u(\boldsymbol{x},\boldsymbol{y})$
- Solution functional $F(\boldsymbol{y})$

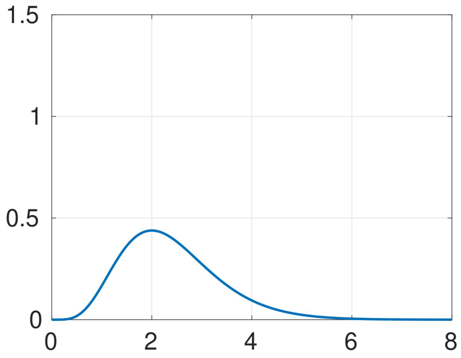
- ***y*** **are random/uncertain**
- Then, $u(\boldsymbol{y})$ *and* $F(\boldsymbol{y})$ are random quantities
- **What is the variability of *u and F* wrt to *y* ?**

The Uncertainty Quantification framework

Parameters ***y*** → PDE/ODE →

- Solution $u(\boldsymbol{x},\boldsymbol{y})$
- Solution functional $F(\boldsymbol{y})$

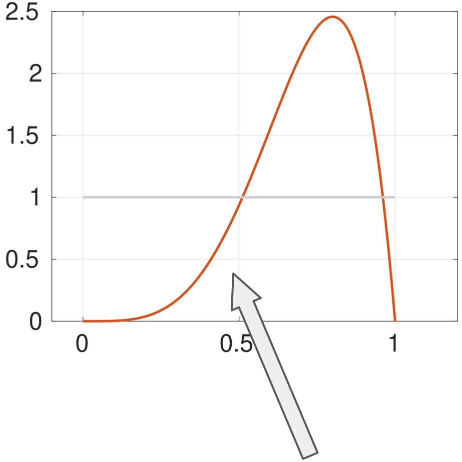


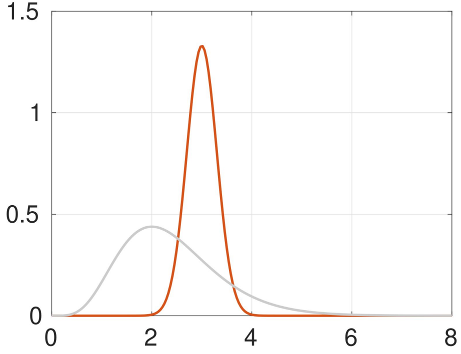


The Uncertainty Quantification framework

Parameters ***y*** ⟹ PDE/ODE ⟹

- Solution *u*(***x***,***y***)
- Solution functional *F*(***y***)





Data-informed pdf, coming e.g. from sequential monte carlo, MCMC, etc

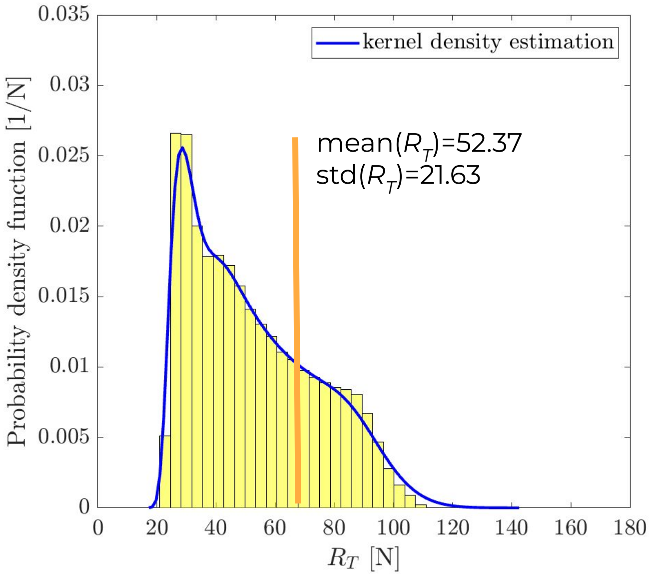
Different kind of analysis are possible

- **Forward UQ (from inputs to outputs):**
  Compute mean, variance, quantiles, probability density function (pdf) of $F(\boldsymbol{y})$

- **Inverse UQ (aka calibration: from “uninformed” to “data-aware” pdf)**
  Can we reduce the uncertainty on $\boldsymbol{y}$ if we measure F?

Example 1: forward UQ for a ferry

- Two operational **uncertain parameters:**
  - **Speed** within the operational range
  - **Draught** ±10% design (±15% design payload)
- F(**y**) = resistance to advancement (ship drag)
- PDE: Navier-Stokes (RANS solver)

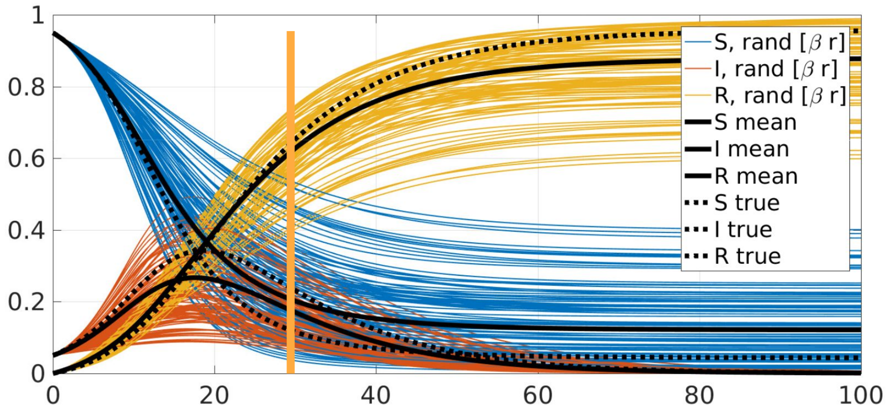


Example 2: inverse + forward UQ for SIR

- Two **uncertain parameters**
  - **Contact probability,** β
  - **Recovery time**, r
- ODE: SIR system

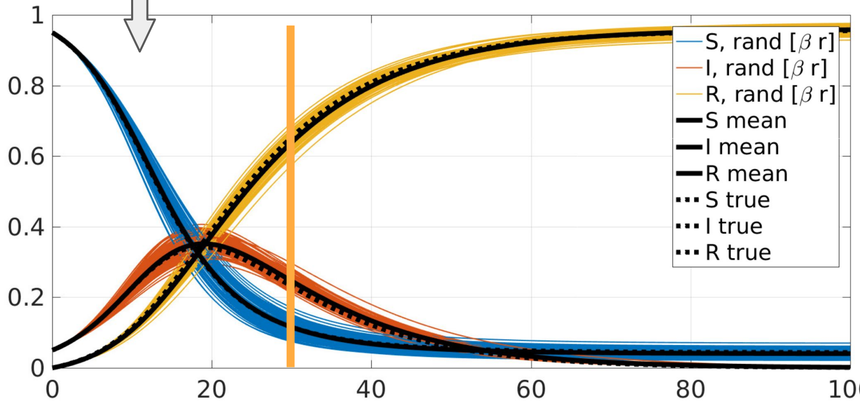
We have seen these plots already

no data are available



β in [0.25, 0.35], r in [0.06,0.18] (literature)

with data on I,R up to t=30



β ~ N(0.285,0.80), r ~ N(0.086,0.26)

Solve, solve, solve!

Most forward / inverse techniques boil down to **repeatedly solving the ODE/PDE** for multiple values of **y** (**sampling**)

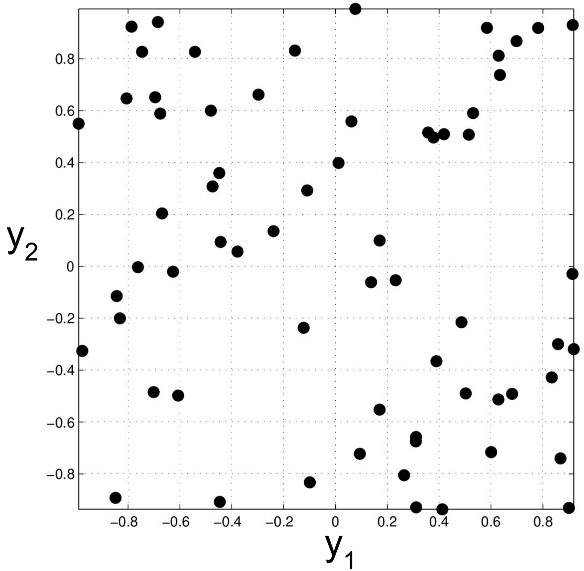
How many samples? For **what values** of y?

Solve, solve, solve!

Most forward / inverse techniques boil down to **repeatedly solving the ODE/PDE** for multiple values of **y** (**sampling**)

How many samples? For **what values** of y?

Alternative 1: Monte Carlo: Robust but **slow,** *Error* $\approx M^{-1/2}$

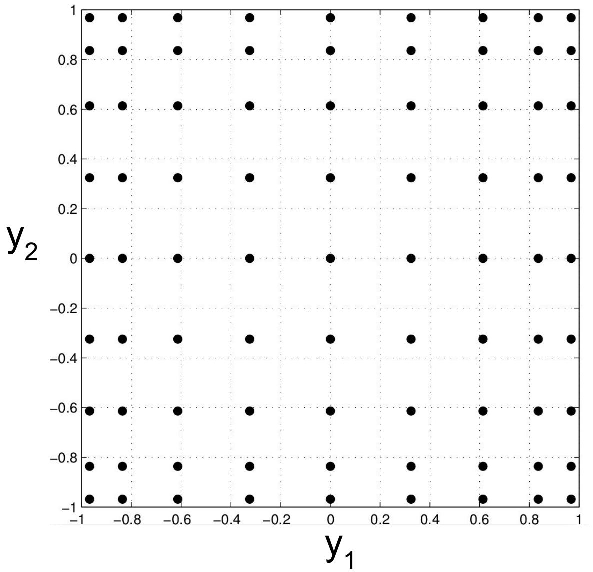


Solve, solve, solve!

Most forward / inverse techniques boil down to **repeatedly solving the ODE/PDE** for multiple values of **y** (**sampling**)

How many samples? For **what values** of y?

Alternative 2: Cartesian grid. More accurate but **expensive**: $M = M_0^N$ **N can be large!!!**

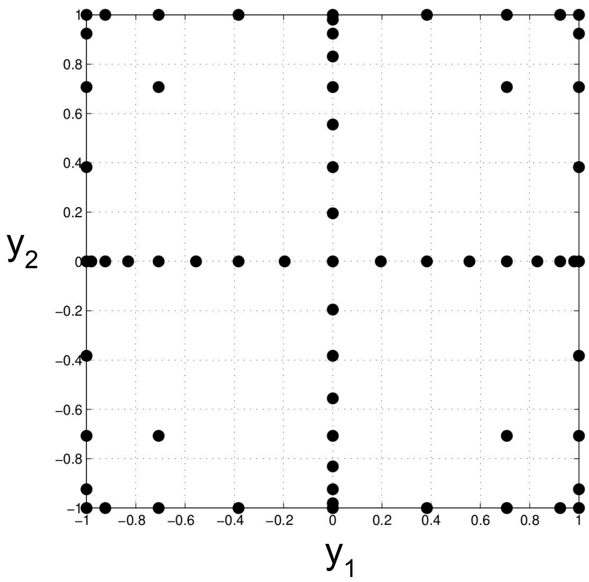


Solve, solve, solve!

Most forward / inverse techniques boil down to **repeatedly solving the ODE/PDE** for multiple values of **y** (**sampling**)

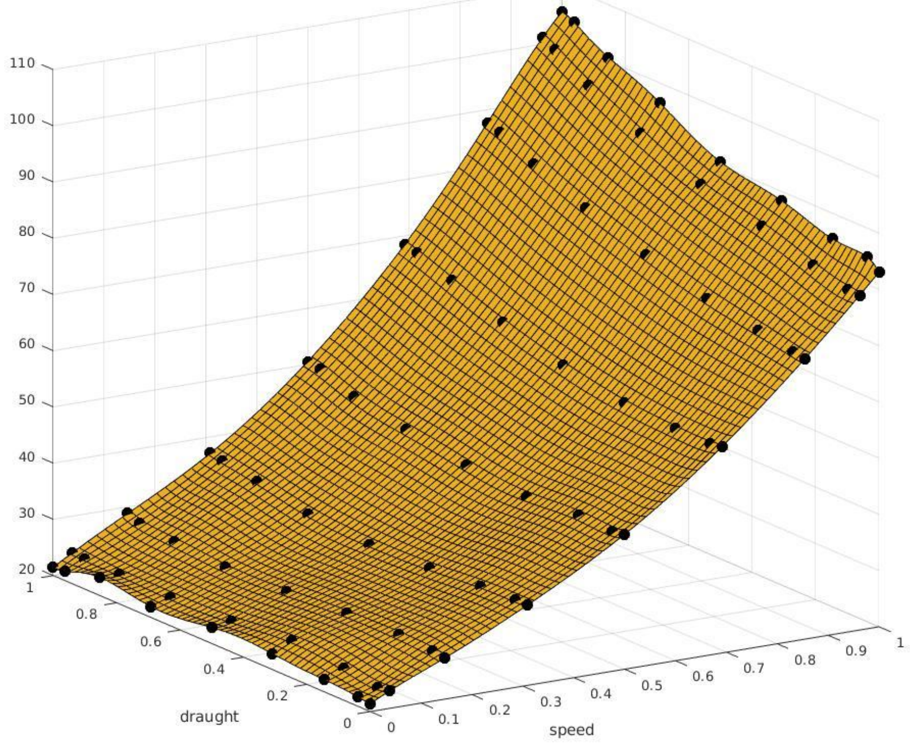
How many samples? For **what values** of y?

Alternative 3: Sparse grid and other advanced sampling



Make it faster: surrogate modeling

- Instead of solving the PDE for all value of ***y***
    a. Solve for a few "selected" ***y***
    b. "Interpolate" the values of *F*(***y***)
    c. Evaluate the surrogate model: **much cheaper!**
    d. Works for **smooth functions** *F*(***y***)



- Many alternatives:
    a. Polynomial Chaos Expansion
    b. Sparse Grids
    c. Reduced Basis
    d. Proper Orthogonal Decomposition
    e. Radial Basis Functions
    f. Gaussian Processes
    g. Neural Networks
    h. ...

Multi-fidelity

- Consider a **hierarchy of approximations** of the same ODE/PDE:
    a. Different **discretizations**
    b. Different **physics:** Euler / Stokes / RANS / Direct Navier Stokes

- Explore the “bulk” of the variability due to y with **many queries of the cheap models** …

- … and correct with a **handful of queries of the high-fidelity models**

- Can (should) be combined with the **surrogate-modeling** paradigm

Summary of UQ framework

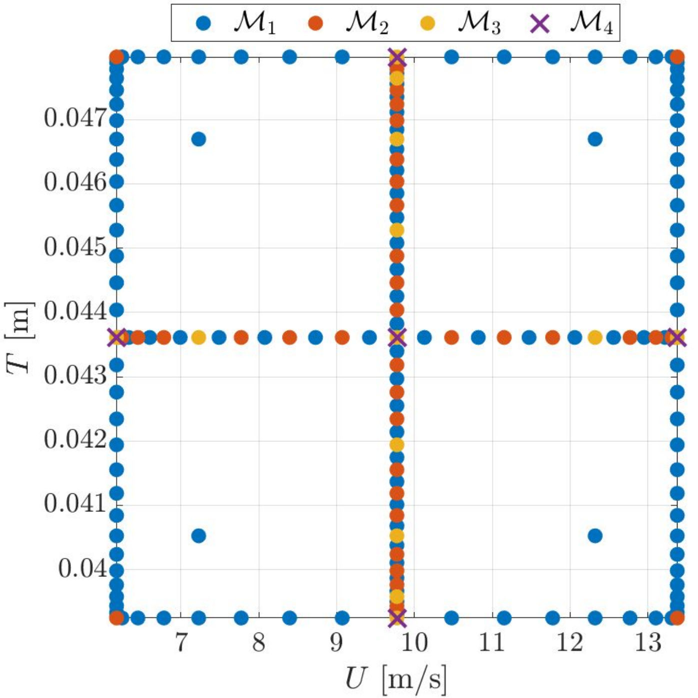
The function to sample is:

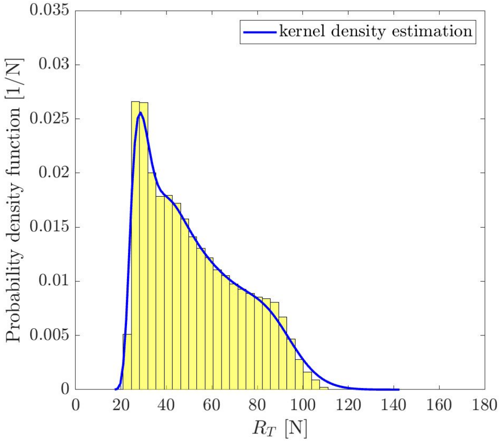
- Implicit & expensive
- high-dimensional

We need:

- Efficient sampling schemes + surrogate models to generate quickly many values
- Multi-fidelity approach

See the big picture here:





… and now, details

Problem setup

Parameters ***y***
- Speed
- Draught (~ payload)

⟹

Unsteady Navier-Stokes equations

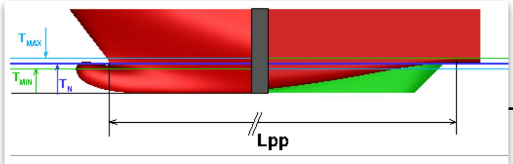
⟹

Advancement resistance R_T

Two operational **uncertain parameters y=[U, T]**:

- **Speed U** within the operational range
- **Draught T** ±10% design (±15% design payload)

We assume **y_1, y_2** to be **uniform random variables**



		Full Scale	Model Scale
Length Between Perpendiculars	L_{PP}	162.85 m	6.0 m
Speed Range	U	[12÷26] knots [6.2÷13.4] m/s	[1.2÷2.6] m/s
Draught Range	T	[6.4÷7.8] m 7.1 m (design)	[0.24÷0.29] m

Our RANS software:

Parameters $\boldsymbol{y}$
- Speed
- Draught (~ payload)

⟹

Unsteady Navier-Stokes equations (**RANS**)

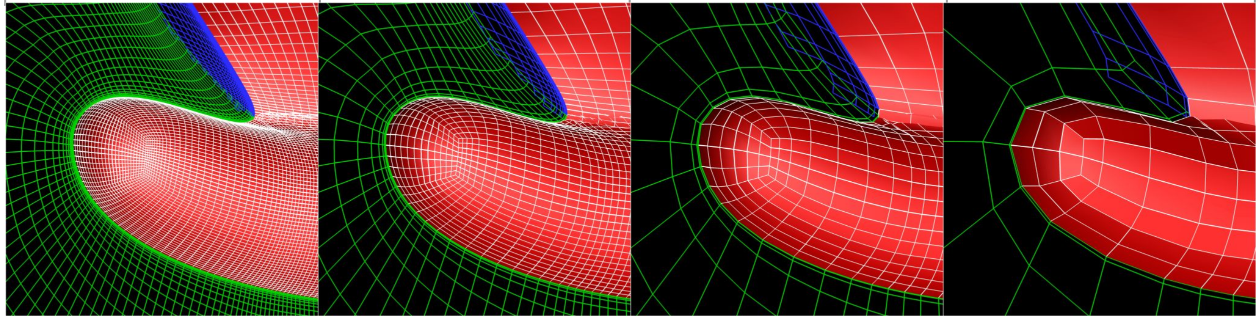
⟹

Advancement resistance R_T

Reynolds Averaged Navier-Stokes (RANS) solver ***X-navis***, developed at CNR-INM

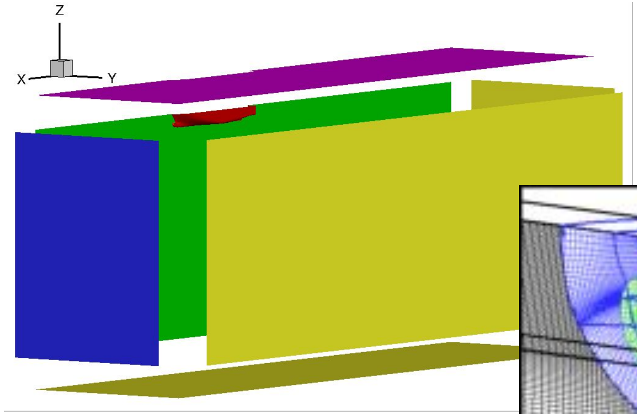
- 2nd-order finite volumes scheme
- **Multi-grid solver**
- **4 grids** by “dyadic derefinement” of an initial fine grid

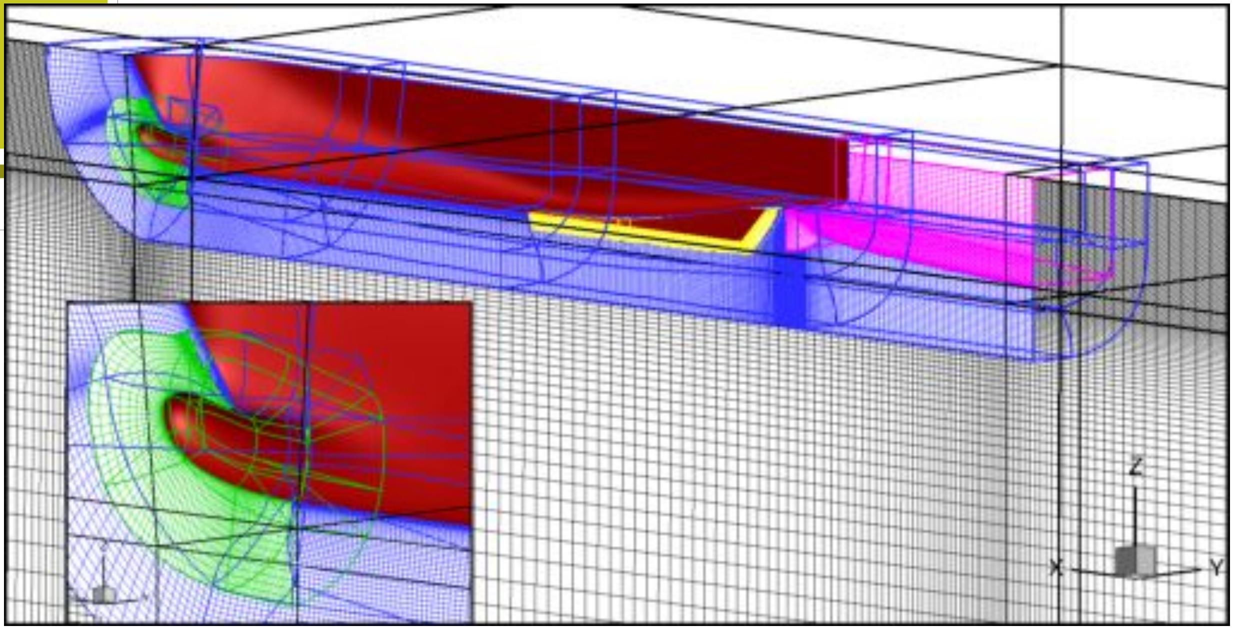
M_4 (5.5M, cost = 512)	M_3 (699k, cost = 64)	M_2 (87k, cost = 8)	M_1 (11k, cost = 1)



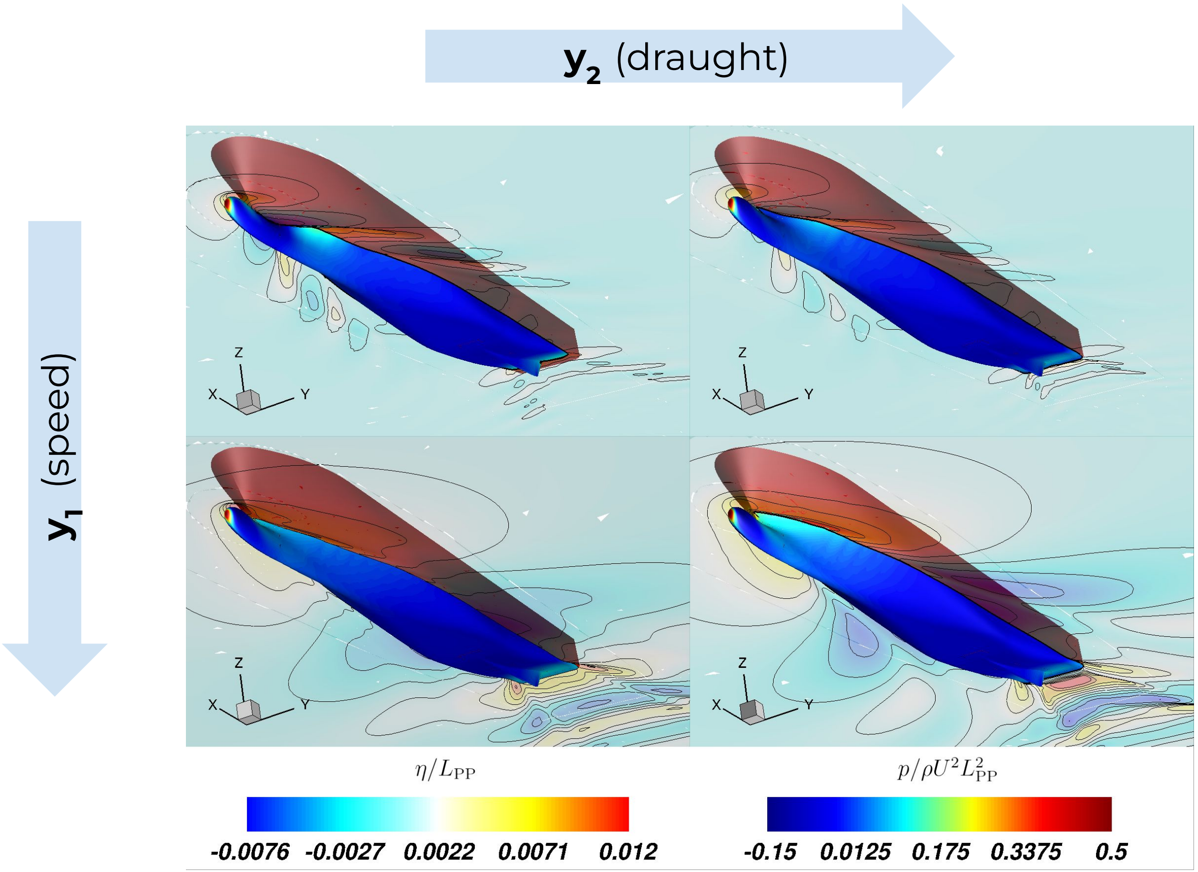
R_{T_4} R_{T_3} R_{T_2} R_{T_1}

Domain and full mesh

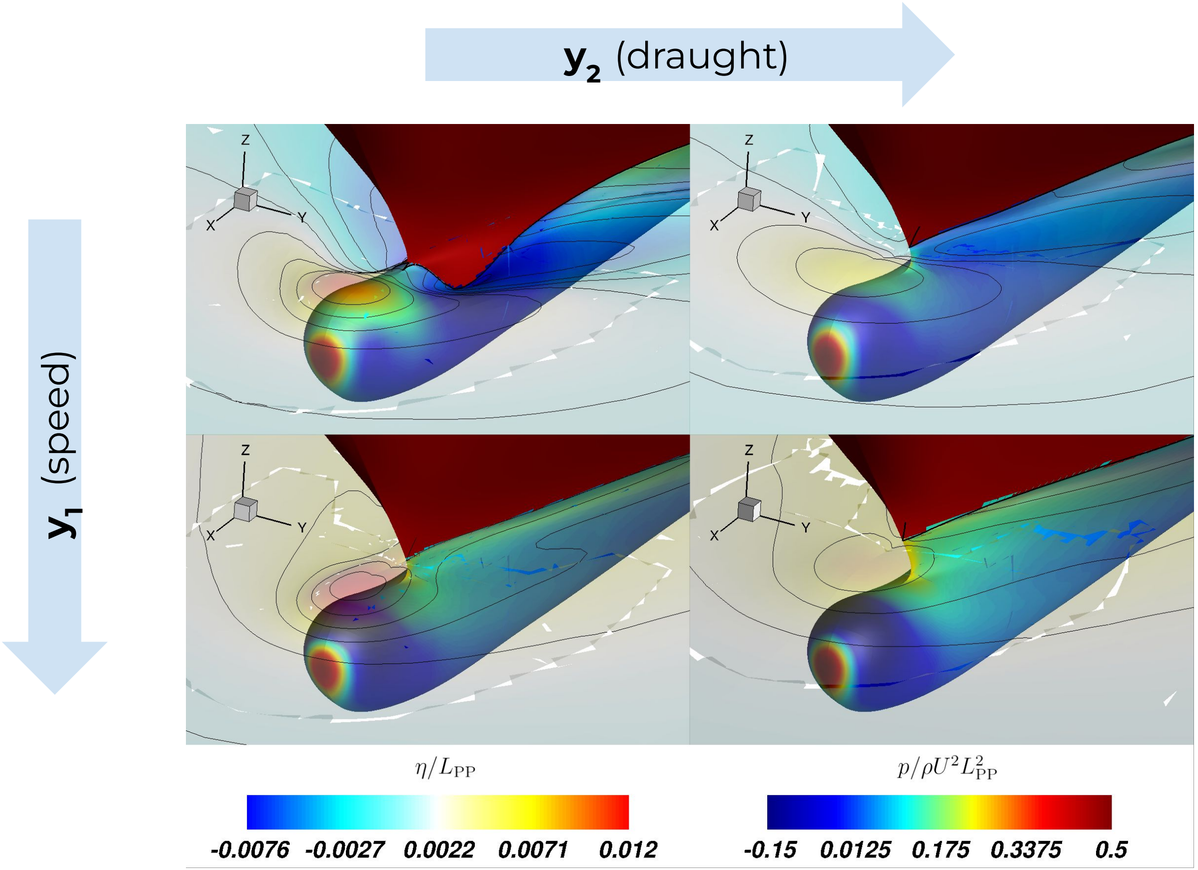




Some simulations

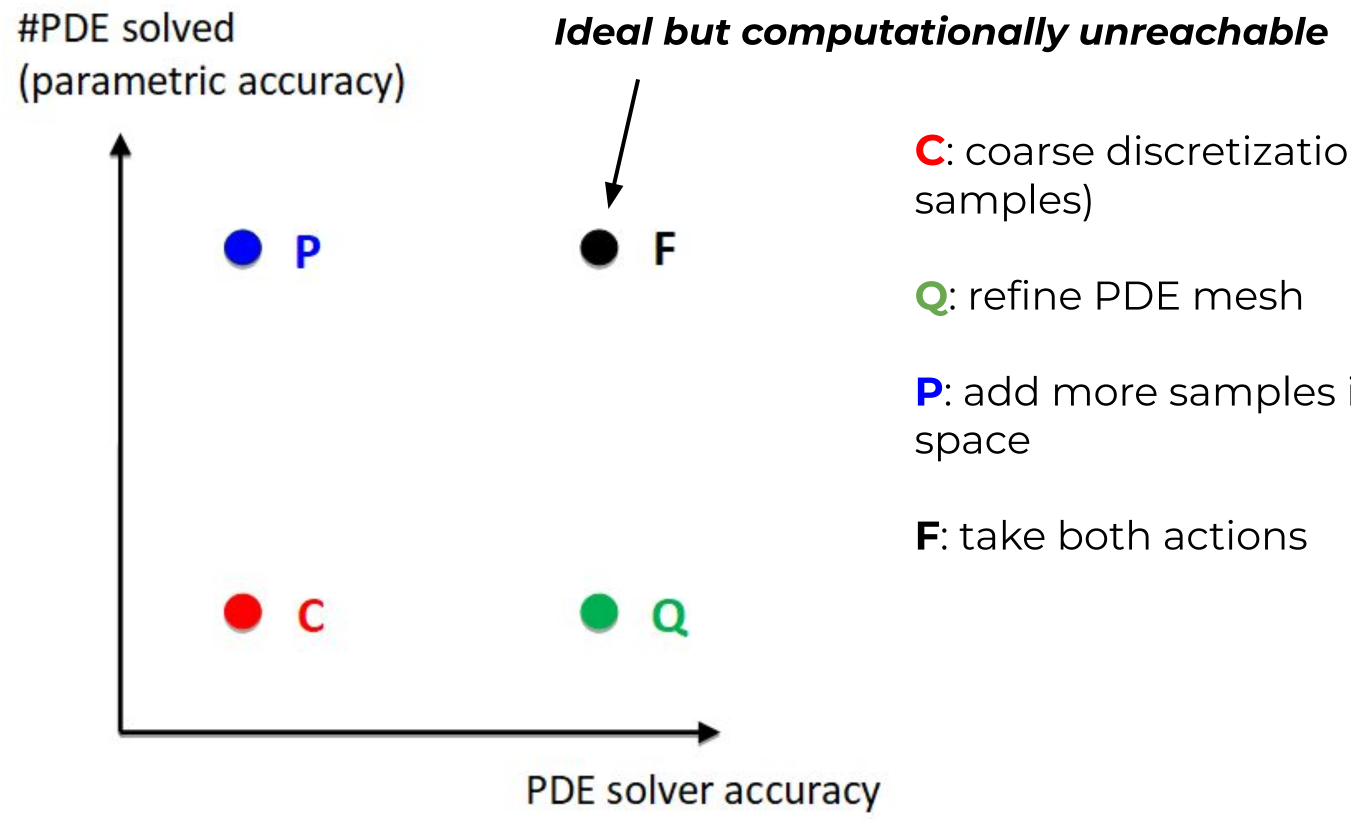


Some simulations (zoom)



Our multi-fidelity method:
Multi-index Stochastic Collocation (MISC)

The sparsification principle



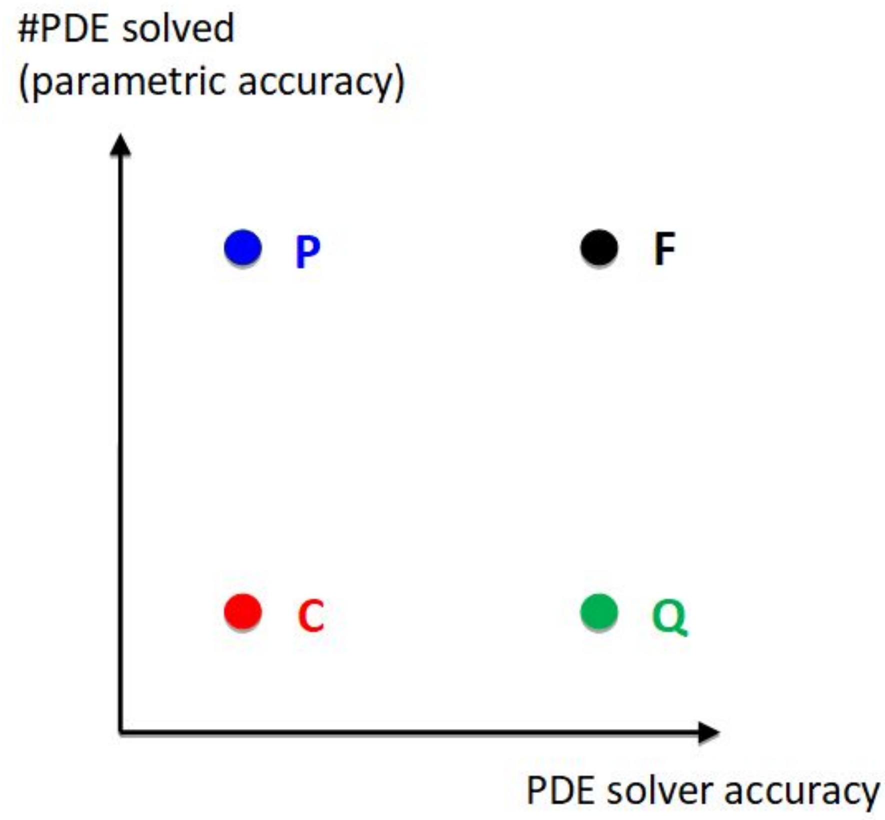
C: coarse discretization (coarse mesh, few samples)

Q: refine PDE mesh

P: add more samples in the parameter space

F: take both actions

The sparsification principle



Write a **telescopic equality**

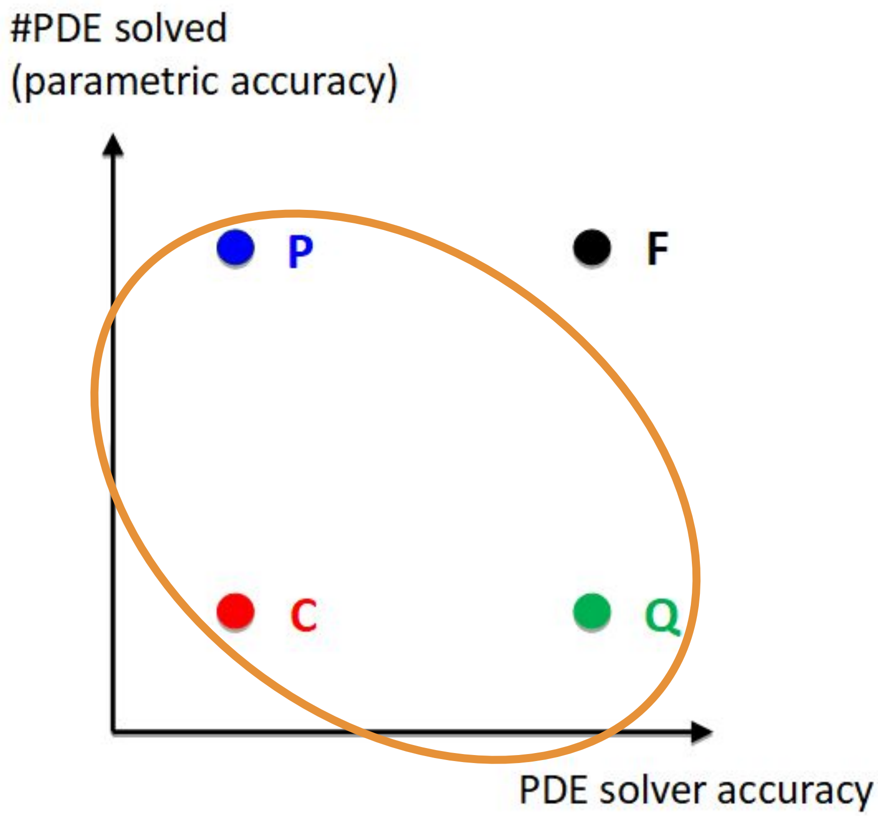
F = **F** + **C** − **C**
+ **C** − **C**
+ **P** − **P**
+ **Q** − **Q**

Rearrange to put in evidence three corrections

F = ~~**C**~~
+ **P** − ~~**C**~~
+ **Q** − **C**
+ ~~**F** − **P** − **Q** + **C**~~

2nd order correction,
small but expensive!
(under regularity assumptions)

The sparsification principle



MISC formula:

$$\mathbf{F} \approx \mathbf{P} + \mathbf{Q} - \mathbf{C}$$

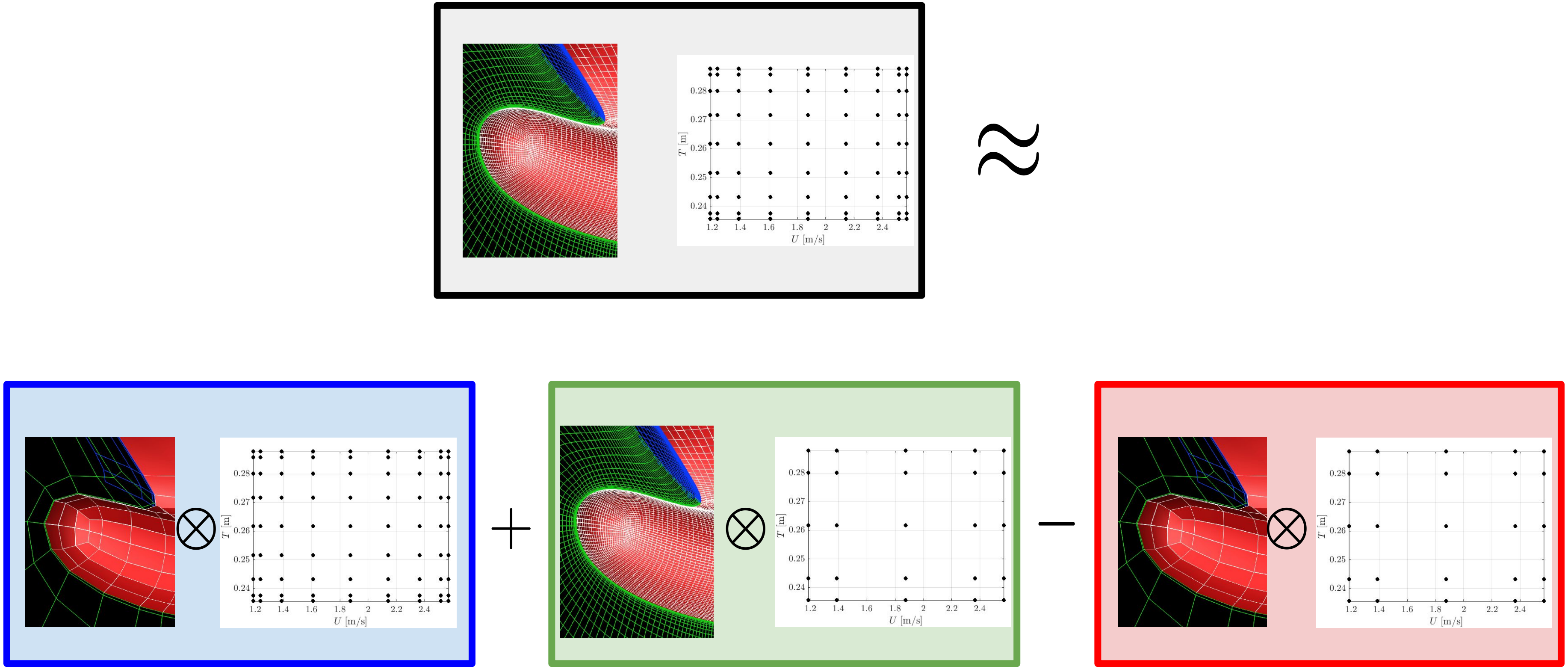
i.e., a **linear combination of three different (cheap) approximations**

Sparsification principle achieved!

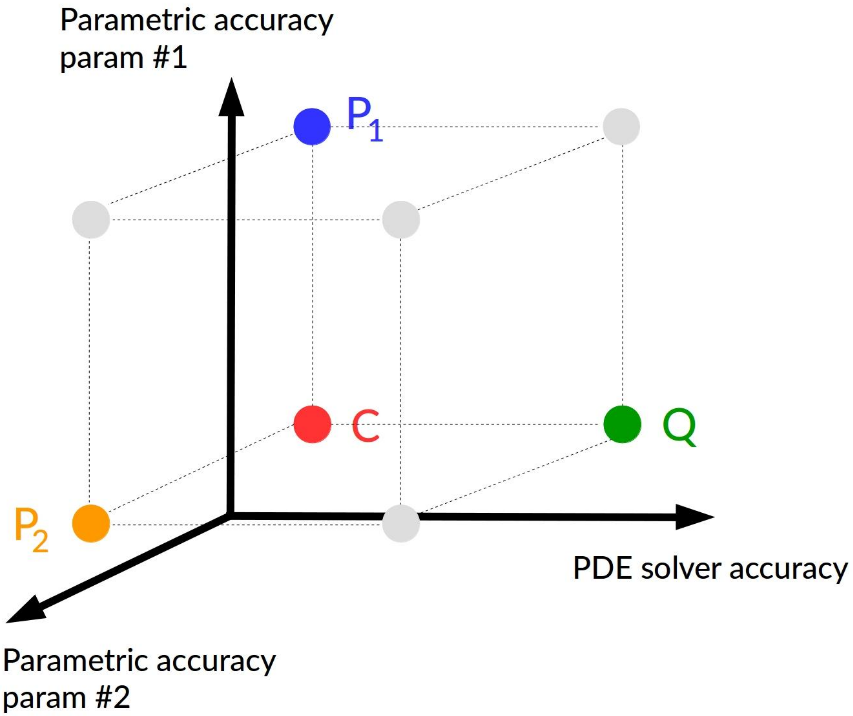
Solve many PDEs on a coarse mesh and few PDEs with refined mesh

Bonus: **balance errors** in solver and parametric space!

So, in the end:



MISC for multiple parameters: y_1, y_2



If we have **two parameters** we can separate also the two decisions:

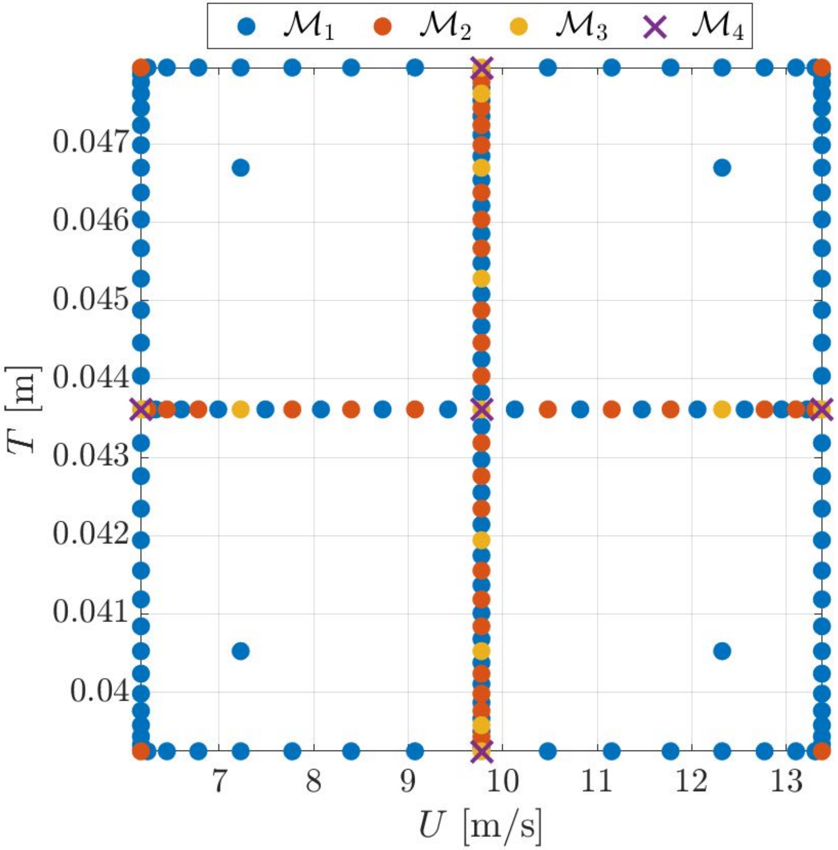
- Refine the sampling of y_1 (**P_1**)
- Refine the sampling of y_2 (**P_2**)

Delay refining both simultaneously!

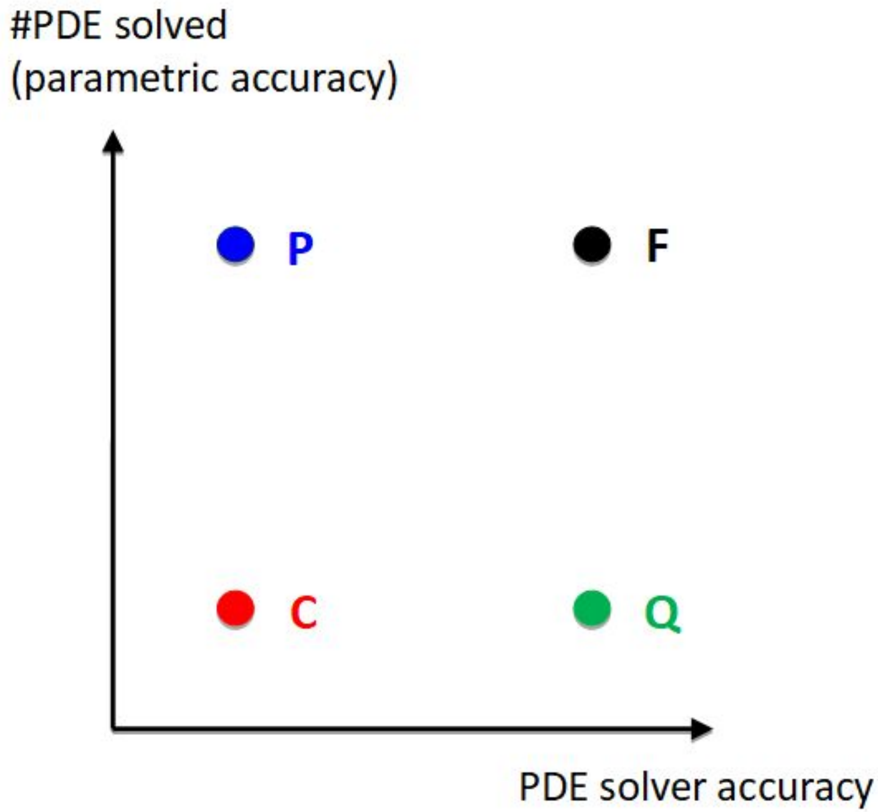
The parameter space is sampled in a **structured but not tensor** way

Bonus: if you have a tensorized solver **(IGA / FD)** you can do the same for the solver!

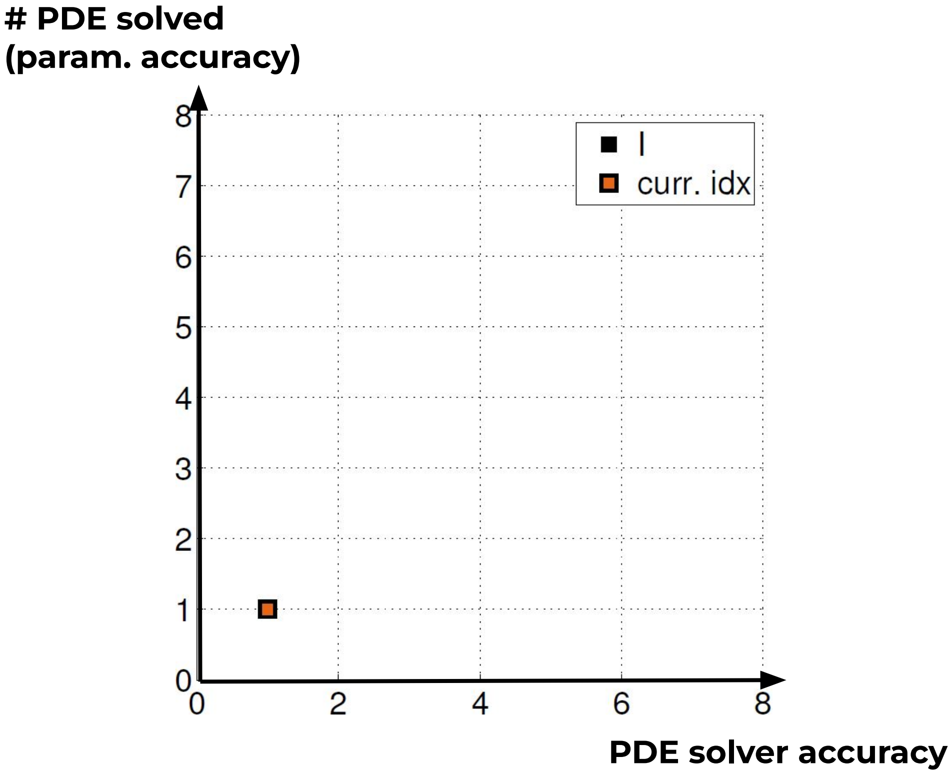
MISC samples and solver calls



MISC: an adaptive algorithm



MISC: an adaptive algorithm

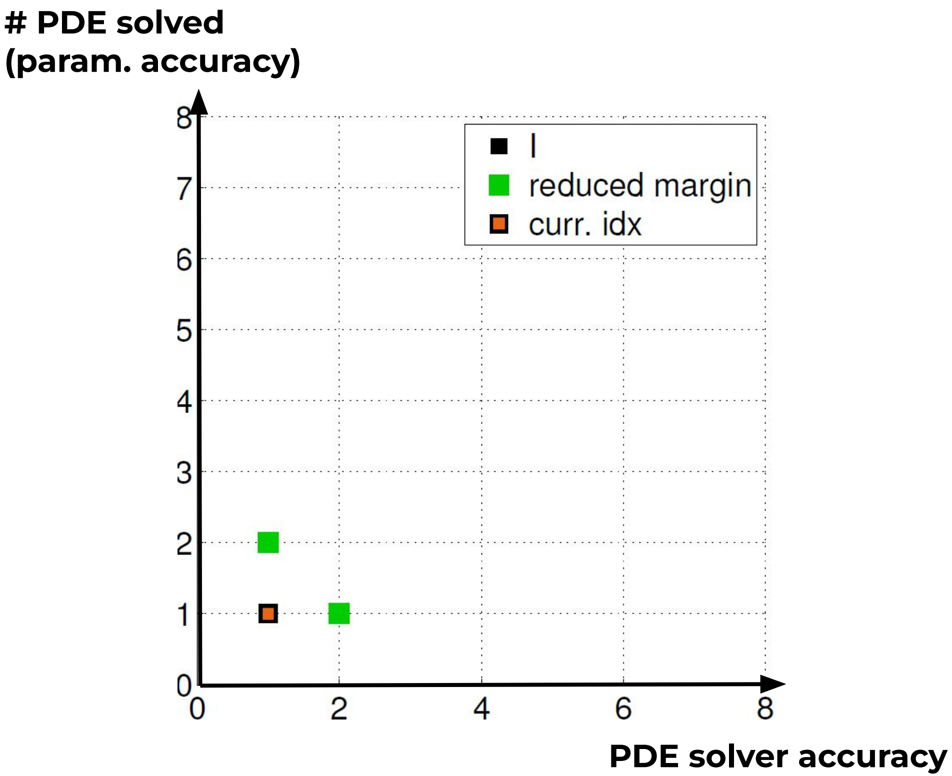


1. Add “**neighboring refinements**”
2. Choose the one that gave the **best improvement**:

   Δ[Expected value] / cost

3. Repeat

MISC: an adaptive algorithm

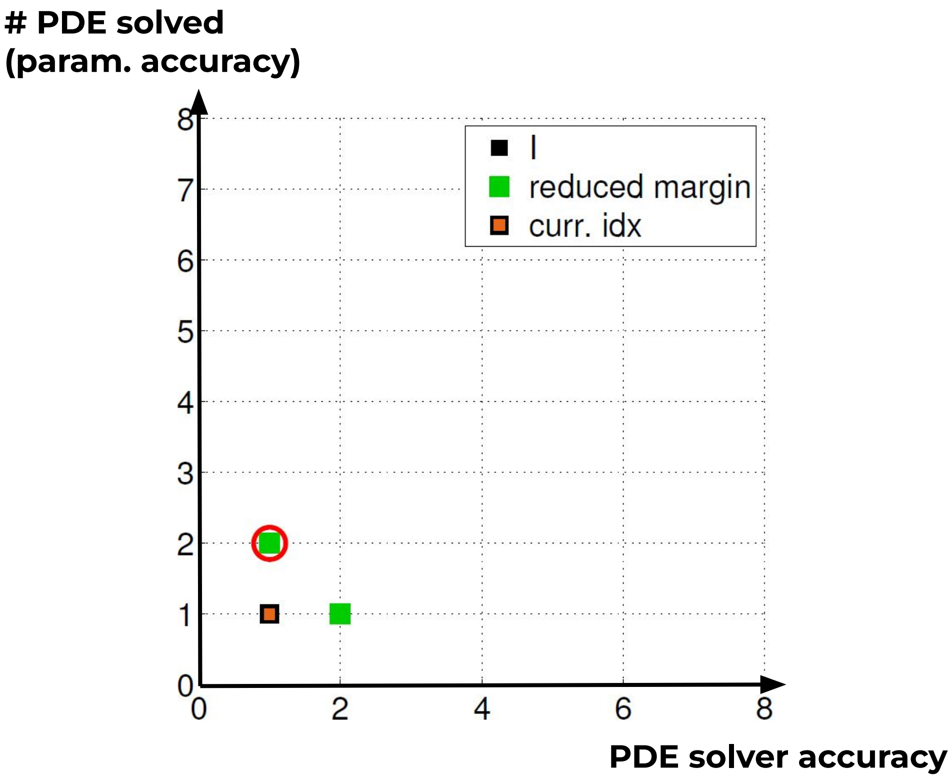


1. Add “**neighboring refinements**”
2. Choose the one that gave the **best improvement**:

   Δ[Expected value] / cost

3. Repeat

MISC: an adaptive algorithm

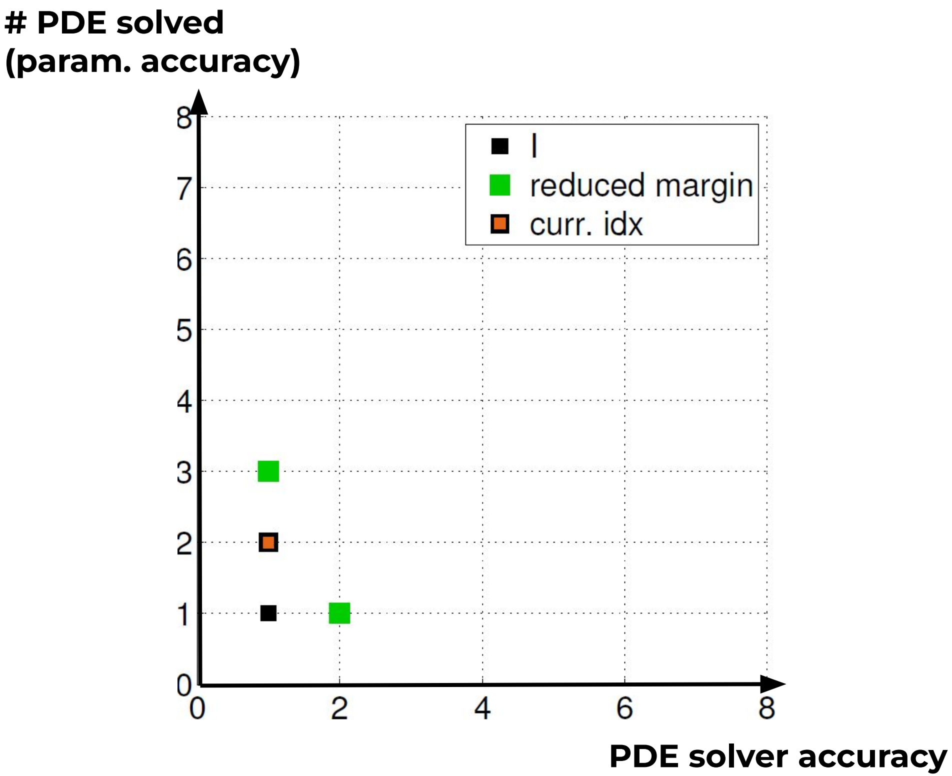


1. Add "**neighboring refinements**"
2. Choose the one that gave the **best improvement**:

   Δ[Expected value] / cost

3. Repeat

MISC: an adaptive algorithm

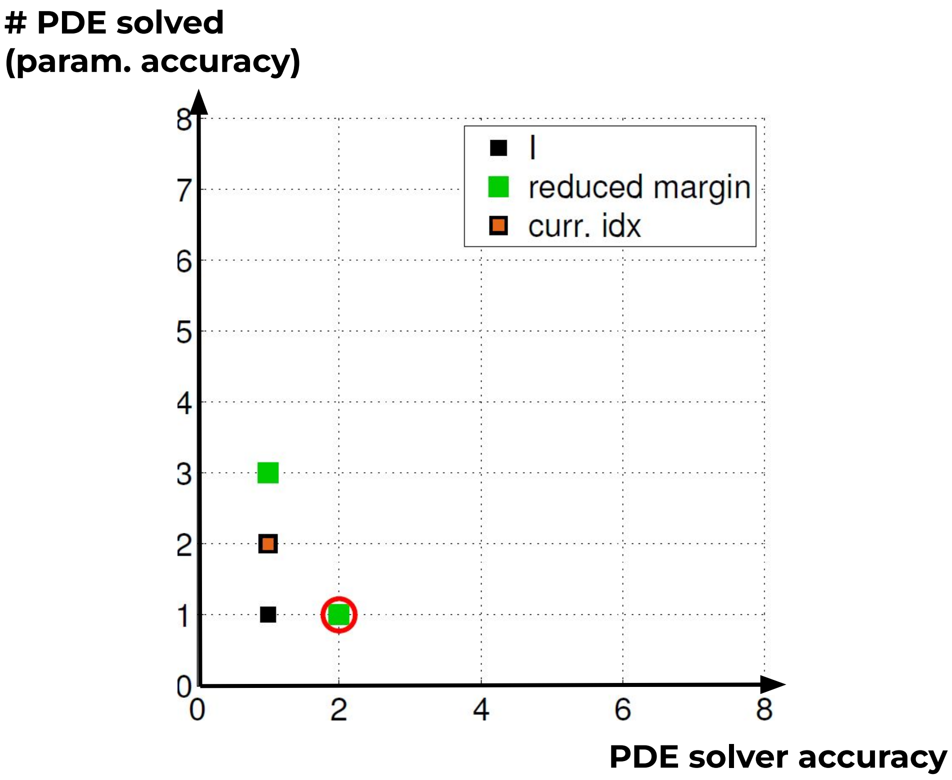


1. Add “**neighboring refinements**”
2. Choose the one that gave the **best improvement**:

   Δ[Expected value] / cost

3. Repeat

MISC: an adaptive algorithm

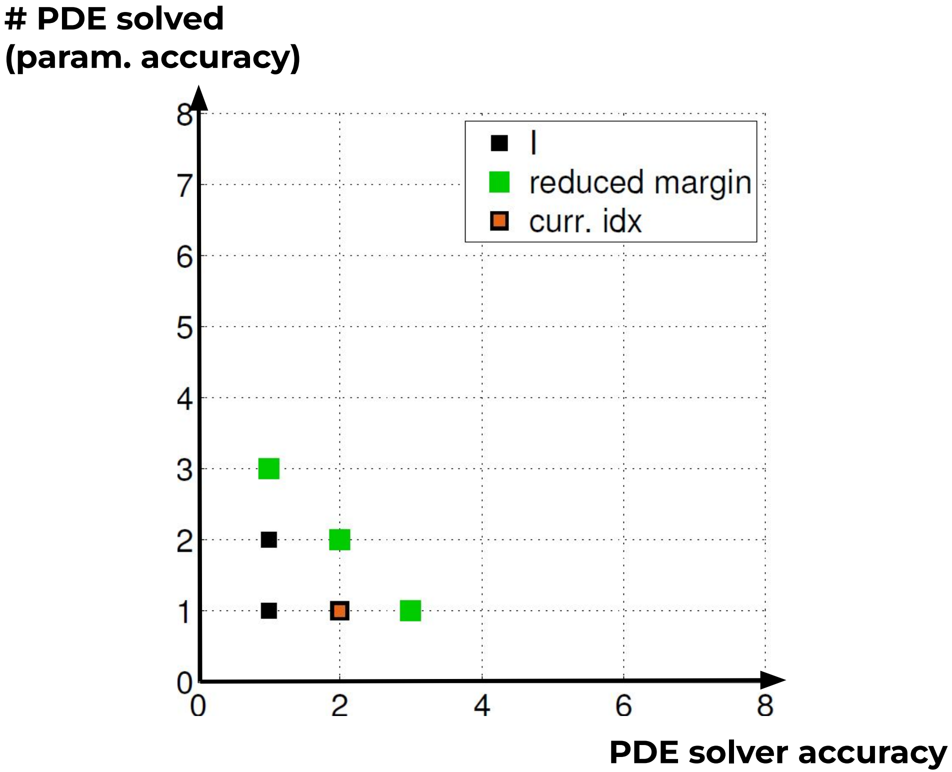


1. Add “**neighboring refinements**”
2. Choose the one that gave the **best improvement**:

   Δ[Expected value] / cost

3. Repeat

MISC: an adaptive algorithm



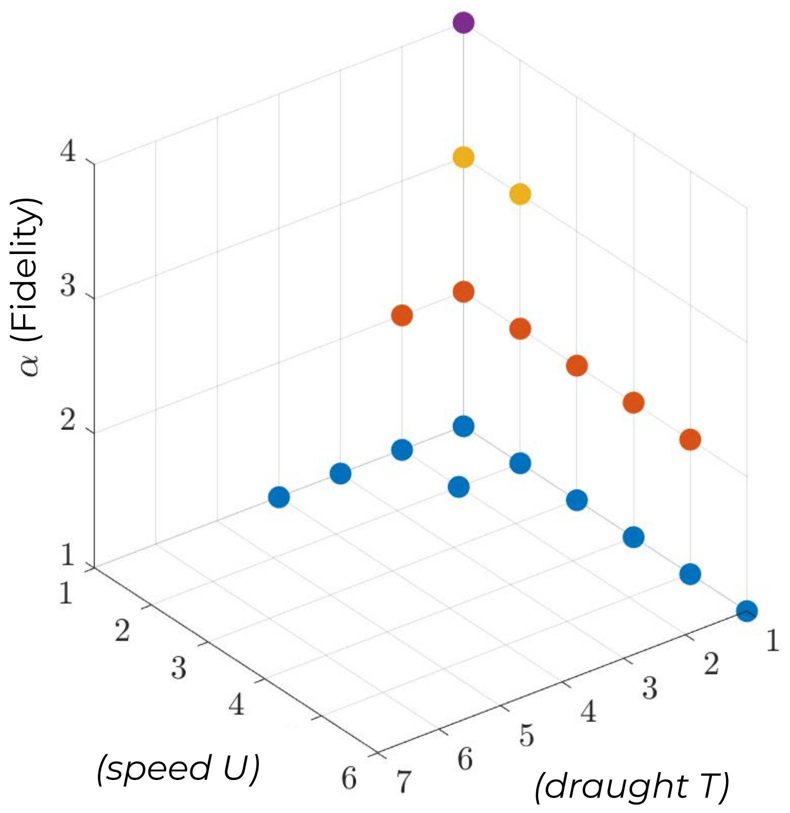
1. Add “**neighboring refinements**”
2. Choose the one that gave the **best improvement**:

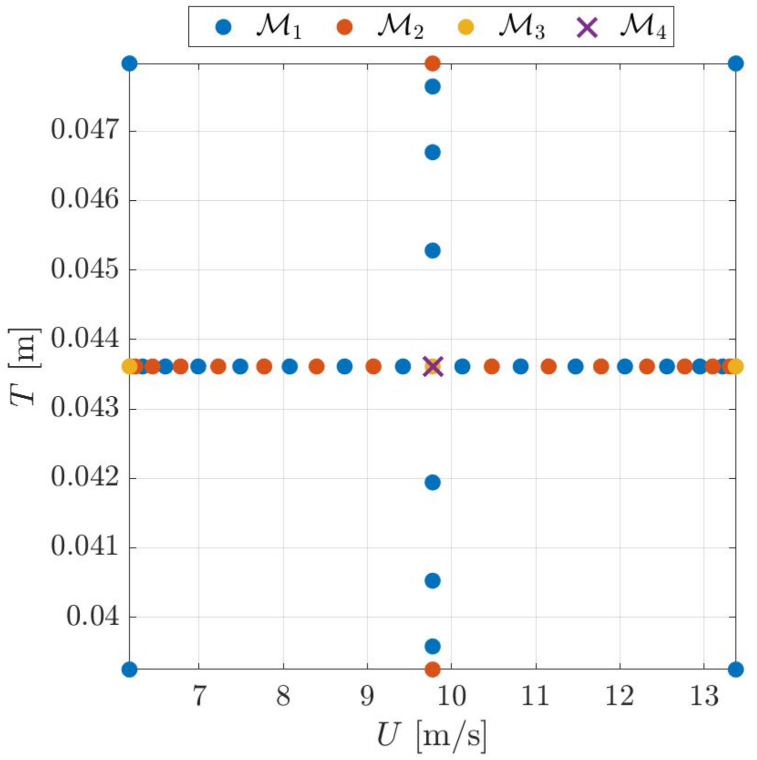
   Δ[Expected value] / cost

3. Repeat

Comments:

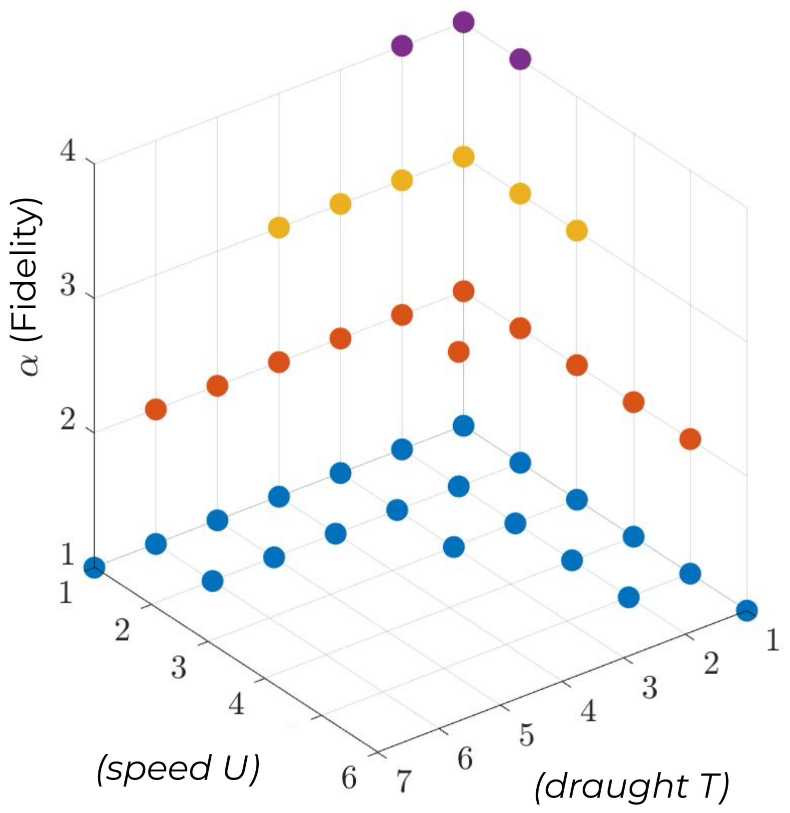
- Expensive (a-posteriori)
- Can we design the set a-priori?
- Other criteria are possible
- Points in [U, T] depend on their pdf



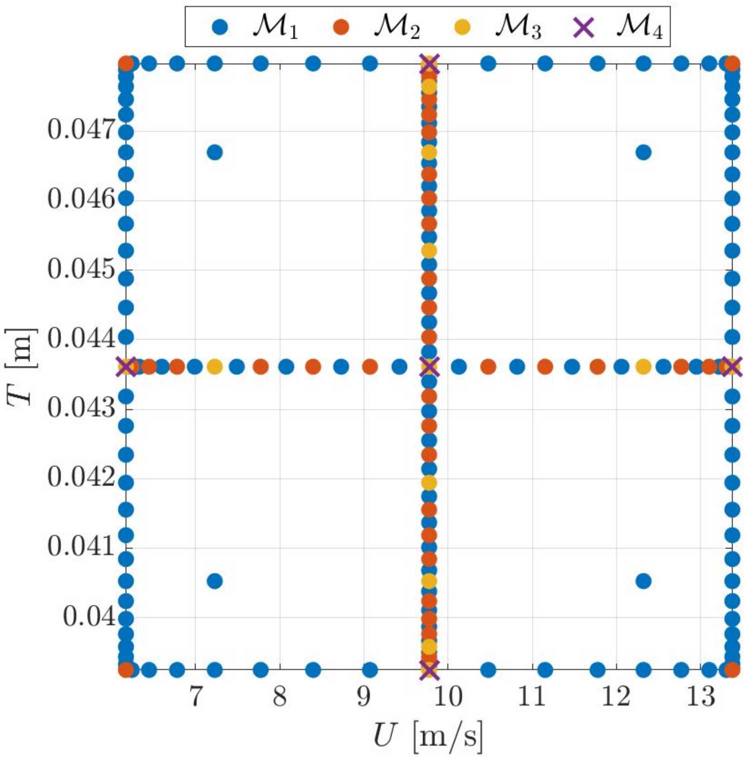


Multi-indices

Sampling



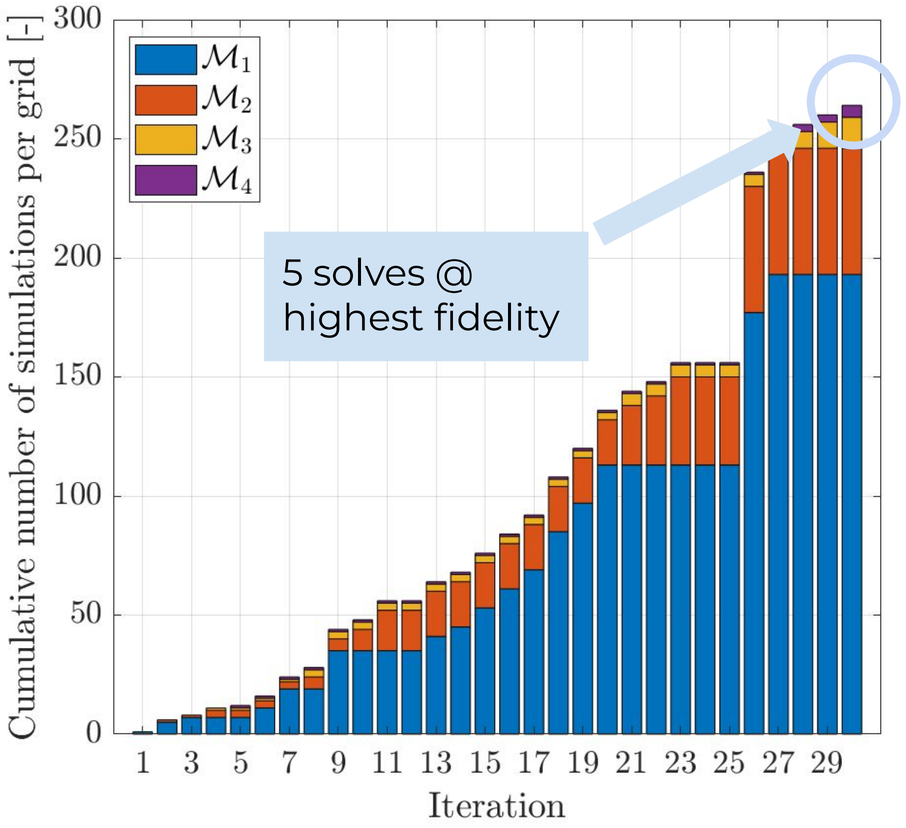
Multi-indices



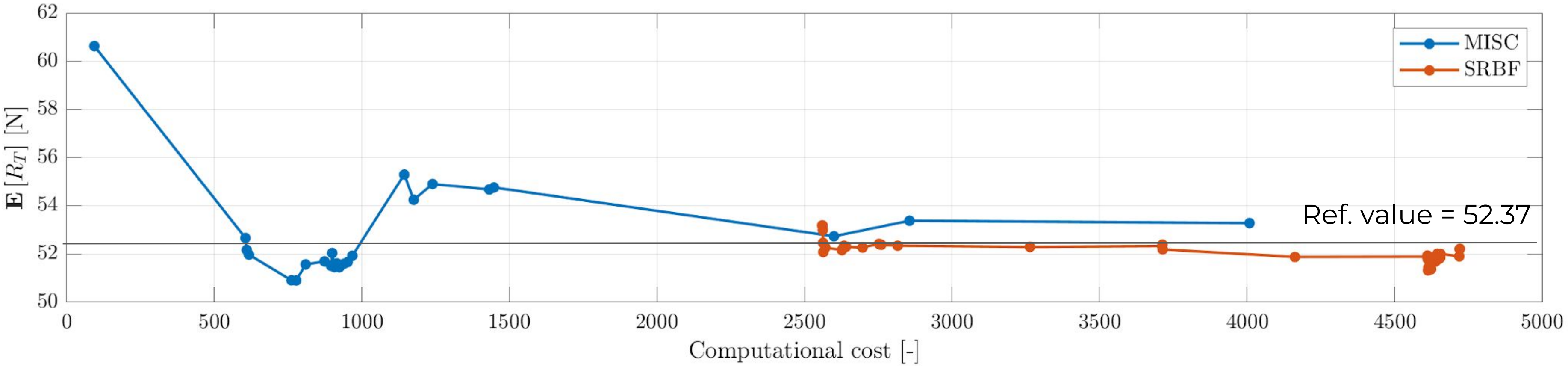
Sampling

Numerical Results

Growth of computational work

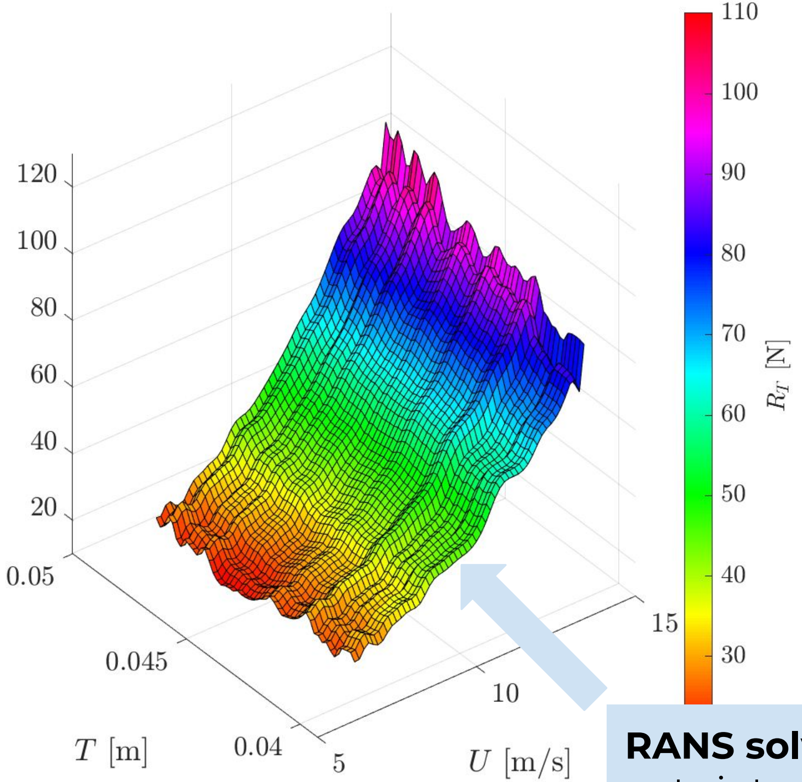


Approximation of E[R_T]



- Low entry cost
- Gets good estimates quickly
- “stagnation”

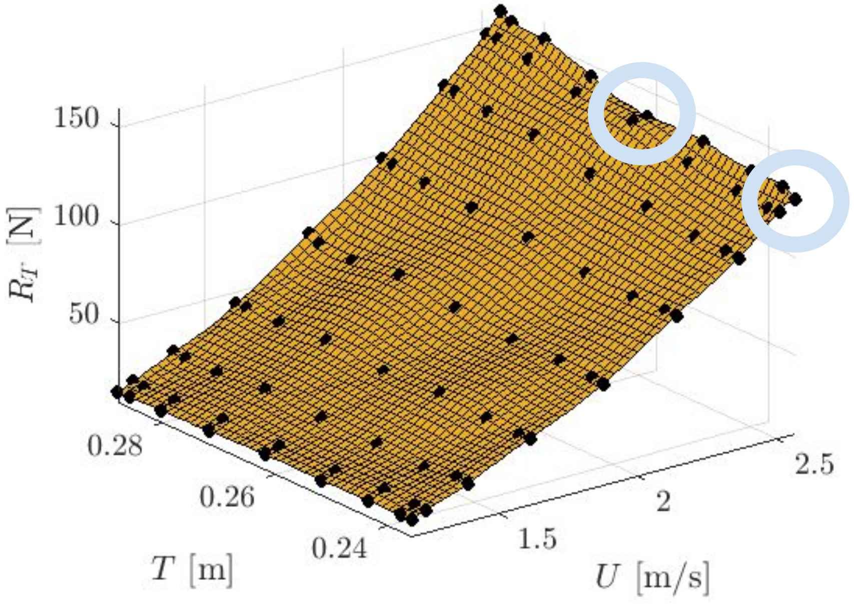
Surrogate models for R_T



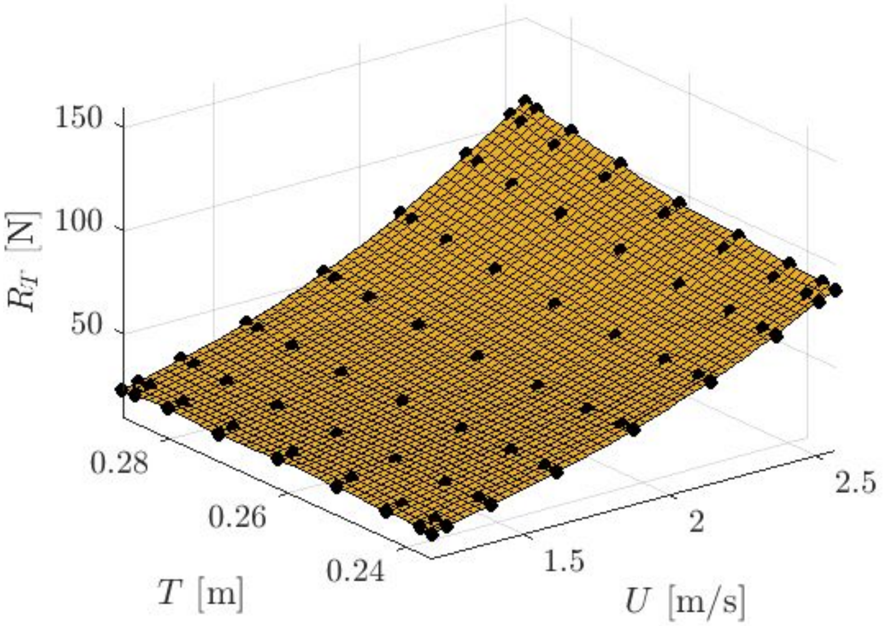
We have discussed this yesterday

RANS solver noise gets interpolated!

Numerical noise

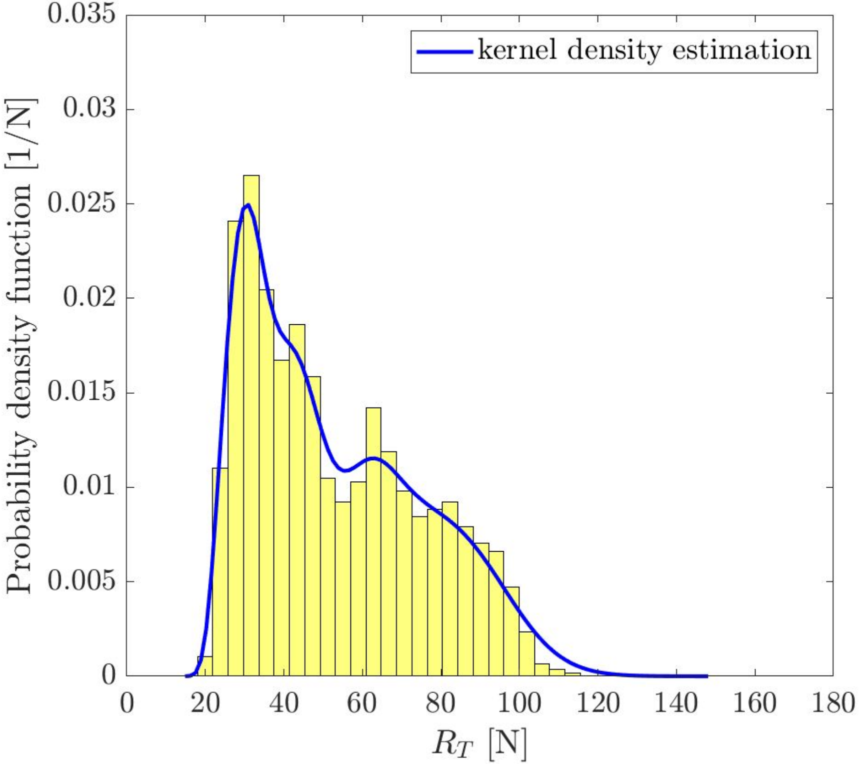


Lowest-fidelity surrogate



Highest-fidelity surrogate

Pdf of R_T



kernel density estimation
Probability density function [1/N]
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
20
40
60
80
100
120
140
160
180
R_T [N]

Bonus slide: convergence result

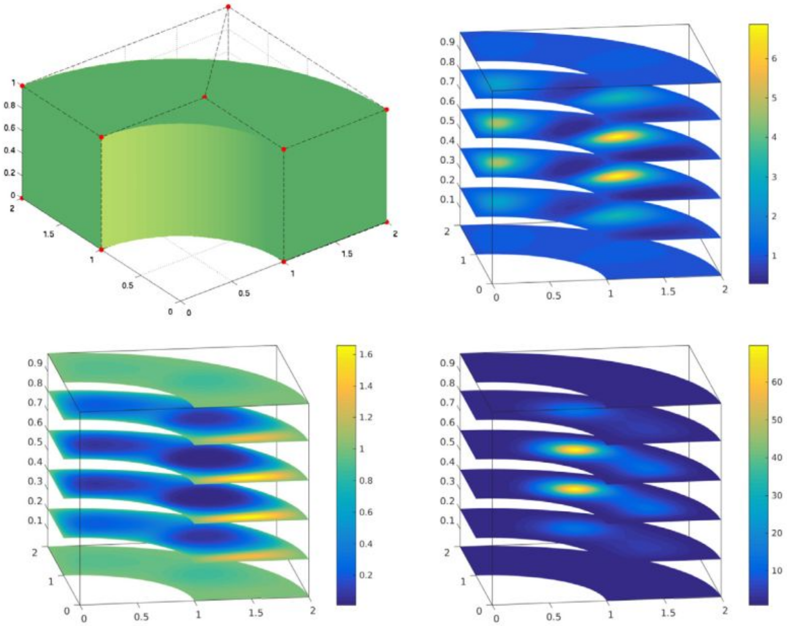
Elliptic PDE with parametric diffusion coefficient with mild regularity

- the solution is **y**-analytic , so it makes sense to use sparsification
- Convergence rate:
  - If **N < ∞**: Err ≤ C $\text{work}^{-s}\log(\text{work})^{r}$ with s,r indep of N, but C might
  - If **N → ∞**: Err ≤ C work^{-r} with s,r,C indep of N

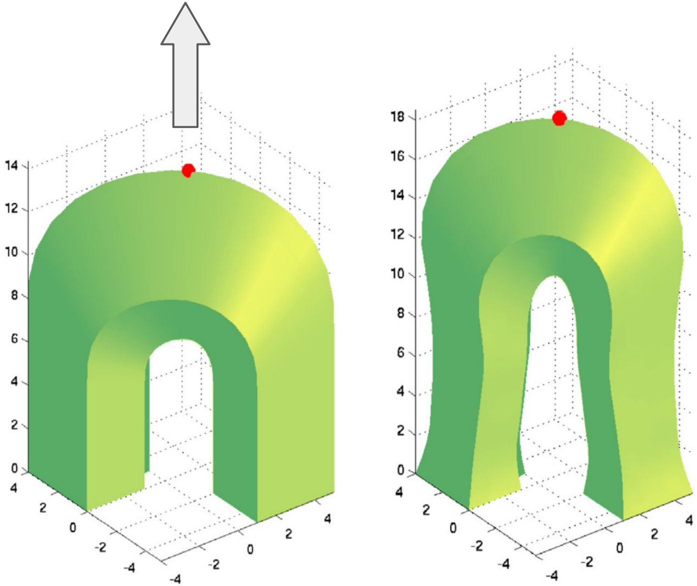
Setting:

- u(**x**,**y**) lives in a Bochner space with mixed Sobolev regularity
- derive decay of the coefficients of the multi-variate spectral expansions wrt **y**
- Choose the best ones by knapsack problem
- Check summability of the truncation error
- connect with multi-variate Lagrange interpolation results

Bonus slide: more tests



Elliptic PDE, random diffusion coeff



Linear elasticity, uncertain *E*, *ν*

Conclusions

Summary of this talk

- Uncertainty Quantification combines statistics, numerical analysis, approximation theory
- The function to sample is:
    - Implicit & expensive
    - high-dimensional

We need:

- Efficient sampling schemes + surrogate models to generate quickly many values
- Multi-fidelity approach

- **MISC** uses sparsification principle. Fast, effective, but issue with noise

Future work

- Software release
- Noise issues
- More efficient adaptive algorithm (ML hybridization)
- Application to inverse UQ, robust optimization, etc...

Bibliography & thanks for the attention!

J. D. Jakeman, S. Friedman, M. Eldred, L. Tamellini, A. Gorodetsky, D. Allaire (2022)

Adaptive experimental design for multi-fidelity surrogate modeling of multi-disciplinary systems.

International Journal for Numerical Methods in Engineering

C. Piazzola, L. Tamellini, R. Pellegrini, R. Broglia, A. Serani, M. Diez (2022)

Comparing Multi-Index Stochastic Collocation and Multi-Fidelity Stochastic Radial Basis Functions for Forward Uncertainty Quantification of Ship Resistance

Engineering with Computers

J. Beck, L. Tamellini, R. Tempone (2019)

IGA-based Multi-Index Stochastic Collocation for random PDEs on arbitrary domains.

Computer Methods in Applied Mechanics and Engineering

A. Haji-Ali, F. Nobile, L. Tamellini, R. Tempone (2016)

Multi-index Stochastic Collocation convergence rates for random PDEs with parametric regularity.

Foundations of Computational Mathematics

A. Haji-Ali, F. Nobile, L. Tamellini, R. Tempone (2016)

Multi-index stochastic collocation for random PDEs

Computer Methods in Applied Mechanics and Engineering

A noisy RANS solver