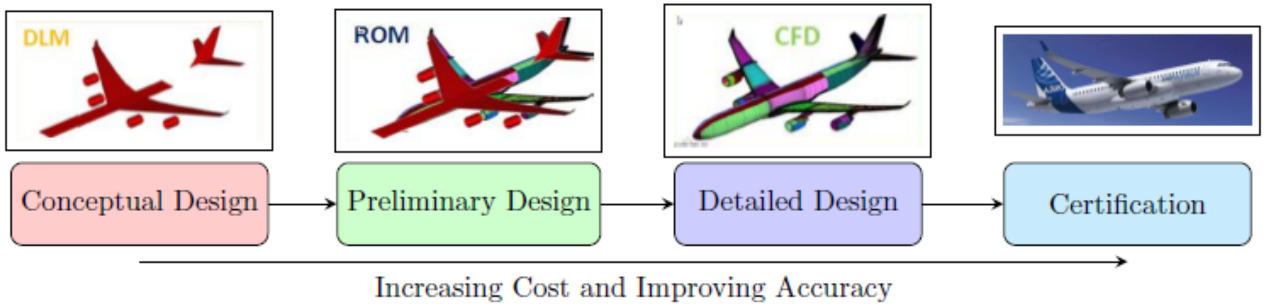
3AF: Big Data & Structures

Sparse and Distributed Gaussian process for Flight test and Structural dynamics

Prof. J. Morlier (SUPAERO), M. Colombo (AIRBUS)

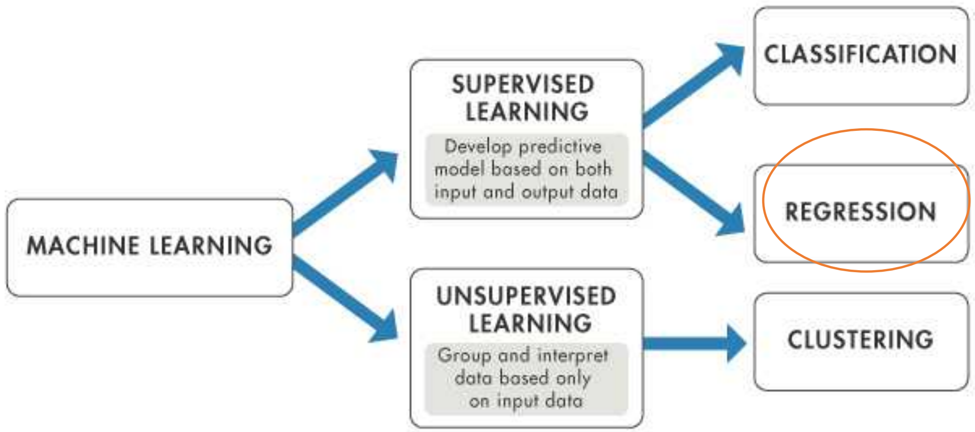
Based on the results of Ankit's Chiplunkar PhD

Our Goals

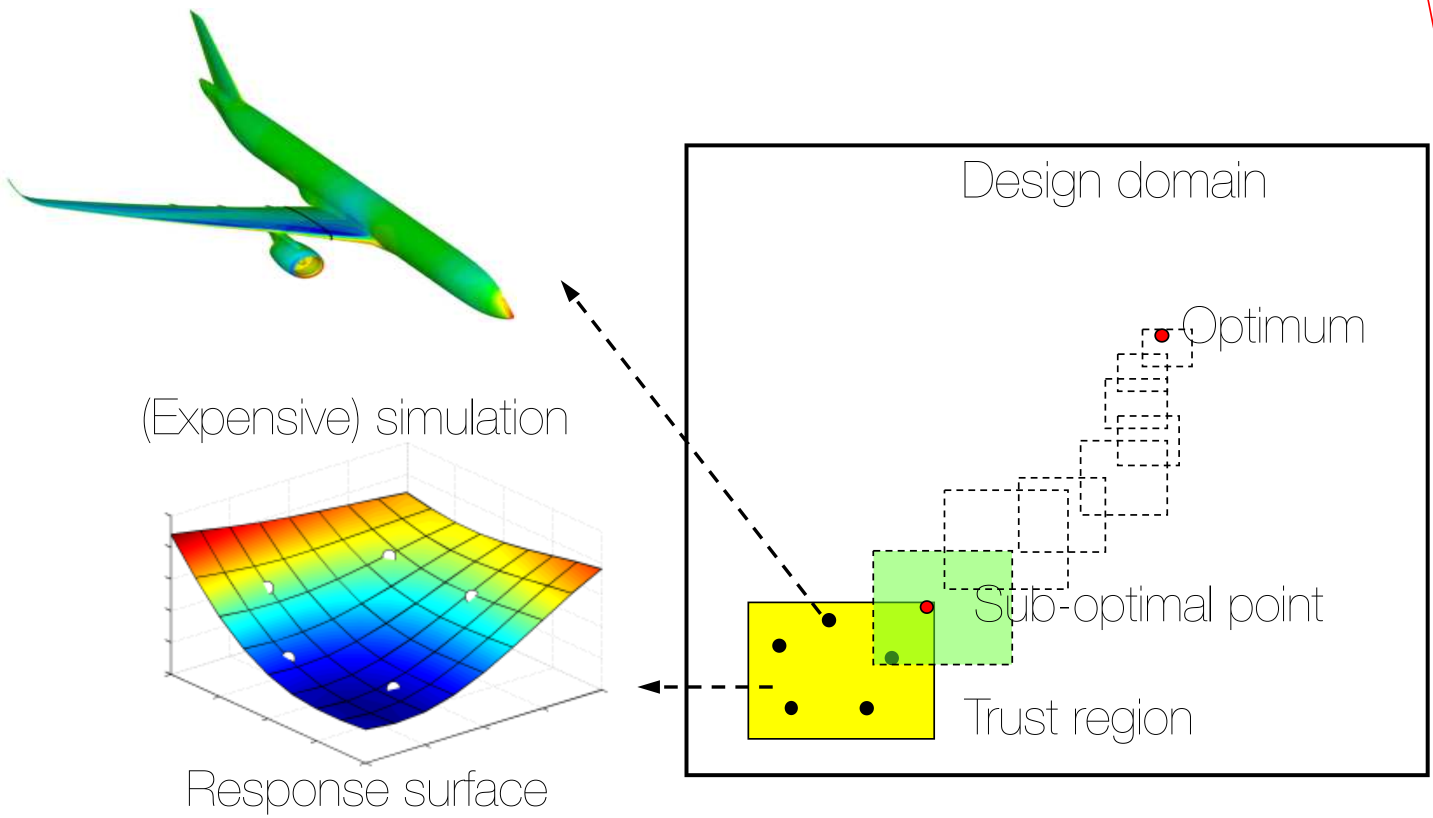


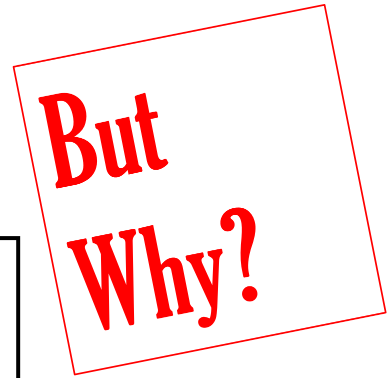
- Designing an aircraft takes almost a decade to complete.
- Design process has been distributed into several design loops
- Each loop requiring more costly simulations

- Hence there is a desire to reduce in a « smart way » the computation time
- It can be done using surrogate modeling or machine learning technics



SURROGATE MODELING (learning for Optimizing)

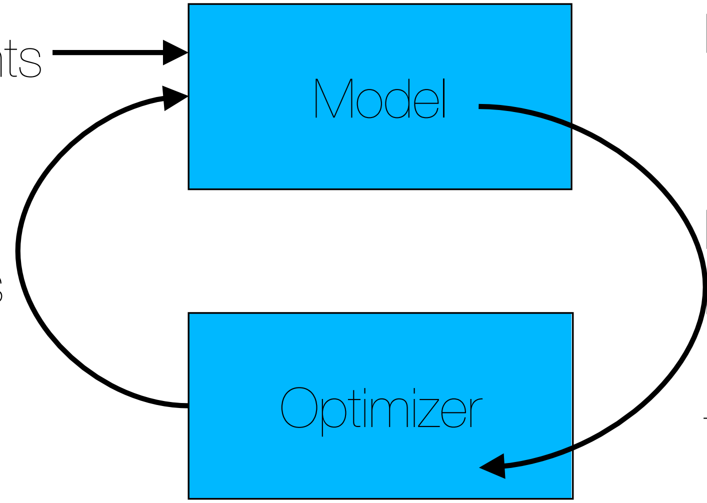




Gradient Based Optimization is costly

Constants

Design variables $\mathbf{x}$



Responses $\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}$

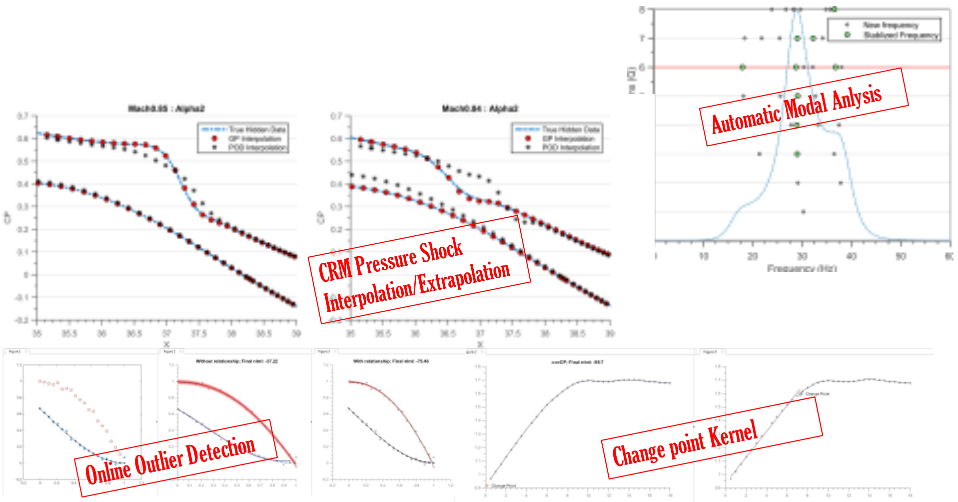
Derivatives of responses (design sensi-tivities*)

$$\frac{\partial f}{\partial x_i}, \frac{\partial g}{\partial x_i}, \frac{\partial h}{\partial x_i}$$

*SOL200 in MSC Nastran for example

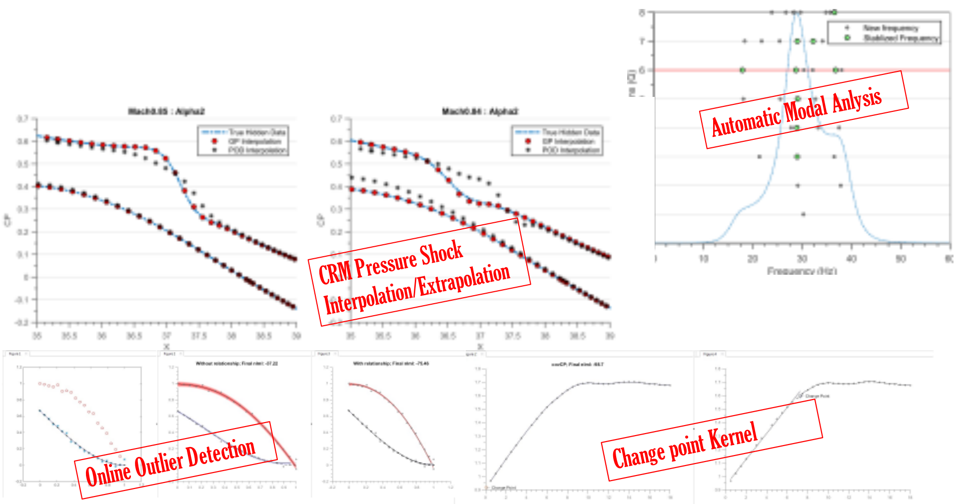
Outlines

1. Challenge
2. Some applications of ML
3. Link to HPC and FT @ Airbus



Outlines

1. Challenge
2. Some applications of ML
3. Link to HPC and FT @ Airbus



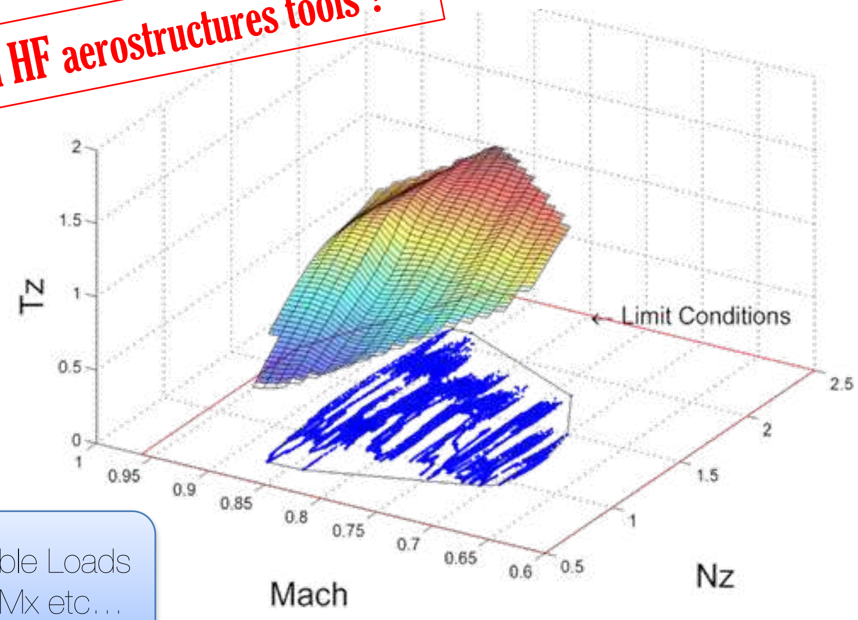
Machine learning for load estimation (Ankit Chiplunkar, AIRBUS FUND)

Kriging (Pionneer)	Gaussian Processes (link with AI)
Developed by Daniel Krige – 1951; formalized by Georges Mathéron in the 60's (Mines Paris)	Neural network with infinite neurons tend to Gaussian Process 1994
Evaluation: minimize error variance	Evaluation: Marginal Likelihood

Loads identification

First flight ready → Flight test of Maneuvers → Capture aero/ structural data → Extract loads → Certification of limit loads

Can we extrapolate limit loads using both measurements and HF aerostructures tools ?



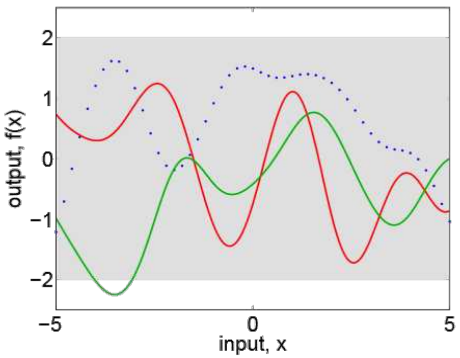
Can we automate the process?

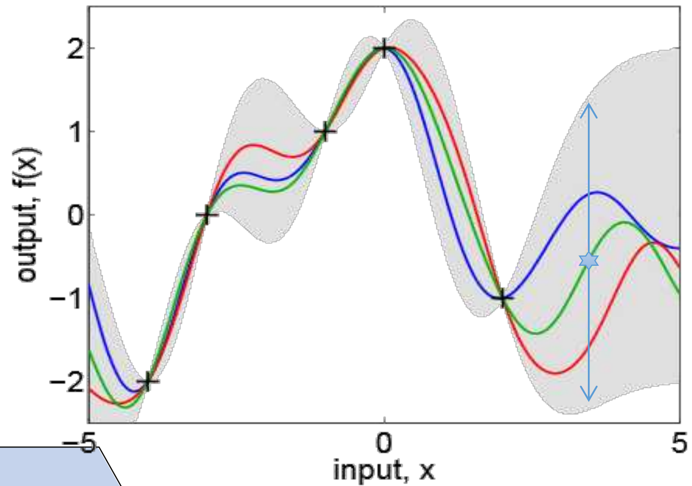
Statistical step

A/C states α, β, M, δ's → Distributed Gaussian Process → Flexible Loads Tz, Mx etc…

Gaussian Process Regression

Image Source: http://mlg.eng.cam.ac.uk/teaching/4f13/1314/





Prior

f|x

$GP(0, K)$

X

Y

Training

y|x

$GP(0, K + \sigma_n^2 I\ )$

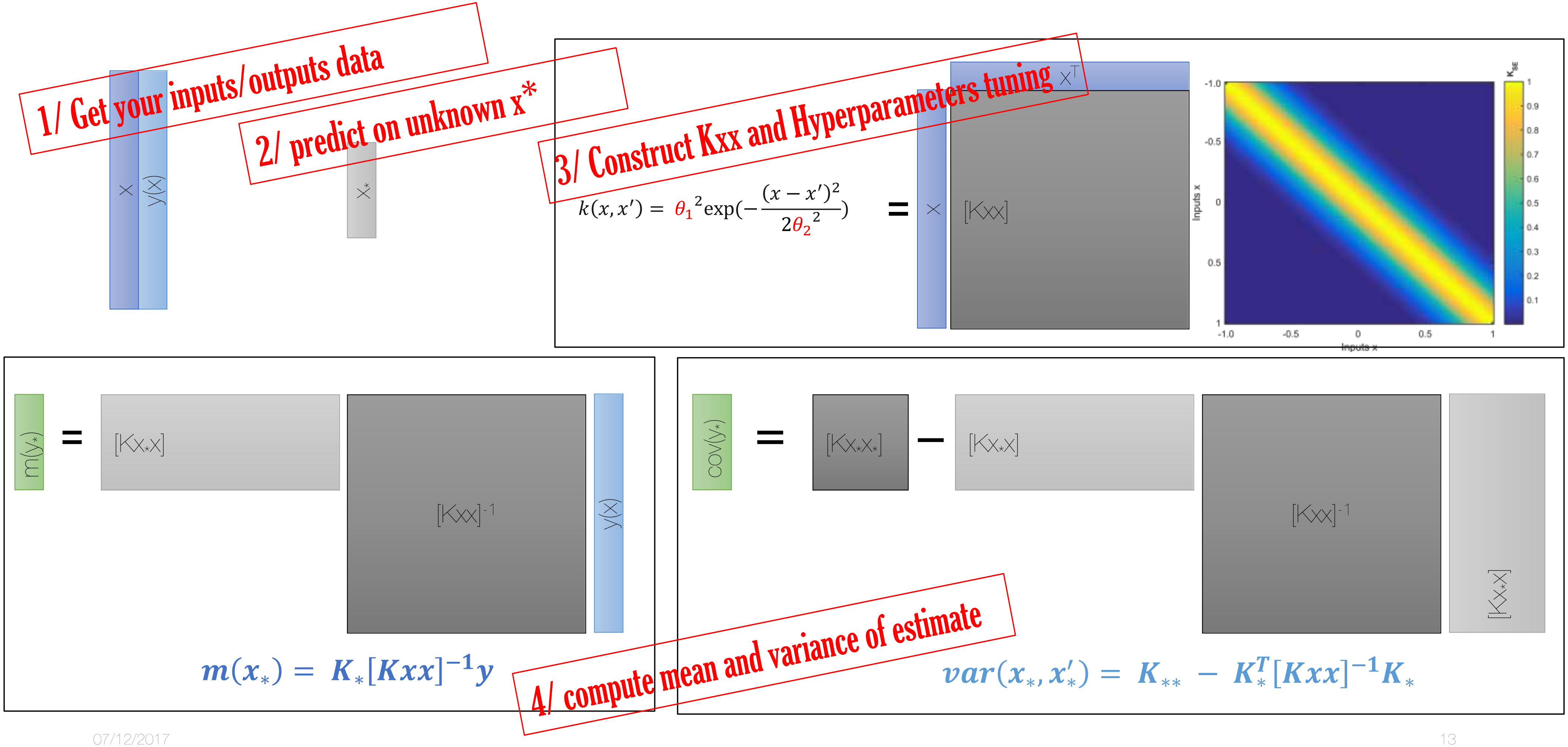
Optimize: for a better hyperparameters and better prior

Posterior on y

$\boldsymbol{y}^* |\ \boldsymbol{x}^*, \boldsymbol{y}, \boldsymbol{x}$

GP ($K_*[K + {\sigma_n}^2 I]^{-1}$y, $K_{**} + \sigma_n^2 I\ -\ K_*^T[K + {\sigma_n}^2 I]^{-1}K_*)$

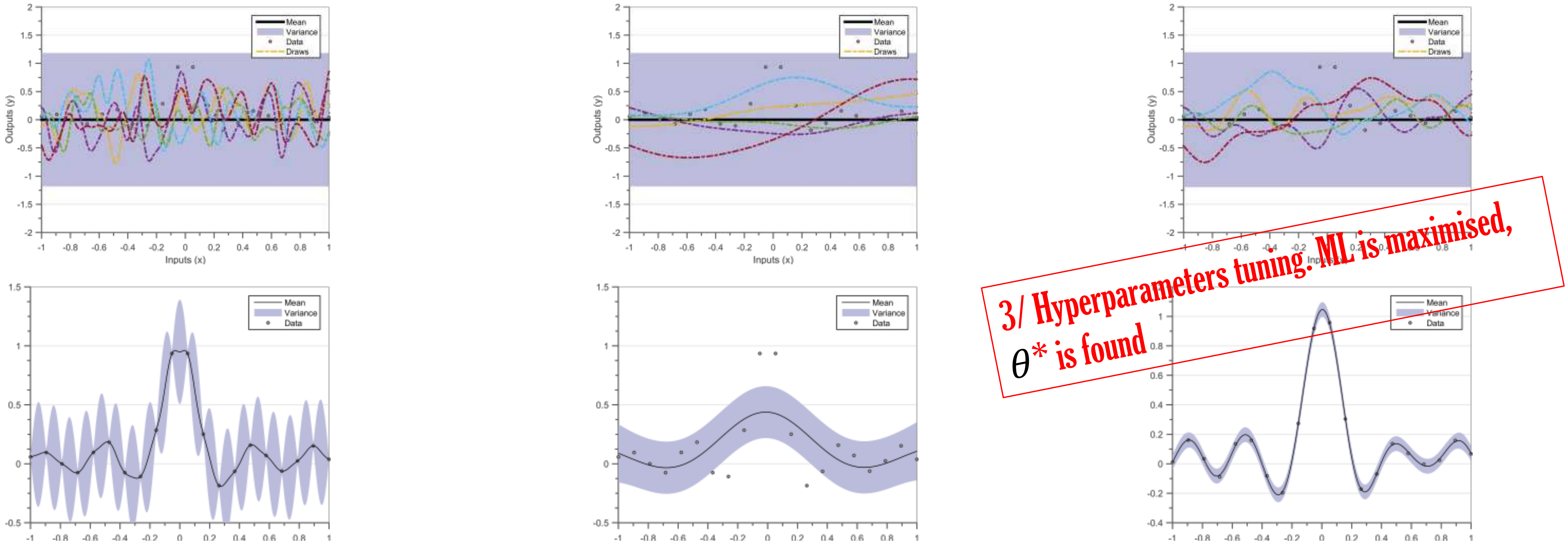
Matrix view of Gaussian Process



Optimizing marginal likelihood (ML)

$$\text{ML} = log\big(p(y|X,\theta)\big) = -\frac{1}{2}y^T K^{-1} y \; - \; \frac{1}{2}log|K| \; - \frac{n}{2}log(2\pi)$$

- It is a combination of **data-fit term**, a **complexity penalty** term and a **normalization term**

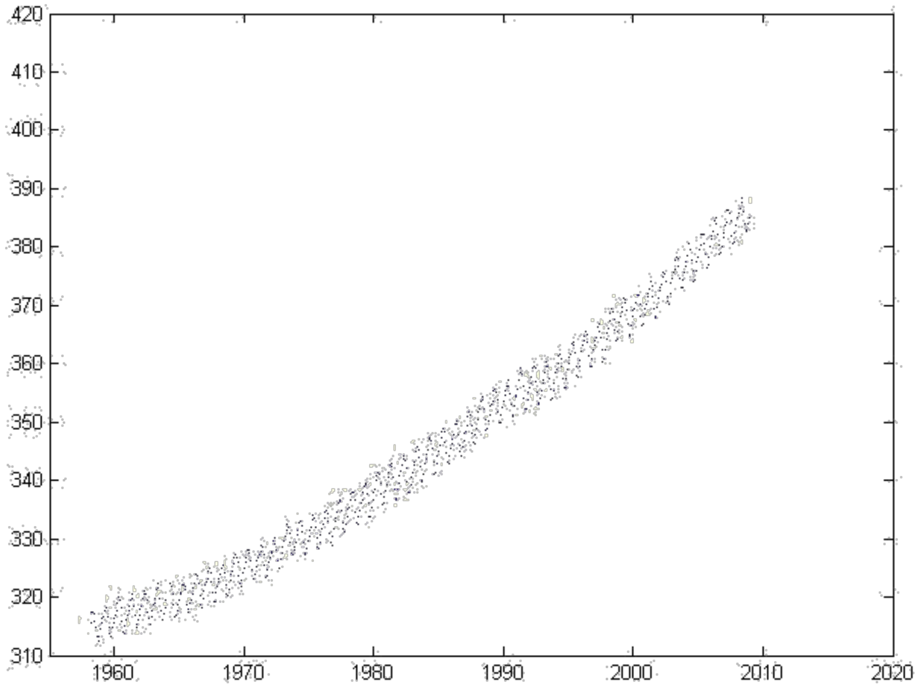


ML = -8.2

ML = -35.3

ML = 6,04

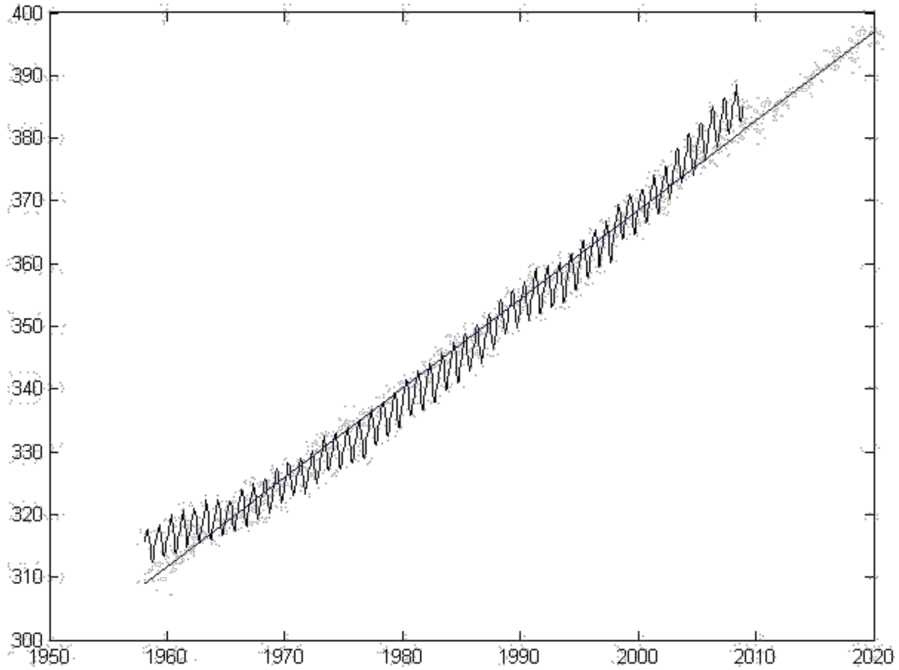
A SIMPle Example



Month-wise data of CO_2 concentration in atmosphere at Hawaii

Image Source: http://mlg.eng.cam.ac.uk/teaching/4f13/1314/

Example – Linear Regression



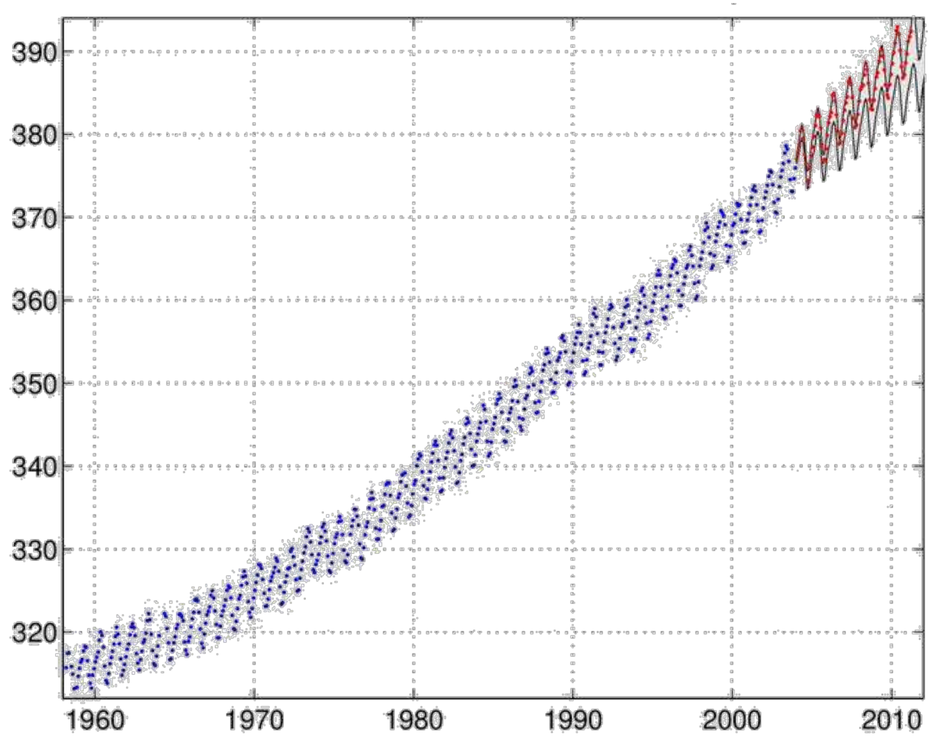
Should we choose a polynomial?

What degree of polynomial should we choose? (overfitting)

For a given degree, what parameters of polynomial should we choose

Image Source: http://mlg.eng.cam.ac.uk/teaching/4f13/1314/

Example – Gaussian Process



Predicted variance after year 2005 in grey, real data-points in red

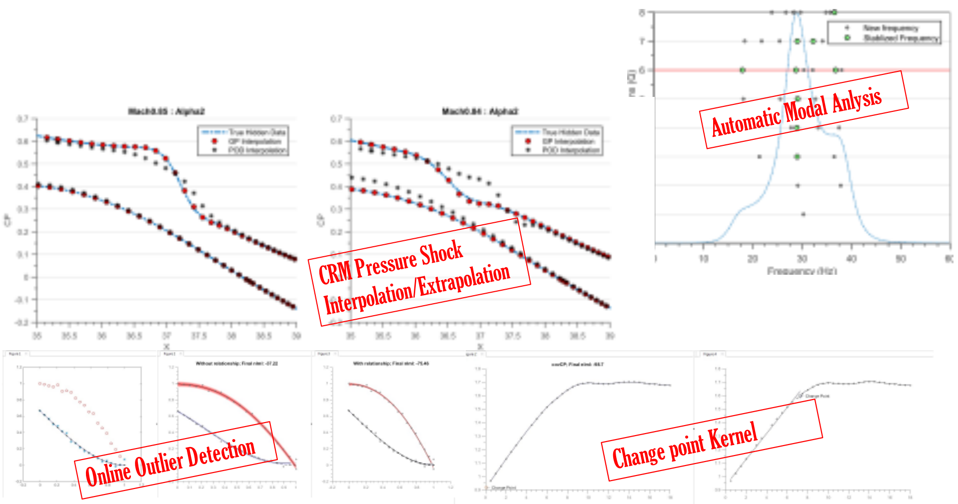
Image Source: http://mlg.eng.cam.ac.uk/teaching/4f13/1314/

Outlines

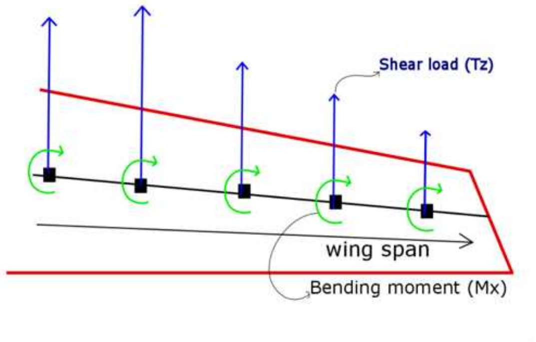
1. Challenge

2. Some applications of ML

3. Link to HPC and FT @ Airbus



Multi-Output Gaussian Process – Flight Test examples



Given: $f_1 = g(f_2, x)$

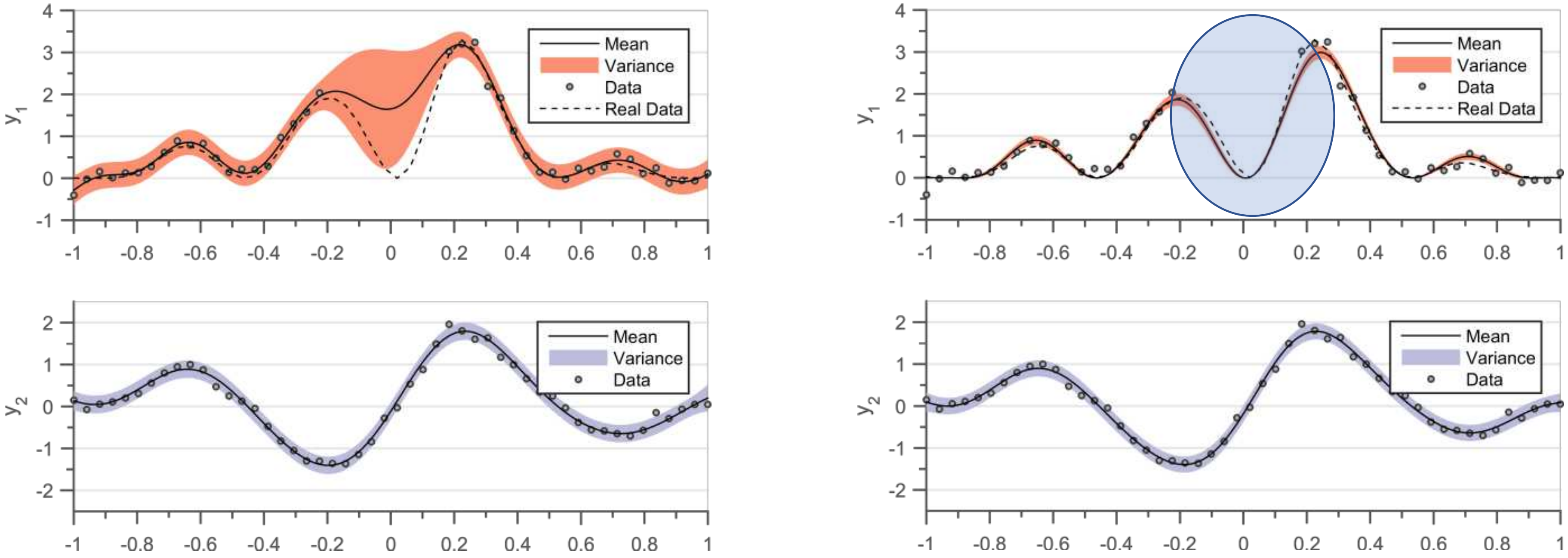
- Earlier examples include **Gradient Enhanced Kriging** (GEK) or **Co-kriging**

- But we want to expand this to integral enhanced kriging, double differential, or any functional relationship between outputs

Forrester, A. I. J., Sobester, A. and Keane, A. J. (2007) Multi-fidelity optimization via surrogate modelling. Proceedings of the Royal Society A, 463(2088), 3251–3269, (doi:10.1098/rspa.2007.1900).

Liu, Weiyu. Development of gradient-enhanced kriging approximations for multidisciplinary design optimization. 2003.

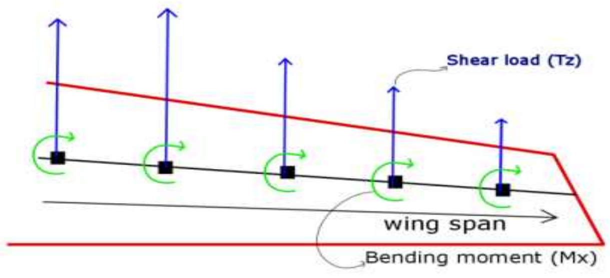
Example 1: Faulty sensors (using synthetic data) $y_1 = (y_2)^2$



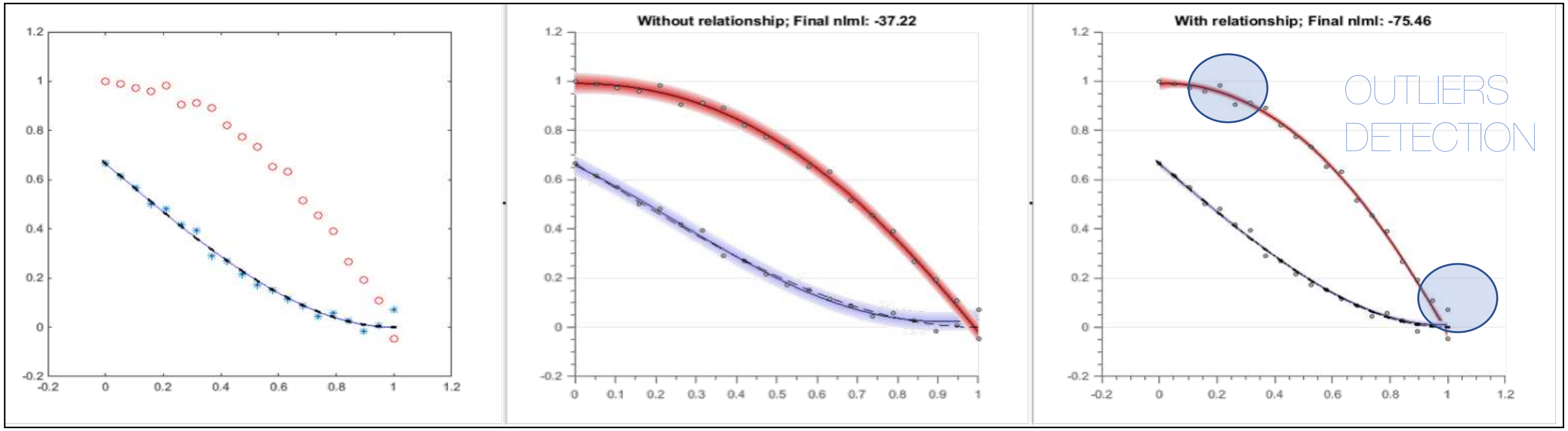
Independent GPs

Related GPs

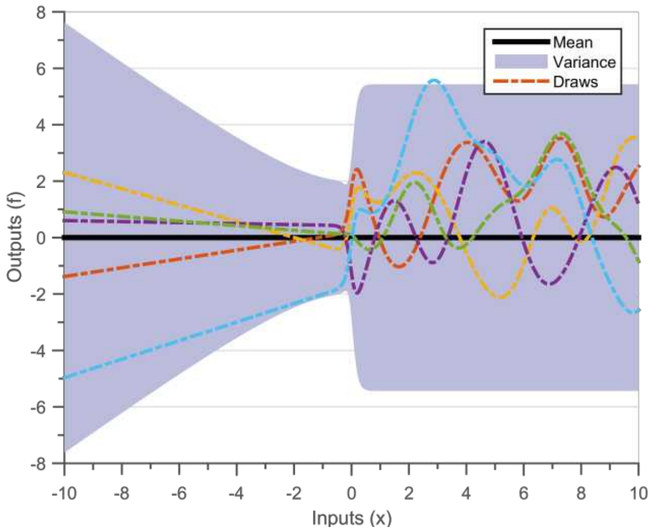
Example 2: use the Relationship between Tz and Mx permits to reduce uncertainties

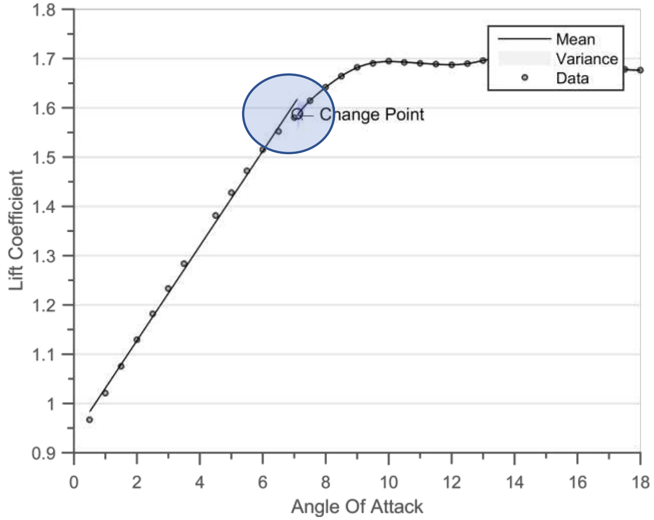


$$Mx = \int_{\eta}^{\eta_{edge}} (x - \eta)\, Tz\, dx$$



Example 3: Identifying onset of non-linearity

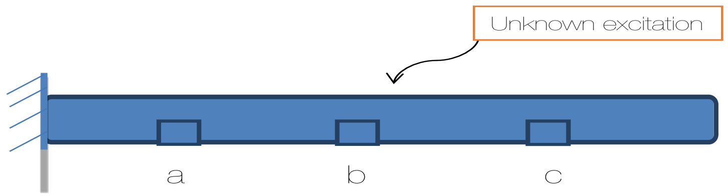


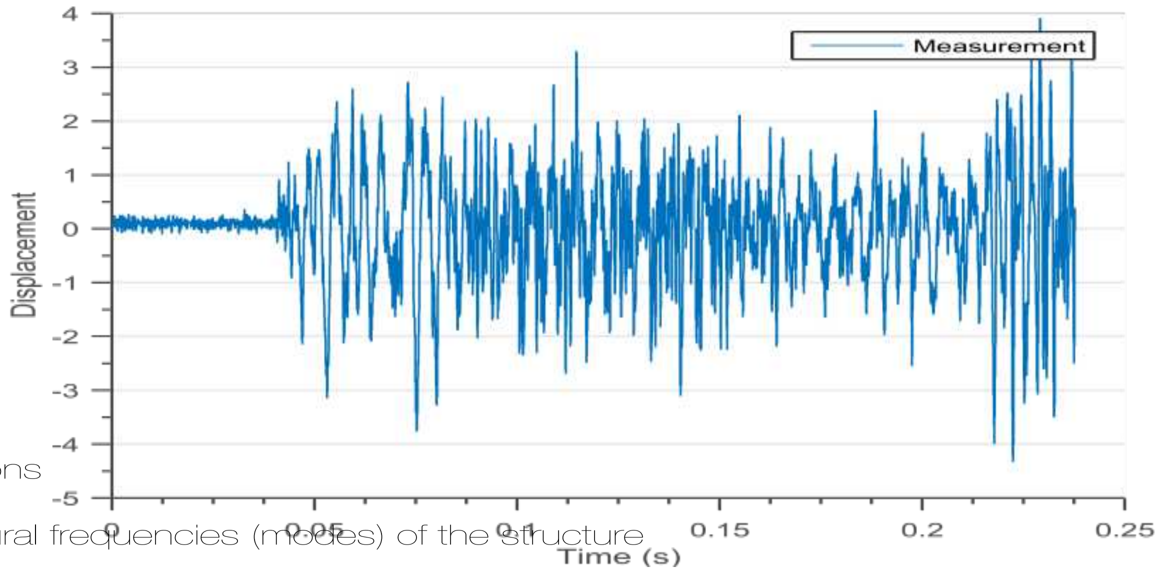


$$k_{CP}(k_1, k_2, x_1, x_2) = sigm(x_1)k_1 sigm(x_2) + (1 - sigm(x_1))k_2(1 - sigm(x_2))$$

- Estimate change in pattern
- Use global optimization to identify the non-lineariity automatically

Example 4: Operational Modal analysis

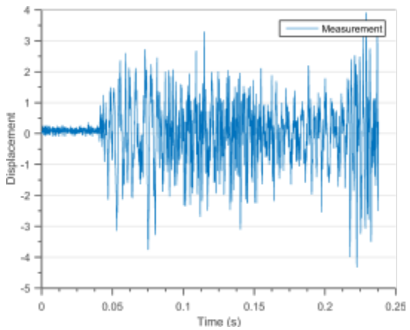


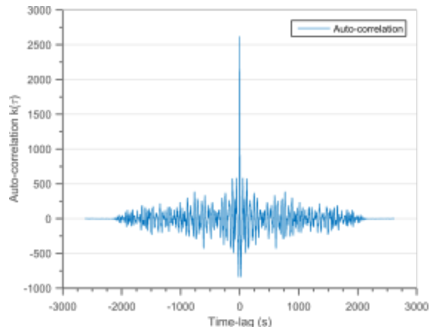


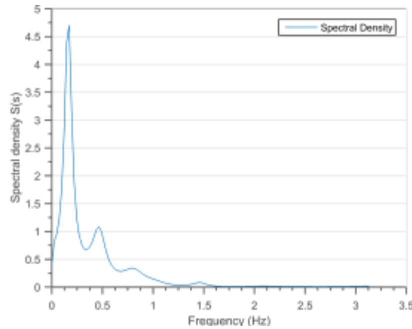
1. Sample is subjected to unknown random excitations
2. Each sensor record time signals
3. Peaks in their power spectral density give the natural frequencies (modes) of the structure

DiazDelaO, F. A., and S. Adhikari. "Structural dynamic analysis using Gaussian process emulators." Engineering Computations 27.5 (2010)

New paradigm: Spectral Mixture

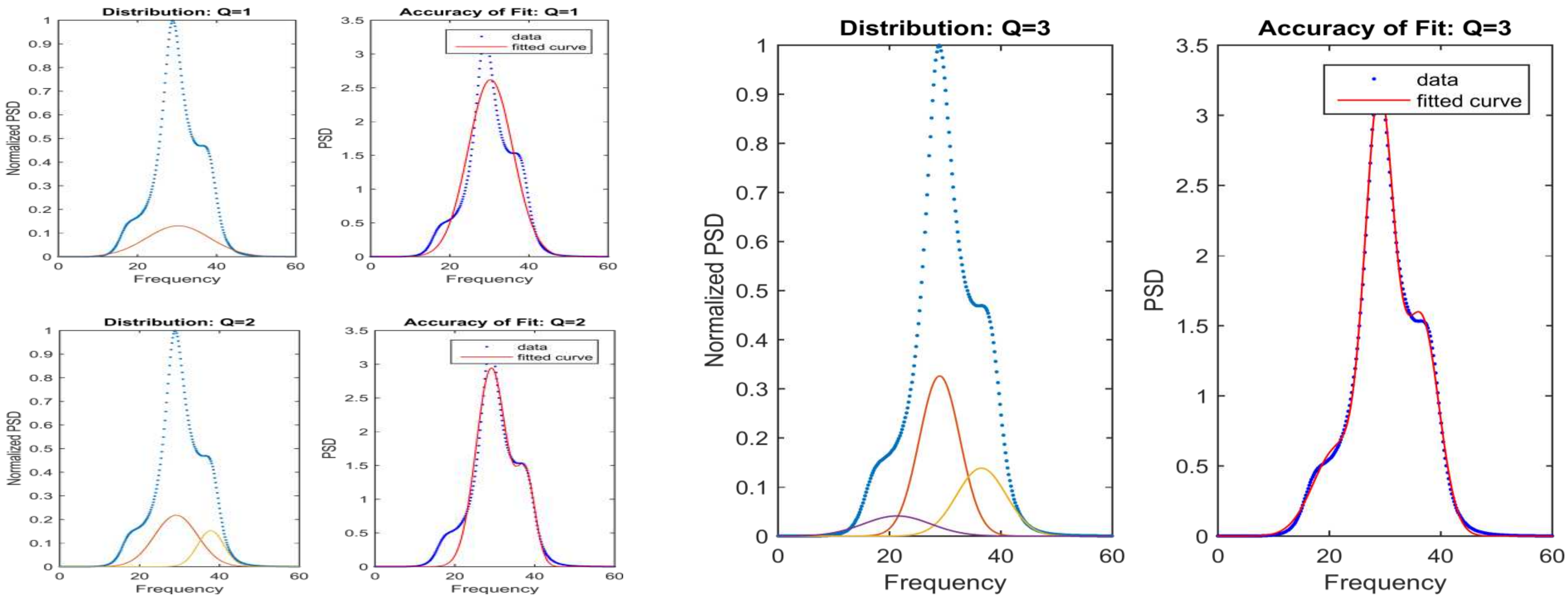






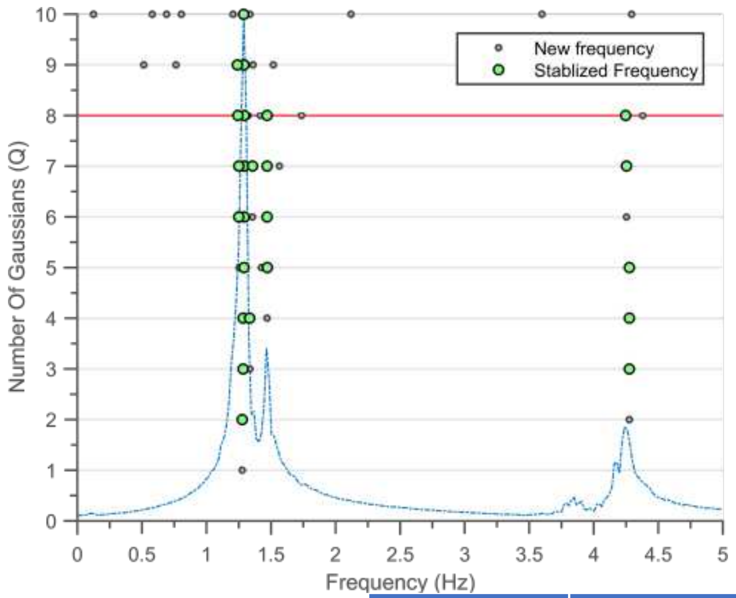
Displacement	Autocorrelation	Power spectral density
$x(t)$	$k(\tau) = \int x(t)x(t-\tau)dt$	$S(s) = Fourier(k(\tau))$
$M\{\ddot{x}(t)\}+C\{\dot{x}(t)\}+K\{x(t)\} = \{f(t)\}$		
	$k(\tau) = \sum A_i exp(-\lambda_i \tau) sin(B_i \tau)$	$S(s) = \frac{\sum a_k(s)^k}{\sum b_l(s)^l}$
Spectral Mixture Covariance		
$x(t) = GP(0, cov_{SM}(t,t'))$	$k_{SM}(d,\mu,\sigma,w) = \sum_{q=1}^{Q} w_q cos(2\pi\mu_q) exp[-2\pi^2 d^2 \sigma_q^2]$	$S_{SM}(s,\mu,\sigma,w) = \sum_{q=1}^{Q} \frac{w_q}{\sqrt{2\pi\sigma_q^2}} \left(\exp\left[-\frac{(s-\mu_q)^2}{2\sigma_q^2}\right] + \exp\left[-\frac{(-s-\mu_q)^2}{2\sigma_q^2}\right] \right)$

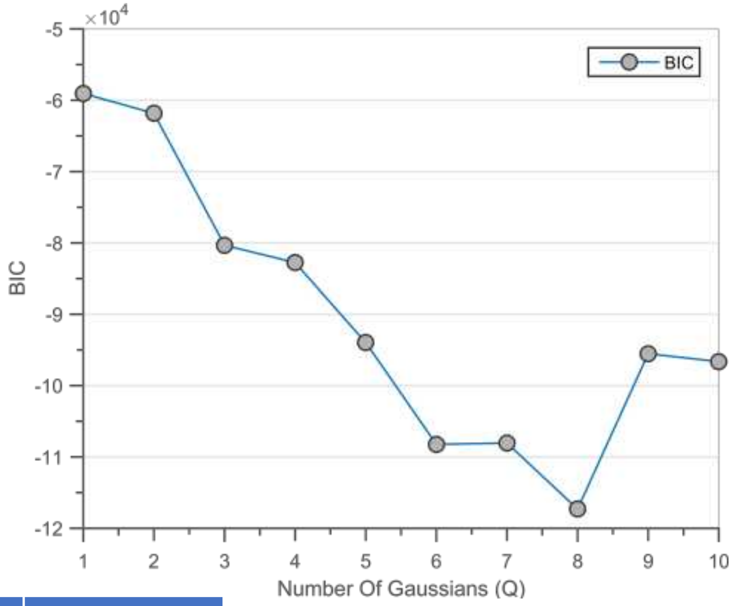
Gaussian Mixture Models



Plataniotis, K. N. and Hatzinakos, D. , Advanced Signal Processing Handbook Theory and Implementation for Radar, Sonar, and Medical

Automatic OMA (Testcase HTC building *)





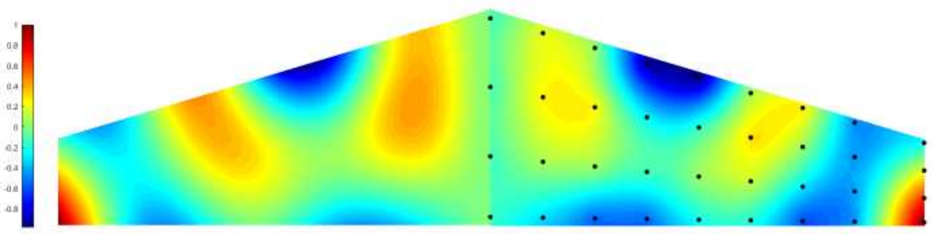
	From paper	GMM
Freq1	1,23	1,242
Freq2	1,27	1,292
Freq3	1,43	1,470
Freq4	3,86	3,866
Freq5	4,25	4,247

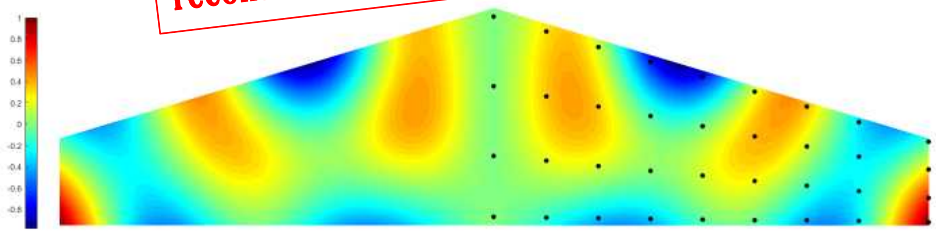
*Brincker, Rune, and Palle Andersen. "Ambient response analysis of the heritage court tower building structure." IMAC, 2000
Data from: http://www.brinckerdynamics.com/oma-toolbox/

Example 5: Sensor Placement Optimization

- Idea : Reduce in a smart way the number of sensors for GVT by using GP/Kriging for Modeshapes reconstruction
- How: By optimizing the sensor positions and use them as variables in a SPO problem

1/ Better Modeshape reconstruction using GP/Kriging


Comparison between HF FEA results (a) and a Linear Reconstruction with a regular grid of 36 sensors (b), for 9th Mode Shape.

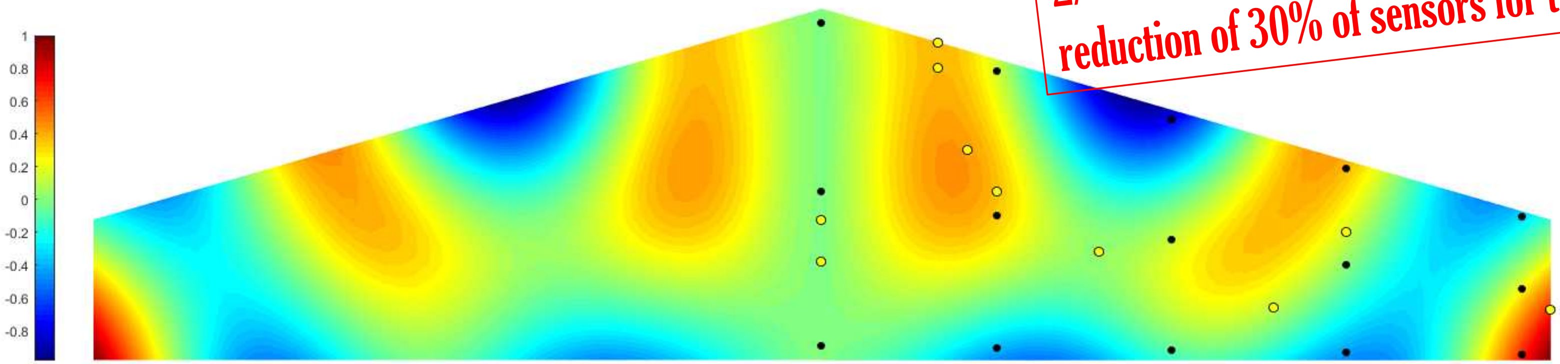

Comparison between HF FEA results (a) and a Kriging Reconstruction with a regular grid of 36 sensors (b), for 9th Mode Shape.

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4), 455-492.

EGO strategy

- Idea : Working at fixed budget
- How: Iterative procedure that maximize the trace of MAC (Modal Assurance Criteria)

2/ EGO-SPO allows a reduction of 30% of sensors for the entire modal basis.



Typical result showing SPO-EGO performance. As example 9th Mode Shape.
(a): HF FEA results. (b): SPO-EGO stategy. The black dots represent the initial DOE (regular grid, 15 sensors).

Papers&conf

Chiplunkar, E. Rachelson, M. Colombo and J. Morlier. Approximate Inference in Related Multi-output Gaussian Process Regression. Lecture Notes in Computer Science. 10163, 88-103. 2017

Chiplunkar and J. Morlier. Operational Modal Analysis in Frequency Domain using Gaussian Mixture Models . Proceedings of IMAC XXXV, 2017

Chiplunkar, E. Bosco and J. Morlier. Gaussian Process for Aerodynamic Pressures Prediction in Fast Fluid Structure Interaction Simulations. Proceedings of WCSMO12 2017

Chiplunkar, E. Rachelson, M. Colombo and J. Morlier. Adding Flight Mechanics to Flight Loads Surrogate Model using Multi-Output Gaussian Processes. AIAA AVIATION 2016

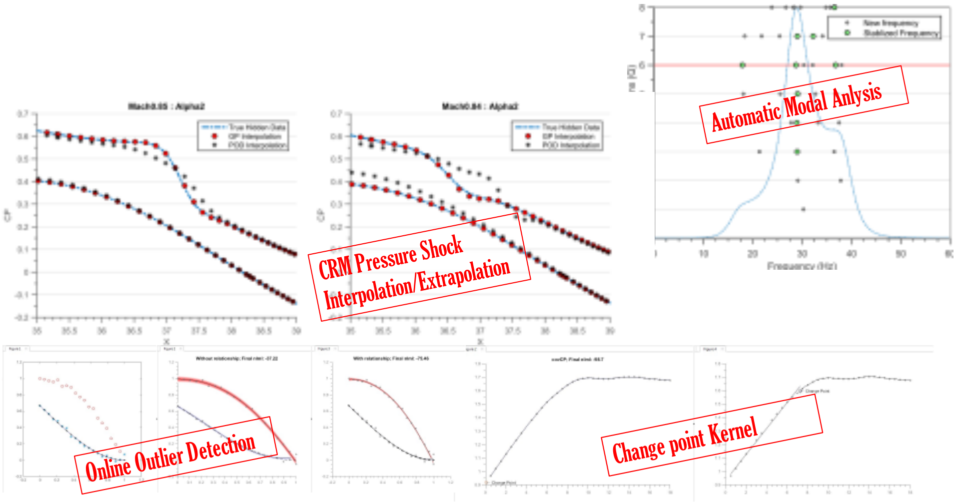
Chiplunkar, E. Rachelson, M. Colombo and J. Morlier. Sparse Physics-Based Gaussian Process for Multi-output Regression using Variational Inferenc. Proceedings of ICPRAM 2016 2016

Several Papers in preparation

Outlines

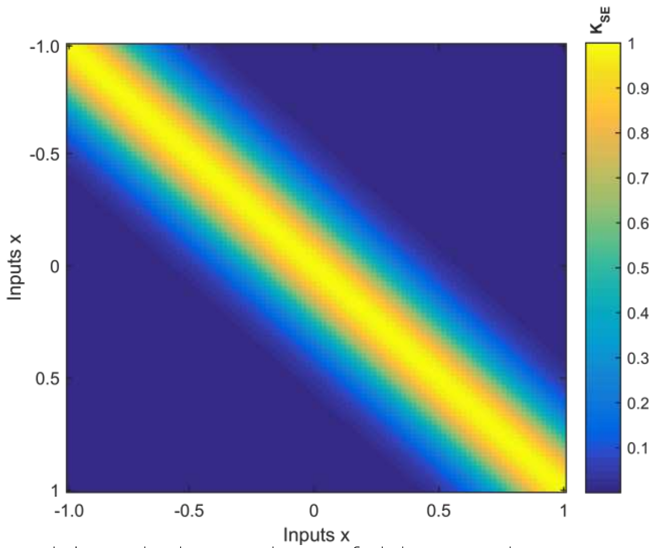
1. Challenge
2. Some applications of ML

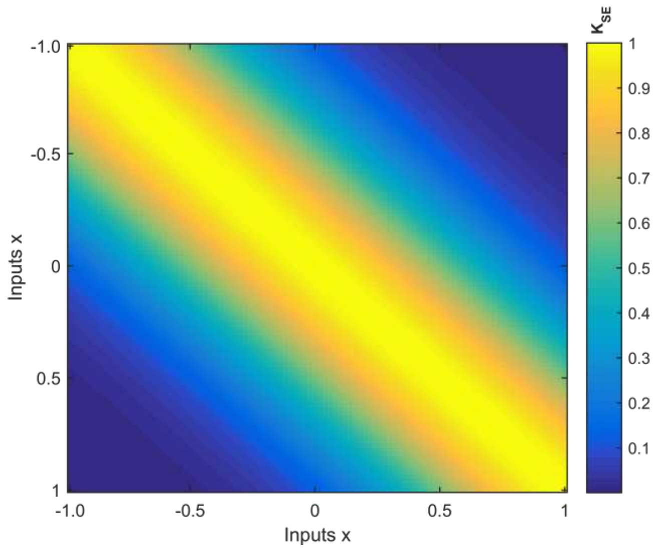
3. Link to HPC and FT @ Airbus



Scaling solutions to a GP: Gaussian Process Order

Computational and Memory requirements			
Learning	Storage	Mean	Variance
O(M2N) training cost and O(M2) pO(n^3)	O(n^2)	O(n)	O(n^2)

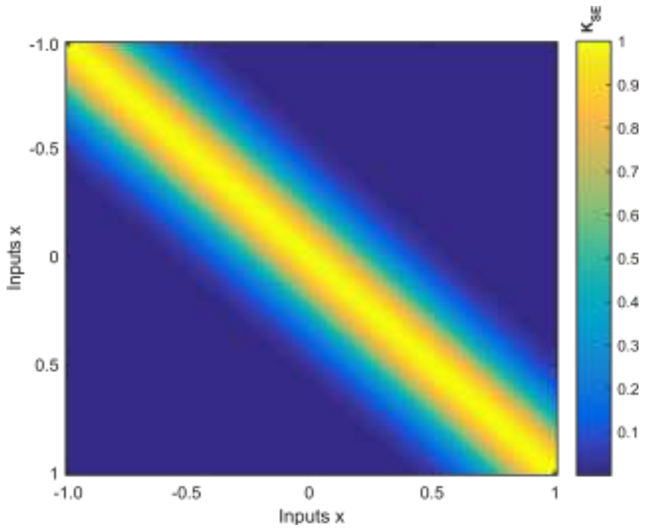


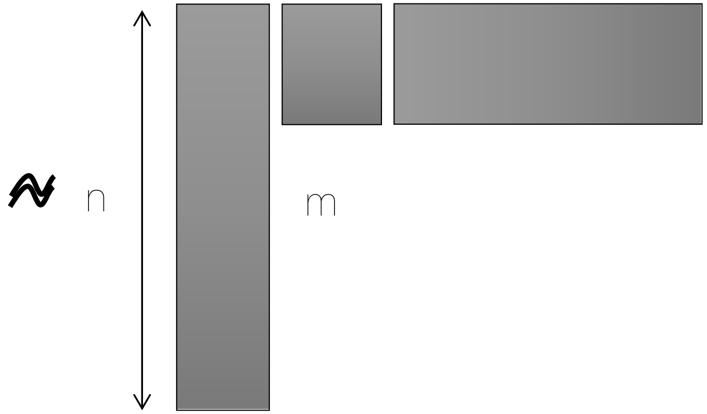


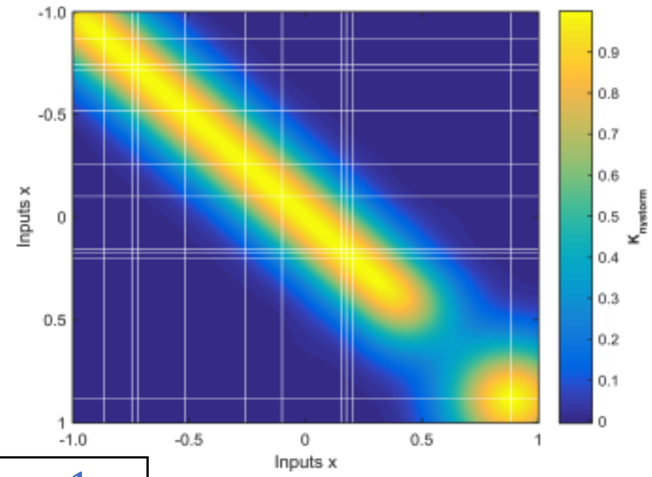
- Core problem is inversion of this matrix
- This limits the applicability of GPs when the number of training samples n grows large

  We run out of patience or worse, computer runs out of memory

Scaling solutions to a GP: Sparse Gaussian Process with inducing inputs







$$m(x_*) = K_{*approximate}\left[Kxx_{approximate}\right]^{-1} y$$

$$var(x_*, x'_*) = K_{**approximate} - K^{T}_{*approximate}\left[Kxx_{approximate}\right]^{-1} K_{*approximate}$$

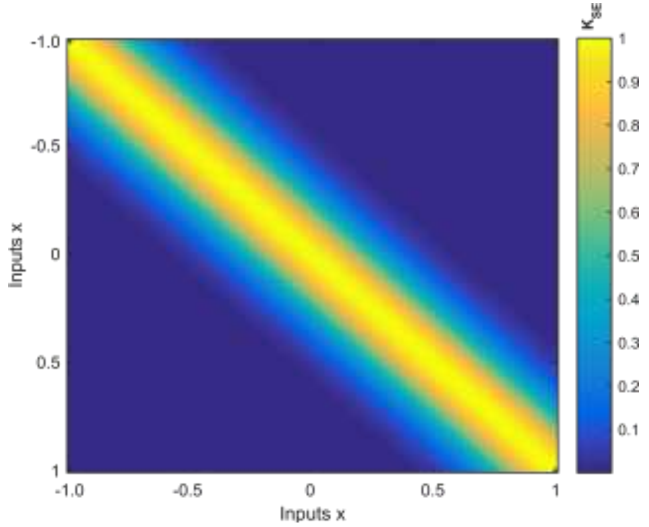
$$log(p(y|X,\theta)) = -\frac{1}{2} y^T K_{approximate}{}^{-1} - \frac{1}{2} log|K_{approximate}| - \frac{n}{2}\log(2\pi)$$

References:

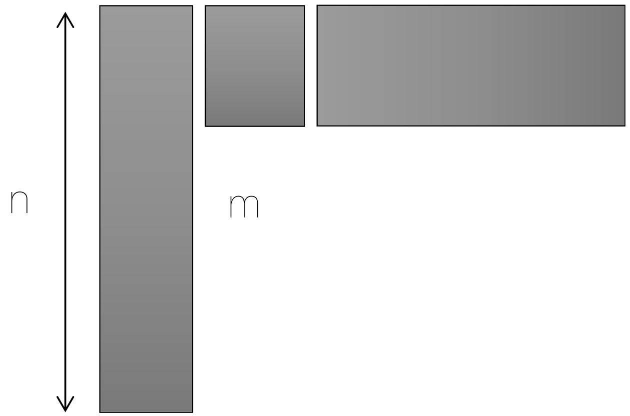
Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In In Artificial Intelligence and Statistics 2009

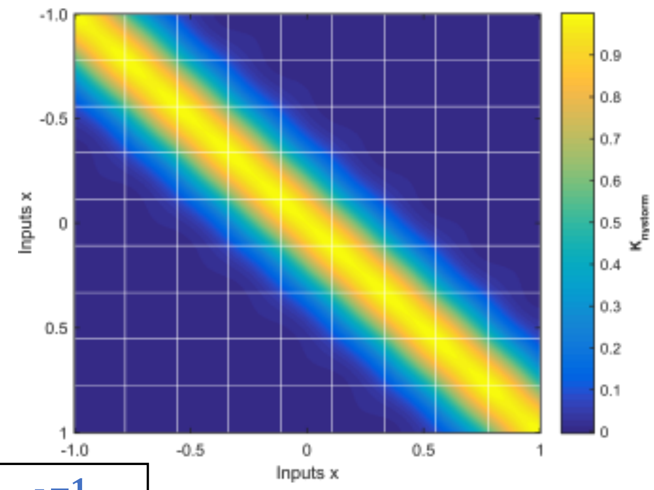
Christopher KI Williams et Matthias Seeger. Using the Nyström method to speed up kernel machines. In Advances in neural information processing systems, 2001

Scaling solutions to a GP: Sparse Gaussian Process with inducing inputs



$\approx$





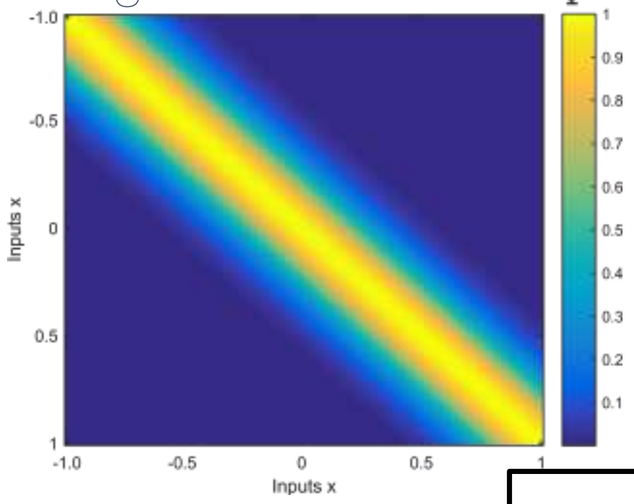
$$m(x_*) = K_{*approximate}\left[Kxx_{approximate}\right]^{-1} y$$

$$var(x_*, x'_*) = K_{**approximate} - K^T_{*approximate}\left[Kxx_{approximate}\right]^{-1} K_{*approximate}$$

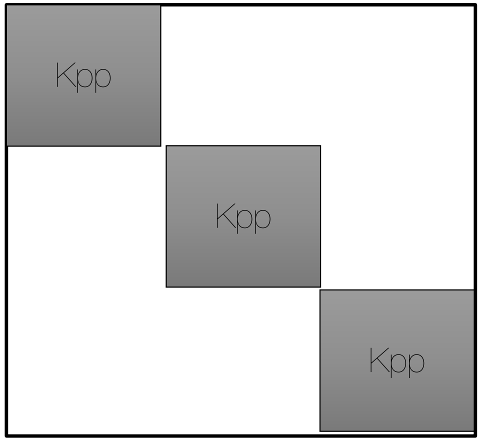
$$log(p(y|X,\theta)) = -\frac{1}{2} y^T K_{approximate}{}^{-1} - \frac{1}{2} log|K_{approximate}| - \frac{n}{2}\log(2\pi)$$

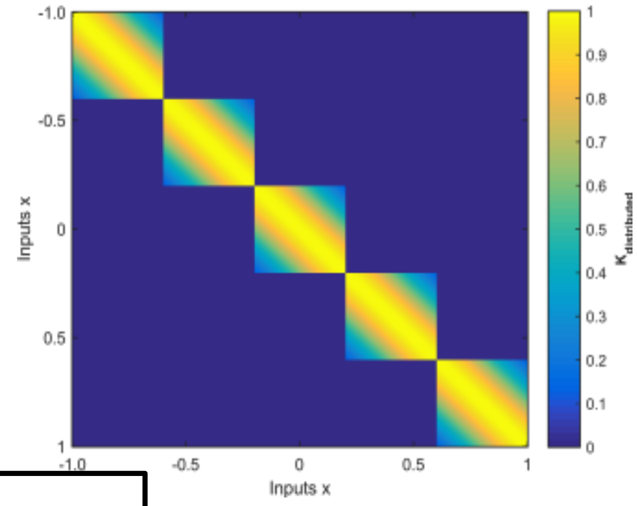
Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In In Artificial Intelligence and Statistics 2009
Christopher KI Williams et Matthias Seeger. Using the Nyström method to speed up kernel machines. In Advances in neural information processing systems, 2001

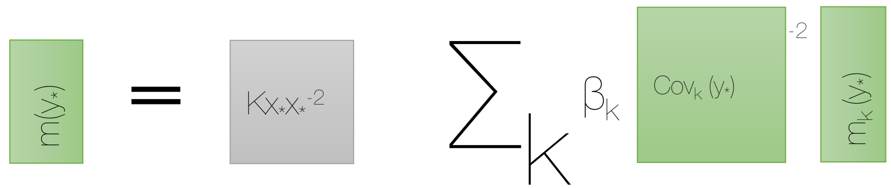
Scaling solutions to a GP: Distributed Gaussian Process



$\approx$



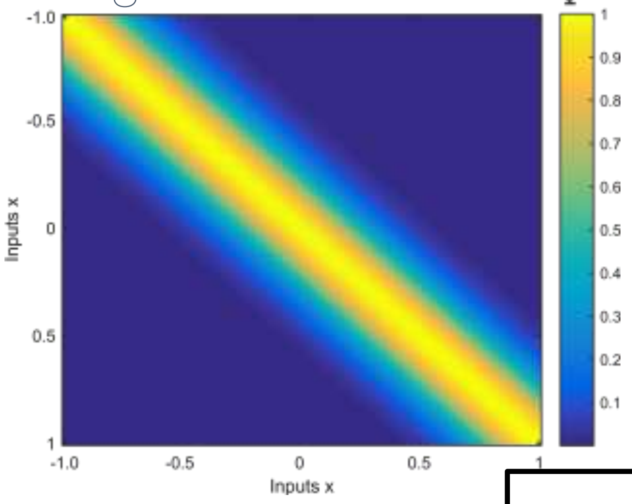




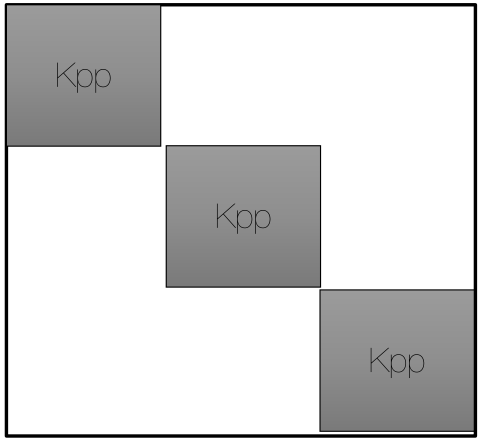
$$\beta_k = \log\left(K_{x_*x_*}{}^{-2}\right) - \log\left(Cov_k(y)^{-2}\right)$$

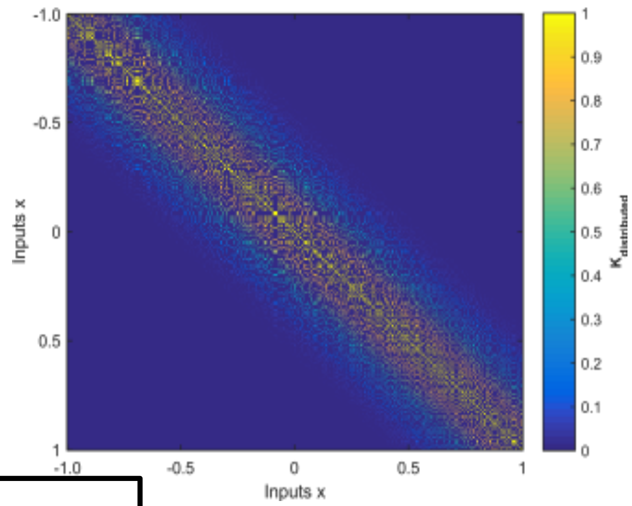
References:
Tao Chen et Jianghong Ren. Bagging for Gaussian process regression. Neurocomputing, 2009.
Marc Peter Deisenroth et Jun Wei Ng. Distributed Gaussian Processes. 2015

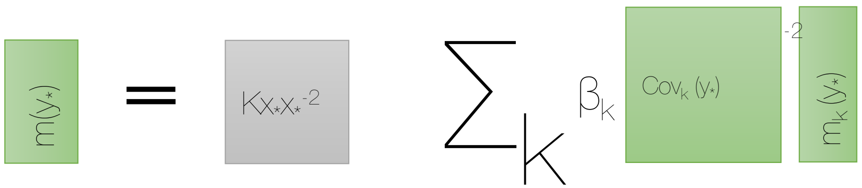
Scaling solutions to a GP: Distributed Gaussian Process



$\approx$





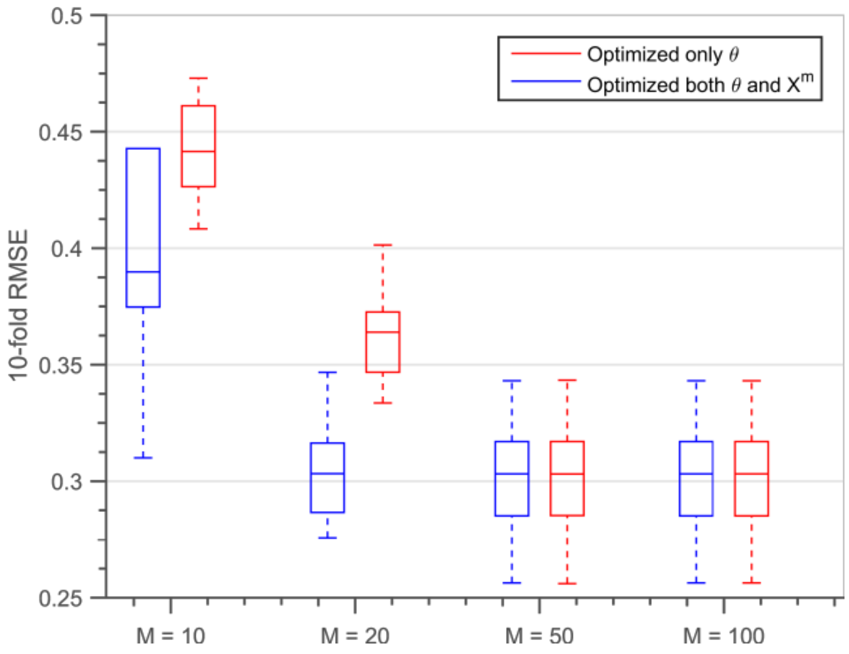


$$\beta_k = \log\left(K_{x_*x_*}{}^{-2}\right) - \log\left(Cov_k(y)^{-2}\right)$$

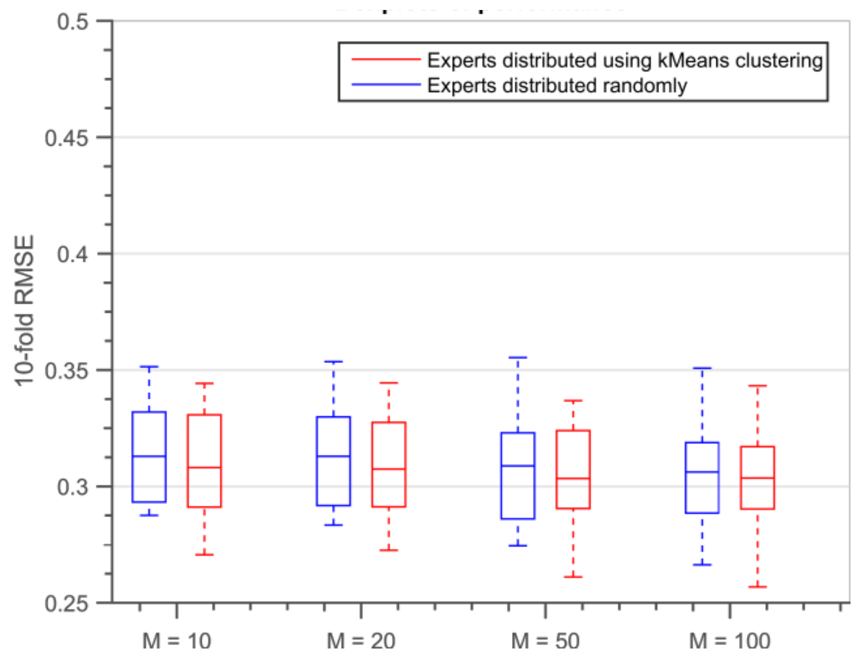
Tao Chen et Jianghong Ren. Bagging for Gaussian process regression. Neurocomputing, 2009.
Marc Peter Deisenroth et Jun Wei Ng. Distributed Gaussian Processes. 2015

Comparison between two approximation methods

- $f(x) = GP(0, K_{se}(1, 0.1))$
- $y(x) = f(x) + (0.3)$*rand
- N = 1000
- 10 fold Cross-validation



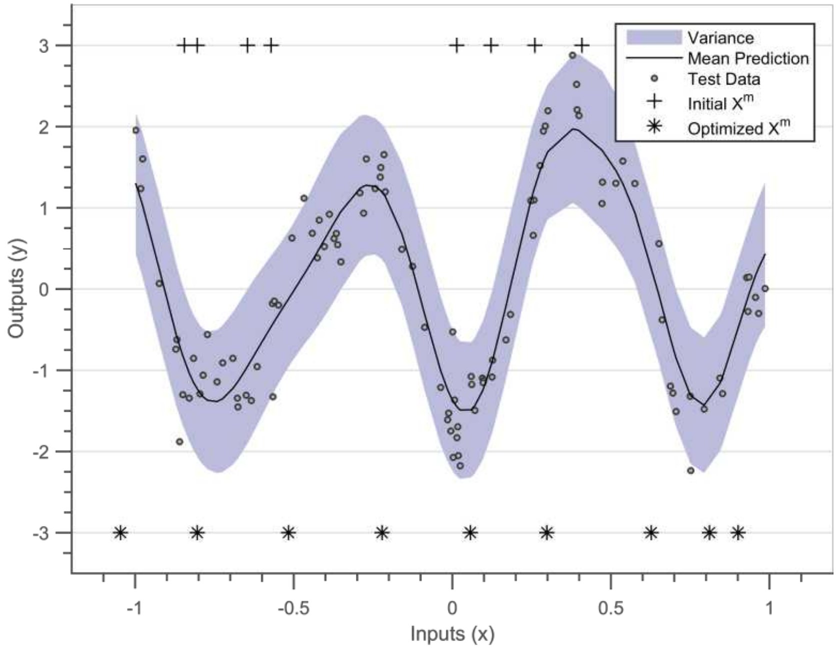
Inducing inputs



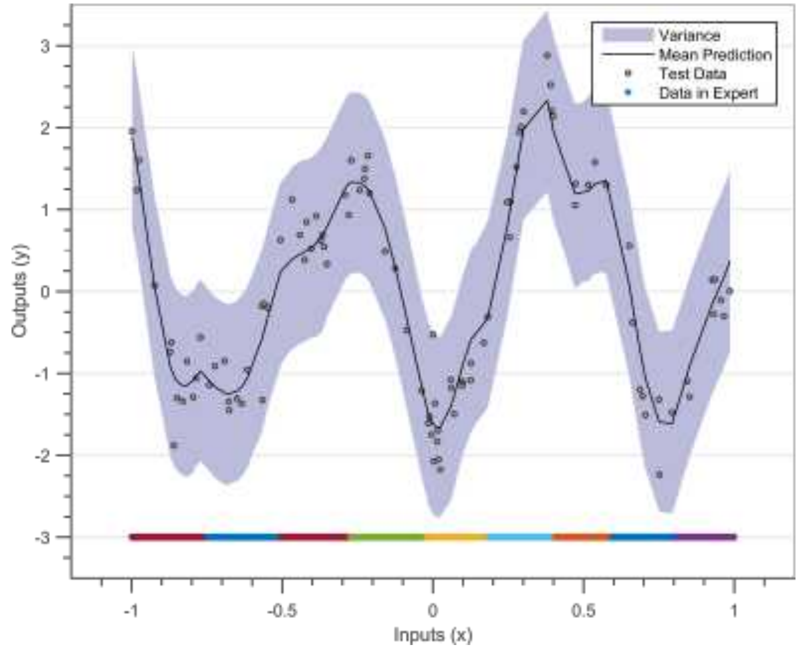
Distributed GP

Comparison between two approximation methods

- $f(x) = GP(0, K_{se}(1, 0.1))$
- $y(x) = f(x) + (0.3)*rand$
- $N = 1000$
- 10 fold Cross-validation



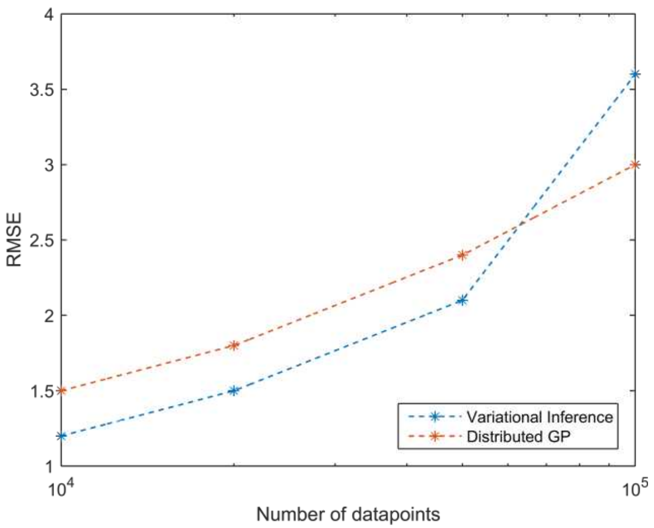
Inducing inputs

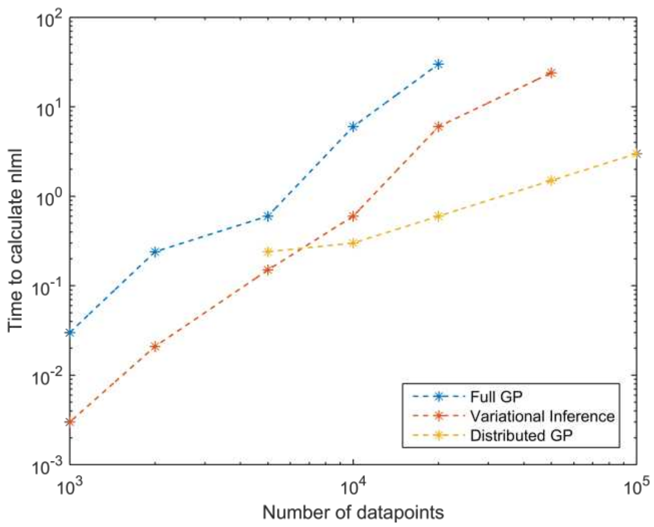


Distributed GP

Scaling solutions to a GP: Sparse Gaussian Process with inducing inputs

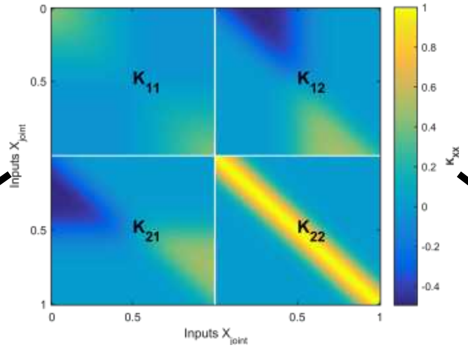
		Learning Computational requirements
Standard	n points	$O(n^3)$
SPGP	m < n pseudo inputs	$O(nm^2)$
DGP	m < n points per P experts	$O(Pm^3)$

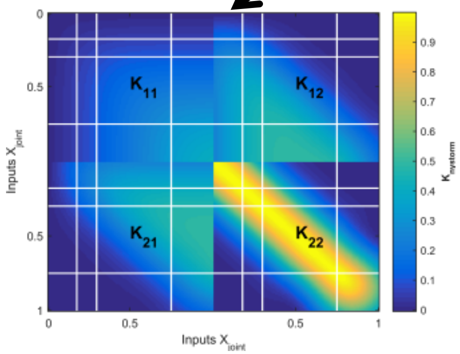


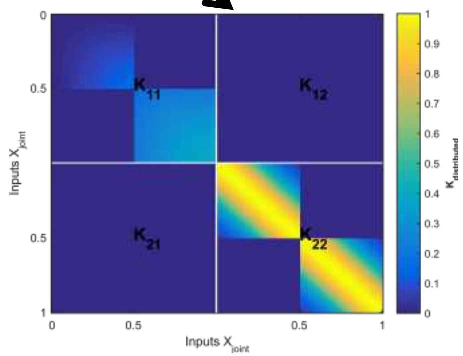


Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In In Artificial Intelligence and Statistics 2009
Christopher KI Williams et Matthias Seeger. Using the Nyström method to speed up kernel machines. In Advances in neural information processing systems, 2001

Scaling solutions to Multi-task GP







Inducing inputs
Same inducing inputs for both outputs

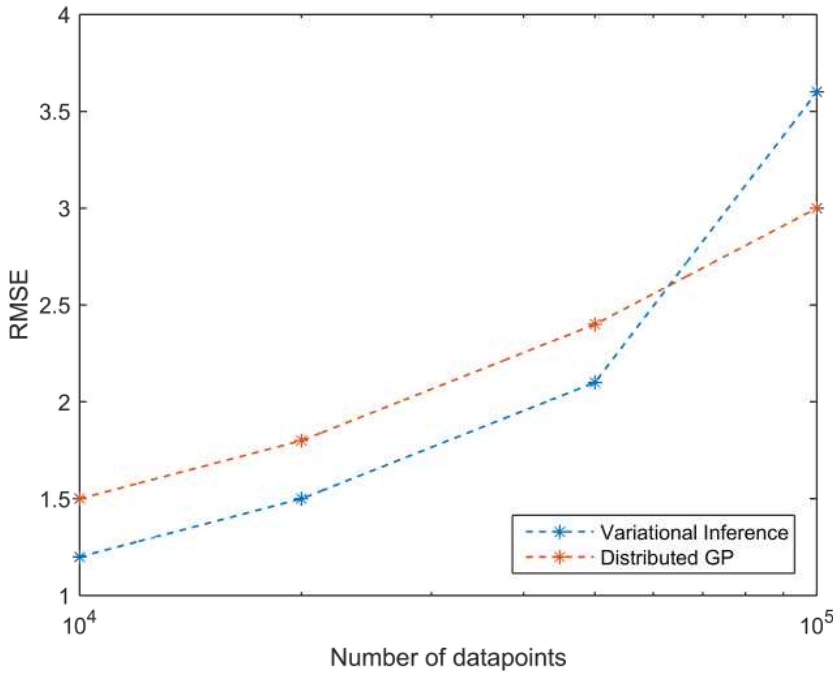
Distributed GP
Same experts for both outputs

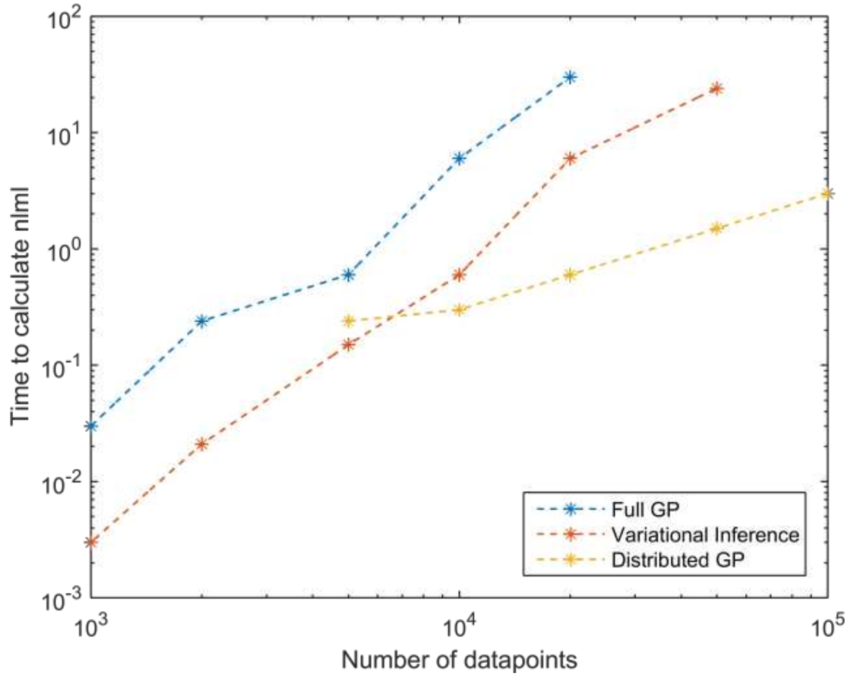
ICPRAM 2016 Sparse Physics-Based Gaussian Process for Multiple outputs
LNCS 2017 Approximate inference in related multi-output Gaussian Process Regression

Scaling solutions to Multi-task GP

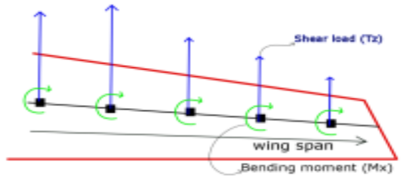
$$f_2 = \frac{df_1}{dx}$$

RMSE calculated by 75% training set 25% test set



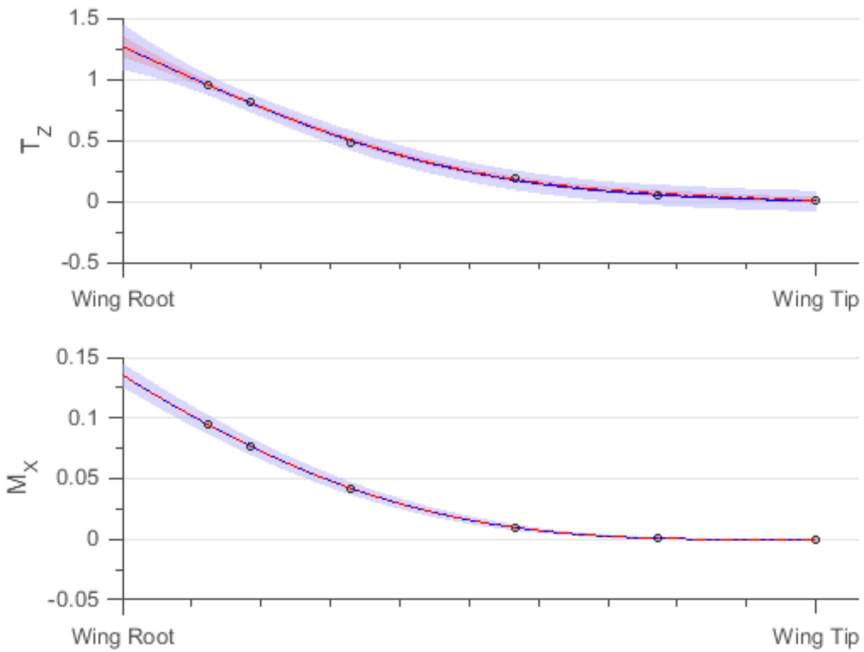


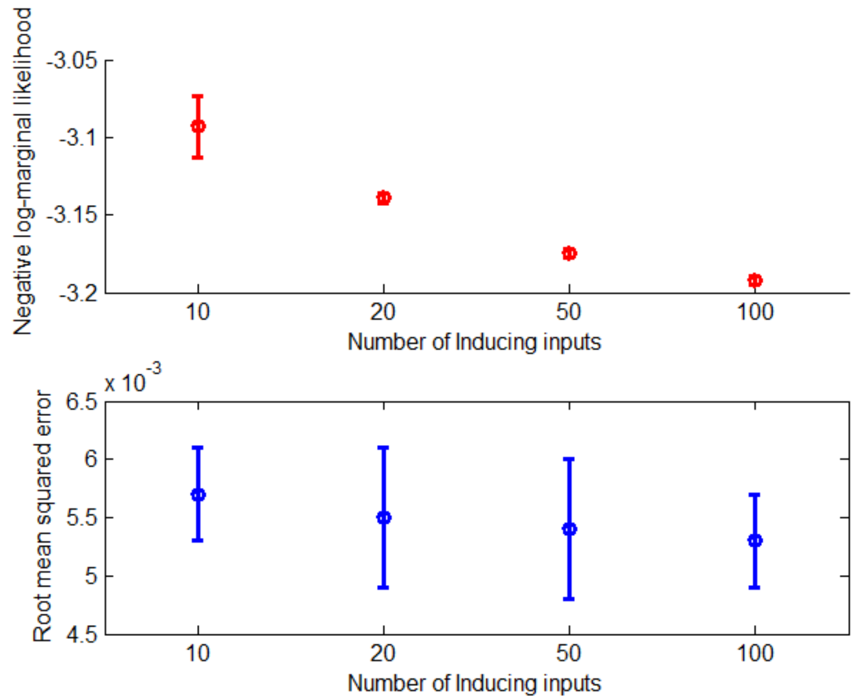
Flight Test: Scaling solutions to Multi-task GP



$$Mx = \int_{\eta}^{\eta_{edge}} (x - \eta)\, Tz\, dx$$

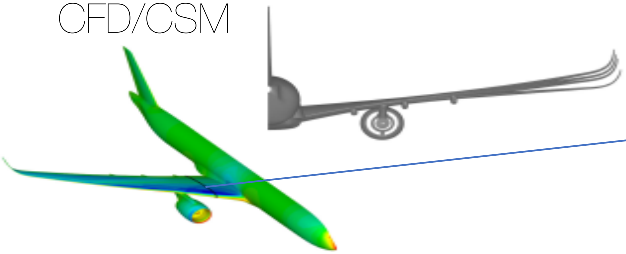
N = 8800

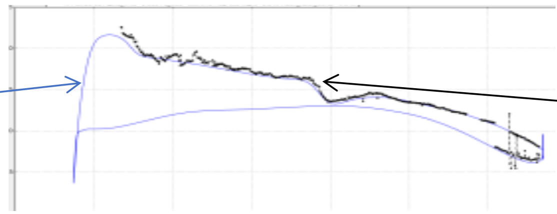




A350-1000 flight test analysis

CFD/CSM



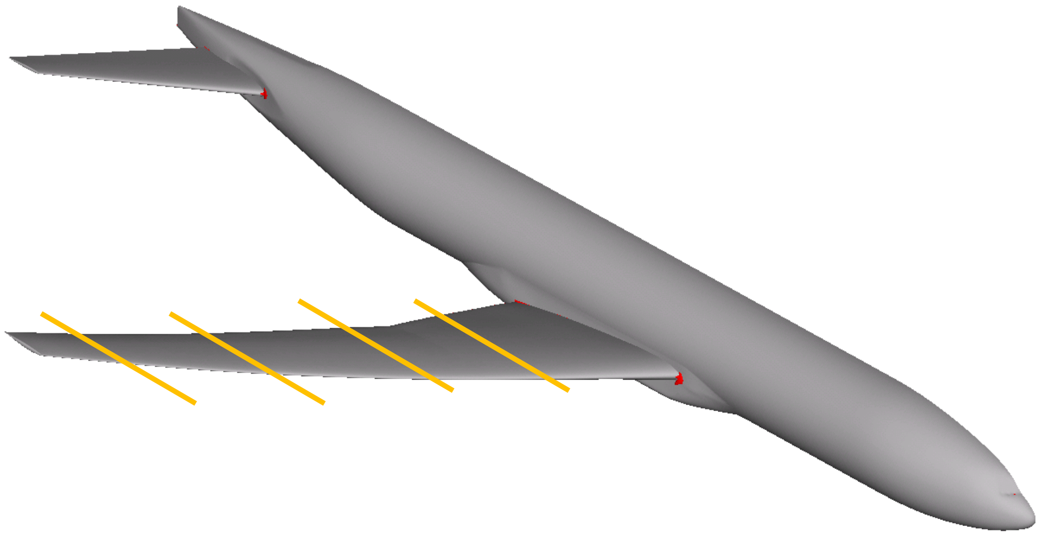


F/T

- Comparing these two sources of data during flight tests enables us to check that the A/C behaves as expected, and that our understanding of its physics (aerodynamics, loads, wing shape) is correct.
- By doing this comparison live in telemetry, we can interact with the ongoing flight test, and optimize configurations (VC/DFS) if needed, for bringing the A/C behaviour as close as possible to design intent.
- These comparisons between CFD/CSM and Flight Test measurements have to be done at identical values of flight parameters : Mach, Alpha, Flight level, VC configuration… so we need to be able to plot instantly the CFD/CSM data for any given combination of these parameters.

Experimental dataset

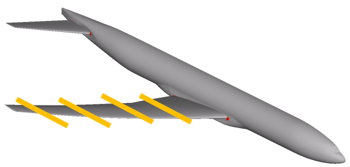
- https://commonresearchmodel.larc.nasa.gov/

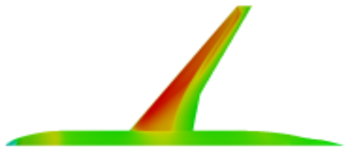


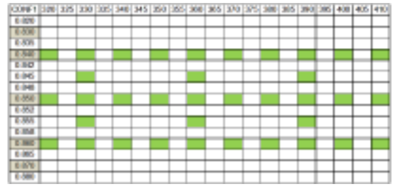
- Simulations run using : Elsa – kOmega-SST
- Same as the one proposed during drag prediction workshop
- Gives better interaction between model fuselage and wing

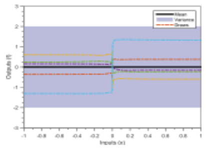
- Alpha = [1 : 0.1 : 3] = **21 alphas**
- Mach = [0.84 : 0.005: 0.86] = **5 machs**
- yLocationCuts = [6.03, 11.99, 17.76, 27.85]

POD vs GP









Pressure dataset $P_i(d_j)$ → Decompose into mode shapes ϕ and amplitude α → Keep Mode shapes constant while Interpolating α between parameters → Linear summation to predict pressures at new points

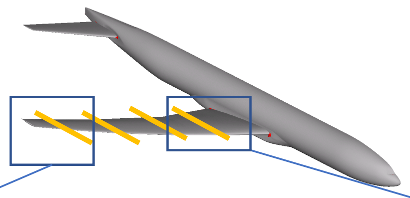
POD

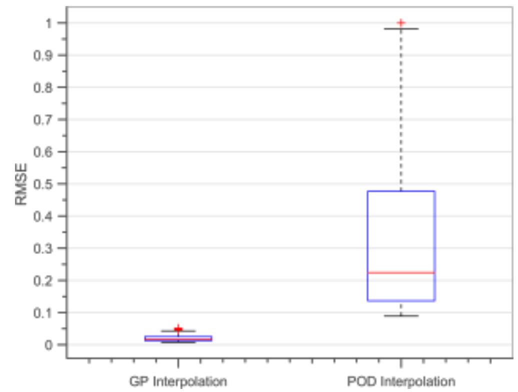
Build a GP model between → Use Neural network Covariance for defining family of functions → Use DGP to accomodate for the huge number of datapoints → Calculate the posterior at new points

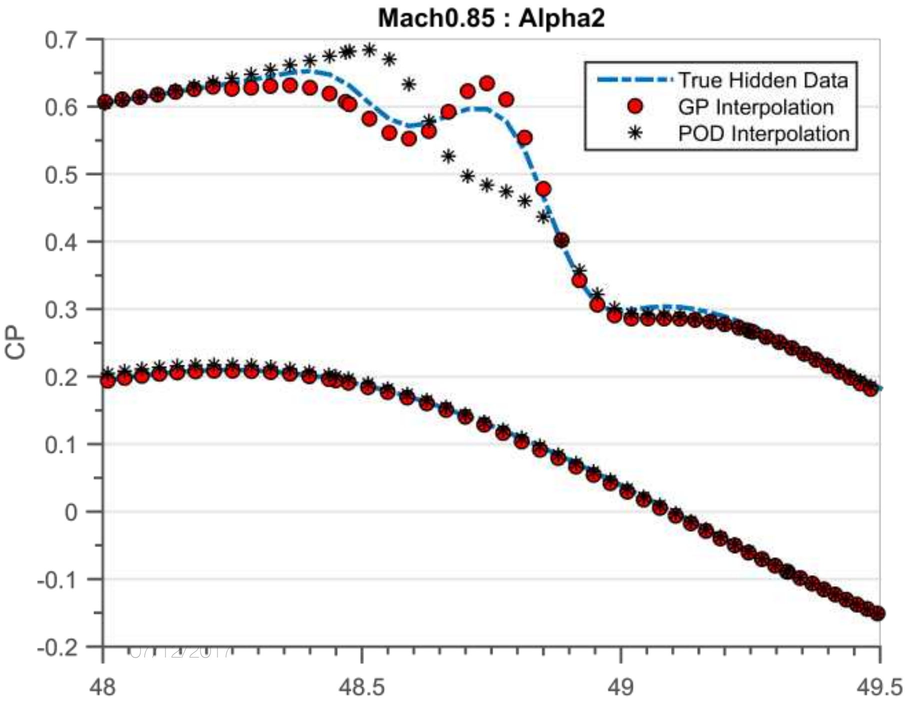
dGP

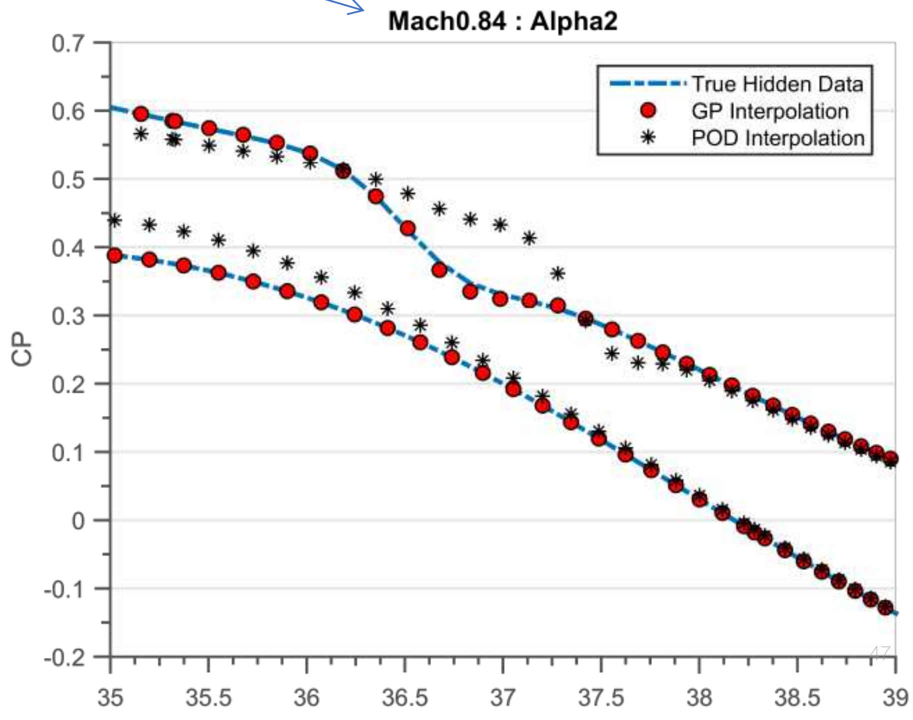
Radford M Neal. Bayesian learning for neural networks, volume 118. Springer Science & Business Media, 2012

Comparison of results: Cut1 & 4

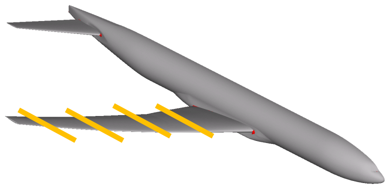


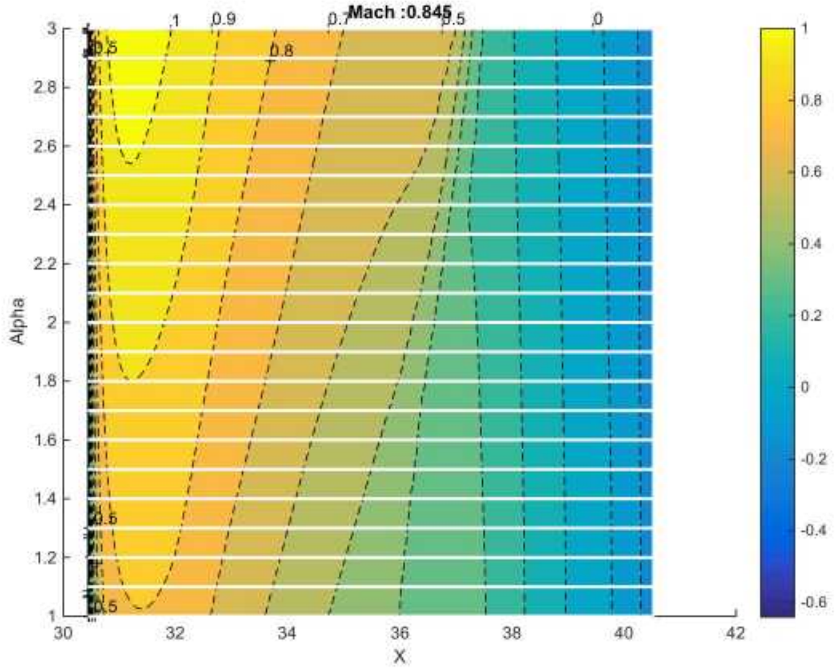


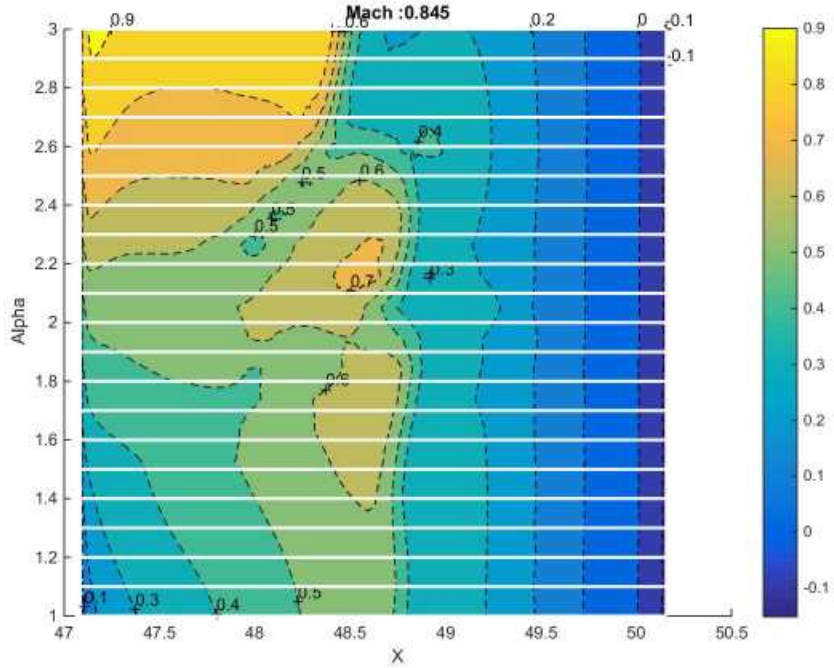




Comparison across Cuts







HPC & Big Data technology

- Work done on Matlab « MDCS » - HPC architecture
- Currently it could be exported easily to support their own cloud computing and structures
  - Amazon EC2 or dedicated Cloud
  - « Tall arrays » data structures

- Possibility to change language / environment
  - Worth it? The Data scientist dilemma

Conclusions

- This research is a good example of Strong link between academics and industry

→PhD of Ankit Chiplunkar funded by AIRBUS via ANRT

- This research is a good example of Strong link between Aerospace Sciences and Machine Learning

→PhD of Ankit Chiplunkar co advise with E. Rachelson (ISAE-SUPAERO/DISC)

- HPC accelerates industrial applications
  - No need to spend excessive time selecting subcases
- Lots of possible new applications (crack prediction…)
  - GP creates surrogate models ideal for industrial applications
  - Great theoretical support for identification
- Since 2010, new courses at SUPAERO including: Big Data / Multidisciplinary Design Optimization

Thanks