A Digital Twin of the Grand'Maison Thrust Bearing

Nicolas Tardieu

EDF R&D Ermes

Contributors : EDF CIH PY Couzon,

EDF DTG A Kuczkowiak, T Lora-Ronco, C Vandalle,

EDF R&D MA Hassini, M Baudin, S Zhang, JC Clément

OpenTURNS user day 2023

Outline

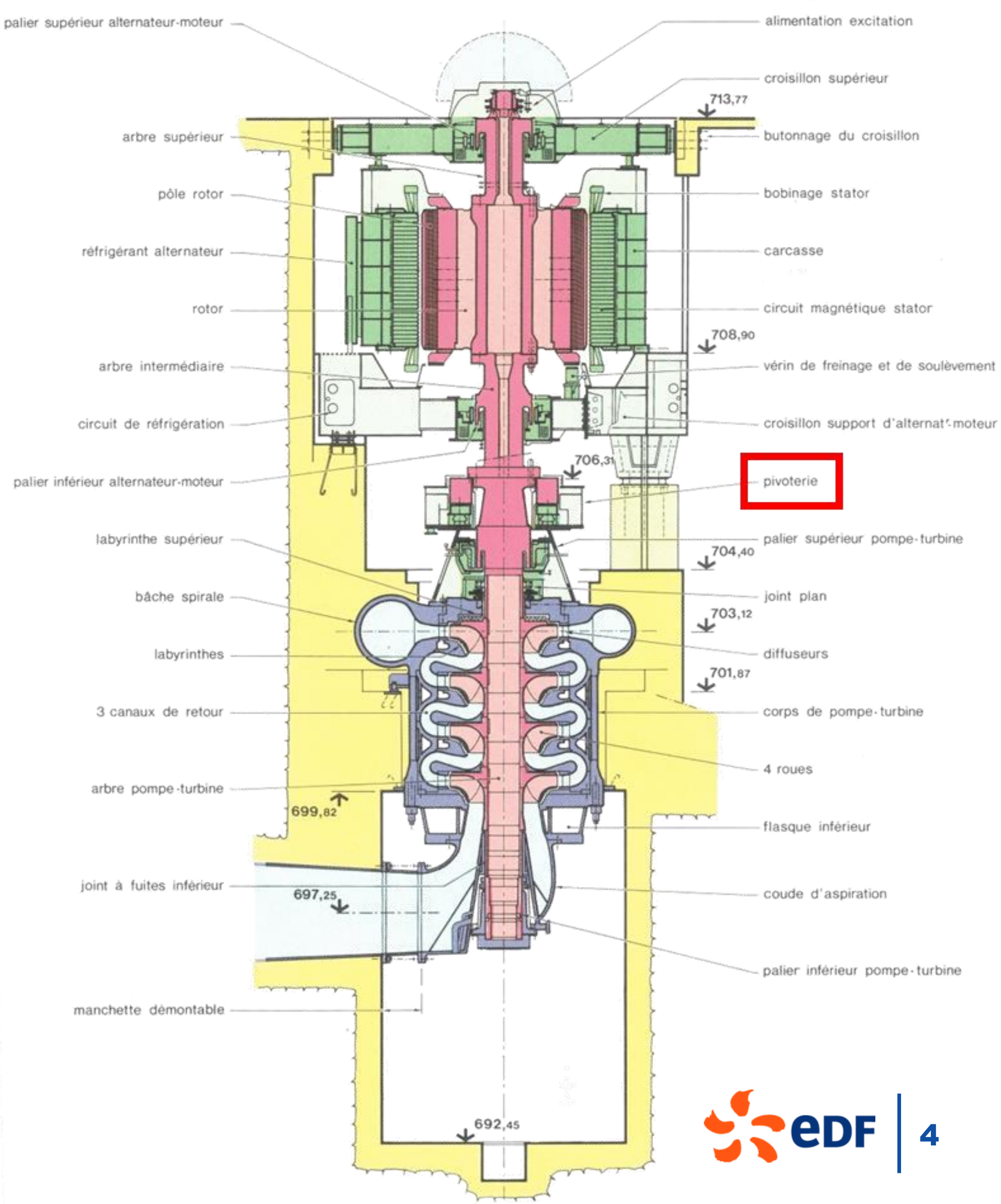
- Industrial Context
- Building the Digital Twin
  - HiFi Model
  - Metamodel
- Using it
  - For Uncertainty Propagation
  - To Monitor a Given Machine
- Conclusion and Outlook

Outline

- Industrial Context
- Building the Digital Twin
  - HiFi Model
  - Metamodel
- Using it
  - For Uncertainty Propagation
  - To Monitor a Given Machine
- Conclusion and Outlook

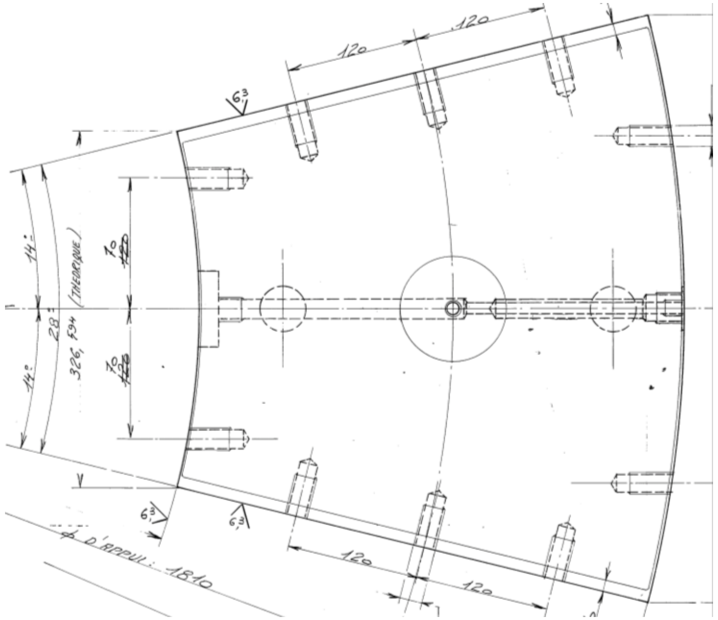
Industrial Context

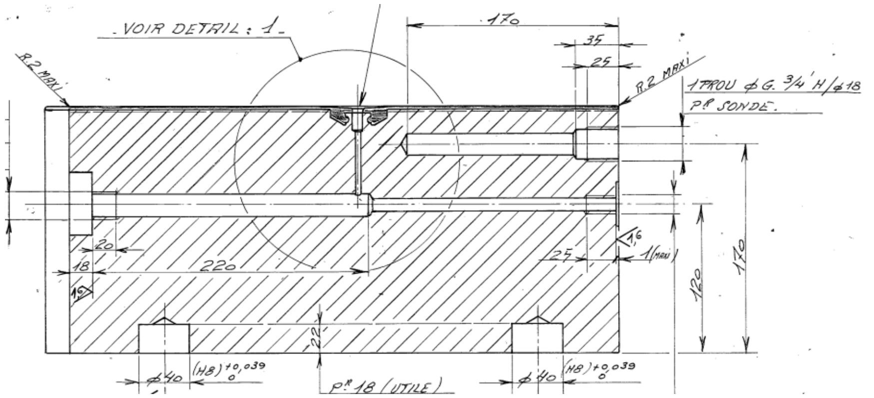
- Hydraulic turbine thrust bearing
  - Key function
    - axial load transfer
  - Costly damage
    - 7M€ from 2016 to 2018



Industrial Context

- Monitored by a single temperature sensor
    - Placed 3cm below the surface and centered in the bearing





Industrial Context

- Digital Twin – R&D vision [2]
  - Possibly high-fidelity model
    - Dedicated to a specific targeted problem
  - Powered by measurements
    - To detect a malfunction (monitoring) or analyze a specific situation (diagnosis)
- Expectation vis-à-vis the Digital Twin of the thrust bearing of Grand'Maison
  - Monitoring

[2] Vision of the ERMES department on the Digital Twin , EDF Technical report H 6125-1722-2021-01109-FR

Outline

- Industrial Context
- Building the Digital Twin
    - HiFi Model
    - Metamodel
- Using it
    - For Uncertainty Propagation
    - To Monitor a Given Machine
- Conclusion and Outlook

High-fidelity Model

Lubrication
Mechanics
Computational Science

- Thermo-Elasto-Hydro-Dynamic (TEHD) model of the thrust bearing by coupling between Legos (lubrication) and code_aster (mechanical) software
  - Multi-physics, multi-scale
- Formulation TEHD

Coupled formulation
Finite volumes and
Finite Elements

TE…

$$\boldsymbol{\sigma} = \boldsymbol{C_{el}\epsilon} - \alpha\Delta \boldsymbol{T C_{el} I} \qquad \begin{cases} \rho C_p \dfrac{\partial T}{\partial t} + div(\phi) = 0 \\ \phi = -\lambda \cdot \nabla T \end{cases}$$

…HD

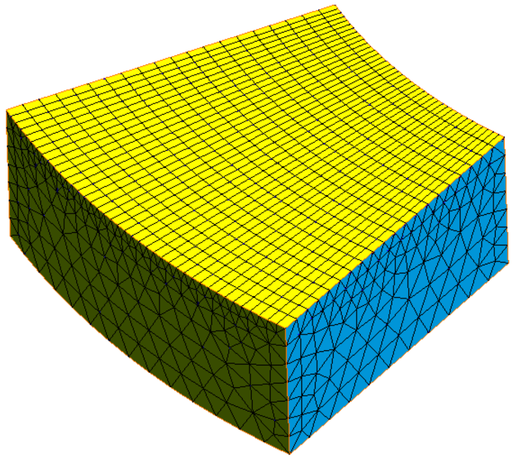
$$\frac{\partial}{\partial x}\left(G_1 \frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial z}\left(G_1 \frac{\partial p}{\partial z}\right) = U \frac{\partial}{\partial x}[(1-\theta)G_2] + \frac{\partial[(1-\theta)\rho h]}{\partial t}$$

$$\rho C_p \left[\frac{\partial T}{\partial t} + \frac{\partial(uT)}{\partial x} + \frac{\partial(vT)}{\partial y} + \frac{\partial(wT)}{\partial z}\right] = \lambda \frac{\partial^2 T}{\partial y^2} + \mu\left[\left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2\right]$$

- Highly non-linear simulation
  - Solution time between 20 min and more than 2h
- Need to drastically reduce the evaluation time

SALOME powered

Model input parameters



- Pad geometry
- Lubricant
  - Lubricant density (RO) [kg/m3]
    - 853.0
  - Specific heat of the lubricant (CHALEUR) [J/kg/K]
    - 2000.0
  - Thermal conduction coefficient (KHL) [W/m/K]
    - 0.13
  - Fluid supply temperature (TALIM) [degree C]
    - 45.0 ∈ [30; 55]

Model input parameters

- Velocity
  - Collar angular velocity (OMEGA) [tr/mn]
    - 600.0
- Loads
  - Norm of external static load (WES) [N]
    - 539550.0 ∈ [250.E3; 540.E3]
- Temperature
  - Collar constant temperature (TABR) [degree C]
    - 55.0 ∈ TALIM+[0; 50]
  - Oil recycling coefficient (RECYC) [%]
    - 85 ∈ [0; 100]

Model input parameters

- Temperature
  - Ambient T. (TAMBP) [degree C]
    - 45.0
  - Inlet T. (TEF) [degree C]
    - 45.0
- Boundary conditions
  - Inlet pressure (PLP) [Pa]
    - 100000.0

Model output parameters

- Relevant physical quantities
  - Hmin, Tmax, Pmax in oil film

- Quantities to be compared with measurements
  - Tprobe

Methodology

- Define a HiFi Thermo-Elasto-Hydro-Dynamic (TEHD) model of the bearing
  - Provided by EDF engineers
- Vary the input parameters of the model and compute the relevant quantities
- Build a metamodel based on previous results

Load
T oil entry
+ other uncertain parameters

Metamodel

h, T, P of the oil film (inaccessible to measurement)
T metal probe

Methodology

Statistics

Computational Science

- Definition of the high-fidelity model G of the bearing
- Identification
  - of the input parameters that characterize the equipment's operation
    - Drawing on available measurements
  - Of the outputs that characterize the equipment's health
    - If possible, to compare with measurements

- Construction of the metamodel $\tilde{G}$
  - Parameters of the metamodel (e.g. degree of the polynomial)
  - Sufficiently large size n_a of the design of learning experiments
  - Generation of a design of experiments $\mathcal{X}_a=\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n_a)}\}$ associated with random vector $\boldsymbol{X}$
  - Evaluation of model outputs $\mathcal{Y}_a=\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n_a)}\}$ by model evaluation G
  - Construction of metamodel $\tilde{G}$, which approximates G with accuracy to be determined

Methodology

- Validation of the metamodel $\tilde{G}$
  - Selection of a sufficiently large validation design size n_v
  - Generation of a design of experiments $\mathcal{X}_v = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n_v)}\}$ associated with the random vector $\boldsymbol{X}$
  - Evaluation of model output $\mathcal{Y}_v = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n_v)}\}$ by evaluation of model G
  - Evaluation of metamodel outputs $\mathcal{Y}'_v = \{\mathbf{y}'^{(1)}, \ldots, \mathbf{y}'^{(n_v)}\}$ by evaluation of metamodel $\tilde{G}$
  - Determination of the deviation between $\mathcal{Y}_v$ and $\mathcal{Y}'_v$, which must satisfy a given criterion

OpenTURNS metamodel

Computational Science

Python Shell SLURM

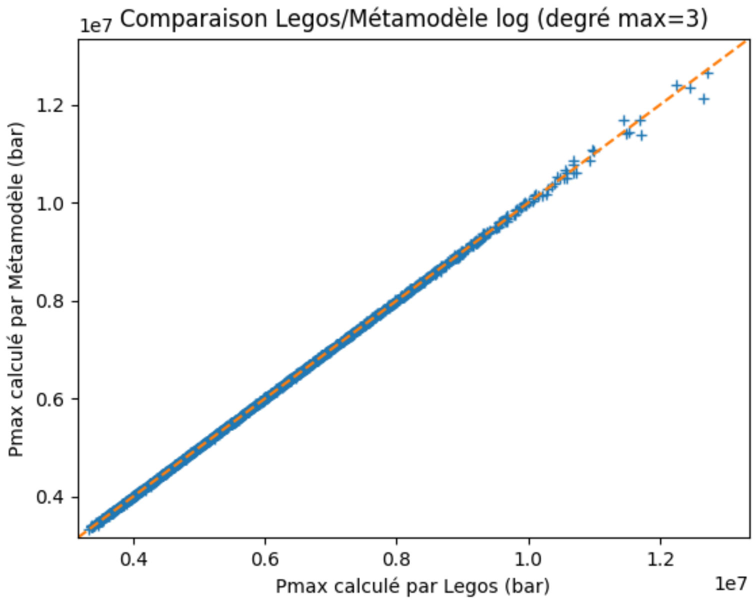
- Input:
  - Load
  - T oil supply
  - T shaft
  - Recycling rate
- Range of variation
  - [250; 540] kN
  - [30; 55] °C
  - Talim + [0; 50] °C
  - [0; 100] %
- Output
  - Pmax
  - Tmax
  - Hmin
  - Tprobe in 3 points around the theoretical position

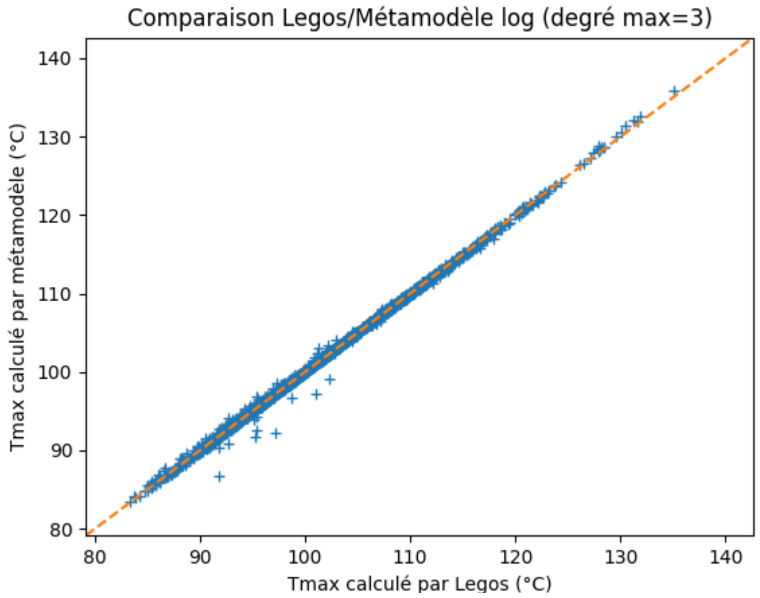
- For the inputs: random draw according to uniform distribution
- Polynomial chaos
- 2000 simulations to build the model
  - ✓ Single solution time: between 20 min and 2h
  - ✓ Parallel solution on EDF R&D cluster: 2500 results in 1 day
- 500 simulations for model validation

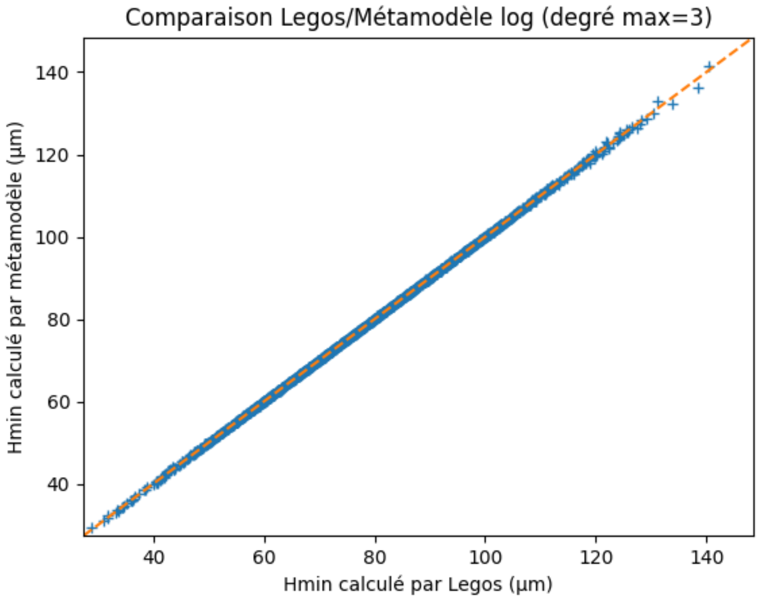
Convergence was achieved for 86% of the simulations

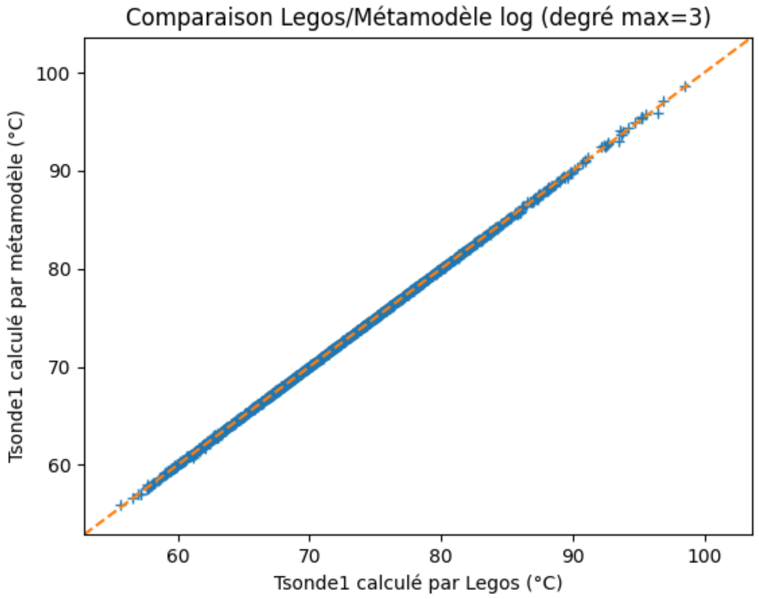
SALOME powered

OpenTURNS metamodel









OpenTURNS metamodel

- Reliability
  - Learning base of 2000 simulations
    - Q_2 = [0.999913, 0.998122, 0.99993, 0.999933, 0.999933, 0.999933]
  - Validation base of 500 simulations
    - Q_2 = [0.999915, 0.998537, 0.999917, 0.999931, 0.999931, 0.999931]
- Performance
  - Metamodel evaluation : ~3.e-5 s
- Export
  - Easily export the metamodel as a polynomial
  - But evaluation in OpenTURNS is preferred (flexibility, performance, scalability)

OpenTURNS metamodel

```python
def polyGM(wes, talim, dt, re):
    def _poly(xx0, xx1, xx2, xx3):
        x0 = 6.896551724137930 81e-06*(xx0-395000)
        x1 = 0.08000000000000001665*(xx1-42.5)
        x2 = 0.04000000000000000833*(xx2-25)
        x3 = 0.02000000000000000416*(xx3-50)
        return np.array([15.5993,102.464,4.34388,72.5981,72.3595,72.8033]) + np.array([0.269326,4.93465,-0.181507,2.10261,2.09978,2.10061]) * (1.73205 * x0) + np.array([0.0337926,4.23664,-0.114819,5.24478,5.27776,5.20956]) * (1.73205 * x1) + np.array([0.000267558,0.837838,-0.0801463,0.701439,0.722108,0.682507]) * (1.73205 * x2) + np.array([0.0710881,4.44855,-0.113681,4.92222,4.96248,4.8793]) * (1.73205 * x3) + np.array([-0.0187358,-0.169745,0.00782219,-0.0835755,-0.082128,-0.0848066]) * (-1.11803 + 3.3541 * x0**2) + np.array([0.00969675,0.350995,-0.0119131,0.107225,0.108172,0.105475]) * ((1.73205 * x0) * (1.73205 * x1)) + np.array([0.0050355,0.797338,-0.0204107,0.143264,0.14622,0.140077]) * ((1.73205 * x0) * (1.73205 * x2)) + np.array([0.0163678,0.28551,-0.0129543,0.260619,0.261321,0.259268]) * ((1.73205 * x0) * (1.73205 * x3)) + np.array([0.00370576,0.404428,-0.00209024,0.21477,0.213017,0.216245]) * (-1.11803 + 3.3541 * x1**2) + np.array([0.00732481,1.19426,-0.0117639,0.406104,0.407277,0.404315]) * ((1.73205 * x1) * (1.73205 * x2)) + np.array([0.00612331,0.203418,0.00237175,0.149927,0.1417,0.158002]) * ((1.73205 * x1) * (1.73205 * x3)) + np.array([0.00421135,0.600557,-0.00868402,0.111148,0.112916,0.10916]) * (-1.11803 + 3.3541 * x2**2) + np.array([0.0166096,1.33418,-0.0228652,1.1064,1.10997,1.10174]) * ((1.73205 * x2) * (1.73205 * x3)) + np.array([0.0113299,0.499431,-0.0117073,0.564053,0.563604,0.564456]) * (-1.11803 + 3.3541 * x3**2) + np.array([0.00263682,0,-0.00090607,0.0126441,0.0123289,0.0129106]) * (-3.96863 * x0 + 6.61438 * x0**3) + np.array([0.000849687,0,-0.000278201,0.0162081,0.0160392,0.0162322]) * ((-1.11803 + 3.3541 * x0**2) * (1.73205 * x1)) + np.array([0.0017717,0.0931773,-0.00126602,0.0353856,0.0354583,0.0350407]) * ((-1.11803 + 3.3541 * x0**2) * (1.73205 * x2)) + np.array([0,-0.124199,0.00100414,-0.0132816,-0.0139134,-0.0126875]) * ((-1.11803 + 3.3541 * x0**2) * (1.73205 * x3)) + np.array([0.001056,-0.0385535,-0.000818417,0.0058737,0.00601778,0.00555466]) * ((1.73205 * x0) * (-1.11803 + 3.3541 * x1**2)) + np.array([0.00354928,0.159613,-0.00430059,0.0805405,0.0812053,0.0793082]) * ((1.73205 * x0) * (1.73205 * x1) * (1.73205 * x2)) + np.array([0.000922633,-0.183642,0.000669539,-0.0304405,-0.0308419,-0.0301453]) * ((1.73205 * x0) * (1.73205 * x1) * (1.73205 * x3)) + np.array([0.0020289,0.136922,-0.00339964,0.0561628,0.0567422,0.0552875]) * ((1.73205 * x0) * (-1.11803 + 3.3541 * x2**2)) + np.array([0.00463657,0.0886418,-0.003131,0.050864,0.0508911,0.0503563]) * ((1.73205 * x0) * (1.73205 * x2) * (1.73205 * x3)) + np.array([0.000893778,-0.203872,0.00119797,-0.0634035,-0.0644517,-0.062312]) * ((1.73205 * x0) * (-1.11803 + 3.3541 * x3**2)) + np.array([0,-0.0395469,-0.000180722,-0.0129769,-0.0128067,-0.0131769]) * (-3.96863 * x1 + 6.61438 * x1**3) + np.array([0.0012834,0.0779777,-0.00162338,0.0174295,0.017607,0.0169858]) * ((-1.11803 + 3.3541 * x1**2) * (1.73205 * x2)) + np.array([-0.000229625,-0.150633,0,-0.0710436,-0.0702665,-0.0718228]) * ((-1.11803 + 3.3541 * x1**2) * (1.73205 * x3)) + np.array([0.00150281,0.268943,-0.00242826,0.0504768,0.0507294,0.0498739]) * ((1.73205 * x1) * (-1.11803 + 3.3541 * x2**2)) + np.array([0.00358788,-0.0451455,-0.00314931,0.0637257,0.0636546,0.0634541]) * ((1.73205 * x1) * (1.73205 * x2) * (1.73205 * x3)) + np.array([-0.00134542,-0.190119,0.00259862,-0.13956,-0.139932,-0.139095]) * ((1.73205 * x1) * (-1.11803 + 3.3541 * x3**2)) + np.array([0.000498105,0.161067,-0.00113193,0.0211751,0.0213776,0.0208633]) * (-3.96863 * x2 + 6.61438 * x2**3) + np.array([0.00161123,0.0593789,-0.00152871,0.0573089,0.056972,0.0573396]) * ((-1.11803 + 3.3541 * x2**2) * (1.73205 * x3)) + np.array([0.0025699,0.0820464,-0.00198255,0.0570014,0.0561445,0.0578167]) * ((1.73205 * x2) * (-1.11803 + 3.3541 * x3**2)) + np.array([0.000469852,-0.0628267,0.000356894,-0.00641126,-0.00696054,-0.00577729]) * (-3.96863 * x3 + 6.61438 * x3**3)

    resu = _poly(wes, talim, dt, re)
    Pmaxlog, Tmax, Hminlog, Tsonde1, Tsonde2, Tsonde3 = resu
    return np.exp(Pmaxlog), Tmax, np.exp(Hminlog), Tsonde1, Tsonde2, Tsonde3
```

Benefits of the Metamodel

- Extreme speed of evaluation with good accuracy

- Easily allows
  - Uncertainty propagation to assess the sensitivity of the Digital Twin
  - Customizing the Digital Twin for a specific machine

Outline

- Industrial Context
- Building the Digital Twin
  - HiFi Model
  - Metamodel
- Using it
  - For Uncertainty Propagation
  - To Monitor a Given Machine
- Conclusion and Outlook

Uncertainty Propagation

Statistics

- Input:
  - Load
  - T oil supply
  - T shaft
  - Recycling rate

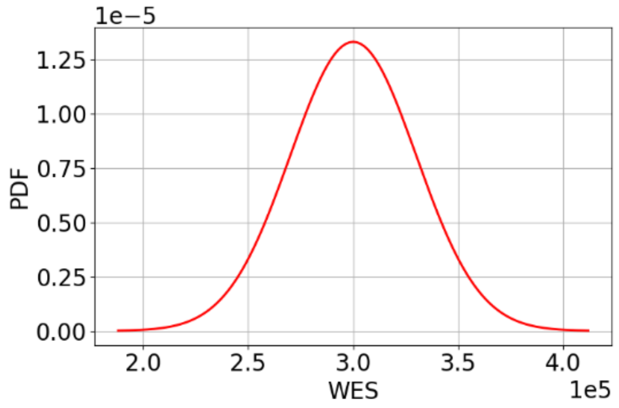
- Range of variation
  - $\mathcal{N}$(WES, 30.e3) kN
  - $\mathcal{N}$(Talim, 2) °C
  - Talim + $\mathcal{N}$(DT, 2) °C
  - $\mathcal{N}$(RECYC, 20) %

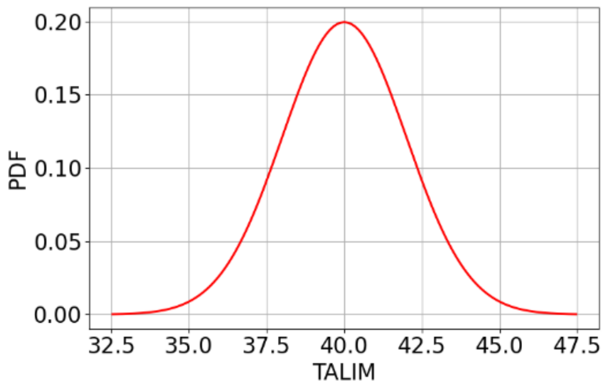
- Output
  - Pmax
  - Tmax
  - Hmin
  - Tprobe in 3 points around the theoretical position

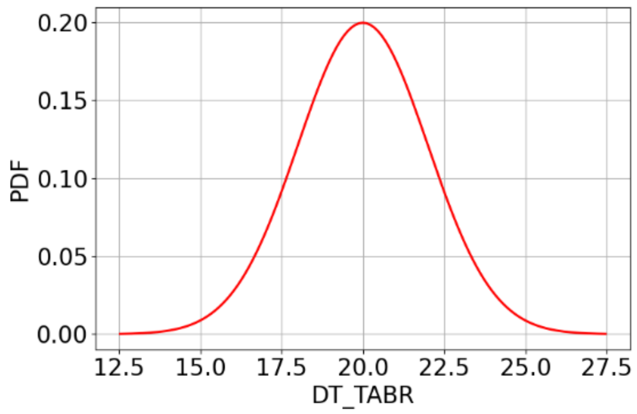
- Propagation of uncertainty on output parameters
  - Drawing of 5000 samples of input parameters
  - Outputs Evaluations
- Outputs with uncertainty
  - Allows quantify the risk of exceeding a given threshold
    - No “*Yes*” or “*No*” but “*You have x% of chance to exceed the threshold*”

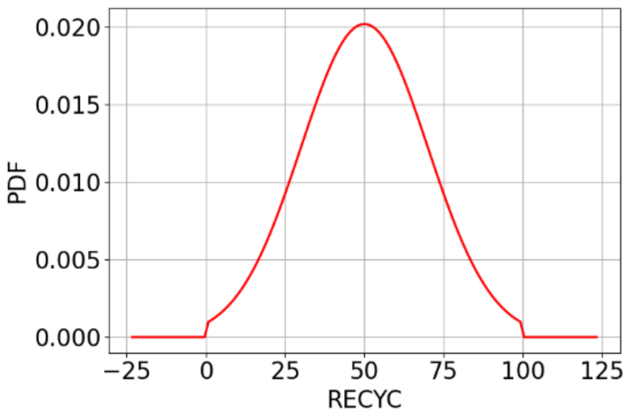
Uncertainty Propagation

- Input parameters



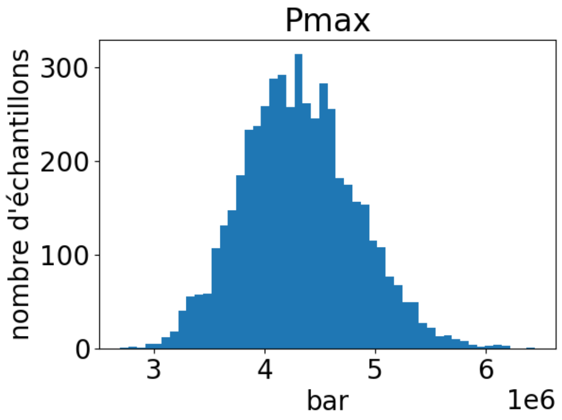


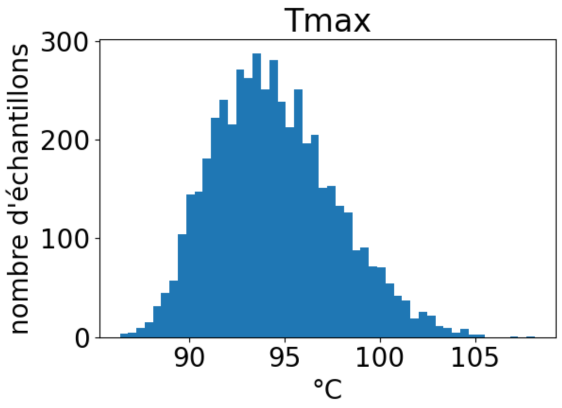


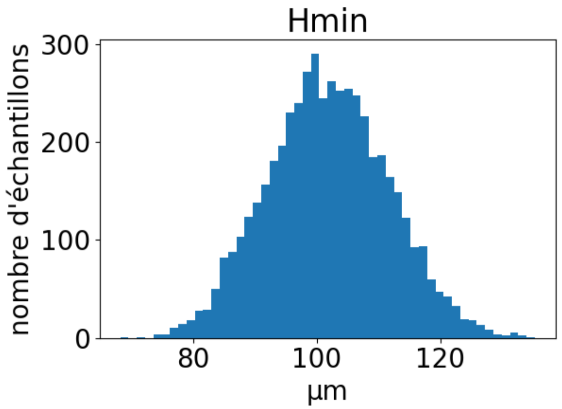


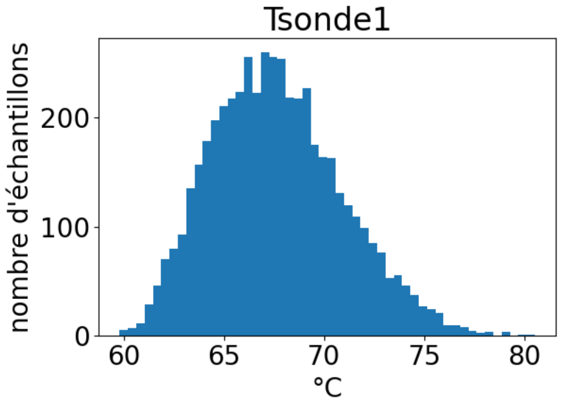
Uncertainty Propagation

- Outputs









Customizing the Digital Twin

- Load measurements carried out at GM in September 2020 by EDF/DTG
  - G06 chosen for this test
  - Temperatures also recorded in the monitoring system during the pumping and turbining measurements

$$\boldsymbol{y}^{obs} = \begin{Bmatrix} T_{\text{measure}}^{\text{pumping}} \\ T_{\text{measure}}^{\text{turbining}} \end{Bmatrix}$$

- Digital Twin's fitting taking into account uncertainties about parameters

$$\boldsymbol{x} = \begin{Bmatrix} \begin{Bmatrix} \text{WES} \\ \text{TALIM} \\ \text{TABR} \\ \text{RECYC} \end{Bmatrix}^{\text{pumping}} \\ \begin{Bmatrix} \text{WES} \\ \text{TALIM} \\ \text{TABR} \\ \text{RECYC} \end{Bmatrix}^{\text{turbining}} \end{Bmatrix}$$

$$\boldsymbol{y} = \begin{Bmatrix} T_{\text{twin}}^{\text{pumping}} \\ T_{\text{twin}}^{\text{turbining}} \end{Bmatrix}$$

3DVar Approach

Optimisation

- Functional to minimize

$$\min_{\boldsymbol{x}} J(\boldsymbol{x})$$

$$J(\boldsymbol{x}) = \left(\boldsymbol{x} - \boldsymbol{x}^b\right)^T \boldsymbol{B}^{-1}\left(\boldsymbol{x} - \boldsymbol{x}^b\right) + \left(\boldsymbol{y} - \boldsymbol{y}^{obs}\right)^T \boldsymbol{R}^{-1}\left(\boldsymbol{y} - \boldsymbol{y}^{obs}\right)$$

  - $\boldsymbol{x}^b$ initial value of $\boldsymbol{x}$
  - $\boldsymbol{B}$ and $\boldsymbol{R}$ Covariance matrices of background noise error and observational error

  - Implemented in the ADAO Module of the Salomé Platfom

SALOME powered

Customizing the Digital Twin

- Initial value for $\boldsymbol{x}$
  - Turbining
    - WES = 398 tons, TALIM=48.8 °C, TABR=58.8°C, RECYC=80%
  - Pumping
    - WES = 492 tons, TALIM=48.1 °C, TABR=58.1°C, RECYC=80%
- Identified values for $\boldsymbol{x}$
  - Turbining
    - WES = 373 tons, TALIM=47.6 °C, TABR=59.6°C, RECYC=30%
  - Pumping
    - WES = 464 tons, TALIM=47.2 °C, TABR=55.3°C, RECYC=36%

  - Significant decreases in variance for these parameters

Customizing the Digital Twin

- Comparing Digital Twin - Measures
  - Pumping
    - Tmetal_metaModel = 71.9 et Tmetal_PI = 65.1 => 10% error
      - Pmax= 5.5 MPa, Tmax=100. °C, Hmin=85. µm
  - Turbining
    - Tmetal_metaModel = 74.3 et Tmetal_ PI = 68.7 => 7% error
      - Pmax= 7.3 MPa, Tmax=105. °C, Hmin=72. µm

- Tailored Digital Twin
  - Dedicated to monitoring this specific machine

Outline

- Industrial Context
- Building the Digital Twin
  - HiFi Model
  - Metamodel
- Using it
  - For Uncertainty Propagation
  - To Monitor a Given Machine
- Conclusion and Outlook

Conclusion

- Metamodel construction methodology taking into account uncertainty on input parameters
  - High-fidelity model + metamodel over a range of input parameter variations
- Enables easy propagation of uncertainty
  - Quantifies the risk of exceeding thresholds
- Allows the Digital Twin to be customized for a given machine
  - Access to quantities that are crucial to equipment health, but inaccessible to measurement

- ⚠ Strong multidisciplinary dimension around the Digital Twin technology
- ⚠ All done in the Salome_Meca platform

Outlook

- Evaluation of the Digital Twin on other Grand'Maison machines
- Integration of the evaluation of the load applied to the thrust bearing
  - Work in progress
  - Allows the Digital Twin to be used for monitoring purposes
- Generalization to other machines

Thank you for your attention