Post-doctoral position at EDF R&D, CHATOU, FRANCE:

Bayesian model selection for the valdation of computer codes

1 Industrial context

An industrial company such as EDF bases many of its decisions on the results of quantitative models.

More generally, increasing efforts are devoted to numerical simulations in industrial studies. Indeed, they tend to complement, or even replace, physical experiments, when these are too costly or dangerous.

Such simulations are used for design (or, more generally, optimization) tasks, as well as reliability assessment and risk analysis.

Because the stakes motivating such studies can be considerable, it is important to guarantee that the computer code used for the simulation 'predicts well enough' in a certain sense the physical phenomena under study. Adressing this task is the object of code validation, an active field of research within the communities of engineering and computer experiments.

At EDF R&D, especially inside the department of industrial risk management (MRI), continuous efforts have been made toward this goal over the last years. In particular, a PhD on this subject has been recently completed [Damblin, 2015], leading to the proposition of a generic approach for computer code validation. This includes the calibration of uncertain code parameters, the quantification of prediction errors and their impact on the final use of the simulations.

Several challenges in the implementation of this approach were addressed. Specifically, a statistical calibration procedure adapted to costly computer codes was developped, based on a sequential experimental design strategy. Then, a test was introduced to detect a bias in the predictions of a computer code, due to a lack of accuracy of the underlying physical model. Taking into account such a bias thus allows to improve code-based predictions.

The two above contributions, though important, are far from exhaustive. These may be seen as a first step in the development of a truly generic code vaidation methodology, which could ideally treat *all* identified sources of uncertainty, in the most automated fashion possible.

The objective of this post-doc is to contribute to the development of this approach, focusing on the bias detection test, limited for now to the simplified case of a low-dimensional computer code that is linear with respect to the uncertain parameters

2 Model selection for the validation of computer codes

2.1 Statistical modeling

We assume that the physical system under study can be assimilated to a function:

$$\boldsymbol{r} : \boldsymbol{x} \in \mathbb{R}^q \quad \mapsto \quad r(\boldsymbol{x}) \in \mathbb{R}^d, \tag{1}$$

where $\boldsymbol{x}$ is a vector of controlable (observable) experimental variables, and $\boldsymbol{r}(\boldsymbol{x})$ the output of interest. Likewise, the computer code is assumed to take the form of a parametric function:

$$\boldsymbol{y_\theta} : \boldsymbol{x} \in \mathbb{R}^q \quad \mapsto \quad y_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}^d, \tag{2}$$

where $\boldsymbol{\theta} \in \mathbb{R}^p$ is a vector of unknown parameters, which we can interpret as physical constants controlling the behavior of the system under study.

Moreover, let $\mathbf{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^\top \in M_{n,d}$ be a series of available measures of the response of interest, for a certain set $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in M_{n,q}$ of input values.

A first statistical model derives from the assumption that the computer code can mimic the physical system perfectly, and that the measurement errors $\boldsymbol{\epsilon}_i$ are independently and identically distributed (iid), following a zero-mean normal law with covariance matrix Σ^2, so that we can write:

$$\begin{array}{rcl} \boldsymbol{z}_i & = & \boldsymbol{y}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i) + \boldsymbol{\epsilon}_i, \\ \epsilon_i & \sim & \mathcal{N}(\mathbf{0}, \Sigma^2) \end{array} \tag{3}$$

where $\boldsymbol{\theta}^*$ is the 'true' value of the parameter vector $\boldsymbol{\theta}$. It is worth noting that (3) is in fact a regression model of the data $\mathbf{Z}$ with respect to the inputs $\mathbf{X}$, where the code $\boldsymbol{y_\theta}$ defines the regression function, which can be non-linear and costly.

This first model is to date the most used in practice, though it relies on the rather strong assumption that the code can predict the physical system perfectly. To relax this assumption, [Kennedy and O'Hagan, 2000] suggest to add a term $b(\boldsymbol{x})$, usually called *model error* or *model bias*. Consequently, (3) is replaced by the following, more general, model:

$$\boldsymbol{z}_i \quad = \quad y_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i) + b(\boldsymbol{x}_i) + \boldsymbol{\epsilon}_i. \tag{4}$$

In [Kennedy and O'Hagan, 2000], $\boldsymbol{\theta}^*$ and $b(\boldsymbol{x})$ are simultaneously estimated from both the experimental data Z and a limited number N of runs $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{y}}_i)_{i=1,\ldots,N}$ of the compute code $y_{\boldsymbol{\theta}}(\boldsymbol{x})$, used to build a Gaussian process emulator in order to reduced the computation time. A second Gaussian process is used as a prior disribution for the model bias $b(\boldsymbol{x})$.

The reason usually advocated for the introduction of this additional term is that it allows to correct pure code predictions, considered as insufficiently accurate approximations of the quantity of interest $\boldsymbol{r}(\boldsymbol{x})$. Such an advantage for predictions has been illustrated for instance in [Kennedy and O'Hagan, 2000], [Bayarri et al., 2007] and [Higdon et al., 2008], among others, on several real-life case-studies.

However, to date we have found no formal justification to the introduction of a model bias, even though it raises several important issues:

- Such a bias makes the statistical model much more complicated to estimate and interpret. In particular, model (4) is not *identifiable*, in the sense that it does not allow to estimate both $\boldsymbol{\theta}^*$ and $b(\boldsymbol{x})$ from the data alone[1]. As a matter of fact, it is the Gaussian process prior distribution on $b(\boldsymbol{x})$ which makes this estimation possible, by imposing additional constraints; hence the prior choice is a crucial and delicate task.

- Accounting for model bias is controverisal in the engineering community. This is especially true among numerical analysts who develop computer codes, and who expect them to predict interest quantities in a sufficiently satisfying way, so that adding a bias-correcting term is in general unnecessary.

In view of these questions, it seems important to develop statistical procedures allowing to test the presence or absence of a nonzero model bias between the code outputs an the interest quantities we wish to predict. In [Loeppky et al., 2006], a maximum likelihood ratio test has been proposed to this end. Still, as is usually the case of frequentist tests, this procedure can detect with good certainty an existing model bias, but not conclude to its absence.

This is why we advocate a Bayesian model selection approach, simpler to interpret, and which allows to conclude with similar confidence levels to both the presence and the absence of a model bias. Another advantage of this approach is that it allows, in absence of a clear model choice, to combine predictions from both pure code and bias-corrected predictions.

2.2 Proposed approach

Informally, Bayesian model selection consists in computing the posterior probability that each model is the 'true' one, given the available data. More precisely, note $\mathcal{M}_0$ the model without bias defined by (3), and $\mathcal{M}_1$ the model with bias, defined by (4). Then, after Bayes' theorem, the posterior probability of model $\mathcal{M}_j$ for $j = 0, 1$ is proportional to :

$$\mathbb{P}[\mathcal{M}_j|\boldsymbol{Z}] \quad \propto \quad m(\boldsymbol{Z}|\mathcal{M}_{\mathrm{j}}) \times \mathbb{P}[\mathcal{M}_{\mathrm{j}}],$$

[1] To see this, note that $\boldsymbol{\theta}^*$ and $b(\boldsymbol{x})$ can be replaced by arbitrary values $\tilde{\boldsymbol{\theta}}^*$ and $\tilde{b}(\boldsymbol{x}) = b(\boldsymbol{x}) + (y_{\tilde{\boldsymbol{\theta}}^*}(\boldsymbol{x}) - y_{\boldsymbol{\theta}^*}(\boldsymbol{x}))$ without changing the sampling distribution of the $\boldsymbol{z}_i$'s.

where $m(\boldsymbol{Z}|\mathcal{M}_\mathrm{j})$ is the *marginal likelihood*, or *evidence*, for model $\mathcal{M}_j$, obtained by integrating the likelihood $\mathcal{L}_j(\boldsymbol{Z}|\boldsymbol{p}_\mathrm{j})$ over the prior distribution of model parameters $\boldsymbol{p}_j$:

$$m(\boldsymbol{Z}|\mathcal{M}_\mathrm{j}) \quad = \quad \int_{\boldsymbol{p}_j} \mathcal{L}_j(\mathbf{Z}|\boldsymbol{p}_j)\pi_j(\boldsymbol{p}_j)d\boldsymbol{p}_j.$$

$\mathbb{P}[\mathcal{M}_j]$ can be interpreted as the prior probability of model $\mathcal{M}_j$, which we will take equal to $1/2$, meaning that we consider both models equally probable *a priori*. This also means that we believe one of them to be the true generative model of the data.

The test then consists to compute the *Bayes factor*, defined as the ratio of marginal likelihoods of both models, which boils down here to the ratio of their posterior probabilities, also know as the *posterior odds*:

$$\begin{aligned} B_{0,1}(\mathbf{Z}) \quad &:= \quad \frac{m(\boldsymbol{Z}|\mathcal{M}_0)}{m(\boldsymbol{Z}|\mathcal{M}_1)} \left(= \frac{\mathbb{P}[\mathbf{Z}|\mathcal{M}_0]}{\mathbb{P}[\mathbf{Z}|\mathcal{M}_1]} \right) \\ &= \quad \frac{\int_{\boldsymbol{p}_j} \mathcal{L}_0(\mathbf{Z}|\boldsymbol{p}_0)\pi_0(\boldsymbol{p}_0)d\boldsymbol{p}_0}{\int_{\boldsymbol{p}_1} \mathcal{L}_1(\mathbf{Z}|\boldsymbol{p}_1)\pi_1(\boldsymbol{p}_1)d\boldsymbol{p}_1}. \end{aligned}$$

It can also be seen as a Bayesian version of the maximum likelihood ratio test in [Loeppky et al., 2006], wherein each model likelihood $\mathcal{L}_j(\mathbf{Z}|\boldsymbol{p}_j)$ is averaged rather than maximised with respect to $\boldsymbol{p}_j$. A Bayes factor greater than 1 means that the absence of a model bias is *a posteriori* more probable than its presence. Furthermore, one can also avoid selecting a single model, for instance if the Bayes factor is too close to 1. In this case, posterior mean predictions of the interest quantities under each model can be added, weighted by the corresponding probabilities $\mathbb{P}[\mathbf{Z}|\mathcal{M}_j]$, effectively integrating out model uncertainty [Hoeting et al., 1999].

From a more practical point of view, the main difficulty consists in evaluating the integrals defining the marginal likelihoods $m(\mathbf{Z}|\mathcal{M}_j)$. In general, these have no closed-form, except in very special cases, for instance when the code is linear with respect to its parameters, and in the absence of model bias; this is why in [Damblin, 2015] we have restricted ourselves to such linear codes.

the choice of prior laws raises other issues, due to their decisive influence on the model selection process, and to the impossibility to use minimally informative reference priors, as suggested in [Berger et al., 2001], when these are improper. The intrinsinc Bayes factor, as introduced in [Berger and Pericchi, 1996], was adopted to circumvent this second problem.

2.3 Results

The approach described above was validated through intensive numerical experiments. In particular, we have shown that our Bayesian test identifies correctly the presence or absence of a model bias, simulated as the realization of a Gaussian process, when the correlation length which controls the regularity of its

paths is moderate. On the other hand, when the correlation length is comparable or exceeds the length of the observation interval, we have observed that the model bias tends to become indistinguishable from the low-order polynomial used in our simulations to represent the code. We stress that this limit is not inherent to our approach, but rather serves to illustrate the always present risk of confounding a computer model and a very regular bias.

3 Post-doctoral work program

3.1 General goal

After these first encouraging results, many questions remain:

- How to implement the Bayesian test of of the presence of a mode bias for a costly, nonlinear compute code?
- How can we perform the Bayes factor computation when it entails a high-dimensional integral with no closed-form?
- What are the numerical and/or physical designs of experiments most adapted to our model selection framework?

The main goal of this post-doc is to contribute some answers to these questions, and to implement them on several case studies, using both simulated and real-life datasets.

3.2 Suggested progression

The following steps may be followed, though the order may be changed, all tasks being strongly inter-dependent:

- Generalization to computer codes that are non-linear with respect to their calibration parameters is one of the main difficulties. Indeed, this means that likelihood integration cannot be done through either closed-form calculations or elementary quadrature formulas. Reasonable approximations will need to be found, either analytically (through Laplace approximations), or numerically, using Monte-Carlo or quasi Monte-Carlo techniques, probably combined with Monte-Carlo Markov chains (MCMC) or Importance sampling in a posterior-guided approach (see [Kass and Raftery, 1995] for a recap of marginal likelihood evaluation methods).
- The extension to costly computer codes, requiring to be approximated by cheap emulators, typically based on Gaussian processes [Sacks et al., 1989], is a very common case which will have to be addressed. The choice of an apropriate design of experiments will need to be treated then, in order to make the Bayes factor estimate as accurate as possible, given the additional uncertainty due to code emulation. One possiblity is to extend the

sequential design strategy adopted in [Damblin, 2015] for the code calibration in the bias-free model, and extend it to the biased model setting. Another possiblity would be to consider the stepwise uncertainty reduction (SUR) approach, as in [Chevalier et al., 2014], based on an iterative reduction of the Bayes factor's expected variance. Finally, for codes with high-dimensional outputs, a pre-processing dimension reduction step will potentially be needed, as suggested in [Higdon et al., 2008].

- The work on prior elicitation needs also to be extended. To begin with, the extension to nonlinear codes may question the validity of the current reference priors on the code parameters, as well as on the hyperparameters of the Gaussian process used to represent the model bias term. Indeed, these priors are well adapted to linear codes, but their behaviour in a non-linear setting needs to be investigated. It is worth noting that a similar work on prior elicitation is being conducted in the context of a PhD in progress in the MRI department, concerning non Gaussian kriging emulation. Another open question concerns the reference prior for the ratio k between the variance λ_1^2 of measurement errors and that σ^2 of the Gaussian process describing the model bias. Indeed, the current Beta prior is an *ad-hoc* choice with no formal justification, though it has been shown to give good results in practice.

3.3 Case studies

Several applications are envisionned for the time being, concerning several industrial computer codes developed by EDF:

- A computational fluid dynamics model, developed using Code_Saturne, is used for the numerical simulation of a welding process, in view of optimizing the settings of the welding operation. The calibration, as well the validation of this model, are made difficult by its high computational cost (several hours per run on multiple cores)

- A hydraulic model, developed using TELEMAC 2D, is used to assess the reliability of industrial installations located next to water bodies, such as rivers or lakes. Its calibration using sequential planing strategies is currently being studied, and can be considered as a preliminary step to its validation.

- Photovoltaic panel models, developed using the systems simulation langage MODELICA, is currently being developed for a PhD thesis on the long-term prediction of power production for photovoltaic plants. Validating such models is crucial in order to guarantee the accuracy of such predictions.

4 Required skills

The questions raised by this subject need to be addressed both from theoretical and operational perspectives. Hence the candidate will have to manipulate complex mathematical tools and methods, while always keeping in mind the expected results on real-life applications.

For these reasons, the post-doctoral fellow will need to possess the following human and technical abilities:

- A good knowledge of probability and statistical theory, and in particular a very firm background in Bayesian approaches. Experience in the field of high-dimensional numerical integration would be an appreciable asset;
- Ease in numerical analysis, computational statistics and programming, with a good knowledge of both R and Python scientific programing environments;
- Curiosity, independence, and a taste for working in multidisciplinary teams;
- Good oral and written communication skills, with a capability to understand and explain with clarity a problem and the methods deployed in order to solve it;

5 Practical information and contacts

- **Advisors:** Merlin KELLER and Nicolas BOUSQUET (EDF R&D), Guillaume DAMBLIN (CEA), Pierre BARBILLON and Eric PARENT (AgroParisTech)
- **Location:** EDF R&D, MRI department, 6, quai Watier, CHATOU and AgroParisTech, 16, rue Claude Bernard, PARIS, FRANCE
- **Duration:** one year, starting October 2016
- **Contacts :** merlin.keller@edf.fr, nicolas.bousquet@edf.fr, guillaume.damblin@cea.fr, pierre.barbillon@agroparistech.fr, eric.parent.agroparistech@gmail.com

References

[Bayarri et al., 2007] Bayarri, M. J., Berger, J. O., Sacks, P. R., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007). A framework for validation of computer models. *Technometrics*, 49(2):138–154.

[Berger et al., 2001] Berger, J., De Oliveira, V., and Sanso, B. (2001). Objective Bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, 96(456):1361–1374.

[Berger and Pericchi, 1996] Berger, J. and Pericchi, L. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122.

[Chevalier et al., 2014] Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., and Richet, Y. (2014). Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):445–465.

[Damblin, 2015] Damblin, F. (2015). *Contributions statistiques au calage et à la validation des codes de calcul*. PhD thesis, Université Paris Saclay.

[Higdon et al., 2008] Higdon, D., Gattiker, J., Williams, B., and Brightley, M. (2008). Computer model calibration using high-dimensional outputs. *Journal of the American Statistical Association*, 103(482):570–583.

[Hoeting et al., 1999] Hoeting, J., Madigan, D., Raftery, A., and Volinsky, C. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417.

[Kass and Raftery, 1995] Kass, R. and Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

[Kennedy and O'Hagan, 2000] Kennedy, M. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.

[Loeppky et al., 2006] Loeppky, D., Bingham, D., and Welch, W. (2006). Computer model calibration or tuning in practice. Technical report.

[Sacks et al., 1989] Sacks, J., W.J., W., Mitchell, T., and Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423.