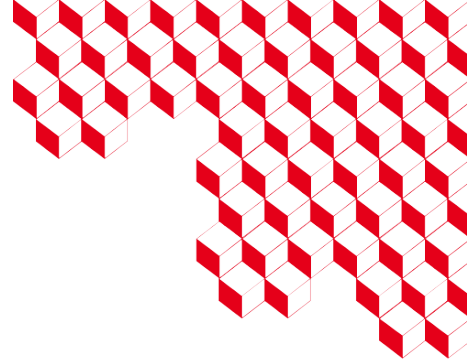




Non-additive discrepancies for Bayesian model calibration with Functional Outputs

| 01/10/2026

Paul Castéras^{1,2,3}, Josselin Garnier¹, Julien Bect² & Gwenaél Salin³

¹ CMAP, CNRS, École polytechnique, Institut polytechnique de Paris, 91120 Palaiseau, France

² Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France

³ CEA, DAM, DIF, F-91297, Arpajon, France, paul.casteras@cea.fr

Table of contents

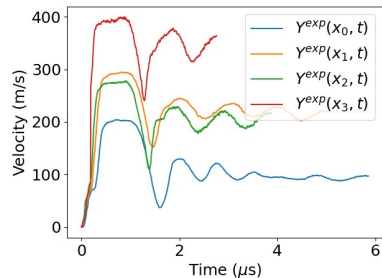
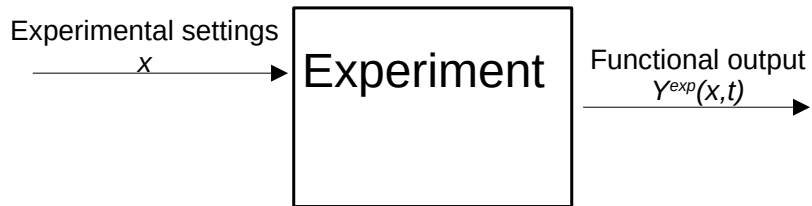
- 1 Bayesian model calibration
- 2 Phase discrepancies
- 3 Application: calibration of an equation of state
- 4 Application: calibration of the Johnson's damage model
- 5 Conclusion



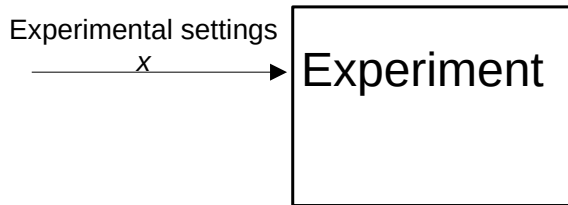
1.

BAYESIAN MODEL CALIBRATION

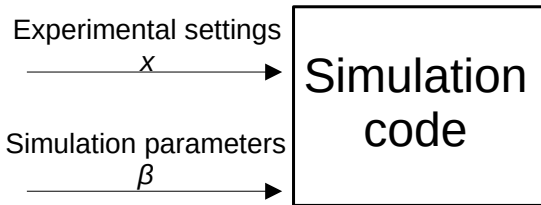
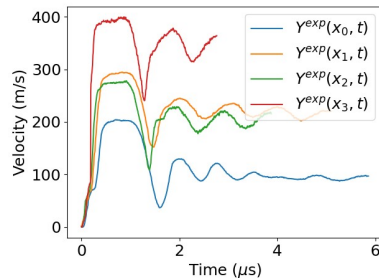
Calibration of functional outputs



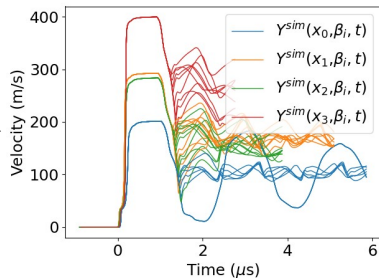
Calibration of functional outputs



Functional output $\gamma^{exp}(X, t)$ →



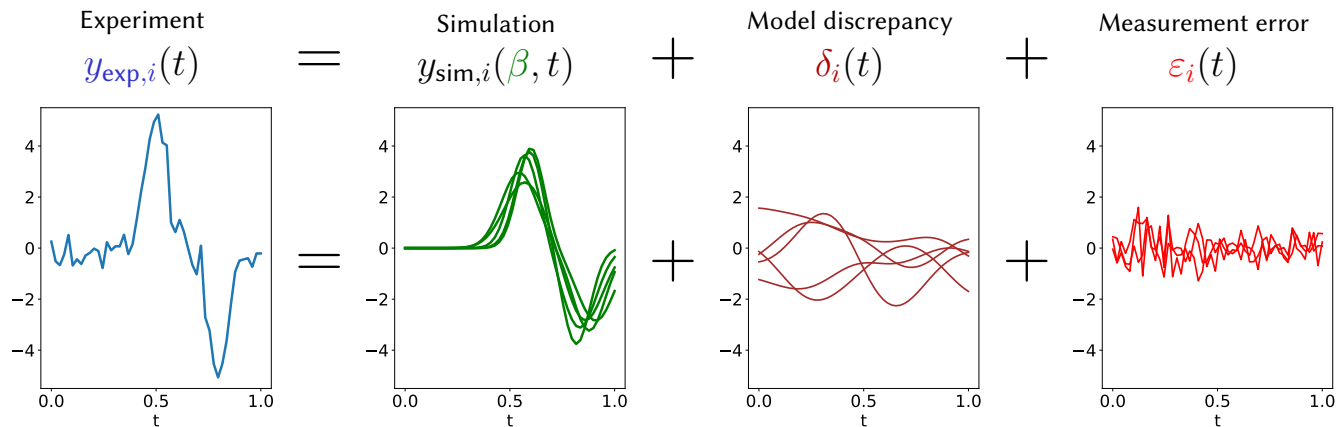
Functional output $\gamma^{sim}(X, \beta, t)$ →



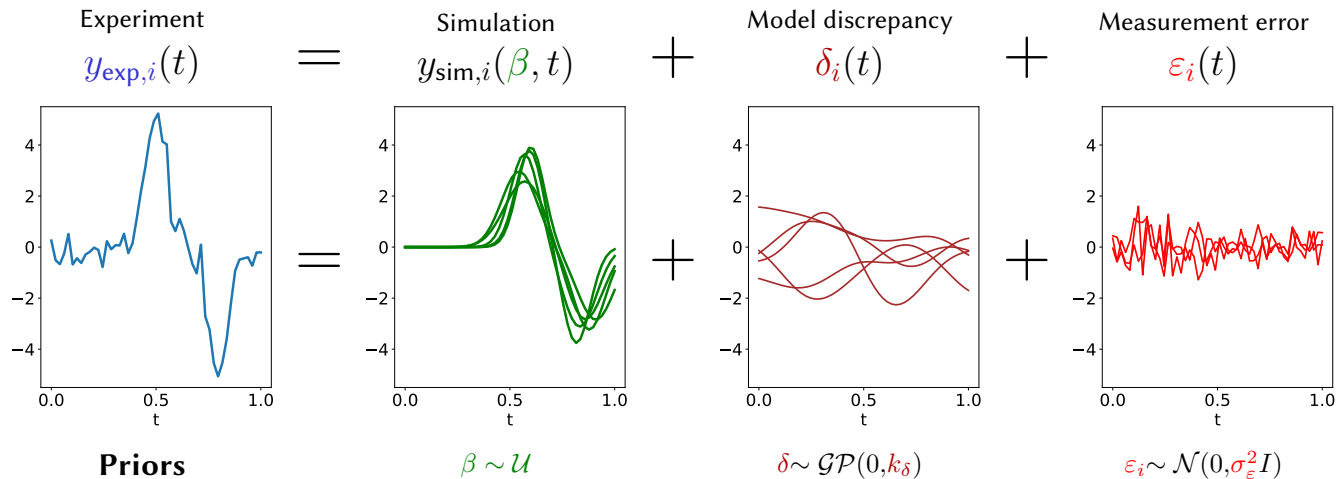
How to estimate/calibrate the parameter β ?

✓, X

Relation between experiment and simulation [Kennedy and O'Hagan, 2001]



Relation between experiment and simulation [Kennedy and O'Hagan, 2001]



Bayesian calibration with functional outputs

Bayesian calibration: [Kennedy and O'Hagan, 2001, Higdon et al., 2008]

We apply Bayes' rule:

$$p(\beta \mid (y_{\text{exp},i})_i) \propto \prod_i p(y_{\text{exp},i} \mid \beta) \pi(\beta)$$

Bayesian calibration with functional outputs

Bayesian calibration: [Kennedy and O'Hagan, 2001, Higdon et al., 2008]

We apply Bayes' rule:

$$p(\beta \mid (y_{\text{exp},i})_i) \propto \prod_i p(y_{\text{exp},i} \mid \beta) \pi(\beta)$$

- **the likelihood:** $y_{\text{exp},i} \mid \beta \sim \mathcal{N}(y_{\text{sim},i}(\beta), \Sigma_\delta + \sigma_\varepsilon^2 I)$.
- **the prior:** $\pi(\beta)$ is chosen based on physical knowledge.
- **the posterior:** $p(\beta \mid (y_{\text{exp},i}))$ provides an estimate of β with quantified uncertainty.

Metamodels: GP with PCA

When simulations are expensive, we run the code N_{sim} times and build a metamodel to replace it. [Perrin, 2020]

Inputs $(\beta_j)_{j=1,\dots,N_{\text{sim}}}$, **Outputs** $(y_{\text{sim},i}(\beta_j, t_k))_{j=1,\dots,N_{\text{sim}}, k=1,\dots,N_t}$

Metamodels: GP with PCA

When simulations are expensive, we run the code N_{sim} times and build a metamodel to replace it. [Perrin, 2020]

Inputs $(\beta_j)_{j=1,\dots,N_{\text{sim}}}$, **Outputs** $(y_{\text{sim},i}(\beta_j, t_k))_{j=1,\dots,N_{\text{sim}}, k=1,\dots,N_t}$

Dimension reduction (PCA) $y_Q^{\text{sim},i} = D_{Q,i}^T y_{\text{sim},i} \in \mathbb{R}^Q, \quad Q \ll N_t$

Metamodels: GP with PCA

When simulations are expensive, we run the code N_{sim} times and build a metamodel to replace it. [Perrin, 2020]

Inputs $(\beta_j)_{j=1,\dots,N_{\text{sim}}}$, **Outputs** $(y_{\text{sim},i}(\beta_j, t_k))_{j=1,\dots,N_{\text{sim}}, k=1,\dots,N_t}$

Dimension reduction (PCA) $y_Q^{\text{sim},i} = D_{Q,i}^T y_{\text{sim},i} \in \mathbb{R}^Q, \quad Q \ll N_t$

Gaussian process regression $\tilde{y}_{Q,l}^{\text{sim},i}(\beta) \mid (\beta_j, y_{Q,l}^{\text{sim},i}(\beta_j))_j \sim \mathcal{GP}(\mu_{Q,i}^{\text{post}}, \sigma_{Q,i}^{\text{post}})$

Metamodels: GP with PCA

When simulations are expensive, we run the code N_{sim} times and build a metamodel to replace it. [Perrin, 2020]

Inputs $(\beta_j)_{j=1,\dots,N_{\text{sim}}}$, **Outputs** $(y_{\text{sim},i}(\beta_j, t_k))_{j=1,\dots,N_{\text{sim}}, k=1,\dots,N_t}$

Dimension reduction (PCA) $y_Q^{\text{sim},i} = D_{Q,i}^T y_{\text{sim},i} \in \mathbb{R}^Q$, $Q \ll N_t$

Gaussian process regression $\tilde{y}_{Q,i}^{\text{sim},i}(\beta) \mid (\beta_j, y_{Q,i}^{\text{sim},i}(\beta_j))_j \sim \mathcal{GP}(\mu_{Q,i}^{\text{post}}, \sigma_{Q,i}^{\text{post}})$

Output reconstruction $\tilde{y}_{\text{sim},i}(\beta) \sim \mathcal{GP}(\mu_{\text{meta},i} = D_{Q,i} \mu_{Q,i}^{\text{post}}, \Sigma_{\text{meta},i} = D_{Q,i}^T \text{diag}(\sigma_{Q,i}^{\text{post}}) D_{Q,i})$

Metamodels: GP with PCA

When simulations are expensive, we run the code N_{sim} times and build a metamodel to replace it. [Perrin, 2020]

Inputs $(\beta_j)_{j=1,\dots,N_{\text{sim}}}$, **Outputs** $(y_{\text{sim},i}(\beta_j, t_k))_{j=1,\dots,N_{\text{sim}}, k=1,\dots,N_t}$

Dimension reduction (PCA) $y_Q^{\text{sim},i} = D_{Q,i}^T y_{\text{sim},i} \in \mathbb{R}^Q$, $Q \ll N_t$

Gaussian process regression $\tilde{y}_{Q,i}^{\text{sim},i}(\beta) \mid (\beta_j, y_{Q,i}^{\text{sim},i}(\beta_j))_j \sim \mathcal{GP}(\mu_{Q,i}^{\text{post}}, \sigma_{Q,i}^{\text{post}})$

Output reconstruction $\tilde{y}_{\text{sim},i}(\beta) \sim \mathcal{GP}(\mu_{\text{meta},i} = D_{Q,i} \mu_{Q,i}^{\text{post}}, \Sigma_{\text{meta},i} = D_{Q,i}^T \text{diag}(\sigma_{Q,i}^{\text{post}}) D_{Q,i})$

Likelihood with the metamodel

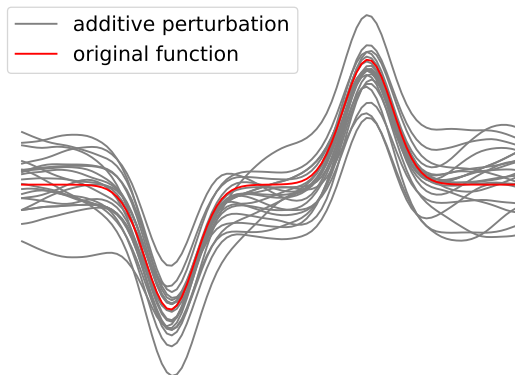
$$y_{\text{exp},i} \mid \beta \sim \mathcal{N}(\mu_{\text{meta},i}(\beta), \Sigma_{\delta} + \sigma_{\epsilon}^2 I + \Sigma_{\text{meta},i}(\beta))$$



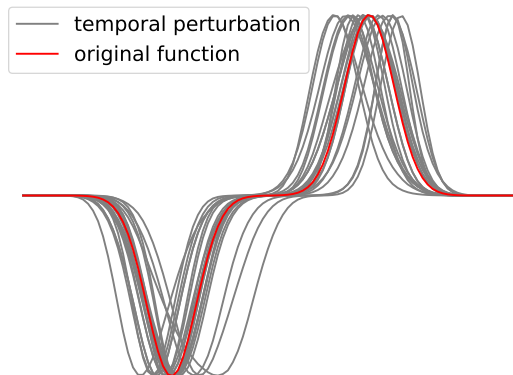
2. **PHASE DISCREPANCIES**

Additive discrepancy vs. phase discrepancy

$f(t) + \delta(t)$ for different δ



$f(\gamma(t))$ for different γ



Priors on the phase discrepancies

Phase space [Srivastava et al., 2011]

$$\Gamma = \left\{ \gamma : [0, 1] \rightarrow [0, 1] \mid \begin{array}{l} \gamma(0) = 0, \quad \gamma(1) = 1, \\ \gamma, \gamma^{-1} \text{ increasing and differentiable} \end{array} \right\}$$

Priors on the phase discrepancies

Phase space [[Srivastava et al., 2011](#)]

$$\Gamma = \left\{ \gamma : [0, 1] \rightarrow [0, 1] \mid \begin{array}{l} \gamma(0) = 0, \quad \gamma(1) = 1, \\ \gamma, \gamma^{-1} \text{ increasing and differentiable} \end{array} \right\}$$

Transformation to the shooting vector space [[Srivastava and Klassen, 2016](#)]

$$v = \exp_{id}^{-1}(\sqrt{\gamma'}) \in V = \left\{ v \in L^2 \mid \int_0^1 v(t) dt = 0 \right\}$$

A prior in the shooting vector space [[Kurtek, 2017](#), [Lu et al., 2017](#), [Tucker et al., 2021](#)]

$$v \sim \mathcal{GP}(0, k_v) \text{ where } k_v \text{ is a Matérn } 5/2 \text{ kernel conditioned to } \int v = 0$$

Priors on the phase discrepancies

Phase space [Srivastava et al., 2011]

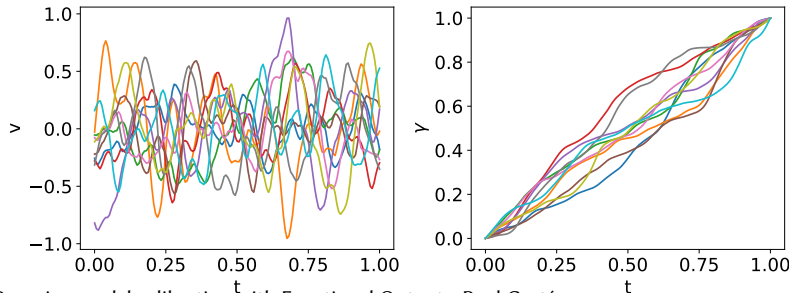
$$\Gamma = \left\{ \gamma : [0, 1] \rightarrow [0, 1] \mid \begin{array}{l} \gamma(0) = 0, \quad \gamma(1) = 1, \\ \gamma, \gamma^{-1} \text{ increasing and differentiable} \end{array} \right\}$$

Transformation to the shooting vector space [Srivastava and Klassen, 2016]

$$v = \exp_{id}^{-1}(\sqrt{\gamma'}) \in V = \left\{ v \in L^2 \mid \int_0^1 v(t) dt = 0 \right\}$$

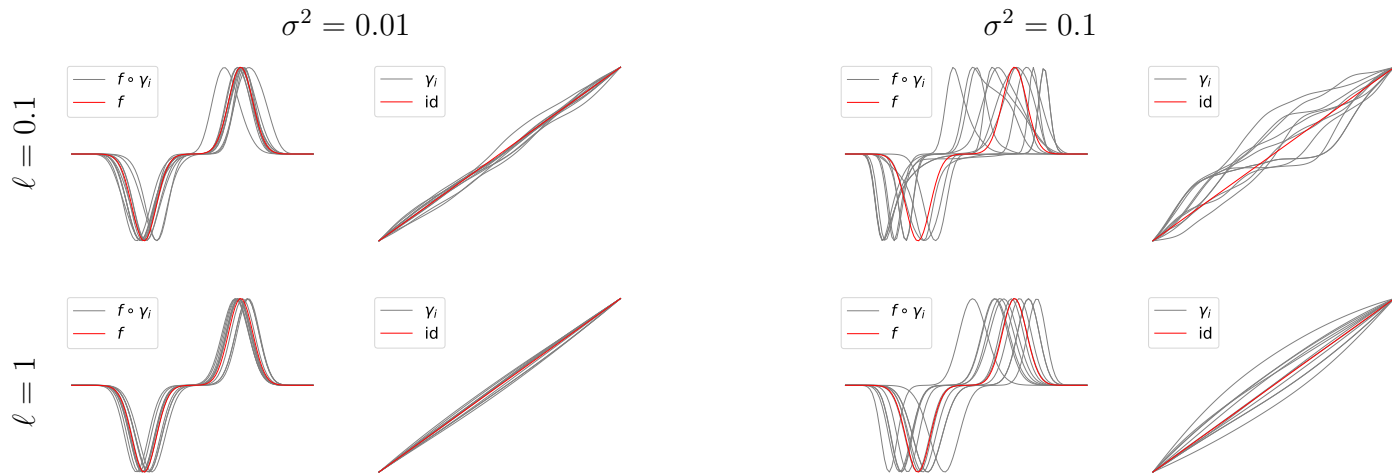
A prior in the shooting vector space [Kurtek, 2017, Lu et al., 2017, Tucker et al., 2021]

$v \sim \mathcal{GP}(0, k_v)$ where k_v is a Matérn 5/2 kernel conditioned to $\int v = 0$



Hyperparameters for the phase prior

$v \sim \mathcal{GP}(0, k_v)$ where k_v dépend on the lengthscale l and the variance σ^2 .



Bayesian model with a phase discrepancy

$$y_{\text{exp},i}(\mathbf{t}) = y_{\text{sim},i}(\beta, \gamma_i(\mathbf{t})) + \delta_i(\mathbf{t}) + \varepsilon_i(\mathbf{t})$$

$$v_i = \exp_{id}^{-1}(\gamma_i) \sim \mathcal{GP}(0, k_v)$$

$$\delta_i \sim \mathcal{GP}(0, k_\delta)$$

$$\varepsilon_i(\mathbf{t}) \sim \mathcal{N}(0, \sigma_\varepsilon^2 I)$$

$$\beta \sim \mathcal{U}$$

We then want to sample from the posterior

$$P(\beta, (v_i)_i \mid (y_{\text{exp},i})) \propto \prod_i \mathcal{N}(y_{\text{exp},i}(\mathbf{t}); y_{\text{sim},i}(\beta, \gamma_i(\mathbf{t})), \sigma_\varepsilon^2 I + \Sigma_\delta) \pi(v_i) \pi(\beta)$$

In practice, we sample the coefficients $(c_j)_j$ of the Karhunen-Loève decomposition: $v(t) = \sum_j c_j b_j(t)$

Bayesian model with a phase discrepancy

$$y_{\text{exp},i}(\mathbf{t}) = y_{\text{sim},i}(\beta, \gamma_i(\mathbf{t})) + \delta_i(\mathbf{t}) + \varepsilon_i(\mathbf{t})$$

$$v_i = \exp_{id}^{-1}(\gamma_i) \sim \mathcal{GP}(0, k_v)$$

$$\delta_i \sim \mathcal{GP}(0, k_\delta)$$

$$\varepsilon_i(\mathbf{t}) \sim \mathcal{N}(0, \sigma_\varepsilon^2 I)$$

$$\beta \sim \mathcal{U}$$

We then want to sample from the posterior

$$P(\beta, (v_i)_i \mid (y_{\text{exp},i})) \propto \prod_i \mathcal{N}(y_{\text{exp},i}(\mathbf{t}); y_{\text{sim},i}(\beta, \gamma_i(\mathbf{t})), \sigma_\varepsilon^2 I + \Sigma_\delta) \pi(v_i) \pi(\beta)$$

In practice, we sample the coefficients $(c_j)_j$ of the Karhunen-Loève decomposition: $v(t) = \sum_j c_j b_j(t)$

Limitations:

- v_i sampled jointly with $\beta \rightarrow$ higher-dimensional sampling.
- γ_i and δ_i can explain the same differences $\rightarrow$ identifiability issues.

Elastic calibration: a tractable approximation

We consider the following approximation of the posterior: [Kahol et al., 2025]

$$p_{\text{elastic}}(\beta \mid (y_{\text{exp},i}(\mathbf{t}))_i) \propto \max_{v_1, \dots, v_m \in V} p(\beta, (v_i)_i \mid (y_{\text{exp},i}(\mathbf{t}))_i) \left(\neq \int_{(v_i)_i} p(\beta, (v_i)_i \mid (y_{\text{exp},i}(\mathbf{t}))_i) (dv_i)_i \right)$$

It may be rewritten as an alignment problem:

$$v_{\text{sim},i}(\beta, \mathbf{t}) = \arg \min_{v_i \in V} \|y_{\text{exp},i}(\mathbf{t}) - y_{\text{sim},i}(\beta, \gamma_i(\mathbf{t}))\|_{\Sigma_\delta + \sigma_\varepsilon^2 I}^2 + \|v_i\|_{\Sigma_v}^2 \quad (1)$$

Elastic calibration: a tractable approximation

We consider the following approximation of the posterior: [Kahol et al., 2025]

$$p_{\text{elastic}}(\beta \mid (y_{\text{exp},i}(\mathbf{t}))_i) \propto \max_{v_1, \dots, v_m \in V} p(\beta, (v_i)_i \mid (y_{\text{exp},i}(\mathbf{t}))_i) \left(\neq \int_{(v_i)_i} p(\beta, (v_i)_i \mid (y_{\text{exp},i}(\mathbf{t}))_i) (dv_i)_i \right)$$

It may be rewritten as an alignment problem:

$$v_{\text{sim},i}(\beta, \mathbf{t}) = \arg \min_{v_i \in V} \|y_{\text{exp},i}(\mathbf{t}) - y_{\text{sim},i}(\beta, \gamma_i(\mathbf{t}))\|_{\Sigma_\delta + \sigma_\varepsilon^2 I}^2 + \|v_i\|_{\Sigma_v}^2 \quad (1)$$

We can rewrite p_{elastic} with the prior on β and two independent likelihoods to get **elastic calibration** relations [Francom et al., 2025]:

$$\begin{aligned} \text{(Amplitude)} \quad z_{\text{exp},i}(\mathbf{t}) &= z_{\text{sim},i}(\beta, \mathbf{t}) + \delta_i(\mathbf{t}) + \varepsilon_i(\mathbf{t}), \\ \text{(Shooting vector)} \quad v_{\text{exp},i}(\mathbf{t}) &= v_{\text{sim},i}(\beta, \mathbf{t}) + \delta_i^v(\mathbf{t}) \end{aligned}$$

where $v_{\text{exp},i}(\mathbf{t}) = \mathbf{0}$ is the shooting vector of the experiment and $z_{\text{sim},i}(\beta, \mathbf{t}) = y_{\text{sim},i}(\beta, \gamma_i(\mathbf{t}))$, $z_{\text{exp},i}(\mathbf{t}) = y_{\text{exp},i}(\mathbf{t})$ are the amplitudes.

Extensions: metamodels and partial elastic calibration

Metamodels

To build metamodels, we first solve (1) for a set of simulations $(y_{\text{sim},i}(\beta_j))_j$ and then fit two metamodels for each experiment: the metamodel of the amplitude $\tilde{z}_{\text{sim},i}$ and of the shooting vector $\tilde{v}_{\text{sim},i}$.

$$\begin{aligned} \text{(Amplitude)} \quad z_{\text{exp},i}(\mathbf{t}) &= \tilde{z}_{\text{sim},i}(\beta, \mathbf{t}) + \delta_i(\mathbf{t}) + \varepsilon_i(\mathbf{t}), \\ \text{(Shooting vector)} \quad v_{\text{exp},i}(\mathbf{t}) &= \tilde{v}_{\text{sim},i}(\beta, \mathbf{t}) + \delta_i^v(\mathbf{t}) \end{aligned}$$

Extensions: metamodels and partial elastic calibration

Metamodels

To build metamodels, we first solve (1) for a set of simulations $(y_{\text{sim},i}(\beta_j))_j$ and then fit two metamodels for each experiment: the metamodel of the amplitude $\tilde{z}_{\text{sim},i}$ and of the shooting vector $\tilde{v}_{\text{sim},i}$.

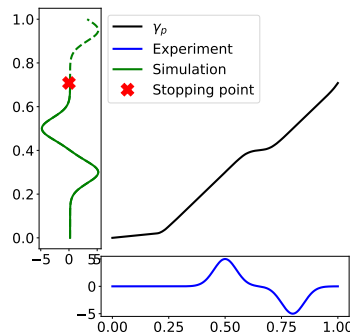
$$\begin{aligned} \text{(Amplitude)} \quad z_{\text{exp},i}(\mathbf{t}) &= \tilde{z}_{\text{sim},i}(\beta, \mathbf{t}) + \delta_i(\mathbf{t}) + \varepsilon_i(\mathbf{t}), \\ \text{(Shooting vector)} \quad v_{\text{exp},i}(\mathbf{t}) &= \tilde{v}_{\text{sim},i}(\beta, \mathbf{t}) + \delta_i^v(\mathbf{t}) \end{aligned}$$

Generalization to partial phases $\Gamma_p = \{s\gamma : \gamma \in \Gamma, s \in (0, t_f]\}$.

Prior

For $\gamma_p = s\gamma$, we consider independent priors on γ and s , where:

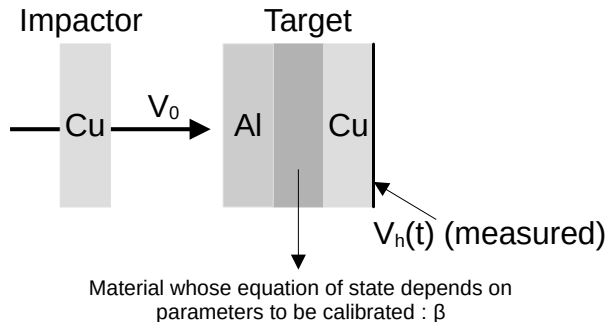
- γ is a phase with a GP prior on its shooting vector v ,
- $s \sim \mathcal{N}(1, \sigma_s^2)$ truncated to $(0, t_f]$.



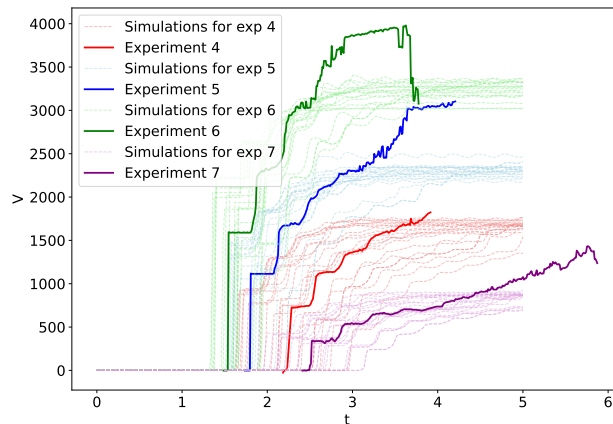


3. APPLICATION: CALIBRATION OF AN EQUATION OF STATE

Application to the calibration of an equation of state



$$\beta \in [0, 1]^4, N_{exp} = 4, N_{sim} = 100.$$



EOS: alignment step

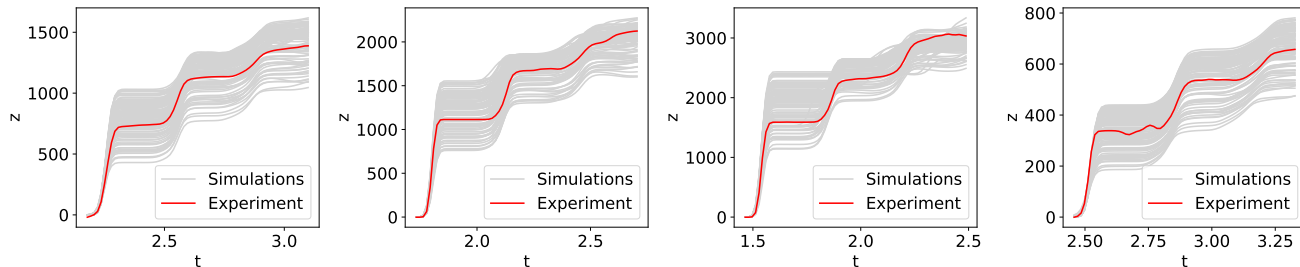


Figure: Partial amplitudes of the equation of state for the different experiments

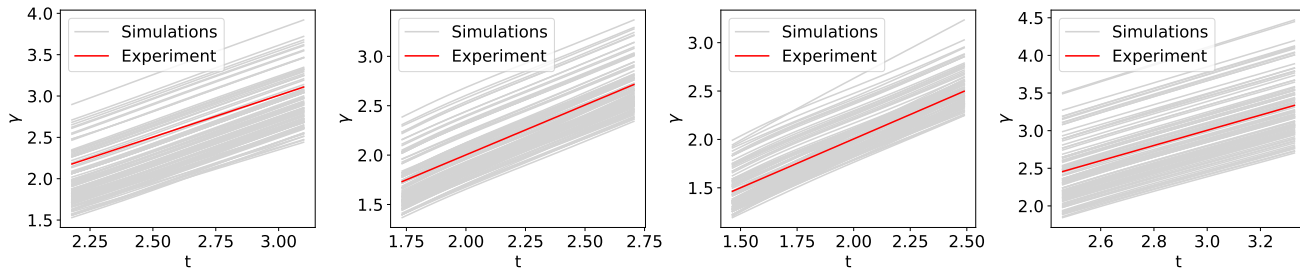


Figure: Partial phases of the equation of state for the different experiments

EOS: metamodels

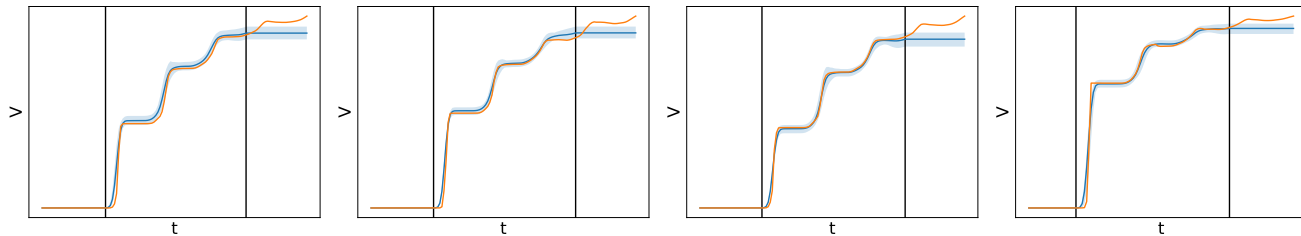


Figure: Predicted outputs for the first experiment with metamodels in the **amplitude** and **phase spaces**.

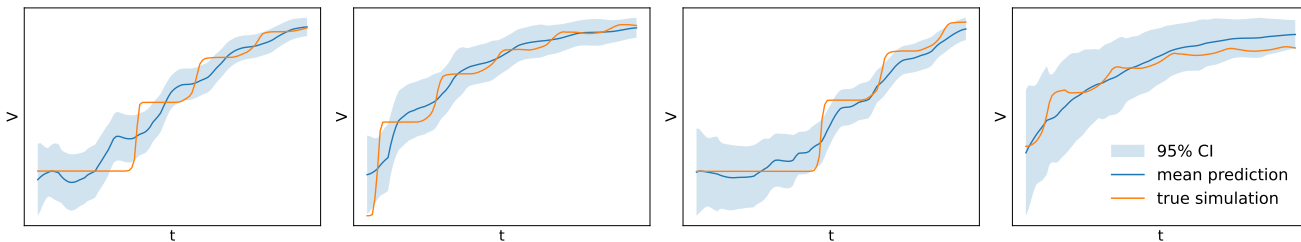
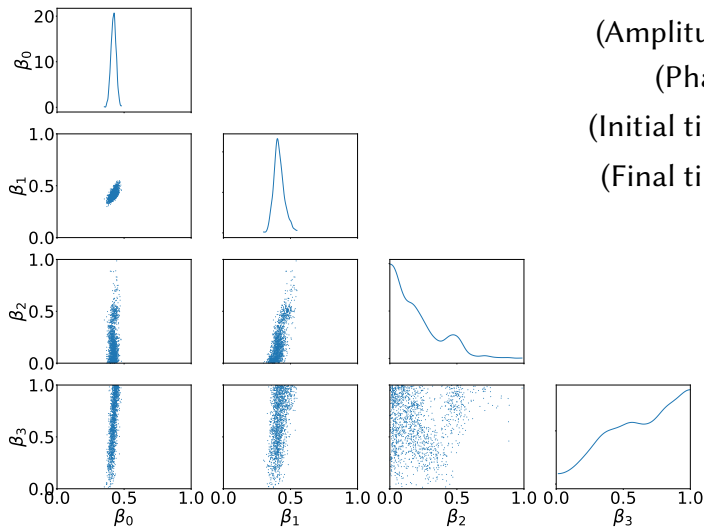


Figure: Predicted outputs for the first experiment with metamodels in the **original space**



(Amplitude) $z_{\text{exp},i}(\mathbf{t}) = \tilde{z}_{\text{sim},i}(\mathbf{t}, \beta) + \delta_{z,i}(\mathbf{t}) + \varepsilon_{z,i}(\mathbf{t})$

(Phase) $v_{\text{exp},i}(\mathbf{t}) = \tilde{v}_{\text{sim},i}(\mathbf{t}, \beta) + \delta_{v,i}(\mathbf{t})$

(Initial time) $t_{\text{exp},i}^0 = \tilde{t}_{\text{sim},i}^0(\beta) + \varepsilon_{0,i}$

(Final time) $s_{\text{exp},i} = \tilde{s}_{\text{sim},i}(\beta) + \varepsilon_{s,i}$

$$\delta_{z,i}(\mathbf{t}) \sim \mathcal{GP}(0, k_z)$$

$$\delta_{v,i}(\mathbf{t}) \sim \mathcal{GP}(0, k_v)$$

$$\varepsilon_{z,i}(\mathbf{t}) \sim \mathcal{N}(0, \sigma_{\text{mes}}^2 I)$$

$$\varepsilon_{t_0,i} \sim \mathcal{N}(0, \sigma_{t_0}^2 + \mathbf{1}_{i=0} \sigma_{\text{mod}}^{2,*})$$

$$\varepsilon_{s,i} \sim \mathcal{N}(0, \sigma_s^2)$$

$$\beta \sim \mathcal{U}([0, 1]^4)$$

Sampled with Sequential Monte Carlo

* Known synchronisation problem for the first experiment

EOS: Predictions after sampling

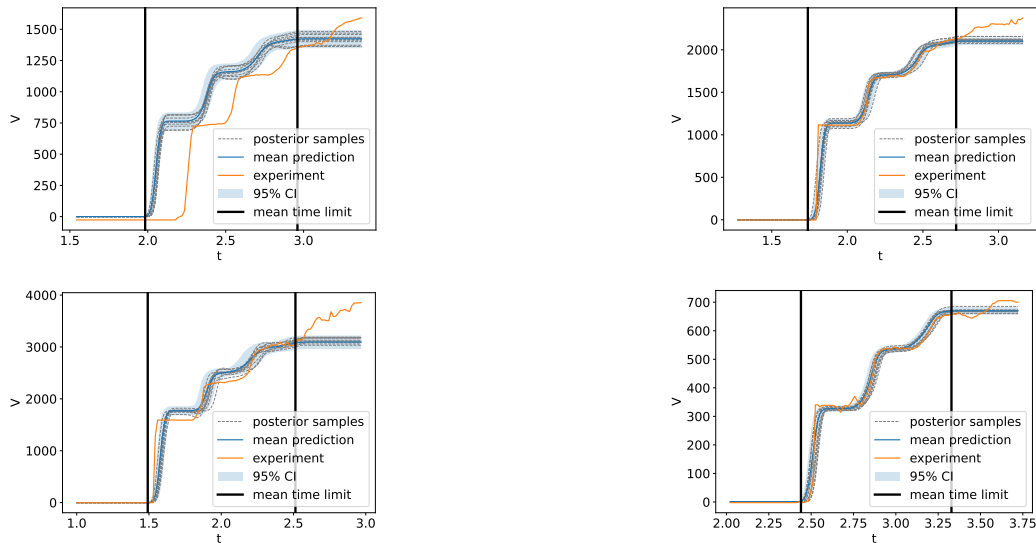


Figure: Posterior prediction after uncertainty propagation for each experiment. The blue curve corresponds to the composition of the mean predictions of the various metamodels, evaluated at the posterior mean of the calibration parameters. The credible intervals and posterior samples are obtained by composing the uncertainties associated with the metamodels and the calibration parameter.



4. APPLICATION: CALIBRATION OF THE JOHNSON'S DAMAGE MODEL

Johnson's damage model

$$\dot{\alpha} = \begin{cases} \frac{(\alpha_0-1)^{2/3}}{\nu} \alpha(\alpha-1)^{1/3} (|P| - P_{\text{eq}}) \text{sgn}(P), & \text{if } |P| > P_{\text{eq}} \\ 0, & \text{if } |P| \leq P_{\text{eq}} \end{cases}$$

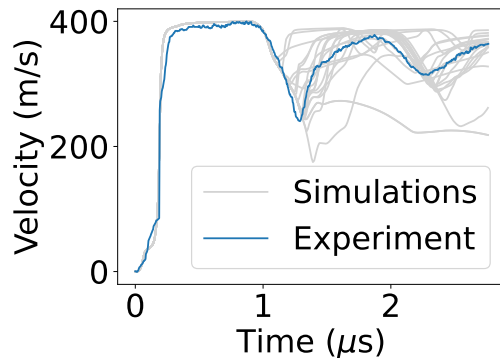
where

- $P_{\text{eq}} = \frac{a_s}{\alpha} \ln \left(\frac{\alpha}{\alpha-1} \right)$: equilibrium pressure
- a_s : effective yield strength
- ν : effective viscosity
- α_0 : initial porosity
- α : porosity

Parameters to calibrate: $\beta = (a_s, \nu, \alpha_L)$

Experimental conditions: $x = (V_0, e^p, e^c)$ where V_0 is the initial velocity and e^p, e^c are plate thicknesses.

$N_{\text{exp}} = 7, N_{\text{sim}} = 1000$



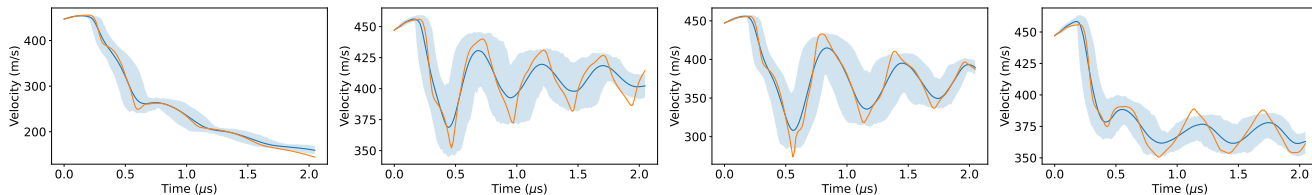


Figure: Predicted outputs for the experiment 18 with metamodels in the **amplitude** and **phase** spaces.

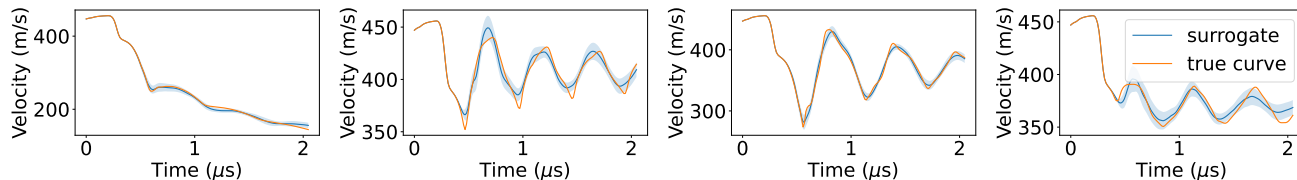
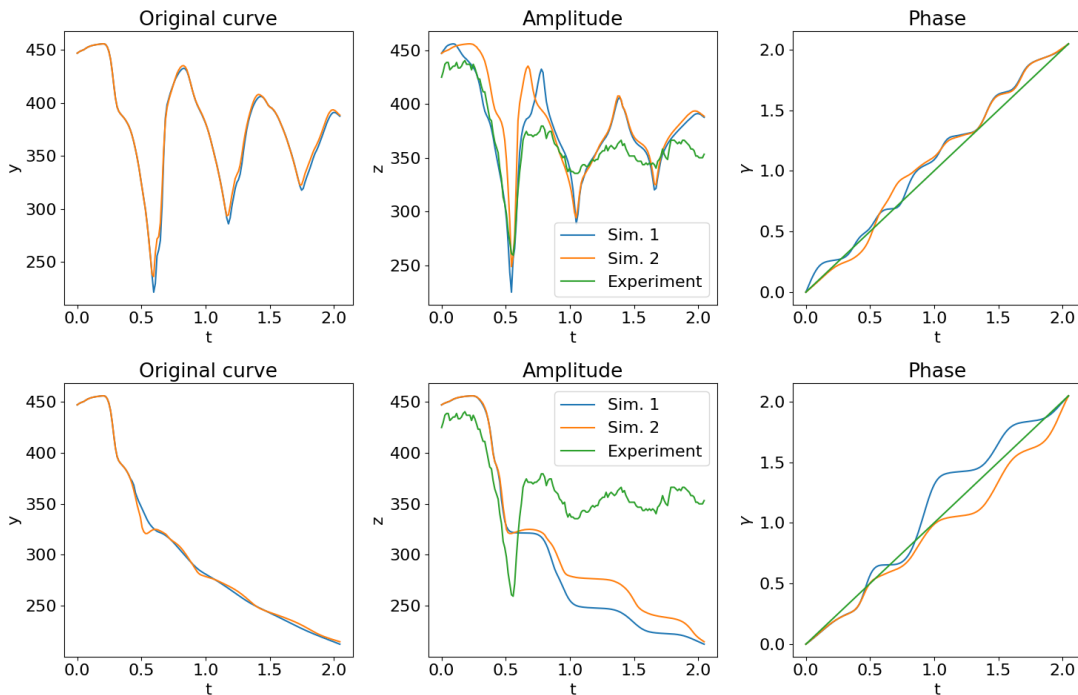
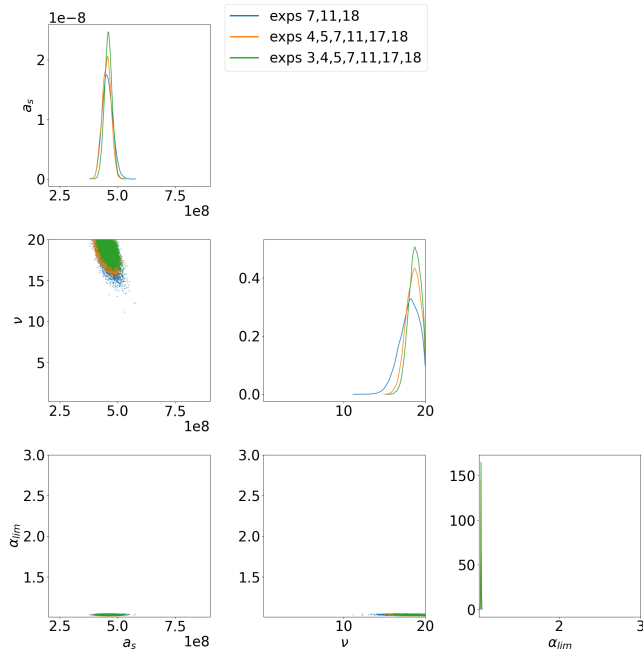


Figure: Predicted outputs for the experiment 18 with metamodels in the **original** space.

Discontinuities in the alignment transformation



Johnson: sampling



$$y_{exp,i}(t) = T_{\rho_i}(\tilde{y}_{sim,i}(\beta, \gamma_i(t))) + \delta_i(t) + \varepsilon_i(t)$$

$$T_{\rho}(y) = e^{-\rho t^2}(y - \bar{y}) + \bar{y},$$

$$\text{where } \bar{y} = \frac{\int (1 - e^{-\rho t^2})^2 y(t) dt}{\int (1 - e^{-\rho t^2})^2 dt}$$

$$\rho \sim \mathcal{E}(1/2)$$

$$\delta \sim \mathcal{GP}(0, k_{\delta})$$

$$v \sim \mathcal{GP}(0, k_v)$$

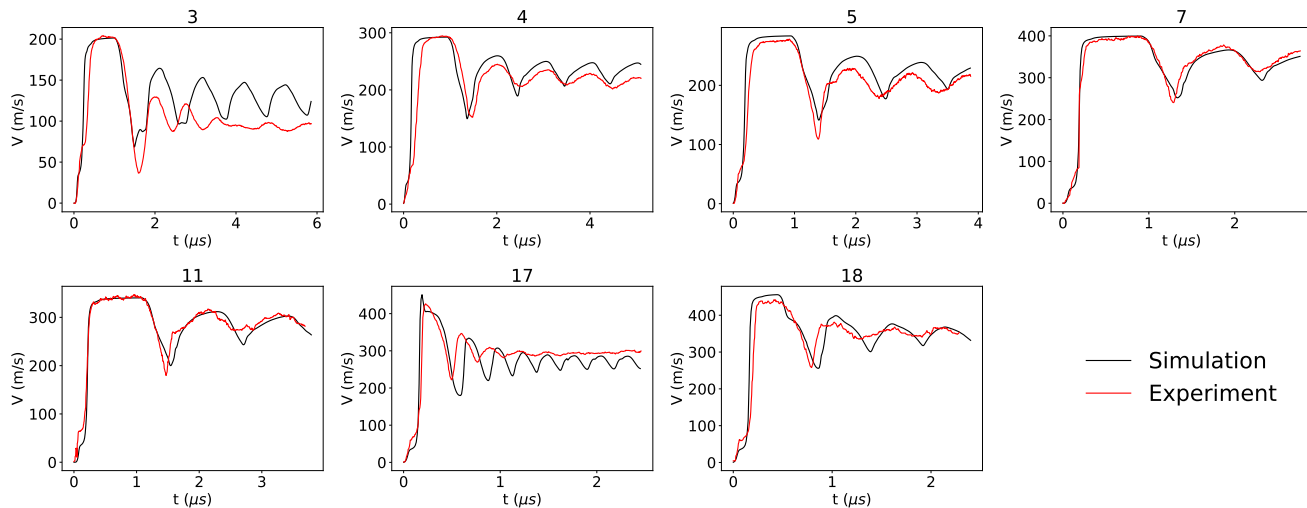
$$\varepsilon \sim \mathcal{N}(0, \sigma_{\varepsilon}^2)$$

$$\beta_0 \sim \mathcal{U}(2 \times 10^8, 9 \times 10^8),$$

$$\beta_1 \sim \mathcal{U}(0.2, 20),$$

$$\beta_2 - 1 \sim \mathcal{W}(0.8, 3).$$

Johnson: Prediction for $\beta^* = (4.2 \times 10^8, 19, 1.036)$



— Simulation
— Experiment



5. CONCLUSION

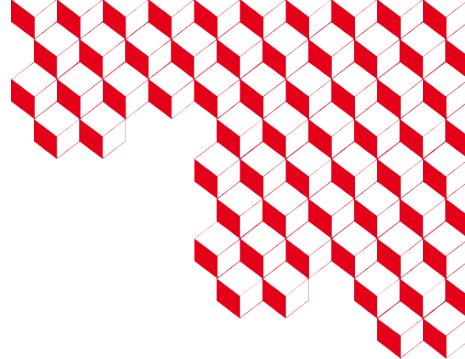
Conclusion

	Phase discrepancy	Elastic calibration
Likelihood	$y_{\text{exp}}(\mathbf{t}) = y_{\text{sim}}(\beta, \gamma(\mathbf{t})) + \delta(\mathbf{t}) + \varepsilon(\mathbf{t})$	$z_{\text{exp}}(\mathbf{t}) = z_{\text{sim}}(\beta, \mathbf{t}) + \delta(\mathbf{t}) + \varepsilon(\mathbf{t}),$ $v_{\text{exp}}(\mathbf{t}) = v_{\text{sim}}(\beta, \mathbf{t}) + \delta^v(\mathbf{t})$
Metamodels	in the original space	in the amplitude/shooting vector spaces
Better metamodels	~	✓ if continuous alignment ✗ otherwise
Always feasible (multimodal, β/v dependencies)	✓	✗
Low dimensional sampling	✗	✓
Discrepancies separation	✗	✓

Conclusion

	Phase discrepancy	Elastic calibration
Likelihood	$y_{\text{exp}}(\mathbf{t}) = y_{\text{sim}}(\beta, \gamma(\mathbf{t})) + \delta(\mathbf{t}) + \varepsilon(\mathbf{t})$	$z_{\text{exp}}(\mathbf{t}) = z_{\text{sim}}(\beta, \mathbf{t}) + \delta(\mathbf{t}) + \varepsilon(\mathbf{t}),$ $v_{\text{exp}}(\mathbf{t}) = v_{\text{sim}}(\beta, \mathbf{t}) + \delta^v(\mathbf{t})$
Metamodels	in the original space	in the amplitude/shooting vector spaces
Better metamodels	~	✓ if continuous alignment ✗ otherwise
Always feasible (multimodal, β/v dependencies)	✓	✗
Low dimensional sampling	✗	✓
Discrepancies separation	✗	✓

Possible improvements: improving the metamodels for discontinuous alignments and characterizing the feasible cases for elastic calibration.



THANK YOU

Paul Castéras
CEA DAM Île-de-France
Bruyères-le-Châtel
91297 Arpajon cedex

References I

- 
-  Abril-Pla, O., Andreani, V., Carroll, C., Dong, L., Fannesbeck, C. J., Kochurov, M., Kumar, R., Lao, J., Luhmann, C. C., Martin, O. A., et al. (2023). Pymc: a modern, and comprehensive probabilistic programming framework in python. *PeerJ Computer Science*, 9:e1516.
-  Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436.
-  Francom, D., Tucker, J. D., Huerta, G., Shuler, K., and Ries, D. (2025). Elastic bayesian model calibration. *SIAM/ASA Journal on Uncertainty Quantification*, 13(1):195–227.
-  Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583.
-  Kahol, O., Congedo, P. M., Le Maître, O., and Goy, E. D. (2025). Efficient treatment of the model error in the calibration of computer codes: the complete maximum a posteriori method. *International Journal for Uncertainty Quantification*, 15(5).
-  Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.
-  Kurtek, S. (2017). A geometric approach to pairwise bayesian alignment of functional data using importance sampling.
-  Lu, Y., Herbei, R., and Kurtek, S. (2017). Bayesian registration of functions with a gaussian process prior. *Journal of Computational and Graphical Statistics*, 26(4):894–904.

References II



Perrin, G. (2020).

Adaptive calibration of a computer code with time-series output.

Reliability engineering & system safety, 196:106728.



Srivastava, A. and Klassen, E. P. (2016).

Functional and shape data analysis, volume 1.

Springer.



Srivastava, A., Wu, W., Kurtek, S., Klassen, E., and Marron, J. S. (2011).

Registration of functional data using fisher-rao metric.

arXiv preprint arXiv:1103.3817.



Tucker, J. D., Shand, L., and Chowdhary, K. (2021).

Multimodal bayesian registration of noisy functions using hamiltonian monte carlo.

Computational Statistics & Data Analysis, 163:107298.

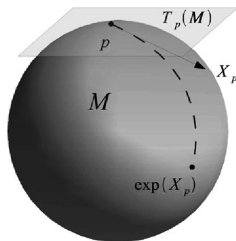
Transformation of the phase space

$$\gamma \in \Gamma = \left\{ \gamma : [0, 1] \rightarrow [0, 1] \mid \begin{array}{l} \gamma(0) = 0, \gamma(1) = 1 \\ \gamma, \gamma^{-1} \text{ increasing and differentiable} \end{array} \right\}$$

First transformation: $\psi = Q(\gamma) = \sqrt{\dot{\gamma}} \in \Psi$

$$\Rightarrow \|\psi\|_2^2 = \int_0^1 \psi(t)^2 dt = 1 \Rightarrow \Psi \subset S_\infty$$

Second transformation:



$$T_\psi(\Psi) = \left\{ v \in L^2 \mid \int_0^1 v(t) \psi(t) dt = 0 \right\}$$
$$\exp_\psi^{-1}(\psi_1) = \frac{\kappa}{\sin(\kappa)} (\psi_1 - \cos(\kappa) \psi)$$

where $\kappa = d_p(\psi, \psi_1) = \cos^{-1}(\int_0^1 \psi \psi_1)$

Figure of [Srivastava and Klassen, 2016]

EOS: metamodels

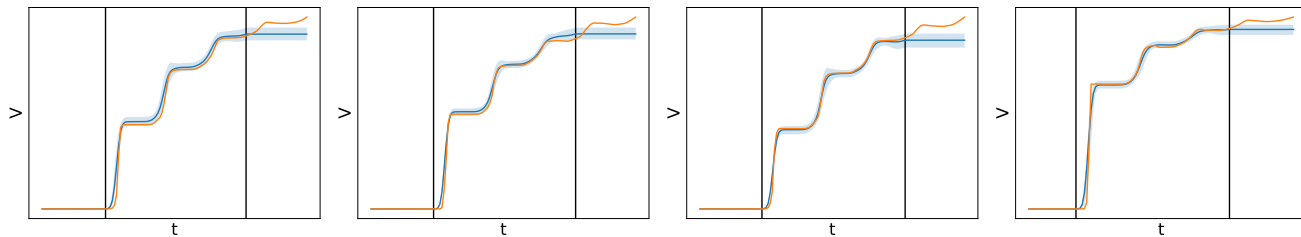


Figure: Reconstructed outputs for the first experiment of the equation of state with partial elastic metamodel.

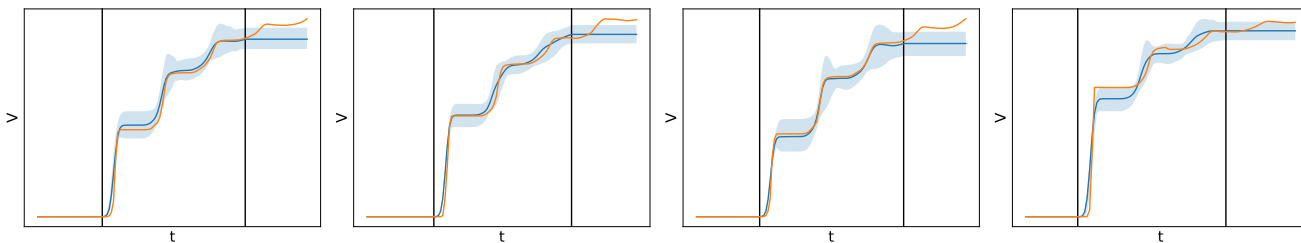


Figure: Reconstructed outputs for the first experiment of the equation of state with affine metamodel.

Anharmonic pendulum: Dataset

$$\frac{d^2\phi}{dt^2} + \alpha_1 \frac{d\phi}{dt} + \alpha_2(\phi - \alpha_0\phi^3) = 0,$$

$$y_{\text{exp}}(\mathbf{t}) = \phi^*(\mathbf{t} + 0.05 \sin(0.1\pi\mathbf{t})) + \eta(\mathbf{t}),$$

$$N_{\text{sim}} = 100$$

(2)

(3)

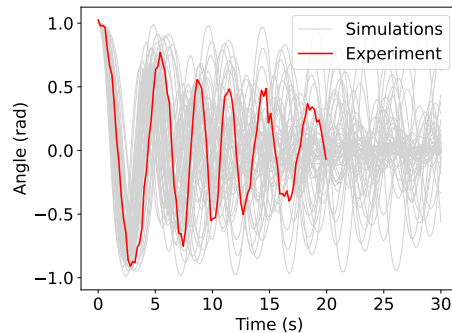
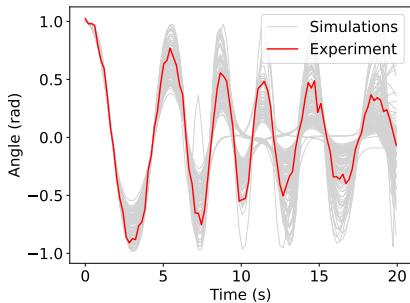


Figure: Dataset for the anharmonic pendulum.

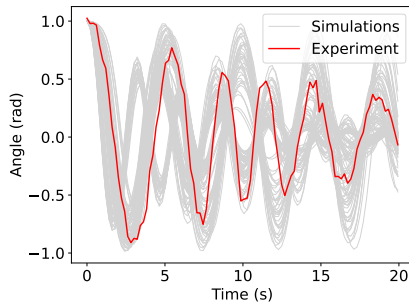
Anharmonic pendulum: Alignment



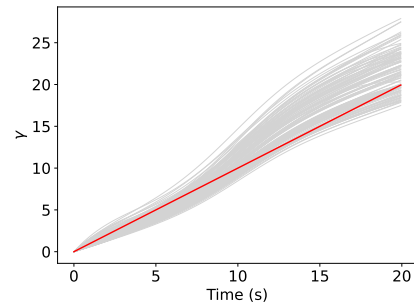
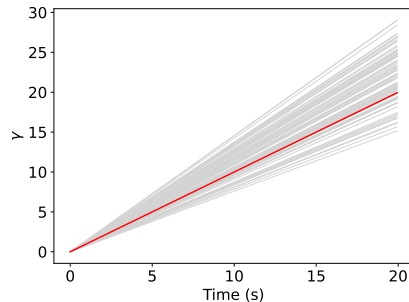
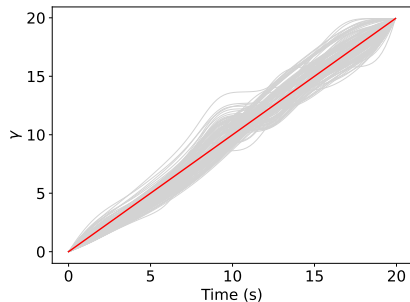
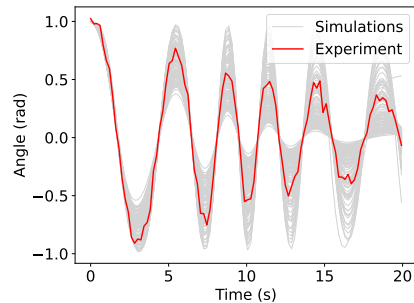
Elastic



Rescaling



Partial elastic



Anharmonic pendulum: metamodel

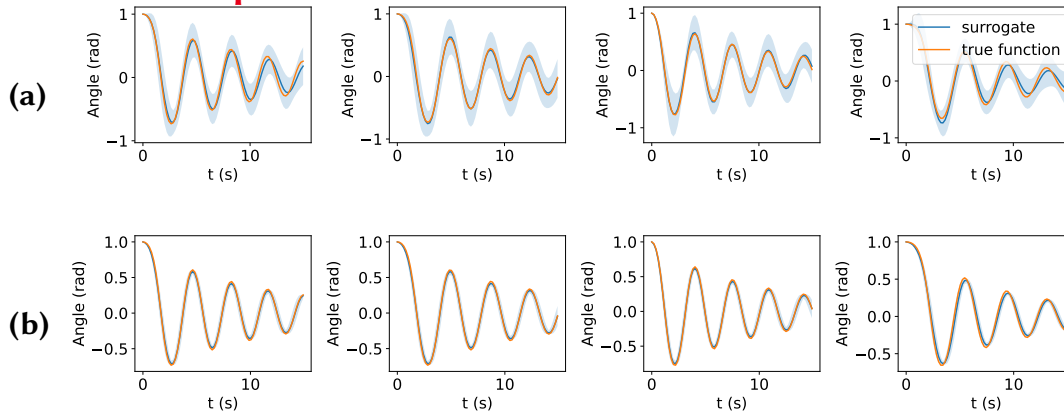


Figure: Reconstructed anharmonic pendulum outputs for four different test inputs for (a) the non-elastic metamodel and (b) the partial elastic metamodel.

	Q^2	IAE	CRPS	N_{PCA}
Non-elastic	0.78	0.10	0.06	7
Partial elastic	0.97	0.06	0.02	7 (3+3+1)

Anharmonic pendulum: Samplings

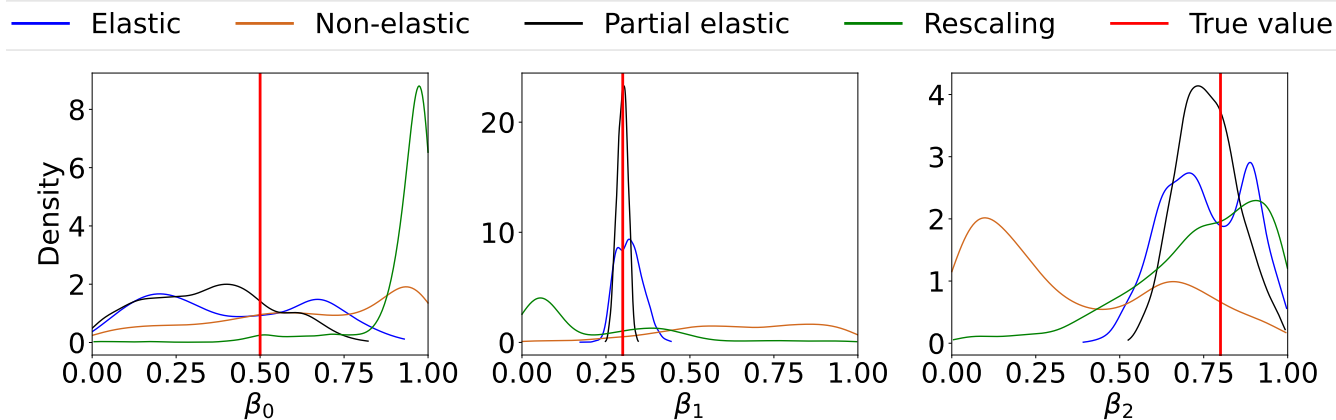


Figure: Marginals of the posterior for the different calibration methods for the anharmonic pendulum.

Anharmonic pendulum: With and without approximation

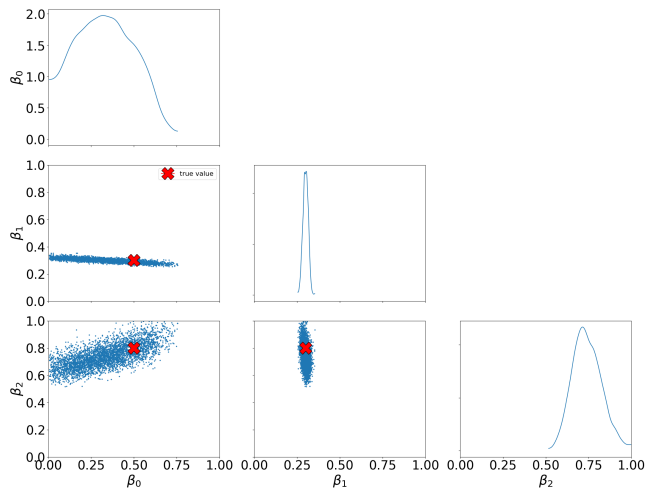


Figure: Pair plot of the posteriors with the maximum approximation.

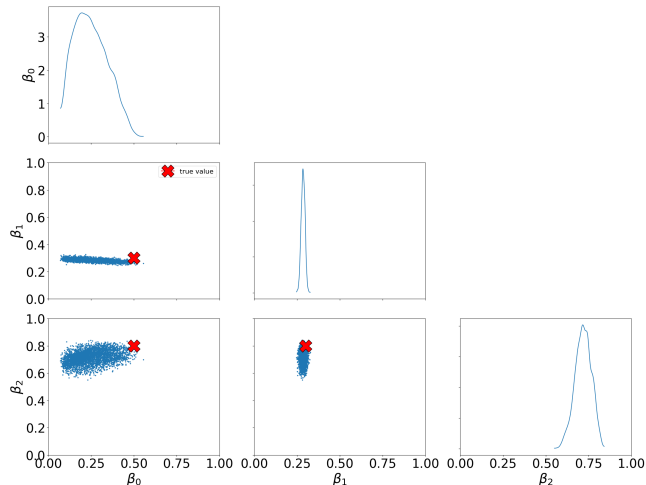
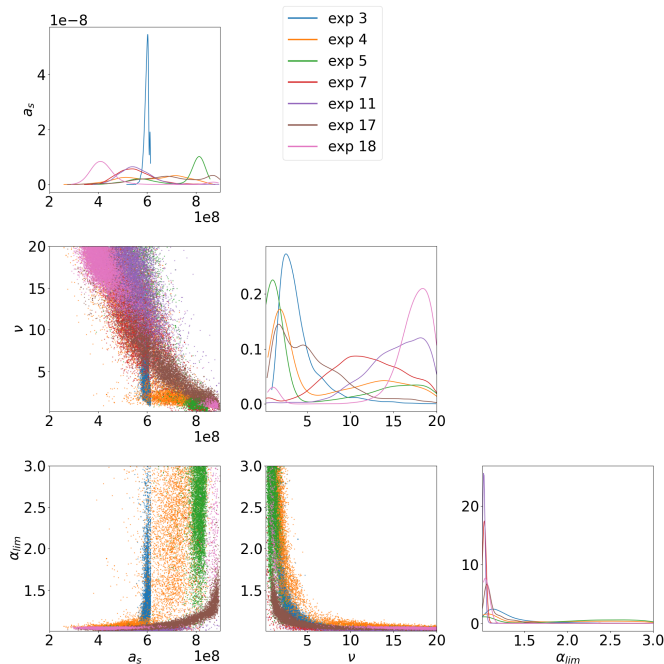


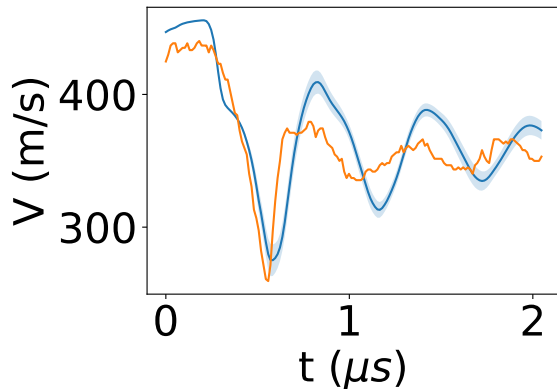
Figure: Pair plot of the posteriors without approximation.

Johnson: Sampling for one experiment

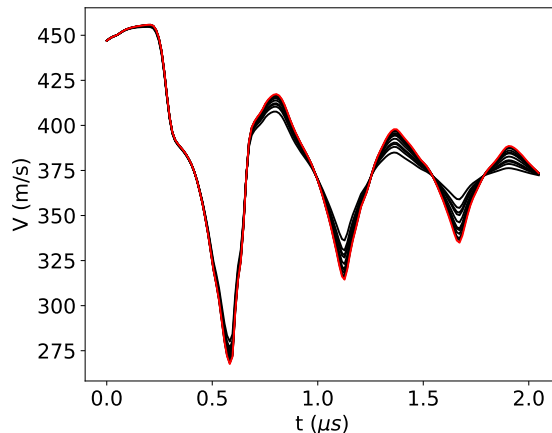


Damping discrepancy

Experiment vs Simulation



Effect of the damping discrepancy



$$T_{\rho}(y) = e^{-\rho t^2}(y - \bar{y}) + \bar{y}, \quad \text{where } \bar{y} = \frac{\int (1 - e^{-\rho t^2})^2 y(t) dt}{\int (1 - e^{-\rho t^2})^2 dt}$$

Sequential Monte Carlo (SMC)

Idea: move a population of particles from the prior to the posterior through a sequence of tempered distributions [[Del Moral et al., 2006](#)]:

$$p_{\alpha}(\beta \mid y) \propto p(y \mid \beta)^{\alpha} p(\beta), \quad 0 = \alpha_0 < \alpha_1 < \cdots < \alpha_K = 1$$

We use the PyMC implementation [[Abril-Pla et al., 2023](#)] of SMC:

- Draw N particles β_i from the prior ($\alpha_0 = 0$)
- Choose α_{k+1} adaptively so that the effective sample size stays near a threshold
- Reweight the particles with $w_i \propto p(y \mid \beta_i)^{\alpha_{k+1} - \alpha_k}$
- Resample the particles, then move each one with a few MCMC steps
- Repeat until $\alpha = 1$