

# Bayesian model calibration with interpolating polynomials based on adaptively weighted Leja nodes

L.M.M. van den Bos<sup>\*1,2</sup>, B. Sanderse<sup>1</sup>, W.A.A.M. Bierbooms<sup>2</sup>, and G.J.W. van Bussel<sup>2</sup>

<sup>1</sup>*Centrum Wiskunde & Informatica, P.O. Box 94079, 1090 GB, Amsterdam*

<sup>2</sup>*Delft University of Technology, P.O. Box 5, 2600 AA, Delft*

7th August 2018

## Abstract

An efficient algorithm is proposed for Bayesian model calibration, which is commonly used to estimate the model parameters of non-linear, computationally expensive models using measurement data. The approach is based on Bayesian statistics: using a prior distribution and a likelihood, the posterior distribution is obtained through application of Bayes' law. Our novel algorithm to accurately determine this posterior requires significantly fewer discrete model evaluations than traditional Monte Carlo methods. The key idea is to replace the expensive model by an interpolating surrogate model and to construct the interpolating nodal set maximizing the accuracy of the posterior. To determine such a nodal set an extension to weighted Leja nodes is introduced, based on a new weighting function. We prove that the convergence of the posterior has the same rate as the convergence of the model. If the convergence of the posterior is measured in the Kullback–Leibler divergence, the rate doubles. The algorithm and its theoretical properties are verified in three different test cases: analytical cases that confirm the correctness of the theoretical findings, Burgers' equation to show its applicability in implicit problems, and finally the calibration of the closure parameters of a turbulence model to show the effectiveness for computationally expensive problems.

**Keywords:** Bayesian model calibration, Interpolation, Leja nodes, Surrogate modeling

## 1 Introduction

Estimating model parameters from measurements is a problem of frequent occurrence in many fields of engineering and many different approaches exist to solve this problem. We consider non-linear calibration problems (or inverse problems) where a forward evaluation of the model is computationally expensive. The approach we follow is of a stochastic nature: the unknown parameters are modeled using probability distributions and information about these parameters is inferred using Bayesian statistics. This approach is often called *Bayesian model calibration*.

Bayesian model calibration [19, 31, 32] is a systematic way to calibrate the parameters of a computational model. By means of a statistical model to describe the relation between the model and the data, the calibrated parameters are obtained in the form of a random variable (called the *posterior*) by means of Bayes' law. These random variables can then be used to assess the uncertainty in the model and to make future predictions. This procedure is well-known in the field of Bayesian statistics, where the goal is to infer unmeasured quantities from data. The calibration approach has already been applied many times, for example to calibrate the closure parameters of turbulence models [7, 11]. A similar example is considered in this work.

Possibly the largest drawback of Bayesian model calibration is the expensive sampling procedure that is necessary. Because the posterior depends to a large extent on the model, which is only known implicitly (e.g. a computer code numerically solving a partial differential equation), determining a sample from the posterior

---

\*Corresponding author: [l.m.m.van.den.bos@cw.i.nl](mailto:l.m.m.van.den.bos@cw.i.nl)

is mostly done using Markov chain Monte Carlo (MCMC) methods [14, 25], requiring many expensive model evaluations. Improvements have been made to accelerate these MCMC methods, e.g. the DREAM algorithm [37] or adaptive sampling [40]. Replacing the sampling procedure itself is also possible, e.g. methods based on sparse grids [6, 22] or Approximate Bayesian Computation [2, 9, 20]. However, this encompasses stringent assumptions on the statistical model or still requires many model runs as the shape of the posterior is unknown.

A different approach is followed in the current article. In essence we are following the approach of Marzouk et al. [24], which has been used several times in literature [1, 4, 23, 27, 30, 41, 42]. The key idea in our procedure is to replace the model in the calibration step with a *surrogate* (or *response surface*) that approximates the computationally expensive model. MCMC can then be used to sample the resulting posterior without a large computational overhead.

It is customary to construct the surrogate with respect to the prior (or only its support). If the likelihood differs a lot from the prior, this is not used. It is even possible that no convergence is observed at all [21].

In this article we create a surrogate model through interpolation (instead of the commonly applied polynomial chaos expansion), built using Leja nodes. Probability density functions can be incorporated using weighted Leja nodes [18, 26]. We extend weighted Leja nodes to *adaptively* refine the interpolating polynomial by using obtained posterior information. As extensive theory about interpolation polynomials exists (e.g. [16]), we can prove convergence of the estimated posterior with mild assumptions on the likelihood. This extends previous work [4, 24], in which the likelihood is assumed to be Gaussian. The end result is an interpolating polynomial that can be used in conjunction with the likelihood and the prior to obtain statistics on the posterior.

To demonstrate the applicability of our methodology, we will employ three different classes of test problems. The first class consist of functions that are known explicitly and can be evaluated fast and accurate. We will use these to show the effectiveness of our nodal set compared to commonly used methods. The second class consists of problems that are defined implicitly, but do not require significant computational power to solve. For this, we employ the one-dimensional Burgers' equation. In this case, it is possible to compare the estimated posterior with a posterior determined using Monte Carlo methods. The last class consists of problems of such large complexity that a quantitative comparison with a true posterior is not possible anymore. As example we consider the calibration of closure coefficients of the Spalart–Allmaras turbulence model.

This paper is set up as follows. First, we discuss Bayesian model calibration and introduce the adaptively weighted Leja nodes. In Section 3 the theoretical properties of the algorithm are studied and its convergence is assessed. Section 4 contains numerical tests that show evidence of the theoretical findings and in Section 5 conclusions are drawn.

## 2 Bayesian model calibration with a surrogate

The focus is on the stochastic calibration of computationally expensive (possibly implicitly defined) models. We denote this model by  $u : \Omega \rightarrow \mathbb{R}$ , with  $\Omega \subset \mathbb{R}^d$  ( $d = 1, 2, \dots$ ). In this work we will not focus on the specific construction of this model, but for example  $u$  arises as a solution of a set of partial differential equations. Without loss of generality, we assume that  $u$  is a scalar quantity.  $u$  depends on  $d$  parameters, which we will denote as vector  $\boldsymbol{\vartheta} = (\vartheta_1, \dots, \vartheta_d)^\top \in \Omega$ . One can think of  $\boldsymbol{\vartheta}$  as parameters inherent to the model, such as fitting parameters or other closure parameters.

The goal of Bayesian model calibration is to infer knowledge about the model parameters, given measurement data of the process modeled by  $u$ . To this end, we assume that a vector of measurements  $\mathbf{z} = (z_1, \dots, z_n)^\top$  is given, with  $z_k \in \mathbb{R}$ . This data can be provided by various means, for example by measurements or by the results of a high-fidelity model. Using parameters  $\boldsymbol{\vartheta}$ , a statistical model is formulated describing a relation between the model  $u(\boldsymbol{\vartheta})$  and the data  $\mathbf{z}$  by means of random variables that model among others discrepancy, error, and uncertainty. For example, these random variables account for measurement errors and numerical tolerances. Using Bayesian statistics [12], the posterior of the parameters is formulated by means of a probability density function (PDF).

Throughout this article we let  $p(\boldsymbol{\vartheta})$  be the prior, a PDF containing all prior information of  $\boldsymbol{\vartheta}$  obtained through physical constraints, assumptions, or previous experiments. The likelihood  $p(\mathbf{z} \mid \boldsymbol{\vartheta})$  is obtained

through the statistical relation between the model  $u$  and the data  $\mathbf{z}$ . Possibly the most straightforward example is  $z_k = u(\boldsymbol{\vartheta}) + \varepsilon_k$ , where  $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^\top$  is assumed to be multivariate Gaussian distributed with mean  $\mathbf{0}$  and covariance matrix  $\Sigma$ . This yields the following likelihood:

$$p(\mathbf{z} \mid \boldsymbol{\vartheta}) \propto \exp \left[ -\frac{1}{2} \mathbf{d}^\top \Sigma^{-1} \mathbf{d} \right], \text{ with } \mathbf{d} \text{ a vector such that } d_k = z_k - u(\boldsymbol{\vartheta}). \quad (1)$$

$\mathbf{d}$  is the so-called misfit. Bayes' law is applied to obtain the posterior  $p(\boldsymbol{\vartheta} \mid \mathbf{z})$ , i.e.

$$p(\boldsymbol{\vartheta} \mid \mathbf{z}) \propto p(\mathbf{z} \mid \boldsymbol{\vartheta})p(\boldsymbol{\vartheta}). \quad (2)$$

The posterior PDF can be used to assess information about the parameters of the model, e.g. by determining the expectation or MAP estimate (i.e. the maximum of the posterior). The uncertainty of these parameters can be quantified by determining the moments of the posterior PDF.

Note that the posterior depends on the likelihood, which requires an expensive evaluation of the model (see (1)). Therefore sampling the posterior through the application of MCMC methods [14, 25] is typically intractable for such models.

Vector-valued models  $u$  can be incorporated in this framework straightforwardly, although the likelihood requires minor modifications. Typically an observation operator is introduced that restricts  $u$  to the locations where measurement data is available. We will discuss an example of this in Section 4.3.

Throughout this article we assume the likelihood is a continuously differentiable, Lipschitz continuous function of the misfit  $\mathbf{d}$  or (more general) of the model  $u$ . This is true for the multivariate Gaussian likelihood and (more general) for any likelihood which has additive errors (see [19] for more examples in the context of Bayesian model calibration). There are no further constraints on the structure of the likelihood and the prior in this work, but we do not incorporate the calibration of *hyperparameters*, i.e. parameters introduced solely in the statistical model (an example would be the calibration of the standard deviation of  $\boldsymbol{\varepsilon}$ ). Moreover, we assume the prior is not improper, although in practice our methods can be applied in such a setting.

The outline of the proposed calibration procedure is as follows. Let  $u_N$  be an interpolating surrogate of  $u$  using  $N$  distinct nodes and model evaluations. Using  $u_N$ , an estimated posterior can be determined, which is used to obtain the  $(N+1)^{\text{th}}$  node. The steps are repeated until convergence is observed. Finally MCMC can be applied to the resulting posterior, because the computationally expensive model is replaced with an explicitly known surrogate.

First, we briefly introduce the interpolation polynomial for sake of completeness. Then the nodal set we will use, the Leja nodes, will be introduced.

## 2.1 Interpolation methods

In general an interpolating polynomial can be defined as follows. Let  $u : \Omega \rightarrow \mathbb{R}$  with  $\Omega \subset \mathbb{R}^d$  be a continuous function. Let  $D$  be given and define the set  $\mathbb{P}(N, d)$  (with  $N = \binom{d+D}{D}$ ) to be all  $d$ -variate polynomials of degree  $D$  and lower. Using a nodal set  $X_{N+1} = \{\mathbf{x}_0, \dots, \mathbf{x}_N\}$  and evaluations of  $u$  at each node (i.e.  $u(\mathbf{x}_k)$  for  $k = 0, \dots, N$ ) the goal is to determine a polynomial  $p \in \mathbb{P}(N, d)$  such that

$$p(\mathbf{x}_k) = u(\mathbf{x}_k), \text{ for } k = 0, \dots, N. \quad (3)$$

This construction can be extended to any  $N = 1, 2, 3, \dots$ , provided that the monomials of the space  $\mathbb{P}(N, d)$  form a well-ordered set. Throughout this article, we use a graded reverse lexicographic order.

### 2.1.1 Univariate interpolation

In the case of  $d = 1$ , it is well-known that if all nodes are distinct the interpolation polynomial can be stated explicitly using Lagrange interpolating polynomials, i.e.

$$u_N(x) = (L_N u)(x) := \sum_{k=0}^N \ell_k^N(x) u(x_k), \text{ with } \ell_k^N(x) = \prod_{\substack{j=0 \\ j \neq k}}^N \frac{x - x_j}{x_k - x_j}. \quad (4)$$

Here  $L_N$  is a linear operator that yields a polynomial of degree  $N$ , which we will denote as  $u_N$ . By construction, the Lagrange basis polynomials  $\ell_k^N$  have the property  $\ell_k^N(x_j) = \delta_{k,j}$  (i.e.  $\ell_k^N(x_j) = 1$  if  $j = k$  and  $\ell_k^N(x_j) = 0$  otherwise). Therefore  $u_N(x_k) = u(x_k)$  for all  $k$ , such that it is indeed an interpolating polynomial.

The barycentric notation [3] can be used to numerically evaluate the interpolation polynomial given a nodal set (which is unconditionally stable [15]).

### 2.1.2 Multivariate interpolation

The Lagrange interpolating polynomials can be formulated in a multivariate setting, by defining them in terms of the determinant of a Vandermonde-matrix:

$$u_N(\mathbf{x}) = (L_N u)(\mathbf{x}) := \sum_{k=0}^N \ell_k^N(\mathbf{x}) u(\mathbf{x}_k), \text{ with } \ell_k^N(\mathbf{x}) = \frac{\det V(\mathbf{x}_0, \dots, \mathbf{x}_{k-1}, \mathbf{x}, \mathbf{x}_{k+1}, \dots, \mathbf{x}_N)}{\det V(\mathbf{x}_0, \dots, \mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_{k+1}, \dots, \mathbf{x}_N)}, \quad (5)$$

where  $V(\mathbf{x}_0, \dots, \mathbf{x}_N)$  is the  $(N+1) \times (N+1)$  Vandermonde-matrix with respect to the nodal set  $\{\mathbf{x}_0, \dots, \mathbf{x}_N\}$ , i.e.

$$V_{i,j}(\mathbf{x}_0, \dots, \mathbf{x}_N) = \mathbf{x}_j^{\alpha_i}. \quad (6)$$

Here,  $\alpha_k \in \mathbb{N}^d$  are defined such that for  $\alpha = (\alpha_1, \dots, \alpha_d)$  and  $\mathbf{x} = (x_1, \dots, x_d)$ , we have  $\mathbf{x}^\alpha = x_1^{\alpha_1} \dots x_d^{\alpha_d}$ . As stated before,  $\alpha_k$  are sorted using the graded reverse lexicographic order (i.e. first compare the total degree, then apply reverse lexicographic order to equal monomials). This implies that  $\|\alpha_k\|_1$  is a sorted sequence in  $k$ . Multivariate interpolation by means of this Vandermonde-matrix is only well-defined if  $V$  is non-singular (or equivalently, if  $X_{N+1}$  is a poised interpolation sequence with respect to  $\mathbb{P}(N, d)$ ). All nodal sequences constructed in this article are poised.

Evaluating a multivariate interpolating polynomial numerically can be done in various ways. A commonly used approach is to rewrite (6) as

$$V_{i,j}(\mathbf{x}_0, \dots, \mathbf{x}_N) = \varphi_j(\mathbf{x}_i), \quad (7)$$

where  $\varphi_j(\mathbf{x})$  are orthogonal basis polynomials (e.g. Chebyshev or Legendre polynomials). The polynomials  $\varphi_j$  form a linear combination of monomials, so mathematically speaking both approaches yield the same solution.

The nodal sets used in this article use the determinant of the Vandermonde-matrix. Therefore we use the QR factorization [13] of the Vandermonde-matrix to determine the interpolating polynomial and reuse the QR factorization to determine the nodal set (see Section 2.2 for details).

## 2.2 Weighted Leja nodes

For the purpose of Bayesian model calibration, we desire an algorithm to determine a nodal set  $X_N$  for any  $N$  having the following properties:

1. **Accuracy:** the nodal sets should yield an accurate posterior. We are mainly interested in estimating the posterior, i.e. it is not strictly necessary to have an accurate surrogate model on the full domain.
2. **Nested:** we require  $X_i \subset X_j$  for  $i < j$ , such that the obtained interpolant can be refined by reusing existing model evaluations.
3. **Weighting:** the goal is to determine the next node based on the posterior obtained so far. The algorithm of the nodal set should allow for this.

In this work we consider weighted Leja nodes, which form a sequence of nodes. The sequence is therefore by definition nested. First, we will define the univariate Leja nodes and generalize those to multivariate Leja nodes. Examples of univariate sequences are depicted in Figure 1.

The definition of weighted Leja nodes is by induction. Let  $\rho : \mathbb{R} \rightarrow \mathbb{R}_+$  be a bounded PDF and let  $\{x_0, \dots, x_N\}$  be a sequences of Leja nodes. Then the next node is defined by maximizing the numerator of  $\ell_{N+1}^{N+1}(x)$ , i.e.

$$x_{N+1} := \arg \max_{x \in \mathbb{R}} \rho(x) |x - x_0| |x - x_1| \dots |x - x_N|. \quad (8)$$

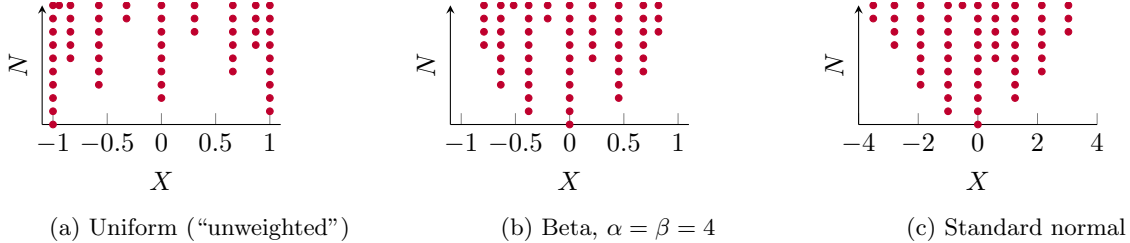


Figure 1: Univariate Leja sequences for various number of nodes and various well-known distributions.

If the maximum is not defined uniquely, any value of  $x$  maximizing the right hand side can be used (we pick the minimal one). The initial node  $x_0$  is defined as any value maximizing  $\rho(x)$ . Unweighted Leja nodes are defined with the uniform weighting function on  $[-1, 1]$ . We want to emphasize that multiplying the weighting function with a constant yields an identical sequence. This property is very useful for our purposes, as it allows us to neglect the constant of proportionality (often called the evidence) in Bayes' law.

Notice that the definition can be rewritten as follows:

$$x_{N+1} = \arg \max_{x \in \mathbb{R}, \rho(x) > 0} \log \rho(x) + \sum_{k=0}^N \log |x - x_k|. \quad (9)$$

Therefore for large  $N$ , the product  $|x - x_0| \cdots |x - x_N|$  will dominate the maximization, i.e. the effect of the weighting function  $\rho$  decreases for increasing  $N$ . Hence weighted Leja nodes become (approximately) unweighted Leja nodes for large  $N$ .

The definition of univariate Leja nodes can be generalized to a multidimensional setting in a similar way as we did in Section 2.1.2 with interpolation. To this end, let  $\rho : \mathbb{R}^d \rightarrow \mathbb{R}_+$  be a multivariate PDF. Let  $\mathbf{x}_0 \in \mathbb{R}^d$  be an initial node with  $\rho(\mathbf{x}_0) > 0$ . Then the next node is defined as follows:

$$\mathbf{x}_{N+1} := \arg \max_{\mathbf{x} \in \mathbb{R}^d} \rho(\mathbf{x}) |\det V(\mathbf{x}_0, \dots, \mathbf{x}_N, \mathbf{x})|. \quad (10)$$

Here,  $V$  is the Vandermonde-matrix defined in Section 2.1.2. The absolute value of the determinant of  $V$  is independent of the set of polynomials that is used to construct  $V$ , so the definition is mathematically the same for both monomials and orthogonal polynomials.

Determining multivariate Leja nodes is less trivial compared to univariate nodes and is typically done by randomly (or quasi-randomly) sampling the space of interest and selecting the node that results in the highest determinant. It is important to be able to use a large number of samples, so it must be possible to calculate the determinant fast. We suggest to calculate the determinant by an extended QR factorization of the  $(N+2) \times (N+1)$ -matrix  $V(\mathbf{x}_0, \dots, \mathbf{x}_N)$  and to add the column containing  $\mathbf{x}$  by applying a rank-1 update. If a QR factorization has been calculated to determine the interpolating polynomial (see Section 2.1.2), it can be reused here. As a rank-1 update is an efficient procedure, a large number of samples can be used and therefore we assume that the approximation error is negligible in this case. Examples of Leja sequences defined by (10) can be found in Figure 2.

### 2.3 Calibration using Leja nodes

In this section we will derive a weighting function to be used in the interpolation procedure discussed in the previous section, with the goal to approximate the posterior. Theoretical details are provided in Section 3.

If the posterior is known explicitly and samples can be readily drawn from it, it is possible to determine weighted Leja nodes with weighting function  $\rho(\boldsymbol{\vartheta}) = p(\boldsymbol{\vartheta} | \mathbf{z})$ . These nodes provide an interpolant that is very suitable for evaluating integrals with respect to the posterior (this is commonly known as Bayesian prediction). However, the posterior is generally not explicitly available because it depends on the model  $u$ , which in itself is not known explicitly and can only be determined on (finitely many) nodes. Therefore the need arises for an interpolation sequence that approximates  $u$  such that the posterior determined with this approximation is accurate.

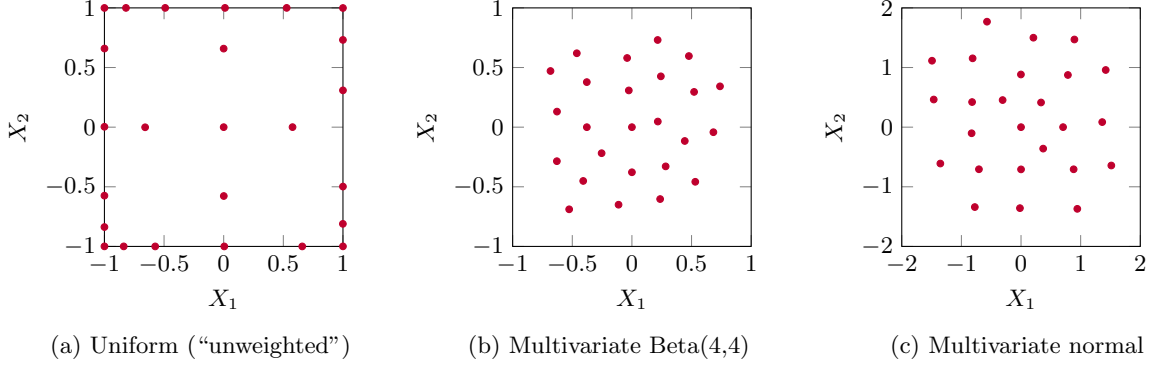


Figure 2: Multivariate Leja sequences of 25 nodes using various well-known distributions.

To this end, let  $p_N(\boldsymbol{\vartheta} | \mathbf{z})$  be the posterior determined using  $u_N(\boldsymbol{\vartheta})$ , i.e. the interpolant of  $u$  using  $N + 1$  nodes in  $\boldsymbol{\vartheta}$ . We will use the definition of the weighted Leja nodes from (10) to determine the next node. The natural idea is to construct  $p_{N+1}(\boldsymbol{\vartheta} | \mathbf{z})$  (i.e. a new approximation of the posterior) by determining a new weighted Leja node using  $p_N(\boldsymbol{\vartheta} | \mathbf{z})$  (i.e. the existing approximation of the posterior). Such a sequence is numerically unstable, because it solely places nodes in regions where the approximate posterior is high and therefore yields spurious oscillations in other regions in the domain.

The key idea is to balance between the accuracy of the interpolant and the accuracy of the posterior. There are various methods to do this, but we choose to temper the effect of the (possibly inaccurate) approximate posterior by adding a constant value  $\zeta$  to it. The higher this  $\zeta$ , the more the posterior tends to a uniform distribution. The weighted Leja nodes therefore tend to unweighted Leja nodes, such that the interpolatory sequence becomes stable.

To introduce this construction formally, we assume that a function  $P : \mathbb{R} \rightarrow \mathbb{R}_+$  (we assume  $0 \in \mathbb{R}_+$ ) exists such that

$$p(\mathbf{z} | \boldsymbol{\vartheta}) = P(u(\boldsymbol{\vartheta})), \quad (11)$$

where  $P$  typically is a PDF which follows from the statistical model. In the example discussed in (1)  $P$  is a Gaussian PDF, i.e.

$$P : \mathbb{R} \rightarrow \mathbb{R}_+, \text{ with } P(u) \propto \exp \left[ -\frac{1}{2} \mathbf{d}^T \Sigma^{-1} \mathbf{d} \right] \text{ and } d_k = z_k - u. \quad (12)$$

We assume that the function  $P$  is globally Lipschitz continuous and continuously differentiable. Many statistical models used in a statistical setting yield Lipschitz continuous  $P$ , because a bounded continuously differentiable function  $P(u)$  with  $P'(u) \rightarrow 0$  for  $u \rightarrow \pm\infty$  is Lipschitz continuous. The domain of definition of  $P$  is the image of the model  $u(\boldsymbol{\vartheta})$ , so functions  $P$  that are only Lipschitz continuous in the set described by the image of  $u(\boldsymbol{\vartheta})$  also fit in this framework (for example the Gamma distribution on the positive real axis).

It follows that for a fixed  $\boldsymbol{\vartheta}$ , we have by application of the mean value theorem:

$$\begin{aligned} |p_N(\boldsymbol{\vartheta} | \mathbf{z}) - p(\boldsymbol{\vartheta} | \mathbf{z})| &= |P(u_N(\boldsymbol{\vartheta})) - P(u(\boldsymbol{\vartheta}))|p(\boldsymbol{\vartheta}) \\ &= |P'(\xi)| |u_N(\boldsymbol{\vartheta}) - u(\boldsymbol{\vartheta})| p(\boldsymbol{\vartheta}), \end{aligned} \quad (13)$$

for some  $\xi \in [u_N(\boldsymbol{\vartheta}), u(\boldsymbol{\vartheta})]$ . Here the derivative  $P'$  is with respect to  $u$ , i.e.

$$P'(u(\boldsymbol{\vartheta})) = \frac{\partial P}{\partial u}(u(\boldsymbol{\vartheta})). \quad (14)$$

We want to emphasize that for the evaluation of  $P'(u_N(\boldsymbol{\vartheta}))$  no costly evaluation of the full model  $u$  is necessary, since  $P'$  is independent of  $u$ . In the example from (1),  $P'$  becomes the following:

$$P' : \mathbb{R} \rightarrow \mathbb{R}, \text{ with } P'(u) \propto -\frac{1}{2} (\mathbf{1}^T \Sigma^{-1} \mathbf{d} + \mathbf{d}^T \Sigma^{-1} \mathbf{1}) \exp \left[ -\frac{1}{2} \mathbf{d}^T \Sigma^{-1} \mathbf{d} \right], \text{ with } d_k = z_k - u, \quad (15)$$

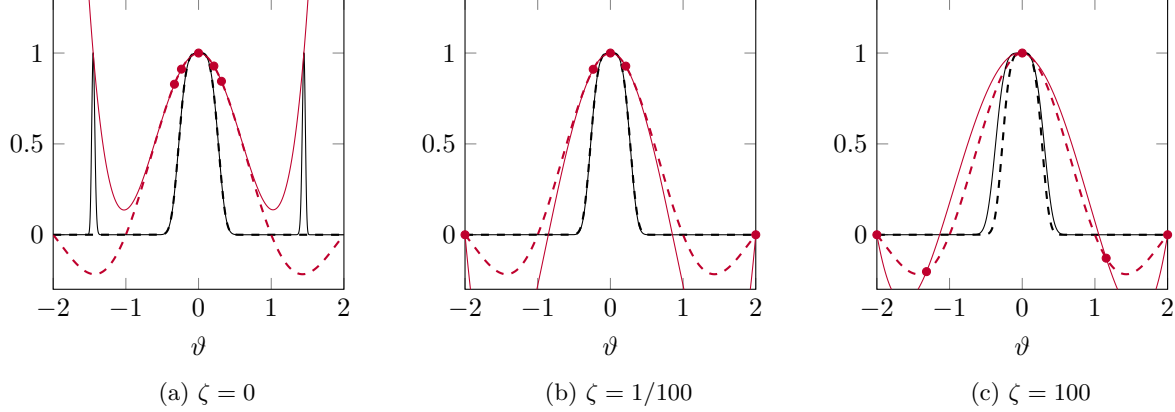


Figure 3: Interpolation of the sinc function using 5 weighted Leja nodes with respect to the posterior and a tempering parameter  $\zeta$ . The model and (unscaled) posterior are depicted in red and black respectively (the “true” value is dashed).

with  $\mathbf{1} = (1, 1, \dots, 1)^T \in \mathbb{R}^n$ . Taking the  $\infty$ -norm in  $\boldsymbol{\vartheta}$  on both sides of (13) yields:

$$\begin{aligned} \|p_N(\boldsymbol{\vartheta} \mid \mathbf{z}) - p(\boldsymbol{\vartheta} \mid \mathbf{z})\|_\infty &= \|P'(\xi)(u_N(\boldsymbol{\vartheta}) - u(\boldsymbol{\vartheta}))p(\boldsymbol{\vartheta})\|_\infty \\ &\leq \| [P'(u_N(\boldsymbol{\vartheta})) + \zeta] u_N(\boldsymbol{\vartheta}) - u(\boldsymbol{\vartheta}) p(\boldsymbol{\vartheta}) \|_\infty, \end{aligned} \quad (16)$$

with  $\zeta \geq |P'(\xi) - P'(u_N(\boldsymbol{\vartheta}))|$ . The constant  $\zeta$  is a measure for how far the likelihood of the interpolant is from the likelihood of the true model. The idea is to construct an accurate interpolant where  $P'(u_N(\boldsymbol{\vartheta})) + \zeta$  is large, while at the same time ensuring stability of the interpolant. Concluding, we propose to construct  $u_N$  using weighted Leja nodes which are determined using the following weighting function:

$$q_N(\boldsymbol{\vartheta} \mid \mathbf{z}) = |P'(u_N(\boldsymbol{\vartheta}))|p(\boldsymbol{\vartheta}) + \zeta p(\boldsymbol{\vartheta}), \text{ where } \zeta > 0 \text{ is a free parameter,} \quad (17)$$

i.e. if  $\boldsymbol{\vartheta}_0, \dots, \boldsymbol{\vartheta}_N$  are the first  $N + 1$  interpolation nodes,  $\boldsymbol{\vartheta}_{N+1}$  is determined as follows:

$$\boldsymbol{\vartheta}_{N+1} = \arg \max_{\boldsymbol{\vartheta}} q_N(\boldsymbol{\vartheta} \mid \mathbf{z}) |\det V(\boldsymbol{\vartheta}_0, \dots, \boldsymbol{\vartheta}_N, \boldsymbol{\vartheta})|. \quad (18)$$

Examples of posteriors obtained with this weighting function compared with the exact posterior are depicted in Figure 3. Here the univariate function  $u(\vartheta) = \text{sinc}(\vartheta) = \sin(\vartheta)/\vartheta$  is “calibrated” using the Gaussian likelihood with  $\sigma = 1/10$  and one data point at  $z_1 = 1$ . If  $\zeta = 0$  (no tempering) the interpolant is indeed accurate with respect to the posterior, but yields an incorrect approximate posterior because the interpolant crosses the value of the data incorrectly around  $\vartheta = \pm 1.5$ . For  $\zeta = 100$ , it can be guaranteed that  $\zeta \geq |P'(\xi) - P'(u_N(\boldsymbol{\vartheta}))|$ , but such a value yields approximately unweighted Leja nodes. However, even for a very small non-negative value of  $\zeta$ , a significant improvement over  $\zeta = 0$  is noticed. This does demonstrate the importance of tempering on the effect of the approximate posterior.

We will discuss the parameter  $\zeta$  in more detail in Section 3.5. Combining this weighting function with (10) (so  $\rho = q_N$ ) yields a nodal set, which can be used in conjunction with (5) to create the interpolant. Finally, using this interpolant a new  $q_N$  can be determined and the process can be repeated. A schematic overview of this procedure is depicted in Figure 4. Convergence can be assessed in various ways, for example using the  $\infty$ -norm or the Kullback–Leibler divergence. We will use the  $\infty$ -norm, as determining the Kullback–Leibler norm in higher-dimensional spaces is numerically challenging.

### 3 Convergence analysis

In this section the convergence of the estimated posterior to the true posterior is studied. First, the convergence properties of interpolation methods are reviewed as these form the basis of our adaptive approach. Based on this, convergence of the estimated posterior to the true posterior in both the  $\infty$ -norm and the Kullback–Leibler divergence can be proved.

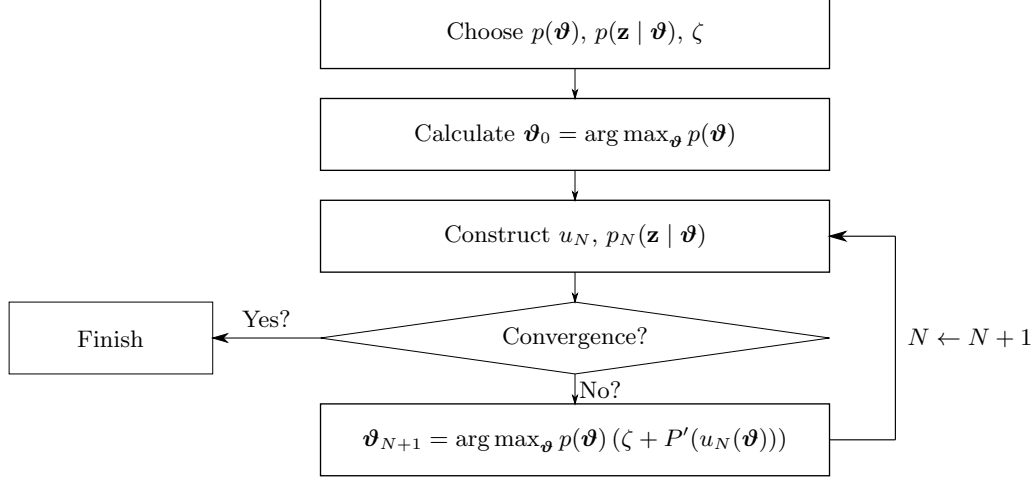


Figure 4: Schematic overview of the algorithm proposed in this article.

### 3.1 Lebesgue constant

The accuracy of interpolation methods can be assessed in two ways: using *point wise* error bounds which are typically based on Taylor expansions and *global* error bounds which are typically based on the Lebesgue inequality. We will use the latter type.

Let  $\mathbb{P}(K) = \mathbb{P}(K, 1)$ , i.e. all univariate polynomials of degree less or equal than  $K$ . We assume  $u \in C[-1, 1]$  (i.e. a continuous function defined in  $[-1, 1]$ ) and  $\|u\|_\infty := \sup_{x \in [-1, 1]} |u(x)|$  if not stated otherwise. It is well-known that  $C[-1, 1]$  equipped with the norm  $\|\cdot\|_\infty$  forms a Banach space.

Let  $X_N = \{x_0, \dots, x_N\} \subset [-1, 1]$  be a set of interpolation nodes and let  $L_N : C[-1, 1] \rightarrow \mathbb{P}(N)$  be the Lagrange interpolation operator (see (4)) that determines the interpolating polynomial given the nodal set  $X_N$ . Then for any polynomial  $\varphi_N$  of degree  $N$  we have

$$\|L_N u - u\|_\infty \leq (1 + \|L_N\|_\infty) \|\varphi_N - u\|_\infty \quad (19)$$

where  $\Lambda_N := \|L_N\|_\infty = \sup_{x \in [-1, 1]} \sum_{k=0}^N |\ell_k^N(x)|$  is the operator norm of  $L_N$  induced by the norm  $\|\cdot\|_\infty$  discussed above.  $\Lambda_N$  is called the Lebesgue constant [16]. The inequality holds for any polynomial  $\varphi_N$ , so an immediate result is the Lebesgue inequality:

$$\|L_N u - u\|_\infty \leq (1 + \Lambda_N) \inf_{\varphi_N \in \mathbb{P}(N)} \|\varphi_N - u\|_\infty. \quad (20)$$

In the procedure used for calibration, nodes are determined using a weighting function  $\rho$ . To assess the accuracy of nodes that use weighting, we introduce the weighted  $\infty$ -norm as follows:

$$\|u\|_\rho := \|\rho u\|_\infty. \quad (21)$$

Here, we assume that  $\rho(x) : \Omega \rightarrow \mathbb{R}_+$  is a bounded PDF<sup>1</sup>, such that that  $C(\Omega)$  equipped with the norm  $\|\cdot\|_\rho$  forms a Banach space. The support of  $u$  (say  $\Omega$ ) is allowed to be unbounded, in contrast to the unweighted case. The unweighted case is a special case of the weighted case.

Using this norm we can derive a similar estimate as (20) by introducing [18]:

$$\Lambda_N^\rho := \|L_N\|_\rho = \sup_{x \in \mathbb{R}} \sum_{k=0}^N \frac{\rho(x) |\ell_k^N(x)|}{\rho(x_k)}. \quad (22)$$

Here,  $\Lambda_N^\rho := \|L_N\|_\rho$  is the weighted Lebesgue constant, i.e. the norm of the operator  $u \rightarrow \rho L_N(u/\rho)$ . We call the result the weighted Lebesgue inequality:

$$\|L_N u - u\|_\rho \leq (1 + \Lambda_N^\rho) \inf_{\varphi_N \in \mathbb{P}(N)} \|\varphi_N - u\|_\rho. \quad (23)$$

<sup>1</sup>The boundedness is not strictly necessary for  $C(\Omega)$  to be a Banach space, but for the cases discussed in this article boundedness suits.



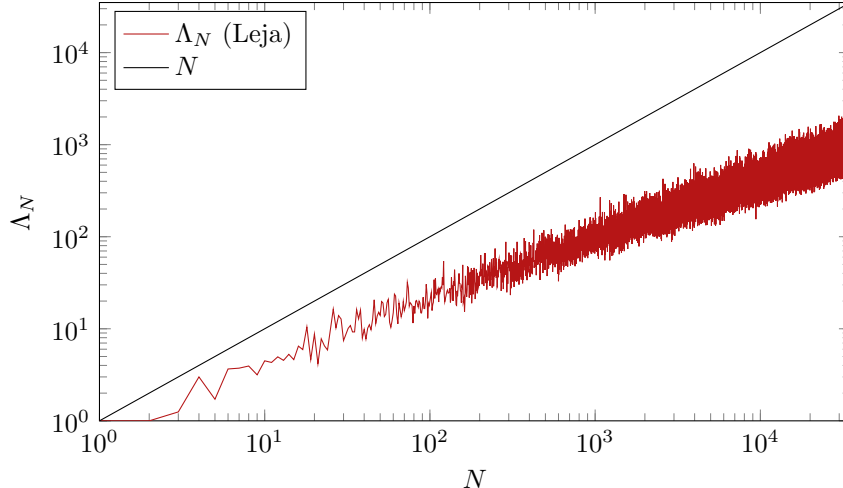


Figure 5: The Lebesgue constant of a univariate unweighted Leja sequence is bounded by  $N$ . This plot has been determined numerically.

### 3.2 Accuracy of nodal sets

It is both an advantage and a disadvantage that the Lebesgue constant solely depends on the nodal set: we do not have to take the model into account to estimate the accuracy, but the resulting estimate does not take any properties of the model into account. The algorithm discussed in Section 2.3 does use the model and therefore cannot be fit directly in the framework set out in the previous section.

Many nodal sets exist with a logarithmically growing Lebesgue constant, which is asymptotically the optimal growth. For example, Chebyshev nodes (i.e. the nodes from the Clenshaw–Curtis quadrature rule) have  $\Lambda_N = \mathcal{O}(\log N)$  [16]. Moreover, we already stated that the Chebyshev nodes are nested such that the nodes for  $N = 2^l + 1$  (for integer  $l$ ) are contained in the nodes for  $N = 2^{l+1} + 1$ . However, the Chebyshev nodes are only defined in an *unweighted* setting. Another well-known example are equidistant nodes, which have an exponentially growing Lebesgue constant. This can be observed by interpolating Runge’s function.

Although it is known that the Lebesgue constant of both weighted and unweighted Leja sequences grows sub-exponentially [18, 35, 36], the exact growth (or a strict upper bound) is not known. We have therefore numerically determined the Lebesgue constant for  $N$  up to 30 000 for several distributions and observed  $\Lambda_N \leq N$ , see Figure 5 for the case of unweighted univariate Leja nodes. For weighted Leja nodes with respect to the beta distribution and the Gaussian distribution the plot is similar.

Moreover, except for some specific cases (such as the unit disk [5]), not much is known about the Lebesgue constant in the multivariate case. Unfortunately, as far as the authors know, it is difficult to determine multivariate Leja nodes accurate enough to create a similar plot as Figure 5. For small number of nodes ( $N \lesssim 100$ ) and low dimensionality ( $d \lesssim 5$ ) the Lebesgue constants do grow similarly, but this number is too small to draw conclusions about general asymptotic behavior. However, the numerically determined growth of the Lebesgue constant is sufficient for the purposes in the current article.

### 3.3 Jackson’s inequality

The interpolation error from (20) contains two main parts:  $\Lambda_N$  and  $\inf_{\varphi \in \mathbb{P}(N)} \|u - \varphi\|_\infty$ . In the previous section we discussed some properties of  $\Lambda_N$ , the first part of this expression. The latter part can be estimated using Jackson’s inequality [17], for which we need the notion of the modulus of continuity.

**Definition 1.** Let  $u \in C[-1, 1]$ . Then the modulus of continuity of  $u$ , denoted by  $\omega_u : [0, \infty) \rightarrow [0, \infty)$  is a function such that

$$|u(x) - u(y)| \leq \omega_u(|x - y|). \quad (24)$$

Jackson’s inequality relates the approximation error with the modulus of continuity of the function  $u$ .

**Theorem 2** (Jackson [17], 1982). *Let  $u \in C[-1, 1]$  and  $N > 0$  be given. Then there exists a  $\varphi \in \mathbb{P}(N)$  such that*

$$\|u - \varphi\|_\infty \leq C\omega_u(N^{-1}). \quad (25)$$

Here,  $C$  is independent of  $N$  and  $u$ .

*Proof.* See [29]. □

Using Jackson's inequality, we can prove a bound on  $\inf_{\varphi \in \mathbb{P}(N)} \|u - \varphi\|_\infty$  and prove convergence of the interpolant, e.g. for Lipschitz continuous functions  $u$ , which have  $\omega_u(\delta) = C\delta$ , we obtain:

$$\|u_N - u\|_\infty \leq (1 + \Lambda_N) \inf_{\varphi \in \mathbb{P}(N)} \|u - \varphi\|_\infty \leq C \frac{1 + \Lambda_N}{N}. \quad (26)$$

So if  $(1 + \Lambda_N)/N \rightarrow 0$  (see Figure 5), convergence is proved. There exist many variants of Jackson's equality, e.g. for higher dimensional spaces and functions that have high-order derivatives. If a function is analytic, exponential convergence can be proved. The interested reader is referred to the accessible introduction in the book of Watson [38] and the references therein for more information.

### 3.4 Convergence of the posterior: the general case

In this section we study convergence of the estimated posterior to the true posterior in the  $\infty$ -norm, given convergence of the interpolant. Convergence of the interpolant can be assessed using both the Lebesgue constant and Jackson's inequality. As discussed previously, let  $p(\boldsymbol{\vartheta})$ ,  $p(\mathbf{z} \mid \boldsymbol{\vartheta})$ , and  $p(\boldsymbol{\vartheta} \mid \mathbf{z})$  be the prior, likelihood, and posterior respectively. Let  $u$  be the model, such that  $u(\boldsymbol{\vartheta})$  is a model evaluation using a fixed set of parameters  $\boldsymbol{\vartheta}$ . We assume an interpolant  $u_N$  is given such that  $u_N$  converges to  $u$  for increasing  $N$  and let  $p_N(\mathbf{z} \mid \boldsymbol{\vartheta})$  and  $p_N(\boldsymbol{\vartheta} \mid \mathbf{z})$  be the likelihood and the posterior constructed with this interpolant, i.e.  $p_N(\mathbf{z} \mid \boldsymbol{\vartheta}) = P(u_N(\boldsymbol{\vartheta}))$ . The main result is that if the interpolant converges to the model (which can be proved using the techniques outlined in the previous section), the estimated posterior converges to the true posterior. We do not need differentiability in this general case.

**Theorem 3.** *Let  $u : \Omega \rightarrow \mathbb{R}$  be a continuous function and let  $u_N$  be an interpolation of  $u$  with  $N$  nodes. Suppose*

$$\|u_N - u\|_{p(\boldsymbol{\vartheta})} \leq CN^{-\alpha} \quad (27)$$

for some positive constants  $C$  and  $\alpha$  (i.e.  $u_N$  converges to  $u$  with rate  $\alpha$ ). Assume the likelihood (i.e. the function  $P$ ) is Lipschitz continuous.

Then

$$\|p_N(\boldsymbol{\vartheta} \mid \mathbf{z}) - p(\boldsymbol{\vartheta} \mid \mathbf{z})\|_\infty \leq KN^{-\alpha}, \quad (28)$$

where  $K$  is a positive constant.

*Proof.* Recall the definition of  $P$  from (11):  $p(\mathbf{z} \mid \boldsymbol{\vartheta}) = P(u(\boldsymbol{\vartheta}))$ . Let  $Q$  be the Lipschitz constant of  $P$ . Convergence readily follows:

$$\begin{aligned} \|p_N(\boldsymbol{\vartheta} \mid \mathbf{z}) - p(\boldsymbol{\vartheta} \mid \mathbf{z})\|_\infty &= \|(p_N(\mathbf{z} \mid \boldsymbol{\vartheta}) - p(\mathbf{z} \mid \boldsymbol{\vartheta}))p(\boldsymbol{\vartheta})\|_\infty \\ &= \|[P(u_N(\boldsymbol{\vartheta})) - P(u(\boldsymbol{\vartheta}))]p(\boldsymbol{\vartheta})\|_\infty \\ &\leq Q\|(u_N(\boldsymbol{\vartheta}) - u(\boldsymbol{\vartheta}))p(\boldsymbol{\vartheta})\|_\infty \\ &= Q\|u_N(\boldsymbol{\vartheta}) - u(\boldsymbol{\vartheta})\|_{p(\boldsymbol{\vartheta})} \\ &\leq QCN^{-\alpha}, \end{aligned} \quad (29)$$

i.e. if  $u_N$  converges to  $u$  in the weighted  $p(\boldsymbol{\vartheta})$ -norm, the estimated posterior converges to the true posterior with at least the same rate of convergence. This concludes the proof of Theorem 3, extending previous work [4, 23] from Gaussian likelihoods to general likelihoods. □

### 3.5 Adaptivity in the weighting function

Recall the weighting function introduced in Section 2.3:

$$q_N(\boldsymbol{\vartheta} \mid \mathbf{z}) = |P'(u_N(\boldsymbol{\vartheta}))|p(\boldsymbol{\vartheta}) + \zeta p(\boldsymbol{\vartheta}), \text{ with } \zeta > 0. \quad (30)$$

This weighting function changes throughout the calibration procedure, because it depends on  $u_N(\boldsymbol{\vartheta})$ .

We have used this function to construct an algorithm that creates an interpolating surrogate using the posterior (see Figure 4). It holds that  $\zeta \approx |P'(\xi) - P'(u_N(\boldsymbol{\vartheta}))|$  yields the fastest convergence, but in practice this value is rarely known exactly. Fortunately, the surrogate will converge for all positive  $\zeta$ , which can be seen as follows.  $q_N$  and  $p$  are both positive functions, hence we can rewrite the definition of Leja nodes as follows:

$$\begin{aligned} \boldsymbol{\vartheta}_{N+1} = \arg \max_{\boldsymbol{\vartheta}} q_N(\boldsymbol{\vartheta}) |\det V(\boldsymbol{\vartheta}_0, \dots, \boldsymbol{\vartheta}_N, \boldsymbol{\vartheta})| = \\ \arg \max_{\boldsymbol{\vartheta}} (\log(|P'(u_N(\boldsymbol{\vartheta}))| + \zeta) + \log p(\boldsymbol{\vartheta}) + \log |\det V(\boldsymbol{\vartheta}_0, \dots, \boldsymbol{\vartheta}_N, \boldsymbol{\vartheta})|). \end{aligned} \quad (31)$$

We have that  $\log(|P'(u_N(\boldsymbol{\vartheta}))| + \zeta)$  is bounded from below (because  $\zeta > 0$ ) and from above (because  $P'$  is bounded). Moreover, the sequence  $\lim_{N \rightarrow \infty} \arg \max_{\boldsymbol{\vartheta}} |\det V(\boldsymbol{\vartheta}_0, \dots, \boldsymbol{\vartheta}_N, \boldsymbol{\vartheta})|$  diverges. Hence for large  $N$ , the last term dominates the maximization, such that the sequences of nodes becomes a sequence of weighted Leja nodes using the prior  $p(\boldsymbol{\vartheta})$ , which yield a converging surrogate (provided that weighted Leja nodes provide a converging surrogate).

The key point in obtaining a converging surrogate is that  $\zeta > 0$ . If  $\zeta = 0$ , the inaccuracy of  $u_N$  can significantly deteriorate the convergence (recall Figure 3), except possibly if  $u$  is globally analytic. If the goal is to optimize  $\zeta$ , we suggest a heuristically adaptive approach. Start with  $\zeta = \zeta_0 > 0$  and for each iteration, multiply  $\zeta$  with a constant  $k > 1$  if the error in the posterior increases and divide  $\zeta$  by  $k$  if the interpolation error decreases. The error can be estimated by comparing two consecutive approximate posteriors. This procedure is however not necessary to obtain convergence for the examples in this article, for which a fixed value of  $\zeta$  is sufficient.

The convergence theorems (e.g. Theorem 3) in the previous sections can be used to prove convergence of the posterior. The rate of convergence of the adaptive interpolating surrogate is similar to the rate derived in Section 3.4. However, the constant of proportionality (denoted with  $Q$  previously) is smaller. In general, a small  $\zeta$  yields fast convergence but if  $\zeta$  is too small, spurious oscillations in the interpolant hinder convergence for small  $N$ . The “best”  $\zeta$  should be picked based on the specifics of the model, the prior, and the likelihood under consideration, but one can pick it slightly too large to prevent numerical instabilities for small  $N$ . We will demonstrate this in Section 4.

### 3.6 Convergence properties

We assess the convergence properties of our algorithm using the Kullback–Leibler divergence, which is often used in a Bayesian setting to measure differences between distributions.

#### 3.6.1 Convergence of the Kullback–Leibler divergence

Given two probability density functions  $p(x)$  and  $q(x)$  defined on a set  $\Omega$ , the Kullback–Leibler divergence is defined as follows:

$$D_{\text{KL}}(p(x) \parallel q(x)) = \int_{\Omega} p(x) \log \frac{p(x)}{q(x)} dx. \quad (32)$$

The KL-divergence is always positive, equals 0 if (and only if)  $p \equiv q$ , and is finite if  $p(x) = 0$  implies  $q(x) = 0$  (here, it is used that  $\lim_{x \rightarrow 0} x \log x = 0$ ).

We are interested in proving bounds on  $D_{\text{KL}}(p_N(\boldsymbol{\vartheta} \mid \mathbf{z}) \parallel p(\boldsymbol{\vartheta} \mid \mathbf{z}))$  or  $D_{\text{KL}}(p(\boldsymbol{\vartheta} \mid \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} \mid \mathbf{z}))$ . This is non-trivial due to the logarithm in the integral. To prove convergence, we first prove pointwise convergence of the logarithm, extend this to convergence of the integral using Fatou’s lemma and finally conclude that

convergence is attained. As the Kullback–Leibler divergence is defined for probability density functions, we have to incorporate the scaling of the posterior again. To this end, let  $\gamma_N$  and  $\gamma$  be defined as follows:

$$\gamma = \int_{\Omega} p(\mathbf{z} \mid \boldsymbol{\vartheta}) p(\boldsymbol{\vartheta}) \, d\boldsymbol{\vartheta}, \quad (33)$$

$$\gamma_N = \int_{\Omega} p_N(\mathbf{z} \mid \boldsymbol{\vartheta}) p(\boldsymbol{\vartheta}) \, d\boldsymbol{\vartheta}. \quad (34)$$

To start off, the following lemma provides pointwise convergence of  $\log \frac{p_N(x)}{p(x)}$ . We omit the proof.

**Lemma 4.** *Let  $g_n : \Omega \rightarrow \mathbb{R}$  be a series of functions with  $g_n(x) \rightarrow g(x)$  for  $n \rightarrow \infty$ , for all  $x \in \Omega$ . Assume  $g_n > 0$  for all  $n$  and  $g > 0$ . Then*

$$\log \frac{g_n(x)}{g(x)} \rightarrow 0, \text{ for } n \rightarrow \infty, \text{ for all } x \in \Omega. \quad (35)$$

Note that the generalization to *uniform* convergence is not trivial. By definition the Kullback–Leibler divergence does not require uniform convergence, but only convergence in the integral. As the functions we are using are probability density functions, Fatou’s lemma is handy. It is well-known and we omit the proof.

**Lemma 5.** *Let  $f_1, f_2, \dots$  be a sequence of extended real-valued measurable functions. Let  $f = \limsup_{n \rightarrow \infty} f_n$ . If there exists a non-negative integrable function  $g$  (i.e.  $g$  measurable and  $\int_{\Omega} g \, d\mu < \infty$ ) such that  $f_n \leq g$  for all  $n$ , then*

$$\limsup_{n \rightarrow \infty} \int_{\Omega} f_n \, d\mu \leq \int_{\Omega} f \, d\mu. \quad (36)$$

We are now in a position to prove convergence of the Kullback–Leibler divergence, given pointwise convergence of the posterior. As uniform convergence of the posterior has been studied extensively in Section 3.4, assuming pointwise convergence is not a restriction. However, we additionally assume positivity of the posterior, as the Kullback–Leibler becomes undefined otherwise.

**Theorem 6.** *Suppose  $\|p_N(\boldsymbol{\vartheta} \mid \mathbf{z}) - p(\boldsymbol{\vartheta} \mid \mathbf{z})\|_{\infty} \rightarrow 0$  for  $N \rightarrow \infty$  and both  $p_N > 0$  and  $p > 0$  in  $\Omega$ . Then*

$$D_{\text{KL}}(p(\boldsymbol{\vartheta} \mid \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} \mid \mathbf{z})) \rightarrow 0. \quad (37)$$

*Proof.* The proof consists of combining Lemma 4 and 5. The result follows from applying Lemma 5 with  $f_n = p \log \frac{p}{p_N}$  and  $f = 0$ , in conjunction with  $g = \sup_N \log \frac{p}{p_N} + 1$ . A direct application of this lemma yields  $D_{\text{KL}}(p(\boldsymbol{\vartheta} \mid \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} \mid \mathbf{z})) \rightarrow 0$ . However, to apply Lemma 5, pointwise convergence of  $\log \frac{p}{p_N}$  to 0 is necessary. This can easily be seen by applying Lemma 4, with  $g_N = p_N$  and  $g = p$ .  $\square$

In a similar way, convergence of  $D_{\text{KL}}(p_N(\boldsymbol{\vartheta} \mid \mathbf{z}) \parallel p(\boldsymbol{\vartheta} \mid \mathbf{z})) \rightarrow 0$  can be proved.

### 3.6.2 Convergence rate of the Kullback–Leibler divergence

So far Theorem 6 only demonstrates convergence of the Kullback–Leibler divergence. The exact rate of convergence (or the constant involved in it) has not been deduced. In this section we will prove that the convergence rate doubles under general assumptions. These assumptions are:

- (A.1).  $\|p_N(\boldsymbol{\vartheta} \mid \mathbf{z}) - p(\boldsymbol{\vartheta} \mid \mathbf{z})\|_{\infty} \leq CN^{-\alpha}$ ,
- (A.2).  $p_N(\mathbf{z} \mid \boldsymbol{\vartheta}) > 0$  and  $p(\mathbf{z} \mid \boldsymbol{\vartheta}) > 0$ ,
- (A.3).  $p(\boldsymbol{\vartheta})$  has compact support.

The first assumption states that the estimated posterior converges with rate  $\alpha$ , which can be shown using Theorem 3. Many statistical settings (such as the model from (1) with a uniform prior) fit in these assumptions. Note that the last assumption restricts the prior such that it has bounded support, so priors on an unbounded domain (e.g. the improper uniform prior or Jackson’s prior) cannot be used in the setting of this section. The convergence result reads as follows.

**Theorem 7.** *Suppose (A.1), (A.2), and (A.3) hold.*

*Then*

$$D_{\text{KL}}(p(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} | \mathbf{z})) \leq KN^{-2\alpha}. \quad (38)$$

*Proof.* The Kullback–Leibler divergence is always positive, hence

$$\begin{aligned} D_{\text{KL}}(p(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} | \mathbf{z})) &\leq D_{\text{KL}}(p_N(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p(\boldsymbol{\vartheta} | \mathbf{z})) + D_{\text{KL}}(p(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} | \mathbf{z})) \\ &= \int_{\Omega} p_N(\boldsymbol{\vartheta} | \mathbf{z}) \log \frac{p_N(\boldsymbol{\vartheta} | \mathbf{z})}{p(\boldsymbol{\vartheta} | \mathbf{z})} d\boldsymbol{\vartheta} + \int_{\Omega} p(\boldsymbol{\vartheta} | \mathbf{z}) \log \frac{p(\boldsymbol{\vartheta} | \mathbf{z})}{p_N(\boldsymbol{\vartheta} | \mathbf{z})} d\boldsymbol{\vartheta} \\ &= \int_{\Omega} (p_N(\boldsymbol{\vartheta} | \mathbf{z}) - p(\boldsymbol{\vartheta} | \mathbf{z})) \log \left( \frac{\gamma}{\gamma_N} \frac{p_N(\mathbf{z} | \boldsymbol{\vartheta})}{p(\mathbf{z} | \boldsymbol{\vartheta})} \right) d\boldsymbol{\vartheta}, \end{aligned}$$

with  $\gamma$  and  $\gamma_N$  according to (33). By taking the absolute value of the integral, we can bound it using the  $\infty$ -norm as follows.

$$\begin{aligned} D_{\text{KL}}(p(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} | \mathbf{z})) &\leq \int_{\Omega} |p_N(\boldsymbol{\vartheta} | \mathbf{z}) - p(\boldsymbol{\vartheta} | \mathbf{z})| \left| \log \left( \frac{\gamma}{\gamma_N} \frac{p_N(\mathbf{z} | \boldsymbol{\vartheta})}{p(\mathbf{z} | \boldsymbol{\vartheta})} \right) \right| d\boldsymbol{\vartheta} \\ &\leq \left\| \log \left( \frac{\gamma}{\gamma_N} \frac{p_N(\mathbf{z} | \boldsymbol{\vartheta})}{p(\mathbf{z} | \boldsymbol{\vartheta})} \right) \right\|_{\infty} \int_{\Omega} |p_N(\boldsymbol{\vartheta} | \mathbf{z}) - p(\boldsymbol{\vartheta} | \mathbf{z})| d\boldsymbol{\vartheta}. \end{aligned}$$

Then, by working out the first formula we obtain:

$$\begin{aligned} D_{\text{KL}}(p(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} | \mathbf{z})) &\leq \|\log(\gamma) - \log(\gamma_N) + \log(p_N(\mathbf{z} | \boldsymbol{\vartheta})) - \log(p(\mathbf{z} | \boldsymbol{\vartheta}))\|_{\infty} \\ &\quad \cdot \int_{\Omega} |p_N(\mathbf{z} | \boldsymbol{\vartheta}) - p(\mathbf{z} | \boldsymbol{\vartheta})| p(\boldsymbol{\vartheta}) d\boldsymbol{\vartheta}. \end{aligned}$$

By application of the triangle inequality the first term can be bounded. By the boundedness of  $\Omega$ , the latter term can be bounded. Both yield  $\infty$ -norms as follows:

$$\begin{aligned} D_{\text{KL}}(p(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} | \mathbf{z})) &\leq (\|\log(\gamma) - \log(\gamma_N)\|_{\infty} + \|\log(p_N(\mathbf{z} | \boldsymbol{\vartheta})) - \log(p(\mathbf{z} | \boldsymbol{\vartheta}))\|_{\infty}) \\ &\quad \cdot \|p_N(\mathbf{z} | \boldsymbol{\vartheta}) - p(\mathbf{z} | \boldsymbol{\vartheta})\|_{\infty}. \end{aligned}$$

As  $\gamma > 0$ , the first term can be bounded using the Lipschitz continuity of the logarithm. Moreover,  $\Omega$  is compact, hence there exists an  $A > 0$  such that  $p(\mathbf{z} | \boldsymbol{\vartheta}) > A$  with  $\boldsymbol{\vartheta} \in \Omega$ . Therefore the second term can also be bounded using the Lipschitz continuity of the logarithm. The last term is already in the right format, and we obtain

$$D_{\text{KL}}(p(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} | \mathbf{z})) \leq \frac{1}{|\log A|} (\|\gamma - \gamma_N\|_{\infty} + \|p_N(\mathbf{z} | \boldsymbol{\vartheta}) - p(\mathbf{z} | \boldsymbol{\vartheta})\|_{\infty}) \|p_N(\mathbf{z} | \boldsymbol{\vartheta}) - p(\mathbf{z} | \boldsymbol{\vartheta})\|_{\infty}.$$

Finally, the result readily follows:

$$D_{\text{KL}}(p(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p_N(\boldsymbol{\vartheta} | \mathbf{z})) \leq (C_1 N^{-\alpha} + C_2 N^{-\alpha}) C_2 N^{-\alpha} = KN^{-2\alpha}. \quad \square$$

Note that in a similar manner it can be proved that  $D_{\text{KL}}(p_N(\boldsymbol{\vartheta} | \mathbf{z}) \parallel p(\boldsymbol{\vartheta} | \mathbf{z})) \leq KN^{-2\alpha}$ .

## 4 Numerical results

Three numerical testcases are employed to show the performance of our methodology. First, in Section 4.1 two explicit testcases are studied, which are cases where an expression for  $u$  is known that can be evaluated accurately such that the exact posterior can be determined explicitly. We use these cases to verify the theoretical properties that have been derived in Section 3. For sake of comparison, these cases are studied using interpolation based on Clenshaw–Curtis nodes, Leja nodes, and the proposed adaptive weighted Leja nodes.

In Section 4.2, we study calibration of the one-dimensional Burgers’ equation. As an explicit solution is not used here, we can show the practical purposes of the interpolation procedure to problems that are

defined implicitly. We determine the interpolant using Clenshaw–Curtis nodes and weighted Leja nodes. The last test case consists of the calibration of the closure parameters of the Spalart–Allmaras turbulence model. Here, a single evaluation of the likelihood is computationally expensive as it requires the numerical solution of the Navier–Stokes equations. In this case straightforward methods (such as MCMC) become intractable and therefore we will only study weighted Leja nodes.

Note that the quantity of interest in each case is the posterior and not the model. Therefore we mainly study the convergence in the posterior and to a lesser extent the convergence in the model.

## 4.1 Explicit testcases

We consider two cases: an analytic (hence smooth) function and a non-analytic smooth function. Both functions are defined for any dimension  $d$ .

### 4.1.1 Analytic function

A well-known class of analytic functions is formed by Gaussian functions. We will use the following function to represent the model:

$$u_d : \mathbb{R}^d \rightarrow \mathbb{R}, \text{ with } u_d(\boldsymbol{\vartheta}) := \exp\left(-\sum_{k=1}^d \left(\vartheta_k - \frac{1}{2}\right)^2\right). \quad (39)$$

This function is a composition of the exponential function and a polynomial, which are both globally analytic. Hence also this function is globally analytic and can therefore be approximated well using polynomials. Consequently, any nodal set can be used to interpolate this function—even an equidistant set—so we use this test case merely for a sanity check of the procedure and the theory.

Two statistical models are employed to demonstrate the independence of our procedure from the likelihood. The first is the statistical model discussed before, i.e.  $z_k = u_d(\boldsymbol{\vartheta}) + \varepsilon_k$ , with  $\varepsilon \sim \mathcal{N}(\mathbf{0}, \Sigma)$ ,  $\Sigma = \sigma^2 I$ , and  $\sigma = 1/10$ . As discussed before, the likelihood equals

$$p_{\mathcal{N}}(\mathbf{z} \mid \boldsymbol{\vartheta}) \propto \exp\left[-\frac{1}{2}(\mathbf{z} - u_d(\boldsymbol{\vartheta}))^T \Sigma^{-1}(\mathbf{z} - u_d(\boldsymbol{\vartheta}))\right], \quad (40)$$

where  $\mathbf{z}$  is the vector containing the data. A data vector of  $n = 20$  elements is generated by sampling from a Gaussian distribution with mean  $u_d(1/4)$  and standard deviation  $\sigma$ . The subscript  $\mathcal{N}$  refers to multivariate normal. For the second model, we do not write an explicit relation between the data and the model, but only impose the following likelihood:

$$p_{\beta}(\mathbf{z} \mid \boldsymbol{\vartheta}) \propto \begin{cases} (1-d)^2(1+d)^2 & \text{if } |d| < 1 \\ 0 & \text{otherwise} \end{cases}, \text{ with } d = \frac{\bar{z} - u_d(\boldsymbol{\vartheta})}{\bar{z}} \text{ and } \bar{z} = \frac{1}{n} \sum_{k=1}^n z_k. \quad (41)$$

We call this likelihood the Beta likelihood (denoted with the  $\beta$ ), which we use because it has different characteristics than the Gaussian likelihood. As the standard deviation is significantly larger in the second case, the posteriors considerably differ. Both likelihoods are continuously differentiable, so we expect similar accuracy when applying the proposed algorithm. In both cases the prior is assumed to be uniform on the domain  $[0, 1]^d$ .

Because the model under consideration is analytic, the value of  $\zeta$  is not very important for the accuracy of the interpolation procedure (even  $\zeta = 0$  works well in this case). We choose  $\zeta = 10^{-3}$ , because then convergence of the posterior can be observed well, which is shown in Figure 6 for  $d = 3$ . It is clearly visible that for the multivariate normal case, the nodes are placed more in the center of the domain (see for comparison Figure 2). This is also true for the second case, but less apparent due to the less intuitive structure of the posterior. The asymmetry between dimensions occurs due to the interpolation with Leja nodes, which are asymmetric by construction.

If we restrict ourselves to a one-dimensional case, the convergence of our algorithm can be assessed with high accuracy as it is possible to determine the Kullback–Leibler divergence with high accuracy using

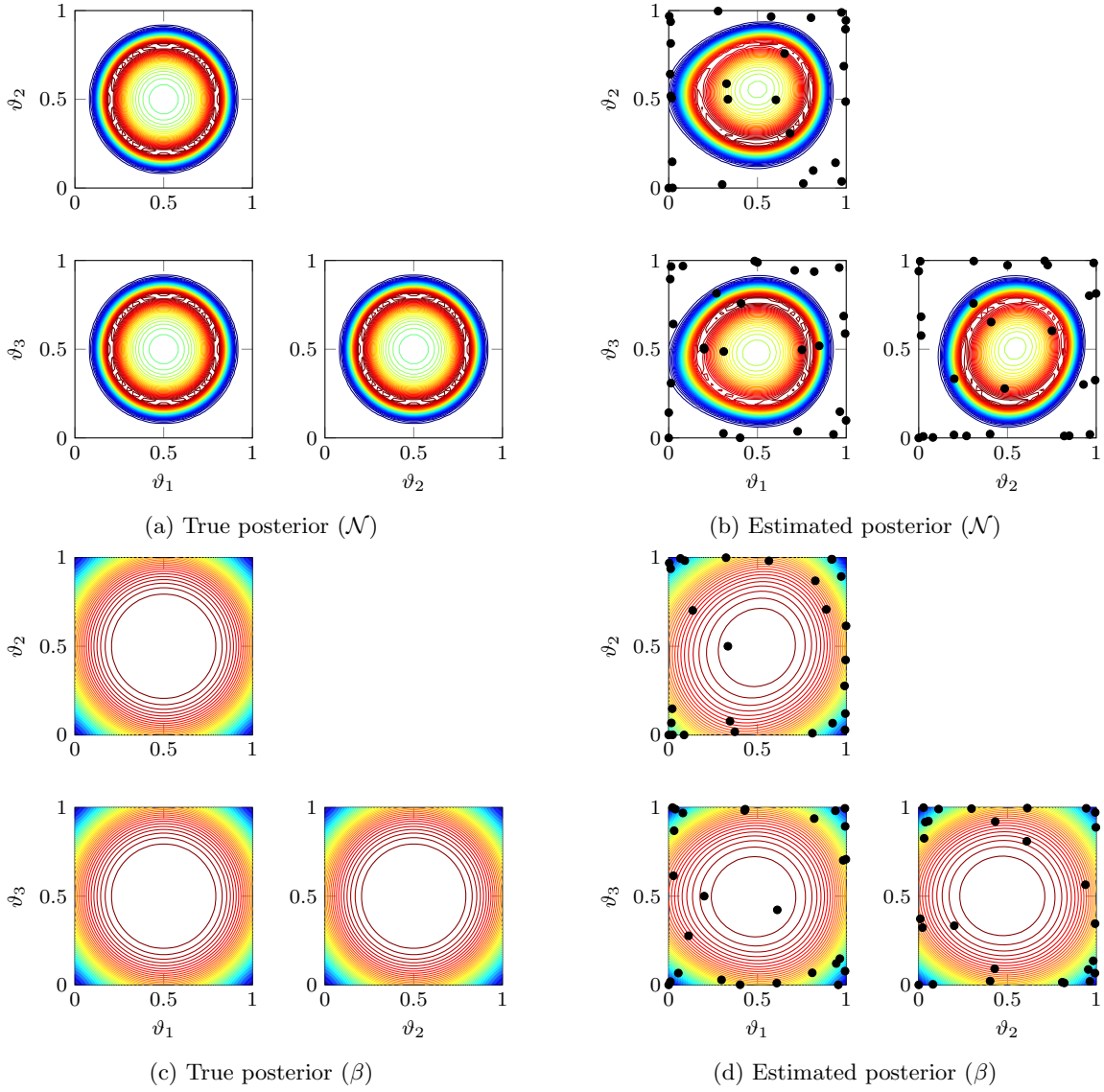


Figure 6: True and estimated posteriors with the analytic Gauss function as model using the multivariate normal ( $\mathcal{N}$ ) likelihood and the Beta ( $\beta$ ) likelihood with 25 Leja nodes. Red is high, blue is low.

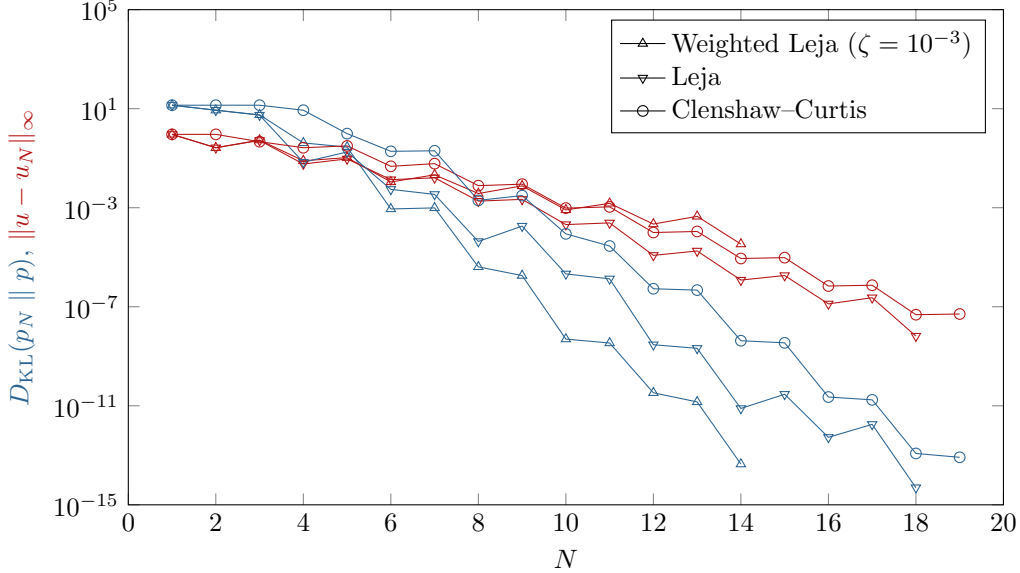


Figure 7: Convergence for the Gaussian likelihood using the Gauss test function. The Kullback–Leibler divergence of the estimated posterior with respect to the true posterior is depicted in blue and the  $\infty$ -norm of the difference between the interpolant and the true model is depicted in red.

a quadrature rule. Moreover, a comparison with an interpolant based on Clenshaw–Curtis nodes can be performed.

The Kullback–Leibler divergence cannot be determined for the Beta likelihood. This is because two interpolants  $u_{N_1}$  and  $u_{N_2}$  in general do not have  $|d| < 1$  exactly at similar  $\vartheta$ . Therefore the set where one model has  $d = 0$  and the other  $d > 0$  (and vice versa) is measurable, rendering the Kullback–Leibler divergence unbounded.

The Kullback–Leibler divergence is determined using posteriors constructed with weighted Leja nodes, unweighted Leja nodes, and the Clenshaw–Curtis nodes (see Figure 7). All three nodal sets yield a model and a posterior that converge to respectively the true model and true posterior. The doubling of the convergence rate, according to Theorem 7, already becomes apparent for the small number of nodes used and it is clearly visible that the weighted Leja nodes are mainly constructed for accuracy of the posterior instead of the model. The weighted Leja nodes show the fastest convergence, but the difference of model evaluations necessary to reach a certain accuracy level is small as the model under consideration is analytic.

#### 4.1.2 Non-analytic function

A well-known test case for interpolation methods is Runge’s function, defined (up to a constant) in the domain  $[0, 1]^d$  as follows:

$$u_d : [0, 1]^d \rightarrow \mathbb{R}, \text{ with } u_d(\mathbf{x}) = \frac{5}{2 + 50 \sum_{k=1}^d (x_k - \frac{1}{2})^2}. \quad (42)$$

This function is infinitely smooth, i.e. all derivatives exist and are continuous. However, its Taylor series has small radius of convergence. Using a nodal set with an exponentially growing Lebesgue constant yields a sequence of interpolants that diverge, which is well-known and typically called Runge’s phenomenon. The Gaussian statistical model, the uniform prior, and the true value  $\vartheta \equiv \frac{1}{4}$  are reconsidered for this test case. Results in terms of convergence using the three different nodal sets discussed previously are shown in Figure 8 (again  $d = 1$ ).

Differences between the nodal sets are more apparent than in the smooth test case. Again, the weighted Leja nodes are mainly constructed to obtain an accurate posterior and the accuracy of the interpolant is less important. It is clear that the weighted Leja nodes outperform the other nodal sets in the Kullback–Leibler



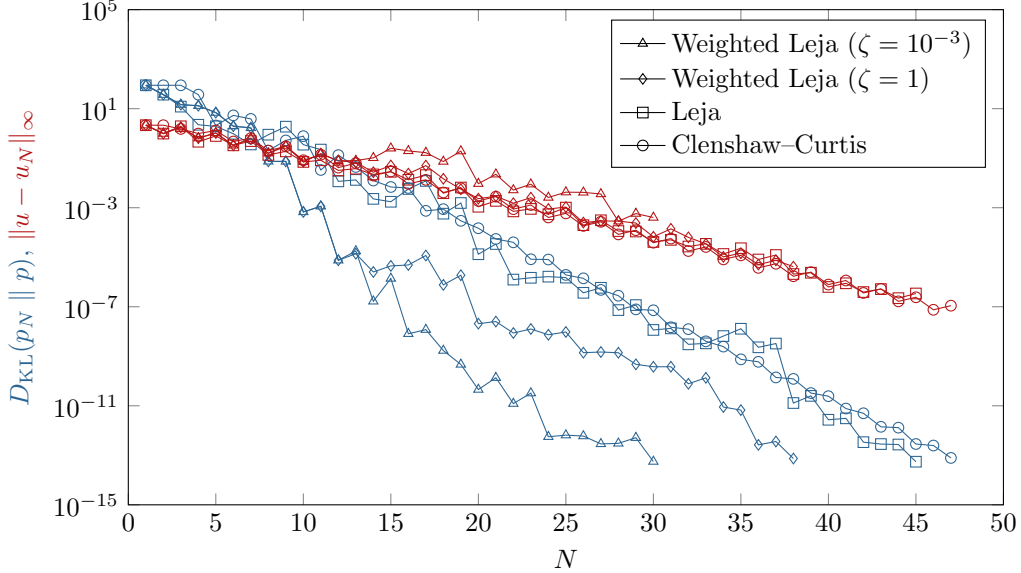


Figure 8: Convergence of the model (in red) and convergence of the Kullback–Leibler divergence (in blue) using four sampling algorithms for Runge’s function.

norm. The unweighted Leja nodes and Clenshaw–Curtis nodes perform equally well, which is surprising as the Leja nodes do not have a logarithmically growing Lebesgue constant (contrary to the Clenshaw–Curtis nodes). Figure 8 also shows that upon increasing the value of  $\zeta$  from  $10^{-3}$  to 1, the convergence rate of the weighted Leja sequence slightly decreases, as expected from theory. For small  $N$ , the posterior dominates the weighting function in Leja nodes, while for large  $N$  the effect of  $\zeta$  becomes important.

## 4.2 Burgers’ equation

In this section the Burgers’ equation is considered where the boundary condition is unknown, extending the example from Marzouk and Xiu [23]. The one-dimensional steady state Burgers’ equation for a solution field  $y(x)$  and diffusion coefficient  $\nu$  is stated as follows:

$$0 = \nu \frac{\partial^2 y}{\partial x^2} - y \frac{\partial y}{\partial x}, \quad (43)$$

where boundary conditions complete the system. In this section, we will consider the equation on an interval  $[-1, 1]$  with  $\nu = 0.1$  and use the boundary conditions  $y(-1) = 1 + \delta$  and  $y(1) = -1$ , where  $\delta$  is unknown.

The inverse problem is as follows. Let  $x_0$  be the zero-crossing of the solution  $y$ , i.e.  $y(x_0) = 0$ . Given a vector of noisy observations of  $x_0$ , we would like to obtain the value of  $\delta$ , i.e. reconstruct the boundary conditions from a specific value of the function. In the notation used so far, we would have a function  $u : \mathbb{R} \rightarrow \mathbb{R}$  with  $u(\delta) = x_0$ . In the current setting, this function is only defined implicitly.

Let  $\mathbf{z}$  be a vector with  $n_d = 20$  noisy observations of  $x_0$ . We generate this vector by sampling  $\delta$  from a normal distribution with mean 0.05 and standard deviation  $\sigma = 0.05$  and determining the corresponding values of  $x_0$ . We use a uniform prior such that  $\delta \sim U[0, 0.1]$ . The likelihood is Gaussian with zero mean and standard deviation  $\sigma$ , i.e. for each “measurement”  $z_k$  (for  $k = 1, \dots, n_d$ ) we have

$$z_k - x_0(\delta) \sim \mathcal{N}(0, \sigma^2). \quad (44)$$

A second-order finite volume scheme is employed to numerically solve the Burgers’ equation equipped with a uniform mesh of  $10^5$  cells. The zero-crossing is determined by piecewise interpolation of the solution.

We use Clenshaw–Curtis, Leja, and weighted Leja nodes to compute the posterior. In all three cases, the interpolant was refined until the Kullback–Leibler divergence of the estimated posterior with respect to the true posterior (computed using a fine quadrature rule) is smaller than  $10^{-7}$ . In Figure 9 a sketch of

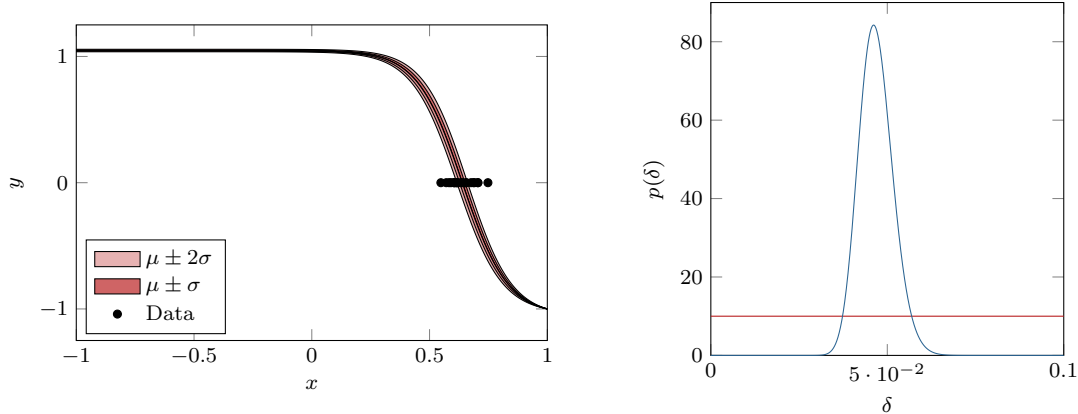


Figure 9: *Left:* Propagation of the posterior through the Burgers' equations. *Right:* The prior (in red) and the posterior (blue) of the Burgers' equation calibration test case.

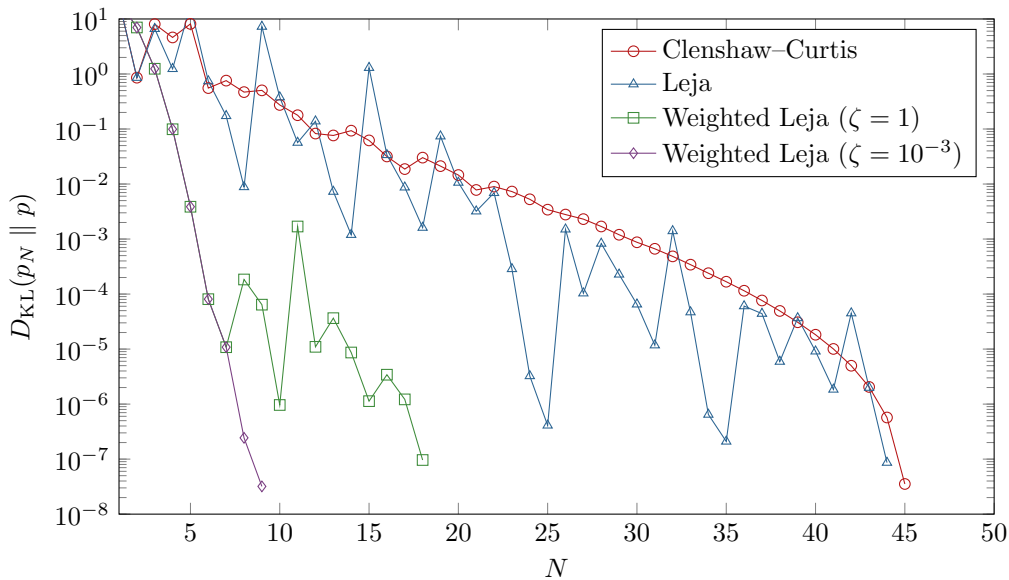


Figure 10: Convergence of the posterior of Burgers' equation using three different sampling procedures.

the distribution of  $u$  after propagation of this posterior and the exact posterior is depicted. The standard deviation of the output variable  $u$  is small because we only take measurement errors into account (hence the standard deviation decreases for larger data sets). The posteriors determined after convergence of the three nodal sequences did not differ visually from the true posterior, so we do not present them separately.

There are large differences in the number of nodes that are necessary to obtain convergence, see Figure 10. For the posteriors determined using Clenshaw-Curtis nodes and Leja nodes, approximately 45 nodes are necessary. For the weighted Leja nodes, just 18 nodes are necessary for  $\zeta = 1$  and only 9 nodes are necessary for  $\zeta = 10^{-3}$ . This significant improvement is because the initial nodes already provide a good approximation of the posterior, which is leveraged when  $\zeta$  is small. Again it is clearly visible that for small  $N$ , the posterior dominates the weighting function, while for large  $N$  the value of  $\zeta$  dominates. In both cases the weighted Leja nodes clearly outperform the other nodal sets.

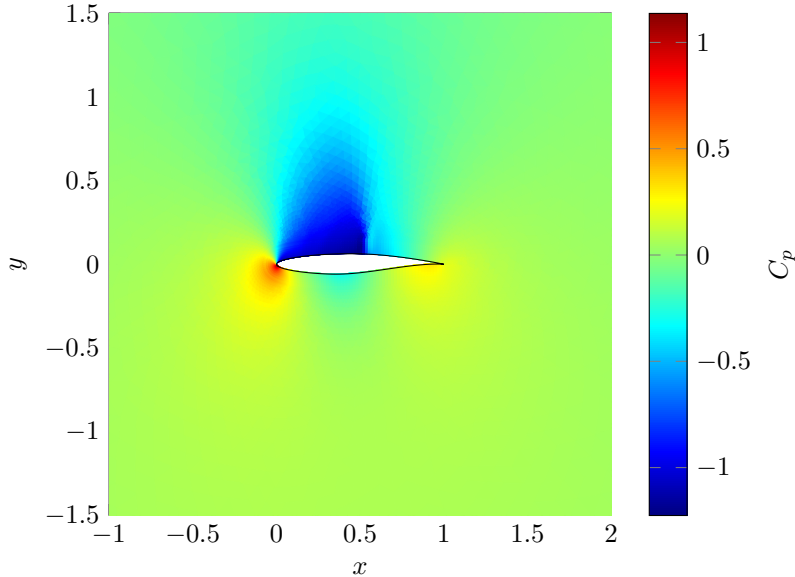


Figure 11: The pressure coefficient around the RAE2822 airfoil using the canonical turbulence coefficients.

### 4.3 Turbulence model closure parameters

In this section we consider the flow around an airfoil (i.e. the two-dimensional cross section of a wing). The airfoil under consideration is the RAE2822, because extensive publicly available measurements have been performed for this particular airfoil. We use the wind tunnel measurements for the pressure coefficient from Cook [8] and study Case 6, i.e. the angle of attack equals  $2.92^\circ$ , the Mach number equals 0.725, and the Reynolds number equals  $6.5 \cdot 10^6$ . These parameters are not corrected for wind tunnel effects, which is necessary for comparison with numerical simulations. See Slater et al. [33] for more information about such a correction procedure and Cook et al. [8] for more information about the measurement data under consideration, as these details are out of the scope of this article. The transonic flow around the airfoil in this case is depicted in Figure 11, determined numerically with the canonical turbulence coefficients of the Spalart–Allmaras turbulence model. It is clearly visible that there is shock formation on the upper side of the airfoil.

#### 4.3.1 Problem description

The flow around an airfoil is often modeled using the Navier–Stokes equations. Due to the large range of scales present in the solution of these equations, typically the Reynolds-averaged Navier–Stokes (RANS) equations are employed. The details are out of the scope of this article, we refer the interested reader to Wilcox [39]. The RANS equations do not form a closed set of equations and require a closure model. Typically, the Boussinesq hypothesis is used, which introduces an eddy viscosity. This viscosity models the effect of turbulence in the flow and requires an additional set of equations called a turbulence model. Throughout the years, many different turbulence models have been developed. These models have fitting parameters, chosen such that the model represents idealized test cases well. Often the coefficients are calibrated in a non-systematic way (e.g. by hand). A turbulence model commonly used for flow over airfoils is the Spalart–Allmaras (SA) turbulence model, which consists of a single equation modeling the transport, production, and dissipation of the eddy viscosity [34].

Several forms of this model exist. The original SA turbulence model has 10 constants, typically defined as follows:

$\sigma$	2/3	$C_{b1}$	0.1355	$C_{b2}$	0.622
$\kappa$	0.41	$C_{w2}$	0.3	$C_{w3}$	2.0
$C_{t3}$	1.2	$C_{t4}$	0.5	$C_{v1}$	7.1

Here we omit  $C_{w1}$ , which is commonly defined as  $C_{w1} = C_{b1}/\kappa^2 + (1 + C_{b2})/\sigma$ .

Although tested extensively, straightforward physical derivations do not exist for many of the parameter values above. In this section, we will calibrate these parameters systematically using Bayesian model calibration and apply the weighted Leja nodes proposed in this article. We will employ  $\zeta = 1$ , to make sure that we have convergence without spurious oscillations. Results on the calibration of turbulence parameters exist in literature (e.g. [7, 10]), where it is customary to calibrate the parameters using MCMC methods.

We will employ SU2 to numerically solve the RANS equations. SU2 is a second-order finite-volume computational fluid dynamics solver with support for the Spalart-Allmaras turbulence model. We have made a few minor modifications to allow the turbulence parameters to be changeable using configuration files. SU2 has been tested extensively to the airfoil test case (for the canonical turbulence parameters) with these options, see [28].

### 4.3.2 Statistical model

We consider the same parameters for calibration as Edeling et al. [10] and Cheung et al. [7], namely  $\boldsymbol{\vartheta} = (\kappa, \sigma, C_{b1}, C_{b2}, C_{v1}, C_{w2}, C_{w3})^T$ . The remaining parameters are fixed at their canonical values.

We are now in the position to state the statistical model. Let  $x$  be the spatial parameter along the contour of the airfoil, i.e.  $x$  parametrizes the airfoil such that each  $x$  has a unique value for the pressure coefficient. Suppose the data (i.e. a vector of pressure coefficients)  $\mathbf{z} = (z_1, \dots, z_n)^T$  is provided at locations  $\mathbf{x} = (x_1, \dots, x_n)^T$  on the airfoil. Then the statistical model under consideration is as follows:

$$\begin{aligned} v(\boldsymbol{\vartheta}; x_k) &= u(\boldsymbol{\vartheta}; x_k) + \delta(x_k), \\ z(x_k) &= v(\boldsymbol{\vartheta}; x_k) + \varepsilon_k, \end{aligned} \tag{45}$$

where  $\delta(x) \sim \mathcal{N}(0, f(x, x'))$  is a Gaussian process with mean 0 and squared exponential covariance function

$$f(x, x') = A \exp \left[ -\frac{|x - x'|^2}{2l^2} \right]. \tag{46}$$

Moreover, we have all  $\varepsilon_k$  independently and identically normally distributed with mean zero and standard deviation  $\tilde{\sigma}$ , which is defined explicitly later (we use  $\tilde{\sigma}$  to overcome confusion with one of the turbulence coefficients). The measurement data from Cook et al. [8] has  $n = 103$  measurement locations along the surface of the airfoil.

We choose the model such that  $v$  is the “true” process that is modeled by  $u$ .  $\delta$  and  $\varepsilon_k$  are the model and measurement error respectively. Both require an estimation based on knowledge of the model and the data, as our calibration framework currently does not include hyperparameters. Bounds for the measurement error are provided in Cook et al. [8], resulting in  $\tilde{\sigma} \approx 0.01$ . This value is larger than the error of the measurement data, thus incorporating potential additional errors. The parameters of the Gaussian process are chosen to be  $A = 0.6$  and  $l = 0.2$ , based on the largest error that we expect (which is approximately 0.5, based on the strength of the shock on top of the airfoil) and the distance between two measurement locations.

### 4.3.3 Prior and likelihood

The prior is assumed to be multivariate uniformly distributed with mean equal to the canonical values of the parameters and a variation of 50%. The support of the prior is therefore as follows:

$\kappa$	[0.205, 0.615]
$\sigma$	[1/3, 1]
$C_{b1}$	[0.0678, 0.2033]
$C_{b2}$	[0.311, 0.933]
$C_{v1}$	[3.55, 10.65]
$C_{w2}$	[0.2, 0.4]
$C_{w3}$	[1, 3]

The likelihood readily follows from the statistical model:

$$p(\mathbf{z} | \boldsymbol{\vartheta}) \propto \exp \left[ -\frac{1}{2} \mathbf{d}^T C^{-1} \mathbf{d} \right], \tag{47}$$

with  $\mathbf{d}$  the misfit and  $C$  the covariance, i.e.

$$d_k = z(x_k) - u(\boldsymbol{\vartheta}; x_k), \quad (48)$$

$$C = \Sigma + C_f, \text{ with } C_f^{[i,j]} = f(x_i, x_j) \text{ and } \Sigma = \bar{\sigma}I. \quad (49)$$

Finally, the posterior is formed by a multiplication of the likelihood with the prior in the usual way.

#### 4.3.4 Results

We applied calibration with weighted Leja nodes to the current problem and used 100 nodes. The convergence was assessed using cross-validation between two consecutive interpolants, up to the point that visually no difference can be observed in the posterior. The one- and two-dimensional marginals from the obtained posterior are depicted in Figure 12. We cannot compare the result with the true posterior, as we did for the previous two test cases, so instead we compare the posterior with results obtained in literature.

The results are similar to Cheung et al. [7], who calibrated the same set of parameters using MCMC using more than 30 000 samples. In their study, the parameters  $\kappa$  and  $C_{v1}$  are informed best, which is also clearly visible in Figure 12. In our results, the parameter  $C_{b1}$  is also informed in comparison with the other parameters, but this is not the case in the study from Cheung et al. There can be various reasons for this, among others the fact that Cheung et al. used a flat plate test case for the calibration and used the skin friction coefficient as data.

In the study from Edeling et al. [10] a similar test case is performed. The parameters  $\kappa$  and  $C_{v1}$  have the highest Sobol' indices and are therefore most influential on the model output (which is the pressure coefficient in this case). This is consistent with our study and the study from Cheung et al. The results for  $C_{b1}$  again differ, which we attribute to the fact that Edeling et al. also used the skin friction coefficient and the flat plate test case.

To conclude, the proposed method is capable of constructing a good approximate posterior using a fraction of the 30 000 model evaluations that were necessary in the MCMC case. The posterior compares very well with both studies. By applying MCMC to the constructed surrogate, we can propagate the posterior and make a prediction under uncertainty. To illustrate this, the obtained uncertainty bound on the pressure coefficient is depicted in Figure 13. It is clearly visible that the largest uncertainty is near the shock on top of the airfoil.

## 5 Conclusion

Bayesian model calibration is an attractive approach for calibrating model parameters of complex physical models. It is customary to sample the posterior using MCMC methods, but these are only tractable if the model under consideration can be evaluated rapidly. We consider problems where this is not the case, such as the calibration of turbulence closure coefficients.

Our proposed method consists of replacing the computationally expensive model with an interpolating surrogate model. The interpolant is determined adaptively using weighted Leja nodes, where the weighting function directly uses the posterior and is changed with each iteration. We have proposed a formulation in which the weighting function consists of two parts: a part incorporating the currently available posterior and a part incorporating the accuracy of the interpolant. The balance between these two can be changed adaptively by the parameter  $\zeta$  and convergence is guaranteed for any  $\zeta > 0$ . The “best” value of  $\zeta$  depends on the specifics of the model, the likelihood, and the posterior, and can be changed adaptively during the calibration procedure. Compared to conventional nodal sets (such as Clenshaw–Curtis nodes), we obtain more accurate results with less nodes.

Theoretically we have proved that if the interpolant converges to the true model in the  $\infty$ -norm, then the estimated posterior converges to the true posterior in the  $\infty$ -norm with a similar rate. Under mild conditions, the Kullback–Leibler divergence between the estimated and exact posterior converges with a doubled rate.

The three conducted numerical experiments confirm these theoretical findings: if the interpolant converges to the model, the Kullback–Leibler divergence converges to zero with doubled rate. For the explicit numerical test cases, this doubled convergence in the Kullback–Leibler divergence was clearly visible. The calibration of the Burgers’ equation shows that the approach also works well for models that are defined implicitly.

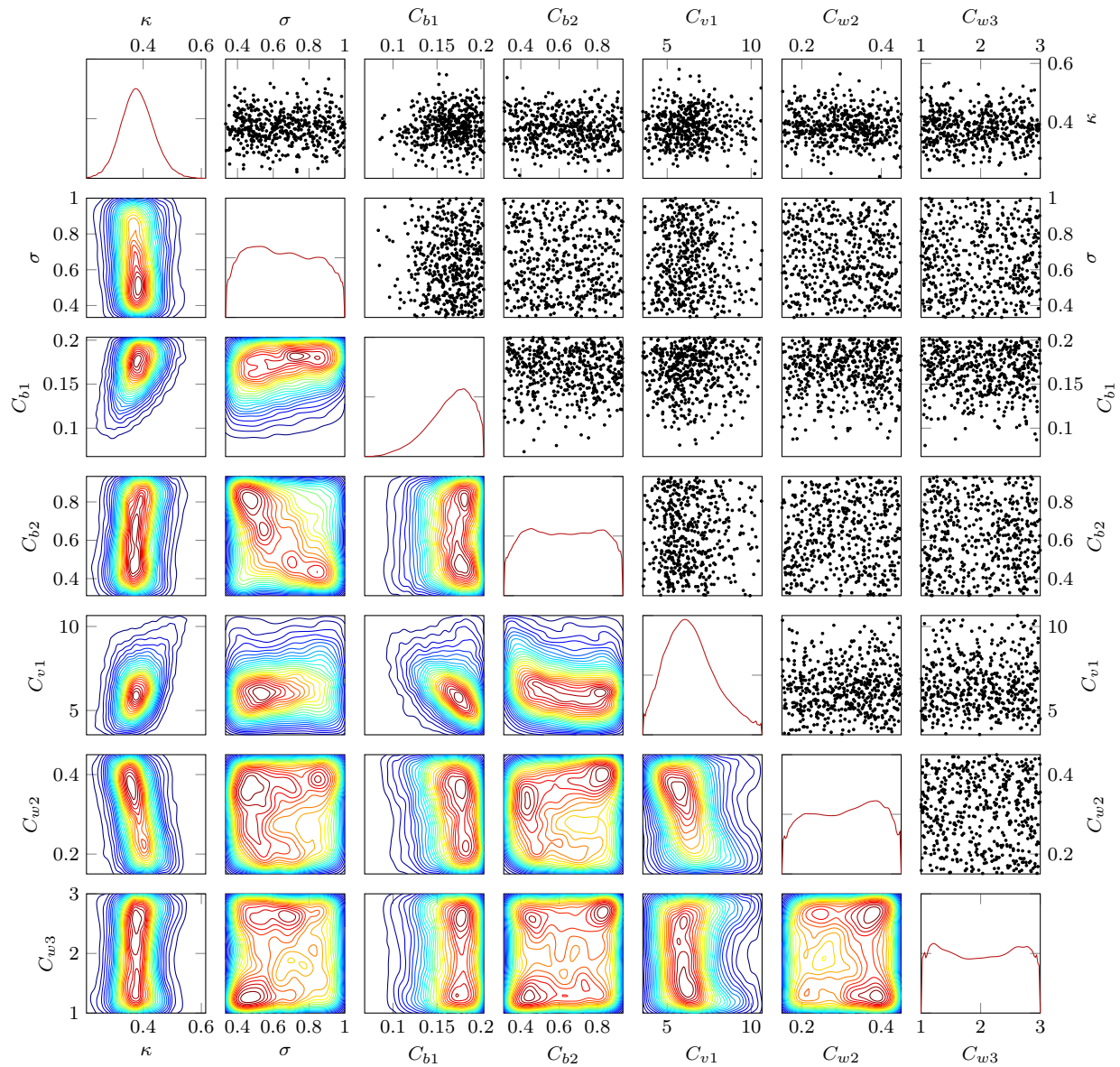


Figure 12: The marginals of the posterior under consideration of the RAE2822 airfoil case. Red is high, blue is low.

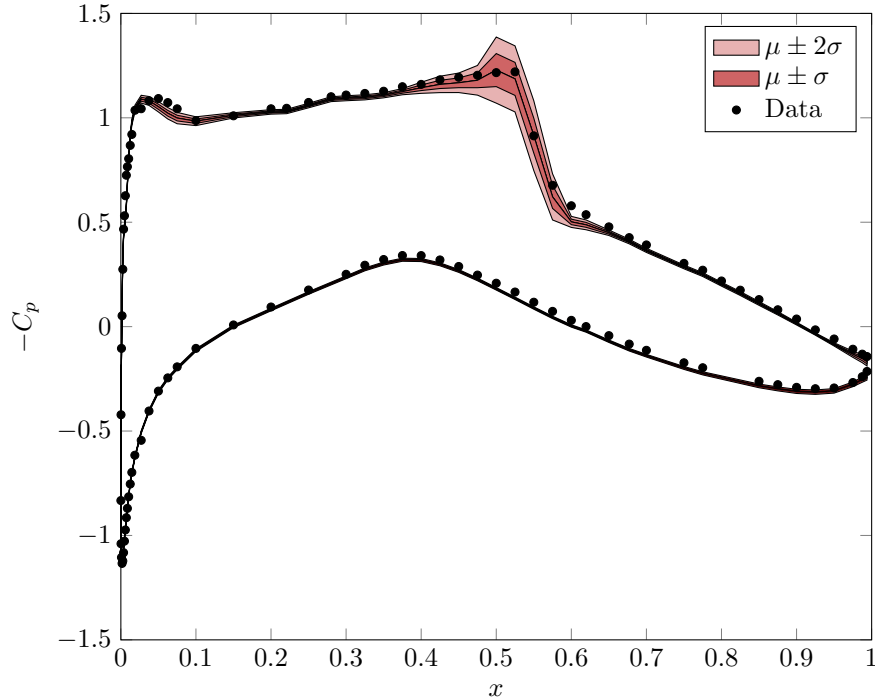


Figure 13: Pressure coefficient along the airfoil determined by propagation of the posterior through the interpolating surrogate. The standard deviation is shaded in red.

Finally, calibration of turbulence closure parameters has been discussed, showing that the approach is truly applicable to computationally expensive models. We have compared the results from our calibration procedure with results conducted using Monte Carlo methods and the posteriors shows good resemblance, at a highly reduced computational cost.

## 5.1 Future work

Firstly, there are some limitations about the statistical model that can be used in our approach. We require that it can be written as a function of the model and no hyperparameters are used in the statistical model. Extending the current approach to such a setting is an important step, as hyperparameters are often employed in Bayesian model calibration.

Secondly, we believe the value of  $\zeta$  should depend on the specifics of the model and further research is required to find the optimal value a priori or either dynamically during the procedure.

## 6 Acknowledgments

We thank Richard Dwight for his valuable remarks on an early draft of this article. This research is part of the Dutch EUROS programme, which is supported by NWO domain Applied and Engineering Sciences and partly funded by the Dutch Ministry of Economic Affairs.

## References

- [1] G. N. Absi and S. Mahadevan. Calibration of system parameters under model uncertainty. In H. S. Atamturktur, B. Moaveni, C. Papadimitriou, and T. Schoenherr, editors, *Model Validation and Uncertainty Quantification*, volume 3, pages 1–10. Springer International Publishing, 2014. doi: 10.1007/978-3-319-04552-8\_1.

- [2] M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.
- [3] J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange interpolation. *SIAM Review*, 46(3):501–517, 2004. doi:10.1137/s0036144502417715.
- [4] A. Birolleau, G. Poëtte, and D. Lucor. Adaptive Bayesian inference for discontinuous inverse problems, application to hyperbolic conservation laws. *Communications in Computational Physics*, 16(1):1–34, 2014. doi:10.4208/cicp.240113.071113a.
- [5] J.-P. Calvi and M. Phung Van. On the Lebesgue constant of Leja sequences for the unit disk and its applications to multivariate interpolation. *Journal of Approximation Theory*, 163(5):608–622, 2011. doi:10.1016/j.jat.2011.02.001.
- [6] D. Chauveau and P. Vandekerckhove. Improving convergence of the Hastings–Metropolis algorithm with an adaptive proposal. *Scandinavian Journal of Statistics*, 29(1):13–29, 2002. doi:10.1111/1467-9469.00064.
- [7] S. H. Cheung, T. A. Oliver, E. E. Prudencio, S. Prudhomme, and R. D. Moser. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliability Engineering & System Safety*, 96(9):1137–1149, 2011. doi:10.1016/j.res.2010.09.013.
- [8] P. H. Cook, M. A. McDonald, and M. C. P. Firmin. Aerofoil RAE 2822 – pressure distributions, and boundary layer and wake measurements. In *Experimental Data Base for Computer Program Assessment*, number 138 in AGARD Advisory Report, chapter 6, pages A6–1 – A6–77. North Atlantic Treaty Organization, 1979.
- [9] K. Csilléry, M. G. B. Blum, O. E. Gaggiotti, and O. François. Approximate Bayesian computation (ABC) in practice. *Trends in Ecology & Evolution*, 25(7):410–418, 2010. doi:10.1016/j.tree.2010.04.001.
- [10] W. N. Edeling, P. Cinnella, and R. P. Dwight. Predictive RANS simulations via Bayesian model-scenario averaging. *Journal of Computational Physics*, 275:65–91, 2014. doi:10.1016/j.jcp.2014.06.052.
- [11] W. N. Edeling, P. Cinnella, R. P. Dwight, and H. Bijl. Bayesian estimates of parameter variability in the  $k - \epsilon$  turbulence model. *Journal of Computational Physics*, 258:73–94, 2014. doi:10.1016/j.jcp.2013.10.027.
- [12] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, third edition, 2013.
- [13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996. ISBN 0-8018-5413-X.
- [14] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi:10.1093/biomet/57.1.97.
- [15] N. J. Higham. The numerical stability of barycentric Lagrange interpolation. *IMA Journal of Numerical Analysis*, 24(4):547–556, 2004. doi:10.1093/imanum/24.4.547.
- [16] B. A. Ibrahimoglu. Lebesgue functions and Lebesgue constants in polynomial interpolation. *Journal of Inequalities and Applications*, 2016(1):93, 2016. doi:10.1186/s13660-016-1030-3.
- [17] D. Jackson. *Theory of Approximation (Colloquium Publications)*. American Mathematical Society, 1982. ISBN 978-0821810118.
- [18] P. Jantsch, C. G. Webster, and G. Zhang. On the Lebesgue constant of weighted Leja points for Lagrange interpolation on unbounded domains. *arXiv preprint arXiv:1606.07093*, 2016.
- [19] M. C. Kennedy and A. O’Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001. doi:10.1111/1467-9868.00294.



- [20] J. S. Lopes and M. A. Beaumont. ABC: A useful Bayesian tool for the analysis of population data. *Infection, Genetics and Evolution*, 10(6):825–832, 2010. doi:10.1016/j.meegid.2009.10.010.
- [21] F. Lu, M. Morzfeld, X. Tu, and A. J. Chorin. Limitations of polynomial chaos expansions in the Bayesian solution of inverse problems. *Journal of Computational Physics*, 282:138–147, 2015. doi:10.1016/j.jcp.2014.11.010.
- [22] X. Ma and N. Zabaras. An efficient Bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method. *Inverse Problems*, 25(3):035013, 2009. doi:10.1088/0266-5611/25/3/035013.
- [23] Y. Marzouk and D. Xiu. A stochastic collocation approach to Bayesian inference in inverse problems. *Communications in Computational Physics*, 6(4):826–847, 2009. doi:10.4208/cicp.2009.v6.p826.
- [24] Y. M. Marzouk, H. N. Najm, and L. A. Rahn. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics*, 224(2):560–586, 2007. doi:10.1016/j.jcp.2006.10.010.
- [25] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087, 1953. doi:10.1063/1.1699114.
- [26] A. Narayan and J. D. Jakeman. Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM Journal of Scientific Computing*, 36(6):A2952–A2983, 2014. doi:10.1137/140966368.
- [27] T. A. Oliver and R. D. Moser. Bayesian uncertainty quantification applied to RANS turbulence models. *Journal of Physics: Conference Series*, 318(4):042032, 2011. doi:10.1088/1742-6596/318/4/042032.
- [28] F. Palacios, T. D. Economou, A. Aranake, S. R. Copeland, A. K. Lonkar, T. W. Lukaczyk, D. E. Manosalvas, K. R. Naik, S. Padron, B. Tracey, A. Variyar, and J. J. Alonso. Stanford university unstructured (SU2): Analysis and design technology for turbulent flows. In *52<sup>nd</sup> Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2014. doi:10.2514/6.2014-0243.
- [29] E. Passow. Another proof of Jackson’s theorem. *Journal of Approximation Theory*, 3(2):146–148, 1970. doi:10.1016/0021-9045(70)90022-5.
- [30] S. Prudhomme and C. M. Bryant. Adaptive surrogate modeling for response surface approximations with application to Bayesian inference. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1), 2015. doi:10.1186/s40323-015-0045-5.
- [31] J. Ray, S. Lefantzi, S. Arunajatesan, and L. J. DeChant. Bayesian calibration of a RANS model with a complex response surface - a case study with jet-in-crossflow configuration. In *45<sup>th</sup> AIAA Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics (AIAA), 2015. doi:10.2514/6.2015-2784.
- [32] K. Sargsyan, H. N. Najm, and R. Ghanem. On the statistical calibration of physical models. *International Journal of Chemical Kinetics*, 47(4):246–276, 2015. doi:10.1002/kin.20906.
- [33] J. W. Slater, J. C. Dudek, and K. E. Tatum. The NPARC alliance verification and validation archive. Technical Report 2000-209946, NASA, 2000.
- [34] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30<sup>th</sup> Aerospace Sciences Meeting & Exhibit*, number 92-0439, pages 6–9, 1992. doi:10.2514/6.1992-439.
- [35] R. Taylor. *Lagrange Interpolation on Leja points*. phdthesis, University of South Florida, 2008.
- [36] R. Taylor and V. Totik. Lebesgue constants for Leja points. *IMA Journal of Numerical Analysis*, 30(2):462–486, 2010. doi:10.1093/imanum/drn082.

- [37] J. A. Vrugt, C. J. F. ter Braak, C. G. H. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon. Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3):273–290, 2009. doi:10.1515/ijnsns.2009.10.3.273.
- [38] G. A. Watson. *Approximation Theory and Numerical Methods*. John Wiley & Sons Ltd, 1980. ISBN 0471277061.
- [39] D. C. Wilcox. *Turbulence modeling for CFD*, volume 2. DCW industries La Cañada, CA, 1998.
- [40] D. J. Wilkinson and S. K. H. Yeung. Adaptive Metropolis–Hastings samplers for the Bayesian analysis of large linear Gaussian systems. *Computational Science and Statistics*, 33:128–138, 2002.
- [41] L. Zeng, L. Shi, D. Zhang, and L. Wu. A sparse grid based Bayesian method for contaminant source identification. *Advances in Water Resources*, 37:1–9, 2012. doi:10.1016/j.advwatres.2011.09.011.
- [42] G. Zhang, D. Lu, M. Ye, M. Gunzburger, and C. Webster. An adaptive sparse-grid high-order stochastic collocation method for Bayesian inference in groundwater reactive transport modeling. *Water Resources Research*, 49(10):6871–6892, 2013. doi:10.1002/wrcr.20467.