Adaptive sampling-based quadrature rules for efficient Bayesian prediction

van den Bos, L. M.M.; Sanderse, B.; Bierbooms, W. A.A.M.

DOI
10.1016/j.jcp.2020.109537

Publication date
2020

Document Version
Accepted author manuscript

Published in
Journal of Computational Physics

Citation (APA)
van den Bos, L. M. M., Sanderse, B., & Bierbooms, W. A. A. M. (2020). Adaptive sampling-based quadrature rules for efficient Bayesian prediction. *Journal of Computational Physics*, *417*, Article 109537. https://doi.org/10.1016/j.jcp.2020.109537

Adaptive sampling-based quadrature rules for efficient Bayesian prediction

L. M. M. van den Bos, B. Sanderse, and W. A. A. M. Bierbooms

28th February 2020

Abstract

A novel method is proposed to infer Bayesian predictions of computationally expensive models. The method is based on the construction of quadrature rules, which are well-suited for approximating the weighted integrals occurring in Bayesian prediction. The novel idea is to construct a sequence of nested quadrature rules with positive weights that converge to a quadrature rule that is weighted with respect to the posterior. The quadrature rules are constructed using a proposal distribution that is determined by means of nearest neighbor interpolation of all available evaluations of the posterior. It is demonstrated both theoretically and numerically that this approach yields accurate estimates of the integrals involved in Bayesian prediction. The applicability of the approach for a fluid dynamics test case is demonstrated by inferring accurate predictions of the transonic flow over the RAE2822 airfoil with a small number of model evaluations. Here, the closure coefficients of the Spalart–Allmaras turbulence model are considered to be uncertain and are calibrated using wind tunnel measurements.

Keywords: Bayesian prediction, Quadrature and cubature formulas, Adaptivity, Interpolation

1 Introduction

Computer simulation with uncertainties often requires calculating the expectation with respect to a probability distribution of a complex and computationally costly function. Various approaches exist to numerically estimate such integrals, for example Monte Carlo approaches [15] or deterministic quadrature rule approaches [10]. In these approaches it is often assumed that samples from the probability distribution can be constructed straightforwardly or are readily available. However, this is not always the case, for instance in the case of Bayesian calibration problems [26, 36] the probability distribution encompasses the computationally costly model and is highly non-trivial. This is known as Bayesian prediction and is the main topic of this article.

A commonly used approach to calculate weighted integrals with respect to such non-trivial distributions is importance sampling [8, 18, 58]. In this approach samples are drawn from a *proposal distribution* and by means of weighting these samples are used to determine integrals with respect to a *target distribution*, i.e. the distribution for which it is difficult to construct samples. If the proposal distribution is chosen well, the constructed samples yield similar convergence rates as samples constructed directly with the target distribution. A major disadvantage is that constructing the proposal distribution is not trivial and often requires a priori knowledge about the target distribution. A well-known extension of importance sampling to a Bayesian framework is formed by Markov chain Monte Carlo methods [31, 41], where the proposal distribution is chosen such that the samples form a Markov chain. In this case the proposal distribution is not constructed beforehand, though a large number of costly evaluations of the posterior is typically necessary to construct the sequence of samples.

There exist various approaches to improve the performance of Monte Carlo techniques in Bayesian inference. For example, using preconditioning based on Laplace's method [51], where an approximation of the posterior is constructed using the maximum a-posteriori estimate and a Gaussian distribution; using transport maps [46], where a low-fidelity model is used to precondition a Markov chain Monte Carlo sampler; or by adaptively tuning the proposal distribution [59]. However, many preconditioning methods are significantly less efficient if the posterior cannot be represented well using a Gaussian distribution. Techniques from the field of machine learning can also be applied [1], though the focus of those techniques is mostly on the treatment of large data sets.

Even if samples can be drawn from the posterior (possibly by using any of these improvements), the error of estimating an integral by means of random sampling decays as $\mathcal{O}(1/\sqrt{K})$ (with K the number of samples). For example, to double the accuracy of the estimate the number of samples must be quadrupled. Therefore using importance sampling techniques, or Monte Carlo techniques in general, in conjunction with a computationally costly model is often infeasible. In summary, estimating integrals by directly sampling from the posterior is often difficult or very expensive with existing methods.

A commonly used approach to alleviate the high computational cost of sampling and the slow convergence is to construct a surrogate (or "response surface") and use the surrogate instead of the complex model for inferring the relevant statistics. Well-known approaches include Gaussian process regression [3, 56] and polynomial approximation [4, 6, 40, 50, 52]. If the weighting distribution is not known explicitly, it is difficult to construct an optimal surrogate (and moreover it often requires analyticity or smoothness), so often the surrogate is constructed such that it is globally accurate (or the distribution is tempered [6, 38], which yields a similarly accurate surrogate). It has been demonstrated rigorously that a posterior determined with a globally accurate surrogate converges to the true posterior [14, 67].

At the same time, in the context of Bayesian prediction the interest is not directly in accurately approximating the *model*, but in calculating *moments* of the model with respect to the posterior. These moments are integral quantities and there exist many efficient numerical integration techniques for this purpose that do not explicitly construct a surrogate. In this article, the focus is specifically on quadrature rules due to their high accuracy and flexibility in the location of the nodes. Existing quadrature rule approaches [10] provide rapid converging estimates of the integral for smooth functions, without suffering from a deteriorated convergence rate if the integrand is not smooth (as often observed in interpolation approaches). However, the state-of-the-art quadrature rule approaches such as Gaussian quadrature rules [28], multivariate sparse grids [32, 44, 54], or heuristic optimization approaches [35, 42] also require much knowledge about the distribution under consideration, so these are not directly applicable to the problems of interest discussed here.

Therefore, the main goal of this work is to propose a novel flexible method for Bayesian prediction that combines the high convergence rates of quadrature rules with the flexibility of adaptive importance sampling. The idea is to construct a sequence of quadrature rules and a sequence of proposal distributions, where the proposal distributions are constructed using all available model evaluations. The sequence of quadrature rules converges to a quadrature rule that incorporates the (expensive) probability distribution (i.e. the posterior). By ensuring that all constructed quadrature rules have high degree and have positive weights, high convergence rates are obtained (depending on the smoothness of the integrand). The rules are specifically tailored to determine moments weighted with a posterior and require a small number of nodes to obtain an accurate estimation. Moreover it can be demonstrated rigorously that estimations made with the sequence of quadrature rules converge to the exact integral for a large class of integrands and distributions.

This article is structured as follows. In Section 2 the problem of Bayesian prediction with a quadrature rule is briefly introduced, with a main focus on quadrature rule methods and their relevant properties. In Section 3 our main contribution is discussed: a method to construct a quadrature rule that converges to a quadrature rule of the posterior. The method is analyzed theoretically in Section 4 and to demonstrate the applicability of the quadrature rule to computational problems, the technique is applied to some test functions in Section 5. Moreover the calibration of the turbulence closure coefficients of the RAE2822 airfoil is discussed. The article is summarized and concluded in Section 6.

2 Preliminaries

2.1 Bayesian prediction

The key goal of this article is to propose an efficient method to infer Bayesian predictions involving a computationally expensive model. However, the setting where the proposed methods can be applied is slightly more general than that. Let $u : \Omega \to \mathbb{R}^n$ be a continuous function with $\Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3, \dots$) that describes the quantity of interest and let $\rho : \Omega \to [0, \infty)$ be a continuous and bounded probability density function (PDF). Throughout this work, it is assumed that u can be evaluated exactly without any significant numerical error. Even though this is not a realistic assumption in practice, the stability of the methods proposed in this work guarantees that any numerical error present in the computational model does not affect the accuracy of the proposed approaches (i.e. errors “do not blow up”). Consequently determining Bayesian predictions can be described as determining the expectation of a function $f : \Omega \to \mathbb{R}$, with $f(\mathbf{X}) = F(u(\mathbf{X}))$ where $\mathbf{X}$ is a d-variate random vector with PDF ρ (so F is a function defined on the *range* of u). This integral will be denoted by the operator $\mathcal{I}^\rho$ throughout this article, i.e.

$$\mathcal{I}^\rho f \coloneqq \mathbb{E}[f(\mathbf{X})] = \int_\Omega f(\mathbf{x})\, \rho(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int_\Omega F(u(\mathbf{x}))\, \rho(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{2.1}$$

Integrals of this type have been studied many times in the framework of uncertainty propagation and many efficient methods exist to approximate them [37, 57, 66]. The approach taken in this work is based on a nearest neighbor interpolant of $\rho(\mathbf{x})$, which is used to construct quadrature rules for approximating the integral of (2.1). Geometric approaches based on nearest neighbor interpolation and the closely related Voronoi tessellation have been used succesfully to construct estimates of integrals as considered in this article [13, 19, 30, 39]. However, in these cases the distribution ρ is known explicitly or is computationally tractable beforehand (possibly up to constant). In this article, the computationally challenging part is that $\rho(\mathbf{x})$ is obtained by Bayesian analysis, i.e. it is, up to a constant, a *posterior*:

$$\rho(\mathbf{x}) \coloneqq q(\mathbf{z} \mid \mathbf{x})\, q(\mathbf{x}),$$

where $\mathbf{z}$, $q(\mathbf{x} \mid \mathbf{z})$, and $q(\mathbf{x})$ are the measurement data, the likelihood, and the prior respectively. The posterior, denoted by $q(\mathbf{z} \mid \mathbf{x})$, is obtained by scaling $\rho(\mathbf{x})$ such that it is a distribution:

$$q(\mathbf{x} \mid \mathbf{z}) = \frac{q(\mathbf{z} \mid \mathbf{x})\, q(\mathbf{x})}{\int_\Omega q(\mathbf{z} \mid \mathbf{x})\, q(\mathbf{x})\, \mathrm{d}\mathbf{x}} = \frac{\rho(\mathbf{x})}{\int_\Omega \rho(\mathbf{x})\, \mathrm{d}\mathbf{x}}.$$

The scaling factor (often called the *evidence*) that is present in this definition of ρ is neglected throughout this article, as it is not of importance for the proposed method. It will be shown that the method proposed in this work is stable (see Section 2.2.1) regardless of the exact value of the evidence. Moreover, this is also demonstrated numerically, by considering an example of a case with small evidence (see Section 5.1.2).

Bayesian prediction consists of assessing the *posterior predictive distribution*, which describes the probability of observing new data $\widehat{\mathbf{z}}$ given the existing data $\mathbf{z}$. It is obtained by marginalizing over the posterior as follows [22]:

$$q(\widehat{\mathbf{z}} \mid \mathbf{z}) = \int_\Omega q(\widehat{\mathbf{z}}, \mathbf{x} \mid \mathbf{z})\, \mathrm{d}\mathbf{x} = \int_\Omega q(\widehat{\mathbf{z}} \mid \mathbf{x}, \mathbf{z})\, q(\mathbf{x} \mid \mathbf{z})\, \mathrm{d}\mathbf{x} = \int_\Omega q(\widehat{\mathbf{z}} \mid \mathbf{x})\, q(\mathbf{x} \mid \mathbf{z})\, \mathrm{d}\mathbf{x}.$$

Here, we assume that $\widehat{\mathbf{z}}$ and $\mathbf{z}$ are conditionally independent given $\mathbf{x}$. Moments of the posterior predictive distribution can be used among others to obtain point predictions. These moments can often be expressed explicitly as integrals over the computational model u.

In general we require throughout this article that the distribution ρ is such that an evaluation of $\rho(\mathbf{x})$ requires an evaluation of $u(\mathbf{x})$. Moreover if u has been evaluated for a specific value of $\mathbf{x}$, the value of $\rho(\mathbf{x})$ can be determined without any significant computational cost. This is often the case if $\rho(\mathbf{x})$ forms a posterior in a Bayesian framework and this property is the key observation that will be leveraged to approximate (2.1).

A well-known example of a prior and a likelihood for a scalar model u are a uniform prior, defined as $q(\mathbf{x}) \propto 1$ for $\mathbf{x} \in \Omega$, and a Gaussian likelihood, defined as follows for given standard deviation σ:

$$q(\mathbf{x} \mid \mathbf{z}) \propto \exp\left[-\frac{1}{2}\frac{\sum_{k=1}^m (u(\mathbf{x}) - z_k)^2}{\sigma^2}\right], \text{ with measurement data } \mathbf{z} = (z_1, \dots, z_m)^\mathrm{T} \text{ known.}$$

In this particular case, the expectation of the posterior predictive distribution can be explicitly expressed as follows:

$$\begin{aligned}\mathbb{E}[\widehat{\mathbf{z}} \mid \mathbf{z}] &= \int_Z \widehat{\mathbf{z}}\, q(\widehat{\mathbf{z}} \mid \mathbf{z})\, \mathrm{d}\widehat{\mathbf{z}} = \int_\Omega \int_Z \widehat{\mathbf{z}}\, q(\widehat{\mathbf{z}} \mid \mathbf{x})\, q(\mathbf{x} \mid \mathbf{z})\, \mathrm{d}\widehat{\mathbf{z}}\, \mathrm{d}\mathbf{x} \\ &= \int_\Omega q(\mathbf{x} \mid \mathbf{z}) \int_Z \widehat{\mathbf{z}}\, q(\widehat{\mathbf{z}} \mid \mathbf{x})\, \mathrm{d}\widehat{\mathbf{z}}\, \mathrm{d}\mathbf{x} = \int_\Omega q(\mathbf{x} \mid \mathbf{z})\, \mathbb{E}[\widehat{\mathbf{z}} \mid \mathbf{x}]\, \mathrm{d}\mathbf{x} = \int_\Omega q(\mathbf{x} \mid \mathbf{z})\, u(\mathbf{x})\, \mathrm{d}\mathbf{x}. \end{aligned} \tag{2.2}$$

Here, Z denotes the space that contains all possible values of the data. It is not necessary to derive this space formally, as it is only used in an intermediate step of the derivation. Notice that the obtained integral is the expectation of the computational model with the parameters distributed according to the posterior. Various statistical models that combine measurement error and model error exist for the purpose of Bayesian model calibration. We refer the interested reader to Kennedy and O'Hagan [36] and will consider such an extensive model in a numerical example discussed in Section 5. Vector-valued u can be incorporated straightforwardly in the framework of this article, since only the posterior distribution $q(\mathbf{x} \mid \mathbf{z})$ is used to construct numerical integration techniques.

Throughout this article, two assumptions are imposed on the statistical setting. Firstly, we assume that the statistical moments of the prior are finite, i.e.

$$\int_{\Omega} \varphi(\mathbf{x})\, q(\mathbf{x})\, \mathrm{d}\mathbf{x} < \infty, \text{ for all polynomials } \varphi.$$

This implies among others that the prior is not improper. Secondly, it is assumed that the posterior is continuous and bounded. These two assumptions are not restrictive, e.g. statistical models with Gaussian random variables fit in this setting. Moreover, these assumptions are only necessary for the theoretical analysis of the approach discussed in this article. The approach requires only straightforward modifications for applying it to cases where an improper prior or discontinuous posterior is considered, and an example of the latter is discussed in more detail in Section 5.1.

2.2 Quadrature rules

The integral from (2.1) is approximated by means of a quadrature rule, consisting of nodes $\{\mathbf{x}_k\}_{k=0}^{N} \subset \Omega$ and weights $\{w_k\}_{k=0}^{N} \subset \mathbb{R}$:

$$\int_{\Omega} f(\mathbf{x})\, \rho(\mathbf{x})\, \mathrm{d}\mathbf{x} \approx \sum_{k=0}^{N} w_k f(\mathbf{x}_k). \tag{2.3}$$

Notice that the distribution $\rho(\mathbf{x})$ does not appear explicitly on the right hand side of this approximation, but that it is implicitly incorporated in the values of the nodes $\mathbf{x}_k$ and the weights w_k. We will denote the quadrature rule approximation by means of the linear operator $\mathcal{A}_N^{\rho}$, i.e.

$$\mathcal{A}_N^{\rho} f \coloneqq \sum_{k=0}^{N} w_k f(\mathbf{x}_k).$$

If $\rho(\mathbf{x}) = q(\mathbf{z} \mid \mathbf{x})\, q(\mathbf{x})$, a quadrature rule with respect to the posterior $q(\mathbf{x} \mid \mathbf{z})$ can be obtained by rescaling the weights of a quadrature rule of ρ such that $\sum_{k=0}^{N} w_k = 1$, provided that $\mathcal{A}_N^{\rho} 1 = \mathcal{I}^{\rho} 1$.

Three properties are relevant for quadrature rules that are applied to computationally expensive models: the exactness of the quadrature rule, positivity of the weights, and nesting. The first two properties ensure numerical stability of the quadrature rule and make that the quadrature rule converges. These two properties and their relevance are discussed in Section 2.2.1. Nesting allows for straightforward refinement of the quadrature rule approximation, and is briefly discussed in Section 2.2.2.

2.2.1 Exactness and positive weights

Often the exactness of a quadrature rule is related to the degree of the polynomials it integrates exactly. We choose to define the exactness slightly more general to facilitate refinements of arbitrary numbers of nodes.

To this end, let $\varphi_0, \dots, \varphi_D$ be $D + 1$ d-variate polynomials defined on Ω and consider the space $\Phi_D = \operatorname{span}\{\varphi_0, \dots, \varphi_D\}$. Then the goal is to construct a quadrature rule such that it integrates all functions $\varphi \in \Phi_D$ exactly. This is, due to linearity, equivalent to

$$\mathcal{A}_N^{\rho} \varphi_j = \mathcal{I}^{\rho} \varphi_j \text{ for } j = 0, \dots, D.$$

Without loss of generality, we assume throughout this article that the polynomials are sorted graded lexicographically. This ensures that referring to φ_k for any k is well-defined and that $\Phi_D \subset \Phi_{D+1}$ for any D, but other monomial orders that are a well-order can be used straightforwardly. Moreover we assume that φ_0 is the constant polynomial, as that allows to straightforwardly scale the quadrature rule weights to incorporate the evidence. In the univariate case, this implies that D is the degree of the quadrature rule, since Φ_D contains all polynomials of degree D and less. In the multivariate case, the degree of the quadrature rule equals Q if $\dim \Phi_D = \dim \mathbb{P}(Q, d)$ where $\mathbb{P}(Q, d)$ denotes all d-variate polynomials of a degree Q.

The close relation between the nodes and the weights of a quadrature rule is described by the $(D+1) \times (N+1)$ *Vandermonde-matrix* $V_D(X_N)$:

$$\underbrace{\begin{pmatrix} \varphi_0(\mathbf{x}_0) & \cdots & \varphi_0(\mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \varphi_D(\mathbf{x}_0) & \cdots & \varphi_D(\mathbf{x}_N) \end{pmatrix}}_{V_D(X_N)} \begin{pmatrix} w_0 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} \mu_0 \\ \vdots \\ \mu_D \end{pmatrix},$$

where $\mu_j = \mathcal{I}^\rho \varphi_j$. If φ_j are monomials, μ_j are the raw moments of the distribution ρ. If $N = D$ the quadrature rule is called *interpolatory* and the linear system can be used to determine the weights given the nodes, provided that the Vandermonde-matrix is non-singular. In the univariate case, this holds if and only if the nodes are distinct. In the multivariate case, it is less straightforward to ensure that the Vandermonde-matrix is non-singular, but all quadrature rules constructed in this work have a non-singular Vandermonde-matrix.

If the weights are non-negative (i.e. $w_k \geq 0$ for all k), the dimension of Φ_D is a measure for the accuracy of the quadrature rule and increasing D yields a converging estimate for sufficiently smooth functions. To see this, assume that Ω is bounded and let $\widehat{\varphi}$ be the best approximation of the function u in Φ_D measured in the ∞-norm (assuming without loss of generality that it exists):

$$\widehat{\varphi} \coloneqq \arg\min_{\varphi \in \Phi_D} \|u - \varphi\|_\infty, \text{ with } \|u - \varphi\|_\infty \coloneqq \sup_{\mathbf{x} \in \Omega} |u(\mathbf{x}) - \varphi(\mathbf{x})|.$$

Then:

$$|\mathcal{I}^\rho u - \mathcal{A}_N^\rho u| \leq \|\mathcal{I}^\rho\|_\infty \|u - \widehat{\varphi}\|_\infty + \|\mathcal{A}_N^\rho\|_\infty \|u - \widehat{\varphi}\|_\infty = (\|\mathcal{I}^\rho\|_\infty + \|\mathcal{A}_N^\rho\|_\infty) \inf_{\varphi \in \Phi_D} \|u - \varphi\|_\infty, \tag{2.4}$$

where

$$\|\mathcal{I}^\rho\|_\infty = \int_\Omega 1 \rho(\mathbf{x}) \,\mathrm{d}\mathbf{x} = \sum_{k=0}^N w_k \leq \sum_{k=0}^N |w_k| = \|\mathcal{A}_N^\rho\|_\infty.$$

The inequality (2.4) is commonly known as the Lebesgue inequality, see e.g. Brass and Petras [10, Theorem 3.1.1]. It is arguably best known for its usage in polynomial interpolation [11, 33], but it straightforwardly carries over to the setting of quadrature rules (then the operator norm $\|\mathcal{A}_N^\rho\|_\infty$ is the Lebesgue constant).

If all weights are positive, then $|w_k| = w_k$ and therefore $\|\mathcal{A}_N^\rho\|_\infty = \|\mathcal{I}^\rho\|_\infty$. Hence

$$|\mathcal{I}^\rho u - \mathcal{A}_N^\rho u| \leq 2 \|\mathcal{I}^\rho\|_\infty \inf_{\varphi \in \Phi_D} \|u - \varphi\|_\infty.$$

As explained before, the weights of the quadrature rule can be scaled such that ρ forms a probability density function. In that case, it holds that $\|\mathcal{I}^\rho\|_\infty = \|\mathcal{A}_N^\rho\|_\infty = 1$ and therefore (2.4) simplifies to

$$|\mathcal{I}^\rho u - \mathcal{A}_N^\rho u| \leq 2 \inf_{\varphi \in \Phi_D} \|u - \varphi\|_\infty.$$

It is well-known that the right hand side of this inequality decays for increasing D if u can be approximated by a polynomial. This is among others the case if u is absolutely continuous and in general the rate of decay of the right hand side of this inequality depends on the smoothness of u. If the function u is univariate and $u \in C^r$, i.e. u is r times differentiable, the error usually decays with rate r, or in other words:

$$\|u - \varphi\|_\infty \leq A D^{-r},$$

where A is a constant that solely depends on u. If u is smooth (i.e. infinitely many times continuously differentiable), the error decays exponentially. The exact rate of decay can be derived using Jackon's inequality [34], but many more results on this topic exist. We refer the interested reader to Watson [60] and the references therein for more information. The upper bound of the Lebesgue inequality is not sharp, as for unbounded Ω and discontinuous u the estimate of a quadrature rule might converge, even though $\inf_\varphi \|u - \varphi\|_\infty$ is unbounded. It is out of the scope of this article to fully discuss all convergence properties of quadrature rules (but see e.g. Brandolini et al. [9] or Brass and Petras [10]).

Notice that the bound described by the Lebesgue inequality depends on the dimension of Ω. For multivariate u, the quantity $\|u - \varphi\|_\infty$ decays significantly slower. In general, it holds that if u is r times differentiable, the error decays at least with rate r/d, where d is the dimension of the space Ω. Estimating the decay of this quantity in multivariate space is significantly less straightforward than estimating it in univariate spaces, since computing the best approximation becomes significantly more difficult. Nonetheless, the convergence rates as stated here are sufficient for the analysis conducted in this work.

A quadrature rule with positive weights and $\sum w_k = 1$ is stable regardless of the computational and statistical model under consideration [10]. In other words, small variations in the integrand do not affect the accuracy of the rule as a whole. In particular, if u is perturbed by a small error $\varepsilon > 0$, say $\tilde{u} = u \pm \varepsilon$, then

$$|\mathcal{A}_N^\rho u - \mathcal{A}_N^\rho \tilde{u}| \leq \|\mathcal{A}_N^\rho\|_\infty \|u - \tilde{u}\|_\infty \leq \varepsilon.$$

This stability result is particularly important for the usage of quadrature rules in a Bayesian setting, as it ensures that the rule yields stable estimates regardless of the exact value of the evidence (recall that the evidence is the scaling factor neglected in the definition of ρ). However, a small evidence will result in slow convergence, as will be demonstrated numerically in Section 5.1.2.

2.2.2 Nesting

Nesting means that smaller sets of quadrature rule nodes are contained in larger sets of quadrature rule nodes, i.e. $X_{N_1} \subset X_{N_2}$ for $N_1 < N_2$. Strictly speaking this forms a *sequence* of quadrature rules, but we will call this with a little abuse of nomenclature simply a nested quadrature rule.

A nested quadrature rule allows for straightforward refinement of the quadrature rule approximation, as computationally costly function evaluations are being reused upon considering a larger quadrature rule. Moreover, if the weights of the quadrature rule are non-negative, it provides a straightforward way to assess the accuracy of a quadrature rule approximation by comparing two consecutive quadrature rule levels. This can be seen as follows:

$$|\mathcal{A}_{N_1}^\rho u - \mathcal{A}_{N_2}^\rho u| \leq |\mathcal{A}_{N_1}^\rho u - \mathcal{I}^\rho u| + |\mathcal{A}_{N_2}^\rho u - \mathcal{I}^\rho u| \leq 4\|\mathcal{I}^\rho\|_\infty \inf_{\varphi \in \Phi_D} \|u - \varphi\|_\infty,$$

where D is such that $\mathcal{A}_{N_1}^\rho \varphi = \mathcal{A}_{N_2}^\rho \varphi = \mathcal{I}^\rho \varphi$ for all $\varphi \in \Phi_D$. The difference between two consecutive quadrature rules also forms an upper bound:

$$|\mathcal{A}_{N_1}^\rho u - \mathcal{A}_{N_2}^\rho u| \geq |\mathcal{A}_{N_1}^\rho u - \mathcal{I}^\rho u| - |\mathcal{A}_{N_2}^\rho u - \mathcal{I}^\rho u|,$$

which is the difference between the errors of the two quadrature rules. This difference vanishes for $N_1, N_2 \to \infty$ if a converging sequence of quadrature rule is used.

3 Bayesian predictions with an adaptive quadrature rule

The main problem addressed in this article is accurate numerical estimation of Bayesian predictions. To this end, the main interest is the construction of quadrature rules to approximate integrals weighted with the posterior, a problem that was summarized in (2.3) as follows:

$$\mathcal{I}^\rho f = \int_\Omega f(\mathbf{x})\, \rho(\mathbf{x})\, \mathrm{d}\mathbf{x} \approx \sum_{k=0}^{N} w_k f(\mathbf{x}_k) = \mathcal{A}_N^\rho f.$$

Here, both f and ρ are functions of u, denoted by $f(\mathbf{x}) = F(u(\mathbf{x}))$ and $\rho(\mathbf{x}) = G(u(\mathbf{x}))$ for suitable F and G. The functions F and G are such that they can be evaluated efficiently and quickly.

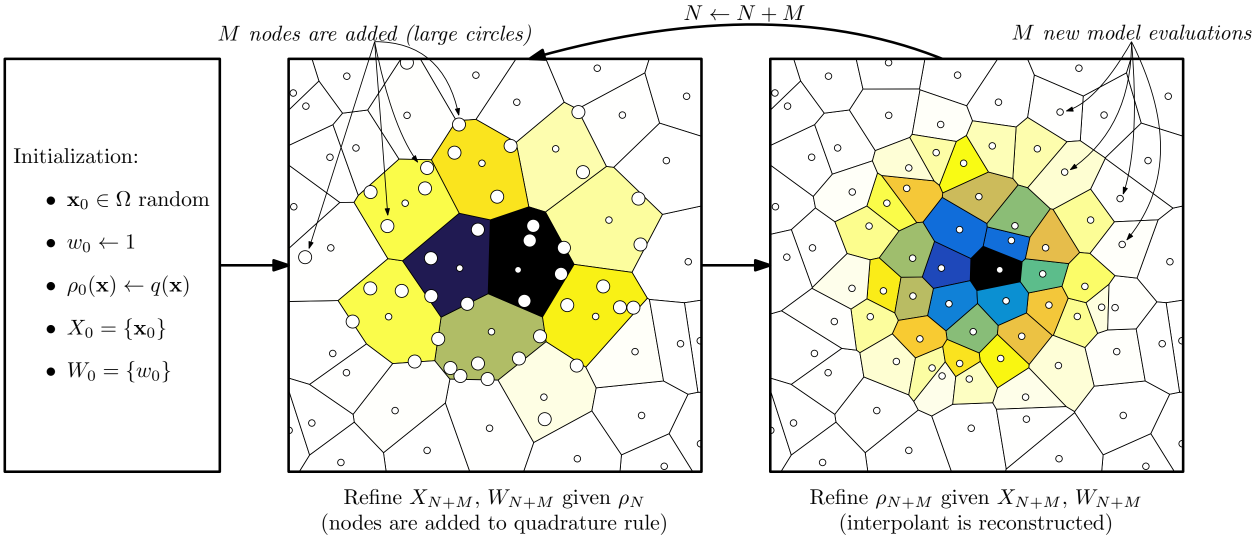

M nodes are added (large circles)
$N \leftarrow N+M$
M new model evaluations
Initialization:
- $\mathbf{x}_0 \in \Omega$ random
- $w_0 \leftarrow 1$
- $\rho_0(\mathbf{x}) \leftarrow q(\mathbf{x})$
- $X_0 = \{\mathbf{x}_0\}$
- $W_0 = \{w_0\}$
Refine X_{N+M}, W_{N+M} given ρ_N
(nodes are added to quadrature rule)
Refine ρ_{N+M} given X_{N+M}, W_{N+M}
(interpolant is reconstructed)

Figure 1: Schematic overview of the two main steps of the method to construct the quadrature rules proposed in this article.

The focus of this article is mainly on using a quadrature rule to accurately infer Bayesian predictions, but various other statistical quantities that are not considered in this article can be inferred straightforwardly without additional model evaluations. For example, moments of the posterior distribution can be determined by integrating suitably chosen polynomials or an empirical cumulative distribution function can be constructed by considering the discrete probability density function $p(\mathbf{x}_k) = w_k$ for all k (which is well-defined because all weights are non-negative).

3.1 Constructing an adaptive quadrature rule

The proposed procedure consists of iteratively refining a pair consisting of a quadrature rule and a distribution. A typical iteration consists of two steps. Firstly, a quadrature rule based on the current estimate of the distribution ρ is constructed. Secondly the estimate of the distribution is refined using all available model evaluations of ρ. The quadrature rules are constructed such that a sequence of nested rules is obtained. These steps are illustrated in Figure 1.

More specifically, at the start of the i-th iteration a distribution ρ_N is given, accompanied by evaluations of the model u and distribution ρ at nodes $\mathbf{x}_0, \dots, \mathbf{x}_N$. The first step consists of constructing a new quadrature rule with $N+M$ nodes that incorporates ρ_N and reuses the available model evaluations, i.e. it is nested. Hence the goal is to determine $\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+M}$ and $w_0, \dots, w_{N+M}$ with M as small as possible such that

$$\sum_{k=0}^{N+M} w_k \, \varphi_j(\mathbf{x}_k) = \int_\Omega \varphi_j(\mathbf{x}) \, \rho_N(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \text{ for all } j = 0, \dots, D_i.$$

Here, D_i is a free parameter which defines the exactness of the rule, i.e. the dimension of Φ_{D_i}. The second step of the iteration consists of constructing a refined approximation of the distribution ρ using all available evaluations of the model u and distribution ρ (including the M nodes that have been added this iteration). In this work, this is done by nearest neighbor interpolation. Consequently ρ_{N+M} is obtained, which is used for the next iteration. The procedure can be initialized using a quadrature rule consisting of a single random node and a proposal distribution $\rho_0(\mathbf{x}) \equiv q(\mathbf{x})$, i.e. by using the prior. The convergence can each iteration be assessed by comparing the current quadrature rule estimate with that of a previous iteration. The algorithm is sketched in Figure 1.

The only free parameter in this construction is D_i, which can be chosen heuristically. For convergence, it is essential to enforce that $D_i \to \infty$ for $i \to \infty$. In this work, we will mainly use two variants: linear growth, i.e. $D_i = \mathcal{O}(i)$, and exponential growth, i.e. $D_i = \mathcal{O}(2^i)$.

The distribution ρ_N is constructed by interpolation of all available point evaluations of ρ. It can be interpreted as a *proposal distribution*, as often used in importance sampling strategies. The efficiency of the approach can be demonstrated mathematically by reusing techniques from importance sampling.

The construction of a quadrature rule that incorporates the distribution ρ_N is non-trivial. We will employ the recently introduced implicit quadrature rule [7] for this purpose, though we emphasize that any methodology to construct quadrature rules for a given distribution can be used in our approach. The implicit quadrature rule is a nested quadrature rule with positive weights that is constructed solely using samples from the distribution ρ_N. We discuss its construction in Section 3.2. Given the quadrature rule nodes, the distribution that is used to refine the quadrature rule is constructed using nearest neighbor interpolation, which is explained in Section 3.3.

3.2 Implicit quadrature rule

The implicit quadrature rule [7] is a quadrature rule constructed using samples from a distribution. Hence let $K+1$ samples $\mathbf{y}_0, \dots, \mathbf{y}_K$ be given, denoted as the set Y_K. In the iterative procedure considered in this work (i.e. Figure 1), these are a large number of samples from ρ_N. Moreover, some model evaluations have been performed in previous iterations, so let $X_N \subset \Omega$ be $N+1$ nodes in Ω where the value of u (and therefore ρ) is known. These nodes are denoted with $\mathbf{x}_0, \dots, \mathbf{x}_N$. Notice that ρ_N is typically obtained by nearest neighbor interpolation of these nodes, which we will discuss in more detail in Section 3.3.

The goal is to find for given D (we omit the subscript i in this section) a subset of M samples from Y_K, say $\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+M}$, with $M \ll K$ such that

$$\sum_{k=0}^{N+M} w_k \, \varphi_j(\mathbf{x}_k) = \frac{1}{K+1} \sum_{k=0}^{K} \varphi_j(\mathbf{y}_k), \text{ for } j = 0, \dots, D. \tag{3.1}$$

The rule is constructed such that the weights w_k are non-negative, i.e. $w_k \geq 0$ (for all k). This ensures that the quadrature rule error is bounded, which can be seen by applying the Lebesgue inequality from (2.4).

Notice that only samples from ρ_N are necessary to construct this quadrature rule, from which the approximate moments are computed. This is an advantage, since in higher dimensional spaces it is often computationally costly to exactly determine the moments of ρ_N.

The nodes $\mathbf{x}_0, \dots, \mathbf{x}_N$ do generally not form a subset of the samples Y_K. Ideally we would like to have that $M = 0$, which would imply that no new costly model evaluations are required, but this is often not possible as D increases with each iteration. However, we ensure that the following bound is valid:

$$0 \leq M \leq D + 1, \tag{3.2}$$

such that we add at most $D+1$ nodes to the quadrature rule, consequently obtaining a rule of at most $N + D + 2$ nodes.

The first step in obtaining such a quadrature rule is to consider the following quadrature rule:

$$\begin{aligned} X_{N+K+1} &= \{\mathbf{x}_0, \dots, \mathbf{x}_N, \mathbf{y}_0, \dots, \mathbf{y}_K\}, \\ W_{N+K+1} &= \Big\{\underbrace{0, \dots, 0}_{N+1}, \underbrace{\frac{1}{K+1}, \dots, \frac{1}{K+1}}_{K+1}\Big\}. \end{aligned} \tag{3.3}$$

It is trivial to see that this quadrature rule is such that (3.1) holds, but it is not such that (3.2) holds. The idea is to iteratively remove nodes from this quadrature rule keeping (3.1) as an invariant, which can be done until (3.2) holds. All constructed intermediate quadrature rules have positive weights.

To start off, notice that (3.3) in conjunction with (3.1) can be rewritten as the following linear system:

$$\begin{pmatrix} \varphi_0(\mathbf{x}_0) & \cdots & \varphi_0(\mathbf{x}_N) & \varphi_0(\mathbf{y}_0) & \cdots & \varphi_0(\mathbf{y}_K) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \varphi_D(\mathbf{x}_0) & \cdots & \varphi_D(\mathbf{x}_N) & \varphi_D(\mathbf{y}_0) & \cdots & \varphi_D(\mathbf{y}_K) \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_N \\ w_{N+1} \\ \vdots \\ w_{N+K+1} \end{pmatrix} = \begin{pmatrix} \mu_0 \\ \vdots \\ \mu_D \end{pmatrix},$$

or denoted equivalently as:

$$V_D(X_{N+K+1})\mathbf{w} = \boldsymbol{\mu}.$$

Here μ_j are the moments determined by means of sampling ("sample moments"): $\mu_j = 1/(K+1)\sum_{k=0}^{K} \varphi_j(\mathbf{y}_k)$. The Vandermonde-matrix of this linear system (see Section 2.2.1) is a $(D+1) \times (N+K+2)$-matrix, so it has $J \coloneqq (N+K+2) - (D+1)$ linearly independent null vectors $\mathbf{c}_1, \dots, \mathbf{c}_J$, with

$$V_D(X_{N+K+1})\mathbf{c}_j = \mathbf{0}, \text{ for } j = 1, \dots, J.$$

Hence it holds that

$$V_D(X_{N+K+1})(\mathbf{w} - \mathbf{c}) = \boldsymbol{\mu}, \text{ for any } \mathbf{c} \in \operatorname{span}\{\mathbf{c}_1, \dots, \mathbf{c}_J\}.$$

These null vectors can be used to remove J nodes from the quadrature rule X_{N+K+1}, W_{N+K+1} [5], which is done by constructing a $\mathbf{c} \in \operatorname{span}\{\mathbf{c}_1, \dots, \mathbf{c}_J\}$ such that:

$$\mathbf{w} - \mathbf{c} \geq 0, \tag{3.4a}$$

$$w_{k_j} - c_{k_j} = 0, \text{ for } J \text{ distinct indices } k_1, \dots, k_J. \tag{3.4b}$$

Condition (3.4a) describes that all weights should be non-negative and condition (3.4b) describes that there should be J weights that equal zero. If such a vector $\mathbf{c}$ is constructed, the nodes $k_1, \dots, k_J$ can be removed from the quadrature rule without loss of accuracy. Then the weights can straightforwardly be determined by $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{c}$.

The algorithm to determine such a $\mathbf{c}$ is initiated with $\mathbf{c} = \mathbf{0}$ and $C = \{\mathbf{c}_1, \dots, \mathbf{c}_J\}$. At each iteration, $\mathbf{c}$ is updated such that firstly (3.4a) is still valid and secondly such that one more weight is equal to zero. Therefore, consider $\mathbf{c}_1$, an element (and without loss of generality, the first element) of C. Since $\mathbf{c}_1$ is a null vector, it holds that $\mathbf{c} + \alpha\mathbf{c}_1$ is a null vector for any $\alpha \in \mathbb{R}$. The goal is to determine α such that conditions (3.4a) and (3.4b) hold. Condition (3.4a) translates to

$$\mathbf{w} - \mathbf{c} - \alpha\mathbf{c}_1 \geq 0,$$

which is equivalent to

$$\alpha_{\min} \leq \alpha \leq \alpha_{\max}, \text{ with}$$

$$\alpha_{\min} = \max\left(\frac{w_k - c_k}{c_{1,k}} \mid c_{1,k} < 0\right) \text{ and } \alpha_{\max} = \min\left(\frac{w_k - c_k}{c_{1,k}} \mid c_{1,k} > 0\right).$$

Here w_k, c_k, and $c_{1,k}$ denote the elements of $\mathbf{w}$, $\mathbf{c}$, and $\mathbf{c}_1$ respectively. Notice that $\alpha_{\min} < 0 < \alpha_{\max}$, since $c_{1,k} < 0$ for $\alpha_{\min}$ and $c_{1,k} > 0$ for $\alpha_{\max}$. Hence α is well-defined in this way. By choosing either $\alpha = \alpha_{\min}$ or $\alpha = \alpha_{\min}$, there exists one k_0 such that

$$w_{k_0} - c_{k_0} - \alpha c_{1,k_0} = 0.$$

Therefore, $\mathbf{c}$ is updated such that $\mathbf{c} \leftarrow \mathbf{c} + \alpha\mathbf{c}_1$ with $\alpha = \alpha_{\min}$ or $\alpha = \alpha_{\max}$.To ensure that the k_0-th weight remains zero in subsequent iterations, all other null vectors in C are updated as follows:

$$\mathbf{c}_j = \mathbf{c}_j - \frac{c_{j,k_0}}{c_{1,k_0}}\mathbf{c}_1, \text{ for all } j > 1.$$

This is essentially a Gaussian reduction with pivot c_{1,k_0} and ensures that $c_{j,k_0} = 0$ for all $j > 1$, which yields that $w_{k_0} - c_{k_0} = 0$ in all subsequent iterations (independently of the chosen α in that iteration). Finally $\mathbf{c}_1$ is removed from C and the process is repeated until C is empty. If C is empty, a $\mathbf{c}$ is obtained that satisfies both conditions (3.4a) and (3.4b).

By determining such a $\mathbf{c}$, the vector $\mathbf{w} - \mathbf{c}$ has J coefficients that are equal to 0. We do not want to remove all J nodes and weights, since the first $N + 1$ weights belong to nodes where the costly computational model has been evaluated. A straightforward solution to this is to construct the following quadrature rule:

$$X_{N+M} = \{\mathbf{x}_k \mid k \leq N \text{ or } w_k - c_k \neq 0\},$$

$$W_{N+M} = \{w_k - c_k \mid k \leq N \text{ or } w_k - c_k \neq 0\}.$$

The vector $\mathbf{w} - \mathbf{c}$ has at least J zero elements. So it has at most $D + 1$ nonzero elements. Hence the quadrature rule based on X_{N+M} and W_{N+M} has at most $(N + 1) + (D + 1)$ nodes. Notice that therefore this is a quadrature rule that satisfies (3.1) and (3.2).

It is not necessary to compute the full null space of the $(D + 1) \times (N + K + 2)$-matrix. Each iteration a null vector of the full matrix can be obtained by computing a null vector of a $(D + 1) \times (D + 2)$ submatrix using $D + 2$ columns that represent nodes that are still present in the quadrature rule, and subsequently padding the obtained vector with elements equal to 0.

There is freedom in the value of α, i.e. both $\alpha = \alpha_{\max}$ and $\alpha = \alpha_{\min}$ yield a valid quadrature rule with positive weights. We use this freedom to determine for each null vector $\mathbf{c}_k$ the α that yields the smallest quadrature rule (i.e. $w_k - c_k = 0$ for as many $k > N$ as possible).

3.3 Construction of the proposal distribution

By using the quadrature rule constructed using the techniques from the previous section, a Bayesian prediction can be inferred by evaluating the model at the quadrature rule nodes and assessing the accuracy using the nesting of the quadrature rules. By using Bayes' rule, evaluations of the (unscaled) posterior are readily available, that is $\rho(\mathbf{x}_k)$ for $k = 0, \dots, N + M$. The goal is to use these evaluations to construct an approximation of the posterior.

Constructing an approximation of a distribution is non-trivial and there are some specific requirements on the interpolation algorithm used in this work. These requirements are further discussed in Section 3.3.1. In this article, all results are obtained using nearest neighbor interpolation, which is explained in Section 3.3.2. There exist various alternative approaches, with varying advantages and disadvantages that warrant our choice for nearest neighbor interpolation. These alternatives are reviewed in Section 3.3.3.

3.3.1 Prerequisites of the interpolant

The goal is to construct an approximate posterior ρ_{N+M} such that the available $N + M + 1$ point evaluations of ρ are taken into account. These point evaluations of the posterior are exact, so in this work we construct ρ_{N+M} such that it is interpolatory, i.e.

$$\rho_{N+M}(\mathbf{x}_k) = \rho(\mathbf{x}_k), \text{ for all } k = 0, \dots, N + M.$$

Obviously, the interpolant should be constructed such that $\rho_{N+M}(\mathbf{x}) \approx \rho(\mathbf{x})$ for all $\mathbf{x} \neq \mathbf{x}_k$ (for any k). It is not necessary that ρ_{N+M} is interpolatory for the construction of the quadrature rule, but it is assumed to be the case in the analysis discussed in Section 4.

It is known that ρ is a probability density function, possibly up to a finite constant. Therefore the interpolant should be such that $\rho_{N+M}(\mathbf{x}) \geq 0$ for all $\mathbf{x}$. Moreover, since a quadrature rule is constructed using this distribution, all moments must be finite, i.e.

$$\int_\Omega \varphi(\mathbf{x})\, \rho_{N+M}(\mathbf{x})\, \mathrm{d}\mathbf{x} < \infty, \text{ for all polynomials } \varphi.$$

These two conditions ensure that ρ_{N+M} is again a probability density function (possibly up to a constant).

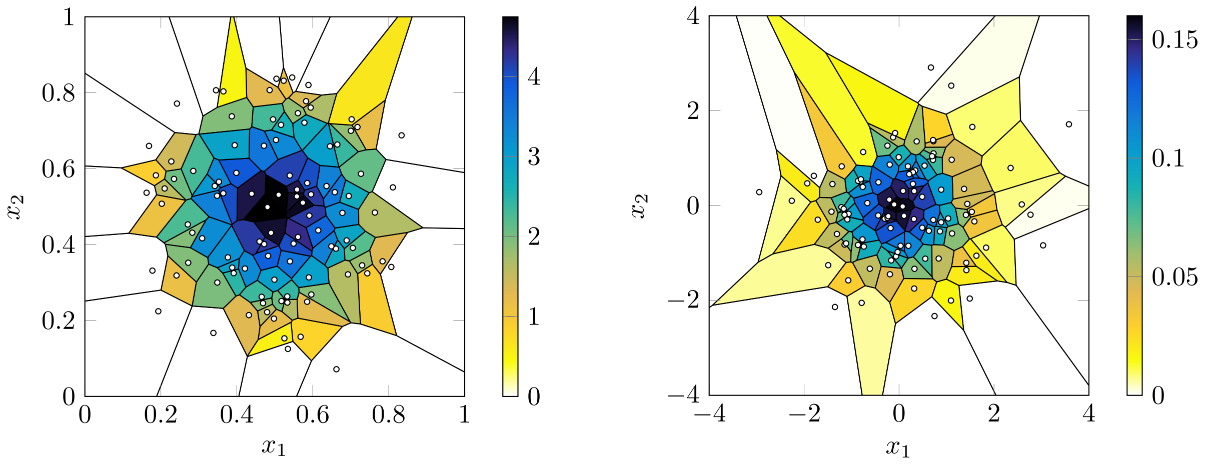
We emphasize that in general it is *not* preferable that the moments of ρ_{N+M} are equal to the moments of the quadrature rule, i.e. in general it should hold that

$$\mathcal{I}^{\rho_{N+M}}\varphi_j = \int_\Omega \varphi_j(\mathbf{x})\, \rho_{N+M}(\mathbf{x})\, \mathrm{d}\mathbf{x} \neq \int_\Omega \varphi_j(\mathbf{x})\, \rho_N(\mathbf{x})\, \mathrm{d}\mathbf{x} = \sum_{k=0}^{N+M} w_k\, \varphi_j(\mathbf{x}_k) = \mathcal{A}_{N+M}^{\rho_N}\varphi_j, \text{ for } j = 0, \dots, D_i.$$

Here, we use the notation $\mathcal{I}^{\rho_{N+M}}$ and $\mathcal{A}_{N+M}^{\rho_N}$ as defined in Section 2.2, i.e.

$$\mathcal{I}^{\rho_{N+M}}u = \int_\Omega u(\mathbf{x})\, \rho_{N+M}(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

$$\mathcal{A}_{N+M}^{\rho_N}u = \sum_{k=0}^{N+M} w_k\, u(\mathbf{x}_k).$$



(a) Beta(4, 4)

(b) $\mathcal{N}(0, 1)$

Figure 2: Nearest neighbor interpolations of random samples based on two distributions. In both cases, x_1 and x_2 are identically and independently distributed.

The moments estimated by the quadrature rule are generally not equal to the exact moments of the true distribution ρ. Hence forcing that the moments of ρ_{N+M} are equal to the moments of the (current) quadrature rule does not lead to a converging sequences of distributions (as all moments would be equal to the approximation of the first quadrature rule).

3.3.2 Nearest neighbor interpolation

In this work, ρ_{N+M} is constructed by means of nearest neighbor interpolation of the likelihood and multiplying the obtained interpolant with the prior. Hence for any $\mathbf{x} \in \Omega$ the interpolant $\rho_{N+M}(\mathbf{x})$ is defined as follows:

$$\rho_{N+M}(\mathbf{x}) = q(\mathbf{z} \mid \mathbf{x}_k)\, q(\mathbf{x}), \text{ with } \mathbf{x}_k = \operatorname*{arg\,min}_{\mathbf{x}_j \in X_{N+M}} \|\mathbf{x} - \mathbf{x}_j\|_2, \text{ with } X_{N+M} = \{\mathbf{x}_0, \dots, \mathbf{x}_{N+M}\}.$$

The advantages of this approach are among others that the obtained interpolant is always non-negative and globally bounded, i.e. for all $\mathbf{x}$ it holds that

$$0 \le \rho_{N+M}(\mathbf{x}) \le \left(\sup_{\mathbf{x}\in\Omega} q(\mathbf{z} \mid \mathbf{x})\right) \left(\sup_{\mathbf{x}\in\Omega} q(\mathbf{x})\right).$$

If an improper prior on an unbounded domain is considered, it might be the case that ρ_{N+M} is not a well-defined distribution. This can be alleviated by restricting the nearest neighbor interpolant to a compact set that is larger than the convex hull of the quadrature rule nodes.

Moreover, for the purposes of this article it can be implemented very efficiently, as the number of interpolation points is relatively small and the exact structure of the interpolant is not of importance: sampling from ρ_{N+M} can simply be done by means of acceptance rejection sampling using the prior as proposal distribution (which requires that the prior is not improper).

In particular, for the nearest neighbor interpolant it is not necessary to construct the Voronoi diagram of the nodes, but only to implement a routine that can find the closest node for any given input sample. This can be done very fast and efficiently for large numbers of samples. Moreover, searching for the nearest node has only a weak dependence of the dimension on the space (contrary to constructing the Voronoi diagram, which is very costly in higher dimensional spaces), which allows to use this approach in high dimensions.

Nearest neighbor interpolation is being used often in statistics [2], especially with a focus on classification problems. It belongs to the broader class of scattered data interpolation methods [24] and is arguably one of the most straightforward methods to use in this regard. Moreover, it is a local interpolation method, whereas the quadrature rule is a global integration method, which naturally balances exploitation of locality with exploration of the space without any free parameters that have to be tuned.

As mentioned briefly before, a nearest neighbor interpolant can also be applied directly to the integral for the construction of quadrature rules [13, 19, 30, 39]. Many of these approaches focus on explicitly constructing the nodal set such that the nodes are the centroids (centers of mass) of the cells of the Voronoi diagram. These approaches cannot be extended straightforwardly to the setting to this article, since it is often assumed that the distribution is computationally accessible. Moreover, it is usually necessary to construct the Voronoi diagram, which hinders the applicability to higher dimensional spaces.

The idea of using a nearest neighbor interpolant to construct a surrogate has been explored successfully before and various improvements can be considered to enhance the accuracy of the interpolant [20, 49]. The advantages of constructing a surrogate in this way are among others that the surrogate exploits locality, as mentioned above. The main difference between the approaches used in [20, 49] and the approach considered in this article is that in this work a surrogate of a probability distribution is generated, which adds several constraints to the approach (see Section 3.3.1).

Nearest neighbor interpolation has been illustrated in Figure 1 (see page 7).Two other examples of interpolated probability density functions are depicted in Figure 2. For sake of illustration, the interpolation nodes of Figure 2 are based on random samples.

3.3.3 Alternative distribution approximation methods

It is non-trivial to consider more accurate interpolation methods in this framework, as assuring that the obtained interpolant is a PDF is difficult. For example, many global methods such as Lagrange interpolation and Gaussian process regression do not yield an interpolant that is non-negative. Extensions to enforce positivity exist, for example for Lagrange interpolation [25, 29]. Often these approaches only work in univariate cases. Gaussian process regression can possibly be used by using a strictly positive prior that is not improper and enforcing finiteness of integrals over the obtained approximation [48], but this is rather involved and conflicts with the convenient Bayesian framework behind Gaussian processes.

Another well-known method to construct a suitable approximation is the kernel density estimate, or more general, the family of methods based on radial basis functions [12]. However, in the cases in this article explicit values of the probability density function are known, which cannot be incorporated in a straightforward fashion in these methods: enforcing specific values of a function in a radial basis function setting does not necessarily yield a positive interpolant.

A promising method that balances the accuracy of polynomial interpolation and piecewise behavior of nearest neighbor interpolation is the Simplex Stochastic collocation method [23, 63, 64, 65], where piecewise high order polynomial interpolation is combined with a Delaunay triangulation of the nodal set. This method converges exponentially for sufficiently smooth functions and is non-oscillatory (i.e. it is bounded from below and above by the minimum and maximum of the model), so it is very suitable for the interpolation of a PDF. However, it requires the explicit construction of the interpolant with the underlying Delaunay triangulation, which severely limits its applicability to multi-dimensional spaces.

Concluding, it is nontrivial to consider a more efficient alternative approach to nearest neighbor interpolation that yields a positive interpolant with similar efficiency as nearest neighbor interpolation. It is out of the scope of this work to derive such a new interpolation approach. Moreover, nearest neighbor interpolation is sufficiently accurate for the examples considered in this work, as demonstrated in Section 5.

4 Error analysis

The accuracy of the proposed method is studied by assessing the convergence of the sequence of quadrature rules. The overall error we are interested in is the integration error, defined for a given continuous function $f : \Omega \to \mathbb{R}$ as follows:

$$e_N(f) \coloneqq |\mathcal{I}^{\rho} f - \mathcal{A}_N^{\rho_N} f|.$$

Here the operator $\mathcal{A}_N^{\rho_N}$ has been constructed using the methods outlined in Section 3, i.e. for known D_i we have

$$\mathcal{A}_N^{\rho_N} \varphi_j = \sum_{k=0}^{N} w_k \, \varphi_j(\mathbf{x}_k) = \int_{\Omega} \varphi_j(\mathbf{x}) \, \rho_N(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \mathcal{I}^{\rho_N} \varphi_j, \text{ for } j = 0, \dots, D_i,$$

where we neglect the effect of using samples for the construction of the quadrature rule.

It can be demonstrated that $e_N \to 0$ if $N \to \infty$, provided that f is sufficiently smooth and $D_i \to \infty$ for $i \to \infty$. The details of this are discussed in Section 4.1.

A central question is whether this error decays faster than the integration error of a conventional quadrature rule constructed using the prior. As constructing such a quadrature rule does not require interpolation or sampling, there are significantly fewer sources of error. This topic can be addressed by using principles from importance sampling, which is outlined in Section 4.2.

The error analysis based on importance sampling neglects any error that occurs due to the usage of samples of ρ_{N+M} instead of the full distribution. We do not consider this a severe limitation, since sampling from ρ_{N+M} does not require costly model evaluations and therefore many samples can be used to minimize this error. Nonetheless, this error will be briefly considered in Section 4.3.

4.1 Decay of the integration error

Notice that the integration error $e_N(f)$ can be decomposed into three components: an interpolation error, a sampling error, and a quadrature rule error:

$$\begin{aligned} e_N(f) = |\mathcal{I}^\rho f - \mathcal{A}_N^{\rho_N} f| &= |\mathcal{I}^\rho f - \mathcal{I}^{\rho_N} f + \mathcal{I}^{\rho_N} f - \mathcal{A}_N^{\rho_N} f| \\ &\leq |\mathcal{I}^\rho f - \mathcal{I}^{\rho_N} f| + |\mathcal{I}^{\rho_N} f - \mathcal{A}_N^{\rho_N} f| \\ &= |\mathcal{I}^\rho f - \mathcal{I}^{\rho_N} f| + |\mathcal{I}^{\rho_N} f - \mathcal{I}_K^{\rho_N} f + \mathcal{I}_K^{\rho_N} f - \mathcal{A}_N^{\rho_N} f| \\ &\leq \underbrace{|\mathcal{I}^\rho f - \mathcal{I}^{\rho_N} f|}_{\text{Interpolation error } I_N} + \underbrace{|\mathcal{I}^{\rho_N} f - \mathcal{I}_K^{\rho_N} f|}_{\text{Sampling error } S_K} + \underbrace{|\mathcal{I}_K^{\rho_N} f - \mathcal{A}_N^{\rho_N} f|}_{\text{Quadrature error } Q_N} \\ &= I_N + S_K + Q_N. \end{aligned} \tag{4.1}$$

Here, $\mathcal{I}_K^{\rho_N} f$ denotes the Monte Carlo estimate of the integral computed using $K+1$ samples from ρ_N:

$$\mathcal{I}_K^{\rho_N} f := \frac{1}{K+1} \sum_{k=0}^{K} f(\mathbf{y}_k).$$

By introducing this operator, the quadrature rule error Q_N can be bounded using the Lebesgue inequality from (2.4) as discussed in Section 2.2.1:

$$Q_N \leq \left(1 + \sum_{k=0}^{N} |w_k|\right) \inf_{\varphi \in \Phi_{D_i}} \|f - \varphi\|_\infty.$$

Since all quadrature rules constructed in this work have positive weights, this error decays if $D_i \to \infty$ (with $i \to \infty$) for sufficiently smooth f.

The interpolation error I_N depends on the specific procedure used to construct the interpolant. It holds that $I_N \to 0$ for $N \to \infty$ if $\|\rho_N - \rho\|_\infty \to 0$. The latter can be demonstrated straightforwardly for nearest neighbor interpolants by demonstrating that the quadrature rule nodes distribute across the space, i.e. the mutual distance between two closest nodes decays to zero. A nearest neighbor interpolant yields a converging interpolant in that case.

To demonstrate that quadrature rule nodes distribute across the space, let $\mathbf{x}_0, \dots, \mathbf{x}_k, \dots$ be a sequence of nodes such that $\{\mathbf{x}_0, \dots, \mathbf{x}_N\}$ form the nodes of a quadrature rule with positive weights with respect to a distribution ρ for all N. Moreover assume that $\mathbf{x}_k \in \Omega$ for all $k \in \mathbb{N}$. Suppose $S \subset \Omega$ is such that there is no $k \in \mathbb{N}$ such that $\mathbf{x}_k \in S$. To demonstrate that the nodes distribute across the space, we will show that S as considered here is a null set, i.e.

$$\int_S \rho(\mathbf{x}) \,\mathrm{d}\mathbf{x} = 0,$$

which implies that the mutual distance between the quadrature rule nodes vanishes. To demonstrate that S is a null set, let $u_S \in C^\infty(\Omega)$ be a smooth function such that $u_S(\mathbf{x}) = 0$ for all $\mathbf{x} \notin S$ and such that

$$\int_\Omega u_S(\mathbf{x}) \,\rho(\mathbf{x}) \,\mathrm{d}\mathbf{x} > 0.$$

Such a u_S only exists if S is not a null set. A univariate example of such a function is the bump function, e.g. if $S = [-1, 1]$ then

$$u_S(x) = \begin{cases} \exp\left(-\frac{1}{1-x^2}\right) & \text{if } |x| < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, multivariate bump functions can be constructed by products of univariate bump functions defined on a hypercube in S. Due to the smoothness of u_S, there exists a sequence of polynomials φ_N such that $\|u_S - \varphi_N\|_\infty \to 0$ for $N \to \infty$. Hence *any* quadrature rule operator $\mathcal{A}_N^\rho$ with positive weights yields a converging estimate of $\mathcal{I}^\rho u_S$, i.e.

$$\mathcal{A}_N^\rho u_S = \sum_{k=0}^{N} w_k \, u_S(\mathbf{x}_k) \to \int_\Omega u_S(\mathbf{x}) \, \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \text{ for } N \to \infty.$$

We assumed that the sequence of nodes is such that there is no $\mathbf{x}_k \in S$ for any $k \in \mathbb{N}$. Since the quadrature rule converges, this yields that the integral of u_S must vanish:

$$\int_\Omega u_S(\mathbf{x}) \, \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \int_S u_S(\mathbf{x}) \, \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 0.$$

This holds for any smooth function defined in this way, which is only possible if

$$\int_S \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 0.$$

Notice that positivity of the weights is essential here, as otherwise it is not guaranteed that the quadrature rule approximation of any smooth function converges.

A similar derivation can be formulated for the nodes deduced with the methodology proposed in this article. To see this, notice that for u_S, as introduced above, it holds that

$$\left| \sum_{k=0}^{N} w_k \, u_S(\mathbf{x}_k) - \int_\Omega u_S(\mathbf{x}) \, \rho_N(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right| \leq \inf_{\varphi \in \Phi_{D_i}} \|u_S - \varphi\|_\infty,$$

which simplifies to (recall that $u_S(\mathbf{x}_k) = 0$ for all k)

$$\left| \int_S u_S(\mathbf{x}) \, \rho_N(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right| \leq \inf_{\varphi \in \Phi_{D_i}} \|u_S - \varphi\|_\infty. \tag{4.2}$$

These inequalities hold for any N. Recall that $\rho(\mathbf{x}) > 0$ for all $\mathbf{x}$, and therefore $\rho_N(\mathbf{x}) > 0$ for all $\mathbf{x}$ and N. Hence, if $\int_S \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x} > 0$, the left hand side of (4.2) is bounded from below by a positive constant that is independent of N. Following the same argument as above, we therefore conclude that S is a null set.

The sampling error S_K is independent from N, and can therefore be reduced without any additional costly model evaluations by considering more samples from ρ. The exact convergence rate depends on the constructed sequence of samples, which will be further discussed in Section 4.3.

Concluding, if $i \to \infty$ and $N \to \infty$, it holds that $e_N(f) \to S_K$ provided that $\inf_{\varphi \in \Phi_{D_i}} \|\varphi - f\|_\infty \to 0$. Moreover, if also $K \to \infty$, it holds that $e_N(f) \to 0$. This demonstrates the convergence of our method for sufficiently smooth functions, but this does not demonstrate that our method converges faster than constructing a conventional quadrature rule with respect to the prior, since such a rule does not have an interpolation error. This is the main topic of Section 4.2.

4.2 Importance sampling

The proposed quadrature rule of this article has three sources of error: an interpolation error I_N, a sampling error S_K, and a quadrature rule error Q_N. A quadrature rule constructed using the prior (e.g. by means of a sparse grid) only has the quadrature rule error, i.e. (4.1) simplifies to $e_N(f) = Q_N$.

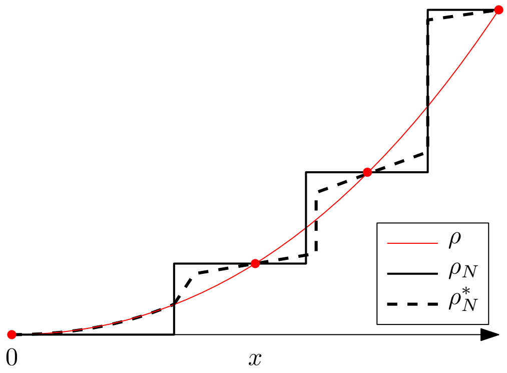


Figure 3: A sketch of the motivation why ρ_N^* is introduced. In this particular case, $\rho(x)/\rho_N(x)$ is unbounded for $x \to 0$. Therefore, a ρ_N^* is introduced that yields a bounded fraction near $x = 0$. Small corrections are necessary to enforce that the raw moments do not change.

The main difference is that a rule that only uses the prior requires a different integrand, i.e. $\mathcal{I}^\rho f$ is approximated by means of $\mathcal{A}_N^{q(.)}(q(\mathbf{z} \mid \mathbf{x}) f(\mathbf{x}))$, whereas the quadrature rule proposed in this work uses $f(\mathbf{x})$ as integrand. In this section we derive in which cases the proposed quadrature rule needs fewer model evaluations than a conventional quadrature rule to reach a similar accuracy. Without loss of generality, we assume that ρ and ρ_N are both probability density functions, which yields that $\|\mathcal{I}^\rho\|_\infty = \|\mathcal{I}^{\rho_N}\|_\infty = 1$.

The analysis is based on the main principle of importance sampling, a technique that is used to estimate integrals with respect to a target distribution if samples from a proposal distribution are known. Even though the main interest is not in drawing samples from the posterior, the following key idea of importance sampling can be readily used: let $\psi(\mathbf{x})$, the so-called proposal distribution, be a probability density function with the same support as ρ. Then

$$\mathcal{I}^\rho f = \int_\Omega f(\mathbf{x})\, \rho(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int_\Omega f(\mathbf{x}) \frac{\rho(\mathbf{x})}{\psi(\mathbf{x})}\, \psi(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathcal{I}^\psi \left(f \frac{\rho}{\psi} \right).$$

Hence it is not necessary to calculate a possibly expensive integral with respect to the distribution ρ, but to calculate one with respect to the proposal distribution ψ.

It is a natural idea to use the proposal distribution $\psi = \rho_N$ for the analysis, but this can severely limit the applicability of the Lebesgue inequality from (2.4), as $\|\rho/\rho_N\|_\infty$ can be large even if $\|\rho - \rho_N\|_\infty$ is small. In many cases, ρ/ρ_N can be globally bounded, for example if a uniform prior and a Gaussian likelihood is considered, ρ_N can be used as proposal distribution. An example where this is not the case is if ρ is a Beta distribution and a node is placed on the boundary of Ω. In that case ρ/ρ_N is unbounded on the boundary.

To alleviate this, we choose a slightly different approach, and use a proposal distribution ρ_N^* such that it is interpolatory and integrates the same moments as ρ_N, i.e.

$$\rho_N^*(\mathbf{x}_k) = \rho_N(\mathbf{x}_k) = \rho(\mathbf{x}_k), \text{ for } k = 0, \dots, N, \tag{4.3a}$$

and

$$\mathcal{I}^{\rho_N^*} \varphi_j = \mathcal{I}^{\rho_N} \varphi_j, \text{ for } j = 0, \dots, D_i. \tag{4.3b}$$

These two conditions define a set of possible proposal distributions, which we will call R in this section, i.e. $\rho_N^* \in R$ if it solves (4.3a) and (4.3b). See Figure 3 for a sketch how such a ρ_N^* can be interpreted. In this example, ρ_N^* is discontinuous and roughly following ρ_N, but this is not necessary for the analysis. The only conditions on ρ_N^* are (4.3a), (4.3b), and $\|\rho/\rho_N^*\|_\infty < \infty$. It is actually not necessary to construct a ρ_N^* to use the method proposed in this work: only samples from ρ_N are used. The distribution ρ_N^* is only a tool in the analysis discussed in this section. Notice that $\rho_N \in R$, so R is a well-defined non-empty set. However, as explained above it is not desirable to use $\rho_N^* = \rho_N$ in the subsequent analysis as that leads to a potentially large $\|\rho/\rho_N^*\|_\infty$.

The two conditions stated above ensure firstly that the moments of ρ_N are equal to the moments of ρ_N^*. Secondly, integrating ρ_N, ρ, or ρ_N^* using a quadrature rule with nodes $\mathbf{x}_0, \dots, \mathbf{x}_N$ yields the same result. The latter is especially useful in importance sampling, since this implies that for all $k = 0, \dots, N$:

$$1 = \frac{\rho(\mathbf{x}_k)}{\rho_N(\mathbf{x}_k)} = \frac{\rho(\mathbf{x}_k)}{\rho_N^*(\mathbf{x}_k)}.$$

This property will be used extensively in the remainder of this section. To this end, choose a $\rho_N^* \in R$, i.e. one that satisfies (4.3a) and (4.3b), as proposal distribution. Then

$$\mathcal{I}^\rho f = \mathcal{I}^{\rho_N} \left(f \frac{\rho}{\rho_N} \right) = \mathcal{I}^{\rho_N^*} \left(f \frac{\rho}{\rho_N^*} \right). \tag{4.4}$$

A similar argument can be applied to rewrite $\mathcal{A}_N^{\rho_N}$:

$$\mathcal{A}_N^{\rho_N} f = \sum_{k=0}^N w_k f(\mathbf{x}_k) = \sum_{k=0}^N w_k \frac{\rho(\mathbf{x}_k)}{\rho_N(\mathbf{x}_k)} f(\mathbf{x}_k) = \mathcal{A}_N^{\rho_N} \left(f \frac{\rho}{\rho_N} \right),$$

where it is being used that $\rho_N(\mathbf{x}_k) = \rho(\mathbf{x}_k)$ for all $k = 0, \dots, N$. Hence the following is obtained:

$$\mathcal{A}_N^{\rho_N} f = \mathcal{A}_N^{\rho_N} \left(f \frac{\rho}{\rho_N} \right) = \mathcal{A}_N^{\rho_N} \left(f \frac{\rho}{\rho_N^*} \right). \tag{4.5}$$

These expressions, i.e. the defining properties of ρ_N^*, as described by (4.3a) and (4.3b), and the derived relations for the integral and quadrature rule operator, as described by (4.4) and (4.5), form sufficient ingredients to bound $e_N(f)$. The derivation is similar to the derivation of the Lebesgue inequality (recall (2.4)), but some bookkeeping is required to keep track of the correct proposal distribution:

$$\begin{aligned} e_N(f) &= |\mathcal{I}^\rho f - \mathcal{A}_N^{\rho_N} f| \\ &= \left| \mathcal{I}^\rho f - \mathcal{A}_N^{\rho_N} \left(f \frac{\rho}{\rho_N^*} \right) \right| \\ &= \left| \mathcal{I}^\rho f - \mathcal{I}^{\rho_N^*} \varphi + \mathcal{I}^{\rho_N^*} \varphi - \mathcal{A}_N^{\rho_N} \left(f \frac{\rho}{\rho_N^*} \right) \right|. \end{aligned}$$

Here, $\varphi \in \Phi_{D_i}$. Hence the quadrature rule is exact for φ and the following is obtained:

$$e_N(f) = \left| \mathcal{I}^{\rho_N^*} \left(f \frac{\rho}{\rho_N^*} \right) - \mathcal{I}^{\rho_N^*} \varphi + \mathcal{A}_N^{\rho_N} \varphi - \mathcal{A}_N^{\rho_N} \left(f \frac{\rho}{\rho_N^*} \right) \right|,$$

where we use that $\mathcal{I}^{\rho_N^*} \varphi = \mathcal{I}^{\rho_N} \varphi = \mathcal{A}_N^{\rho_N} \varphi$ for $\varphi \in \Phi_{D_i}$, which follows from (4.3b). Concluding, the following inequality is obtained:

$$\begin{aligned} e_N(f) &\leq \left| \mathcal{I}^{\rho_N^*} \left(f \frac{\rho}{\rho_N^*} \right) - \mathcal{I}^{\rho_N^*} \varphi \right| + \left| \mathcal{A}_N^{\rho_N} \varphi - \mathcal{A}_N^{\rho_N} \left(f \frac{\rho}{\rho_N^*} \right) \right| \\ &\leq \left(\|\mathcal{I}^{\rho_N^*}\|_\infty + \|\mathcal{A}_N^{\rho_N}\|_\infty \right) \left\| f \frac{\rho}{\rho_N^*} - \varphi \right\|_\infty \\ &= 2 \left\| f \frac{\rho}{\rho_N^*} - \varphi \right\|_\infty. \end{aligned} \tag{4.6}$$

Notice that in a Bayesian framework the ratio ρ / ρ_N^* can be expressed independently from the prior:

$$\frac{\rho(\mathbf{x})}{\rho_N^*(\mathbf{x})} = \frac{q(\mathbf{z} \mid \mathbf{x})\, q(\mathbf{x})}{q_N^*(\mathbf{z} \mid \mathbf{x})\, q(\mathbf{x})} = \frac{q(\mathbf{z} \mid \mathbf{x})}{q_N^*(\mathbf{z} \mid \mathbf{x})}.$$

Here, with a little abuse of notation, $q_N^*(\mathbf{z} \mid \mathbf{x})$ is the nearest neighbor interpolant of the likelihood. Similarly to ρ_N^*, it is not necessary to gain any knowledge about $q_N^*(\mathbf{z} \mid \mathbf{x})$, since it is only used here to demonstrate independence from the prior.

By choosing $\rho_N^* \in R$ and $\varphi \in \Phi_{D_i}$ such that the obtained norm from (4.6) is minimal, an inequality similar to the Lebesgue inequality from (2.4) is obtained:

$$e_N(f) = |\mathcal{I}^\rho f - \mathcal{A}_N^{\rho_N} f| \leq 2 \inf_{\rho_N^*} \inf_{\varphi \in \Phi_{D_i}} \left\| f \frac{\rho}{\rho_N^*} - \varphi \right\|_\infty = 2 \inf_{q_N^*} \inf_{\varphi \in \Phi_{D_i}} \left\| f \frac{q(\mathbf{z} \mid \cdot)}{q_N^*(\mathbf{z} \mid \cdot)} - \varphi \right\|_\infty . \tag{4.7}$$

Here, $\rho_N^*(\mathbf{x}) = q_N^*(\mathbf{z} \mid \mathbf{x})\, q(\mathbf{x}) \in R$ is according to (4.3a) and (4.3b).

It is instructive to compare the obtained inequality with that of a conventional quadrature rule constructed with respect to the prior. If such a quadrature rule, say $\mathcal{A}_N^{q(\cdot)}$, is given, then by using the Lebesgue inequality its error can be bounded as follows (notice the difference with (4.6)):

$$|\mathcal{I}^\rho f - \mathcal{A}_N^{q(\cdot)} \left(f\, q(\mathbf{z} \mid \cdot)\right)| = |\mathcal{I}^{q(\cdot)} \left(f\, q(\mathbf{z} \mid \cdot)\right) - \mathcal{A}_N^{q(\cdot)} \left(f\, q(\mathbf{z} \mid \cdot)\right)| \leq 2 \inf_{\varphi \in \Phi_{D_i}} \| f\, q(\mathbf{z} \mid \cdot) - \varphi \|_\infty, \tag{4.8}$$

where we use that the prior is not improper (as otherwise $\|\mathcal{I}^{q(\cdot)}\|_\infty \neq 1$).

It is not generally true that the right hand side of (4.7), i.e. the error of our proposed adaptive rule, is smaller than the right hand side of (4.8), i.e. the error of conventional rules with respect to the prior. For example, the latter is smaller if both f and $q(\mathbf{z} \mid \cdot)$ are polynomials (which is rarely the case in Bayesian model calibration). In general, it is the case that a quadrature rule constructed using the iterative procedure of this article outperforms a quadrature rule constructed using the prior if f can be approximated much more efficiently by polynomials than $f q(\mathbf{z} \mid \cdot)$. To see this, we assume that the approximation of f converges *strictly* faster than the approximation of $f q(\mathbf{z} \mid \cdot)$, so using little-o notation:

$$\inf_{\varphi \in \Phi_{D_i}} \| f - \varphi \|_\infty = o \left(\inf_{\varphi \in \Phi_{D_i}} \| f\, q(\mathbf{z} \mid \cdot) - \varphi \|_\infty \right). \tag{4.9}$$

To compare both quadrature rules, recall that ρ_N is constructed such that $\rho_N \to \rho$. Then there exists a $\rho_N^* \in R$ such that

$$\|\rho / \rho_N^*\|_\infty \to 1, \text{ for } N \to \infty.$$

Hence there exist sequences $\{a_N\}_N$ and $\{b_N\}_N$ such that $a_N \leq \|\rho/\rho_N^*\|_\infty \leq b_N$ for all N with $a_N \uparrow 1$ and $b_N \downarrow 1$ for $N \to \infty$[*]. Hence

$$\begin{aligned} \inf_{\varphi \in \Phi_{D_i}} \left\| f \frac{\rho}{\rho_N^*} - \varphi \right\|_\infty &\leq \max \left(\inf_{\varphi \in \Phi_{D_i}} \| a_N f - \varphi \|_\infty ;\ \inf_{\varphi \in \Phi_{D_i}} \| b_N f - \varphi \|_\infty \right) \\ &\leq \max \left(a_N \inf_{\varphi \in \Phi_{D_i}} \| f - \varphi \|_\infty ;\ b_N \inf_{\varphi \in \Phi_{D_i}} \| f - \varphi \|_\infty \right). \end{aligned}$$

So for all $\varepsilon > 0$ there exists an N_1 such that

$$\inf_{\varphi \in \Phi_{D_i}} \left\| f \frac{\rho}{\rho_N^*} - \varphi \right\|_\infty \leq (1 + \varepsilon) \inf_{\varphi \in \Phi_{D_i}} \| f - \varphi \|_\infty, \text{ for all } N \geq N_1.$$

Thus, for $N \geq N_1$ the following bound is obtained:

$$e_N(f) \leq 2(1 + \varepsilon) \inf_{\varphi \in \Phi_{D_i}} \| f - \varphi \|_\infty. \tag{4.10}$$

By combining this bound with (4.9), it follows that (4.7) is sharper than (4.8). In other words, we have shown that the decay of the error of the adaptive quadrature rule proposed in this work, denoted by $e_N(f)$ in (4.10), depends solely on whether f can be approximated well by polynomials. Recall that whether f can be approximated well using polynomials depends mostly on its smoothness properties, see Section 2.2.1 for more discussion on this topic.

Concluding, if incorporating the likelihood in the integrand (obtaining $f q(\mathbf{z} \mid \cdot)$) yields an integrand that is difficult to approximate using polynomials, the proposed adaptive method has a sharper error bound than conventional rules. This is described by (4.9) and is often the case if the likelihood is informative

[*] In other words, a_N converges to 1 from below and b_N converges to 1 from above for $N \to \infty$.

(i.e. it has small standard deviation) or if the model is irregular (which yields an even more irregular likelihood). For large N the error of the adaptive quadrature rule proposed in this work behaves similar to that of a conventional quadrature rule constructed with respect to the *posterior*, as described by (4.10), even though this conventional quadrature rule can usually not be determined without requiring large numbers of evaluations of the posterior.

4.3 Sampling error

As an arbitrary sized sample set can be easily drawn from the distribution ρ_N for any N, the sampling error at least decays with rate $\sqrt{K}$, i.e. for $K \to \infty$, the error converges to a Gaussian distribution as follows:

$$|\mathcal{I}^{\rho_N} f - \mathcal{I}^{\rho_N}_K f| \sim \mathcal{N}(0, \sigma^2),$$

with $\sigma = \sigma(f)/\sqrt{K}$. Here, $\sigma(f)$ is the standard deviation of f with respect to ρ_N, i.e.

$$(\sigma(f))^2 = \mathcal{I}^{\rho_N}\left((f - \mathcal{I}^{\rho_N} f)^2\right).$$

In this article, acceptance rejection sampling is used to draw large numbers of samples from ρ_N. Several improvements exist to construct sequences that convergence faster, for example quasi-Monte Carlo sequences [15, 43]. We emphasize that these approaches can be used straightforwardly in the framework explained in this article.

The number of samples K can be chosen independently from the number of nodes N under consideration, so the sampling error can be made arbitrarily small without any costly model evaluations. Choosing a larger sample set does however mean that it takes more computational effort to determine the implicit quadrature rule. In practice, the interpolation and quadrature rule error dominate both the error of the quadrature rule and the cost of the procedure (through model evaluations). Only if the model is analytic and the likelihood is not informative, the quadrature rule error and interpolation error decay rapidly enough to make the sampling error dominate.

5 Numerical experiments

The properties of the method derived in this work are illustrated using various numerical examples. Two types of cases are discussed.

In Section 5.1 four different classes of analytical test cases are discussed to illustrate the key properties of our method. Firstly, in Section 5.1.1 a univariate case is discussed where the sampling and interpolation error as introduced in (4.1) can be observed well. Secondly, in Section 5.1.2 three two-dimensional test functions are calibrated using numerically generated data. The test functions, based on Genz test functions [27], are chosen such that the likelihood is very informative. Thirdly, in Section 5.1.3 the six Genz test functions are calibrated using numerically generated data. The domain of these functions is five-dimensional and a comparison is made with conventional quadrature rules such as the tensor grid and the Smolyak sparse grid. Finally, it is instructive to assess the affect of varying dimension on the error of the quadrature rule. Therefore, in Section 5.1.4 the calibration of the six Genz test functions is considered for varying dimension of Ω.

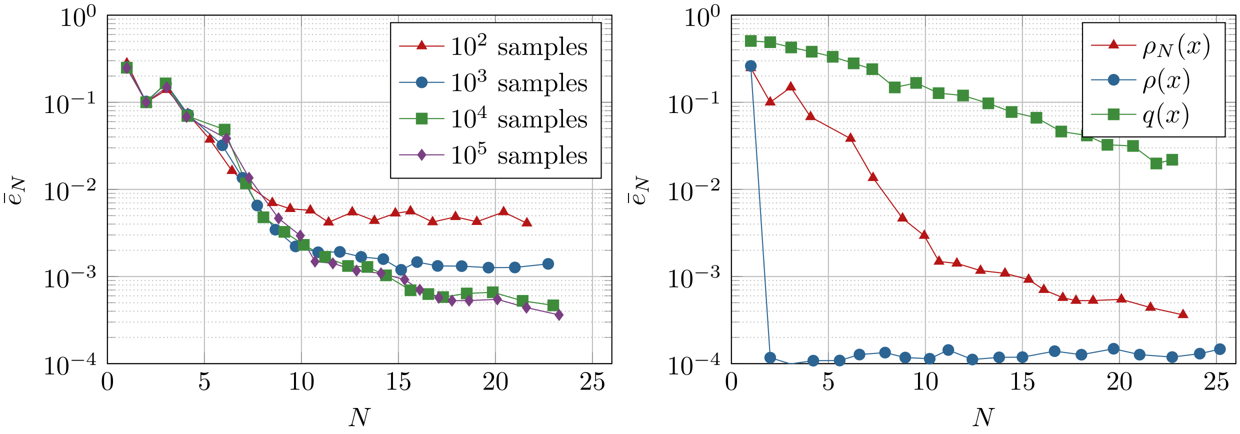
The main motivation for this work is to apply Bayesian calibration to the closure coefficients of turbulence models. To demonstrate the applicability of our method to this case, we calibrate the closure coefficients of the Spalart–Allmaras one-equation model using measurement data from the flow over a transonic airfoil. This problem is further considered in Section 5.2.

5.1 Analytical test cases

5.1.1 Effect of sampling and interpolation error

Let $\Omega = [0, 1]$ and consider a uniformly distributed prior in conjunction with a Beta$(40, 60)$-distributed likelihood, which yields:

$$\rho(x) \propto x^{40}(1 - x)^{60}.$$



(a) Effect of varying number of samples

(b) Effect of varying distribution

Figure 4: Integration error as a function of the number of nodes. In the left figure, the error of an adaptive quadrature rule is depicted for various numbers of samples. In the right figure, the error of an implicit quadrature rule constructed using various distributions is depicted, for fixed $K = 10^5$.

To assess the effect of the *sampling* error, the proposed adaptive quadrature rule is constructed using $K = 10^n$ samples with $n = 2, 3, 4, 5$. The obtained quadrature rule is used to approximate the mean of $\rho(x)$:

$$\int_\Omega \varphi_1(x)\, \rho(x)\, \mathrm{d}x = \int_\Omega x\, \rho(x)\, \mathrm{d}x \approx \sum_{k=0}^{N} w_k\, x_k = \sum_{k=0}^{N} w_k\, \varphi_1(x_k). \tag{5.1}$$

The obtained estimate is compared to the true mean (which is 2/5). The sampling error depends on the randomly generated samples, so the experiment is repeated 50 times and the obtained errors are averaged to illustrate general behavior of the error. The results are gathered in Figure 4a, where the averaged error is denoted by $\bar{e}_N$.

For small N, it is clear that the errors of all quadrature rules converge geometrically, which implies that the quadrature error or the interpolation error dominates. For large N, the error does not decay further due to the limited number of samples. The point where the error stops decaying depends on the number of samples used, i.e. a larger number of samples results in a smaller overall error.

To assess the effect of the *interpolation* error, three different implicit quadrature rules are constructed. The first rule is constructed using the adaptive method proposed in this work, i.e. using $\rho_N(x)$ like in Figure 4a. This quadrature rule has a sampling error and interpolation error, but no quadrature error since we are comparing the first moment (see (5.1)). The second rule is constructed using the *exact* distribution, i.e. $\rho(x)$, which results in a rule that only has a sampling error. The third rule is constructed using the uniform distribution, i.e. using $q(x)$. This rule has a sampling error and a quadrature error, as the likelihood needs to be incorporated in the integrand. All three quadrature rules are used to approximate the mean of the posterior. The rules constructed using $\rho_N(x)$ and $\rho(x)$ can estimate the mean using (5.1), whereas the rule that is constructed using $q(x)$ requires integration of $f(x) = \rho(x) \cdot x$. All rules are constructed using 10^5 samples and again the experiment is repeated 50 times to demonstrate general behavior of the rules. The results are gathered in Figure 4b.

The rule constructed using samples from $\rho(x)$ immediately saturates to a level where the sampling error dominates, as no interpolation is used and the quadrature rule error vanishes since the error estimate from (5.1) is based on the mean (i.e. the first moment). The rule constructed using the prior $q(x)$ converges geometrically, as the PDF of the Beta(40, 60)-distribution is a polynomial. A significant improvement is noticed by using our proposed adaptive proposal distribution $\rho_N(x)$. Initially a larger rate of convergence is obtained (up to approximately $N = 10$), as the interpolant provides rapid convergence. For large N, the quadrature error starts dominating and the rate of convergence equals the rate of the rule constructed using $q(x)$.

This test case demonstrates that using samples that have been obtained directly from $\rho(x)$ is preferable, but this is intractable in practice. The adaptive proposal distribution has a clear advantage over using a rule that only incorporates the prior.

5.1.2 Two-dimensional Genz test functions

The previously considered test case of Section 5.1.1 demonstrates the advantages of using an adaptive quadrature rule. To illustrate the nodal placement of this rule and to further compare this rule to conventional quadrature rules constructed with respect to the prior, the calibration of three two-dimensional Genz test functions [27] with artificially generated data is considered. Therefore let $d = 2$, $\mathbf{x}^* = (1/2, 1/2)^{\mathrm{T}}$, and consider the following functions:

$$\begin{aligned} u_1(\mathbf{x}) &= \prod_{i=1}^{d} \left[\frac{1}{4} + \left(x_i - \frac{1}{2} \right)^2 \right]^{-1}, && \text{(Product Peak)} \\ u_2(\mathbf{x}) &= \exp\left(- \sum_{i=1}^{d} \left| x_i - \frac{1}{2} \right| \right), && \text{(C0 function)} \\ u_3(\mathbf{x}) &= \begin{cases} 0 & \text{if } x_1 > 3/5 \text{ and } x_2 > 3/5, \\ \exp\left(\sum_{i=1}^{d} x_i \right) & \text{otherwise.} \end{cases} && \text{(Discontinuous function)} \end{aligned}$$

These functions are specifically crafted such that they have difficult characteristics for numerical integration routines at or around $\mathbf{x} = \mathbf{x}^*$. The statistical model under consideration is

$$z_k = u_g(\mathbf{x}) + \varepsilon, \text{ with } \varepsilon \sim \mathcal{N}(0, \sigma^2) \text{ and } \sigma^2 = 1/5, \text{ for } k = 1, \dots, 20. \tag{5.2}$$

Here, u_g (with $g = 1, 2, 3$) denotes one of the functions under consideration and $\mathbf{z} = (z_1, \dots, z_{20})^{\mathrm{T}}$ are 20 "measurements" obtained by $z_k = u_g(\mathbf{x}^*) + n_k$ with n_k a sample from $\mathcal{N}(0, \sigma^2)$. Hence the data is centered around the defining ("difficult") characteristics of the test functions.

The goal is to infer $\mathbf{x}$ from (5.2), or equivalently, characterize the posterior $q(\mathbf{x} \mid \mathbf{z})$ with a uniform prior and Gaussian likelihood defined in $\Omega = [0, 1]^d$, i.e. we use $\rho(\mathbf{x}) = q(\mathbf{z} \mid \mathbf{x})\, q(\mathbf{x})$ with

$$\begin{aligned} q(\mathbf{x}) &= \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega, \\ 0 & \text{otherwise,} \end{cases} \\ q(\mathbf{z} \mid \mathbf{x}) &\propto \exp\left[-\frac{1}{2} \frac{\sum_{k=1}^{20} (u(\mathbf{x}) - z_k)^2}{\sigma^2} \right]. \end{aligned} \tag{5.3}$$

The quantity used to measure the error of the quadrature rules is the maximum error of the mean of the posterior, i.e.

$$e_N = \begin{cases} \max\left(|\mathcal{A}_N \varphi_1 - \mathcal{I}^\rho \varphi_1|,\ |\mathcal{A}_N \varphi_2 - \mathcal{I}^\rho \varphi_2| \right) & \text{if } \mathcal{A}_N \text{ w.r.t. } \rho_N(\mathbf{x}), \\ \max\left(\left| \mathcal{A}_N \left(\varphi_1 q(\mathbf{z} \mid x_1) \right) - \mathcal{I}^\rho \varphi_1 \right|,\ \left| \mathcal{A}_N \left(\varphi_2 q(\mathbf{z} \mid x_2) \right) - \mathcal{I}^\rho \varphi_2 \right| \right) & \text{if } \mathcal{A}_N \text{ w.r.t. } q(\mathbf{x}), \end{cases}$$

where $\mathcal{A}_N$ denotes a quadrature rule operator constructed using the adaptive method proposed in this article (i.e. with respect to ρ_N) or solely using the prior (i.e. with respect to $q(\mathbf{x})$). In both cases the weights are scaled such that $\sum_k w_k = 1$. All quadrature rules are constructed using a linearly growing D_i, i.e. in iteration i it is enforced that the quadrature rule integrates all $\varphi \in \Phi_i$ exactly (with Φ_i as introduced in Section 2.2.1). Recall that if $\mathbf{x} = (x_1, x_2)$ we have that $\varphi_1(\mathbf{x}) = x_1$ and $\varphi_2(\mathbf{x}) = x_2$.

There is randomness both in the generation of the quadrature rules and in the generation of the measurement data. Therefore, the error e_N is determined 25 times and the obtained errors are averaged, which is denoted by $\bar{e}_N$. The convergence results are summarized in Figure 5. Examples of quadrature rules obtained by applying the proposed procedure are depicted in Figure 6. There are still oscillations visible, regardless of the averaging, though the error decay can clearly be observed.

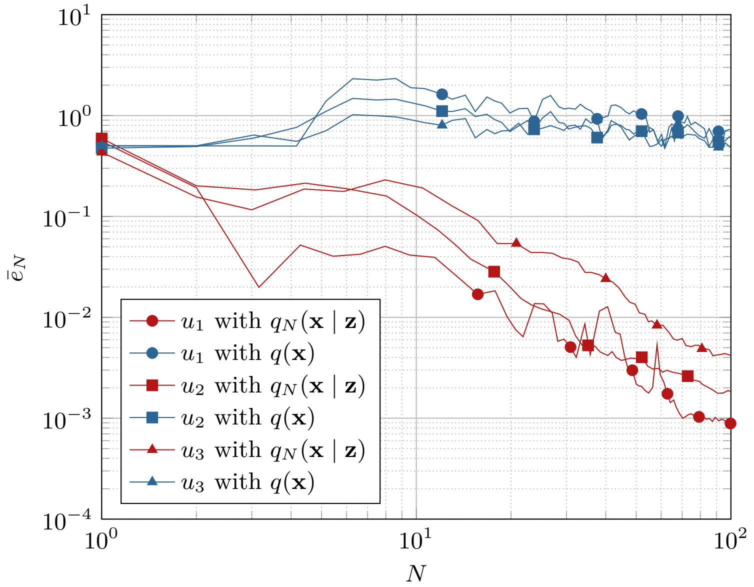
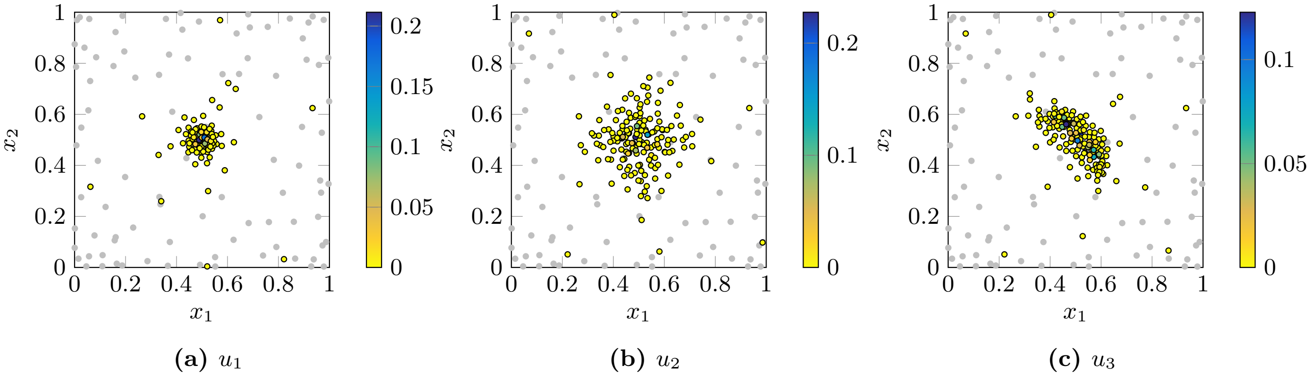


Figure 5: Convergence of the mean error determined with the new adaptive implicit quadrature rule or a quadrature rule determined using the prior. Equal colors refer to the same quadrature rule. Equal symbols refer to the same function.



(a) u_1

(b) u_2

(c) u_3

Figure 6: Examples of quadrature rules with $D = 65$ (exact for all polynomials up to degree 10) obtained by the adaptive implicit quadrature rule. The colors indicate the weights of the quadrature rule nodes. The nodes of a quadrature rule of the same polynomial degree, but constructed with respect to the prior, are depicted in gray.

Notice that the adaptive implicit quadrature rule outperforms the conventional quadrature rule in all three cases. The scatter plots of the nodes illustrate that the quadrature rules are indeed tailored specifically to the posterior under consideration. Since all rules have positive weights, estimates determined using these rules are unconditionally stable. This is an aforementioned advantage of the proposed method: the quadrature rules exhibit both exploitation (due to the interpolant) and exploration (due to the spread of the nodes) without suffering from instability.

5.1.3 Five-dimensional Genz test functions

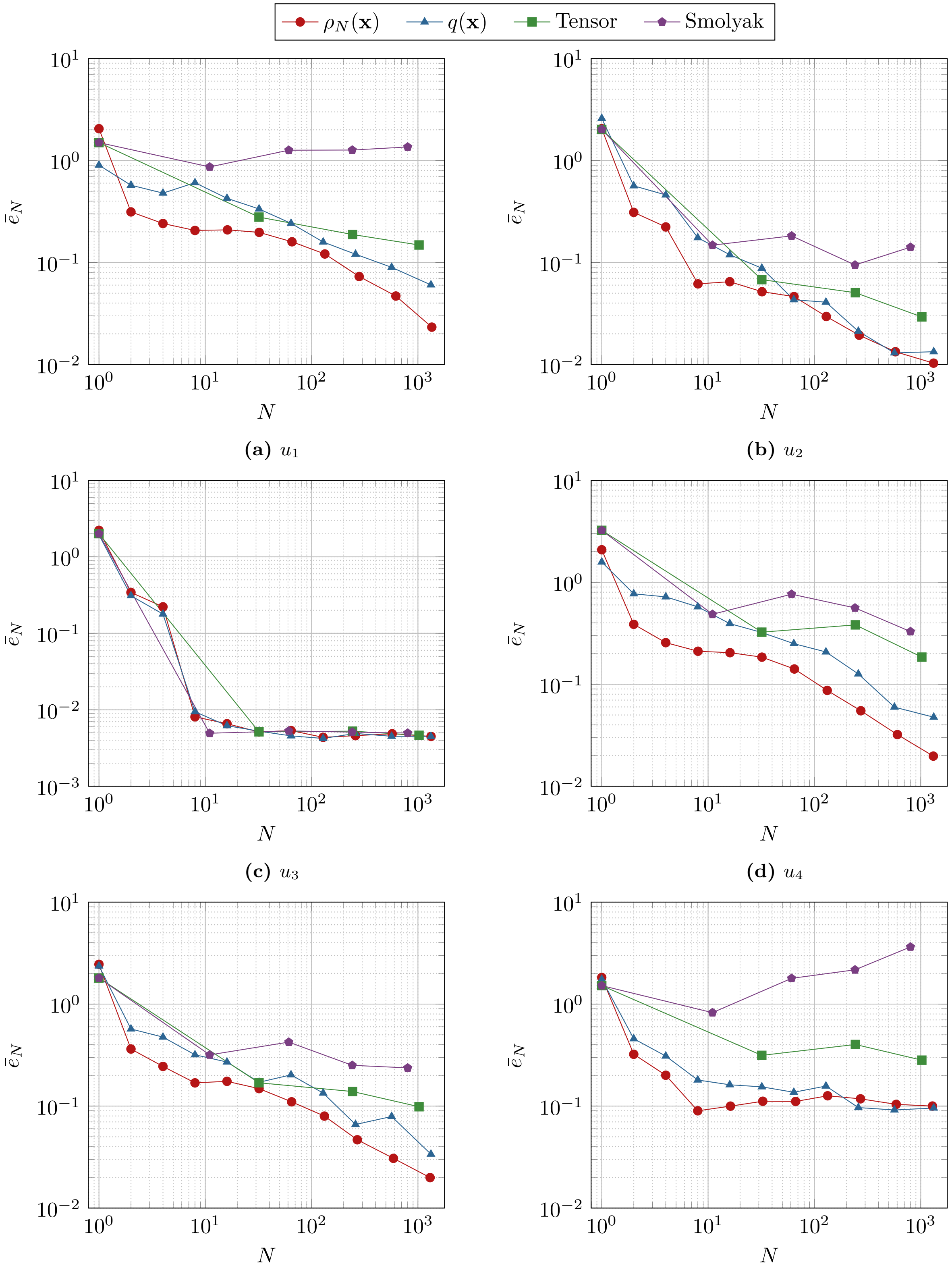
In this section, the calibration of the five-dimensional Genz test functions is considered. The setting is similar to the one of the previous section, i.e. let $d = 5$, $\mathbf{x}^* \in \mathbb{R}^d$ with $x_k^* = 1/2$, and consider the following functions:

$$
\begin{aligned}
u_1(\mathbf{x}) &= \cos\left(2\pi b_1 + \sum_{i=1}^{d} a_i x_i\right), && \text{(Oscillatory)}\\
u_2(\mathbf{x}) &= \prod_{i=1}^{d} \left(a_i^{-2} + (x_i - b_i)^2\right)^{-1}, && \text{(Product Peak)}\\
u_3(\mathbf{x}) &= \left(1 + \sum_{i=1}^{d} a_i x_i\right)^{-(d+1)}, && \text{(Corner Peak)}\\
u_4(\mathbf{x}) &= \exp\left(-\sum_{i=1}^{d} a_i^2 (x_i - b_i)^2\right), && \text{(Gaussian)}\\
u_5(\mathbf{x}) &= \exp\left(-\sum_{i=1}^{d} a_i |x_i - b_i|\right), && \text{(C_0 function)}\\
u_6(\mathbf{x}) &= \begin{cases} 0 & \text{if } x_1 > b_1 \text{ or } x_2 > b_2,\\ \exp\left(\sum_{i=1}^{d} a_i x_i\right) & \text{otherwise.}\end{cases} && \text{(Discontinuous)}
\end{aligned}
$$

The vectors $\mathbf{a} = (a_1, \ldots, a_d)^{\mathrm{T}}$ and $\mathbf{b} = (b_1, \ldots, b_d)^{\mathrm{T}}$ are shape and translation vectors respectively and are chosen randomly in this test case. The elements of $\mathbf{a}$ enlarge the defining "difficult" property of the function and $\mathbf{b}$ translates the function in space. The statistical model under consideration is the same model considered in the previous section, see (5.2) and (5.3). Notice that, due to the random nature of $\mathbf{b}$, the probability that the difficult property of the functions is around $\mathbf{x} = \mathbf{x}^*$ is small.

Four quadrature rules are considered to determine the mean of the posterior. Firstly, the proposed quadrature rule with a nearest neighbor interpolant is used. At each iteration the exactness of the quadrature rule is doubled, starting with $D_0 = 2^0$ up to $D_{10} = 2^{10}$. In other words, subsequent quadrature rules integrate larger numbers of polynomials (recall that Φ_D denotes the space of $D + 1$ polynomials, sorted graded lexicographically). Secondly, an implicit quadrature rule without interpolation is used, generated using solely the prior. Finally, to compare the results with conventional quadrature rules, a tensor grid and a Smolyak sparse grid generated with a Clenshaw–Curtis quadrature rule are used. A linearly growing sequence of Clenshaw–Curtis quadrature rules is considered, so the tensor grid and the Smolyak sparse grid do not form a sequence of nested quadrature rules. This is done to keep the number of nodes of the multivariate quadrature rules tractable. We do not present the results obtained with Markov chain Monte Carlo, since the number of model evaluations considered here is too small to obtain a "burned-in" sequence.

For each quadrature rule, the experiment is repeated 25 times and the reported errors are averaged. For each experiment, the vectors $\mathbf{a}$ and $\mathbf{b}$ are drawn randomly from the five-dimensional unit hypercube and $\mathbf{a}$ is subsequently scaled such that $\|\mathbf{a}\|_2 = 5/2$. Moreover, for each experiment different samples are used to generate the implicit quadrature rules. To introduce randomness in the tensor grid and Smolyak sparse grid, the "exact" value $\mathbf{x}^*$ is randomized in the domain. This *improves* the results of calibration with the Clenshaw–Curtis rules, as it otherwise would have a node at $\mathbf{x} = \mathbf{x}^*$ and therefore significantly overestimate the moments. The quantity used to measure the error of the quadrature rules is the same as in the previous test case, i.e. e_N is the maximum error of all first-order raw moments (which are $\varphi_1, \ldots, \varphi_5$). The "exact"



(a) u_1

(b) u_2

(c) u_3

(d) u_4

(e) u_5

(f) u_6

Figure 7: Convergence of the mean of the posterior determined using three different quadrature rules. The six functions under consideration are the Genz test functions.

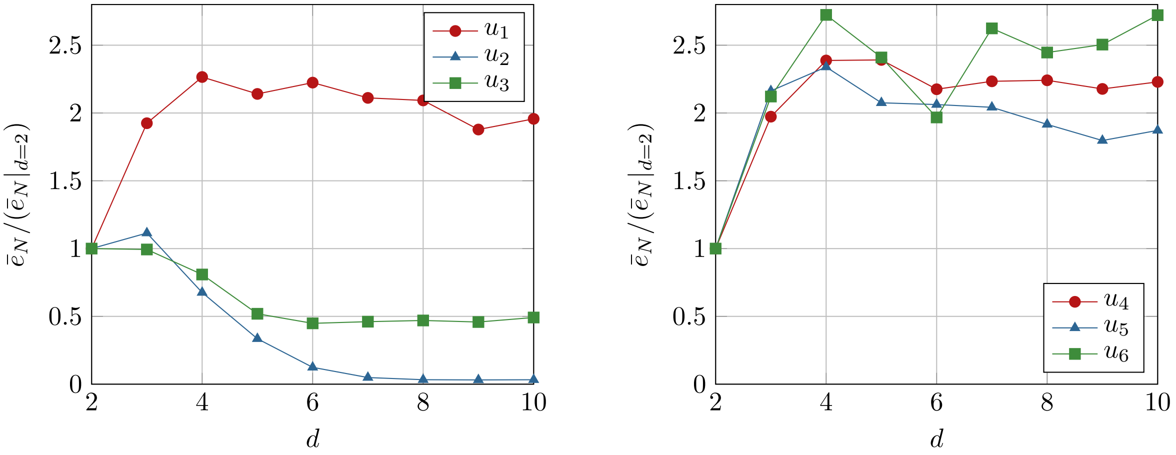


Figure 8: The integration error of the mean, averaged over 100 runs, considering the Genz test functions for varying dimension.

mean is determined using a Monte Carlo sample from the prior. The number of samples used is the same number used to generate the implicit quadrature rules, such that the sampling error can be observed well (in that case all errors saturate). The obtained results are depicted in Figure 7.

The first four test functions are analytic, so for these test functions the quadrature rules perform best. The flattening of the error of the third test function occurs due to the sampling error of the *exact* mean (and not due to the sampling error of the quadrature rule). Moreover, the results of the third test function are best, since the corner peak varies only near a corner (hence the name) and therefore yields an almost uniform posterior.

The fifth Genz test function is only continuous (and not smooth), so should theoretically yield inferior performance. However, it is not differentiable only at a single point in the five-dimensional domain, so it is very unlikely that that point is of importance for the accuracy of the quadrature rules.

The sixth Genz test function is discontinuous and all rules do not yield a rapidly converging estimate. This is expected, since the quadrature rules are based on approximation using smooth functions. Notice that the convergence of the proposed adaptive procedure is significantly worse than in the two-dimensional case considered in Section 5.1.2, since detecting the discontinuity in the posterior using nearest neighbor interpolation in five dimensions requires significantly more model evaluations than in two dimensions. However, the accuracy remains similar to the other quadrature rules.

Overall, the adaptive quadrature rule proposed here consistently performs better or on the same level as the other quadrature rules. This demonstrates that the proposed procedure works for non-informative posteriors, since it is rarely the case that the posterior of the Genz test functions in conjunction with the randomized parameters is informative. Intuitively one would expect that piecewise interpolation is in these cases not necessarily beneficial, but the results demonstrate that it does not deteriorate the performance.

The Smolyak sparse grid performs significantly worse than the other quadrature rules for u_1 and u_6. This happens due to the negative weights of the sparse grid, or more specifically, due to the large positive weights present in the Smolyak sparse grid. These large weights significantly deteriorate the accuracy for a small number of the 25 runs, which is reflected in the averaged errors presented in Figure 7. Notice that this effect can be significantly alleviated with an adaptive Smolyak sparse grid, as considered by Schillings and Schwab [50], though negative weights will remain present.

5.1.4 d-dimensional Genz test functions

It is well-known that the convergence rate of quadrature rules based on polynomial approximation deteriorates in larger dimensional spaces. This follows among others from the Lebesgue inequality, see (2.4), since approximating multivariate functions requires larger numbers of basis polynomials. It is not straightforward to precisely assess the effect of the dimension on the accuracy of the quadrature rule, since the properties of the statistical model and the Genz test functions also depend on the dimension. However, if the accuracy of the quadrature rule deteriorates rapidly, the quadrature rule error will dominate the overall integration error, which is the quantity reported in this work.

Therefore, the Genz test functions are reconsidered in combination with the previously discussed statistical model, see (5.3). The key difference is that the experiment is repeated for different dimensions d. Again the mean integration error $\bar{e}_N$ is computed, but contrary to the previous cases the error is determined by averaging over 100 experiments, since the error varies significantly less (and otherwise noise would pollute the results). The mean integration error is computed using a quadrature rule of degree 20, which is determined iteratively using a linearly growing sequence $D_i = i$ (hence 20 iterations of the algorithm are necessary to construct a quadrature rule). All other parameters are chosen as in Section 5.1.3. The results are reported in Figure 8, where the error is scaled with the integration error of the bivariate case (i.e. $d = 2$).

The effect of the dimension on the statistical model and the Genz test functions is clearly visible. For u_2 and u_3, the error is *decreasing* for increasing dimension, which conflicts with the theoretical expectations. This is due to the region that encompasses the defining properties of the product peak (i.e. u_2) and corner peak (i.e. u_3), which becomes smaller in multivariate spaces. This effect is also observed in the results of Section 5.1.3. The other functions yield a rapidly increasing error for $d \leq 5$, but the error stagnates (or even decreases slightly) for larger d. Again, this is the result of the varying properties of the test functions and the statistical model.

Even though it is not straightforward to isolate the error of the quadrature rule, this test case demonstrates that the growth of the quadrature error for increasing dimension does not dominate the growth of the integration error. In other words, the effect of dimension on the quadrature rule is significantly less than the effect of dimension on the test functions and the statistical model. Therefore the quadrature rules yield a viable approach for Bayesian prediction in multivariate spaces.

5.2 Transonic flow over an airfoil

The applicability of the approach to complex test cases is demonstrated by considering the transonic flow over the RAE2822 airfoil. The problem under consideration is the calibration of the closure parameters of the Spalart–Allmaras turbulence model using wind tunnel measurements. It is computationally expensive to determine an accurate numerical solution of this problem, so the usage of efficient calibration approaches is of importance.

This section is split into three parts. Firstly, the problem of modeling the transonic flow over an airfoil and the measurement data under consideration is briefly introduced in Section 5.2.1. Secondly, the statistical model under consideration that relates the measurement data to the model output is discussed in Section 5.2.2. The obtained Bayesian predictions are presented in Section 5.2.3.

5.2.1 Modeling the flow over an airfoil

The flow over the airfoil under consideration is modeled using the Reynolds-averaged Navier–Stokes (RANS) equations, which only model the time-averaged mean flow over the airfoil. These equations do not form a closed system and require a closure model. For this purpose, the Boussinesq hypothesis is used, which introduces an eddy viscosity, which is subsequently modeled using a turbulence model. The details of this derivation are out of the scope of this article and we refer the interested reader to Wilcox [61]. The turbulence model under consideration is the Spalart–Allmaras model [55], which is commonly used to model the flow over an airfoil. Various variants of the model exists, though in this work the original most commonly used variant from Spalart and Allmaras [55] is considered. The model has 10 constants, which are typically defined as follows:

σ	2/3	C_{b1}	0.1355	C_{b2}	0.622
κ	0.41	C_{w2}	0.3	C_{w3}	2.0
C_{t3}	1.2	C_{t4}	0.5	C_{v1}	7.1

Here C_{w1} is omitted, since it depends on the other coefficients by the following expression:

$$C_{w1} = C_{b1}/\kappa^2 + (1 + C_{b2})/\sigma.$$

The closure coefficients of turbulence models are usually obtained by fitting the model using canonical test cases, for which analytical or accurate numerical solutions are available. There is no clear physical interpretation of many of these coefficients. A more structured approach to obtain values for these coefficients is to calibrate these coefficients statistically using Bayesian model calibration [6, 16, 22], which has been discussed in this article. The posterior can be used to infer credible intervals of the parameters and can be propagated to infer Bayesian predictions.

A major advantage of the approach discussed in this article is that the determined quadrature rule can directly be used to infer prediction of the flow incorporating the uncertainty of the closure coefficients. The costly model evaluations have already been performed during the construction of the rule. No Markov chain Monte Carlo routines are necessary to obtain the predictions.

In this article, SU2 [45] is employed to numerically solve the RANS equations. It uses a second-order finite volume discretization and has support for the Spalart–Allmaras turbulence model. Moreover, it has been tested extensively to model the flow over an airfoil and the turbulence model parameters can be made configurable due to its freely available source code. As mentioned before, the airfoil under consideration is the RAE2822. Measurements of this airfoil for various values of the Reynolds number, Mach number, and angle of attack are freely available [17]. We consider Case 6 (see Cook et al. [17] for the other cases), with Reynolds number $6.5 \cdot 10^6$, Mach number 0.725, and angle of attack 2.92°. The measurements encompass $n_z = 103$ point measurements of the pressure coefficient on the surface of the airfoil. To use these values in conjunction with SU2, we correct these values to incorporate wind tunnel effects [53]. The flow over the RAE2822 airfoil is transonic in this case and there is shock formation with a supersonic regime on the upper part of the airfoil.

5.2.2 Statistical model

The quantity of interest used for the calibration is the pressure coefficient on the surface of the airfoil. Following Edeling et al. [22], the parameters considered for calibration are $\boldsymbol{\vartheta} = [\kappa, \sigma, C_{b1}, C_{b2}, C_{v1}, C_{w2}, C_{w3}]^{\mathrm{T}}$. Let $s \in [0, 2]$ be the spatial parameter that runs from the trailing edge in anticlockwise direction over the airfoil. Then $u(\boldsymbol{\vartheta}; s)$ denotes the mapping that yields the pressure coefficient at s using the turbulence parameters $\boldsymbol{\vartheta}$. We denote the calibration parameters by $\boldsymbol{\vartheta}$ (instead of $\mathbf{x}$) to avoid confusion between $\boldsymbol{\vartheta}$ and the usual spatial coordinates x and y. A single evaluation of SU2 yields the pressure coefficient at all spatial coordinates in the discrete mesh for given $\boldsymbol{\vartheta}$ and we use linear interpolation to obtain the pressure coefficient at the measurement locations.

The statistical model under consideration incorporates both model uncertainty, denoted by δ, and measurement error, denoted by ε. It is, following Kennedy and O'Hagan [36], as follows:

$$z_k = u(\boldsymbol{\vartheta}; s_k) + \delta(s_k) + \varepsilon_k, \text{ with } \varepsilon_k \sim \mathcal{N}(0, \sigma^2) \text{ for } k = 1, \dots, n_z. \tag{5.4}$$

Here, z_k denotes the measured pressure coefficient at location s_k. The random process $\delta(s) \sim \mathcal{N}(0, \mathrm{Cov}(s, s' \mid A, l))$ models the discrepancy between the model and the truth by means of a Gaussian process with zero mean and squared exponential covariance:

$$\mathrm{Cov}(s, s' \mid A, l) = A \exp\left[-\left(\frac{s - s'}{L10^l}\right)^2\right].$$

The values of A and l are hyperparameters and are being inferred from calibration whereas L is a fixed parameter whose value is provided later. Measurement error is modeled by ε_k, which is Gaussian distributed with known standard deviation $\sigma = 0.01$. This value is based on the measurement errors reported by Cook et al. [17].

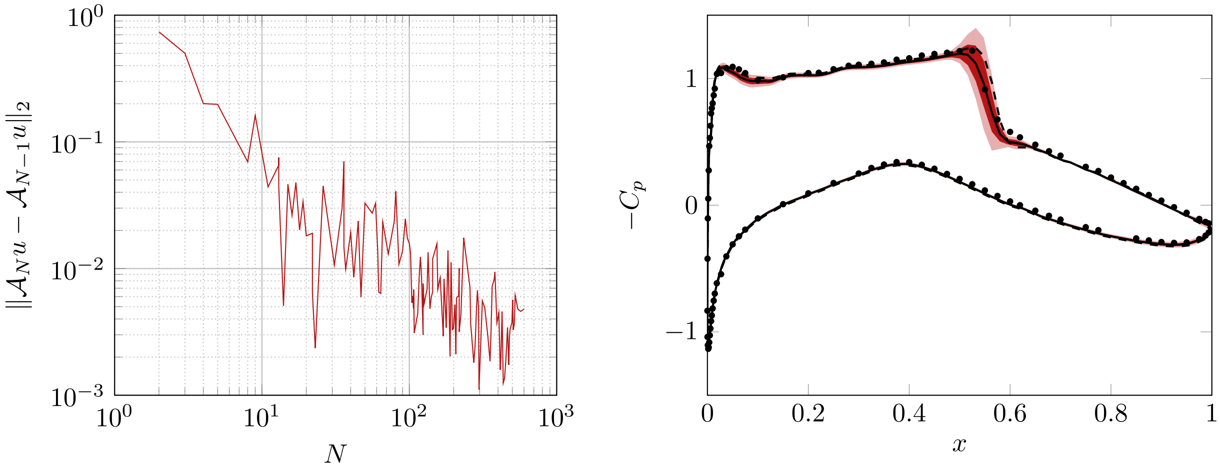
The model from (5.4) yields the following likelihood:

$$q(\mathbf{z} \mid \boldsymbol{\vartheta}, A, l) \propto \exp\left[-\frac{1}{2}\mathbf{d}^{\mathrm{T}} K^{-1} \mathbf{d}\right], \tag{5.5}$$

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

$$\begin{aligned} d_k &= z_k - u(\boldsymbol{\vartheta}; s_k), \\ K &= \Sigma + C, \text{ with } C_{i,j} = \mathrm{Cov}(s_i, s_j \mid A, l) \text{ and } \Sigma = \sigma I. \end{aligned}$$

The prior of the turbulence closure parameters is as follows:



(a) Error behavior

(b) $C_p - x$

Figure 9: *Left:* convergence of consecutive differences, *Right:* Mean and standard deviation of $C_p - x$ determined using the proposed approach (solid line and red area), compared to using canonical coefficients (dashed) and measurement data (scattered)

κ	$[0.205, 0.615]$
σ	$[1/3, 1]$
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}	$[1, 3]$

These values are such that they encapsulate all commonly used variants of the turbulence model and are such that SU2 converges for all possible combinations.

The hyperparameters A and l are also being calibrated and therefore also require a prior. These are respectively $A \sim \mathcal{U}(0, 0.01)$ and $l \sim \mathcal{U}(0, 1)$. We choose $L = 0.01$, such that the maximal covariance length is significantly larger than the length of a single numerical cell on the surface of the airfoil. The computational model does not depend on A, l, and L, so given $\boldsymbol{\vartheta}$ the distribution of these parameters is fully known analytically.

By combining the prior with the likelihood from (5.5), Bayes' law yields the posterior:

$$q(\boldsymbol{\vartheta}, A, l \mid \mathbf{z}) \propto q(\mathbf{z} \mid \boldsymbol{\vartheta}, A, l)\, q(\boldsymbol{\vartheta})\, q(A)\, q(l).$$

We are not interested in the exact distribution of the hyperparameters and the evidence can be neglected by rescaling the quadrature rule weights. Hence we apply the adaptive implicit quadrature rule to the following distribution:

$$\rho(\boldsymbol{\vartheta}) = \iint q(\mathbf{z} \mid \boldsymbol{\vartheta}, A, l)\, q(\boldsymbol{\vartheta})\, q(A)\, q(l) \,\mathrm{d}l \,\mathrm{d}A.$$

Afterwards, the quadrature rule weights are scaled such that $\sum_{k=0}^{N} w_k = 1$ and the rule is used to infer Bayesian predictions, e.g. as derived in (2.2):

$$\mathbb{E}[\widehat{\mathbf{z}} \mid \mathbf{z}](s) = \int_\Omega u(\boldsymbol{\vartheta}, s)\, q(\boldsymbol{\vartheta} \mid \mathbf{z}) \,\mathrm{d}\boldsymbol{\vartheta} \approx \sum_{k=0}^{N} w_k\, u(\boldsymbol{\vartheta}_k, s), \text{ with } q(\boldsymbol{\vartheta} \mid \mathbf{z}) \propto \rho(\boldsymbol{\vartheta}).$$

5.2.3 Results

The quadrature rule constructed in this test case has polynomial degree 3, which constitutes a polynomial space with 120 basis polynomials. This number of polynomials is a good balance between accuracy in the predictions and overall running time. The quadrature rules are constructed with linearly increasing exactness, so at the i-th iteration a quadrature rule of degree i is constructed that incorporates all available model evaluations (up to iteration $i-1$).

The obtained rule incorporates an approximation of the posterior. There is no exact solution available, so it is not straightforward to assess the accuracy of the quadrature rule. For this purpose, each iteration the mean of the pressure coefficient is determined and these are consecutively compared using the 2-norm, i.e. if N_i is the number of nodes of the quadrature rule after i iterations, then the error measure is

$$e_{N_i} = \|\mathcal{A}_{N_i} u - \mathcal{A}_{N_{i-1}} u\|_2.$$

We will denote this with a little abuse of notation simply as

$$e_N = \|\mathcal{A}_N u - \mathcal{A}_{N-1} u\|_2.$$

The obtained errors are depicted in Figure 9a.

The quadrature rule clearly yields a converging mean and the error converges approximately linearly to zero. The oscillations are the result of the random sampling of the proposal distributions in combination with the fact that this test case was run only once. Obtaining a smoother decay would require repeating the experiment various times and averaging the obtained error (as done in the previous test cases), but this is too computationally costly in this case. In practical computations, the error decay can be assessed by regressing the convergence rate by means of least squares (i.e. by fitting analytic error decay).

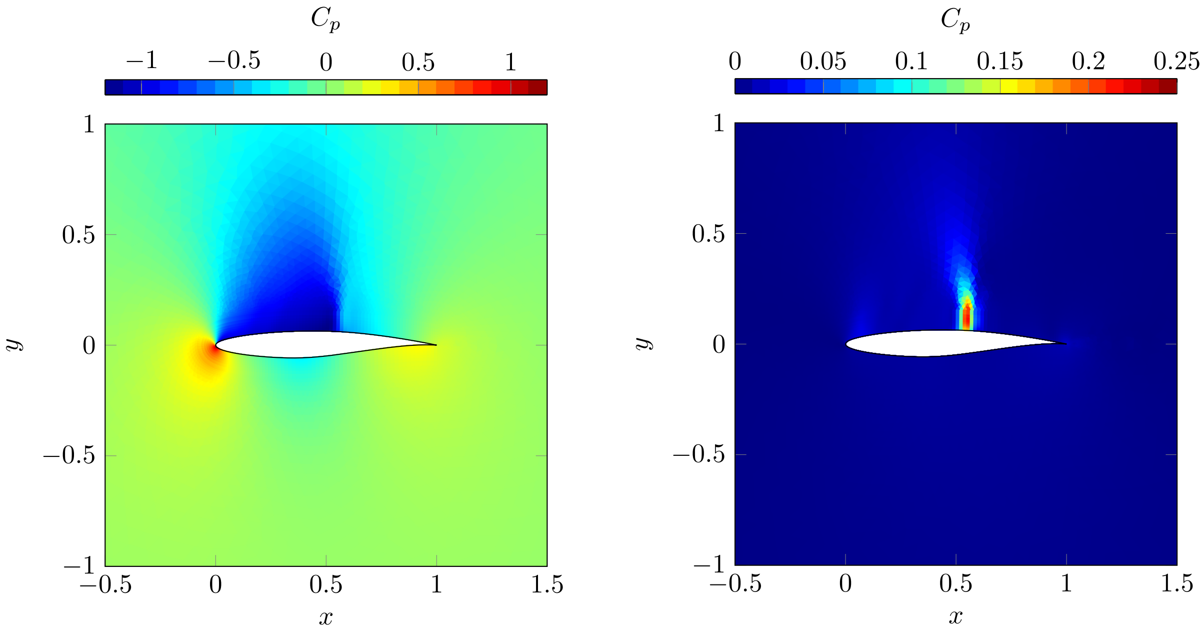
Each iteration SU2 is evaluated for each additional quadrature rule node, so obtaining Bayesian predictions of the pressure coefficient is a straightforward post-processing step. The obtained predictions of the pressure coefficient are depicted in Figure 9b. The measurement data, uncertainty bounds, and the deterministic pressure coefficient determined using the canonical turbulence coefficients are also depicted. It is clearly visible that the largest uncertainty occurs near the shock on top of the airfoil, which is in line with existing results on uncertain transonic flows around the RAE2822 airfoil, either using uncorrelated prescribed distributions [47, 62] or calibrated parameters in a Bayesian setting [6]. Notice that the number of evaluations of SU2 needed to infer these predictions is of similar order of magnitude as is common for uncertainty propagation with *prescribed* distributions. It is a significant reduction compared to commonly used Markov chain Monte Carlo methods [16, 21, 22].

SU2 yields the full flow field around the airfoil and therefore it is also possible to predict the statistical moments of the pressure coefficient away from the airfoil surface, as depicted in Figure 10. We want to emphasize the importance of positive weights, since these ensure that the prediction of the variance is positive. Notice that the pressure coefficients depicted in these figures (i.e. away from the airfoil) have *not* been used in the calibration process, as only measurement data on the airfoil surface is available. So technically it not evident whether these results are correct or not. It is not straightforward to make general claims about this, since the specific properties (such as smoothness) of the model under consideration determine to a large extent whether it is viable to infer predictions of quantities using parameters that have been calibrated with a different quantity.

6 Conclusions

In this article a new methodology is proposed to construct quadrature rules with positive weights and high degree for the purpose of inferring Bayesian predictions. It consists of two steps. Firstly a quadrature rule is constructed using an approximation of the posterior. For this step the so-called implicit quadrature rule has been used. Secondly the approximation of the posterior is refined using all available model evaluations. For the second step nearest neighbor interpolation has been used. It has been demonstrated rigorously that our approach yields a quadrature rule whose error behaves like a quadrature rule with respect to the exact posterior for increasing number of nodes.

The performance of the quadrature rule has been demonstrated by calibrating the Genz test functions in a Bayesian framework. The results demonstrate that the quadrature rules consistently outperform (or perform as well as) conventional numerical integration approaches. The performance gains are most significant if the posterior is very informative, e.g. if its standard deviation is small. If, on the other hand, the posterior



(a) Mean

(b) Standard deviation

Figure 10: Predictions of pressure coefficients in vicinity of the airfoil using calibrated coefficients.

resembles approximately the prior, the performance is comparable to a quadrature rule determined using solely the prior.

The applicability of the approach to an expensive fluid dynamics model has been demonstrated by using the quadrature rule to calibrate the transonic flow over the RAE2822 airfoil. It has been demonstrated numerically that estimates of the rule converge and predictions of the mean and standard deviation of the pressure coefficient have been inferred. The results are in line with existing research, though significantly less model evaluations are necessary compared to existing Bayesian calibration approaches.

There are various opportunities to further extend the approach. The nearest neighbor interpolation procedure does not leverage smoothness in either the distribution or the posterior. A major opportunity is replacing nearest neighbor interpolation by a more efficient interpolation approach, which could allow for more rapid convergence. This is however far from trivial, since our method requires that the obtained interpolant is a distribution.

Acknowledgments

This research is part of the Dutch EUROS program, which is supported by NWO domain Applied and Engineering Sciences and partly funded by the Dutch Ministry of Economic Affairs.

References

[1] S. Ahn, A. Korattikara, and M. Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[2] G. Biau and L. Devroye. *Lectures on the Nearest Neighbor Method*. Springer International Publishing, 2015. `doi:10.1007/978-3-319-25388-6`.

[3] I. Bilionis and N. Zabaras. Solution of inverse problems with limited forward solver evaluations: a Bayesian perspective. *Inverse Problems*, 30(1):015004, 2013. `doi:10.1088/0266-5611/30/1/015004`.

[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] L. M. M. van den Bos, B. Koren, and R. P. Dwight. Non-intrusive uncertainty quantification using reduced cubature rules. *Journal of Computational Physics*, 332:418–445, 2017. `doi:10.1016/j.jcp.2016.12.011`.

[6] L. M. M. van den Bos, B. Sanderse, W. A. A. M. Bierbooms, and G. J. W. van Bussel. Bayesian model calibration with interpolating polynomials based on adaptively weighted Leja nodes. *Communications in Computational Physics*, 27(1):33–69, 2020. `doi:10.4208/cicp.oa-2018-0218`.

[7] L. M. M. van den Bos, B. Sanderse, W. A. A. M. Bierbooms, and G. J. W. van Bussel. Generating nested quadrature rules with positive weights based on arbitrary sample sets. *SIAM/ASA Journal on Uncertainty Quantification*, 8(1):139–169, 2020.

[8] C. Botts, W. Hörmann, and J. Leydold. Transformed density rejection with inflection points. *Statistics and Computing*, 23(2):251–260, 2011. `doi:10.1007/s11222-011-9306-4`.

[9] L. Brandolini, C. Choirat, and L. Colzani. Quadrature rules and distribution of points on manifolds. *Annali della Scuola Normale Superiore - Classe di Scienze*, 2014. ISSN 2036-2145. `doi:10.2422/2036-2145.201103_007`.

[10] H. Brass and K. Petras. *Quadrature Theory*, volume 178 of *Mathematical Surveys and Monographs.* American Mathematical Society, 2011. `doi:10.1090/surv/178`.

[11] L. Brutman. Lebesgue functions for polynomial interpolation-a survey. *Annals of Numerical Mathematics*, 4: 111–128, 1996.

[12] M. D. Buhmann. Radial basis functions. *Acta Numerica*, pages 1–38, 2000. `doi:10.1007/978-3-540-69909-5_3`.

[13] T. Butler, L. Graham, S. Mattis, and S. Walsh. A measure-theoretic interpretation of sample based numerical integration with applications to inverse and prediction problems under uncertainty. *SIAM Journal on Scientific Computing*, 39(5):A2072–A2098, 2017. `doi:10.1137/16m1063289`.

[14] T. Butler, J. D. Jakeman, and T. Wildey. Convergence of probability densities using approximate models for forward and inverse problems in uncertainty quantification. *SIAM Journal on Scientific Computing*, 40(5): A3523–A3548, 2018. `doi:10.1137/18m1181675`.

[15] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *ANU*, 7:1, 1998. `doi:10.1017/s0962492900002804`.

[16] 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.ress.2010.09.013`.

[17] 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.

[18] G. Derflinger, W. Hörmann, and J. Leydold. Random variate generation by numerical inversion when only the density is known. *ACM Transactions on Modeling and Computer Simulation*, 20(4):1–25, 2010. `doi:10.1145/1842722.1842723`.

[19] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999. `doi:10.1137/s0036144599352836`.

[20] M. S. Ebeida, S. A. Mitchell, L. P. Swiler, V. J. Romero, and A. A. Rushdi. POF-darts: Geometric adaptive sampling for probability of failure. *Reliability Engineering & System Safety*, 155:64–77, 2016. `doi:10.1016/j.ress.2016.05.001`.

[21] 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`.

[22] 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`.

[23] W. N. Edeling, R. P. Dwight, and P. Cinnella. Simplex-stochastic collocation method with improved scalability. *Journal of Computational Physics*, 310:301–328, 2016. doi:10.1016/j.jcp.2015.12.034.

[24] R. Franke. Scattered data interpolation: tests of some methods. *Mathematics of Computation*, 38(157):181–200, 1982. doi:10.1090/s0025-5718-1982-0637296-4.

[25] F. N. Fritsch and R. E. Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, 1980. doi:10.1137/0717021.

[26] 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.

[27] A. Genz. Testing multidimensional integration routines. In *Proceedings of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*, pages 81–94. Elsevier North–Holland, 1984.

[28] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106): 221–230, 1969. doi:10.1090/s0025-5718-69-99647-1.

[29] L. A. Grzelak, J. A. S. Witteveen, M. Suárez-Taboada, and C. W. Oosterlee. The stochastic collocation Monte Carlo sampler: highly efficient sampling from 'expensive' distributions. *Quantitative Finance*, pages 1–18, 2018. doi:10.1080/14697688.2018.1459807.

[30] A. Guessab and G. Schmeisser. Construction of positive definite cubature formulae and approximation of functions via Voronoi tessellations. *Advances in Computational Mathematics*, 32(1):25–41, 2008. doi:10.1007/s10444-008-9080-9.

[31] 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.

[32] J. Hewitt and J. A. Hoeting. Approximate Bayesian inference via sparse grid quadrature evaluation for hierarchical models. *ArXiv 1904.07270*, 2019.

[33] 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.

[34] D. Jackson. *Theory of Approximation (Colloquium Publications)*. American Mathematical Society, 1982. ISBN 978-0821810118.

[35] J. D. Jakeman and A. Narayan. Generation and application of multivariate polynomial quadrature rules. *Computer Methods in Applied Mechanics and Engineering*, 338:134–161, 2018. doi:10.1016/j.cma.2018.04.009.

[36] 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.

[37] O. P. Le Maître and O. M. Knio. Spectral methods for uncertainty quantification. *Scientific Computation*, 2010. ISSN 1434-8322. doi:10.1007/978-90-481-3520-2.

[38] J. Li and Y. M. Marzouk. Adaptive construction of surrogates for the Bayesian solution of inverse problems. *SIAM Journal on Scientific Computing*, 36(3):A1163–A1186, 2014. ISSN 1095-7197. doi:10.1137/130938189.

[39] C. Luo, J. Sun, and Y. Wang. Integral estimation from point cloud in d-dimensional space. In *Proceedings of the 25th annual symposium on Computational geometry - SCG '09*. ACM Press, 2009. doi:10.1145/1542362.1542389.

[40] 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.

[41] 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.

[42] A. Narayan and J. D. Jakeman. Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM Journal on Scientific Computing*, 36(6):A2952–A2983, 2014. doi:10.1137/140966368.

[43] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial & Applied Mathematics (SIAM), 1992. doi:10.1137/1.9781611970081.

[44] E. Novak and K. Ritter. Simple cubature formulas with high polynomial exactness. *Constructive Approximation*, 15(4):499, 1999. ISSN 1432-0940. `doi:10.1007/s003659900119`.

[45] F. Palacios, T. D. Economon, 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^{nd} Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2014. `doi:10.2514/6.2014-0243`.

[46] B. Peherstorfer and Y. Marzouk. A transport-based multifidelity preconditioner for Markov chain Monte Carlo. *Advances in Computational Mathematics*, pages 1–28, 2019. `doi:10.1007/s10444-019-09711-y`.

[47] M. Pisaroni, F. Nobile, and P. Leyland. Continuation multi-level Monte-Carlo method for uncertainty quantification in turbulent compressible aerodynamics problems modeled by RANS. Technical Report 10.2017, École Polytechnique Fédérale de Lausanne, 2017.

[48] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation And Machine Learning)*. The MIT Press, 2006. ISBN 0-262-18253-X.

[49] A. A. Rushdi, L. P. Swiler, E. T. Phipps, M. D'Elia, and M. S. Ebeida. VPS: Voronoi piecewise surrogate models for high-dimensional data fitting. *International Journal for Uncertainty Quantification*, 7(1):1–21, 2017. `doi:10.1615/int.j.uncertaintyquantification.2016018697`.

[50] C. Schillings and C. Schwab. Sparse, adaptive Smolyak quadratures for Bayesian inverse problems. *Inverse Problems*, 29(6):065011, 2013. `doi:10.1088/0266-5611/29/6/065011`.

[51] C. Schillings, B. Sprungk, and P. Wacker. On the convergence of the Laplace approximation and noise-level-robustness of Laplace-based Monte Carlo methods for Bayesian inverse problems. *ArXiV 1901.03958*, 2019.

[52] C. Schwab and A. M. Stuart. Sparse deterministic approximation of Bayesian inverse problems. *Inverse Problems*, 28(4):045003, 2012. `doi:10.1088/0266-5611/28/4/045003`.

[53] J. W. Slater, J. C. Dudek, and K. E. Tatum. The NPARC alliance verification and validation archive. Technical Report 2000-209946, NASA, 2000.

[54] S. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics, Doklady*, 4:240–243, 1963.

[55] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30^{th} Aerospace Sciences Meeting & Exhibit*, number 92-0439, pages 6–9, 1992. `doi:10.2514/6.1992-439`.

[56] A. M. Stuart and A. L. Teckentrup. Posterior consistency for Gaussian process approximations of Bayesian posterior distributions. *Mathematics of Computation*, 87(310):721–753, 2017. `doi:10.1090/mcom/3244`.

[57] T. J. Sullivan. *Introduction to Uncertainty Quantification*. Springer International Publishing, 2015. `doi:10.1007/978-3-319-23395-6`.

[58] S. T. Tokdar and R. E. Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2009. `doi:10.1002/wics.56`.

[59] 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`.

[60] G. A. Watson. *Approximation Theory and Numerical Methods*. John Wiley & Sons Ltd, 1980. ISBN 0471277061.

[61] D. C. Wilcox. *Turbulence modeling for CFD*, volume 2. DCW industries La Cañada, CA, 1998.

[62] J. A. S. Witteveen, A. Doostan, R. Pecnik, and G. Iaccarino. Uncertainty quantification of the transonic flow around the RAE 2822 airfoil. *Center for Turbulence Research, Annual Briefs, Stanford University*, 2009.

[63] J. A. S. Witteveen and G. Iaccarino. Simplex elements stochastic collocation for uncertainty propagation in robust design optimization. In *48^{th} AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics (AIAA), 2010. `doi:10.2514/6.2010-1313`.

[64] J. A. S. Witteveen and G. Iaccarino. Simplex stochastic collocation with random sampling and extrapolation for nonhypercube probability spaces. *SIAM J. Sci. Comput.*, 34(2):A814–A838, 2012. `doi:10.1137/100817504`.

[65] J. A. S. Witteveen and G. Iaccarino. Simplex stochastic collocation with ENO-type stencil selection for robust uncertainty quantification. *Journal of Computational Physics*, 239:1–21, 2013. `doi:10.1016/j.jcp.2012.12.030`.

[66] D. Xiu. *Numerical Methods for Stochastic Computations.* Princeton University Press, 2010. ISBN 0691142122.

[67] L. Yan and Y.-X. Zhang. Convergence analysis of surrogate-based methods for Bayesian inverse problems. *Inverse Problems*, 33(12):125001, 2017. `doi:10.1088/1361-6420/aa9417`.