Computers and Fluids 181 (2019) 173–187

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Benchmark solutions

An efficient approach for quantifying parameter uncertainty in the SST turbulence model

Jincheng Zhang, Song Fu*

School of Aerospace Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 14 April 2018
Revised 5 December 2018
Accepted 14 January 2019
Available online 15 January 2019

Keywords:
Uncertainty quantification
Surrogate model
Gaussian process machine learning
Bayesian inference
High dimensional model representation
RANS Turbulence model

ABSTRACT

In this work, an efficient approach for quantifying the parameter uncertainty for expensive computer models is proposed, which combines the high dimensional model representation (HDMR) technique and the Gaussian process machine learning (GPML) method to construct the surrogate model, then the so-constructed surrogate is used in the Bayesian inference procedure to obtain the posterior distribution of the model parameters. The applications of the proposed approach to simple mathematical functions are investigated, demonstrating its efficiency and accuracy for both continuous function and function with discontinuities. The computer design of the sampling points based on GPML is also proposed and the results show that the proposed method is promising in terms of both efficiency and extensibility to high dimensional problems. After testing the proposed approach with simple mathematical functions, it is applied to the two equation SST turbulence model for hypersonic flow over flat plate with a wide range of Mach numbers. A Bayesian scenario-averaging method based on the flow quantities that can characterize both the flow scenario and the model's performance in the scenario is proposed and it is employed for the model predictions of new flow scenarios. The results show that the prediction mean values match well with the DNS data and the corresponding uncertainties are well captured.

1. Introduction

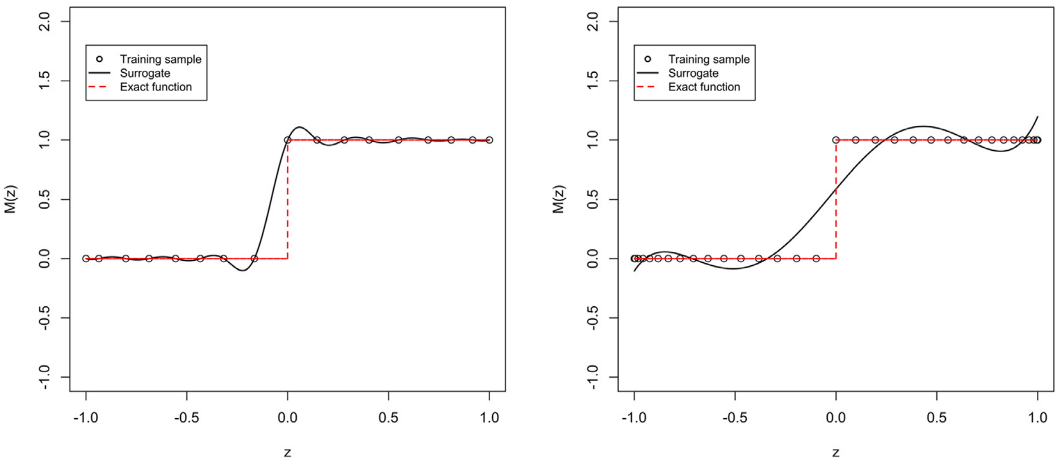
Computer models have been widely used for numerically simulating complex physical systems, such as in the areas of weather forecasting, climate simulation, and aircraft design. In order to make reliable predictions of these complex systems, not only the quantities of interest (QoI) but also the underlying uncertainties should be quantified rigorously. The empirical parameters of the computer models are a major source of uncertainties and a number of studies have focused on quantifying the uncertainties through the model parameters [1–6]. Often, due to the inherent complexity of the physical systems, the corresponding computer models can be quite complex, rendering the intrusive uncertainty quantification (UQ) approach intractable. And for practical problems in engineering applications, the computer models can also be very expensive to evaluate, making the direct Monte-Carlo method computationally prohibitive. Thus a non-intrusive, efficient UQ method is needed to capture the uncertainty of the QoIs while keeping the cost low.

An efficient Bayesian UQ approach is recently proposed in Ref [7], which combines the adaptive high dimensional model representation technique (HDMR) [8] and stochastic collocation (SC) method based generalized polynomial chaos (gPC) [9] to construct the surrogate for sampling procedure in calibration step of the Bayesian UQ framework [10]. Specifically, the adaptive HDMR technique is used to decompose the original high dimensional problems into several lower-dimensional subproblems, which are subsequently solved with the gPC-based SC method. This approach is applied to both a simple mathematical function and a complex fluid dynamic model, i.e. $k-\omega-\gamma$ transition model [11], demonstrating both its efficiency and accuracy. However, as the surrogate modeling for the subproblems is based on global polynomials, the so-constructed surrogate suffers from the problems of oscillations, especially when the response surface has local discontinuities. Moreover, even with the use of the Smolyak algorithm [12] for sparse grid design, the resulting design nodal set is by no means optimal. Thus a more efficient surrogate modeling method needs to be employed, which should also capture the response surface well even in the presence of discontinuities.

In order to resolve successfully discontinuities in the random space, Ma&Zabaras [13] proposed an adaptive hierarchical sparse grid collocation(ASGC) algorithm, in which they employed piecewise multi-linear hierarchical basis functions for surrogate modeling and hierarchical surplus is used as an error indicator to automatically detect the discontinuities. In Ref [14], Witteveen

* Corresponding author.
E-mail address: fs-dem@tsinghua.edu.cn (S. Fu).

https://doi.org/10.1016/j.compfluid.2019.01.017
0045-7930/© 2019 Elsevier Ltd. All rights reserved.



(a) The surrogate constructed by GPML

(b) The surrogate constructed by gPC-based SC (the polynomial degree is 5)

Fig. 1. The surrogates for the 1D step function, constructed by the GPML method and gPC-based SC method. Here the final size of the training set is 17 for GPML and 33 for gPC-based SC method.

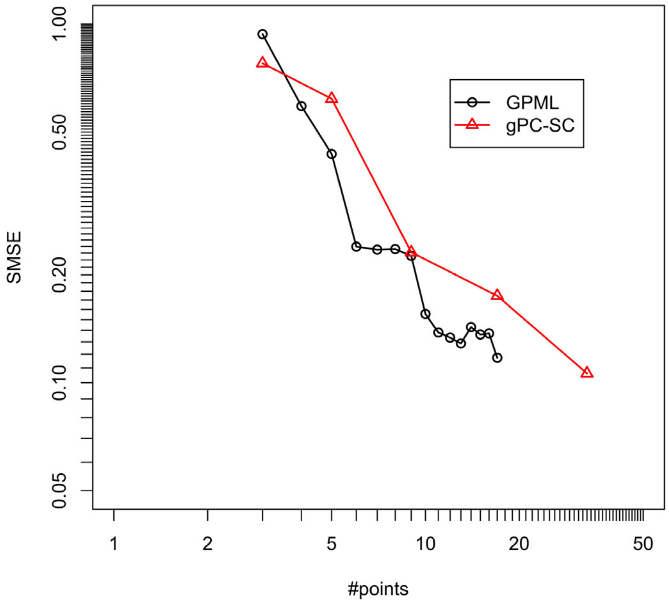


Fig. 2. The standard mean squared errors (SMSE) between the surrogate and the exact model (1D step function), against the size of the training set.

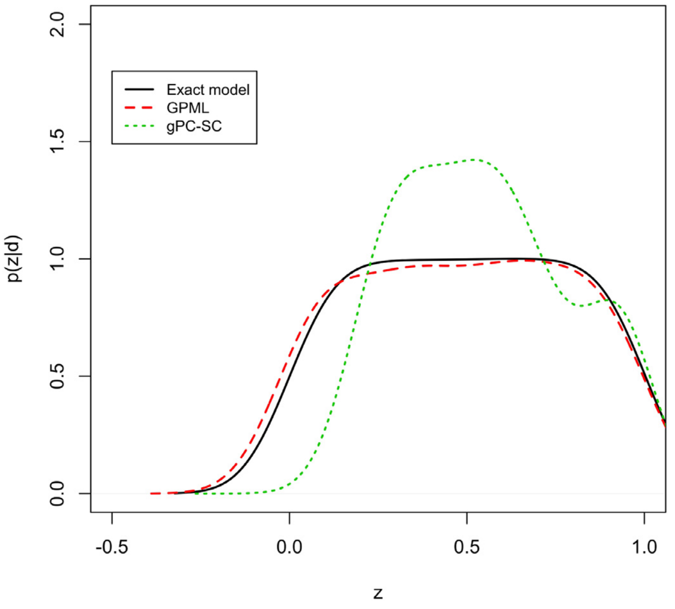
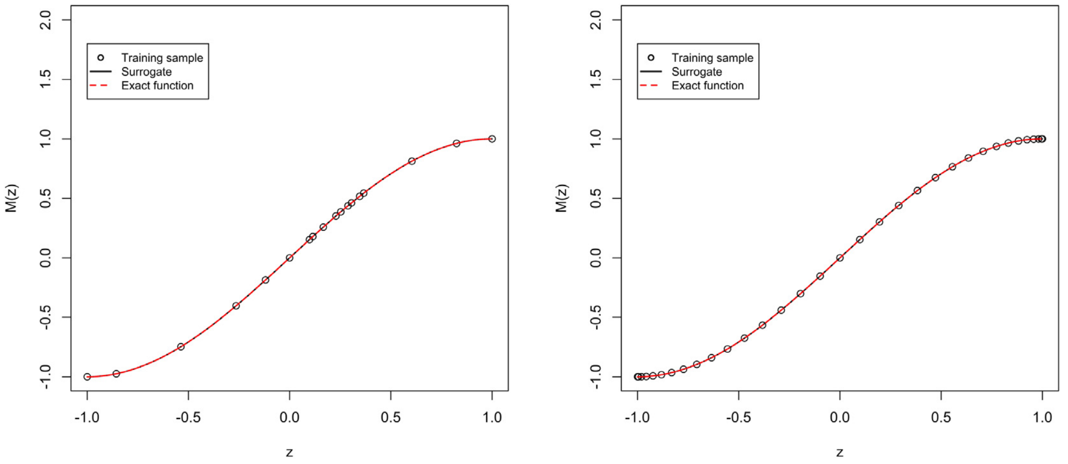


Fig. 3. The posterior distribution of the model parameter z of the 1D step function, using the exact model, the GPML surrogate and the gPC-based SC surrogate.

et al. proposed the Simplex-stochastic collocation(SSC) method, suppressing the possible unphysical oscillations by imposing the so-called Local-Extremum Conserving Limiter to each simplice. Edeling et al. [15] improved the original SSC method by proposing an alternative interpolation stencil technique based on the Set-Covering problem. In Ref [16], it is demonstrated that Gaussian process machine learning (GPML) approach can be used to successfully construct the surrogate for the response surface with discontinuities. In the literature, Gaussian process has already been used as a powerful tool for surrogate construction in the context of UQ, including [10,17–19] among many others. Moreover, the GPML approach provides a consistent way for the efficient design of the sample points. The main idea is that, with GPML method we can obtain not only the posterior mean but also the posterior variance of the output as a function of the input random variables, and the posterior variance can be used as an indicator of the error between the constructed GP model and the true response surface, which provides a consistent way for the design of the new sampling point.

The aim of this paper is to efficiently construct the surrogate, irrespective of whether the response surface is smooth or discontinuous. Then the so-constructed surrogate is used for sampling procedure in Bayesian UQ. To do so, we combine the adaptive HDMR technique [8] with GPML method [16], to construct the surrogate model. Then this surrogate model is used for both the forward propagation and the inverse problem. The proposed approach can be seen as an improved version of the UQ approach proposed in Ref [7], by replacing the gPC-based SC method with GPML method.

The application of the proposed approach to hypersonic turbulent flow simulations is of primary interest to the present authors. As is pointed out in Ref [20], turbulence modeling remains a major source of uncertainty in the computational prediction of aerodynamic forces and heating for hypersonic turbulent flows, thus the uncertainty originating from turbulence models for this



(a) The surrogate constructed by GPML

(b) The surrogate constructed by gPC-based SC (the polynomial degree is 5)

Fig. 4. The surrogates for the 1D smooth function, constructed by the GPML method and gPC-based SC method. Here the final size of the training set is 18 for GPML and 33 for gPC-based SC method.

Table 1
The prior range for model parameters and the hyper-parameter σ_h.

Coefficient	Nominal value	Left boundary	Right boundary
a_1	0.31	0.2604	0.434
κ	0.41	0.246	0.574
β^*	0.09	0.054	0.126
σ_{k_1}	0.85	0.51	1.19
σ_{ω_1}	0.5	0.3	0.7
β_1	0.075	0.045	0.105
σ_{k_2}	1.0	0.6	1.4
σ_{ω_2}	0.856	0.5136	1.1984
β_2	0.0828	0.04968	0.11592
σ_h	*	0	0.5

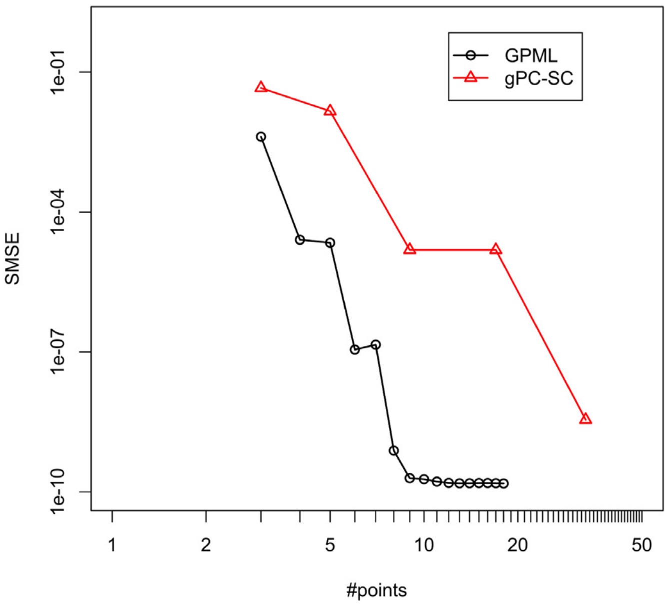
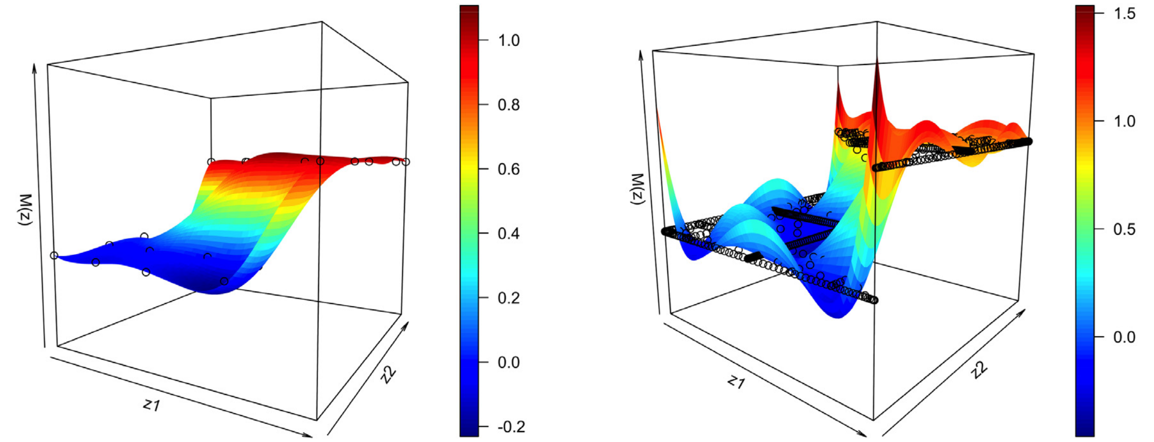


Fig. 5. The standard mean squared errors (SMSE) between the surrogate and the exact model (1D smooth function), against the size of the training set.

flow configuration should be quantified in order to make reliable predictions. Moreover, with the increase of the Mach number, the compressibility effects may further undermine the turbulence model's ability for accurate predictions of QoIs. Although there are several compressibility corrections existing in the literature, most widely-used turbulence models do not include them [21]. In this paper, the two-equation SST model of Menter [22] is used for the computational predictions of the surface drag for hypersonic turbulent flows, and the uncertainty is accounted for through the model parameters. The model inadequacy is also a major source of uncertainty in turbulence modeling, and it has raised the research interests in the field recently, e.g. [23–27]. In this work we model the inadequacy term simply by a multiplicative Gaussian and focus on the parameter uncertainties.

In this work, the flows over a wide range of Mach numbers are investigated, with the use of the DNS database recently reported in Ref [28]. When making predictions for a new flow scenario, a Bayesian scenario-averaging procedure is necessary in order to take account of all the information of the calibration results. Smart scenario-averaging methods have been proposed in the literature, e.g. Ref [29]. In this work, a physics-based method to determine the scenario weights is proposed, which is based on the mean squared value of the difference of certain quantities between scenarios. The choice of the quantities to characterize each scenario depends on the problems at hand and the computer models that are used in the simulations. Specifically, the quantities that can characterize the flow field and also the model's performance in different scenarios should be chosen, as these quantities can reveal the variability of the posterior distributions of model parameters across scenarios.

The paper is organized as follows: the previously proposed Bayesian UQ approach is described in Section 2. In Section 3 the surrogate model construction approach based on GPML is described. In Section 4 we demonstrate the accuracy and efficiency of the proposed method through simple mathematical functions, including both smooth functions and functions with discontinuities. A comparison between the proposed approach and the baseline approach is given. After testing our approach with these simple mathematical functions, we apply the approach to two equation SST turbulence model in hypersonic turbulence simulations over a wide range of Mach numbers in Section 5. In Section 6 the model predictions with quantified uncertainties are carried out and the Bayesian scenario-averaging method based on the flow quantities that can characterize both the flow field and the model's performance in the scenario (e.g. turbulence Mach number in our



(a) The surrogate constructed by GPML

(b) The surrogate constructed by gPC-based SC

Fig. 6. The surrogate for the 2D step function, constructed by the GPML method and gPC-based SC method. Here the final size of the training set is 23 for GPML and 705 for gPC-based SC method. The training samples are also shown.

Table 2
Flow conditions and the DNS data from Ref. [28].

Case	Mach number	Reynolds number (m^{-1})	Boundary layer thickness $\delta 99(mm)$	Skin friction coefficient
1	3	5.55×10^6	8.85	2.17×10^{-3}
2	4	7.4×10^6	12.0	1.61×10^{-3}
3	5	9.25×10^6	15.1	1.31×10^{-3}
4	6	1.11×10^7	19.7	1.08×10^{-3}
5	7	1.30×10^7	28.1	8.51×10^{-4}
6	8	1.48×10^7	31.8	7.80×10^{-4}

application) is employed for the prediction of scenarios with no experimental data. Finally the conclusion is drawn in Section 7.

2. Baseline uncertainty quantification approach

In this section, the previously proposed Bayesian UQ approach is briefly described. The interested reader may refer to [7] for further details.

2.1. Bayesian uncertainty quantification framework

In the Bayesian framework, various forms of uncertainty, whether aleatoric or epistemic, are all represented through probability, which are usually characterized by their probability density functions (PDF). During the Bayesian calibration, the posterior distributions of the parameters are obtained through Bayes' rule:

$$p(\mathbf{z}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{z})p(\mathbf{z}) \tag{1}$$

where random vector **z** represents the model parameters and **d** the experimental observation. Here p(**z**) is the prior distribution of the model parameters and p(**d**|**z**) is the likelihood, the calculation of which requires the computer model output and the experimental observation, combined by the constructed stochastic model. After calibration, the model prediction is done by propagating the PDF of the input parameters through the simulation code to obtain the posterior PDF of the QoIs, which is expressed as:

$$p(\tilde{\mathbf{q}}|\mathbf{d}) = \int p(\tilde{\mathbf{q}}, \mathbf{z}|\mathbf{d})d\mathbf{z} = \int p(\tilde{\mathbf{q}}|\mathbf{d}, \mathbf{z})p(\mathbf{z}|\mathbf{d})d\mathbf{z} = \int p(\tilde{\mathbf{q}}|\mathbf{z})p(\mathbf{z}|\mathbf{d})d\mathbf{z} \tag{2}$$

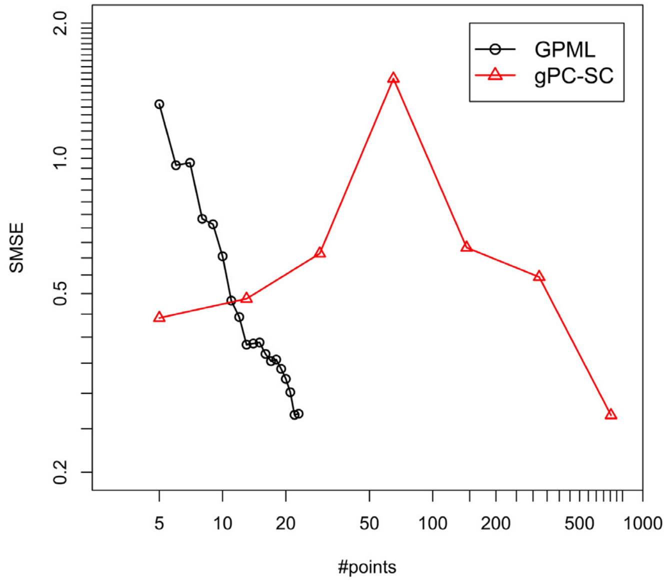


Fig. 7. The standard mean squared errors (SMSE) between the surrogate and the exact model (2D step function), against the size of the training set.

where $\tilde{\mathbf{q}}$ represents the QoIs. In this work the stochastic model is constructed simply by accounting for the model inadequacy through a multiplicative Gaussian random variable :

$$\tilde{\mathbf{q}} = (1 + \boldsymbol{\eta})\mathscr{M}(\mathbf{x}, \mathbf{z}) \tag{3}$$

where $\boldsymbol{\eta}$ is a random vector with each component η_i as zero mean, independent and identically distributed Gaussian: i.e. $\eta_i \sim \mathscr{N}(0, \sigma^2)$. $\mathscr{M}(\mathbf{x}, \mathbf{z})$ is the output of QoIs from our simulation code, depending on the explanatory variable **x**(e.g. Mach number,

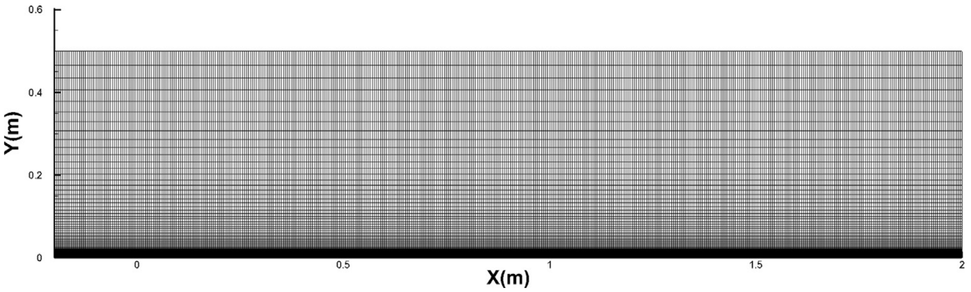
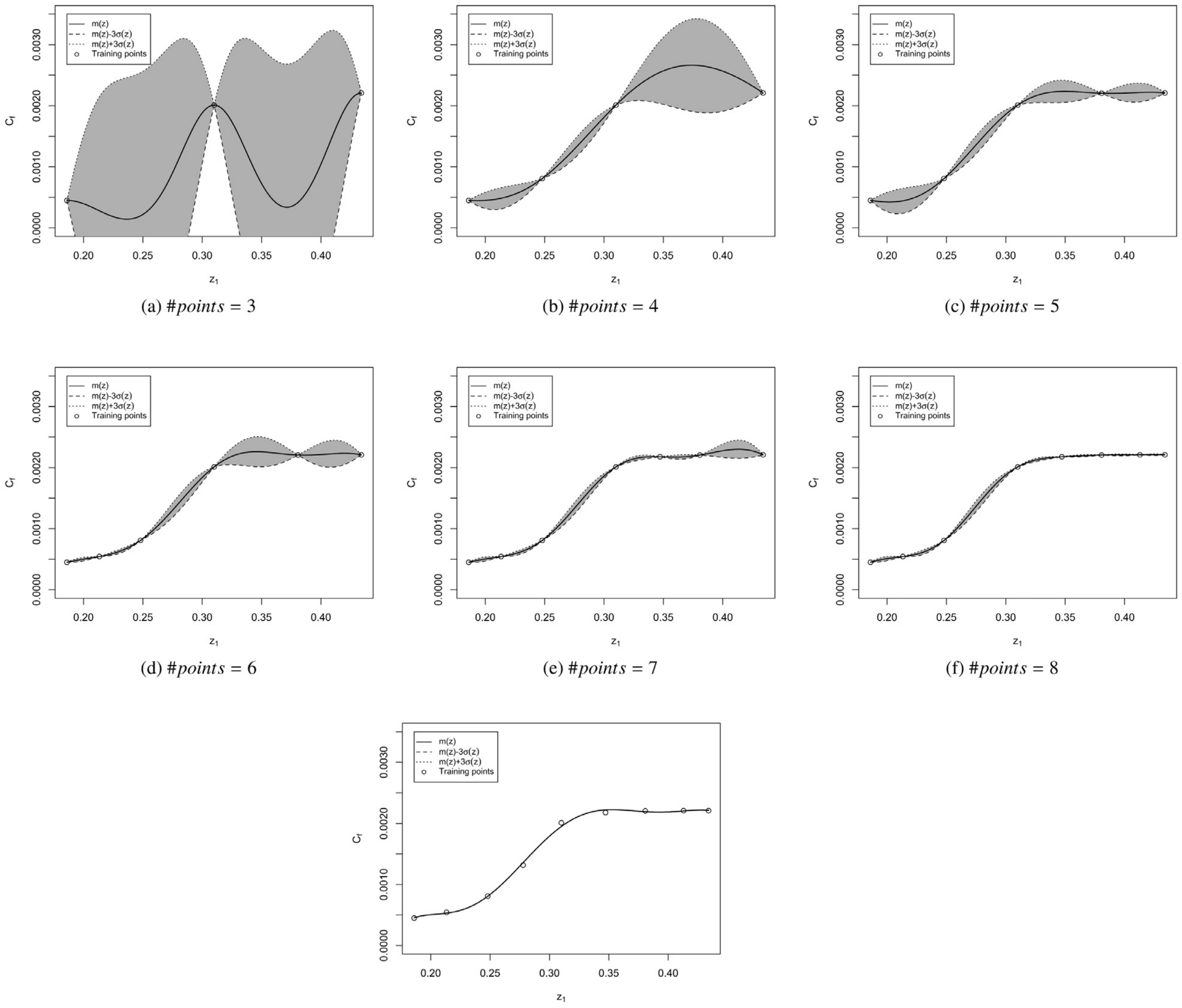


Fig. 8. The computational domain and the final mesh for Case 1 in Table 2.



(a) #$points$ = 3

(b) #$points$ = 4

(c) #$points$ = 5

(d) #$points$ = 6

(e) #$points$ = 7

(f) #$points$ = 8

(g) #$points$ = 9

Fig. 9. The design process based on GPML for the surrogate construction of $\mathscr{M}(\mathbf{z}_1)|_{\mathbf{z}=\bar{\mathbf{z}}\backslash\mathbf{z}_1}$. The training points, $m(z)$ and $m(z)\pm3\sigma(z)$ are shown, where $m(z)$ is the mean function of the posterior GP and $\sigma(z)$ the variance.

Reynolds number etc.) and the model parameter $\mathbf{z}$. $\tilde{\mathbf{q}}$ can be related to the experimental observation $\mathbf{d}$ as:

$$\mathbf{d} = \tilde{\mathbf{q}} + \mathbf{e} \tag{4}$$

Here $\mathbf{e}$ represents the measurement error, which is modeled as a zero mean, independent and identically distributed Gaussian, i.e. $e_i \sim \mathcal{N}(0, \sigma_e^2)$. σ_e is determined from the corresponding experiments. Thus from Eqs. (3) and (4) we can relate the model output to the experimental observation, and we can obtain:

$$\mathbf{d}|\sigma, \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\lambda}) \tag{5}$$

where

$$\boldsymbol{\mu} = \mathscr{M}(\mathbf{x}, \mathbf{z}) \text{ and } \boldsymbol{\lambda} = \mathscr{M}^T(\mathbf{x}, \mathbf{z})\sigma^2\mathscr{M}(\mathbf{x}, \mathbf{z}) + \sigma_e^2\boldsymbol{I} \tag{6}$$

With the prior distribution specified by the modeler and the stochastic model constructed as Eq. (5), we can recast Eq. (1) as:

$$p(\boldsymbol{\theta}_M|\mathbf{d}) \propto \frac{1}{\sqrt{(2\pi)^{N_d}|\boldsymbol{\lambda}|}} exp\left(-\frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{\lambda}^{-1}\boldsymbol{\delta}\right)p(\boldsymbol{\theta}_M) \tag{7}$$

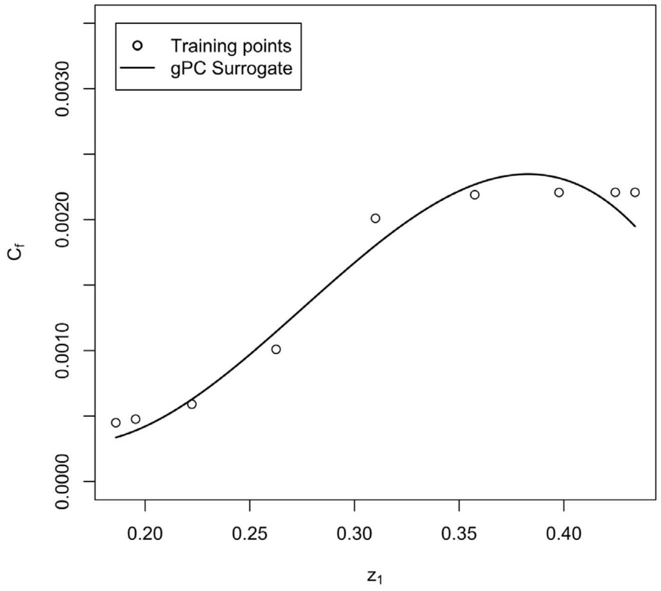


Fig. 10. The surrogate constructed by gPC-based SC method with sparse grid level $l = 4$.

where $\boldsymbol{\theta}_M$ denotes $\{\sigma, \mathbf{z}\}$, N_d is the dimension of the experimental observation, $|\boldsymbol{\lambda}|$ represents the determinant of $\boldsymbol{\lambda}$, and $\boldsymbol{\delta} = \mathbf{d} - \mathscr{M}(\mathbf{x}, \mathbf{z})$. Then a sampler is employed to obtain samples according to Eq. (7) and the kernel estimation is used to evaluate the posterior distributions of the model parameters. In this work an adaptive Metropolis-Hastings MCMC sampler [30], as implemented in the R [31] package MHadaptive [32], is employed. The sampling procedure involves a large number of evaluations of $\mathscr{M}(\mathbf{x}, \mathbf{z})$ thus the construction of a surrogate model for $\mathscr{M}(\mathbf{x}, \mathbf{z})$ is usually needed.

2.2. Surrogate modeling

To construct the surrogate for $\mathscr{M}(\mathbf{z})$(we omit the explanatory variable x hereafter for brevity), the adaptive HDMR is employed to decompose the original moderately high dimensional problems into several subproblems with lower dimensions, which are then solved with gPC-based SC method. The CUT-HDMR of $\mathscr{M}(\mathbf{z})$ is defined as [8,33]

$$\mathscr{M}(\mathbf{z}) = \sum_{\mathbf{u}\subset\mathscr{D}} f_{\mathbf{u}}(z_{\mathbf{u}}) = \sum_{\mathbf{u}\subset\mathscr{D}} \sum_{\mathbf{v}\subset\mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \mathscr{M}(z_{\mathbf{v}})|_{\mathbf{z}=\bar{\mathbf{z}}\setminus z_{\mathbf{v}}} \tag{8}$$

where $\mathscr{D} = \{1, 2, \ldots, N_z\}$, $f_{\mathbf{u}}(z_{\mathbf{u}}) = f_{i_1\ldots i_s}(z_{i_1}, \ldots, z_{i_s})$ for $\mathbf{u} = \{i_1, i_2, \ldots, i_s\}$ and by convention $f_\emptyset(z_\emptyset) = f_0$, N_z represents the input dimension and $\{\bar{z}_i\}$ is the reference point. Specifically, the zeroth-order component function $f_0 = \mathscr{M}(\bar{\mathbf{z}})$, the 1st-order term $f_i(z_i) = \mathscr{M}(\bar{z}_1, \ldots, z_i, \ldots, \bar{z}_{N_z}) - f_0$ and the 2nd-order term $f_{i,j}(z_i, z_j) = \mathscr{M}(\bar{z}_1, \ldots, z_i, \ldots, z_j, \ldots, \bar{z}_{N_z}) - f_i(z_i) - f_j(z_j) - f_0$. Respectively they represent the mean, the 1st-order correlation and the 2nd-order correlation effects of the input parameters on the model output. In this work, the implementation of HDMR is the same as in our previous work [7] and the interested reader may refer to [7] for all the details.

After decomposing the original high dimensional problems into subproblems, the gPC-based SC method is employed to construct the surrogate for the subproblems. In this work the GPC [34] R package is employed for the code implementation. As the so-constructed surrogate suffers from the problems of oscillations, especially when the response surface has local discontinuities, we propose an alternative approach based on GPML to solve the subproblems.

3. Gaussian process machine learning

In this section we describe the GPML approach used for constructing the surrogate. The interested reader may refer to [10,16,17] for further details.

In Gaussian process regression, $\mathscr{M}(\mathbf{z})$ is assumed as a random function with the prior distributions:

$$\mathscr{M}(\mathbf{z}) \sim GP(m(\mathbf{z}), k(\mathbf{z}, \mathbf{z}')) \tag{9}$$

Here $m(\mathbf{z})$ is the mean function, $k(\mathbf{z}, \mathbf{z}')$ is the covariance function of the Gaussian process. Usually the mean function is taken to be zero for notational simplicity. Given the experimental design $\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots \mathbf{z}^{(N_z)}\}$ and the corresponding model evaluations $\{\mathscr{M}(\mathbf{z}^{(1)}), \mathscr{M}(\mathbf{z}^{(2)}), \ldots \mathscr{M}(\mathbf{z}^{(N_z)})\}$, the joint distribution of the training outputs $\mathscr{M}(\mathbf{z}^{(Tr)})$ and the test outputs $\mathscr{M}(\mathbf{z}^{(*)})$ according to Eq. (9) is

$$\begin{bmatrix} \mathscr{M}(\mathbf{z}^{(Tr)}) \\ \mathscr{M}(\mathbf{z}^{(*)}) \end{bmatrix} \sim GP\left(\mathbf{0}, \begin{bmatrix} K(Z^{Tr}, Z^{Tr}) & K(Z^{Tr}, Z^*) \\ K(Z^*, Z^{Tr}) & K(Z^*, Z^*) \end{bmatrix}\right) \tag{10}$$

where $K(Z^{Tr}, Z^*)$ denotes the $N_z \times N_*$ matrix of the covariance function $k(\mathbf{z}, \mathbf{z}')$ evaluated at all pairs of training and test points, and similarly for the definitions of $K(Z^{Tr}, Z^{Tr})$, $K(Z^*, Z^{Tr})$ and $K(Z^*, Z^*)$. After conditioning this joint Gaussian prior distribution on the observations we can obtain the posterior distribution of $\mathscr{M}(\mathbf{z})$:

$$\mathscr{M}(\mathbf{z}) \sim GP(K(Z^*, Z^{Tr})K(Z^{Tr}, Z^{Tr})^{-1}\mathscr{M}(\mathbf{z}^{(Tr)}),$$

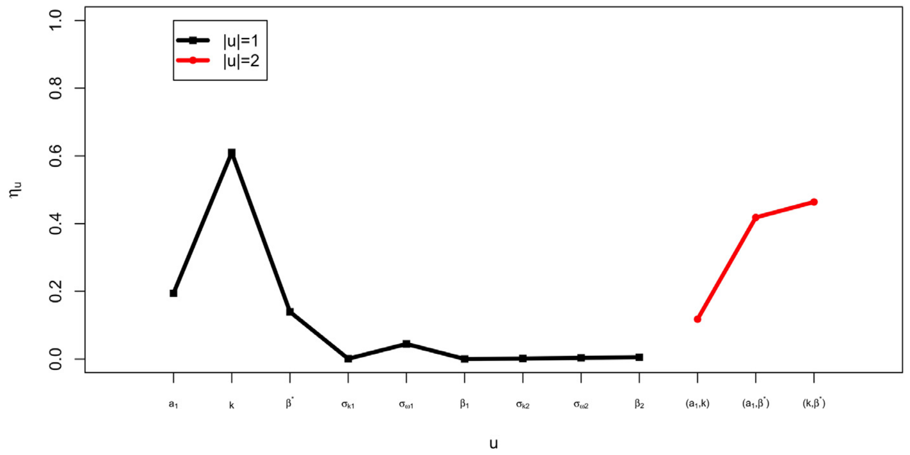


Fig. 11. The sensitivity analysis for the selection of HDMR terms. The details of weight definitions can be found in [7].

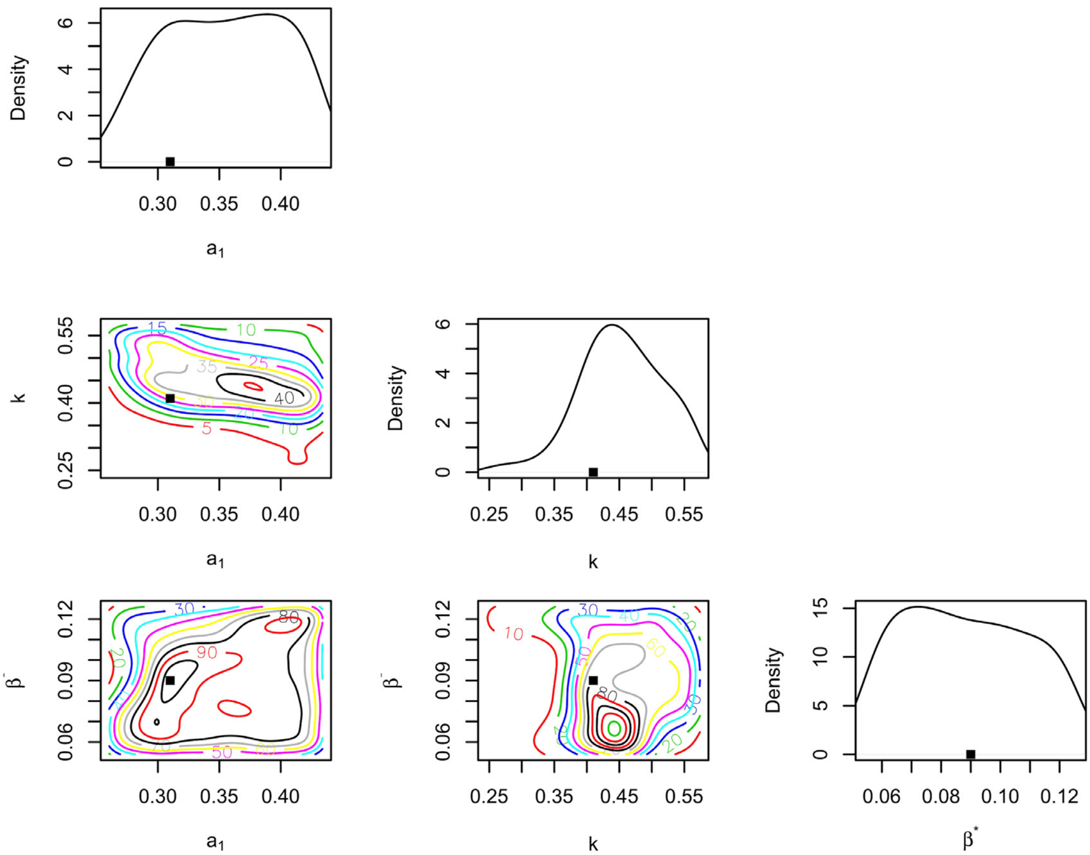


Fig. 12. The posterior distributions of the model parameters (a_1, κ, β^*), with surface drag as the calibration dataset.

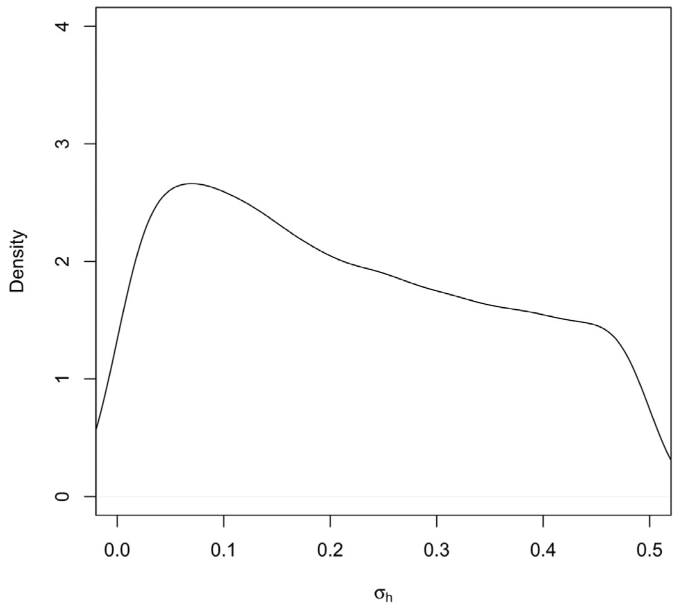


Fig. 13. The posterior distributions of the hyperparameter σ_h, with surface drag as the calibration dataset.

$$K(Z^*, Z^*) - K(Z^*, Z^{Tr})K(Z^{Tr}, Z^{Tr})^{-1}K(Z^{Tr}, Z^*)) \qquad (11)$$

Then for constructing the surrogate, the mean function is used as the surrogate of the response surface and the variance function is used as the error indicator:

$$\mathscr{M}_S(\mathbf{z}^*) = K(\mathbf{z}^*, Z^{Tr})K(Z^{Tr}, Z^{Tr})^{-1}\mathscr{M}(\mathbf{z}^{(Tr)}) \qquad (12)$$

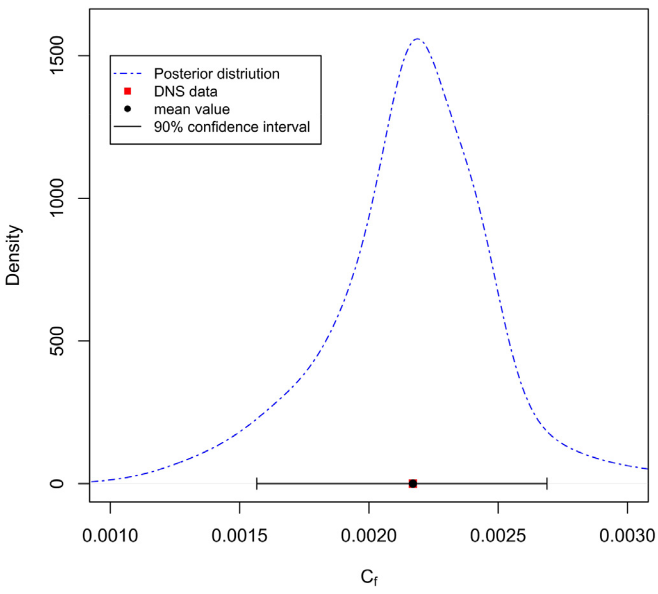


Fig. 14. The posterior distribution of the skin friction coefficient C_f, with surface drag as the calibration dataset.

$$Var_{\mathscr{M}_S}(\mathbf{z}^*) = K(\mathbf{z}^*, \mathbf{z}^*) - K(\mathbf{z}^*, Z^{Tr})K(Z^{Tr}, Z^{Tr})^{-1}K(Z^{Tr}, \mathbf{z}^*) \qquad (13)$$

The choice of the covariance function $k(\mathbf{z}, \mathbf{z}')$ is a key issue in GPML and various covariance functions can be used in practice, depending on the problem at hand and the available training data. Here in this work the sum of a squared exponential covariance

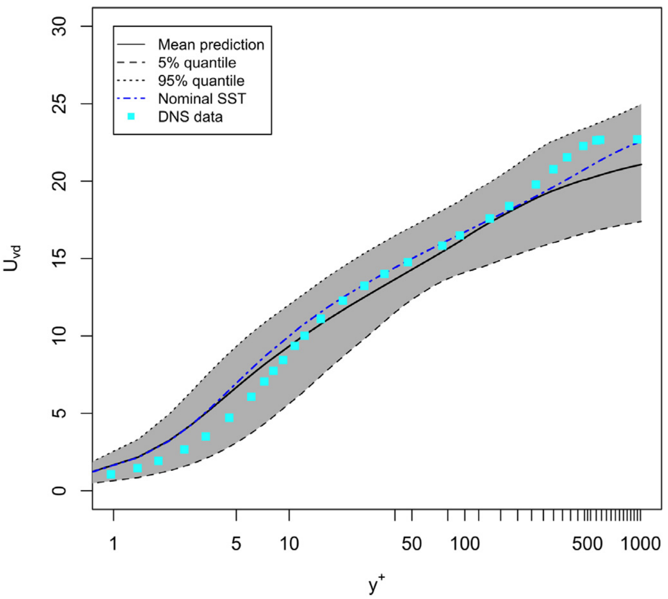


Fig. 15. The posterior distribution of the Van Driest transformed velocity U_{vd}, with surface drag as the calibration dataset.

function and an independent white noise covariance function is employed:

$$k(\mathbf{z},\mathbf{z}') = k^{SE}(\mathbf{z},\mathbf{z}') + k^{white}(\mathbf{z},\mathbf{z}') = \sigma_1^2 exp\left(-\frac{(\mathbf{z}-\mathbf{z}')^2}{2l^2}\right) + \sigma_2^2\delta_{\mathbf{zz}'} \tag{14}$$

More complicated, non-stationary covariance functions can be employed when the problem at hand is more complex(e.g. high dimensional problems) and more training data can be obtained. The determination of the hyperparameters $(\sigma_1^2, 1/l^2, \sigma_2^2)$(denoted as $\boldsymbol{\theta}$ hereafter) is done by maximizing the log marginal likelihood:

$$logp(\mathscr{M}|\mathbf{z},\boldsymbol{\theta}) = -\frac{1}{2}\mathscr{M}(\mathbf{z}^{(Tr)})^T K(Z^{Tr},Z^{Tr})^{-1}\mathscr{M}(\mathbf{z}^{(Tr)}) - \frac{1}{2}log|K(Z^{Tr},Z^{Tr})| - \frac{N_z}{2}log(2\pi) \tag{15}$$

In this work, the gptk R package [35] is employed for the GPML, in which an optimizer based on conjugate gradients is employed to determine the hyperparameters.

3.1. Design of the sampling points

The variance function $Var_{\mathscr{M}_S}(\mathbf{z}^*)$ obtained by GPML method can be used as the error indicator of the constructed surrogate, which provides a consistent way for the design of the sampling points in computer experiments. Specifically, a new sampling point should be added where the corresponding variance is the greatest. In practice, the newly-added sampling point may be too close to the already-selected points, making the matrix $K(Z^{Tr}, Z^{Tr})$ ill-conditioned, thus a random noise is added to the design position of the new sampling points to prevent this situation. Specifically, we iterate the design process of the new point by adding a random noise until the new point is not too close to the previous ones. In addition, for some cases during the sequential design, the optimizer may obtain extreme values for the hyperparameters, making the constructed surrogate highly irregular. In this case, the optimized hyperparameters from the previous design are employed. Also, the design based on the variance function become unreasonable when the value of the variance function is too small, i.e. 10^{-6} of the mean function magnitude. However, this poses no real problem for the current investigations, since it is sufficient to stop the design procedure when the variance function decreases to such a small value. The overall algorithm for the design of sampling points is summarized below as Algorithm 1.

Algorithm 1 the surrogate construction method based on GPML.

```
Initialize the set of training points $\mathscr{T}_r$ by including only the points at the vertices and center of the input random space $\mathscr{R}$. Evaluate corresponding model output $\mathscr{M}(\mathbf{z})$ for $\mathbf{z} \in \mathscr{T}_r$ and store them in the set $\mathscr{M}_{Tr}$.
Set the set of testing points $\mathscr{T}_e$ by including the regular tensor grid points or Monte-Carlo sampling points of the input space $\mathscr{R}$.
Set the initial hyperparameters $\theta_{init}$.
Set the allowed space $\Theta$ for optimizing the hyperparameters.
Set the maximum number of model evaluations $N_z$.
$i \leftarrow dim(\mathscr{T}_r)$.
while $i < N_z + 1$ do
    % Set the allowed minimum distance between the training points.
    $Dist = \epsilon_1/i$;
    % Set the standard deviation of the noise to be added to the new point.
    $sd_N = n_1 * Dist$.
    Based on the training set $\{\mathscr{T}_r, \mathscr{M}_{Tr}\}$, compute $\mathscr{M}_S(\mathbf{z})$ and $Var_{\mathscr{M}_S}(\mathbf{z})$ for $\mathbf{z} \in \mathscr{T}_e$ with the GPML method with the hyper-parameter $\theta_{opt}$, which is determined by maximizing the log marginal likelihood with an initial value $\theta_{init}$.
    if $\theta_{opt} \notin \Theta$ then
        Compute the $\mathscr{M}_S(\mathbf{z})$ and $Var_{\mathscr{M}_S}(\mathbf{z})$ for $\mathbf{z} \in \mathscr{T}_e$ with the hyperparameter $\theta_{init}$
        $\theta_{opt} \leftarrow \theta_{init}$
    end if
    $\theta_{init} \leftarrow \theta_{opt}$.
    % Set the new sampling point $\mathbf{z}_{new}$ as the point where the variance $Var_{\mathscr{M}_S}(\mathbf{z})$ reach maximum.
    $\mathbf{z}_{new} \leftarrow argmax(Var_{\mathscr{M}_S}(\mathbf{z})) + max\{min[\mathscr{N}(0, sd_N), 3sd_N], -3sd_N\}$
    if $max(Var_{\mathscr{M}_S}(\mathbf{z})) < \epsilon_2$ then
        $i \leftarrow N_z + 1$
    end if
    if $|\mathbf{z} - \mathbf{z}_{new}| > Dist$ for $\forall \mathbf{z} \in \mathscr{T}_r$ then
        % Update the training set $\{\mathscr{T}_r, \mathscr{M}_{Tr}\}$
        $\mathscr{T}_r \leftarrow \mathscr{T}_r \cup \{\mathbf{z}_{new}\}$
        $\mathscr{M}_{Tr} \leftarrow \mathscr{M}_{Tr} \cup \{\mathscr{M}(\mathbf{z}_{new})\}$
        $i \leftarrow i + 1$
    end if
end while
% The training set $\{\mathscr{T}_r, \mathscr{M}_{Tr}\}$, the constructed surrogate $\mathscr{M}_S(\mathbf{z})$ and the corresponding error assessment $Var_{\mathscr{M}_S}(\mathbf{z})$ are finally obtained.
```

Remark:In this work, $\theta_{init} = (1, 1, 10^{-8})$, $\Theta = (10^{-2}, 10^2) \times (0, 10) \times (10^{-8}, 10^{-1})$, $\epsilon_1 = 10^{-1}$, $n_1 = 1$, $\epsilon_2 = 10^{-6}$. The design process is not sensible to θ_{init}, ϵ_1 and n_1 as long as a reasonable initial value is specified to make the process start. Θ is determined by trial and error and we try to keep it as large as possible as long as the optimization process does not result in extreme values and fast oscillatory response surfaces. We have used the full tensor uniform grid as the testing set in the following applications (6000 points are used for the 1D cases and 80*80 points are used for the 2D case).

4. Application to simple mathematical functions

Here we apply the proposed UQ approach to simple mathematical functions, including both smooth functions and functions with

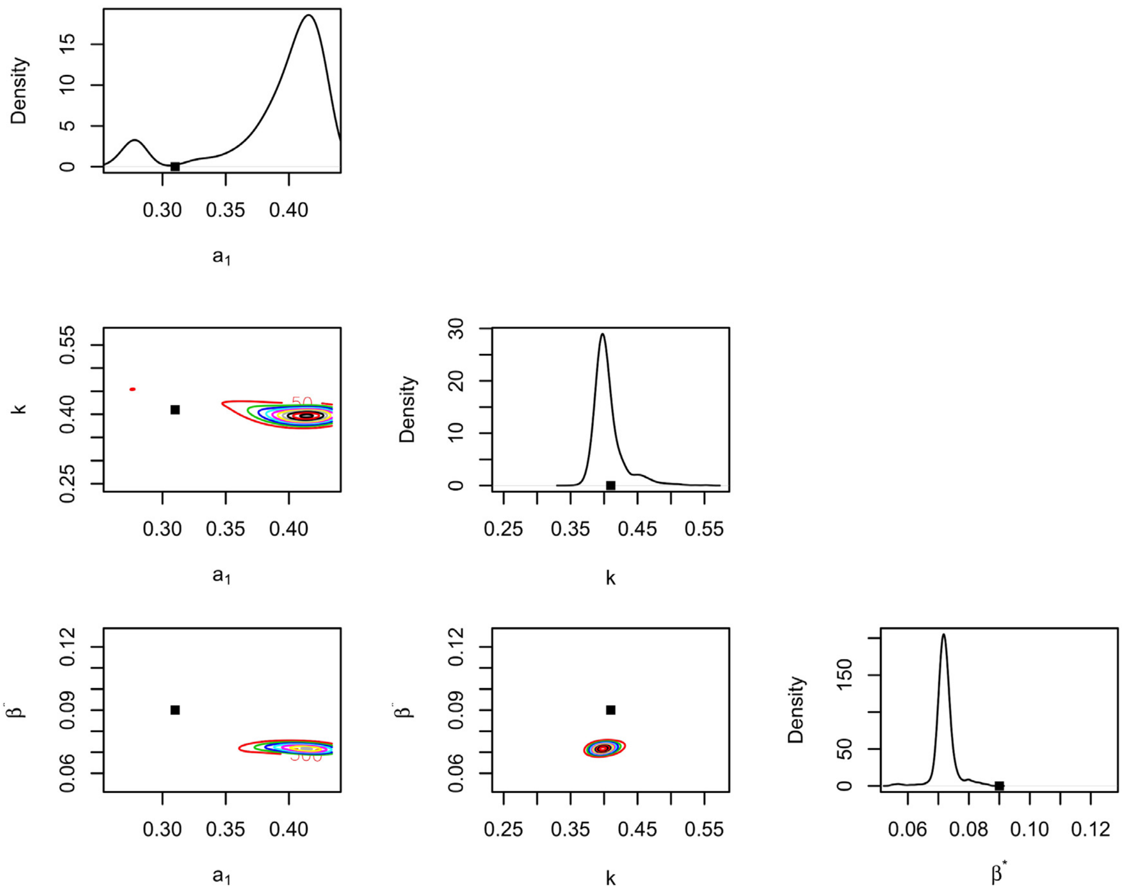


Fig. 16. The posterior distributions of the model parameters (a_1, κ, β^*), with velocity profile as the calibration dataset.

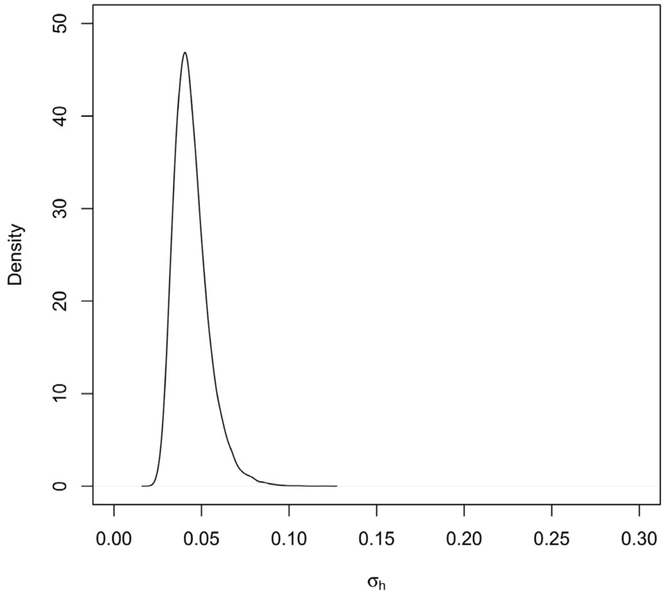


Fig. 17. The posterior distributions of the hyperparameter σ_h, with velocity profile as the calibration dataset.

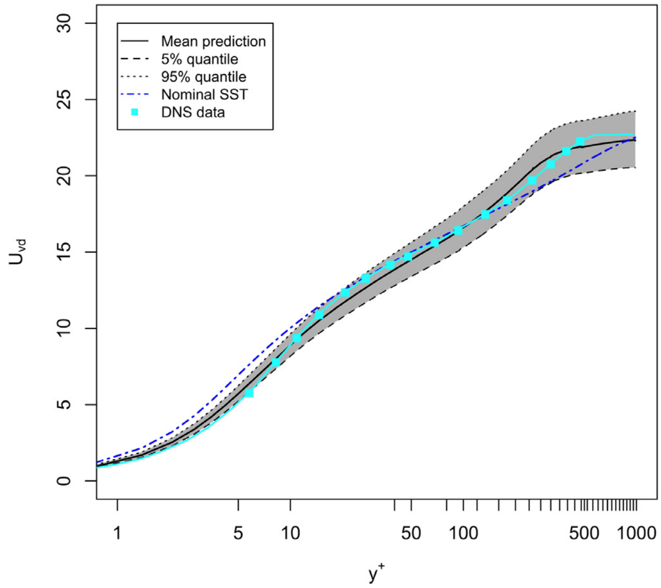


Fig. 18. The posterior distribution of the Van Driest transformed velocity U_{vd}, with velocity profile as the calibration dataset.

discontinuities, to demonstrate its accuracy and efficiency. A comparison with the baseline UQ approach is also shown. The details of the gPC implementation can be found in [7].

4.1. 1D Step function

The proposed approach is first applied to a function with discontinuities, i.e. the 1D step function defined on $[-1, 1]$:

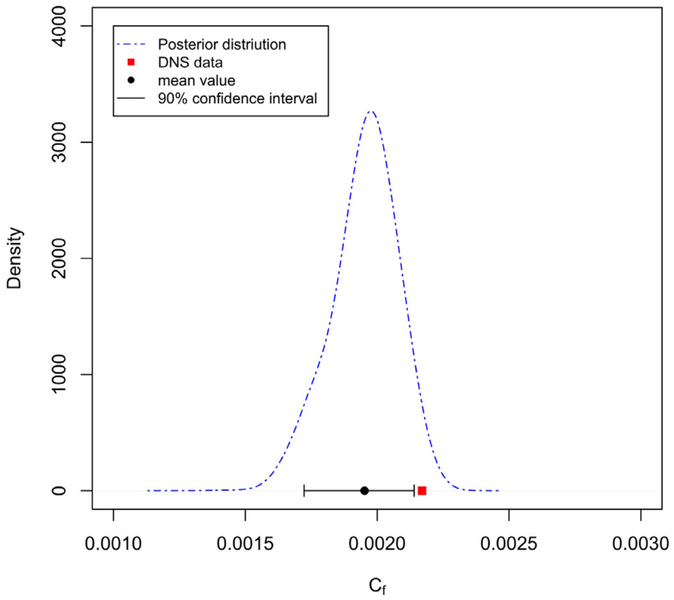


Fig. 19. The posterior distribution of the skin friction coefficient C_f, with velocity profile as the calibration dataset.

$$\mathscr{M}(\mathbf{z}) = \begin{cases} 0, & \mathbf{z} \in [-1, 0) \\ 1, & \mathbf{z} \in [0, 1] \end{cases} \tag{16}$$

The prior distribution of the model parameter is assumed as: $\mathbf{z} \sim \mathscr{U}(-1, 1)$. The model inadequacy term is neglected and the posterior is expressed as:

$$p(\mathbf{z}|\mathbf{d}) \propto \frac{1}{\sqrt{(2\pi)^{N_d}|\boldsymbol{\lambda}|}} exp\left(-\frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{\lambda}^{-1}\boldsymbol{\delta}\right)p(\mathbf{z}) \tag{17}$$

where the dimension of the experimental observation $N_d = 1$, $\boldsymbol{\lambda} = \sigma_e^2 I$ where $\sigma_e = 0.1$ and $\boldsymbol{\delta} = \mathbf{d} - \mathscr{M}(z)$. Here a single observation is artificially generated by $\mathbf{d} = \mathscr{M}(\mathbf{z}_{exact}) + \mathbf{e}$ where $\mathbf{z}_{exact} = 0.3$ and $\mathbf{e} \sim \mathscr{N}(0, \sigma_e^2)$.

The surrogates constructed by the GPML method and gPC-based SC method are shown in Fig. 1. Oscillation characteristics of Gibbs' phenomena are observed for the gPC-based SC method and the discontinuity of the forward model is captured well by the GPML method. To further examine the accuracy and efficiency of the proposed surrogate modeling method, we define the standardized mean squared error (SMSE) between the surrogate and the exact function as:

$$SMSE = \frac{1}{N_{test}} \sum_{1}^{N_{test}} (\mathscr{M}_S(\mathbf{z}_i) - \mathscr{M}(\mathbf{z}_i))^2 / Var[\mathscr{M}(\mathbf{z}_i)] \tag{18}$$

In this work the uniform grid with the dimension $N_{test} = 6000$ is used as the test points to evaluate the SMSE. Fig. 2 shows the SMSE with respect to the size of the training set. The GPML method performs better than the gPC-based SC method. After examining the forward model approximation, the inverse problem is solved and the results are shown in Fig. 3. The GPML surrogate has oscillations that are much more localized in space near the discontinuity, thus only a small discrepancy is observed nearby z=0.0 in Fig. 3, while the oscillations in gPC surrogate lead to much larger discrepancies at around z= 0.5 and z=0.8.

4.2. 1D Smooth function

After testing the proposed approach for a function with discontinuities, its application to a smooth function, i.e. $\mathscr{M}(\mathbf{z}) = sin(\pi\mathbf{z}/2)$, is investigated for completeness.

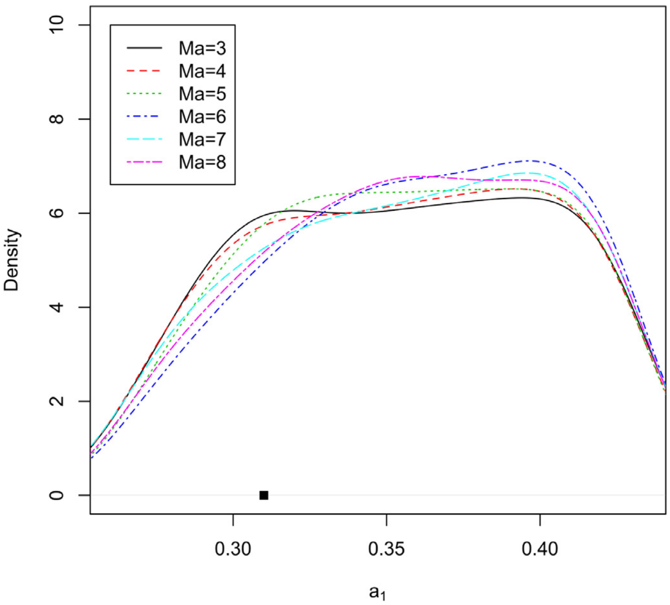


Fig. 20. The posterior distribution of a_1 over a wide range of Mach numbers.

The surrogates constructed by the GPML method and gPC-based SC method are shown in Fig. 4 and the standard mean squared errors are shown in Fig. 5. The surrogates constructed by both method match the exact function well. The SMSE results show that the GPML method is much more efficient than the gPC-based SC method.

4.3. 2D Step function

The GPML method can be easily extended to high dimensional problems, while the design of the nodal set for high dimensional problems is a delicate issue for the gPC-based SC method. Here we demonstrate the efficiency and the flexibility of the GPML method by applying it to a 2D step function:

$$\mathscr{M}(\mathbf{z}) = \begin{cases} 0, & z_1 + z_2 \in [-1, 0) \\ 1, & z_1 + z_2 \in [0, 1] \end{cases} \tag{19}$$

The surrogates constructed by the GPML method and gPC-based SC method are shown in Fig. 6 and the standard mean squared errors are shown in Fig. 7. As can be seen, the SMSEs reach the same level for the two methods but gPC-based SC method needs much more training samples than the GPML. In addition, the GPML exhibits an oscillatory behavior that is much less serious than the gPC approach.

5. Application to SST turbulence modeling

After testing the surrogate modeling method based on GPML for simple mathematical functions, we investigate the application of the proposed approach to the two-equation SST model for the hypersonic flow configurations.

5.1. SST turbulence model formulation

In this section, the two-equation SST model of Menter [22] is briefly described and the corresponding model parameters are identified. The SST model is formulated as:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\mu + \sigma_k \mu_t)\frac{\partial k}{\partial x_j}\right] + P_k - \beta^* \rho k \omega \tag{20}$$

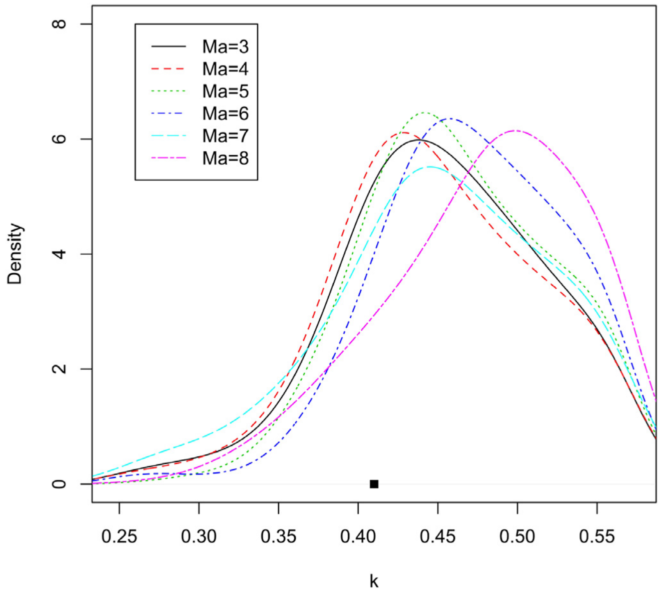


Fig. 21. The posterior distribution of κ over a wide range of Mach numbers.

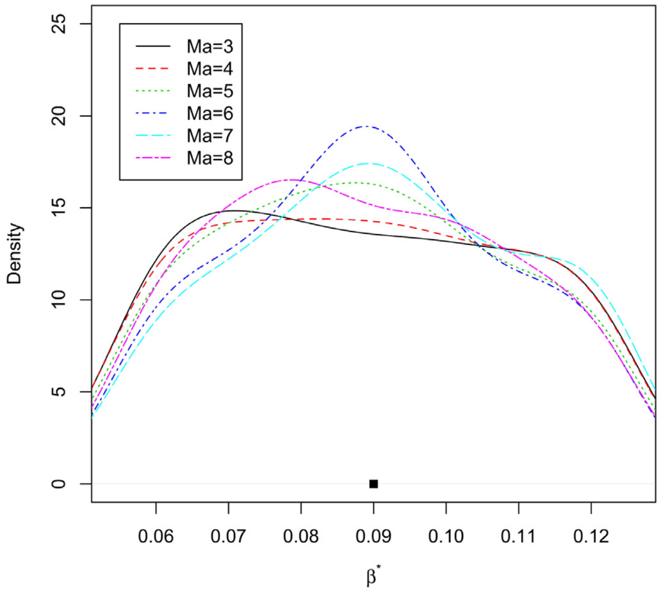


Fig. 22. The posterior distribution of β^* over a wide range of Mach numbers.

$$\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho u_j\omega)}{\partial x_j}=\frac{\partial}{\partial x_j}\left[(\mu+\sigma_\omega\mu_t)\frac{\partial\omega}{\partial x_j}\right]+\alpha\rho S^2-\beta\rho\omega^2$$
$$+2(1-F_1)\frac{\rho\sigma_{\omega_2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j} \qquad (21)$$

where F_1 is defined by:

$$F_1=tanh\left\{\left[min\left(max\left(\frac{\sqrt{k}}{0.09\omega d},\frac{500\mu}{\rho d^2\omega}\right);\frac{4\rho\sigma_{\omega_2}k}{CD_{k\omega}d^2}\right)\right]^4\right\} \qquad (22)$$

with

$$CD_{k\omega}=max\left(\frac{2\rho\sigma_{\omega_2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j};10^{-20}\right) \qquad (23)$$

The turbulent eddy viscosity is defined as:

$$\mu_t=min\left[\frac{\rho k}{\omega},\frac{\rho a_1 k}{\Omega F_2}\right] \qquad (24)$$

where F_2 is defined as:

$$F_2=tanh\left\{\left[max\left(2\frac{\sqrt{k}}{0.09\omega d},\frac{500\mu}{\rho d^2\omega}\right)\right]^2\right\} \qquad (25)$$

And all the model parameters are obtained by $\phi=F_1\phi_1+(1-F_1)\phi_2$, where ϕ represents the parameter in $\{\sigma_k,\ \sigma_\omega,\ \beta,\ \gamma\}$. The model parameters $\{a_1,\kappa,\beta^*,\sigma_{k_1},\sigma_{\omega_1},\beta_1,\sigma_{k_2},\sigma_{\omega_2},\beta_2\}$ are specified and $\{\gamma_1,\ \gamma_2\}$ are calculated by $\gamma_1=\beta_1/\beta^*-\sigma_{\omega_1}\kappa^2/\sqrt{\beta^*}$ and $\gamma_2=\beta_2/\beta^*-\sigma_{\omega_2}\kappa^2/\sqrt{\beta^*}$. The total 9 model parameters with their nominal values are reported in Table 1. The prior distribution needs to be specified to carry out the Bayesian calibration. In this work, the uniform distribution is used and the prior range is obtained by perturbing the nominal values by ±40%. This is determined by trial and error in order to keep the parameter range as large as possible while keeping the computational cost manageable.

5.2. Numerical methods and grid generation

All the numerical simulations are carried out in our in-house three-dimensional compressible Navier-Stokes solver. The compressible Navier-Stokes equations are solved with Roe's finite volume, upwind algorithm. By means of the monotone upstream-centered schemes for conservation laws interpolation of the primitive variables, the quantity in the inviscid fluxes is obtained. The viscous flux terms are calculated by a second-order central difference scheme. Lower Upper Symmetric Gauss Seidel (LU-SGS) scheme is used for temporal integration. The no-slip, constant wall temperature conditions are imposed. The mesh independence study has been carried out and the final mesh dimension is 161 in wall-normal direction and ranges from 401 to 801 in streamwise direction for different scenarios, depending on the size of the computational domain. The near wall spacing is $10^{-6}m$, making y^+ less than one over most part of the plate. The final mesh used for Case 1 in Table 2 is shown in Fig. 8. The leading edge of the plate is located at $X=0$ and the symmetrical boundary condition is imposed prior to the leading edge. In addition, the meshes used for other scenarios only differ with this one in streamwise dimension as the domain length is different for each scenario in order to include the DNS measurement location, which is determined by matching the boundary layer thickness.

5.3. Specification of the flow class and QoIs

The flow configurations of our primary interest are the hypersonic turbulent flows over flat plate with a wide range of Mach numbers and we are interested in predicting the skin friction coefficients (C_f) with quantified uncertainty. The collection of the corresponding experimental data is a key issue for solving the inverse problems, and a number of experimental databases and the DNS databases can be found in the literature. Here in this work, the skin frictions for the hypersonic flows over flat plate of the DNS database reported by Duan et al. [28] are used as the experimental data. The corresponding flow conditions and experimental data are reported in Table 2. The skin friction is reported at the location with a specified boundary layer thickness in this DNS database. In this work we calibrate our RANS results against their DNS results at the streamwise location where the boundary layer thickness is matched. In practice, we find that the flow remains laminar for certain parameter range and the corresponding boundary layer growth is very slow, thus a very long domain will be required in RANS simulations in order to match

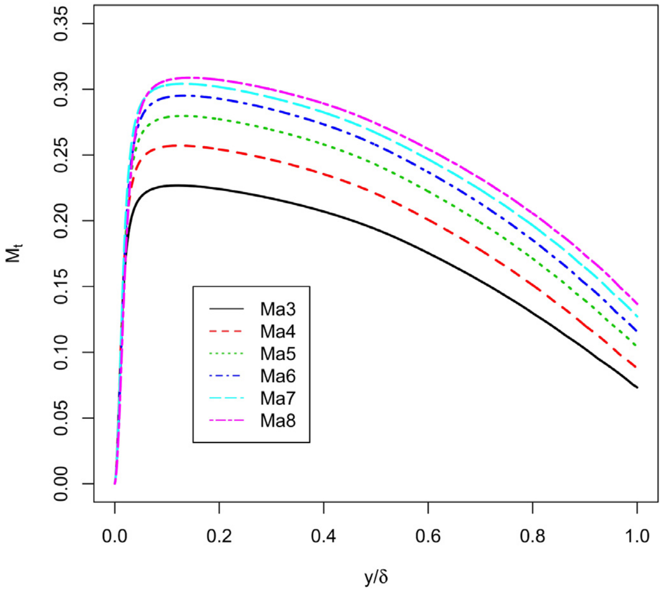


Fig. 23. the turbulence Mach number profiles $M_t(y)$ for all the flow cases listed in Table 2.

the boundary layer thickness with DNS. To prevent this situation, we further narrow down the prior range of a_1. It is worth noting that this has little impact on the final results, since the parameter range causing laminar results will lead to very small posterior probabilities if included in the prior. The final range of the model parameters is reported in Table 1.

5.4. Calibration results

The input-output mapping is denoted as $\mathscr{M}(\mathbf{z})$ with $\mathbf{z} = (a_1, k, \beta^*, \sigma_{k_1}, \sigma_{\omega_1}, \beta_1, \sigma_{k_2}, \sigma_{\omega_2}, \beta_2)$. The prior distribution of the model parameter is determined by Table 1 and the stochastic model is constructed following Section 2.1. The measurement error is neglected for stochastic model construction and the mismatches between the RANS predictions and DNS predictions are accounted for through the model inadequacy. 3×10^5 MCMC samples are generated to estimate the posterior distributions, with a burn-in length of 10^4.

First, the Mach 3 flow over a flat plate is investigated. The surrogates for $\mathscr{M}(\mathbf{z}_1)|_{\mathbf{z}=\bar{\mathbf{z}}\backslash\mathbf{z}_1}$ constructed by the GPML method and gPC-based SC method are compared. The design process based on GPML is shown in Fig. 9 and the surrogate constructed by gPC-based SC method is shown in Fig. 10. Each subfigure of Fig. 9 shows the variation of C_f measured at the point corresponding to the nominal thickness of Case 1 in Table 2 as a function of a_1 with the other model parameters taken fixed to their nominal values. As can be seen, the surrogate modeling error drops gradually with the inclusion of new design points. The final surrogate with GPML method performs well while oscillation characteristics are observed for the gPC-based SC method. All the following results are obtained with the GPML method.

The sensitivity analysis is then carried out for the selection of the HDMR terms. As can be seen from Fig. 11, a_1, κ and β^* are the first three most important dimensions and account for over 90% of the total variance of the first order terms. By adaptively selecting the HDMR terms, only f_0, $f_1(\mathbf{z}_1)$, $f_2(\mathbf{z}_2)$, $f_3(\mathbf{z}_3)$, $f_{(1,3)}(\mathbf{z}_1, \mathbf{z}_3)$ and $f_{(2,3)}(\mathbf{z}_2, \mathbf{z}_3)$ are included in the HDMR of $\mathscr{M}(\mathbf{z})$. The posterior distributions of the model parameters (a_1, k, β^*) and hyperparameter σ_h are shown in Figs. 12 and 13. As can be seen, the model parameter κ is well informed, and κ has the most significant impact on the surface drag as it affects significantly the log layer profile. The turbulence drag reduction research usually pays attention to the log layer profile in order to analyze the drag reduction mechanism. The results also show that the coefficients affecting the turbulence viscosity term (a_1) and the dissipation term of k equation(β^*) are also important for the drag prediction. This sensitivity analysis is true for all the scenarios listed in Table 2.

After obtaining the posterior distributions of the model parameters, the posterior model check is done by just propagating the model parameters through the SST model. We do not include the contribution of the model inadequacy terms in the following results so that we can also make predictions for other flow quantities. The results are shown in Fig. 14. The predicted skin friction with quantified uncertainty matches well with the DNS data, indicating that the Bayesian calibration is done successfully. The predictions of other quantities can also be carried out, such as the velocity profile, which is shown in Fig. 15. As can be seen, the predicted uncertainty overlaps well with the DNS data. However, the calibrated mean results are not so good as the nominal one in the outer region, which is reasonable since we calibrate the SST model against the surface drag (the near wall information) but not the velocity profile. The prediction of different quantities clearly depends on the dataset used during calibration.

In addition to the results calibrated against C_f, the calibration results against the velocity profile are also given here, as is shown in Figs. 16 and 17. The posterior prediction results of the velocity profile and the surface drag are shown in Figs. 18 and 19. As can be seen, the model parameters are clearly identified and the predictive uncertainties of both C_f and U_{vd} are greatly reduced. Even though the prediction of U_{vd} matches well with DNS data, the C_f prediction performs worse than the results calibrated against C_f itself (shown in Fig. 14). Thus we use only C_f (the QoIs) as our calibration dataset in this work. The inclusion of both surface drag and flow field information (e.g. velocity profiles) in the calibration dataset is also worth investigating in the future, in order to achieve a better compromise between the accurate prediction of QoIs and the reduction of the prediction uncertainty.

After the successful calibration of the Mach 3 flow, we carry out the Bayesian calibration for flows over a wide range of Mach numbers as listed in Table 2, and the posterior distributions of the model parameters are shown in Figs. 20–22. As can be seen, the calibration results are different for each scenario, and there is a clear parameter variability across scenarios. These results serve as the preparation step for the model predictions with quantified uncertainty. In the next section, a Bayesian scenario-averaging method is proposed to combine all these calibration results in order to make predictions for new scenarios.

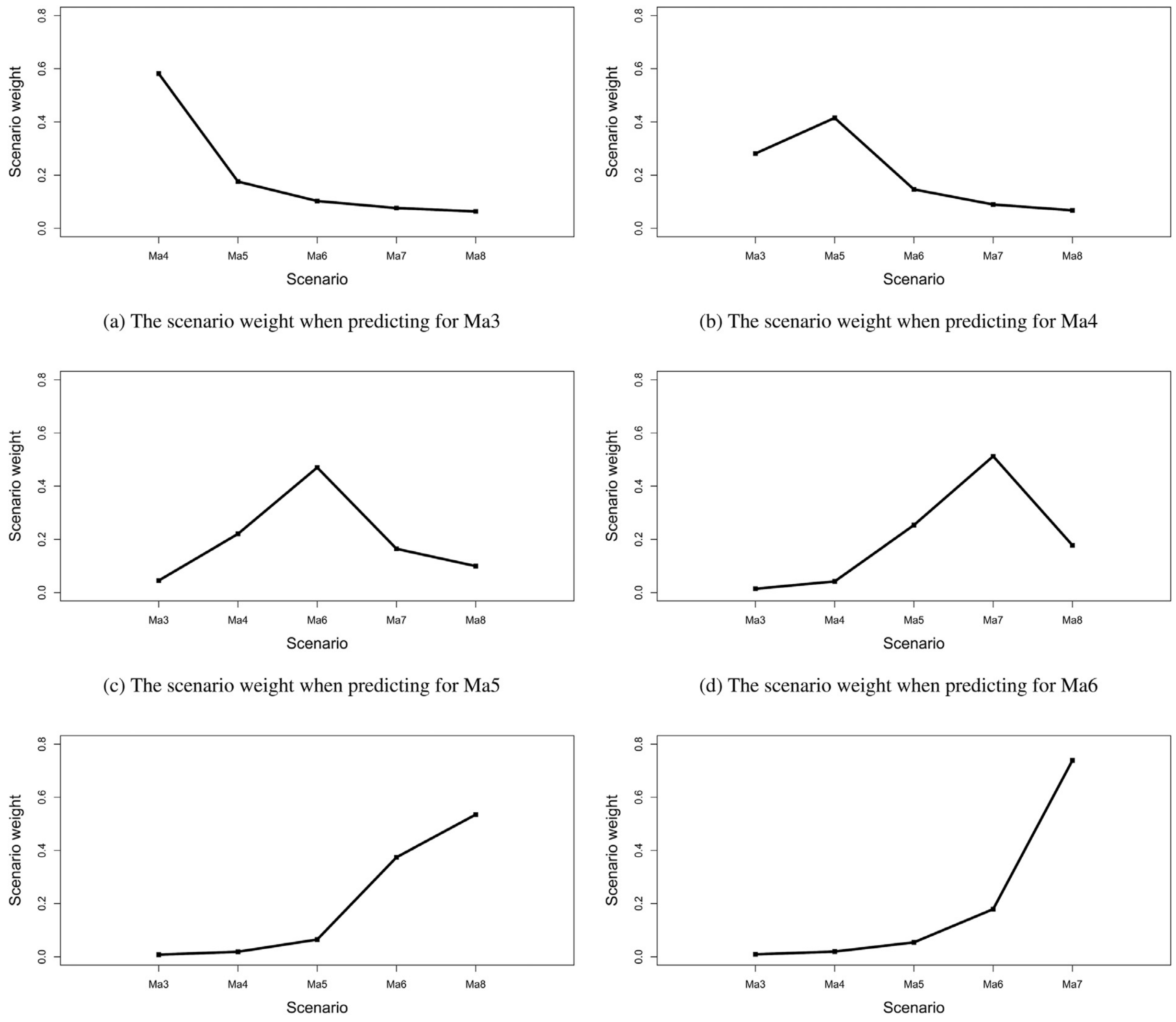
6. Model prediction with quantified uncertainties

6.1. Bayesian scenario-averaging

After Bayesian calibration, the model prediction for a new scenario is carried out by taking all the calibration results into account. The posterior distribution of the QoIs is expressed as:

$$p(\tilde{\mathbf{q}}|\mathbf{d}) = \sum_{i=1}^{I_s} \int p(\tilde{\mathbf{q}}, S_i, \mathbf{z}|\mathbf{d}_i) d\mathbf{z} = \sum_{i=1}^{I_s} \int p(\tilde{\mathbf{q}}|\mathbf{z}) p(\mathbf{z}|\mathbf{d}_i, S_i) P(S_i) d\mathbf{z} \tag{26}$$

where S_i represents the i^{th} scenario, $\mathbf{d}_i$ represents the corresponding experimental data, I_s represents the total number of scenarios and $P(S_i)$ represents the probability mass function (PMF) of the i^{th} scenario. Then the prediction mean and variance of the QoI are expressed as:



(a) The scenario weight when predicting for Ma3

(b) The scenario weight when predicting for Ma4

(c) The scenario weight when predicting for Ma5

(d) The scenario weight when predicting for Ma6

(e) The scenario weight when predicting for Ma7

(f) The scenario weight when predicting for Ma8

Fig. 24. The scenario weights determined as described in Section 6.2.

$$E[\tilde{\mathbf{q}}|\mathbf{d}] = \sum_{i=1}^{I_s} P(S_i) E_i[\tilde{\mathbf{q}}|\mathbf{d}_i] \tag{27}$$

$$Var[\tilde{\mathbf{q}}|\mathbf{d}] = \sum_{i=1}^{I_s} P(S_i) Var_i[\tilde{\mathbf{q}}|\mathbf{d}_i] + \sum_{i=1}^{I_s} P(S_i)(E_i[\tilde{\mathbf{q}}|\mathbf{d}_i] - E[\tilde{\mathbf{q}}|\mathbf{d}])^2 \tag{28}$$

where $E_i[\tilde{\mathbf{q}}|\mathbf{d}_i]$ and $Var_i[\tilde{\mathbf{q}}|\mathbf{d}_i]$ represent the posterior expectation and variance of the QoIs with the model parameters calibrated against the i^{th} scenario.

6.2. Determination of the scenario weights

The probability mass function $P(S_i)$ needs to be specified in order to make predictions. Here a physics-based method to determine the scenario weights is proposed, which is based on the mean squared value of the difference of certain quantities between scenarios. As the numerical simulations with SST turbulence models may be greatly undermined when the compressibility effects are important, we choose the turbulence Mach number, which is defined as follows, to characterize the compressibility effects and determine the corresponding weights of each scenario.

$$M_t(x, y) = \frac{\sqrt{2k(x, y)}}{a(x, y)} \tag{29}$$

where k is the turbulent kinetic energy, a is the local speed of sound and (x, y) represents the location in streamwise and wall-normal direction. Here we use only the profile $M_t(y)$ at the streamwise location where the skin friction is measured. Fig. 23 gives the corresponding turbulence Mach number profiles $M_t(y)$ for all the flow cases listed in Table 2. We denote the mean squared value of the difference of $M_t(y)$ between scenario S_i and S_j as $D_{i,\,j}$, then the scenario weight $P(S_j)$ when predicting for the scenario i is determined by:

$$P(S_j) = \frac{1/D_{i,j}}{\sum_{j=1}^{I_s} 1/D_{i,j}} \tag{30}$$

Here the purpose of using turbulence Mach number instead of external Mach number is to characterize the scenarios by flow field features, in order to extend the approach in the future to more general cases such as different wall temperatures, different geometries. In such cases, the turbulent Mach number may still be useful but the external one cannot characterize the compressibility effects correctly.

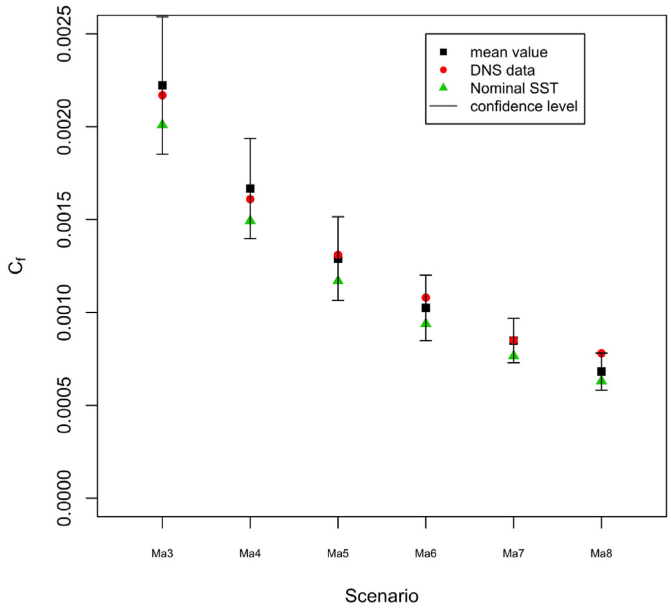


Fig. 25. The prediction mean of the QoIs with quantified uncertainty($\pm\sigma$). The DNS data and the nominal SST model results are also shown.

6.3. Prediction results

In this section, the model prediction is carried out, with the model parameters calibrated in Section 5.4. The prediction for the i^{th} scenario is done by excluding the calibration results of the i^{th} scenario and taking all the others into account. The scenario weights are determined as described in Section 6.2 and the results are shown in Fig. 24. After determining the scenario weights, the surrogate is constructed for the forward propagation and the posterior distribution of the QoIs is obtained according to Eqs. (26)–(28). The results are shown in Fig. 25. As can be seen, the prediction confidence level matches well with the DNS data, and the prediction mean values perform better than the nominal SST. As the prediction for each scenario does not make use of the corresponding DNS data of the same scenario, the results demonstrate the ability of the proposed approach in making reliable predictions for new, unknown scenarios.

7. Conclusion

In this work, an efficient Bayesian UQ approach is proposed. It is demonstrated that it can handle both continuous response surfaces and discontinuous surfaces, and the computational burden for the application in high dimensional problems is reduced. Its applications to simple mathematical functions are investigated, demonstrating its efficiency and accuracy for both continuous function and function with discontinuities. The computer design of the sampling points based on GPML is also proposed and the results show that the proposed method is promising in terms of both efficiency and extensibility to high dimensional problems.

After testing the proposed approach with simple mathematical functions, it is applied to the two equation SST turbulence model for hypersonic flow over flat plate over a wide range of Mach numbers. The sensitivity analysis shows that the model parameters a_1, κ and β^* are the first three most important dimensions and the inclusion of only the first-order and second-order terms in the HDMR is sufficient. The Bayesian calibration is done by employing the so-constructed surrogate and the posterior model check shows that the model predictions with quantified uncertainty match well with the DNS data, for both the variable used for calibration and the other flow variable, i.e. velocity profile.

When making predictions for a new flow scenario, the Bayesian scenario-averaging is employed and a method based on the mean squared values of the difference of chosen physical quantities between scenarios is proposed to determine the scenario weights. As the numerical simulations with SST turbulence models may be greatly undermined when the compressibility effects are important, we choose the turbulence Mach number to characterize the compressibility effects and determine the corresponding weights of each scenario. The results show that the prediction mean values match well with the DNS data and the corresponding uncertainties are well captured. In terms of efficiency, 50 code runs are typically required for the final prediction of each scenario, comparing tens of thousands of code runs for direct model evaluations. The choice of the quantities to characterize each scenario depends on both the problems at hand and the computer models that are used in the simulations. When applying the proposed approach with other computer models or for other flow scenarios, the choice may be different. In addition, through Bayesian scenario-averaging, the predicted uncertainties will be quite large if the predicted scenario is different from all the scenarios in the calibration database. This is due to the automatic account of the parameter variability across scenarios through the 2nd term in Eq. (28). Thus this approach may capture the possible inappropriateness of the calibration database and show them in terms of predicted uncertainties.

Future work may involve the further inclusion of the other scenarios in the DNS database reported in the literature and the investigation of the impact of using different quantities as the calibration dataset. After calibration with the reported database, it is very promising to apply the proposed approach to more complex flow configurations, e.g. HIFiRE models, so that the predicted uncertainties can provide useful information for decision-making in real engineering applications.

References

[1] Cheung SH, Oliver TA, Prudencio EE, Prudhomme S, Moser RD. Bayesian uncertainty analysis with applications to turbulence modeling. Reliab Eng Syst Safety 2011;96(9):1137–49.
[2] Oliver TA, Moser RD. Bayesian uncertainty quantification applied to rans turbulence models. In: Journal of Physics: Conference Series, 318. IOP Publishing; 2011. p. 042032.
[3] Edeling W, Cinnella P, Dwight RP, Bijl H. Bayesian estimates of parameter variability in the k–ε turbulence model. J Comput Phys 2014;258:73–94.
[4] Ray J, Lefantzi S, Arunajatesan S, Dechant L. Bayesian parameter estimation of ak-ε model for accurate jet-in-crossflow simulations. AIAA J 2016:2432–48.
[5] Margheri L, Meldi M, Salvetti MV, Sagaut P. Epistemic uncertainties in rans model free coefficients. Comput Fluid 2014;102:315–35.
[6] Schaefer J, Hosder S, West T, Rumsey C, Carlson J-R, Kleb W. Uncertainty quantification of turbulence model closure coefficients for transonic wall-bounded flows. AIAA J 2016;55(1):195–213.
[7] Zhang J, Fu S. An efficient bayesian uncertainty quantification approach with application to k-ω-γ transition modeling. Comput Fluid 2018;161:211–24.
[8] Ma X, Zabaras N. An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. J Comput Phys 2010;229(10):3884–915.
[9] Marzouk Y, Xiu D. A stochastic collocation approach to bayesian inference in inverse problems. Commun Comput Phys 2009;6(4):826–47.
[10] Kennedy MC, O'Hagan A. Bayesian calibration of computer models. J R Statist Soc 2001;63(3):425–64.
[11] Fu S, Wang L. Rans modeling of high-speed aerodynamic flow transition with consideration of stability theory. Prog Aerosp Sci 2013;58:36–59.
[12] Smoljak SA. Quadrature and interpolation formulae on tensor products of certain function classes. Dokl Akad Nauk SSSR 1963;4(5):240–3.
[13] Ma X, Zabaras N. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. J Comput Phys 2009;228(8):3084–113.
[14] Witteveen JAS, Iaccarino G. Simplex stochastic collocation with eno-type stencil selection for robust uncertainty quantification. J Comput Phys 2013;239(8):1–21.

[15] Edeling W, Dwight RP, Cinnella P. Simplex-stochastic collocation method with improved scalability. J Comput Phys 2016;310:301–28.

[16] Rasmussen CE, Williams CKI. Gaussian processes for machine learning (adaptive computation and machine learning). Cambridge, Massachusetts, USA: The MIT Press; 2005.

[17] Higdon D, Kennedy M, Cavendish JC, Cafeo JA, Ryne RD. Combining field data and computer simulations for calibration and prediction. SIAM J Sci Comput 2004;26(2):448–66.

[18] Williams B, Higdon D, Gattiker J, Moore L, McKay M, Keller-McNulty S, et al. Combining experimental data and computer simulations, with an application to flyer plate experiments. Bayesian Anal 2006;1(4):765–92.

[19] Higdon D, Gattiker J, Williams B, Rightley M. Computer model calibration using high-dimensional output. J Am Stat Assoc 2008;103(482):570–83.

[20] Roy CJ, Blottner FG. Review and assessment of turbulence models for hypersonic flows. Prog Aerosp Sci 2006;42(7–8):469–530.

[21] Rumsey CL. Compressibility considerations for k-w turbulence models in hypersonic boundary-layer applications. J Spacecraft Rockets 2010;47(1):11–20.

[22] Menter FR. Two-equation eddy-viscosity turbulence models for engineering applications. AIAA J 1994;32(8):1598–605.

[23] Dow E, Wang Q. Quantification of structural uncertainties in the k–ω turbulence model. AIAA Paper 2011;1762:2011.

[24] Emory M, Larsson J, Iaccarino G. Modeling of structural uncertainties in reynolds-averaged navier-stokes closures. Phys Fluid 2013;25(11):110822.

[25] Gorlé C, Iaccarino G. A framework for epistemic uncertainty quantification of turbulent scalar flux models for reynolds-averaged navier-stokes simulations. Phys Fluid 2013;25(5):055105.

[26] Duraisamy K, Zhang ZJ, Singh AP. New approaches in turbulence and transition modeling using data-driven techniques. In: 53rd AIAA Aerospace Sciences Meeting; 2015. p. 1284.

[27] Xiao H, Wu J-L, Wang J-X, Sun R, Roy C. Quantifying and reducing model-form uncertainties in reynolds-averaged navier–stokes simulations: a data-driven, physics-informed bayesian approach. J Comput Phys 2016;324:115–36.

[28] DUAN L, BEEKMAN I, MARTIN MP. Direct numerical simulation of hypersonic turbulent boundary layers. part 3. effect of mach number. J Fluid Mech 2011;672(4):245–67.

[29] Edeling W, Cinnella P, Dwight RP. Predictive rans simulations via bayesian model-scenario averaging. J Comput Phys 2014;275:65–91.

[30] Haario H, Saksman E, Tamminen J. An adaptive metropolis algorithm. Bernoulli 2001;7(2):223–42.

[31] R Core Team. R: alanguage and environment for statistical computing. R Foundation for Statistical Computing; Vienna, Austria; 2017. URL https://www.R-project.org/.

[32] Chivers C. MHadaptive: general Markov Chain Monte Carlo for Bayesian inference using adaptive Metropolis-Hastings sampling; 2012. R package version 1.1–8; URL https://CRAN.R-project.org/package=MHadaptive.

[33] Griebel M, Holtz M. Dimension-wise integration of high-dimensional functions with applications to finance. J Complex 2010;26(5):455–89.

[34] Zuniga M.M., Ko J. GPC: generalized polynomial chaos; 2014. R package version 0.1; URL https://CRAN.R-project.org/package=GPC.

[35] Kalaitzis A., Honkela A., Gao P., Lawrence N.D. gptk: gaussian processes Tool-Kit; 2014. R package version 1.08; URL https://CRAN.R-project.org/package=gptk.