Journal of Computational Physics 305 (2016) 758–774

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A paradigm for data-driven predictive modeling using field inversion and machine learning

Eric J. Parish, Karthik Duraisamy*

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

Article history:
Received 9 July 2015
Received in revised form 27 September 2015
Accepted 7 November 2015
Available online 10 November 2015

Keywords:
Data-driven modeling
Machine learning
Closure modeling

ABSTRACT

We propose a modeling paradigm, termed field inversion and machine learning (FIML), that seeks to comprehensively harness data from sources such as high-fidelity simulations and experiments to aid the creation of improved closure models for computational physics applications. In contrast to inferring model parameters, this work uses inverse modeling to obtain corrective, spatially distributed functional terms, offering a route to directly address model-form errors. Once the inference has been performed over a number of problems that are representative of the deficient physics in the closure model, machine learning techniques are used to reconstruct the model corrections in terms of variables that appear in the closure model. These reconstructed functional forms are then used to augment the closure model in a predictive computational setting. As a first demonstrative example, a scalar ordinary differential equation is considered, wherein the model equation has missing and deficient terms. Following this, the methodology is extended to the prediction of turbulent channel flow. In both of these applications, the approach is demonstrated to be able to successfully reconstruct functional corrections and yield accurate predictive solutions while providing a measure of model form uncertainties.

Even with the tremendous growth in computational power during the past decade, simulations based on first-principles models of physical systems remain prohibitively expensive for most practical problems. As a result, one has to rely on coarse-grained models to characterize or predict the overall state of a complex system or its statistical properties. Derivation of the more affordable models, however, involves a number of additional assumptions that can limit their accuracy. The pursuit of accurate *closures* in coarse-grained or intermediate/low-fidelity models is typically a central issue and pacing item in many scientific disciplines. At the same time, it is becoming feasible to run first-principles or high-fidelity simulations under idealized conditions and in some regimes of interest. Concurrently, experimental techniques have evolved to a point where high resolution information can be provided at many scales, including those that are of direct relevance to problems of interest for physicists and engineers. Against this backdrop, data mining techniques have already made their mark in many disciplines of science and engineering by providing improved physical insight as well as quantitative data for modeling. Unprecedented opportunities exist in going one step further and directly utilizing available data to improve and generate *predictive models* that can be used in *practical* analysis and design in a *robust* manner.

Physical modeling has always been data-driven to a degree. Typically, a theory or set of theories are formulated and unknown model coefficients and functions are empirically determined by correlating the response of the model with available data. Over the past decade, more formal calibration procedures have emerged in many different fields of application.

* Corresponding author.
E-mail addresses: parish@umich.edu (E.J. Parish), kdur@umich.edu (K. Duraisamy).

http://dx.doi.org/10.1016/j.jcp.2015.11.012
0021-9991/© 2015 Elsevier Inc. All rights reserved.

Discipline-specific reviews are presented in Navon [1] for data assimilation in weather forecasting and Kerschen et al. [2] for system identification in Structural Dynamics. Given some data $\mathbf{G}_d$ and a model output $\mathbf{G}_m(\mathbf{Q}, \lambda)$, where $\mathbf{Q}$ are state variables and λ are model parameters, Frequentist (typically Least Squares) or Bayesian procedures are formulated to infer optimal values of λ. This is usually accomplished by minimizing the difference between the data and the model output. The Least Squares procedure is conceptually simple and can offer probability measures on the output, whereas the Bayesian approach can be formulated more rigorously and can account for a more precise prescription of prior knowledge and probability structure, though at a much higher expense. Nevertheless, both types of techniques have been successfully used for parameter estimation.

Errors in the underlying structure of the model may result in inadequacies [3], even if the best possible set of parameters have been inferred. Predicted values of the outputs may never match the true value in a deterministic or statistical sense for many models. A widely used approach to address model inadequacy is the Bayesian calibration framework of Kennedy and O'Hagan (Ref. [4] and its derivatives [5,6], etc.). The essence of their approach is to represent the output quantity of interest as

$$\mathbf{G}_d = \mathbf{G}_m(\mathbf{Q}, \lambda) + \delta(\mathbf{Q}) + \varepsilon, \tag{1}$$

where δ is the model discrepancy term and ε represents measurement errors. Typically, Gaussian Process models are assumed for $\mathbf{G}_m$, δ, and ε and Bayesian inversion is used to derive a posterior distribution for the hyperparameters of the Gaussian process as well as the parameters λ. A key advantage of this approach is that the procedure is general enough that, in theory, a deep understanding of the underlying physical system is not required to quantify the discrepancy, although physically motivated priors may be required to guarantee identifiable solutions [7]. While this approach has been successful in exposing the idea of model discrepancy and has been widely employed, several shortcomings have been noted [8,9]. A primary drawback is that there is no logical way to distinguish the role of the parameters λ and the discrepancy function δ. Additional concerns include that (i) the discrepancy model δ is an empiricism that is unconstrained by physical laws, (ii) in its originally proposed (and widely employed) form based on Gaussian processes, non-Gaussian phenomena cannot be accurately represented, and (iii) the definition of this discrepancy is tied to a particular quantity of interest and does not inform the prediction of other quantities at a more fundamental level.

Over the past few years, researchers have focused on addressing some of the aforementioned limitations. It has to be mentioned that the process of addressing model inadequacies cannot ignore the underlying physics, and hence many attempts at model calibration may be hidden in discipline-specific literature and thus a comprehensive review is difficult.

Cheung et al. [10] and Edeling et al. [11] use the concept of Stochastic Model Class [12] to calibrate the coefficients of underlying sub-models (in this case, eddy-viscosity-based turbulence models). In this approach, given a number of models, the plausibility of each model class M_i is quantified by appropriate prior and likelihood functions. Again, Bayes' theorem is used to find the posterior distribution $p(\theta|\mathbf{G}_d, M_i)$. The above approaches can be successful when a large amount of data is available and when the model parameters are fixed.

Berliner et al. [13] try to improve one-dimensional ice-sheet models by incorporating additive parameters to specific model components (for instance, the basal shear stress). More recently, Sargsyan et al. [14] have introduced random variables into kinetic rate coefficients in reaction models and reduce the calibration problem to one of estimating the kernel density of the random variables. These works embed the discrepancy function into the model or sub-model equations (rather than at the output level) and thus operate under conditions that satisfy physical constraints (for instance, conservation properties [14]). However, as discussed by Loeppky et al. [15], accounting for the discrepancy can also lead to poor choices of the parameters λ, which should now be viewed as *tuning* parameters. Soize [16] addresses model form uncertainties directly by constructing *stochastic models of the operators* of structural mechanics instead of introducing a probabilistic model of the prediction errors. The approach uses random matrix theory [17] to construct the prior probability distribution of matrix operators in a structural dynamic system. This approach has been used to represent modeling uncertainty that appears in a class of problems with specific structure (*i.e.* classical second order dynamical systems typical of structural mechanics). It is not clear how this method can be extended to more general non-linear large scale systems that do not have a matrix-based dynamical system representation.

The objective of the present work is to move beyond parameter estimation and seek detailed distributions of *field variables* that can effectively address structural uncertainties. The underlying Bayesian problem is one of field inversion. Functional relationships are extracted from the field variable distribution using *machine learning* techniques once the inversion over a sufficiently large number of problems is complete. The ultimate goal is to offer predictive capability based on the knowledge gained from various datasets. This two-step process, which we refer to as field inversion and machine learning (FIML) is demonstrated in a predictive setting to account for model form uncertainties in a scalar ordinary differential equation as well as in a turbulence modeling application.

1. Mathematical setting

Consider a physical system that is governed by a set of non-linear equations (partial differential or otherwise). The truth-model of the system is given by

$$\mathcal{R}_T(\mathbf{Q}_T(\mathbf{x}, t)) = \mathbf{0}, \tag{2}$$

along with well-posed initial and boundary conditions. The operator $\mathcal{R}_T$ contains the governing equations of the system while $\mathbf{Q}_T$ contains the model variables. The physical system is modeled by

$$\mathcal{R}_m(\mathbf{Q}_m(\mathbf{x},t),\mathcal{M})=\mathbf{0}, \tag{3}$$

where $\mathcal{R}_m \neq \mathcal{R}_T$ and it is possible $\mathbf{Q}_m$ involves different model variables than $\mathbf{Q}_T$. These model equations contain unclosed functions/operators $\mathcal{M}$ that arise either from a deficient understanding of the underlying physics or from coarse-graining.[1] In other words, if exact values of $\mathcal{M}$ are inserted into the model equations $\mathcal{R}_m$, one can obtain accurate solutions[2] that satisfy the truth model. However, $\mathcal{M}$ is usually not determinable from first principles and so to provide closure to the above set of equations, a secondary set of model variables $\mathbf{Q}_s(\mathbf{x},t)\in\mathbb{R}^{n_s}$ are typically introduced. These secondary variables may have associated model equations $\mathcal{R}_s(\mathbf{Q}_m,\mathbf{Q}_s)=\mathbf{0}$, which themselves will have inexact aspects. Using these equations, an approximation for $\mathcal{M}$ is typically sought as a function $\mathbf{M}(\mathbf{Q}_m,\mathbf{Q}_s)$. Thus, the model of the system can be represented as

$$\mathcal{R}(\mathbf{Q},\mathbf{M}(\mathbf{Q}))=\mathbf{0}; \tag{4}$$

with $\mathcal{R}$ and $\mathbf{Q}$ denoting the combination $\{\mathcal{R}_m;\mathcal{R}_s\}$ and $\{\mathbf{Q}_m;\mathbf{Q}_s\}$ respectively.

In the proposed approach, the model in Eq. (4) will be replaced by a stochastic system

$$\mathcal{R}(\mathbf{Q},\mathbf{M}(\mathbf{Q},\beta(\omega)))=\mathbf{0}, \tag{5}$$

where $\beta:\Omega\ni\omega$ is a random function that is a result of a data-driven inversion/machine learning process. Determining β is the essence of this work. A significant aspect of our approach is that realizations (sampled in the space Ω) of β are not parameters, but *spatio-temporally varying field variables*. To be useful in *predictive modeling*, however, the spatio-temporal dependence of β has to be transformed by extracting the functional relationship $\beta(\eta)$, where $\eta(\mathbf{Q})$ are input features available in the closure model. The input features are generally extensions of the model variables. **Such a functional relationship will have to be elicited while considering the output of a large number of inverse problems that are representative of the physical phenomena to be modeled.** Note that the stochastic nature of β will be implicit in the discussion below and thus the dependence on ω will not be included for clarity.

It is beneficial to relate the concepts defined above to a physical system. As an example, direct numerical simulations (DNS) of the Navier–Stokes equations can be considered to be the truth model of the system, $\mathcal{R}_T$. The computational cost associated with DNS can be quite high in practical flows, and thus filtering or averaging the solutions can lead to a *modeled* set of equations such as Large Eddy (LES) or Reynolds-averaged Navier–Stokes (RANS) simulations, represented by $\mathcal{R}_m$. The averaging/filtering procedure produces unclosed terms $\mathcal{M}$. It is not possible to derive an exact set of closed equations for $\mathcal{M}$; instead the solution approach is to develop a closure model with a secondary set of equations, $\mathcal{R}_s$, which will be dependent on the averaged/filtered quantities.

In the proposed approach, the stochastic function β is introduced to the model system. The inverse methods described in Section 2 are used to infer the probability density function (PDF) of $\beta(\mathbf{x})$. To be of use in a predictive setting, inferred solutions for β over a wide class of problems are used as inputs to the machine learning methods described in Section 3. The machine learning algorithms are then used to extract a functional relationship for $\beta(\eta)$, with η being input features (such as a notional time-scale of the turbulence) available from the closure model.

2. Field inversion

The challenge of creating the stochastic system in Eq. (5) is in the estimation of the distribution of the function $\beta(\eta)$. Model discrepancy inhibits direct extraction of the functional form of $\beta(\eta)$ from available data. Instead, an inverse problem is posed to infer the distribution of β such that realizations of the stochastic system are consistent with underlying physics; potentially both in the mean and higher order statistics. Bayesian inversion is used to obtain $\beta(\omega)$ in the form of functional corrections. **The functional correction is obtained by inferring β at every grid point in the computational domain**. The process of inferring β can be thought of as follows: We start by having an estimate of β, along with a certain amount of confidence in that estimate. This is the prior probability of β, given by $p(\beta)$. Next, we observe an external system and obtain an observational dataset $\mathbf{d}$ along with its associated uncertainty. For a given β, there is some probability that the model will reproduce the dataset $\mathbf{d}$. This is given by the likelihood function $h(\mathbf{d}|\beta)$. Given $p(\beta)$ and $h(\mathbf{d}|\beta)$, there exists some probability of β *given* the observations $\mathbf{d}$. This is the posterior probability $q(\beta|\mathbf{d})$. The goal of the inverse is to obtain $q(\beta|\mathbf{d})$. Mathematically, the posterior probability distribution is given by Bayes' theorem

$$q(\beta|\mathbf{d})=\frac{h(\mathbf{d}|\beta)p(\beta)}{c}, \tag{6}$$

where $c=\int h(\mathbf{d}|\beta)p(\beta)d\beta$. The solution of Eq. (6) can be made tractable via assumptions regarding the distribution of $\mathbf{d}$ and β. In the case that the dataset $\mathbf{d}$ and random function β are Gaussian, and the distribution $h(\mathbf{d}|\beta)$ is Gaussian, it can be

[1] Most continuum models have this issue in one form or the other.

[2] In this paper, the focus is on the closure model and not on issues such as numerical discretization errors, initial/boundary condition uncertainties, etc.

shown [18] that the problem of determining the distribution of β is reduced to estimating the *maximum a posteriori* (MAP) solution, which is found by solving a deterministic optimization problem

$$\beta_{map} = arg\ \min \frac{1}{2}\left[\left(\mathbf{d} - h(\beta)\right)^T \mathbf{C_m}^{-1}\left(\mathbf{d} - h(\beta)\right) + \left(\beta - \beta_{prior}\right)^T \mathbf{C}_\beta{}^{-1}\left(\beta - \beta_{prior}\right)\right], \tag{7}$$

where $\mathbf{C_m}$ and $\mathbf{C}_\beta$ are the observational and prior covariance matrices, respectively. The observational covariance is determined by the statistics of the observed dataset $\mathbf{d}$ and the prior covariance is determined by prior knowledge of the system. The parameter to observable map $h(\beta)$ is a subset of the governing equations. The parameter being optimized in Eq. (7) is β. The term being minimized is referred to as the cost function $\mathfrak{J}$; *i.e.*

$$\beta_{map} = arg\ \min \mathfrak{J}. \tag{8}$$

The dimensionality of the optimization problem scales with the number of discrete parameters being optimized. In this case, β is being optimized at every point in the domain so the dimensionality of the optimization problem scales with the number of mesh points. In the linear case, the covariance of the posterior is given by the inverse of the Hessian of the cost function $\mathfrak{J}$ evaluated at the MAP point

$$\mathbf{C}_{\beta_{map}} = \mathbf{H}^{-1}\bigg|_{\beta_{map}} = \left(\frac{d^2\mathfrak{J}}{d\beta_i d\beta_j}\right)^{-1}\bigg|_{\beta_{map}}. \tag{9}$$

In the non-linear case, Eq. (9) becomes an *approximation* that is a result of a linearization about the MAP point. Once the MAP solution and posterior covariance are found, realizations of β can be drawn from the posterior distribution. A standard method is to perform a Cholesky decomposition on $\mathbf{C}_{\beta_{map}}$ such that

$$\mathbf{RR^T} = \mathbf{C}_{\beta_{map}}. \tag{10}$$

Random samples of the posterior can then be drawn by

$$\beta = \beta_{map} + \mathbf{Rs}, \tag{11}$$

where $\mathbf{s}$ is a vector, the components of which are independent standard normal variates.

It is well-recognized that the Gaussian assumption is a strong one. In a general setting, the data $\mathbf{d}$ and prior β may not be truly Gaussian. In non-linear problems, even if $\mathbf{d}$ and the prior β are Gaussian, the posterior distribution may not be Gaussian in nature. In our approach, we consider the approximation to make our approaches feasible in high-dimensional problems. The error introduced by the Gaussian assumption is problem-dependent and can be estimated by analyzing the model response, but in the non-linear case, the posterior distribution must be viewed as an approximation. Computing non-Gaussian posterior distributions require expensive sampling based methods, such as Markov chain Monte Carlo (MCMC) methods (see Appendix A). These methods are especially costly in high-dimensional state spaces. The inversion procedure described in this paper is scalable, while sampling based methods are not.

In the following subsections, different aspects of the inversion process are discussed.

2.1. Gradient and Hessian computations

For field inversion, the dimensionality of the inversion scales with the number of mesh points. Though simplifications may be performed by constructing a surrogate representation of β over the computational domain, we pursue the more detailed approach of estimating β at every grid point in the computational domain. The resulting optimization problem is high-dimensional and efficient methods of minimizing the cost function are needed. Gradient-based methods are used to solve the inverse problem in this work. These methods require derivatives with respect to a large number of parameters, which are efficiently calculated using a discrete adjoint [19] formulation. To determine the gradient, the adjoint equation is first solved for ψ,

$$\left(\frac{\partial \mathcal{R}}{\partial \mathbf{Q}}\right)^T \psi = -\left(\frac{\partial \mathfrak{J}}{\partial \mathbf{Q}}\right)^T, \tag{12}$$

where $\mathfrak{J}$ is the cost function, $\mathcal{R}$ is the residual of the primal equations, and $\mathbf{Q}$ are the model variables. The gradient is then computed using

$$G = \frac{d\mathfrak{J}}{d\beta} = \frac{\partial \mathfrak{J}}{\partial \beta} + \psi^T \frac{\partial \mathcal{R}}{\partial \beta}. \tag{13}$$

The optimization problem is solved using BFGS [20], or in problematic cases, using steepest decent. The solution of the adjoint equation requires the computation of the Jacobian of the primal equations, which is calculated analytically.

To determine the posterior covariance, a Hessian calculation is required. In this work, an adjoint-adjoint method is used to compute the Hessian. For a system with M discrete model variables and N optimization parameters (the model variables are $\mathbf{Q}$ at each grid point and the optimization parameter is β at each grid point), the Hessian is computed by [21]:

$$H_{ij} = \frac{\partial^2 \mathfrak{J}}{\partial \beta_i \partial \beta_j} + \psi_m \frac{\partial^2 \mathcal{R}_m}{\partial \beta_i \partial \beta_j} + \mu_{i,m} \frac{\partial \mathcal{R}_m}{\partial \beta_j} + \nu_{i,m} \frac{\partial^2 \mathfrak{J}}{\partial Q_n \partial \beta_j} + \nu_{i,n} \psi_m \frac{\partial^2 \mathcal{R}_m}{\partial Q_n \partial \beta_j} \qquad m, n \in [1, M] \tag{14}$$

where

$$\nu_{i,n} \frac{\partial \mathcal{R}_m}{\partial Q_n} = -\frac{\partial \mathcal{R}_m}{\partial \beta_i} \qquad m, n \in [1, M] \quad i \in [1, N] \tag{15}$$

$$\mu_{i,m} \frac{\partial \mathcal{R}_m}{\partial Q_k} = -\frac{\partial^2 F}{\partial \beta_i \partial u_k} - \psi_m \frac{\partial^2 \mathcal{R}_m}{\partial \beta_i \partial Q_k} - \nu_{i,n} \frac{\partial^2 \mathfrak{J}}{\partial Q_n \partial Q_k} - \nu_{i,n} \psi_m \frac{\partial^2 \mathcal{R}_m}{\partial Q_n \partial Q_k} \qquad k, m, n \in [1, M] \quad i \in [1, N]. \tag{16}$$

A low-rank approximation is useful for a diagonal observational covariance matrix. A diagonal covariance assumes that the data are uncorrelated. This is a reasonable approximation when measurement error dominates the observational variance. It is additionally relevant for cases where data is not available to build a complete covariance matrix. For a cost function with a diagonal observational covariance matrix and no prior, the cost function simplifies to

$$\mathfrak{J} = \frac{1}{2}\left[\sum_{i}^{M} \left(\frac{h(\beta)_i - d_i}{\sigma_i}\right)^2\right]. \tag{17}$$

To approximate the Hessian, M scalar valued functions are defined,

$$f_i(\beta) = \frac{h(\beta)_i - d_i}{\sigma_i} \qquad \text{for } i = 1, 2, \ldots, M. \tag{18}$$

The gradient of the non-regularized cost function can computed as

$$\nabla \mathfrak{J}(\beta) = \sum_{i=1}^{M} \nabla f_i(\beta)^2 = \sum_{i=1}^{M} 2 f_i(\beta) \nabla (f_i(\beta)). \tag{19}$$

Or equivalently

$$\nabla \mathfrak{J}(\beta) = 2\mathbf{J}(\beta)^T \mathbf{F}(\beta), \tag{20}$$

where

$$\mathbf{F} = \begin{bmatrix} f_1(\beta) & f_2(\beta) & \ldots & f_M(\beta) \end{bmatrix}^T. \tag{21}$$

The Jacobian of the scalar valued functions can then be used to approximate the Hessian by

$$\mathbf{H} \approx \mathbf{J}(\beta)^T \mathbf{J}(\beta). \tag{22}$$

The contribution of the prior is easily computed analytically. It is noted that calculation of the low-rank Hessian involves computing adjoints for each of the scalar valued functions. The low-rank approximation was generally found to be an excellent approximation to the true Hessian near the MAP point. For cases where the inversion was severely ill-conditioned, the approximate Hessian calculation generally produced a positive definite covariance, while the exact Hessian did not.

2.2. Observational covariance

In the case of an uninformed prior with a high covariance, the magnitude of the observational covariance will generally be much less than that of the prior. The solution of the inverse problem is expectedly sensitive to the specification of $\mathbf{C_m}$. In general, $\mathbf{C_m}$ is determined by the statistics of the observational data. The availability of observational statistics varies from case to case; and it is thus important to quantify the performance of the inversion for different forms of $\mathbf{C_m}$. In this work, three different models are considered. The simplest possible model we consider takes the form

$$\mathbf{C_m} = \sigma_{obs}^2 \mathbf{I}, \tag{23}$$

where σ_{obs} is a scalar that is representative of some mean variance of the observations. Such a model neglects all covariances. The second model considered is an extension of the above

$$\mathbf{C_m} = \vec{\sigma}_{obs}^2 \mathbf{I}, \tag{24}$$

where $\vec{\sigma}_{obs}$ is a vector containing the variances for each observation. This model assumes that $\vec{\sigma}_{obs}$ can be determined from available data. The third model considered assumes the availability of a complete set of statistics, in which case the exact covariance matrix of the data vector $\mathbf{D}$ is given by

$$\mathbf{C_m} = E\left[\left(D_i - \overline{D_i}\right)\left(D_j - \overline{D_j}\right)\right], \tag{25}$$

where E is the expectation operator.

The complexity of the observational covariance has a minimal impact on the computational cost of the inverse problem. The bandwidth of the covariance matrix affects the cost of evaluating the objective function and its derivatives. This does not play a significant role in the computational cost, although analytic derivatives of the objective function become unwieldy as the bandwidth increases. For the cases considered, it was observed that the optimization algorithm converged significantly faster using the full covariance matrix.

3. Machine learning

The inversion produces solutions of the correction β that are in a spatio-temporal (or in the present work, spatial) form, for specific problems. If the inversion is performed over a large number of problems and objective functions, *problem-specific inference* can be converted to *general modeling knowledge* via Supervised machine learning [22,23] algorithms. These techniques can be used to elicit the functional relation $\beta(\eta)$ where $\eta(\mathbf{Q})$ are the model input features. The examples provided in this paper use Gaussian Processes [24] (GPs), and the interaction between the inverse and ML formulation is limited to variances, *i.e.* the ML algorithm does not use the entire covariance matrix generated in the inversion. In the GP formulation, it is assumed that the output function at the training points, $\mathbf{Y}_{train}$ (where $\mathbf{Y}_{train}$ consists of inferred spatio-temporal fields of β for a wide class of problems), is drawn from the distribution

$$\mathbf{Y}_{train} \approx \mathcal{N}(0, \phi + \lambda \mathbf{I}), \tag{26}$$

where ϕ is the covariance matrix of the training data and λ is the variance predicted by the inversion. The mean of the distribution is assumed to be zero since it is unknown. The covariance is constructed using a radial basis function (RBF) kernel given by

$$\phi(\eta_m, \eta_n) = e^{-\frac{||\eta_m - \eta_n||^2}{h^2}}. \tag{27}$$

The L2 norm is used in the RBF. The decision to use an RBF kernel is not unique, it is possible that more expressive kernels may produce better results. The model output, $\beta(\eta_{test})$, is defined by a conditional probability distribution $p(\beta|\mathbf{Y}_{train})$. It can be shown that the probability distribution is given by

$$p(\eta|\mathbf{Y}_{train}) \approx \mathcal{N}\left(\underline{\phi}^T(\phi + \lambda \mathbf{I})^{-1}\mathbf{Y}_{train}, 1 + \lambda - \underline{\phi}^T(\phi + \lambda\mathbf{I})^{-1}\underline{\phi}\right). \tag{28}$$

It directly follows that the expected value and variance of β are given by

$$\beta(\eta) = \underline{\phi}^T\left(\phi + \lambda\mathbf{I}\right)^{-1}\mathbf{Y}_{train}, \tag{29}$$

$$\sigma_f^2 = 1 + \lambda - \underline{\phi}^T\left(\phi + \lambda\mathbf{I}\right)^{-1}\underline{\phi}, \tag{30}$$

where $\underline{\phi}$ is a vector whose elements are $\phi(\eta_{train,i}, \eta_{test})$. It is noted that in the current implementation the machine learning process only returns variances. The covariance of the training data assumed in Eq. (27) is a mathematical construct designed to help the machine learning process and has no direct relation to the true covariance of the training data.

The hyperparameters h in Eq. (27) are found by maximizing the probability of obtaining an output distribution $\mathbf{Y}_{train}$ given input features η and hyperparameters h. This is done by maximizing the log marginal likelihood function,

$$\log p(\mathbf{Y}_{train}|\eta_{train}, h) = -\frac{1}{2}\log|\phi + \lambda| - \frac{1}{2}\mathbf{Y}_{train}(\phi + \lambda)^{-1}\mathbf{Y}_{train} - \frac{N}{2}\log(2\pi), \tag{31}$$

where N is the number of training points.

4. Application to model problem

The framework is now applied to a scalar non-linear ordinary differential equation that resembles one-dimensional heat conduction with radiative and convective heat sources. The "true" model is taken to be

$$\frac{d^2T}{dz^2} = \varepsilon(T)\left(T_\infty^4 - T^4\right) + h(T_\infty - T), \quad z \in [0..1], \tag{32}$$

with homogeneous boundary conditions. We will refer to ε as the emissivity of the material and h as the convection coefficient. In the true process, the emissivity is a stochastic non-linear function of temperature and is given by

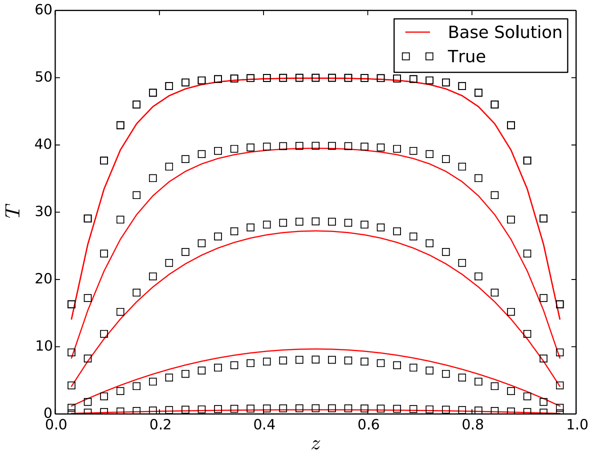


Fig. 1. Solutions of the base model compared to the mean of the true process.

Table 1
Summary of the conditions used for the inversion.

Case	T_∞	σ_{prior}	β_{prior}
1	5	20	1
2	10	2	1
3	15	1	1
4	20	1	1
5	25	0.5	1
6	30	1	1
7	35	1	1
8	40	1	1
9	45	1	1
10	50	0.8	1

$$\varepsilon(T) = \left[1 + 5\sin\left(\frac{3\pi}{200}T\right) + \exp(0.02T) + \mathcal{N}(0, 0.1^2)\right] \times 10^{-4}. \tag{33}$$

The convection coefficient is taken to be a constant of $h = 0.5$. To demonstrate the framework, we consider the case where the true process (Eqs. (32) and (33)) is unknown. The process is imperfectly modeled by

$$\frac{d^2T}{dz^2} = \varepsilon_0\left(T_\infty^4(z) - T^4\right) \tag{34}$$

with $\varepsilon_0 = 5 \times 10^{-4}$. Eq. (34) will be referred to as the base model. The resulting model outputs are shown in Fig. 1. The model particularly suffers when T_∞ is low, where the ignored linear term is significant. The inverse is posed by adding a spatial multiplier to ε_0

$$\frac{d^2T}{dz^2} = \beta(z)\varepsilon_0\left(T_\infty^4(z) - T^4\right). \tag{35}$$

The goal of the framework is to obtain $\beta(z)$ from the inversion, and then to learn $\beta = \beta(T, T_\infty)$. Note that β will encapsulate both the true form of ε and the convective heat transfer term. The true solution for β is

$$\beta(T, T_\infty) = \beta_r + \beta_c = \frac{1}{\varepsilon_0}\left[1 + 5\sin\left(\frac{3\pi}{200}T\right) + \exp(0.02T) + \mathcal{N}(0, 0.1)\right] \times 10^{-4} + \frac{h}{\varepsilon_0}\frac{T_\infty - T}{T_\infty^4 - T^4}. \tag{36}$$

Synthetic data is generated by solving 100 realizations of the true process (Eq. (32)) for $T_\infty \in [5, 10, \ldots, 50]$. The governing equation is solved using second order central differences on a uniform mesh with 31 grid points. These synthetic data are used as observational data for the inverse calculations. Note that the inversion is performed on the same computational grid. A summary of the conditions used for the inversion are given in Table 1.

The inversion is performed using the various models for the observational covariance matrix $\mathbf{C_m}$ that were previously discussed. For $\mathbf{C_m} = \sigma_{obs}^2\mathbf{I}$, the observational data is used to compute a single mean variance. For $\mathbf{C_m} = \vec{\sigma}_{obs}^2\mathbf{I}$, the observational data is used to compute the variance of temperature at each grid point. An uninformative prior is selected that corresponds with the baseline model, i.e. $\beta_{prior} = 1$. The prior variance is selected such that the 2σ limits of the prior PDF of temperature encompass the observed solution. The prior PDF for temperature is determined by solving the forward model for samples of β_{prior}. In this case, the forward model was sampled 100 times. Elementary statistical formulae can be used as a general guideline to determine the number of required samples. Given n statistically independent samples, the error on the mean $\sigma_{\overline{X}}$ and the error on the (co)variances σ_S can be approximated by

$$\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}, \qquad \sigma_S^2 = \sigma^2 \sqrt{\frac{2}{(n-1)}}. \tag{37}$$

For clarity, the entire solution process is outlined:

1. Sample the prior distribution of β via Equations (10) and (11) with the assumed prior covariance matrix $\mathbf{C}_\beta$.
   (a) For each sample of β solve the model equation (Equation (35)) to determine distributions for temperature.
   (b) Determine if $\mathbf{C}_\beta$ was appropriately chosen by ensuring that the observed temperature profile falls within the $\pm 2\sigma$ limits of the distribution predicted by the model equation.
2. Solve the inverse problem with the various models for $\mathbf{C_m}$ by solving the optimization problem.
3. Sample the posterior distribution of β_{map} via Equations (10) and (11) with $\mathbf{C}_{\beta_{map}}$.
   (a) For each sample of β, solve the model equation to determine the posterior distributions for temperature about the MAP point.

The results of the inversion for $T_\infty = 50$ are of the most interest and are shown in Fig. 2. It is first seen that the MAP solution for temperature coincides with the observed value for all models. The MAP solution for β, however, only coincides with the true solution when the complete observational covariance is used. For the diagonal models of $\mathbf{C_m}$ that ignore covariance, the posterior variance is too high in the center of the domain, as seen in Figs. 2a and 2b. However, the posterior variance is directly correlated to the local accuracy of β_{map}. When the complete observational covariance is used the correct posterior distribution is inferred across the entire domain.

Fig. 3 gives a compiled summary of the inferred β_{map} for the 10 cases. For plotting purposes, β_r and β_c are extracted from β_{map} and the uncertainty bounds are attached to β_r. **The missing and deficient terms in the model equation have been effectively inferred, especially in Fig. 3c.** For the lower-order representations of $\mathbf{C_m}$, β was inferred correctly over most of the domain; with error being present at low and high temperatures. This error is well reflected by the posterior variance. The solutions using the complete observational covariance are seen to yield extremely accurate inferences for β and σ over the entire domain.

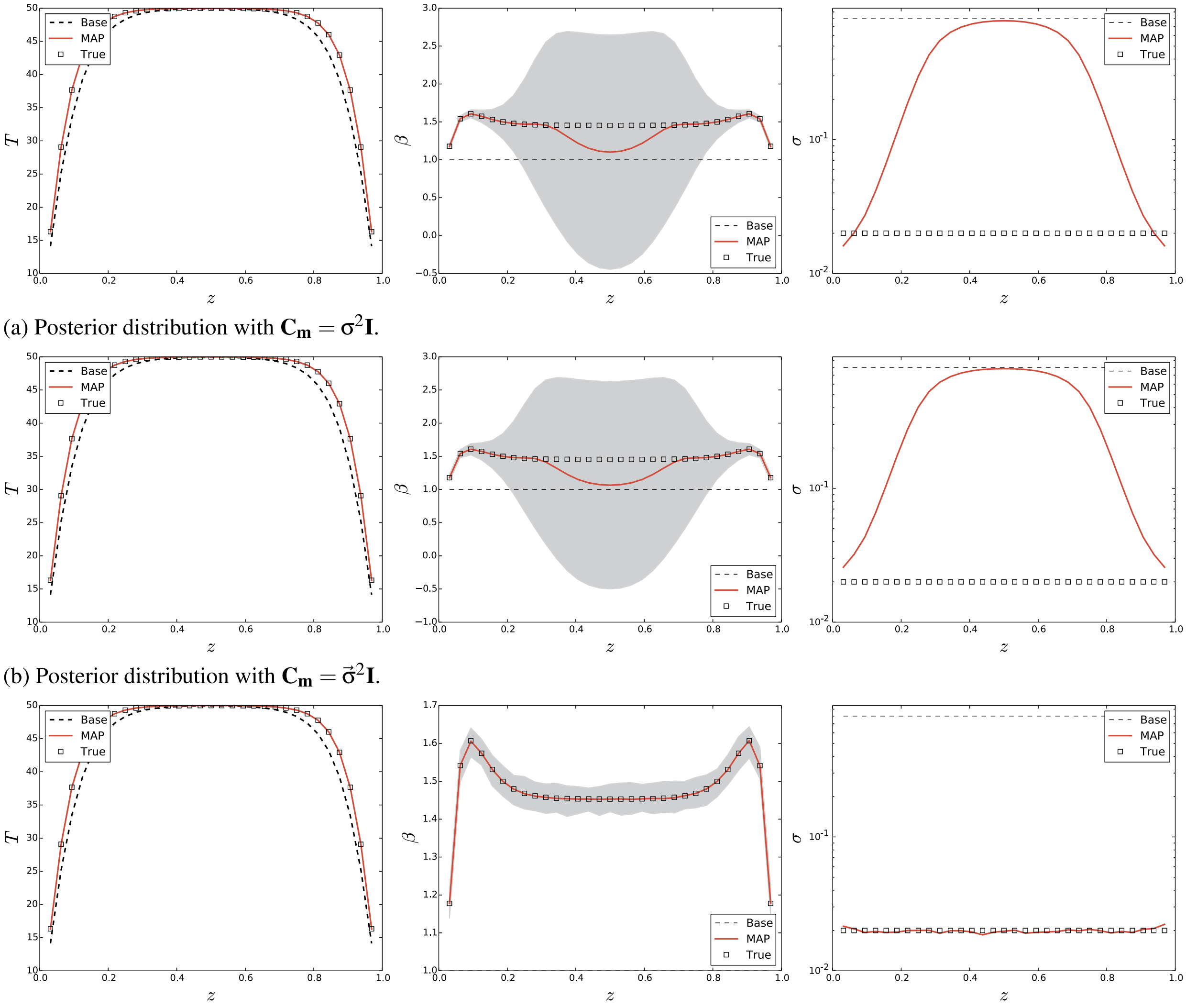
Several conclusions can be drawn from the inference step. First, the performance of the inversion was comparable for $\mathbf{C_m} = \sigma^2\mathbf{I}$ and $\mathbf{C_m} = \vec{\sigma}^2\mathbf{I}$. This shows that an accurate observational covariance is needed to correctly infer the posterior distribution of β. Second, if a simpler observational covariance is used, the posterior distribution can still provide information on the accuracy of the inference; but the posterior distribution is not representative of the underlying physics. Here we make an important note that the objective of this model problem was to infer the correction β and its proper posterior distribution. However, it is often desirable to infer a correction for a mean quantity (as in the next example). Under these settings, the covariance matrices should be constructed differently than in the procedure described above. For example, $\sigma_{obs}^{mean} = \sigma_{obs}/\sqrt{N_{samples}}$ is a more appropriate standard deviation for mean quantities. In general, an arbitrarily low variance can always be set for the observable, in which case the resulting MAP solutions will be in closer agreement with the observed data (the discrepancies between the MAP and true solutions in Fig. 2 can be eliminated with this method).

In this section it was shown that, unless the correct observational covariance was used, the statistics of the resulting posterior distribution can be inaccurate. It is well argued that, in the absence of the true observational covariance, the inversion could be carried out for low order statistics such as mean quantities.

4.1. Predictive simulations with machine learning injection

As described in Section 3, a machine learning algorithm utilizing GPs is used to elicit the functional relationship $\beta(T, T_\infty)$ from the spatial data generated in the inversion. The data generated from the inversion using the exact observational covariance is used for training. It is noted that the ML formulation employed only makes use of the variance predicted by the inverse, rather than the entire covariance matrix. The hyperparameters for the GP are optimized off-line, and then the resulting model is injected into the solver at every iteration of the solution. Systematically, the solver calls the residual calculation routine which in turn calls the ML algorithm. The ML algorithm is queried with T and T_∞ and returns β_{ML} and σ_{ML}.

The machine-learned predictive model was evaluated for a variety of new predictive cases. Table 2 gives a summary of the conditions and the performance of the ML model. Case 1 provides examples with similar “physics” as the training cases (i.e. $T_\infty = \text{constant}$), and the results of the implementation are excellent. Cases 2 and 3 explore the performance of the model for regions of $T_\infty > 50$, for which there were no training data. **For these cases, an improvement is observed over the baseline model**. Cases 4 and 5 explore the performance of the ML model for lower values of T_∞, where the linear heat transfer term becomes important. Again, it is seen that the performance of the model is much improved. The solutions for Case 2 and Case 5 with the predictive model are given in Figs. 4 and 5. In both cases, the ML solution for temperature and β is much improved from the base model, but error is still present. However, excellent correlation is seen between model



(a) Posterior distribution with $\mathbf{C_m} = \sigma^2\mathbf{I}$.

(b) Posterior distribution with $\mathbf{C_m} = \vec{\sigma}^2\mathbf{I}$.

(c) Posterior distribution with the full observational covariance $\mathbf{C_m}$.

Fig. 2. Posterior model distributions at $T_\infty = 50$ for each model of $\mathbf{C_m}$. From left to right T and β with 2σ limits, σ, and $\mathbf{C}_{\beta_{map}}$ are shown, respectively.

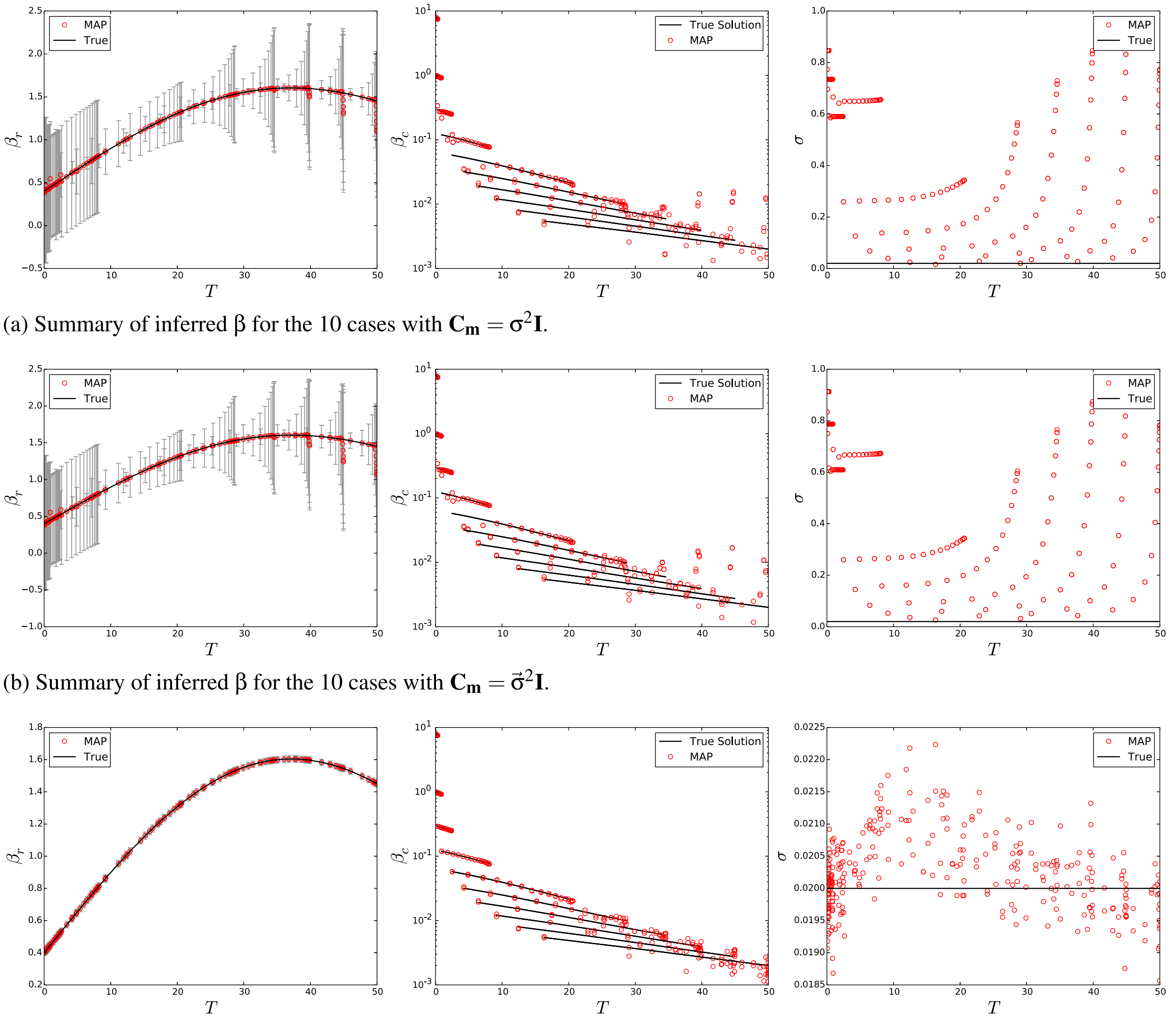
Table 2
Summary of cases used to test the predictive model. The L2 norm is used to compute the errors reported in column 5.

Case	Grid points	T_∞	Training data	$\lvert e_{map} - e_{model}\rvert / e_{model} \times 100$
1	71	28	$\mathbf{R} = \sigma^2\mathbf{I}$	95.5%
2	71	55	$\mathbf{R} = \sigma^2\mathbf{I}$	64.5%
3	71	$35 + 20\sin(2\pi z)$	$\mathbf{R} = \sigma^2\mathbf{I}$	76.0%
4	71	$35 - 15z$	$\mathbf{R} = \sigma^2\mathbf{I}$	98.3%
5	71	$15 + 5\cos(\pi z)$	$\mathbf{R} = \sigma^2\mathbf{I}$	88.5%

error and the predicted variance. Similar correlations were seen for all cases, suggesting that the posterior variance is an indicator of local model accuracy. This feature is extremely useful in a predictive setting.

5. Application to turbulent channel flow

The modeling of turbulent flows has been a long-standing obstacle to the application of computational fluid dynamics (CFD) to many practical problems. Direct numerical simulations (DNS) attempt to resolve all scales of turbulence but the resolution requirements make this technique infeasible for most flows of engineering interest. To compute practical high Reynolds number flows, near-wall modeling is performed using a Reynolds-averaged Navier–Stokes (RANS)-type closure. RANS-based methods are typically formulated using a combination of theory and intuition. Traditionally, a number of free



(a) Summary of inferred β for the 10 cases with $\mathbf{C_m} = \sigma^2\mathbf{I}$.

(b) Summary of inferred β for the 10 cases with $\mathbf{C_m} = \vec{\sigma}^2\mathbf{I}$.

(c) Summary of inferred β for the 10 cases with the full observational covariance.

Fig. 3. Summary of Posterior model distributions for each model of $\mathbf{C_m}$. From left to right, β_r, β_c, and σ are shown.

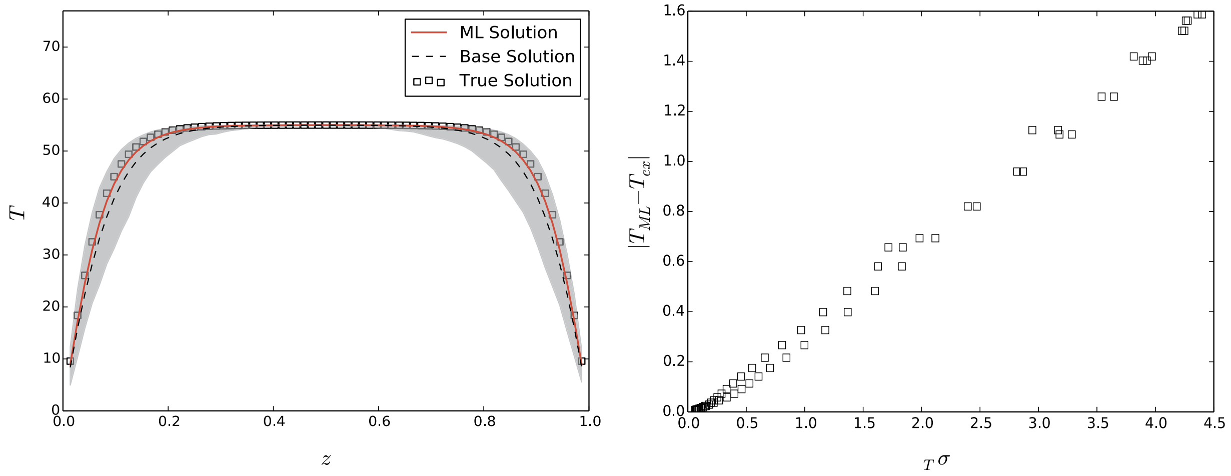
parameters remain in the model and these are calibrated using empirical fitting and are often found to be deficient in many flows. The key issue is that the main source of error is in the functional form of the model terms. Functional relationships elicited directly from high-fidelity simulation or experimental data will not translate to RANS model improvements since the inference has to be within the context of the model. The technique outlined in this section infers and reconstructs a correction that is *consistent with the low fidelity (RANS) model.*

The FIML framework is applied to turbulent channel flow with a $k - \omega$ turbulence model [25]. The Reynolds-averaged momentum equation for incompressible fully-developed channel flow is given by

$$\frac{\partial}{\partial y}\left[\mu\frac{\partial \overline{u}}{\partial y} - \rho\overline{u'v'}\right] - \frac{\partial \overline{p}}{\partial x} = 0, \tag{38}$$

where $\overline{p}$ and $\overline{u}$ are the mean pressure and velocity, respectively. The process of Reynolds averaging introduces the unclosed Reynolds stresses, $\tau_{ij} = -\rho\overline{u'v'}$. Determining τ_{ij} is the fundamental challenge of turbulence modeling. The $k - \omega$ model makes use of the Boussinesq approximation, where the Reynolds-stress tensor is assumed to take the form

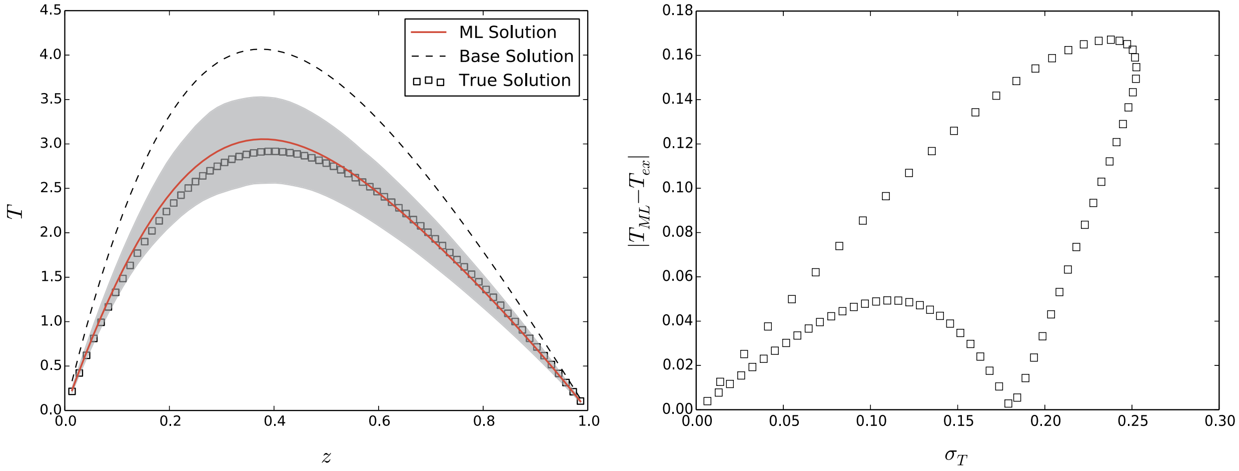
$$\tau_{ij} = 2\nu_t S_{ij} - \frac{2}{3}k\delta_{ij}, \tag{39}$$



(a) Machine learned solution for temperature.

(b) Predicted temperature error as a function of variance

Fig. 4. Results of the predictive model for Case 2.



(a) Machine learned solution for temperature.

(b) Predicted temperature error as a function of variance

Fig. 5. Results of the predictive model for Case 5.

with ν_t being the turbulent eddy viscosity, k the turbulent kinetic energy, and S_{ij} the mean strain-rate tensor. The still unclosed turbulent eddy viscosity is then determined by introducing transport equations for the turbulent kinetic energy k and the specific dissipation rate ω. On dimensional grounds, the turbulent eddy viscosity is modeled by

$$\nu_t = C_\mu \frac{k}{\omega}, \tag{40}$$

where C_μ is a constant of proportionality. For the case of planar channel flow, the transport equations for k and ω become ordinary differential equations of the form

$$\nu_t \left(\frac{\partial \bar{u}}{\partial y}\right)^2 - \alpha^* k\omega + \frac{\partial}{\partial y}\left[\left(\nu + \sigma^* \frac{k}{\omega}\right)\frac{\partial k}{\partial y}\right] = 0 \tag{41}$$

$$\gamma \left(\frac{\partial \bar{u}}{\partial y}\right)^2 - \alpha\omega^2 + \frac{\partial}{\partial y}\left[\left(\nu + \sigma \frac{k}{\omega}\right)\frac{\partial \omega}{\partial y}\right] = 0. \tag{42}$$

The standard closure coefficients for the Wilcox $k - \omega$ model are used and are given in Table 3, along with the associated boundary conditions for the channel flow. Equations (38) through (42) will be referred to as the base model. Numerically, these governing equations are discretized with second-order finite differences and the system is solved by introducing pseudo-time derivatives to the left hand side of Equations (38), (41), and (42). Implicit time integration is then used to iterate the system to a steady state. The equations are solved on a geometrically graded mesh with the first grid point

Table 3
Summary of model coefficients and boundary conditions for the $k-\omega$ model in planar channel flow. The channel wall is at $y=0$ and the mid plane of the channel is at $y=h/2$.

C_μ	α^*	σ^*	γ	α	σ	$y=0$	$y=h/2$
1.00	0.09	0.6	13/25	0.09	0.5	$u,k=0;\ \omega=\omega_w$	$\partial/\partial y(u,k,\omega)=0$

placed well into the viscous sublayer at $y^+\approx 0.05$. At the wall, the boundary condition for ω becomes singular, which is numerically handled by analyzing asymptotics [25] of ω.

5.1. Inversion

The functional correction $\beta(y)$ is introduced as a multiplier to the production term in the turbulent kinetic energy equation,

$$\nu_t\left(\frac{\partial\overline{u}}{\partial y}\right)^2\beta(y)-\alpha^*k\omega+\frac{\partial}{\partial y}\left[\left(\nu+\sigma^*\frac{k}{\omega}\right)\frac{\partial k}{\partial y}\right]=0. \tag{43}$$

Introducing β to the production term modifies the entire turbulence model, and is equivalent to adding an additive source term. DNS data from Jimenez et al. [26] are used in the inverse modeling. Of this data, the velocity profiles are targeted; i.e. $\mathbf{d}=\mathbf{u}_{DNS}$. Since the DNS data provides a near perfect observation of truth, the observational covariance is taken to be $\mathbf{C_m}=\sigma^2_{obs}\mathbf{I}$ where $\sigma_{obs}=10^{-10}$. This choice neglects covariance in the observed data. Constructing a more accurate observational covariance, as was done in the previous problem, requires statistics that are not readily available from DNS. In this case, the two-point correlation of the mean streamwise velocity in the wall normal direction could be used to build an accurate observational covariance. However, a plausible alternative could be to use an approximation to the two-point correlation to build a more accurate observational covariance.

The prior distribution is determined by the same process discussed previously, where $\sigma_p=0.5$ was selected such that the DNS velocity profile falls within the 2σ limits of the prior model. With $\sigma_{obs}=10^{-10}$ and $\sigma_p=0.5$, the dependence on the prior has been effectively eliminated for regions where the inferred function is sensitive to the data.

The inversion is performed for different wall-shear stress-based Reynolds numbers $Re_\tau\in[180,550,950,2000,4200]$. An example of the resulting posterior distribution for inferred velocity and β is given in Fig. 6. The MAP solution is seen to match the DNS data very well. Due to the low observational variance, the posterior distribution for velocity collapses on the MAP solution. The posterior distribution for β also collapses on the MAP estimate for $y^+>5$. Turbulent production within the viscous sublayer ($y^+<5$) is very small and thus the value of β in this region is largely inconsequential and cannot be inferred with a high degree of confidence.

A summary of the inferred corrections for all Reynolds numbers is given in Fig. 7. A universal scaling is seen with y^+ within the inner layer, and with y near the center of the channel, both results being consistent with the underlying physics. A Reynolds number dependence that is usually missed in traditional turbulence models is additionally observed. A detailed summary of the inversion is provided in [27].

5.2. Predictive simulation with machine learning injection

Gaussian processes are again used to extract the functional relationship $\beta(\eta)$. The non-dimensional input features η considered in this process are the inverse solutions for $\{Sk/\varepsilon, d\sqrt{k}/\nu, P/\varepsilon, y^+\}$ at $Re_\tau\in[180,550,950,4200]$; $Re_\tau=2000$ was omitted from this training data set. In the training process, only input features within the inner layer were considered. It is well recognized that a functional relationship $\beta(\eta)$ between the model correction and input features may not exist; or minimally the accuracy of the functional extraction may vary across the solution space. Injecting the ML algorithm into the solver may not be sufficient for cases where local ML predictions are highly inaccurate. One method to make an appropriate model update is to consider an additional Bayesian update step after the machine learning has completed. This final Bayesian update step (with the appropriate assumptions) is given by

$$\beta_{post}=arg\ \min\frac{1}{2}\left[\left(\beta-\beta_{ML}\right)^T\mathbf{C}_{\beta ML}^{-1}\left(\beta-\beta_{ML}\right)+\left(\beta-\beta_{prior}\right)^T\mathbf{C}_{\beta_{prior}}{}^{-1}\left(\beta-\beta_{prior}\right)\right]. \tag{44}$$

Note that $C_{\beta_{ML}}$ and β_{ML} are functions of both the inverse and the machine learning algorithms, while β_{prior} is specified in the inverse. The posterior model will assimilate to the ML model for regions where the variance of the ML model is low. For regions of high variance, the posterior model will assimilate to the prior model. Although not shown, the results using this method for the previous model problem were comparable to those reported in Table 2.

The posterior model described above is now tested at $Re_\tau=2000$. The ML model is queried once at the inverse solution to construct a machine learned correction. This correction is applied during the predictive solution to obtain the final posterior model. The impact of the final Bayesian update is seen in Fig. 8. The ML prediction performs well within the inner layer, as such a low training variance is predicted and the posterior model assimilates to the ML model. In the channel core,

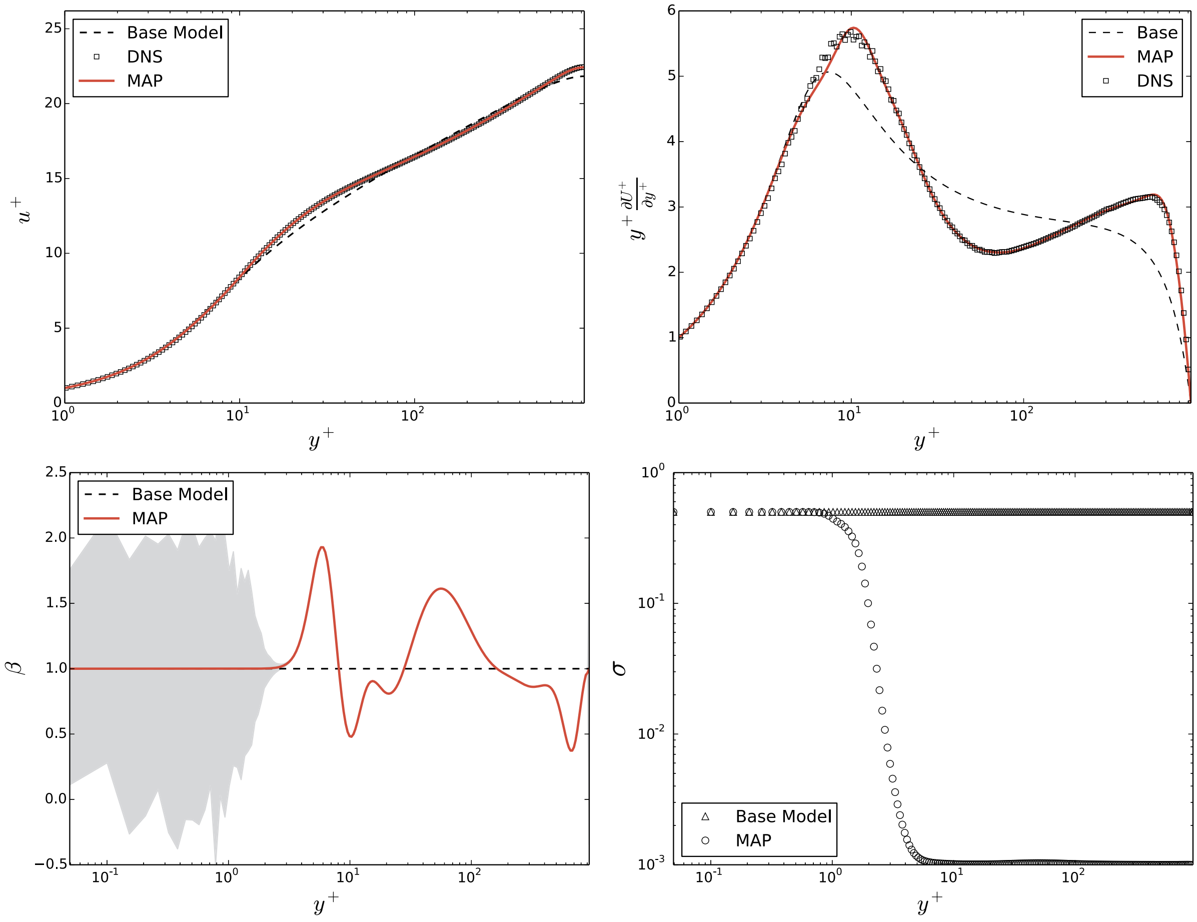


Fig. 6. Posterior model distributions for planar channel flow at $Re_\tau = 934$. The $\pm 2\sigma$ limits are shaded in both figures.

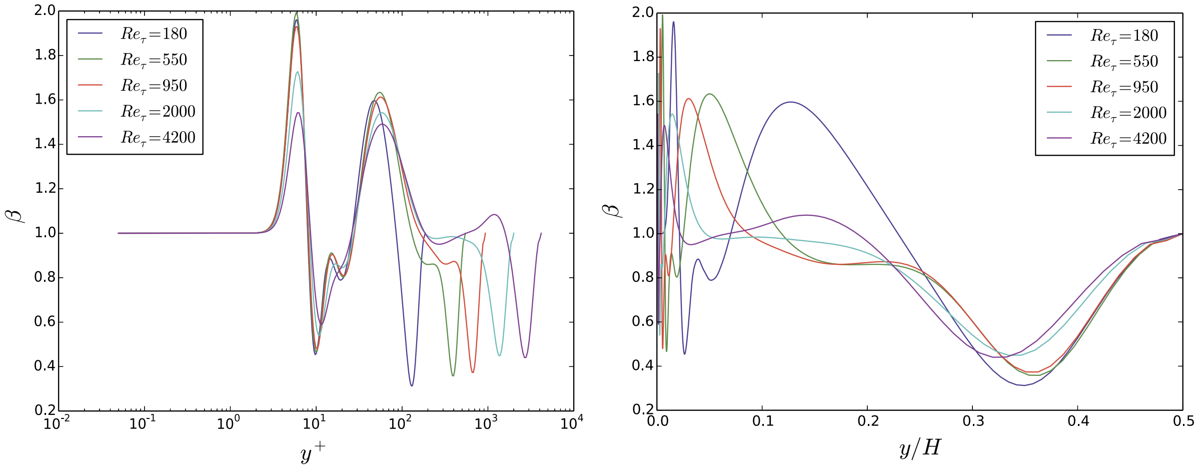


Fig. 7. Summary of inferred β for $Re_\tau \in [186, 547, 934, 2004, 4200]$.

the ML model is unable to extract an accurate functional relationship. This lack of accuracy is well reflected by the high ML variance in this solution region, as the posterior model assimilates to the prior model.

Fig. 9 shows the resulting velocity predictions and the associated 95% confidence intervals. Two features worth noting are the improved performance within the inner layer and the correlation between the confidence intervals and model error. The MAP solution within the inner layer is much improved, in particular the slight bump that characterizes the buffer layer is well captured in the posterior model. In the outer layer, however, the mean velocity is under-predicted. In this region,

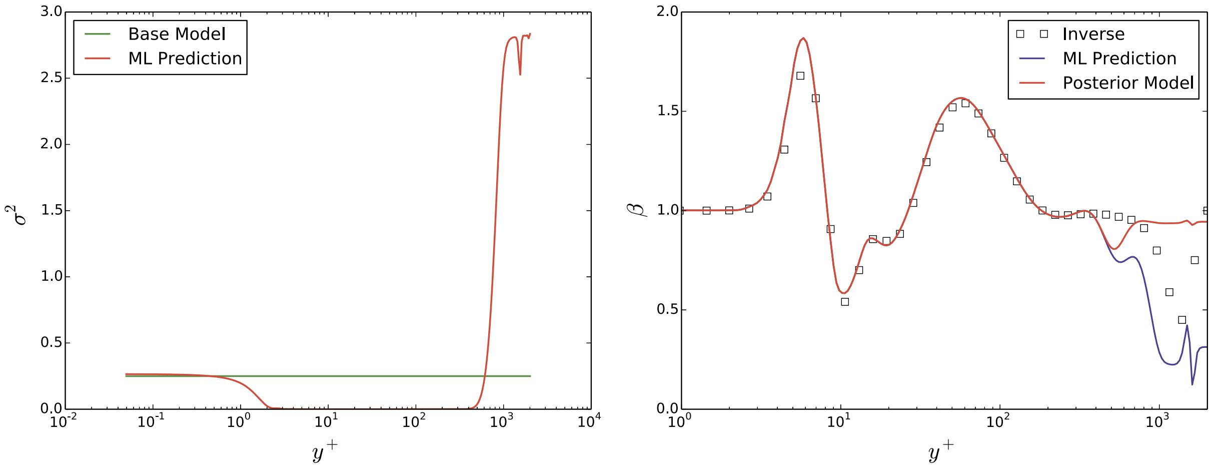


Fig. 8. Posterior model predictions for the correction β at $Re_\tau = 2000$.

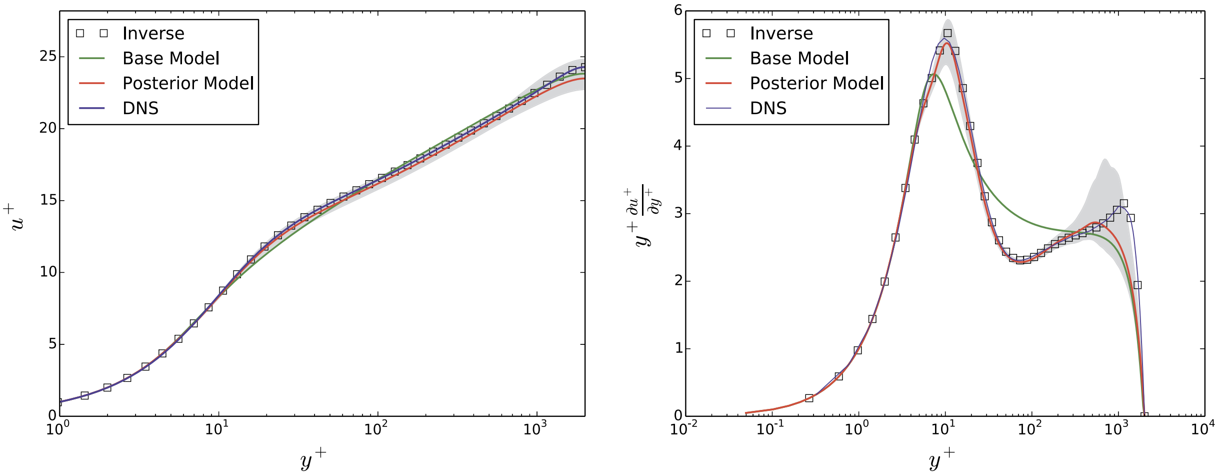


Fig. 9. Posterior model predictions for velocity at $Re_\tau = 2000$.

a high ML variance was predicted and the posterior model reverts to the prior model. The failure to predict the increased destruction of TKE in the channel core leads to too high of an eddy viscosity and an under-prediction of velocity. It is worth noting that the turbulent eddy viscosity predicted by the baseline $k - \omega$ model is too high in the channel core. When the correct behavior within the inner layer is captured and the baseline model is used in the channel core, an under-prediction in velocity is expected. The confidence intervals given by the posterior model again provide a reasonable estimate of the underlying uncertainty of the model. While the resulting PDFs should not be viewed as exact, they provide information about local model accuracy, which is of paramount value to the practitioner.

6. Conclusions and perspectives

The wealth of available data from high-fidelity simulations and high resolution experiments provides unprecedented opportunities to more comprehensively inform closure models. In this work, a data-driven modeling approach, which we refer to as FIML (field inversion and machine learning) was presented. The proposed approach moves beyond parameter calibration and uses data to directly infer information about the functional form of model discrepancies. The inference process generates function correction information for *specific* problems. Once the inference is applied over a number of problems, machine learning is used to reconstruct the inferred function in terms of variables that will be available during predictive simulations using lower fidelity models. This step aims to create *generic* modeling knowledge from the inferred information. The reconstructed function is then embedded into a predictive solution process. In contrast to existing calibration frameworks, our approach uses data to directly infer information about underlying model discrepancies and provides a methodology to generalize the inferred information. This approach provides insight into model error at a fundamental level, rather than at the level of the output.

The framework was applied to a scalar non-linear ODE model problem, in which missing and deficient terms were reconstructed and the predictive capability of the improved model was confirmed. A second application was extended to turbulent channel flow, where DNS data was used to inform a standard Reynolds-averaged closure model. While it was shown that precise observational statistics may be needed to precisely quantify the posterior distribution, simple approximations for the prior statistics and linearized Gaussian assumptions for the posterior proved to be sufficient to obtain mean solutions and posterior distributions that are *representative of the modeling error*.

The field inversion process directly provides comprehensive information about model discrepancies, which is of great use to the modeler in the quest to formulate more accurate closures. The machine learning step could be considered as *one* tool that can be used to reconstruct the discrepancy in terms of low fidelity model information. It was demonstrated that, for the simple problems considered, it is possible to use machine learning methods to elicit functional relationships and the associated uncertainties for the corrections obtained in the inference process. This extraction allows for predictive modeling.

The examples in this paper are illustrative in nature. For the framework to be able to offer improved predictions in practical situations, inverse problems must be solved over a wide class of problems (and over multiple objective functions of interest) that will be representative of the deficient physics in the baseline model. Concurrently, the tendency of the learning process to over-fit data must also be avoided. At every stage of the process, the underlying physical insight is irreplaceable and thus it is left to the modeler to make judicious choices about the data, prior information and introduction of one or more correction functions. Further, physical considerations such as realizability and consistency with asymptotic limits should be enforced.

A number of challenges remain for a full-scale implementation in complex problems. These include grid/numerical scheme dependence of the inferred corrections, solver convergence, scalability, learning errors, accounting for non-Gaussian behavior, etc. The present work has nevertheless demonstrated that the FIML method can play a significant role in using data to more comprehensively inform predictive models, offering a route to creating improved closure approximations while providing measures of model-form uncertainties.

Acknowledgements

This work was supported by NASA LEARN project NNX15AN98A and by the NSF via grant 1507928.

Appendix A. Markov chain Monte Carlo sampling

The inverse procedure outlined in this paper makes strong assumptions about the Gaussian nature of the underlying PDFs. In our approach, it is assumed that distributions for the prior probability $p(\beta)$, observational data $\mathbf{d}$, likelihood $h(\mathbf{d}|\beta)$, and conditional probability $q(\beta|\mathbf{d})$ are all Gaussian. In the linear case, if $\mathbf{d}$ and $p(\beta)$ are Gaussian, the posterior distribution will be Gaussian. This need not be true in the non-linear case. To accurately infer the posterior PDF in the non-linear case, more expensive methods (such as sampling) need to be utilized. Since the non-linear heat problem presented in Section 4 is relatively simple, Markov Chain Monte Carlo (MCMC) simulations were performed. Sampling was performed with the Python package PyMC [28], which utilizes the Metropolis–Hastings step, to determine the posterior distribution. Fig. 10 shows the posterior distribution determined by MCMC sampling for $T_\infty = 50$ with the complete observational covariance matrix (Eq. (25)) and compares it to the MAP solution obtained through Bayesian inversion. It is seen that the comparison between the two methods is excellent. The MAP solution coincides almost perfectly with the mean MCMC solution. Additionally, the posterior PDFs are Gaussian across the entire domain and compare well with the MAP solution. While it is not prudent to generalize these results to other non-linear problems, the Gaussian assumption appears to be reasonable in this specific problem.

Appendix B. Computational cost

The primary computational cost in the FIML framework arises from functional inversion and construction of the machine-learned model, both of which are off-line processes. The inverse problem requires the solution of a high-dimensional optimization problem, which could be simplified by using a parametric representation of the function β. The number of iterations required by optimization algorithms varies from problem to problem, but the computational cost is $O(100)$ solves of the forward model. In the presented work, the forward model is additionally solved for realizations of β. This sampling process is performed for both the prior and posterior distributions. The number of samples required depends on the desired accuracy of statistical quantities, but $O(100)$ samples serves as a representative estimate. Additionally, the sampling process is embarrassingly parallel. Note that the number of forward solves required for sampling and optimization (assuming gradient-based methods) does not scale with dimensionality. Constructing the Gaussian Process ML model requires an $N \times N$ inversion (with N being the number of training points). Although this is an off-line process, it can become prohibitively expensive for very large training sets, demanding sparse and approximate solvers. Additionally, determining the GP hyperparameters requires the matrix inversion at each iteration of the optimization algorithm. For high-dimensional learning problems, more efficient ML algorithms (such as neural networks or approximate GPs [23]) need to be considered.

The on-line cost of the FIML framework is realized in the evaluation of the ML model. In the case of GPs, this evaluation requires a matrix-vector multiplication. In practice, the introduction of the ML model may have an impact on solver convergence and stability, both of which could affect the computational cost.

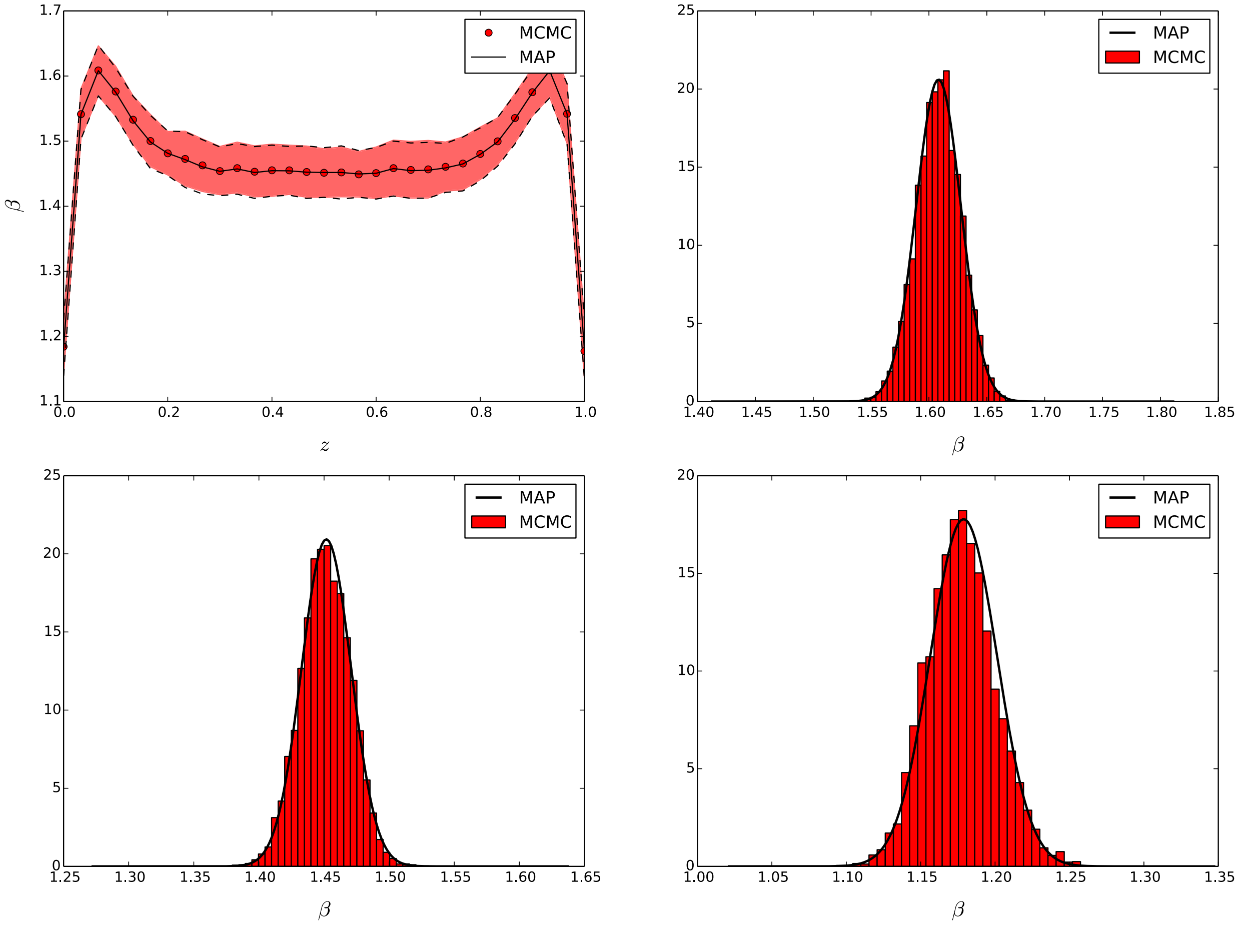


Fig. 10. The posterior distribution of β as obtained through MCMC is compared to the MAP solution. The upper left figure compares the mean solution obtained with MCMC and the 95% confidence intervals. The upper right figure shows the PDF for β at $z = 0.1$. The lower left and right figures show the PDF for β at $z = 0.5$ and $z = 0.96$ respectively.

References

[1] I.M. Navon, Data assimilation for numerical weather prediction: a review, in: Data Assimilation for Atmospheric, Oceanic, and Hydrologic Applications, Springer, 2009.

[2] G. Kerschen, K. Worden, A. Vakakis, J. Golinval, Past, present, and future of system identification in structural dynamics, Mech. Syst. Signal Process. 20 (2006) 505–592.

[3] N.R. Council, Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification, National Academies Press, 2012.

[4] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc., Ser. B, Stat. Methodol. 63 (3) (2001) 425–464.

[5] M.C. Kennedy, C.W. Anderson, S. Conti, A. O'Hagan, Case studies in Gaussian process modelling of computer codes, Reliab. Eng. Syst. Saf. 91 (10) (2006) 1301–1309.

[6] S. Conti, J.P. Gosling, J.E. Oakley, A. O'hagan, Gaussian process emulation of dynamic computer codes, Biometrika 96 (3) (2009) 663–676.

[7] J. Brynjarsdóttir, A. O'Hagan, Learning about physical parameters: the importance of model discrepancy, Inverse Probl. 30 (11) (2014) 114007.

[8] P.D. Arendt, D.W. Apley, W. Chen, Quantification of model uncertainty: calibration, model discrepancy, and identifiability, J. Mech. Des. 134 (10) (2012) 100908.

[9] R.C. Smith, Uncertainty Quantification: Theory, Implementation, and Applications, Computational Science and Engineering, vol. 12, SIAM, 2013.

[10] S.H. Cheung, T.A. Oliver, E.E. Prudencio, S. Prudhomme, R.D. Moser, Bayesian uncertainty analysis with applications to turbulence modeling, Reliab. Eng. Syst. Saf. 96 (9) (2011) 1137–1149.

[11] W.N. Edeling, P. Cinnella, R.P. Dwight, Bayesian estimates of parameter variability in the $k - \varepsilon$ turbulence model, 2014.

[12] J.L. Beck, L.S. Katafygiotis, Updating models and their uncertainties. I: Bayesian statistical framework, J. Eng. Mech. 124 (4) (1998) 455–461.

[13] L.M. Berliner, K. Jezek, N. Cressie, Y. Kim, C. Lam, C.V.D. Veen, Modeling dynamic controls on ice streams: a Bayesian statistical approach, J. Glaciol. 54 (2008) 705–714.

[14] K. Sargsyan, H. Najm, R. Ghanem, On the statistical calibration of physical models, Int. J. Chem. Kinet. 47 (4) (April 2015) 246–276.

[15] J.L. Loeppky, D. Bingham, W.J. Welch, Computer model calibration or tuning in practice, Technical report, University of British, Columbia, 2006.

[16] C. Soize, Stochastic modeling of uncertainties in computational structural dynamics recent theoretical advances, J. Sound Vib. 332 (10) (2013) 2379–2395.

[17] M.L. Mehta, Random Matrices, Pure and Applied Mathematics, vol. 142, Academic Press, 2004.

[18] R. Aster, Parameter Estimation and Inverse Problems, Elsevier Academic Press, 2005.

[19] M.B. Giles, M.C. Duta, J.-D. Müller, N.A. Pierce, Algorithm developments for discrete adjoint methods, AIAA J. 41 (2) (2003) 198–205.

[20] J.E. Dennis Jr., J.J. Moré, Quasi-Newton methods, motivation and theory, SIAM Rev. 19 (1) (1977) 46–89.

[21] P. Caplan, Numerical computation of second derivatives with applications to optimization problems, Unpublished academic report, MIT.

[22] B.D. Tracey, K. Duraisamy, J.J. Alonso, A machine learning strategy to assist turbulence model development, in: 53rd AIAA Aerospace Sciences Meeting, The American Institute of Aeronautics and Astronautics, 2015.

[23] Z.J. Zhang, K. Duraisamy, Machine learning methods for data-driven turbulence modeling, in: AIAA Aviation and Aeronautics Forum and Exposition, Dallas, Texas, June 2015.

[24] C.E. Rasmussen, Gaussian processes for machine learning, 2006.

[25] D.C. Wilcox, Turbulence Modeling for CFD, vol. 2, DCW Industries, La Canada, CA, 1998.

[26] J. Jimenez, S. Hoyas, Turbulent fluctuations above the buffer layer of wall-bounded flows, J. Fluid Mech. 611 (2008) 215–236.

[27] E. Parish, K. Duraisamy, Quantification of turbulence modeling uncertainties using full field inversion, in: AIAA Aviation and Aeronautics Forum and Exposition, Dallas, Texas, June 2015.

[28] A. Patil, D. Huard, C. Fonnesbeck, PyMC: Bayesian stochastic modelling in Python, J. Stat. Softw. 35 (2010) 4.