PHILOSOPHICAL TRANSACTIONS A

royalsocietypublishing.org/journal/rsta

Research

Cite this article: Goldstein M, Vernon I, Cumming JA. 2025 Systematic structural discrepancy assessment for computer models. *Phil. Trans. R. Soc. A* **383**: 20240214. https://doi.org/10.1098/rsta.2024.0214

Received: 13 September 2024
Accepted: 6 February 2025

One contribution of 13 to a theme issue 'Uncertainty quantification for healthcare and biological systems (Part 2)'.

Subject Areas:
statistics, computer modelling and simulation

Keywords:
emulation, computer models, discrepancy, model inadequacy

Author for correspondence:
Ian Vernon
e-mail: i.r.vernon@durham.ac.uk

THE ROYAL SOCIETY
PUBLISHING

Systematic structural discrepancy assessment for computer models

Michael Goldstein, Ian Vernon and Jonathan A. Cumming

Department of Mathematical Sciences, Durham University, Durham, UK

IV, 0000-0002-9161-9946; JAC, 0000-0002-4855-7460

Model or structural discrepancy is an essential component in the analysis of computer simulators, representing the differences between the outputs of the simulator and the real-world system that the simulator seeks to represent. This discrepancy can arise from various sources such as simplifications of the model science in the simulator, choices made in our particular implementation of that science, and epistemic uncertainties such as the absence of features or science that we did not know to include or have yet to discover. In this paper, we define and distinguish two types of discrepancy: internal discrepancy that can be assessed by experiments on the simulator itself; and external discrepancy which lies outside the scope of such experiments. We present a tractable methodology and workflow for the assessment of structural discrepancy on the basis of collections of experiments applied to the computer model and illustrate our approach in the context of a simple biological model.

This article is part of the theme issue 'Uncertainty quantification for healthcare and biological systems (Part 2)'.

1. Introduction

Suppose that we have a simulator for a physical system, that we have observations of the real-world system against which to compare the simulator and that we wish to make statements about the real world using this information. No matter how complex a simulation model of a physical process is, there will always be differences between the outputs of the simulator and the real process that the model is intended to represent. Inevitably, there will be simplifications in the model science based on features that are too complicated for us to include, features that we do not know that we should include, mismatches between the scales on which the model and the system operate, simplifications and approximations in solving the equations determining the system, and so on.

If we are interested in making statements about the real system using results from the simulator, we must incorporate these differences into our analysis. Failure to do so leads to grossly inaccurate inferences, such as overfitting of the model to historical data, wrongly ruling out potentially useful models and overconfidence in subsequent predictions [1].

Informally, model or structural discrepancy is the difference between reality and appropriately chosen simulator output. As we will be uncertain about this difference, it is natural to express our knowledge probabilistically, to be incorporated, for example, within a Bayesian analysis. This article is concerned with systematic methods to quantify our knowledge about structural discrepancy in a form that we can use to make inferences about the real physical process. In particular, we will emphasize the value of separation of structural discrepancy into internal and external components, corresponding to features which may be assessed by computer experiments and features which lie outside such experiments, and explain the role of emulation for integrating each aspect of the discrepancy assessment within a manageable workflow.

2. Specifying model discrepancy

Suppose that we have a model $M(\cdot)$ for a physical system. The model takes as inputs a vector $\underline{x}$ related to system properties and outputs a vector $M(\underline{x})$ representing some features $\underline{y}$ of the behaviour of the physical system. The model is implemented as a computer simulator $\underline{f}(\underline{x})$. Often, we have historical observations, z, made, with error, on a subset, $\underline{y}_h$, of the elements of y, with corresponding functional outputs $f_h(\underline{x})$.

Sometimes, we may consider that there is a unique 'true' or 'best' choice, $\underline{x}^*$, for the input vector $\underline{x}$, and we use the observed value of $\underline{z}$ to make an inference for this value, for example assessing a Bayesian posterior distribution for $\underline{x}^*$. This process is termed *calibration* [2]. At other times, we do not consider that there is such a unique true value, for example if the inputs are tuning parameters. In such cases, we may wish to identify the collection of all input choices x for which the simulator is able to reproduce the observed system history. By this, we mean that the outputs $f_h(\underline{x})$ are acceptably close to the observed values of $\underline{z}$, when we have taken into account all of the uncertainties relevant for assessing the quality of the match. This process is termed *history matching* [3,4].

To carry out either of these procedures, we need a probabilistic representation of the difference between the simulator and the physical system, which we can evaluate for each choice of $\underline{x}$. A very common way to introduce such model discrepancy is by supposing that, if $\underline{x}^*$ is an appropriate choice for $\underline{x}$, then the state of the world, $\underline{y}$, is given by

$$\underline{y} = f(\underline{x}^*) + \epsilon_d, \tag{2.1}$$

where ϵ_d is a random 'error' vector capturing the deficiencies and missing features of the model, and which is independent of everything else. This choice has the great virtue of simplicity—we just add, say, a 10% error to all uncertainty statements about the real world produced by the simulator.

This is far better than ignoring discrepancy altogether, but we usually can, and should, be more careful. For example, suppose that the main reason for structural discrepancy is that the implementation of the model as a simulator involves certain approximations for the solutions of the underlying system equations. Suppose further that for some input choices these approximations have negligible effects whereas for other choices they have substantial effects. Then acceptable history matches for the former choices would require a closer fit to the data than matches for the latter choices, and similarly the likelihood for a Bayesian calibration would need to include an additional component of uncertainty for those input choices with larger model implementation errors.

The difference between a model and the corresponding physical system is a complex structure which has many different possible representations. We will suggest general forms which are sufficient to support many different aspects of structural discrepancy. Much of our description relies on the construction of emulators for elements of structural discrepancy. Emulators are a familiar feature of uncertainty quantification problems. Often, in practice, we are only able to evaluate f for a limited number of values of $\underline{x}$, because of time and resource constraints. In such cases, we usually construct emulators for the elements of f itself.

An *emulator* for a function is a fast surrogate model for the function, with known uncertainty of approximation, which allows us to carry out detailed exploration of model behaviour [2,3]. A common form for such an emulator, for an individual component $f_i(\underline{x})$ of f, which will be sufficient for our discussion, is to represent the function as

$$f_i(\underline{x}) = \sum_j a_{ij} g_{ij}(\underline{x}) + u_i(\underline{x}), \tag{2.2}$$

where a_{ij} are constants to determine, $g_{ij}(\underline{x})$ are deterministic functions, for example polynomials, of $\underline{x}$, and $u_i(\underline{x})$ is a stationary stochastic process, often with $\mathbb{E}\left[u_i(\underline{x})\right] = 0, \forall \underline{x}$ and a correlation function reflecting the smoothness of the function. For example, a common choice is

$$\mathbb{C}\mathrm{orr}\left[u_i(\underline{x}_1), u_i(\underline{x}_2)\right] = e^{-(\underline{x}_1-\underline{x}_2)^T}\Sigma\,(\underline{x}_1-\underline{x}_2),$$

where Σ is an appropriate scaling matrix. We may specify a complete probabilistic form for $u_i(\underline{x})$, for example a Gaussian process, for a standard Bayesian calibration analysis. Alternately, for history matching, we only require means and variances, so that we can specify only the first and second moments of $\underline{u}$, as this is sufficient for a Bayes linear analysis [3,5].

In what follows, we will assume that the computer simulator is represented by a corresponding emulator and explain how the notion of structural discrepancy is represented by appropriate modifications to this emulator. In order to do this, we will structure such discrepancy by separating it into two categories, namely *internal* and *external* model discrepancy.

Internal discrepancy arises from the specific technical choices made in the implementation of the model as the current simulator, f, and can be assessed by direct experiments on the simulator itself. Therefore, we learn about internal discrepancy by performing such experiments. External discrepancy comprises all of the features arising from limitations of the modelling process and which, therefore, cannot be quantified through such simulator experiments. Some of these features may correspond to processes that we know we have omitted from the model. Others may come from our recognition of the limitations of our understanding about the processes underlying the model. Finally, some may arise simply as we lack the time, expertise or resource to carry out the appropriate computer experiments, and thus we must count such aspects of structural discrepancy as external as well.

We will now discuss each of these two forms of discrepancy.

3. Internal discrepancy

(a) Internal discrepancy experiments

Internal discrepancy refers to any aspect of discrepancy variation that we choose to quantify by direct experiments on the computer simulator. These assess the effect of the various simplifying assumptions made in the model and the simulator implementation. As with any other aspect of modelling, we must choose the level of detail with which we carry out the internal analysis. Care at this stage will reduce the effort required at the later stages when we must account for all other aspects of structural discrepancy which were not included in the internal assessment. Here are some examples.

(1) We may vary parameters that are usually held fixed at pre-assigned values in the planned evaluations of the simulator, where this choice has been made in order to reduce dimension for the input space.
(2) We may have several subgroups, for example classified by gender, age and so forth, where the simulator uses the same parameter input values for each subgroup, for reasons of simplicity and ease of model fitting. The simulator may similarly impose coarse partitions for continuous effects, for example young, middle aged and old for effects which are continuous in age.
(3) We can add extra flexibility to our input parametrization, for example allowing some constant parameter values to vary over time and space.
(4) If the model science is based on the deterministic propagation of a state vector, then we may add a small amount of random noise at each propagation step, in recognition that the model rules oversimplify the true propagation of the state vector.
(5) If the model uses a simplified solver, then we may explore the effect of more careful equation solvers, for example increasing the grid resolution and number of iterations for the solver.
(6) The simulator might use fixed initial conditions, boundary conditions or external forcing functions, which we might choose to vary.

The aim of these experiments is not to fit a more complex model, as we judge that would be too expensive to fit to data or for real-world use. Instead, we quantify the effect of making the collection of simplifying assumptions given our intention to use our original simulator. However, if the experiments do identify such features as very large discrepancy bias or variance, then we would want to identify which features of the experiments were the major causes of these effects and consider modifying the simulator accordingly.

In each case, our ability to vary the corresponding feature depends in part on how the simulator is coded. Varying fixed parameters, for example, will usually be straightforward. Adding noise to the state vector depends entirely on whether this vector is accessible to access and modify in the simulator code. As a general design principle, the implementation of the model as a simulator should take careful account of the internal discrepancy experiments that we will wish to carry out in order to ensure that the simulator is fit for real-world use.

(b) Assessing internal discrepancy

We assess internal discrepancy through our chosen experiments as follows.

(1) For a single input parameter choice $\underline{x}$ with model run $f(\underline{x}) = (f_1(\underline{x}), f_2(\underline{x}), \ldots)$, we choose a set of perturbations $d_1, d_2, \ldots, d_k$, where each d_i is a perturbation of each one of the attributes selected for the experiment. We are treating discrepancy judgements as part of the modelling process. Therefore, the perturbations are chosen to reflect the degree of uncertainty that the modeller wishes to introduce. For example, random choices for a parameter value usually held fixed are made by consideration of the level of variation that is thought appropriate to introduce for the parameter. Similarly, we may replace a parameter fixed over time with the output of a continuous stochastic process, with mean equal to the parameter value, small variance and high correlation to represent the modified parameter at each time point, for example by judging the anticipated amount of drift in the parameter over a choice of fixed time intervals that would be considered acceptable and choosing parameters of the process to match these conditional judgements. As with any modelling process, we may explore different choices for such effects, for example choosing sets of samples representing different levels of variation.

(2) We evaluate the collection of k model runs, $F(\underline{x}) = (f(\underline{x}, d_1), f(\underline{x}, d_2), \dots, f(\underline{x}, d_k))$. $F(\underline{x})$ is a sample from the internal discrepancy distribution at input $\underline{x}$.

(3) For each output, $f_r(\underline{x})$, we look at the empirical distribution of $F_r(\underline{x})$, from which we may choose to extract simple summary statistics. Natural choices are the bias and variance:

$$B_r(\underline{x}) = f_r(\underline{x}) - \frac{1}{k}\sum_{i=1}^{k} f_r(\underline{x}, d_i), \tag{3.1}$$

$$V_r(\underline{x}) = \frac{1}{k-1}\sum_{j=1}^{k}\left[f_r(\underline{x}, d_j) - \frac{1}{k}\sum_{i=1}^{k} f_r(\underline{x}, d_i)\right]^2. \tag{3.2}$$

(4) We repeat this experiment for a range of input choices $\underline{x}_1, \dots, \underline{x}_n$, giving discrepancy samples $F(\underline{x}_1), \dots, F(\underline{x}_n)$. If the sample distributions are very similar, for each $\underline{x}_i$, then the average is a practical working choice for internal discrepancy.

(5) Otherwise, we extract summary statistics, for each $\underline{x}_i$, and build emulators for our chosen discrepancy summaries, for example $B(\underline{x})$ and $V(x)$, across the input space. Often this emulation is easy as these turn out to be simple smooth monotonic functions. If the number of experiment repetitions, k, is large, then we can view the observed sample statistics as being equal to the underlying population values for each chosen input value. Otherwise, we consider the sample statistics as estimates of these population values and add appropriate standard errors of estimation for our uncertainty about each population value, given the sample. This is similar to the standard way in which we build emulators for summary measures, such as the mean response, for stochastic simulators [6].

(6) We now have a value, with estimated uncertainty, for the internal discrepancy, at each choice $\underline{x}$ in the input space. Further, each internal discrepancy experiment gives samples from the full joint distribution of the internal discrepancy variables for all of the outputs. Therefore, we can compute any sample summaries for the joint distribution that we need; for example, we can assess the sample correlation between any output pair $F_r(\underline{x}_i)$ and $F_s(\underline{x}_i)$ for each $\underline{x}_i$.

(7) We now choose a form to incorporate internal discrepancy into our emulator for f. If we have been mainly focusing on bias and variance, then a simple representation would be of the form

$$f_I(\underline{x}) = f(\underline{x}) + \mu_I(\underline{x}) + \sigma_I(\underline{x})\epsilon, \tag{3.3}$$

where $\mu_I(\underline{x})$ is a vector of bias terms and $\sigma_I(\underline{x})$ is a vector of scale parameters, dependent on $\underline{x}$, given by the emulators built from the internal discrepancy experiments. We may view ϵ as a vector with zero mean and unit variance, independent of everything else, with correlation structure based on a combination of the corresponding correlation structures evaluated by the internal discrepancy experiments. If these vary greatly, we may take more care and emulate the general structure of this correlation matrix.

4. External discrepancy

External discrepancy arises from the inherent limitations of the modelling process embodied in the simulator and cannot be assessed by simulator experiments. This adds onto the internal discrepancy as

$$f^*(\underline{x}) = Mf_I(\underline{x}) + \epsilon_E(\underline{x}). \tag{4.1}$$

Here, $\epsilon_E(x)$ has some stochastic specification, given $\underline{x}$, as we will describe, and M is a scaling matrix, typically diagonal and often the identity, which allows us to rescale the simulator output to adjust for any mismatch between the scales of the simulator and the real-world phenomena.

One way to analyse the external component of structural discrepancy is described in [7] and is as follows. The function f describes how system properties (the inputs) affect system behaviour (the outputs). Our simulator approximates both the properties of the system and the rules for assessing system behaviour given system properties. Therefore, we may consider that our simulator is an approximation to a more detailed form, f^*, sometimes called the *reified model* (from *reify*—to consider an abstract concept to be real). f^* embodies all of our judgements about refinements to the science, improvements to solution accuracy, etc., so that additional structural discrepancy on top of this model will be unstructured.

We apply the simple discrepancy model (2.1) to f^* and to f^* alone, i.e.

$$\underline{y} = f^*(\underline{x}^*, \underline{w}^*) + \epsilon^*, \tag{4.2}$$

where ϵ^* is independent of everything else and $\underline{w}^*$ are any extra model parameters that we might introduce for the reified form.

In this construction, the model, f, is informative for the actual system, $\underline{y}$, because f is informative for reified model f^*. We cannot evaluate f^* but we can emulate it, so all of the analyses (history matching, calibration, forecasting) that we habitually carry out using the emulator of f can be transferred directly to the emulator of the reified form.

To see how we might emulate f^*, consider the simplest case, with one input x and one output $f(x)$, and an emulator for f of form $f(x) = a + bx + \epsilon(x)$, where a, b are constants and $\epsilon(x)$ is a stationary stochastic process. A simple emulator for f^* might be $f^*(x) = a^* + b^*x + \epsilon^*(x)$, where a^*, b^* are uncertain constants (with prior means a, b) and ϵ^* is a stationary process correlated with ϵ; for example we might set $\epsilon^*(x) = \gamma\epsilon(x) + \delta\epsilon'(x)$, where ϵ and ϵ' are independent. This expresses our judgement that, with more careful modelling, the global form of the emulator will not change but the rate of change of $f(x)$ with x is very likely to change. This form allows us to express structured judgements as to the potential effects of more detailed modelling. In contrast, the simple form for discrepancy, (2.1), is equivalent to imposing the simplified version of the reified emulator of form $f^*(x) = a^* + bx + \epsilon(x)$.

This approach is termed *direct* reification. In more generality, suppose that our emulator for component i of f is of form (2.2). Then our simplest emulator for f^* would then be

$$f_i^*(\underline{x}, \underline{w}) = \sum_j a_{ij}^* g_{ij}(\underline{x}) + u_i^*(\underline{x}) + u_i^{**}(\underline{x}, \underline{w}), \tag{4.3}$$

where we might model a_{ij}^* as $a_{ij}^* = c_{ij}a_{ij} + \nu_{ij}$ with, for example, c_{ij} treated as known, reflecting modelling judgements as to the relative rates of change of the two functions (often we will set these to one), and ν_{ij} treated as uncertain. We may choose to correlate $u(\underline{x})$ and $u^*(\underline{x})$, if we consider that divergences from the global form will share common features, but we will usually leave $u^{**}(\underline{x}, \underline{w})$ uncorrelated.

If we have more detailed judgements about particular deficiencies in the simulator, then we may build an emulator f' which represents the particular effects that we are considering and then link f' to f^* through direct reification. This is termed *structural* reification. For a discussion of the assessment of all of these quantities and an example of structural reification, see [7], which illustrates each part of this assessment in the context of a compartmental model for the potential shutdown of the Thermohaline circulation in the Atlantic ocean, by adding a notional additional compartment to represent aspects of circulation not captured in the given model and assessing the effect of this modified flow on each component of the model.

5. Structural discrepancy workflow

Given data, $\underline{z}$, corresponding to simulator outputs $f_h(\underline{x})$, we are usually interested in structural discrepancy mainly for those input choices which give acceptable matches to $\underline{z}$. In such cases, we can greatly simplify the workflow for discrepancy quantification. First, we identify, by history matching, the subspace of input values for which the simulator output is sufficiently close to the observed system history to be of interest as a possible choice for real-world uses of the simulator.

We can use a simple and cautious overall order of magnitude discrepancy assessment for this purpose. A good software package for carrying out such history matching is HMER [8]. Then, we re-sample and re-emulate the simulator, carry out the internal discrepancy experiments and derive discrepancy forms, such as (3.3) and (4.3), all within this input subspace. This will often be a much simpler task than to emulate and assess discrepancy accurately for the whole of the input space, partly as the new space is much smaller and partly because function behaviour within the reduced space is usually much more consistent. We usually go through a final stage of history matching with the more careful emulators and discrepancy assessments.

Depending on the problem at hand, the history match may be the endpoint of the analysis. Alternately, if we prefer to calibrate the model, for example, if we wish to make inferences about the likely values of some of the model parameters which have clear physical meanings, then we may carry out a full Bayesian analysis within the reduced parameter space using the disrepancy structure that we have constructed within this space.

We may have further goals for our modelling. For example, we may want to forecast, which requires careful structural discrepancy assessment across past and future outcomes for plausible choices of inputs, $\underline{x}^*$. If we want to predict some future system outcomes, $\underline{y}_p$, corresponding to function outputs $f_p(\underline{x})$, given observed historical data $\underline{z}$, then we update uncertainties for $\underline{y}_p$ given $\underline{z}$, for each acceptable choice of $\underline{x}^*$, using the decompositions

$$\underline{z} = f_h(\underline{x}^*) + \epsilon_h^*(\underline{x}^*) + e_h, \quad \underline{y}_p = f_p(\underline{x}^*) + \epsilon_p^*(\underline{x}^*), \tag{5.1}$$

where $\epsilon_p^*, \epsilon_h^*$ are structural discrepancies for $\underline{y}_p, \underline{y}_h$ and e_h is measurement error for $\underline{z}$ [3]. Where our beliefs are described by a second-order specification, the appropriate mechanism for updating those beliefs are the Bayes linear [5]:

$$\begin{aligned}\mathbb{E}_{\underline{z}}[\underline{y}_p] &= \mathbb{E}[\underline{y}_p] + \mathbb{C}\mathrm{ov}[\underline{y}_p, \underline{z}]\mathbb{V}\mathrm{ar}\,[\underline{z}]^{-1}\,(\underline{z} - \mathbb{E}[\underline{z}]),\\ \mathbb{V}\mathrm{ar}_{\underline{z}}[\underline{y}_p] &= \mathbb{V}\mathrm{ar}[\underline{y}_p] - \mathbb{C}\mathrm{ov}[\underline{y}_p, \underline{z}]\mathbb{V}\mathrm{ar}\,[\underline{z}]^{-1}\,\mathbb{C}\mathrm{ov}[\underline{z}, \underline{y}_p],\end{aligned} \tag{5.2}$$

where $\mathbb{E}_{\underline{z}}[\underline{y}_p]$ and $\mathbb{V}\mathrm{ar}_{\underline{z}}[\underline{y}_p]$ represent the adjusted expectation and variance of the future outcomes, $\underline{y}_p$, given the observations, $\underline{z}$ - in other words, our forecasts. Where beliefs about $\underline{z}$ and $\underline{y}_p$ are described probabilistically then the distribution $\underline{y}_p \mid \underline{z}$ is required, commonly obtained with Markov chain Monte Carlo methods.

6. Discussion

In this paper, we treat the assessment of structural discrepancy as part of the modelling process. This may be contrasted to approaches based on variants of (2.1); see for example [2] in which model discrepancy depends only on controllable parameters such as time. In such approaches, assessment of model discrepancy becomes a problem of statistical estimation, which raises important technical challenges as there is confounding between the model response and the structural discrepancy component. Various methods have been developed to address these challenges, for example, the modularization approach, introduced in the context of computer models in [9], in which these problems are handled by separating the probabilistic specification into discrete sub-modules and carefully controlling message passing between them. This is similar to the idea of 'cutting feedback' as implemented in the popular Bayesian software package Winbugs [10]. A common estimation technique in this area to address such issues, introduced in [11], is based on projection using the L_2 norm of the function, which can be viewed as a continuous analogue of ordinary least squares methods for parameter choice which minimize squared differences between physical outputs and simulation outputs. A Bayesian formulation for such approaches was introduced in [12]. A recent overview of advancements in addressing the challenges of the unidentifiability issues raised when incorporating model inadequacy, and related issues, is given in [13].

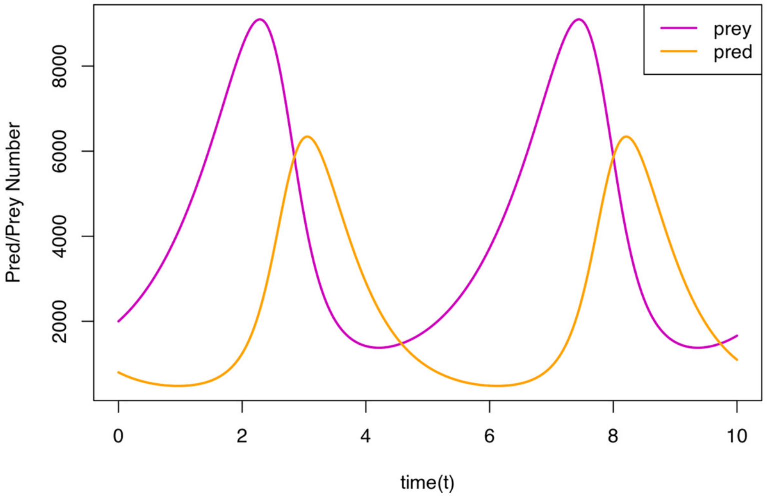


Figure 1. Output from a single run of the Lotka–Volterra model against time. Note that prey (purple line) exhibits the classic peaks in population level, while the predator population (orange lines) has similar peaks that follow that of the prey, with the whole system exhibiting oscillatory behaviour.

In our formulation, such confounding is not an issue as each part of our uncertainty specification is separately modelled and assessed. Further, the basic issue with any approach based on variants of (2.1) is that it implies there is a value x^* for which all of the information that the simulator provides is contained in the single evaluation $f(x^*)$. Usually, in practice, this is unlikely to be the case. [7] provides a careful analysis of the potential logical inconsistencies in such a view, for example by considering the thought experiment of constructing an improved simulator which respects the physical behaviour of the system more closely than does the current simulator and demonstrating contradictions within the single sufficient evaluation view. See also [14] for a somewhat more structured approach to model discrepancy assessment, albeit within a fairly specific setting.

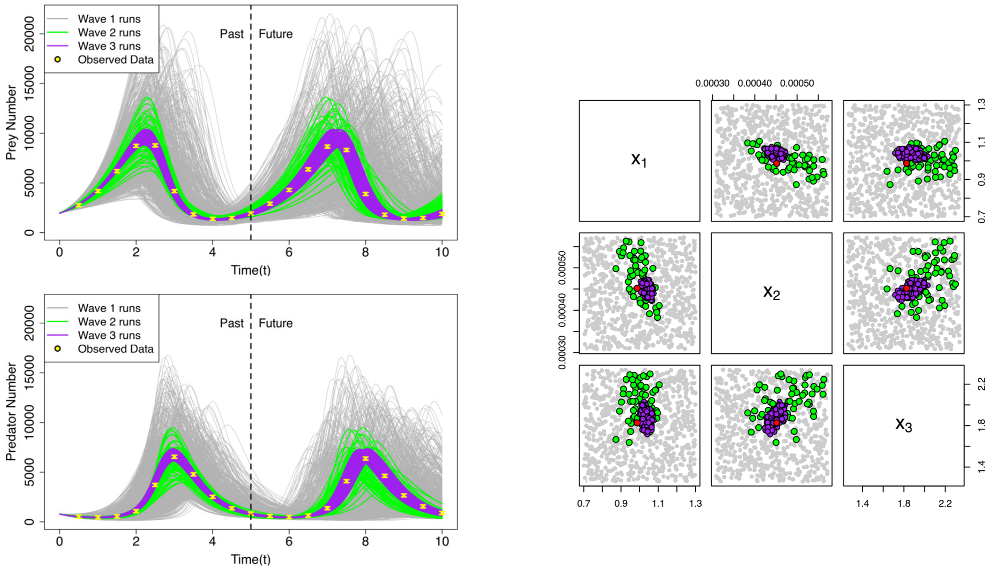
7. Example: the predator–prey model

To illustrate the methods for assessing model discrepancy, we employ a relatively simple and fast Lotka–Volterra Predator–Prey (LVPP) model. This will allow us to focus on demonstrating the structural discrepancy assessment, however when analysing more expensive and/or more complex models we would follow the same workflow, but make more use of emulation and history matching (to perform a more extensive input parameter search), as discussed below. See [15] for an analysis of a more complex variant of the LVPP model. The LVPP model represents the dynamics, over time, of two interacting species—the prey, f_1, and the predators, f_2—describing the changing populations of each species over time and generating a two-dimensional time series of species counts, indexed by time t. The system's dynamics are described by the following pair of differential equations:

$$\frac{\mathrm{d}f_1}{\mathrm{d}t} = x_1 f_1 - x_2 f_1 f_2, \quad \frac{\mathrm{d}f_2}{\mathrm{d}t} = x_2 f_1 f_2 - x_3 f_2, \tag{7.1}$$

where $\underline{x} = (x_1, x_2, x_3)$ are the inputs to the model and comprise three rate parameters that govern the speed of reproduction of prey, the predator–prey interaction and the death rate of predators, respectively. The output from a single evaluation of the Lotka–Volterra model is shown in figure 1, exhibiting the classic lag between peaks of prey and predator populations over time.

For our analysis, the input parameters to the model are assumed to have ranges of $x_1 \in [0.7, 1.3]$, $x_2 \in [3.1 \times 10^{-5}, 5.7 \times 10^{-4}]$ and $x_3 \in [1.3, 2.3]$, representing partially informed prior information.



(a) Model evaluations for prey and predator outputs (top and bottom respectively). Vertical dashed line represents the $t = 5$ divide between past and future.

(b) 3-dimensional input parameter run locations, projected onto 2-dimensional subspaces, for all three waves.

Figure 2. Lotka–Volterra evaluations in output space (left) and input space (right) for wave 1 (grey), wave 2 (green) and wave 3 (purple). Red point (right) is the wave 2 test point. Yellow points and error bars (left) are the observed data.

Note that, wider ranges would require a more detailed multi-wave history match [16], which although perfectly feasible, is not our focus here. The LVPP equation (7.1) can be numerically integrated to generate a time series of simulator output once the initial conditions have been specified, which we here take to be $[f_1(t=0), f_2(t=0)] = [2000, 800]$. To explore the full ranges of input parameter values, a first *wave* of 750 runs of the model were designed over the three-dimensional input space using a maximin Latin hypercube [17], and the collection of resulting model outputs are drawn as the grey curves in figure 2a, with their corresponding input values shown as grey points in figure 2b.

To illustrate our structural discrepancy assessment workflow for this model we require observed data, $\underline{z}$. As this model is entirely synthetic, pseudo-observations are generated from a single evaluation of a more complex and stochastic version of the Lotka–Volterra model simulated using the Gillespie algorithm (see e.g. [18]), under slightly different initial conditions $(1910, 710)$, and using time-varying inputs $\underline{x} \to g(t)\underline{x}$ that vary with a quadratic dependence centred around 1: $g(t) = 1 - ((t-5)^2/25 - 0.5)/10$, representing a subtle seasonal, periodic change to the reaction rates of the real system. We, therefore, obtain a set of system outputs $\underline{y}$ that cannot be perfectly recreated using the LVPP model described above—just as would be expected in any analysis of a real-world system and its corresponding data. Uncorrelated observation error $\underline{e}$ with $\sigma_{\underline{e}} = 50$ (representing the accuracy of the observation process) is added to obtain pseudo-observational data $\underline{z} = \underline{y} + \underline{e}$, and is shown in figure 2a as yellow points and error bars.

(a) Initial history matching

Given the data $\underline{z}$, we are interested in assessing the structural discrepancy of our simulator for input choices that give acceptable matches to the observations. However, a cursory inspection of the model evaluations in figure 2a reveals that a sizeable majority of the first wave of model evaluations show little to no correspondence to our observations. Thus, we follow the outlined workflow and begin by seeking to refine our initially broad and conservative input space to a more focused set of parameter combinations which could feasibly yield outputs that are a close enough match to the data to be of further interest. The methodology here is that of history matching [6,19] using a conservative order of magnitude model discrepancy assessment of ±15% of model output values. Our model discrepancy choice is deliberately cautious at this stage and will be refined in the subsequent analysis; its purpose here is to simply ensure that obviously incompatible input parameter combinations can be readily identified and removed from consideration. We anticipate that the refined assessment will be smaller than this initial, cautious ±15% value. Given this discrepancy and simple univariate emulators for the simulator outputs, we compute the *implausibility*, $I\left(\underline{x}\right)$, at each of our wave 1 input points:

$$I\left(\underline{x}\right)=\frac{\left|\mathbb{E}\left[f(\underline{x})\right]-\underline{z}\right|}{\sqrt{\mathbb{V}\mathrm{ar}\left[f(\underline{x})-\underline{z}\right]}}=\frac{\left|\mathbb{E}\left[f(\underline{x})\right]+\mathbb{E}\left[\epsilon_h^*\right]-\underline{z}\right|}{\sqrt{\mathbb{V}\mathrm{ar}\left[f(\underline{x})\right]+\mathbb{V}\mathrm{ar}\left[\epsilon_h^*\right]+\mathbb{V}\mathrm{ar}\left[e_h\right]}}, \tag{7.2}$$

where $\mathbb{E}\left[f(\underline{x})\right]$ and $\mathbb{V}\mathrm{ar}\left[f(\underline{x})\right]$ are the expectation and variance of our emulator of the simulation at $\underline{x}$ (required for slow models while here we use the model output directly), $\underline{z}$ is the observed data, e_h is the observational data error and ϵ_h^* is our structural model discrepancy. This yields an implausibility value for every input parameter choice, $\underline{x}$, and every output component of $\underline{y}$. The implausibility measure takes large values in the presence of strong disagreement between a particular simulator output and its corresponding data, and small values in the presence of either good matches or high uncertainty [19]. These implausibility measures are then combined into a single implausibility value, for each input point $\underline{x}$, via the maximum implausibility:

$$I_M\left(\underline{x}\right)=\max_i I_{(i)}(\underline{x}), \tag{7.3}$$

where $I_{(i)}(\underline{x})$ is the implausibility (7.2) calculated for the ith output component of the simulator. This ensures that if the simulator fails to match the data on any single component, it is judged an implausible match. Conversely, good matches are only declared when all output components have correspondingly small implausibilities and so are close to the data with low uncertainty for all model outputs. Here, to demonstrate our methods, we take the historical data z to be outputs from the first peak only, such that $0\leq t\leq 5$. The time $t=5$ is viewed as the 'present day' (shown as the vertical dashed line in figure 2a) and all other future data for $t>5$ is shown for comparative purposes but not used in the history match, nor in the subsequent forecasts.

To distinguish these implausible input points to exclude from future study from the remaining non-implausible set of points, we apply a threshold to the maximum implausibility computed via (7.3) and retain all those input points with lower implausibilities as our *wave 2* input points. We choose a threshold of $I_M\left(\underline{x}\right)=3$ motivated by Pukelsheim's 3-sigma rule [20] which states that for any uni-modal continuous distribution, 95% of its probability lies within 3-sigma of its mean. When applied directly to the 750 runs, this results in 60 suitable model runs that are deemed sufficiently compatible with the observed data to warrant further analysis. These are shown as green lines in figure 2a and in the context of the input space in figure 2b as green points, which now more closely mirror the observed data over the first peak ($0\leq t\leq 5$) and correspond to a smaller region within the input parameter space. This refinement stage can be repeated multiple times with additional design points generated in this reduced input space and emulators re-fit after each wave to focus further if required. See [16,21,22] for details regarding history matching, including applications in cosmology, systems biology and epidemiology and also for comparisons to alternative approaches such as MCMC and ABC.

(b) Internal discrepancy assessment

We perform the internal model discrepancy assessment following the steps of §3(b). We begin with a single input location chosen from our wave 2 runs (step (i)), hereinafter referred to as the *wave 2* *test point* and highlighted in red on figure 2b. We select a set of perturbations $d_1, \ldots, d_k$ of the model inputs to be applied to the test point, in this case formed by varying initial conditions and—given the time series nature of this simulation—adding temporal variation to the inputs. Specifically, for each perturbation, the initial conditions were drawn from $N(2000, 40^2)$ and $N(800, 40^2)$ for prey and predator, respectively, rather than being fixed at the values of 2000 and 800 that were used in the main model runs, with the SD of 40 chosen to reflect the scientist's uncertainty over these previously fixed initial conditions. Time-varying inputs were introduced by modifying the LVPP equation (7.1) to multiply the rate constants by functions of time, such that each x_i is replaced by $g_i(t)x_i$ for $i = 1, 2, 3$. The function $g_i(x)$ was chosen to be cyclical about the value of 1 and took the form

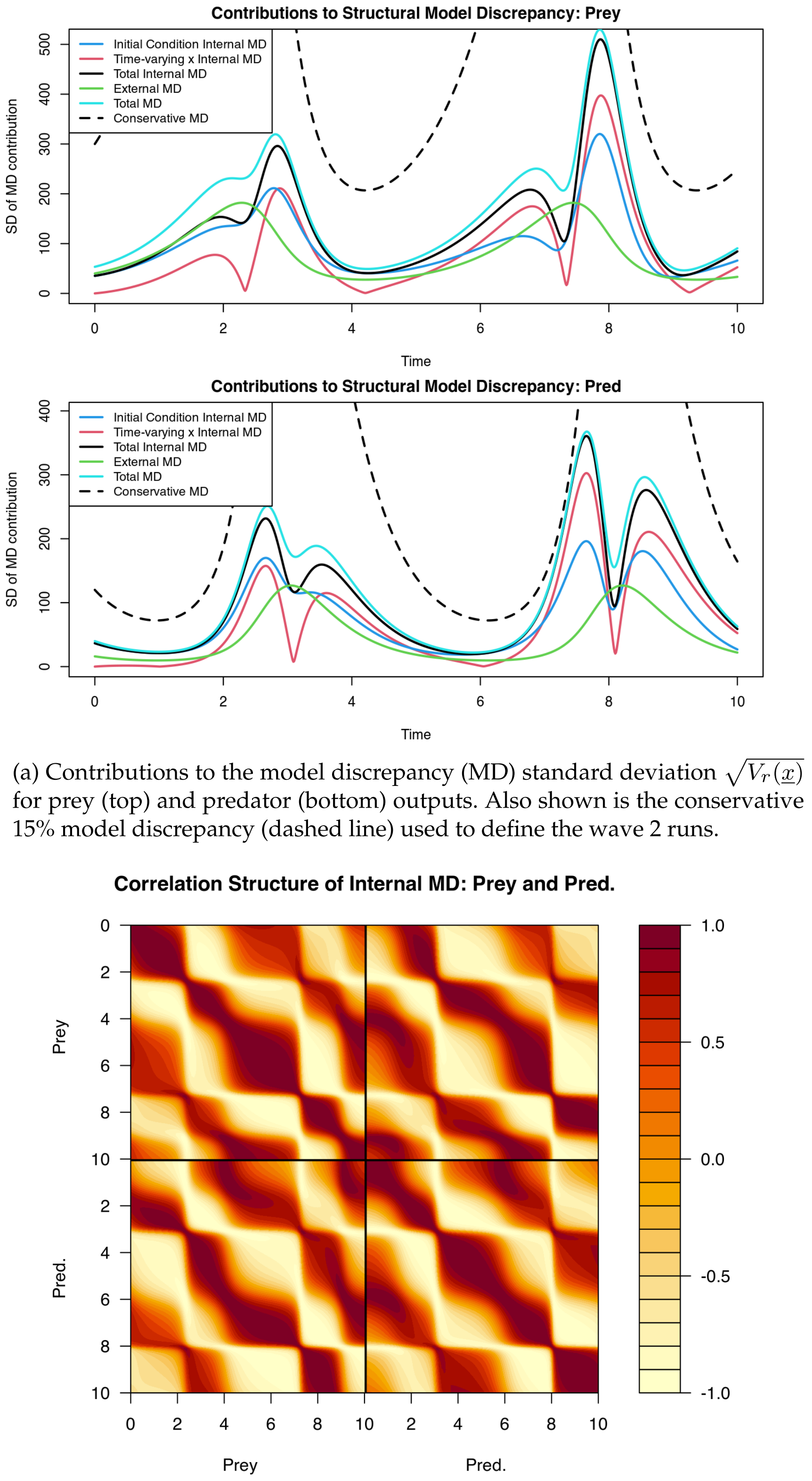
$$g_i(t) = 1 + c_i \sin\left[(2\pi\{t - a_i\}b_i)/T\right], \tag{7.4}$$

where $T = 10$, and where $a_i \sim U(4.5, 5.5)$, $b_i \sim U(0.6, 0.9)$ and $c_i \sim N(0.02, 0.05^2)$ are drawn independently for each perturbation. Note that the time variation here is sinusoidal and hence different from the quadratic variation used to generate the real system $\underline{y}$. Hence, it along with the distributional choices for a_i, b_i and c_i is designed to capture the scientist's uncertain judgements regarding a possible subtle seasonal effect (any remaining uncertainty, not captured by this parameterised form, can be included in the external discrepancy). Following this scheme, we construct 50 perturbations for each of the initial conditions and the time-varying inputs separately, as well as 200 perturbations where both elements were varied together in order to assess the impact of each modification and compare their relative effects.

We now evaluate the collection of simulator runs under each of our permutations, $F(\underline{x}) = [f(\underline{x}, d_1), f(\underline{x}, d_2), \ldots, f(\underline{x}, d_k)]$, per step (ii). The resulting collection of evaluations form a sample of the internal discrepancy distribution, which can be summarised (step iii) by simple statistics such as bias (3.1) and variance (3.2). In figure 3a, we focus on the latter, where we plot the contributions of the initial conditions and the time-varying inputs to the standard deviation of the internal discrepancy, i.e. $\sqrt{V_r(\underline{x})}$. The contributions to the internal discrepancy due to varying initial conditions and time varying x inputs are shown as the blue and red lines, respectively (calculated using the two sets of 50 perturbations that varied each internal feature), while the total internal discrepancy $\sqrt{V_r(\underline{x})}$ is given by the black line (calculated using the 200 perturbations that varied both internal features). A nominal external discrepancy equal to 2% of the model output is employed to represent the remaining structural model discrepancy not captured by the internal analysis and is given by the green line. The total model discrepancy is shown as the light blue line. The conservative model discrepancy used above to define the wave 2 runs (15% of the model output) is shown as the dashed black lines.

First, we note that the discrepancy contributions are not uniform, and clearly vary with time with peaks in the vicinity of those observed in the data. The initial condition uncertainty (blue line) is a dominant component of discrepancy at early times, but while the effects of the time-varying inputs (red line) are small at first they occasionally become the dominant source of discrepancy at late times. These results are intuitive, with simulation output most sensitive to initial conditions at early times and becoming less relevant as the simulation progresses. Note also that the total model discrepancy is substantially less than the conservative 15% used in the initial history matching stage, so there is no risk of mistakenly excluding viable parameter combinations.

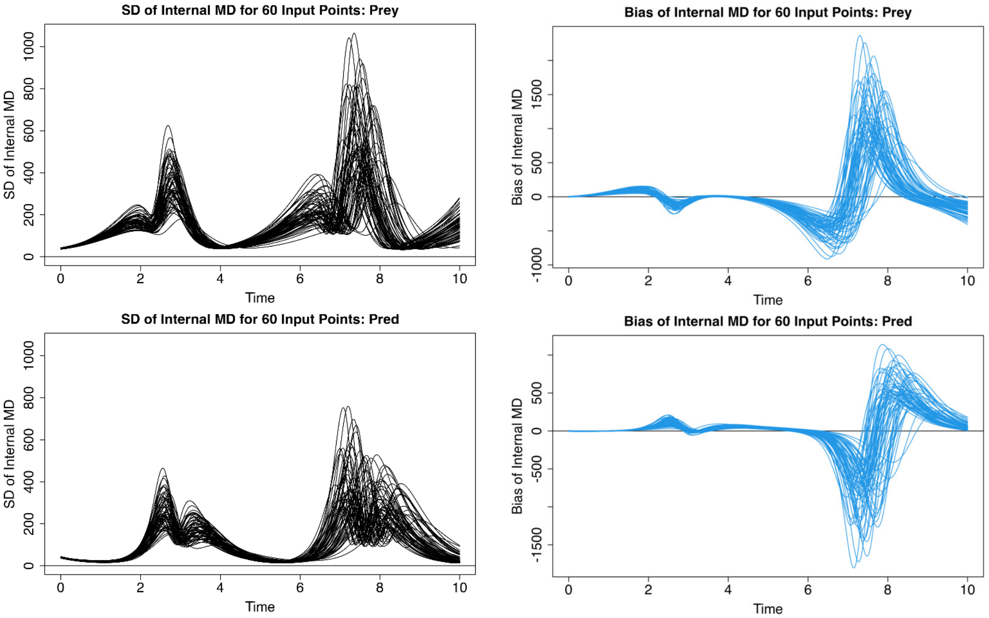
A particularly insightful feature of this approach to the assessment of internal discrepancy assessment is that we can also examine the correlation structure (across all outputs over time, and of prey and predator type) of the induced internal discrepancy. From the collection of 200 permutations where we varied both sources of discrepancy, we can construct the sample correlation matrix which is shown in figure 3b. This approximates the correlation $\mathrm{Corr}[F_r(\underline{x}_i), F_s(\underline{x}_i)]$, where r and s label both the time and output type (prey or predator). Here, we see substantial structure between the discrepancy over the different output components, with strong positive and negative correlations induced by the coupled and oscillatory behaviour of the simulator outputs. This



(a) Contributions to the model discrepancy (MD) standard deviation $\sqrt{V_r(\underline{x})}$ for prey (top) and predator (bottom) outputs. Also shown is the conservative 15% model discrepancy (dashed line) used to define the wave 2 runs.

(b) Induced correlation structure of the internal discrepancy.

Figure 3. Results of internal discrepancy analysis at the wave 2 test input point.



(a) Internal Discrepancy SD $= \sqrt{V_r(\underline{x})}$.

(b) Internal Discrepancy Bias $B_r(\underline{x})$.

Figure 4. The standard deviation $\sqrt{V_r(\underline{x})}$ and bias $B_r(\underline{x})$ of the internal discrepancy, calculated at each of the 60 wave 2 input points (giving each of the 60 black/blue lines), plotted against time ($r = t$) for the prey and predator outputs.

provides valuable information that would be entirely overlooked were we to take a naive and unstructured model discrepancy specification.

While we can glean substantial information from this analysis alone, we have thus far only explored a single point from the wave 2 runs, and it is reasonable to suspect that the observed internal discrepancy properties may vary with $\underline{x}$. Therefore, the natural progression of the analysis is to repeat the permutation experiment with each of the remaining wave 2 input points (step iv).

Recalculating the internal model discrepancy standard deviation, $\sqrt{V_r(\underline{x})}$, for each of the 60 cases we obtain the results in figure 4a. Similar calculation of the biases, $B_r(\underline{x})$, yield the results in figure 4b. Each curve in the plots represent the results of the same calculation as that presented in figure 3a, only now applied to each of the wave 2 input points. We can see clearly that both $\sqrt{V_r(\underline{x})}$ and $B_r(\underline{x})$ vary substantially with input location $\underline{x}$, with similar overall shapes to those observed above albeit with notable variation in magnitude of the peaks. Given that these expressions of the discrepancy clearly vary as the inputs to the simulator vary, it would be inappropriate to simply reduce these results to a simple summary such as an average. If we were to do so, we would grossly oversimplify our assessment of our internal model discrepancy. Instead, we proceed to construct emulators for these quantities over the wave 2 input space (step v) for use in subsequent calculations.

The discrepancy standard deviation, $\sqrt{V_r(\underline{x})}$, and bias, $B_r(\underline{x})$, were then emulated over the wave 2 locations. Simple emulators based on linear regressions were used, noting that the regression R^2 was above 0.85 for all outputs. This provides a more detailed and nuanced description of model discrepancy over the space of $\underline{x}$ that we use to revisit and refine our collection of feasible input points. We can now refine our emulator for the model (step vii) into the form (3.3) by using the emulators of the internal discrepancy bias and standard deviation as the $\mu_I(\underline{x})$ and $\sigma_I(\underline{x})$ components. This provides a more detailed and structured description of the internal discrepancy over the input space, which can be reintroduced to our history matching calculations in §7*a*. By doing so, we can further refine our collection of suitable input parameter combinations using the updated implausibility given the additional information gained during the structural discrepancy analysis. At this stage, we could generate new input candidate points and, by rejection sampling, retain only those which satisfied our implausibility threshold under the new model discrepancy. Doing so yields the new input combinations shown in purple on figure 2b, which could form a potential wave 3 of our analysis were we to continue iterating the history match.

(c) Using structural discrepancy to enhance forecasting

In the previous section, we carefully assessed the internal model discrepancy via a series of experiments on the computer simulator. While this has already refined our understanding of the gap between our simulation and the real system, the same information can greatly aid other calculations we may seek to perform—such as forecasting, as discussed in §5. Suppose our rationale for exploring this simulation is to predict the next peak population (in terms of time and magnitude) for both species on the basis of only the data observed until time $t = 5$, with data beyond this point as yet unseen. Specifically, let us consider how we can use information on the location of the first population peak alongside our improved discrepancy specification to update the model discrepancy and improve potential forecasts for the second peak in the two populations.

First, our focus shifts from the time series outputs of the simulator, $f(\underline{x}, t) = [f_1(\underline{x}, t), f_2(\underline{x}, t)]$, for the two species, to derived quantities of the timing and magnitude of the kth peak in the population for the ith species, denoted by $\tilde{f}^{(k)}(\underline{x}) = [\tilde{f}^{(k)}_{1,\mathrm{time}}(\underline{x}), \tilde{f}^{(k)}_{1,\mathrm{mag}}(\underline{x}), \tilde{f}^{(k)}_{2,\mathrm{time}}(\underline{x}), \tilde{f}^{(k)}_{2,\mathrm{mag}}(\underline{x})]$. Thus from every evaluation of the simulator, we can determine the properties of the first simulated population peak, $\tilde{f}^{(1)}(\underline{x})$, and the second peak, $\tilde{f}^{(2)}(\underline{x})$. In the notation of (5.1), our historical simulator outcomes are $f_h(\underline{x}) = \tilde{f}^{(1)}(\underline{x})$, and our future outcomes to be predicted are $f_p(\underline{x}) = \tilde{f}^{(2)}(\underline{x})$, and any forecasting calculation will require a structural discrepancy component for both the past, ϵ^*_h, and future values, ϵ^*_p. Before assessing the internal discrepancy, we begin by specifying a simple zero mean uncorrelated external discrepancy of 2% and ± 0.03 for the magnitude and the timing of the peaks, respectively, for both species. Additionally, we adopt a zero-mean uncorrelated observational error e_h with standard deviations 50 and 0.025 for the magnitude and timing of the peaks, representing an imperfect peak observation process.

A simple approach to the prediction problem would be to identify a single input choice, $\underline{x}$, that we viewed as an acceptable candidate for $\underline{x}^*$ and to explore the forecast this particular choice would give. A naive first step would simply use the simulator output for the second peak, $\tilde{f}^{(2)}(\underline{x})$, as the forecast, supplemented by the additional uncertainties we had specified for our discrepancies, ϵ_p. However, there are various correlations present between our model discrepancy components - between peaks of predator and prey, and between past and future—which mean we can transfer information about our ability to predict at the first peak to our prediction for the second peak via the model discrepancy. More formally, assuming that $\underline{x}$ is a suitable $\underline{x}^*$, the correlations that exist between ϵ_h and ϵ_p induced by the internal discrepancies, in turn, induce correlations between the past observations $\underline{z}$ (peak 1) and the future system value $\underline{y}_p$ (peak 2) via (5.1). These correlations can then be used to update our beliefs about the behaviour of $\underline{y}_p$ given $\underline{z}$, represented by $\mathbb{E}_{\underline{z}}[\underline{y}_p]$ and $\mathbb{V}\mathrm{ar}_{\underline{z}}[\underline{y}_p]$ as given by (5.2).

In figure 5, we show an example of this calculation for a single wave 2 input combination, where the model output, $f(\underline{x}, t)$, is given as the solid black line and its 200 perturbations from §7*b* indicated as purple lines. For each member of this collection of 200 evaluations $F(\underline{x})$, we can extract the 8 peak outputs of interest, $[\tilde{f}^{(1)}(\underline{x}), \tilde{f}^{(2)}(\underline{x})]$, and generate a 200×8 matrix of simulator outputs from which we can assess the 8×8 discrepancy covariance matrix, $V(\underline{x})$, using (3.2) from §7*b*. This provides valuable information on the discrepancy correlations between all of the 8 peak outputs—the first four of which correspond to the first peak, $\tilde{f}^{(1)}(\underline{x})$, and the remainder to the second peak, $\tilde{f}^{(2)}(\underline{x})$. Combining this internal discrepancy information with the unstructured external discrepancy specification gives an assessment of the overall discrepancy variance for the 8 peak output variables.

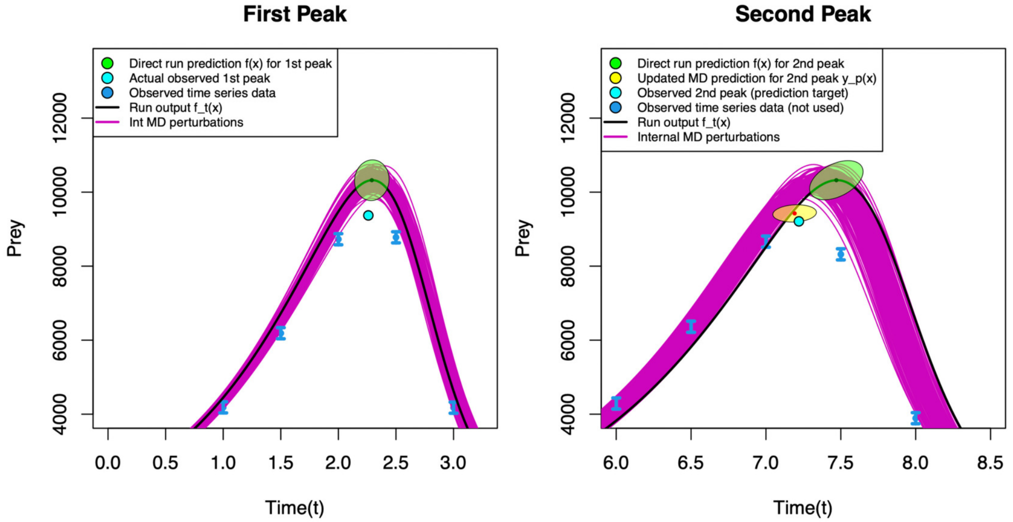
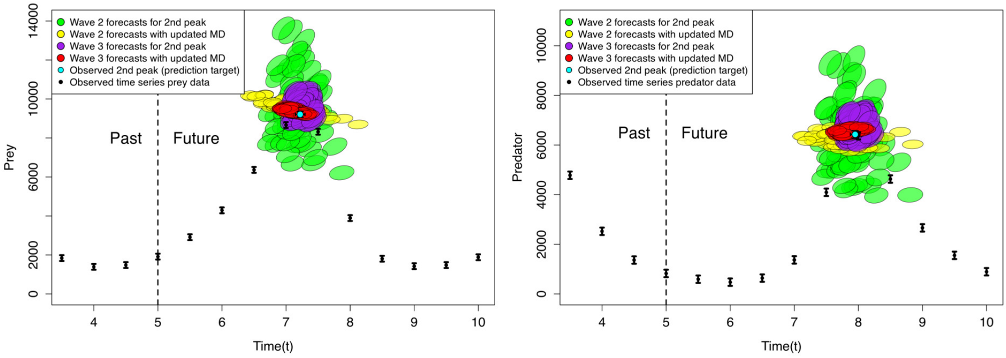


Figure 5. Impact of updating model discrepancy on the second peak location forecast from a single wave 2 evaluation. Left: observed first peak location (light blue point) used to update the model discrepancy. Right: forecast before update (green ellipse) and after update (yellow ellipse). Actual second peak location (the target for the forecast): light blue point. Note that, time series data for the second peak was not used, and is just shown for comparative purposes.

Using this information, we draw the green ellipses in figure 5 to represent a naive prediction that is centred on $f(\underline{x})$ with the ellipse orientation and axis lengths describing the assessed discrepancy uncertainties represented by $\mathbb{V}\text{ar}[\epsilon_h]$ and $\mathbb{V}\text{ar}[\epsilon_p]$. It is clear from the left panel that this yields an inadequate prediction to the first peak, as the observation (cyan point) is both lower and slightly earlier than this simple simulator-based forecast suggests. Therefore, it is reasonable to expect a similar deficiency in our forecast of the second peak using this particular evaluation of the simulator. Using what we have learned about the prediction at the first peak, we can adjust our discrepancy, ϵ_p, and hence our prediction for $\underline{y}_p$ to $\mathbb{E}_{\underline{z}}[\underline{y}_p]$ via (5.2). In the right panel of figure 5, the original naive prediction in green has been adjusted to give the improved forecast for $\underline{y}_p$, $\mathbb{E}_{\underline{z}}[\underline{y}_p]$, as the red point, with associated uncertainty, $\mathbb{V}\text{ar}_{\underline{z}}[\underline{y}_p]$, in yellow, giving a more accurate prediction of the second peak location after this update. It is important to note that this calculation is looking only at a single input choice, $\underline{x}$, effectively assuming it is a good candidate for $\underline{x}^*$. This will not be universally true for all $\underline{x}$, and this calculation will not rescue a bad prediction made from an inappropriate input choice; instead the input should be deemed implausible and discarded after a later history match that used the second peak observation.

It is instructive to repeat this predictive calculation for all of the cautious wave 2 input points. We thus obtain the results in figure 6 with the simple forecasts for each input as a green ellipse, alongside the corresponding updated forecasts as yellow ellipses. We can clearly see the effect of the updated model discrepancy by substantially reducing the uncertainty around each of the forecasts and moving the forecasts closer to the prediction target. The set of yellow ellipses already deliver a vastly improved forecast for the peaks and represents a somewhat robust forecast as it is based on a very cautious set of wave 2 runs (that were defined using a conservative, large initial model discrepancy assessment). The substantial impact of an additional wave of history matching to refine our space of plausible input parameters can be seen by contrasting these results with the naive (purple) and adjusted (red) predictions using the wave 3 evaluations, which have greatly reduced uncertainties and concentrate predictions in a much tighter region around the prediction target. Refer to figure 2 for the locations and outputs of the wave 2 and 3 runs. We assessed the internal model discrepancy for the wave 3 runs in the same way as for wave 2, but again for a slower model, emulation of the relevant covariance matrices could be employed and would dramatically reduce the total number of runs required.



(a) Second peak predictions for prey. (b) Second peak predictions for predators.

Figure 6. Forecasts for the second peak of the LVPP model: wave 2 predictions without updating the model discrepancy (green), and after updating model discrepancy (yellow); wave 3 predictions without updating the model discrepancy (purple), and after updating model discrepancy (red). Actual second peak location (the target for the forecast): light blue point.

The above forecasts are in alignment with the history matching paradigm where we do not seek to probabilize the input space and seek to employ only a limited set of uncertainty judgements [16]. However, a further step would be to consider a weighting of the input points used in the forecast according to their fit to data, and thereby weighting the corresponding forecasts they produce. For example, adopting Uniform or Gaussian prior distributions over the inputs lead to tractable forecasts without the need for extensive numerical integration [23]. Alternatively, choices of more general prior distributions effectively leads to a fully Bayesian calibration and forecast [2], though this does require making a number of distributional judgements that may be harder to justify and possibly unnecessary, e.g. if the current forecast dictates a very clear decision choice which will be unaltered by further probabilistic nuance.

8. Conclusion

Model or structural discrepancy is an essential component in the analysis of computer models as it reflects the uncertainty that surrounds our simulator's ability to reproduce the real system it attempts to model. While discrepancy can sometimes be a challenging concept to reason about, we have described various potential strategies for assessing sources of structural discrepancy that are applicable to a wide range of models.

Careful discrepancy assessment will: (i) correct our overconfidence in our projections (by adding appropriate levels of additional uncertainty), (ii) increase our forecast accuracy (by making better choices for $\underline{x}^*$, avoiding overfitting and correcting for systematic biases in our simulator), (iii) help us to make reliable control choices for future outcomes (by recognising the real-world risks of our various control choices), and (iv) allow us to have a reasoned view as to how the quality of the model affects the quality of forecasts (by modifying features such as the magnitude of discrepancy variances and repeating our calculations).

Data accessibility. Data are simulated, and code to generate data and perform the analysis is available at: https://www.maths.dur.ac.uk/users/i.r.vernon/papers/health_bio/

Declaration of AI use. We have not used AI-assisted technologies in creating this article.

Authors' contributions. M.G.: methodology, writing—original draft; I.V.: investigation, methodology, writing—original draft, writing—review and editing; J.C.: writing—original draft, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. No funding has been received for this article.

References

1. Brynjarsdóttir J, O'Hagan A. 2014 Learning about physical parameters: the importance of model discrepancy. *Inverse Probl.* **30**, 114007. (doi:10.1088/0266-5611/30/11/114007)
2. Kennedy MC, O'Hagan A. 2001 Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B* **63**, 425–464. (doi:10.1111/1467-9868.00294)
3. Goldstein M, Huntley N. 2017 Bayes Linear Emulation, History Matching, and Forecasting for Complex Computer Simulators. In *Handbook of uncertainty quantification* (eds G Ghanem, H Higdon, O Owhad), pp. 9–32. Cham, Switzerland: Springer International Publishing. (doi:10.1007/978-3-319-12385-1_14)
4. Craig PS, Goldstein M, Seheult AH, Smith JA. 1997 Pressure matching for hydrocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments. In *Case studies in Bayesian statistics lecture notes in statistics* (eds C Gatsonis, JS Hodges, RE Kass, R McCulloch, P Rossi, ND Singpurwalla). New York, NY: Springer New York. (doi:10.1007/978-1-4612-2290-3_2)
5. Goldstein M, Wooff D. *Bayes linear statistics: theory and methods*. Chichester, UK: Wiley.
6. Andrianakis I, Vernon I, McCreesh N, McKinley TJ, Oakley JE, Nsubuga RN, Goldstein M, White RG. 2017 History matching of a complex epidemiological model of human immunodeficiency virus transmission by using variance emulation. *J. R. Stat. Soc. Ser. C* **66**, 717–740. (doi:10.1111/rssc.12198)
7. Goldstein M, Rougier J. 2009 Reified Bayesian modelling and inference for physical systems. *J. Stat. Plan. Inference* **139**, 1221–1239. (doi:10.1016/j.jspi.2008.07.019)
8. Scarponi D *et al*. 2023 Demonstrating multi-country calibration of a tuberculosis model using new history matching and emulation package - hmer. *Epidemics* **43**, 100678. (doi:10.1016/j.epidem.2023.100678)
9. Bayarri MJ, Berger JO, Liu F. 2009 Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Anal.* **4**, A404 119–50. (doi:10.1214/09-ba404)
10. Sat DJ, Best NG, Lunn D. 2004 *WinBUGS version 2.0 users manual*. MRC biostatistics unit Cambridge.
11. Tuo R, Wu CFJ. 2015 Efficient calibration for imperfect computer models. *Ann. Stat.* **43**, S1314 2331–52. (doi:10.1214/15-aos1314)
12. Plumlee M. 2017 Bayesian calibration of inexact computer models. *J. Am. Stat. Assoc.* **112**, 1274–1285. (doi:10.1080/01621459.2016.1211016)
13. Sung CL, Tuo R. 2024 A review on computer model calibration. *WIREs Comput. Stat.* **16**, e1645. (doi:10.1002/wics.1645)
14. Braverman A, Hobbs J, Teixeira J, Gunson M. 2021 Post hoc uncertainty quantification for remote sensing observing systems. *SIAM/ASA J. Uncertain. Quantif.* **9**, 1064–1093. (doi:10.1137/19m1304283)
15. Lazarus A, Husmeier D, Papamarkou T. Multiphase MCMC sampling for parameter inference in nonlinear ordinary differential equations. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 84, pp. 1252–1260, PMLR. https://proceedings.mlr.press/v84/lazarus18a.html.
16. Bower RG, Goldstein M, Vernon I. 2010 Galaxy formation: a Bayesian uncertainty analysis. *Bayesian Anal.* **5**, A524 619–70. (doi:10.1214/10-ba524)
17. McKay MD, Beckman RJ, Conover WJ. 1979 A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239. (doi:10.2307/1268522)
18. Wilkinson DJ. 2012 *Stochastic modelling for systems biology*, 2nd edn. Chapman, Hall/CRC.
19. Craig PS, Goldstein M, Seheult AH, Smith JA. 1996 Bayes linear strategies for matching hydrocarbon reservoir history. In *Bayesian statistics 5* (eds JM Bernardo, JO Berger, AP Dawid, AFM Smith), pp. 69–96. Oxford, UK: Oxford University Press. (doi:10.1093/oso/9780198523567.003.0004)
20. Pukelsheim F. 1994 The three sigma rule. *Am. Stat.* **48**, 88. (doi:10.2307/2684253)
21. Vernon I, Liu J, Goldstein M, Rowe J, Topping J, Lindsey K. 2018 Bayesian uncertainty analysis for complex systems biology models: emulation, global parameter searches and evaluation of gene functions. *BMC Syst. Biol.* **12**, 1. (doi:10.1186/s12918-017-0484-3)

22. McKinley TJ, Vernon I, Andrianakis I, McCreesh N, Oakley JE, Nsubuga RN, Goldstein M, White RG. 2018 Approximate Bayesian computation and simulation-based inference for complex stochastic epidemic models. *Stat. Sci.* **33**. (doi:10.1214/17-sts618)

23. Craig PS, Goldstein M, Rougier JC, Seheult AH. 2001 Bayesian forecasting for complex systems using computer simulators. *J. Am. Stat. Assoc.* **96**, 717–729. (doi:10.1198/016214501753168370)