Bayesian Analysis (2023) **18**, Number 1, pp. 165–191

Efficient Emulation of Computer Models Utilising Multiple Known Boundaries of Differing Dimension

Samuel E. Jackson[*] and Ian Vernon[†]

Abstract. Emulation has been successfully applied across a wide variety of scientific disciplines for efficiently analysing computationally intensive models. We develop known boundary emulation strategies which utilise the fact that, for many computer models, there exist hyperplanes in the input parameter space for which the model output can be evaluated far more efficiently, whether this be analytically or just significantly faster using a more efficient and simpler numerical solver. The information contained on these known hyperplanes, or boundaries, can be incorporated into the emulation process via analytical update, thus involving no additional computational cost. In this article, we show that such analytical updates are available for multiple boundaries of various dimensions. We subsequently demonstrate which configurations of boundaries such analytical updates are available for, in particular by presenting a set of conditions that such a set of boundaries must satisfy. We demonstrate the powerful computational advantages of the known boundary emulation techniques developed on both an illustrative low-dimensional simulated example and a scientifically relevant and high-dimensional systems biology model of hormonal crosstalk in the roots of an Arabidopsis plant.

Keywords: Bayes linear emulation, simulators, boundary conditions, systems biology.

1 Introduction

Computer models, or simulators, have been used across a wide range of disciplines to help understand the behavioural dynamics of physical systems. Such computer models are often high-dimensional, due to them possessing large numbers of input parameters, and take a substantial amount of time to evaluate. As a result, performing a full uncertainty analysis of model behaviour – a critical part of any scientific study that requires model evaluations at a vast number of parameter combinations – may be unfeasible. For this reason, emulators are frequently used as fast statistical approximations to computer model output, providing a predicted value at any input and a corresponding measure of uncertainty, given that the model has been evaluated for a set of training inputs (Sacks et al., 1989; Higdon et al., 2004; Bowman and Woods, 2016). Emulation has been successfully applied across a variety of scientific disciplines, such as astrophysics (Higdon et al., 2004; Kaufman et al., 2011; Vernon et al., 2014), climate science (Castel-

[*] Department of Mathematical Sciences, Durham University, Stockton Road, Durham, DH1 3LE, United Kingdom, samuel.e.jackson@durham.ac.uk

[†] Department of Mathematical Sciences, Durham University, Stockton Road, Durham, DH1 3LE, United Kingdom, i.r.vernon@durham.ac.uk

https://doi.org/10.1214/22-BA1304

letti et al., 2012; Williamson et al., 2013; Edwards et al., 2021), engineering (Du et al., 2021), epidemiology (Andrianakis et al., 2015; McKinley et al., 2018), and volcanology (Gu and Berger, 2016; Marshall et al., 2019). Improved emulation strategies therefore have the potential to benefit many scientific areas, allowing more accurate analysis at lower computational cost.

Vernon et al. (2019) describe an advance in emulation strategy that can lead to substantial improvements in emulator performance by exploiting the fact that there often exist parameter settings for which computer model output can be evaluated far more efficiently (whether this be analytically or just significantly faster using a simpler numerical solver). This may be possible as a result of allowing various modules to decouple from more complex parts of the model, particularly when certain parameters are set to zero. Such parameter settings commonly lie across boundaries or hyperplanes of the input parameter space, hence leading to effectively known model behaviour on these boundaries that impose constraints on the emulator itself (note that such Dirichlet boundary conditions on the emulator are distinct from the Dirichlet boundary conditions that could be imposed on the computer model itself). The information on these known boundaries can be incorporated into the emulation process via analytical update, thus involving no additional computational cost. This is preferable to the approach explored by Tan (2018), which uses substantial extra modelling and multiple extra emulator parameters (each requiring estimation) to ensure consistency with the known boundary. In contrast, the approach in Vernon et al. (2019) includes no extra modelling, and zero additional parameters, instead updating the Gaussian Process (GP) style emulator with the boundary information in a natural way.

In this article, we extend the work of the literature to show that such analytical updates are available for multiple boundaries of various dimensions. In particular, we demonstrate which configurations of boundaries such analytical updates are available for. The results of this article both provide analytical insights and are directly applicable to the analysis of many realistic physical systems represented by computer models. We demonstrate this by applying the methodology to a scientifically relevant model of hormonal crosstalk in the roots of Arabidopsis Thaliana. Due to the ease and substantial benefits of including known boundaries when emulating the Arabidopsis model, we would suggest that future Uncertainty Quantification (UQ) analyses of scientific models include a phase of identification and incorporation of known boundaries, if they are found to exist, as standard practice.

The remainder of this article is organised as follows. In Section 2, we review and extend the work of Vernon et al. (2019) to the case of a single known boundary of any dimension (as opposed to $p-1$, where p is the number of input components to the computer model). Section 3 extends the theory to multiple boundaries of various dimensions, covering which configurations of boundaries may be incorporated into an emulator analytically, before exhibiting a low-dimensional illustrative example. This example can be thoroughly investigated via associated code available at `https://github.com/samjacksonstats/KBE`. Section 4 applies the emulation techniques to a current systems biology model of Arabidopsis Thaliana, with the article being concluded in Section 5. All referenced appendices can be found in the supplementary material (Jackson and Vernon, 2022).

2 Known Boundaries of Dimension $p - k$

This section reviews the work presented in Vernon et al. (2019), whilst extending it by allowing the known boundaries to be of any dimension.

2.1 Emulation of Computer Models

We consider a computer model $f(x)$, where $x \in \mathcal{X}$ denotes a p-dimensional vector containing the computer model's input parameters, and $\mathcal{X} \subset \mathbb{R}^p$ is a pre-specified input parameter space of interest. We assume that $f(x)$ is univariate, however, the results presented directly generalise to the corresponding multivariate case, with acceptable correlation structure, as discussed further in Appendix H. We make the judgement, consistent with much of the computer model literature, that $f(x)$ has a product correlation structure:

$$\mathrm{Cov}[f(x), f(x')] \;=\; \sigma^2\, r(x - x') \;=\; \sigma^2 \prod_{j=1}^{p} r_j(x_j - x'_j), \tag{1}$$

with $r_j(0) = 1$, corresponding to deterministic $f(x)$. For example, a common choice is the Gaussian correlation function, given by

$$\mathrm{Cov}\left[f(x), f(x')\right] = \sigma^2 \exp\left(-\sum_{j=1}^{p} \left\{ \frac{x_j - x'_j}{\theta_j} \right\}^2 \right). \tag{2}$$

If we perform a set of runs at locations $X_D = \{x^{(1)}, \dots, x^{(n)}\}$ over the input space of interest $\mathcal{X}$, giving computer model outputs as the column vector $D = (f(x^{(1)}), \dots, f(x^{(n)}))^T$, then we can update our beliefs about the computer model $f(x)$ in light of D. While this can be done using Bayes theorem (if, say, $f(x)$ is assumed to be a Gaussian Process), we instead prefer a less fully specified framework. Hence, we treat expectation as primitive (motivated by de Finetti (1974)), give a partial prior belief specification only in terms of expectations, variances and covariances, and employ the Bayes linear update which is the natural choice when operating under such a partial specification (Goldstein, 1999; Goldstein and Wooff, 2007):

$$\begin{aligned}
\mathrm{E}_D[f(x)] &= \mathrm{E}[f(x)] + \mathrm{Cov}\left[f(x), D\right] \mathrm{Var}[D]^{-1}(D - \mathrm{E}[D]), &(3)\\
\mathrm{Var}_D[f(x)] &= \mathrm{Var}[f(x)] - \mathrm{Cov}\left[f(x), D\right] \mathrm{Var}[D]^{-1} \mathrm{Cov}\left[D, f(x)\right], &(4)\\
\mathrm{Cov}_D\left[f(x), f(x')\right] &= \mathrm{Cov}\left[f(x), f(x')\right] - \mathrm{Cov}\left[f(x), D\right] \mathrm{Var}[D]^{-1} \mathrm{Cov}\left[D, f(x')\right], &(5)
\end{aligned}$$

where $\mathrm{E}_D[f(x)]$, $\mathrm{Var}_D[f(x)]$ and $\mathrm{Cov}_D\left[f(x), f(x')\right]$ are the expectation, variance and covariance of $f(x)$ adjusted by D (Goldstein, 1999; Goldstein and Wooff, 2007). Although we will work within the Bayes linear formalism, the derived results would apply to a version of the fully specified probabilistic Bayesian emulation case, were one willing to make the additional assumption of full normality that use of a GP entails, and to condition on various emulator parameters. See Goldstein (1999) and Goldstein and

Wooff (2007) for a further discussion of the Bayes linear approach and its foundational motivation.

As discussed in Vernon et al. (2019), since the results of this article rely on the product correlation structure of the emulator, extension of these methods to more general emulator forms requires further calculation.

2.2 Known Boundary Emulation

Let $\mathcal{K} = \{x \in \mathbb{R}^p | x_j = \alpha_j^K, j \in J_K\}$ be a $p-k$-dimensional hyperplane defined by fixing the value of a subset of the inputs, with $J_K = \{j_1, \ldots, j_k\} \subset P = \{1, \ldots, p\}$ being the indices of the fixed inputs. We consider the situation where f is analytically solvable along $\mathcal{K}$ (that is, we know $f(x)$ for any $x \in \mathcal{K}$). We wish to update our emulator, and hence our beliefs about $f(x)$ at input point $x \in \mathcal{X}$, in light of $\mathcal{K}$. We capture model behaviour along $\mathcal{K}$ by evaluating $K = (f(y^{(1)}), \ldots, f(y^{(m)}))$ for a large but finite number m of points on $\mathcal{K}$, denoted $y^{(1)}, \ldots, y^{(m)}$, however we structure our calculations so that they can be easily generalised to the case of continuous model evaluations on $\mathcal{K}$, as shown in Vernon et al. (2019). Naively plugging these m runs into the Bayes Linear update Equations (3), (4) and (5) by replacing D with K may be infeasible due to the size of the $m \times m$ matrix inversion $\text{Var}[K]^{-1}$ (m may need to be extremely large to capture all the information available from $\mathcal{K}$). A direct update of the emulator is therefore non-trivial, hence we show from first principles that this update can be performed analytically for a wide class of emulators. This is done by exploiting a sufficiency argument briefly described in the supplementary material of Kennedy and O'Hagan (2001) though, to our knowledge, only utilised for the first time in the context of known boundary emulation in Vernon et al. (2019).

We begin by evaluating $f(x^K)$, where x^K is the orthogonal projection of x onto the boundary $\mathcal{K}$, and extending the collection of boundary evaluations, K, to be the $m+1$ column vector $K = (f(x^K), f(y^{(1)}), \ldots, f(y^{(m)}))^T$. Note that $x_j^K = \alpha_j^K$ for $j \in J_K$. Crucially, we have that

$$\text{Cov}\left[f(x^K), K\right] \text{Var}[K]^{-1} = (1, 0, \cdots, 0), \tag{6}$$

arising from the first row of the somewhat trivial equation $\text{Var}[K]\text{Var}[K]^{-1} = \boldsymbol{I}_{(m+1)}$, where $\boldsymbol{I}_{(m+1)}$ is the identity matrix of dimension $(m+1)$.

Equation (6) is of particular value when considering the behaviour of f at the point of interest x. As we have defined x^K as the orthogonal projection of x onto $\mathcal{K}$, we can define $a^K = x - x^K$ to be the p-vector of shortest distance from boundary $\mathcal{K}$ to x. Note that the elements of a^K have the property that:

$$a_j^K = \left\{ \begin{array}{rll} x_j - x_j^K & \text{if} & j \in J_K \\ 0 & \text{if} & j \in P^K = P \backslash J_K \end{array} \right. ,$$

where we define (for two sets A, B) $A \backslash B$ to be the elements in A but not B. We also define:

$$r_J(q) = \prod_{j \in J} r_j(q_j), \tag{7}$$

for a generic collection of indices J and vector of constants q. By partitioning the dimension indices $P = \{1, \ldots, p\}$ into $P = \{J_K, P^K\}$ we obtain the following covariance expressions:

$$\begin{aligned} \mathrm{Cov}[f(x), f(x^K)] \;&=\; \sigma^2\,\mathrm{r}_P(x - x^K) \;=\; \sigma^2\,\mathrm{r}_{J_K}(a^K)\,\mathrm{r}_{P^K}(0) \;=\; \sigma^2\,\mathrm{r}_{J_K}(a^K)\,, \qquad (8) \\ \mathrm{Cov}[f(x), f(y^{(s)})] \;&=\sigma^2\,\mathrm{r}_P(x - y^{(s)}) \;=\; \sigma^2\,\mathrm{r}_{J_K}(a^K)\;\mathrm{r}_{P^K}(x^K - y^{(s)}) \\ &\qquad\qquad\qquad\;=\; \mathrm{r}_{J_K}(a^K)\;\mathrm{Cov}[f(x^K), f(y^{(s)})], \qquad (9) \end{aligned}$$

since $r_j(0) = 1$, components J_K of x^K and $y^{(s)}$ must be equal as they all lie on $\mathcal{K}$ (that is, $x_j^K = y_j^{(s)}$ for $j \in J_K$), and $x_j - y_j^{(s)} = x_j^K - y_j^{(s)}$ for $j \in P^K$. From (8) and (9), the covariance between $f(x)$ and the set of boundary evaluations K is given by

$$\mathrm{Cov}\left[f(x), K\right] = \mathrm{r}_{J_K}(a^K)\;\mathrm{Cov}\left[f(x^K), K\right]. \qquad (10)$$

Using (6) and (10) we obtain the important result that:

$$\mathrm{Cov}\left[f(x), K\right]\mathrm{Var}[K]^{-1} \;=\; \mathrm{r}_{J_K}(a^K)\,(1, 0, \cdots, 0). \qquad (11)$$

As we have avoided the need to explicitly evaluate the intractable matrix inverse $\mathrm{Var}[K]^{-1}$, we can find the Bayes Linear adjusted expectation for $f(x)$ with respect to K analytically by combining (3) and (11):

$$\begin{aligned} \mathrm{E}_K[f(x)] \;&=\; \mathrm{E}[f(x)] + \mathrm{r}_{J_K}(a^K)\,(1, 0, \cdots, 0)(K - \mathrm{E}[K]) \\ &=\; \mathrm{E}[f(x)] + \mathrm{r}_{J_K}(a^K)\,\Delta f(x^K), \qquad (12) \end{aligned}$$

where we have defined $\Delta f(\cdot) = f(\cdot) - \mathrm{E}[f(\cdot)]$. We have thus eliminated the need to explicitly invert the large matrix $\mathrm{Var}[K]$ entirely by exploiting the symmetric product correlation structure and (6). Similarly, we find the adjusted covariance between $f(x)$ and $f(x')$ given the boundary $\mathcal{K}$, where $f(x')$ is the model output at a second point x', using (5) and (11), and again exploiting the partition $P = \{J_K, P^K\}$:

$$\begin{aligned} &\mathrm{Cov}_K\left[f(x), f(x')\right] \\ &\quad= \;\mathrm{Cov}\left[f(x), f(x')\right] - \mathrm{r}_{J_K}(a^K)\,(1, 0, \cdots, 0)\mathrm{Cov}\left[K, f(x')\right] \\ &\quad= \;\mathrm{Cov}\left[f(x), f(x')\right] - \mathrm{r}_{J_K}(a^K)\,\mathrm{Cov}\left[f(x^K), f(x'^K)\right]\mathrm{r}_{J_K}(a'^K) \\ &\quad= \;\sigma^2\,\mathrm{r}_{J_K}(a^K - a'^K)\,\mathrm{r}_{P^K}(x - x') - \sigma^2\,\mathrm{r}_{J_K}(a^K)\,\mathrm{r}_{J_K}(0)\,\mathrm{r}_{P^K}(x - x')\,\mathrm{r}_{J_K}(a'^K) \\ &\quad= \;\sigma^2\,R_{J_K}(a^K, a'^K)\;\mathrm{r}_{P^K}(x - x'), \qquad (13) \end{aligned}$$

where the ‘updated correlation component’ in the x_{J_K} directions is given as

$$R_{J_K}(a^K, a'^K) \;=\; \mathrm{r}_{J_K}(a^K - a'^K) - \mathrm{r}_{J_K}(a^K)\,\mathrm{r}_{J_K}(a'^K)\,. \qquad (14)$$

By setting $x = x'$, we obtain an expression for the adjusted variance of $f(x)$:

$$\mathrm{Var}_K[f(x)] = \sigma^2\,(1 - \mathrm{r}_{J_K}(a^K)^{\,2}). \qquad (15)$$

Equations (12) and (15) give the expectation and variance of the emulator at a point x, updated by a known boundary $\mathcal{K}$. As they require only evaluations of the analytic boundary function and the correlation function they can be implemented with trivial computational cost in comparison to a direct update by K. Useful insights into the sufficiency, stationarity and limiting behaviour, along with a generalisation of the above to continuous boundary evaluations K, are discussed in Vernon et al. (2019). Note that when using a general product correlation structure, as given by (1), we require the known boundary $\mathcal{K}$ (and all subsequent known boundaries) to be axially aligned. However, if the Gaussian/squared exponential correlation structure is used which is invariant under rotations in $\mathbb{R}^p$, the initial boundary $\mathcal{K}$ can have any orientation, and subsequent boundaries (see Section 3) can be included that are parallel to $\mathcal{K}$ or perpendicular to $\mathcal{K}$ (in a basis where all correlation lengths are equal).

2.3 Three-dimensional Example

For illustration, we consider the problem of emulating the three-dimensional function:

$$f(x) = \sin\left(\frac{x_1}{\exp(x_2)}\right) + \cos(x_3), \tag{16}$$

over an input domain of interest given by $[-2\pi, 2\pi] \times [-\pi/4, \pi/4] \times [-2\pi, 2\pi]$. This simulated example will, throughout this article, take a prior expectation $\mathrm{E}[f(x)] = 0$, and a product Gaussian covariance structure, as given by (2), with correlation length parameters $\theta = (\pi, \pi/8, \pi)$ and variance parameter $\sigma^2 = 2$. These values are adequate for the example presented, as illustrated in the diagnostic panels of the figures that follow. Having said this, it is important and informative to explore the effect of varying the correlation length parameters on emulator predictions, particularly in combination with known boundaries. We explore varying these parameters for the Arabidopsis model application in Section 4.

We begin by assuming a known boundary $\mathcal{K} = \{x \in \mathbb{R}^3 | (x_2, x_3) = (0, 0)\}$, hence that we can evaluate $f(x^K) = \sin(x_1) + 1$ for any point on the boundary $x^K \in \mathcal{K}$. We hence apply the emulator expectation and variance update given by (12) and (15).

In order to illustrate the effect of the known boundary on the emulator, we examine emulator behaviour across two-dimensional slices (keeping one variable fixed) of the three-dimensional input space, as shown in Figure 1. The top row depicts the input space as a cube, with the one-dimensional boundary being illustrated by the red line. The green planes are two-dimensional slices of the input space over which emulator and simulator behaviour are compared in the remaining plots. The remaining rows show (from top to bottom) simulator $f(x)$ (for comparison purposes), emulator expectation $\mu(x) = \mathrm{E}_K[f(x)]$, emulator variance $\nu(x) = \mathrm{Var}_K[f(x)]$ and standardised diagnostic values $s(x) = (\mathrm{E}_K[f(x)] - f(x))/\sqrt{\mathrm{Var}_K[f(x)]}$. In each case, the variable with smaller index is along the horizontal axis. The left column of the figure shows the results for $x_2 = 0$. Since this slice contains the known boundary, we see that for $x_3 = 0$ emulator expectation precisely matches the true simulator function, and the variance goes to zero. As we move further away from the boundary in the x_3 direction, the variance increases. Note that, since $\mathcal{K}$ is parallel to the x_1 direction, altering the value of this variable doesn't alter the emulator variance. The middle column shows a slice away

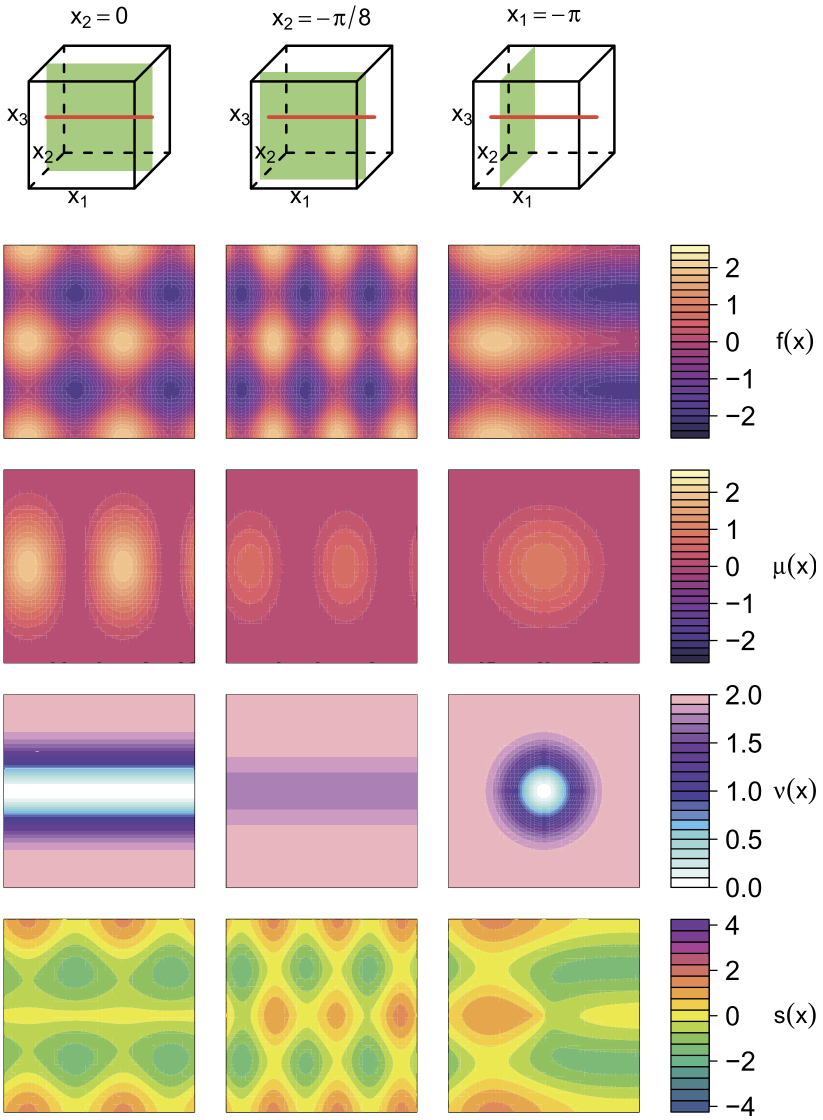

Figure 1: Updating the emulator for the three-dimensional function given by (16) by a single boundary $\mathcal{K}$. Rows from top to bottom, show: 1) position of known boundary (red line) in the three-dimensional input space, along with the position of the two-dimensional slices (green planes) illustrating the two-dimensional plane in the three-dimensional input space, over which the remaining plots in the same column are plotted, 2) simulator function $f(x)$, 3) emulator expectation $\mu(x)$, 4) emulator variance $\nu(x)$, 5) standardised errors $s(x)$. Columns from left to right show results on the three planes $x_2 = 0$, $x_2 = -\pi/8$ and $x_1 = -\pi$ respectively, shown as the green planes in the top row. Note that for each two dimensional plot, the variable with smaller index is along the horizontal axis.

from the boundary ($x_2 = -\pi/8$). Again, the smallest variance is at $x_3 = 0$, however, now it is not zero. The right column shows $x_1 = -\pi$. In this case, the function is only known at the centre point $(x_2, x_3) = (0, 0)$ with variance increasing radially away from this point. The diagnostic plots provide evidence for the validity of the emulator, with few parts of the input space having standardised errors greater than 2. Associated code for the examples contained in Sections 2 and 3 can be found at `https://github.com/samjacksonstats/KBE`. This code permits user-friendly investigation into varying all features of the presented example, including toy function, parameter ranges, known boundaries and diagnostic slices.

2.4 Updating by Further Model Evaluations

Since we have analytic expressions for $\mathrm{E}_K[f(x)]$, $\mathrm{Var}_K[f(x)]$ and $\mathrm{Cov}_K\left[f(x), f(x')\right]$, we are now able to include additional computer model evaluations into the emulation process. To do this, we perform n (expensive) evaluations of the computer model across $\mathcal{X}$ to obtain $D = (f(x^{(1)}), \ldots, f(x^{(n)}))$, and use these to supplement the evaluations, K, available on the boundary. We want to update the emulator by the union of the evaluations D and K, that is to find $\mathrm{E}_{K\cup D}[f(x)]$, $\mathrm{Var}_{K\cup D}[f(x)]$ and $\mathrm{Cov}_{K\cup D}\left[f(x), f(x')\right]$. This can be achieved via a sequential Bayes Linear update:

$$\mathrm{E}_{K\cup D}[f(x)] = \mathrm{E}_K[f(x)] + \mathrm{Cov}_K\left[f(x), D\right]\mathrm{Var}_K[D]^{-1}(D - \mathrm{E}_K[D]), \tag{17}$$

$$\mathrm{Var}_{K\cup D}[f(x)] = \mathrm{Var}_K[f(x)] - \mathrm{Cov}_K\left[f(x), D\right]\mathrm{Var}_K[D]^{-1}\mathrm{Cov}_K\left[D, f(x)\right], \tag{18}$$

$$\mathrm{Cov}_{K\cup D}\left[f(x), f(x')\right] = \mathrm{Cov}_K\left[f(x), f(x')\right] - \mathrm{Cov}_K\left[f(x), D\right]\mathrm{Var}_K[D]^{-1}\mathrm{Cov}_K\left[D, f(x')\right], \tag{19}$$

where we first update our emulator analytically by K, and subsequently update these quantities by the evaluations D (Goldstein and Wooff, 2007). n is typically of small/modest size due to the relative expense of evaluating the computer model, hence these calculations (in particular $\mathrm{Var}_K[D]^{-1}$) will remain tractable, leading to an overall $\mathcal{O}(n^3)$ calculation. It is worth comparing this to the brute force approach of including a large number m of points on the $(p-k)$-dimensional known boundary $\mathcal{K}$, and updating the emulator directly, which leads to an $\mathcal{O}((m+n)^3)$ operation. For example, we may consider a fine grid of points that are half a correlation length $(\theta/2)$ apart in each direction to be enough to capture most of the information from the boundary $\mathcal{K}$. In this case we would have $\mathcal{O}((2/\theta)^{(p-k)} + n)^3)$ assuming $\mathcal{X} = [0, 1]^p$, which if $\theta = 1/4$, say, is $\mathcal{O}((8^{(p-k)} + n)^3)$. For the Arabidopsis application, $p = 38$ and $k = 4$. This represents a vastly less efficient calculation compared to the above analytic $\mathcal{O}(n^3)$ approach.

It is worth noting that users of standard black-box GP emulation packages may be unable to implement directly the formulae of (12) and (15). However, we see that due to underlying sufficiency arguments, such a user can simply add the $(n+1)$ trivial boundary evaluations $f(x^K)$ and $D^K = (f(x^{(1)K}), \ldots, f(x^{(n)K}))$ to D to give $D^* = \{D, D^K, f(x^K)\}$, and then their black box Gaussian process package will produce results that precisely match (17)-(19). This would only require the inversion of a $(2n+1) \times (2n+1)$ matrix, and hence be of $\mathcal{O}((2n+1)^3)$. However, this is only true for a single emulated point, while typically a user would wish to emulate at a large number n' of input locations. Naively adding all corresponding n' boundary evaluations directly to D^* would lead to an unnecessary $\mathcal{O}((n'+2n)^3)$ calculation with $n' \gg n$. However, splitting the n' points into N' batches improves calculation efficiency to $\mathcal{O}(N'(\frac{n'}{N'}+2n)^3)$, which has a computationally optimal batch size of n, thus an optimal number of batches $N' = n'/n$. This results in the calculation being $\mathcal{O}(N'(\frac{n'}{N'}+2n)^3) = \mathcal{O}(3^3 n' n^2) = \mathcal{O}(n'n^2)$, where again $n' \gg n$. As typical values may be say $n = 10^2$ and $n' = 10^6$, whereby $n' = n^3$, were this relationship to scale, we would have $\mathcal{O}(n^5)$. In comparison, the above proposed analytic method ((17)-(19) combined with (12), (13), and (15)) is of $\mathcal{O}(n^3)$ regardless of the size of n', and hence is seen to be clearly superior. For further discussion of this issue and its exacerbation for multiple known boundaries, see Appendix G.

3 Multiple Boundaries of Various Dimensions

In this section, we begin by discussing the requirements for being able to analytically update by a second known boundary, before considering larger numbers of boundaries.

3.1 Two Known Boundaries

Given the results of Section 2, we now proceed to consider analytical updating by a second known boundary $\mathcal{L} = \{x \in \mathbb{R}^p | x_j = \alpha_j^L, j \in J_L\}$ of dimension $p - l$. As this section progresses, we will restrict the form of $\mathcal{L}$, for example, to being either intersecting and orthogonal, or parallel to $\mathcal{K}$, however, for now we consider the general case. We define $L = \left(f(x^L), f(z^{(1)}), \ldots, f(z^{(m)})\right)^T$ to be a vector of model evaluations, where $z^{(1)}, \cdots, z^{(m)}$ constitute a large but finite number m of points along $\mathcal{L}$, and denote the p-vector of shortest distance to x from its orthogonal projection x^L as $a^L = x - x^L$. We also define x^{LK} to be the sequential orthogonal projection of x first onto $\mathcal{L}$ and then onto $\mathcal{K}$, and correspondingly $a^{LK} = x - x^{LK}$ to be the p-vector of shortest distance to x from this sequential projection. In Vernon et al. (2019), it was demonstrated that analytic updating of two perpendicular or parallel $p-1$-dimensional boundaries $\mathcal{K}, \mathcal{L}$ could be achieved. As an example, the update for two perpendicular $p-1$-boundaries was given by:

$$\mathrm{E}_{K\cup L}[f(x)] = \mathrm{E}[f(x)] + \mathrm{r}_1(a^K)\,\Delta f(x^K) + \mathrm{r}_2(a^L)\,\Delta f(x^L) - \mathrm{r}_1(a^K)\,\mathrm{r}_2(a^L)\,\Delta f(x^{LK}), \tag{20}$$

$$\mathrm{Cov}_{K\cup L}\left[f(x), f(x')\right] = \sigma^2\, R_1(a^K, a'^K)\; R_2(a^L, a'^L)\; \prod_{j=3}^{p} \mathrm{r}_j(x - x'), \tag{21}$$

where it is assumed that $\mathcal{K}$ and $\mathcal{L}$ are defined by $x_1 = \alpha^K$ and $x_2 = \alpha^L$, and we have utilised the notation of (7) and (14). The inclusion-exclusion nature of this result will also feature in the general results that follow, both in this section and in Section 3.2.

More generally, to permit analytic updating by a second boundary $\mathcal{L}$ of dimension $p - l$, as defined above, we need to find an analogous version of (10), which relates $\mathrm{Cov}_K\left[f(x), L\right]$ to $\mathrm{Cov}_K\left[f(x^L), L\right]$. We begin by examining the expression for

$\mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right]$ in light of (13), exploiting the notation of (7) and (14), and using the partition $P^K = \{J_L \backslash J_K, P^{K\cup L}\}$ where $P^{K\cup L} = P\backslash(J_K \cup J_L)$:

$$\begin{aligned}
&\mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right] \\
&\quad= \sigma^2 R_{J_K}(x^L - x^{LK}, z^{(s)} - z^{(s)K})\,\mathrm{r}_{P^K}(x^L - z^{(s)}) \\
&\quad= \sigma^2\,\mathrm{r}_{P^{K\cup L}}(x^L - z^{(s)})\,\mathrm{r}_{J_L\backslash J_K}(x^L - z^{(s)}) \\
&\qquad\qquad \times \left(\mathrm{r}_{J_K}(x^L - z^{(s)}) - \mathrm{r}_{J_K}(x^L - x^{LK})\,\mathrm{r}_{J_K}(z^{(s)} - z^{(s)K})\right) \\
&\quad= \sigma^2\,\mathrm{r}_{P^{K\cup L}}(x - z^{(s)}) \\
&\qquad\qquad \times \left(\mathrm{r}_{J_K\backslash J_L}(x - z^{(s)}) - \mathrm{r}_{J_K\backslash J_L}(a^K)\,\mathrm{r}_{J_K\backslash J_L}(z^{(s)} - z^{(s)K})\,\mathrm{r}_{J_K\cap J_L}(LK)^{\,2}\right),
\end{aligned} \tag{22}$$

since $x_j^L - z_j^{(s)} = 0$ for $j \in J_L$, $\mathrm{r}_{P^{K\cup L}}(x_j^L - z_j^{(s)}) = \mathrm{r}_{P^{K\cup L}}(x_j - z_j^{(s)})$ and $x_j^L - x_j^{LK} = LK_j$ for $j \in J_L$, where LK_j is a constant giving the orthogonal distance from $\mathcal{L}$ to $\mathcal{K}$ in the x_j-direction. In addition, note that we define throughout $r_\emptyset(\cdot) = 1$. We then examine:

$$\begin{aligned}
&\mathrm{Cov}_K\left[f(x), f(z^{(s)})\right] \\
&\quad= \sigma^2 R_{J_K}(x - x^K, z^{(s)} - z^{(s)K})\,\mathrm{r}_{P^K}(x - z^{(s)}) \\
&\quad= \sigma^2\,\mathrm{r}_{P^{K\cup L}}(x - z^{(s)})\,\mathrm{r}_{J_L\backslash J_K}(x - z^{(s)})\left(\mathrm{r}_{J_K}(x - z^{(s)}) - \mathrm{r}_{J_K}(x - x^K)\,\mathrm{r}_{J_K}(z^{(s)} - z^{(s)K})\right) \\
&\quad= \sigma^2\,\mathrm{r}_{P^{K\cup L}}(x - z^{(s)})\,\mathrm{r}_{J_L\backslash J_K}(a^L)\Big(\mathrm{r}_{J_K\backslash J_L}(x - z^{(s)})\,\mathrm{r}_{J_K\cap J_L}(a^L) \\
&\qquad\qquad - \mathrm{r}_{J_K\backslash J_L}(a^K)\,\mathrm{r}_{J_K\cap J_L}(a^K)\,\mathrm{r}_{J_K\backslash J_L}(z^{(s)} - z^{(s)K})\,\mathrm{r}_{J_K\cap J_L}(LK)\Big).
\end{aligned} \tag{23}$$

By combining (22) and (23) we obtain:

$$\begin{aligned}
&\mathrm{Cov}_K\left[f(x), f(z^{(s)})\right] = \\
&\frac{\mathrm{r}_{J_K\backslash J_L}(x - z^{(s)})\,\mathrm{r}_{J_K\cap J_L}(a^L) - \mathrm{r}_{J_K\backslash J_L}(a^K)\,\mathrm{r}_{J_K\cap J_L}(a^K)\,\mathrm{r}_{J_K\backslash J_L}(z^{(s)} - z^{(s)K})\,\mathrm{r}_{J_K\cap J_L}(LK)}{\mathrm{r}_{J_K\backslash J_L}(x - z^{(s)}) - \mathrm{r}_{J_K\backslash J_L}(a^K)\,\mathrm{r}_{J_K\backslash J_L}(z^{(s)} - z^{(s)K})\,\mathrm{r}_{J_K\cap J_L}(LK)^2} \\
&\qquad \times \mathrm{r}_{J_L\backslash J_K}(a^L)\,\mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right].
\end{aligned} \tag{24}$$

In order to obtain an equation analogous to (10) we need to be able to write $\mathrm{Cov}_K\left[f(x), f(z^{(s)})\right]$ as a product of $\mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right]$ and a function that does not depend on $z^{(s)}$, thus permitting replacement of $f(z^{(s)})$ by L in the $\mathrm{Cov}_K\left[\,\cdot\,, f(z^{(s)})\right]$ terms. In general, this is not possible for a second boundary, since the required product correlation structure no longer exists, thus resulting in the appearance of $z^{(s)}$ several times in the quotient in the expression on the right hand side of (24). However, there are two general and commonly occurring cases when this dependency does not exist and our methods permit further analytic update by a second known boundary (and indeed further known boundaries, as discussed in Sections 3.2 and 3.3). The first case is if we wish to update by a known boundary which is a hyperplane which is orthogonal to and intersecting the first, and the second case is if we wish to update by a known boundary that is a hyperplane which is parallel to the first, or a subplane thereof. We discuss these two cases in the following sections.

Two Intersecting Orthogonal Known Boundaries

If $\mathcal{K}$ and $\mathcal{L}$ are two intersecting orthogonal boundaries, that is $\mathcal{K} \cap \mathcal{L} \neq \emptyset$, then $\alpha_j^K = \alpha_j^L, a_j^K = a_j^L$ and $LK_j = 0$ for $j \in J_K \cap J_L$. In this case we can rewrite (24) as:

$$\begin{aligned}
&\mathrm{Cov}_K \left[f(x), f(z^{(s)})\right] \\
&\quad = \frac{\mathrm{r}_{J_K \setminus J_L}(x - z^{(s)})\, \mathrm{r}_{J_K \cap J_L}(a^L) \;-\; \mathrm{r}_{J_K \setminus J_L}(a^K)\, \mathrm{r}_{J_K \cap J_L}(a^L)\, \mathrm{r}_{J_K \setminus J_L}(z^{(s)} - z^{(s)K})\, \mathrm{r}_{J_K \cap J_L}(0)}{\mathrm{r}_{J_K \setminus J_L}(x - z^{(s)}) \;-\mathrm{r}_{J_K \setminus J_L}(a^K)\, \mathrm{r}_{J_K \setminus J_L}(z^{(s)} - z^{(s)K})\, \mathrm{r}_{J_K \cap J_L}(0)^2} \\
&\qquad\qquad \times \mathrm{r}_{J_L \setminus J_K}(a^L)\, \mathrm{Cov}_K \left[f(x^L), f(z^{(s)})\right] \\
&\quad = \mathrm{r}_{J_K \cap J_L}(a^L)\, \mathrm{r}_{J_L \setminus J_K}(a^L)\, \mathrm{Cov}_K \left[f(x^L), f(z^{(s)})\right] \\
&\quad = \mathrm{r}_{J_L}(a^L)\, \mathrm{Cov}_K \left[f(x^L), f(z^{(s)})\right], \qquad (25)
\end{aligned}$$

so that

$$\mathrm{Cov}_K \left[f(x), L\right] = \mathrm{r}_{J_L}(a^L)\, \mathrm{Cov}_K \left[f(x^L), L\right]. \qquad (26)$$

We can now avoid explicit evaluation of the intractable $\mathrm{Var}_K[L]^{-1}$ term by combining (26) with sequential update (17)-(19) to give:

$$\mathrm{E}_{K \cup L}[f(x)] = \mathrm{E}[f(x)] + \mathrm{r}_{J_K}(a^K)\, \Delta f(x^K) + \mathrm{r}_{J_L}(a^L)\, \Delta f(x^L) - \mathrm{r}_{J_K \cup J_L}(a^{LK})\, \Delta f(x^{LK}), \quad (27)$$

$$\mathrm{Cov}_{K \cup L} \left[f(x), f(x')\right] = \sigma^2\, \mathrm{r}_{P K \cup L}(x - x')\, R_{K,L}(x, x'), \qquad (28)$$

$$\text{with} \qquad R_{K,L}(x, x') = \sum_{i=0}^{2} (-1)^i \sum_{T \subseteq \{K,L\},\, |T| = i} \mathrm{r}_{(J_K \cup J_L) \setminus J_T}(x - x')\, \mathrm{r}_{J_T}(a^{LK})\, \mathrm{r}_{J_T}(a'^{LK}),$$

where $J_T = \bigcup_{t \in T} J_t$. Extended derivation of the general Expressions (27) and (28) for updating by any two intersecting orthogonal boundaries can be found in Appendix A. Note that if $\mathcal{K}$ is defined by $x_1 = \alpha^K$, and $\mathcal{L}$ is defined by $x_2 = \alpha^L$, these expressions collapse back to those given by (21). We can see that Expressions (27) and (28) are invariant under the interchange of the two boundaries. This should be as expected, since the boundaries are orthogonal to and intersecting each other.

Two Parallel Boundaries

Consider now that $\mathcal{L}$ is such that $J_K \subseteq J_L$. In other words, $\mathcal{L}$ is either a hyperplane which is parallel to $\mathcal{K}$, or a subplane thereof. Note that now $x^L = x^{KL} \neq x^{LK}$, that is, the order of the boundaries matters, and that $x^{LK} \neq x^K$ in general (unless $J_K = J_L$). We also define LK to be the p-vector of shortest distance from (any point on) $\mathcal{L}$ to $\mathcal{K}$. In this case, (24) can be rewritten as

$$\begin{aligned}
&\mathrm{Cov}_K \left[f(x), f(z^{(s)})\right] \\
&\quad = \frac{\mathrm{r}_{J_K}(a^L) \;-\mathrm{r}_{J_K}(a^K)\, \mathrm{r}_{J_K}(LK)}{1 - \mathrm{r}_{J_K}(LK)}\, \mathrm{r}_{J_L \setminus J_K}(a^L)\, \mathrm{Cov}_K \left[f(x^L), f(z^{(s)})\right] \\
&\quad = \frac{R_{J_K}(a^K, LK)}{R_{J_K}(LK, LK)}\, \mathrm{r}_{J_L \setminus J_K}(a^L)\, \mathrm{Cov}_K \left[f(x^L), f(z^{(s)})\right], \qquad (29)
\end{aligned}$$

hence we have that:

$$\mathrm{Cov}_K\left[f(x), L\right] = \frac{R_{J_K}(a^K, LK)}{R_{J_K}(LK, LK)} \mathrm{r}_{J_L \setminus J_K}(a^L)\,\mathrm{Cov}_K\left[f(x^L), L\right]. \tag{30}$$

Equation (30) allows us to again avoid explicit evaluation of the intractable $\mathrm{Var}_K[L]^{-1}$ term. The adjusted expectation and covariance can then be calculated using the sequential update (17)-(19), to be:

$$\begin{aligned}
&\mathrm{E}_{K\cup L}[f(x)] \\
&= \mathrm{E}_K[f(x)] + \frac{R_{J_K}(a^K, LK)}{R_{J_K}(LK, LK)} \mathrm{r}_{J_L\setminus J_K}(a^L)\,(f(x^L) - \mathrm{E}_K[f(x^L)]) \\
&= \mathrm{E}[f(x)] + \mathrm{r}_{J_K}(a^K)\,\Delta f(x^K) + \frac{R_{J_K}(a^K, LK)}{R_{J_K}(LK, LK)} \mathrm{r}_{J_L\setminus J_K}(a^L)\,\Delta f(x^L) \\
&\quad - \frac{R_{J_K}(a^K, LK)}{R_{J_K}(LK, LK)} \mathrm{r}_{J_L\setminus J_K}(a^L)\,\mathrm{r}_{J_K}(KL)\,\Delta f(x^{LK}),
\end{aligned} \tag{31}$$

$$\mathrm{Cov}_{K\cup L}\left[f(x), f(x')\right] \quad = \quad \sigma^2\ \mathrm{r}_{P^{K\cup L}}(x - x')\,R^{(2)}_{K,L}(x, x'), \tag{32}$$

where we define:

$$\begin{aligned}
&R^{(2)}_{K,L}(x, x') \\
&= \ R_{J_K}(a^K, a'^K)\,\mathrm{r}_{J_L\setminus J_K}(x - x') \ - \ \frac{R_{J_K}(a^K, LK)\,R_{J_K}(LK, a')}{R_{J_K}(LK, LK)} \mathrm{r}_{J_L\setminus J_K}(a^L)\,\mathrm{r}_{J_L\setminus J_K}(a'^L).
\end{aligned}$$

Extended derivation of Expressions (31) and (32) can be found in Appendix B.

We observe that, for the case when $J_K \subset J_L$, the result is not invariant under the interchange of the two boundaries $\mathcal{K} \leftrightarrow \mathcal{L}$, as expected. Although the order in which we update by the two boundaries should not affect the final result, whilst we were able to provide the analytical solution above for the case where we updated by the boundary of largest dimension first, this is not the case if we first update by the boundary of lower dimension. As discussed in Section 3.1, a problem arises in the latter case due to us being unable to write $\mathrm{Cov}_K\left[f(x), f(z^{(s)})\right]$ as a product of $\mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right]$ and a function of x only. Therefore, we cannot obtain an expression analogous to (11) which enables analytic updating of $f(x)$ by $\mathcal{K}$ and $\mathcal{L}$ by avoiding the explicit inversion of $\mathrm{Var}_K[L]^{-1}$. In the case when $J_K = J_L$, the result is invariant under $\mathcal{K} \leftrightarrow \mathcal{L}$, as shown in Appendix C. In addition, if $\mathcal{K}$ is given by $x_1 = \alpha^K$ and $\mathcal{L}$ is given by $x_1 = \alpha^L$, Expressions (31) and (32) collapse to the result in Vernon et al. (2019).

3.2 Multiple Known Boundaries

Following Section 3.1, it is logical to assume that analytic updating would be possible for further intersecting orthogonal and parallel hyperplanes along which model behaviour is known. We hence proceed to discuss the form of an emulator updated by h boundaries $\mathcal{K}_H = \mathcal{K}_1, \ldots, \mathcal{K}_h$, where boundary $\mathcal{K}_i = \{x \in \mathbb{R}^p | x_j^{K_i} = \alpha_j^{K_i}, j \in J_{K_i}\}$ is of dimension $p - k_i$. In this section, we first consider intersecting orthogonal boundaries, then we consider parallel boundaries. Similar to the previous sections, we define $a^{K_i} = x - x^{K_i}$ to be the vector of shortest distance from boundary $\mathcal{K}_i$ to x, where x^{K_i} is the orthogonal projection of x onto boundary $\mathcal{K}_i$. Letting $T = (t_1, \ldots, t_\tau)$ be a sequence of boundary indices, we also define $\mathcal{K}_T$ to be the sequence of boundaries $\mathcal{K}_{t_1}, \ldots, \mathcal{K}_{t_\tau}$, x^{K_T} to denote x projected sequentially onto boundaries $\mathcal{K}_T$ in reverse order (that is, $\mathcal{K}_{t_\tau}, \mathcal{K}_{t_{\tau-1}}$, etc.), and $a^{K_T} = x - x^{K_T}$.

Multiple Intersecting Orthogonal Boundaries

In this section, we consider that the h boundaries are all orthogonal to and intersecting each other, in other words that $\mathcal{K}_1 \cap \cdots \cap \mathcal{K}_h \neq \emptyset$.

Theorem 3.2.1. *The expectation and covariance of $f(x)$ sequentially adjusted by boundaries $\mathcal{K}_H$, such that $\mathcal{K}_1 \cap \cdots \cap \mathcal{K}_h \neq \emptyset$, are given by:*

$$\mathrm{E}_{K_H}[f(x)] = \mathrm{E}[f(x)] + \sum_{i=1}^{h} (-1)^{i+1} \sum_{T \subseteq H,\, |T|=i} \mathrm{r}_{J_T}(a^{K_T})\, \Delta f(x^{K_T}), \quad (33)$$

$$\mathrm{Cov}_{K_H}[f(x), f(x')] = \sigma^2 R_{K_H}(x, x')\, \mathrm{r}_{P^H}(x - x'), \quad (34)$$

$$\textit{with} \quad R_{K_H}(x, x') = \sum_{i=0}^{h} (-1)^i \sum_{T \subseteq H,\, |T|=i} \mathrm{r}_{J_H \setminus J_T}(x - x')\, \mathrm{r}_{J_T}(a^{K_H})\, \mathrm{r}_{J_T}(a'^{K_H}),$$

where $H = 1, \ldots, h$, $J_T = \bigcup_{t \in T} J_{K_t}$, $J_H = \bigcup_{i \in H} J_{K_i}$ and $P^H = P \backslash J_H$.

Theorem 3.2.1 provides the general form for analytically updating our emulator by multiple intersecting orthogonal boundaries. We can see that Expressions (33) and (34) are invariant under the interchange of the h boundaries. This should be as expected, since all boundaries are orthogonal to and intersecting each other. Proof of Theorem 3.2.1 by induction is presented in Appendix D.

Multiple Parallel Boundaries

In this section, we consider that the h boundaries are such that $J_{K_{i-1}} \subseteq J_{K_i}$. In other words, for all $i \geq 2$, $\mathcal{K}_i$ is either a hyperplane which is parallel to K_{i-1}, or a subplane thereof. Such ordering of the boundaries by decreasing dimension size is required in order to leave the correlation structure in the appropriate product form to perform all the calculations analytically at each stage (see the discussion in the main part of Section 3.1 and at the end of Section 3.1 for more detail).

Theorem 3.2.2. *The expectation and covariance of $f(x)$ adjusted by $h \geq 2$ parallel boundaries $\mathcal{K}_H$, with $J_{K_{i-1}} \subseteq J_{K_i}$ for $i = 2, \ldots, h$, are given by:*

$$
\begin{aligned}
&\mathrm{E}_{K_H}[f(x)] \qquad (35)\\
&= \mathrm{E}[f(x)] + \mathrm{r}_{J_1}(a^{K_1})\,\Delta f(x^{K_1}) + \sum_{\gamma=2}^{h} \frac{R^{(\gamma-1)}_{K_{\Gamma-1}}(x, K_\gamma)}{R^{(\gamma-1)}_{K_{\Gamma-1}}(K_\gamma, K_\gamma)} \mathrm{r}_{J_\gamma \setminus J_{\Gamma-1}}(a^{K_\gamma}) \Bigg(\Delta f(x^{K_\gamma}) + \\
&\sum_{i=2}^{\gamma} \sum_{b \subset \Gamma,\, b_1 < \ldots < b_i = \gamma} (-1)^{i+1} \prod_{l=1}^{i-1} \frac{R^{(b_l-1)}_{K_{B_l-1}}(K_\gamma, K_{b_l})}{R^{(b_l-1)}_{K_{B_l-1}}(K_{b_l}, K_{b_l})} \mathrm{r}_{J_{b_l} \setminus J_{B_l-1}}(K_{b_{l+1}} K_{b_l})\,\Delta f(x^{K_b}) \Bigg),
\end{aligned}
$$

$$
\mathrm{Cov}_{K_H}\left[f(x), f(x')\right] \;=\; \sigma^2\, \mathrm{r}_{PH}(x - x')\, R^{(h)}_{K_H}(x, x'), \qquad (36)
$$

where $\Gamma - 1 = 1, \ldots, \gamma - 1$*,* $B_l = 1, \ldots, b_l$*,* $B_l - 1 = 1, \ldots, b_l - 1$*,* $r_\emptyset(\cdot) = 1$*,* $K_{i_1} K_{i_2}$ *is the p-vector of shortest distance from* $\mathcal{K}_{i_1}$ *to* $\mathcal{K}_{i_2}$*, and* $R^{(\gamma)}$ *is defined recursively by:*

$$
\begin{aligned}
R^{(\gamma)}_{K_\Gamma}(x, x') \;=\; &\Bigg(R^{(\gamma-1)}_{K_{\Gamma-1}}(x, x')\, \mathrm{r}_{J_\gamma \setminus J_{\Gamma-1}}(x - x') \\
&- \frac{R^{(\gamma-1)}_{K_{\Gamma-1}}(x, K_\gamma)\, R^{(\gamma-1)}_{K_{\Gamma-1}}(x', K_\gamma)}{R^{(\gamma-1)}_{K_{\Gamma-1}}(K_\gamma, K_\gamma)} \mathrm{r}_{J_\gamma \setminus J_{\Gamma-1}}(a^{K_\gamma})\, \mathrm{r}_{J_\gamma \setminus J_{\Gamma-1}}(a'^{K_\gamma}) \Bigg),
\end{aligned}
$$

with $R^{(0)} = 1$*, and defining* $R^{(\gamma)}_{K_\Gamma}(x, K_i) = R^{(\gamma)}_{K_\Gamma}(x, y)$*,* $R^{(\gamma)}_{K_\Gamma}(K_i, K_{i'}) = R^{(\gamma)}_{K_\Gamma}(y, y')$ *for any points* $y \in \mathcal{K}_i, y' \in \mathcal{K}_{i'}$ *respectively.*

Note that for a single boundary $R^{(1)}_K(x, x') = R_{J_K}(a^K, a'^K)$. Theorem 3.2.2, proved by induction in Appendix E, provides the general formulae for analytically updating our emulator by multiple parallel boundaries. Expressions (35) and (36) are not invariant under interchange of the h boundaries due to the need for the boundaries to be taken in order of decreasing dimension size in order for the calculations to be performed analytically.

3.3 Additional Sets of Known Boundaries

Section 3.2 demonstrated that analytic update calculations are possible given a set of mutually orthogonal known boundaries or for sets of known boundaries where each boundary is a hyperplane which is parallel to the previous one, or a subset thereof. Given these results, the natural question to ask is: for which combinations of known boundaries can an emulator be updated, whilst allowing all of the necessary calculations to be performed analytically? We now state the following proposition to answer this general question.

Proposition 3.3.1. *Given that beliefs about model output have been analytically updated given information on a sequence of boundaries* $\mathcal{K}_{H-1} = \mathcal{K}_1, \ldots, \mathcal{K}_{h-1}$*, with* $\mathcal{K}_i = \{x \in \mathbb{R}^p | x_j^{K_i} = \alpha_j^{K_i}, j \in J_{K_i}\}$ *, we can update by a further boundary* $\mathcal{K}_h$ *if and only if, for each* $i = 1, \ldots, h-1$*, either* $\mathcal{K}_h \cap \mathcal{K}_i \neq \emptyset$ *or* $J_{K_i} \subseteq J_h$*.*

In other words, for each $i = 1, \ldots, h-1$, $\mathcal{K}_h$ must either be an intersecting and orthogonal hyperplane to $\mathcal{K}_i$, or be a hyperplane (or subplane thereof) which is parallel to $\mathcal{K}_i$.

As discussed in Section 3.1, in order to perform an analytical update by a further known boundary, we needed to be able to write $\mathrm{Cov}_K\left[f(x), f(z^{(s)})\right]$ as a product of $\mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right]$ and a function that does not depend on $z^{(s)}$. In other words, we needed an appropriate product correlation structure. This same criterion extends to further known boundaries, and must hold between every pair of a set of boundaries for analytic update to be performed. The general formula is somewhat complex, although the analytic update formulae can be derived by iteratively applying the sequential update (17)-(19) with appropriate analogous equations to (10).

3.4 Three-Dimensional Example

We continue the example of Section 2.3 by adding two extra boundaries. We add a one-dimensional boundary $\mathcal{L} = \{x \in \mathbb{R}^3 | (x_2, x_3) = (0, -\pi)\}$ which is parallel to the first, and also a two-dimensional boundary $\mathcal{M} = \{x \in \mathbb{R}^3 | x_1 = 0\}$, which is orthogonal to the others. $\mathcal{K}, \mathcal{L}$ and $\mathcal{M}$ therefore form a set of known boundaries satisfying the conditions given in the propostion in Section 3.3. The update formulae for this particular set of boundaries can be given as:

$$
\begin{aligned}
&\mathrm{E}_{K\cup L\cup M}[f(x)] \\
&\quad= \mathrm{E}[f(x)] + \mathrm{r}_1(a^M)\,\Delta f(x^M) + \mathrm{r}_{\{2,3\}}(a^K)\left(\Delta f(x^K) - \mathrm{r}_1(a^M)\,\Delta f(x^{MK})\right) \\
&\qquad + \frac{R_{\{2,3\}}(a^K, LK)}{R_{\{2,3\}}(LK, LK)}\left(\Delta f(x^L) - \mathrm{r}_1(a^M)\,\Delta f(x^{ML})\right) \\
&\qquad - \frac{R_{\{2,3\}}(a^K, LK)}{R_{\{2,3\}}(LK, LK)}\mathrm{r}_{\{2,3\}}(LK)\left(\Delta f(x^{LK}) - \mathrm{r}_1(a^M)\,\Delta f(x^{MLK})\right), \qquad (37)
\end{aligned}
$$

$$
\mathrm{Var}_{K\cup L\cup M}[f(x)] \quad = \quad \sigma^2\, R^{(2)}_{K,L}(x,x)\, R_1(a^M, a^M). \qquad (38)
$$

The derivation of (37) and (38) is provided in Appendix F. The emulator outputs, derived using these equations, are shown in Figure 2. We can see that with these three boundaries, much is learnt across each of the displayed two-dimensional slices of the input space. Variance is particularly reduced for $x_2 = 0$ (left-hand column), this column also essentially containing the story of a smaller two-dimensional example (that when $x_2 = 0$) with three one-dimensional boundaries. The emulator predicts the model across much of the input space well; only in the top left and top right corners, when x_3 is large and x_1 small or large, is behaviour really uncertain.

The middle column shows the plane $x_2 = \pi/8$. We can see that the intersecting known boundary $\mathcal{M}$ at $x_1 = 0$ has much greater influence on the adjusted beliefs across the plane of interest than the lower dimensional known boundaries $\mathcal{K}$ and $\mathcal{L}$, these being subplanes of a plane parallel to the one of interest in this case. In contrast, if the plane of interest is parallel to the two-dimensional plane, for example $x_1 = \pi$ in the right-hand column, then the intersecting lines have a greater influence, although concentrated over a smaller area of the plane. The right-hand column particularly highlights the advantages of having as many known boundaries as possible. The intersecting lines provide much

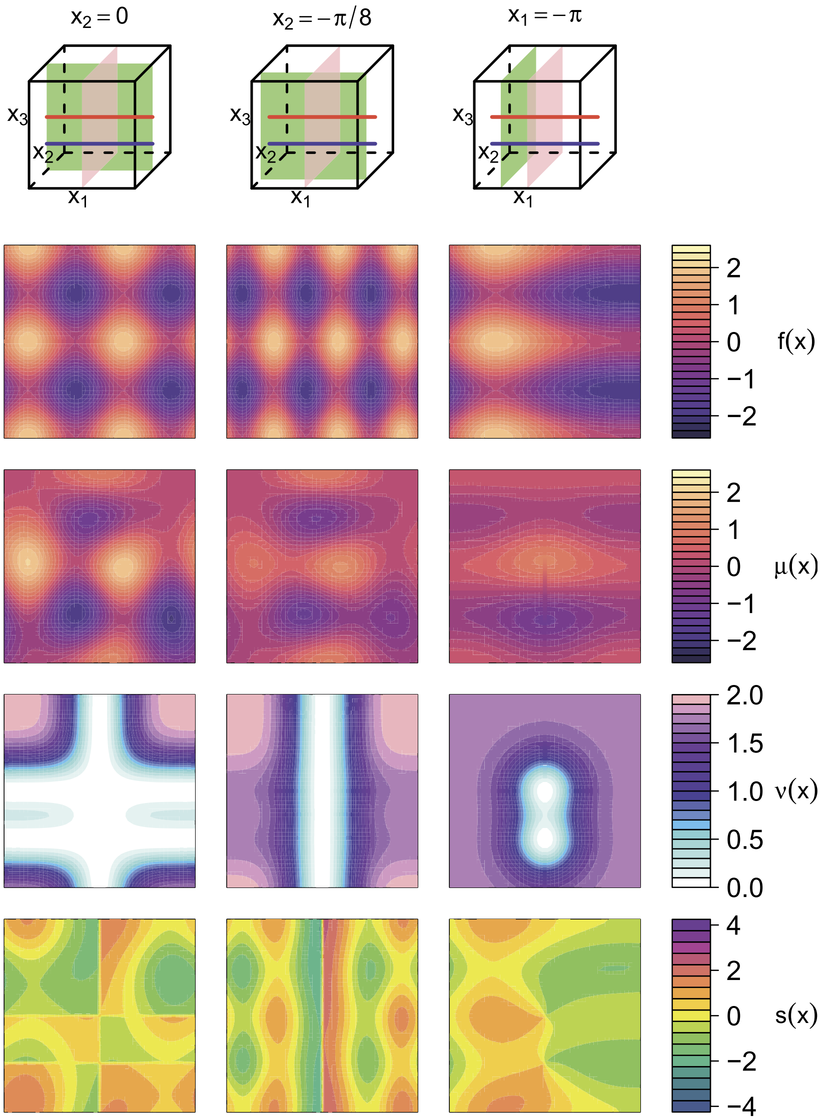

Figure 2: Updating the emulator for the three-dimensional function given by (16) by two sets of perpendicular boundaries with two and one boundaries respectively. Rows from top to bottom, show 1) position of known boundaries (red line $\mathcal{K}$, blue line $\mathcal{L}$ and pink plane $\mathcal{M}$) in the three-dimensional input space, along with the position of the two-dimensional slices (green planes) illustrating the two-dimensional plane in the three-dimensional input space, over which the remaining plots in the same column are plotted, 2) simulator function $f(x)$, 3) emulator expectation $\mu(x)$, 4) emulator variance $\nu(x)$, 5) standardised errors $s(x)$. Columns from left to right show results on the three planes $x_2 = 0$, $x_2 = -\pi/8$ and $x_1 = -\pi$ respectively, shown as the green planes in the top row. Note that for each two-dimensional plot, the variable with smaller index is along the horizontal axis.

increased precision over a smaller area, whilst the parallel plane reduces variance slightly (though still to a worthwhile degree) across the whole plane. In addition, the diagnostics are satisfactory across each plane in the example.

To summarise, for computer model applications where such sets of known boundaries exist, the gains of including them in the analysis using the general results derived in this section can be substantial, and therefore they should be included whenever possible.

4 Application of Methods to Arabidopsis Model

In the previous sections of this article, we have presented methodology for utilising knowledge of the behaviour of computer models along particular boundaries of the input space to aid emulation across the whole input space. In this section, we explore the implications such boundaries can have on a higher-dimensional scientifically relevant systems biology model of the hormonal crosstalk in the roots of an Arabidopsis plant.

4.1 Model of Hormonal Crosstalk in Arabidopsis Thaliana

Arabidopsis Thaliana is a small flowering plant that is widely used as a model organism in plant biology (Initiative, 2000). We demonstrate our known boundary emulation techniques on a model of hormonal crosstalk in the root of an Arabidopsis plant that was constructed by Liu et al. (2013). This Arabidopsis model represents the crosstalk of auxin, ethylene and cytokinin in Arabidopsis root development as a set of 18 differential equations, given in Table 2 of Appendix I, which must be solved numerically. The model takes an input vector of 45 rate parameters $(k_1, k_{1a}, k_2, \ldots)$, although we will be interested in a subset of 38 of them, as discussed in Appendix I, and returns an output vector of 18 chemical concentrations $([Auxin], [X], [PLSp], \ldots)$. This Arabidopsis model has been successfully emulated in the literature in the context of history matching (Vernon et al., 2018).

For the purposes of this article, we are interested in modelling the important output component $[ET]$, which represents the concentration of ethylene (Burg, 1973; Swarup et al., 2007), at early time $t = 2$. The ranges over which we allowed the inputs to vary are given in Table 3 in Appendix I, these being elicited as ranges of interest deemed sensible by the biological experts (Liu et al., 2013), and square rooted and mapped to a $[-1, 1]$ scale prior to analysis.

4.2 Establishing Known Boundaries

Establishing known boundaries requires some understanding of the scientific model. It is not uncommon for one or more known boundaries to occur in a model for some output components. Often, setting certain parameters to specific values will decouple smaller subsections of the system, which may allow subsets of the model equations to be solved analytically, for particular output components, as is the case for the Arabidopsis model.

We consider known boundaries for output component $[ET]$ by considering its rate equation:

$$\frac{d[ET]}{dt} \;=\; k_{12} + k_{12a}[Auxin][CK] - k_{13}[ET]. \tag{39}$$

A known boundary exists when rate parameter $k_{12a} = 0$, since in this case:

$$\frac{d[ET]}{dt} \;=\; k_{12} - k_{13}[ET] \quad\Rightarrow\quad [ET] \;=\; \frac{([ET^0]k_{13} - k_{12})\exp(-k_{13}t) + k_{12}}{k_{13}}, \tag{40}$$

where $[ET^0]$ is the initial condition of the $[ET]$ output component, and we see that $[ET]$ has been entirely decoupled from the rest of the system. $[ET]$ can now be obtained along the boundary $k_{12a} = 0$ with negligible computational cost. Note that this boundary is of dimension $p - 1 = 38 - 1 = 37$. The second (perpendicular) known boundary for $[ET]$ is a $p - 4 = 34$-dimensional boundary given by $k_{1a} = k_{2a} = k_{3a} = k_{18a} = 0$, which decouples the combined system of $[Auxin]$, $[CK]$ and $[ET]$. In this case, we can solve for $[Auxin]$ and $[CK]$ first:

$$\begin{aligned}
\frac{d[Auxin]}{dt} &= k_2 - k_3[Auxin] \\
\Rightarrow \quad [Auxin] &= \frac{([Auxin^0]k_3 - k_2)\exp(-k_3t) + k_2}{k_3}, \qquad (41) \\
\frac{d[CK]}{dt} &= -k_{19}[CK] \\
\Rightarrow \quad [CK] &= [CK^0]\exp(-k_{19}t). \qquad (42)
\end{aligned}$$

Inserting these solutions into the rate equation for $[ET]$ then yields:

$$\begin{aligned}
\frac{d[ET]}{dt} &= k_{12} + k_{12a}[CK^0]\exp(-k_{19}t)\left(\frac{([Auxin^0]k_3 - k_2)\exp(-k_3t) + k_2}{k_3}\right) - k_{13}[ET] \\
\Rightarrow\ [ET] &= \frac{k_{12}}{k_{13}}(1 - \exp(-k_{13}t)) + \frac{k_{12a}[CK^0]k_2}{k_3(k_{19} - k_{13})}\big(1 - \exp\big((k_{13} - k_{19})t\big)\big) \\
&\quad + \frac{k_{12a}[CK^0]([Auxin^0]k_3 - k_2)}{k_3(k_3 + k_{19} - k_{13})}\big(1 - \exp\big((k_{13} - (k_3 + k_{19}))t\big)\big), \qquad (43)
\end{aligned}$$

which can now be solved analytically with negligible computational cost, given the initial conditions $[Auxin^0]$ and $[CK^0]$ for Auxin and Cytokinin respectively. In this case, we have $[CK^0] = [ET^0] = [Auxin^0] = 0.1$ as the initial conditions suggested by the biological experts. The remaining initial conditions are shown in Table 4 in Appendix I. We will refer to this $p - 4$-dimensional boundary as $\mathcal{K}$ and the earlier presented $p - 1$-dimensional boundary as $\mathcal{L}$ in order to show the effect of the smaller-dimension boundary in comparison to the larger-dimension one. In addition, it is important to note that both boundaries $\mathcal{K}$ and $\mathcal{L}$ lie outside the input space of interest $\mathcal{X}$ as given by Table 3 in Appendix I. Despite this, assuming the behaviour of the model is reasonable in the vicinity of the boundaries, the information provided by the analytical solutions along the boundary can be useful for predicting model behaviour inside $\mathcal{X}$.

4.3 Emulator Structure and Specification

We restrict the form of our emulator to have the covariance structure as given by (1). We used a product Gaussian correlation function of the form given by (2), as we assumed that the solution to the Arabidopsis model would most likely be smooth and that many orders of derivatives would exist.

The prior emulator expectation and variance were taken to be constant, that is $\mathrm{E}[f(x)] = \beta$ and $\mathrm{Var}[f(x)] = \sigma^2$, where β and σ^2 were estimated to be the sample mean and variance of a set of previously evaluated scoping runs. In this section, we specify a common correlation length parameter $\theta = 3$ for each input, a choice consistent with the argument for approximately assessing correlation lengths presented in Vernon et al. (2010). This value of θ was also checked for adequacy using standard emulator diagnostics (Bastos and O'Hagan, 2008). We made this relatively simple emulator specification for illustrative purposes, the reason being that we wish to demonstrate that there are benefits to utilising the known boundaries regardless of how the parameters may have been estimated. To this end, in Section 4.5 we compare the effects of several different values of θ on an analysis with and without the known boundaries, but for now keep the value fixed at $\theta = 3$.

4.4 Comparison of Results

In this section, we compare the emulators of the above form constructed with and without use of the known boundaries $\mathcal{K} : k_{1a} = k_{2a} = k_{3a} = k_{18a} = 0$ and $\mathcal{L} : k_{12a} = 0$, and also with and without the addition of training points. The design for the additional training points is obtained by constructing a Maximin Latin hypercube design of size 1000 across the 38-dimensional input space, this then being sampled from to explore the effects of using different numbers of training points up to 1000. Bayes linear updates were carried out using the single and two perpendicular boundary updates given by (12), (13) and (33), (34) respectively. Additional updating is then performed using the sequential update formulae given by (17)-(19).

Equivalent plots to those shown in Figures 1 and 2 are substantially more difficult to visualise across all dimensions of a high-dimensional input space. We will use numerical diagnostics to assess these emulators in Section 4.5, but in this section we will restrict comparison of the emulators to visual diagnostics. Figure 3 shows model output against emulator expectation ±3 standard deviations for a set of 100 diagnostic test points for each of six emulators; first row: no boundaries (0KB); second row: 1 known boundary $\mathcal{K}$ (1KB); third row: two known boundaries $\mathcal{K}$ and $\mathcal{L}$ (2KB). The left column shows diagnostics for emulators without any training points D in the bulk of the input space (0TP), and the right column shows diagnostics for emulators that include 500 training points (500TP). Since the error bars for most of the points intersect the line $f(x) = \mathrm{E}[f(x)]$, this gives evidence to the fact that these emulators are valid, in the sense of generating predictions with appropriate associated uncertainty estimates, without making reference to the accuracy of the predictions. This heuristic appeals to Pukelsheim's Three Sigma Rule (Pukelsheim, 1994) which states that 95% of the probability mass of any unimodal distribution lies within 3 standard deviations of the mean.

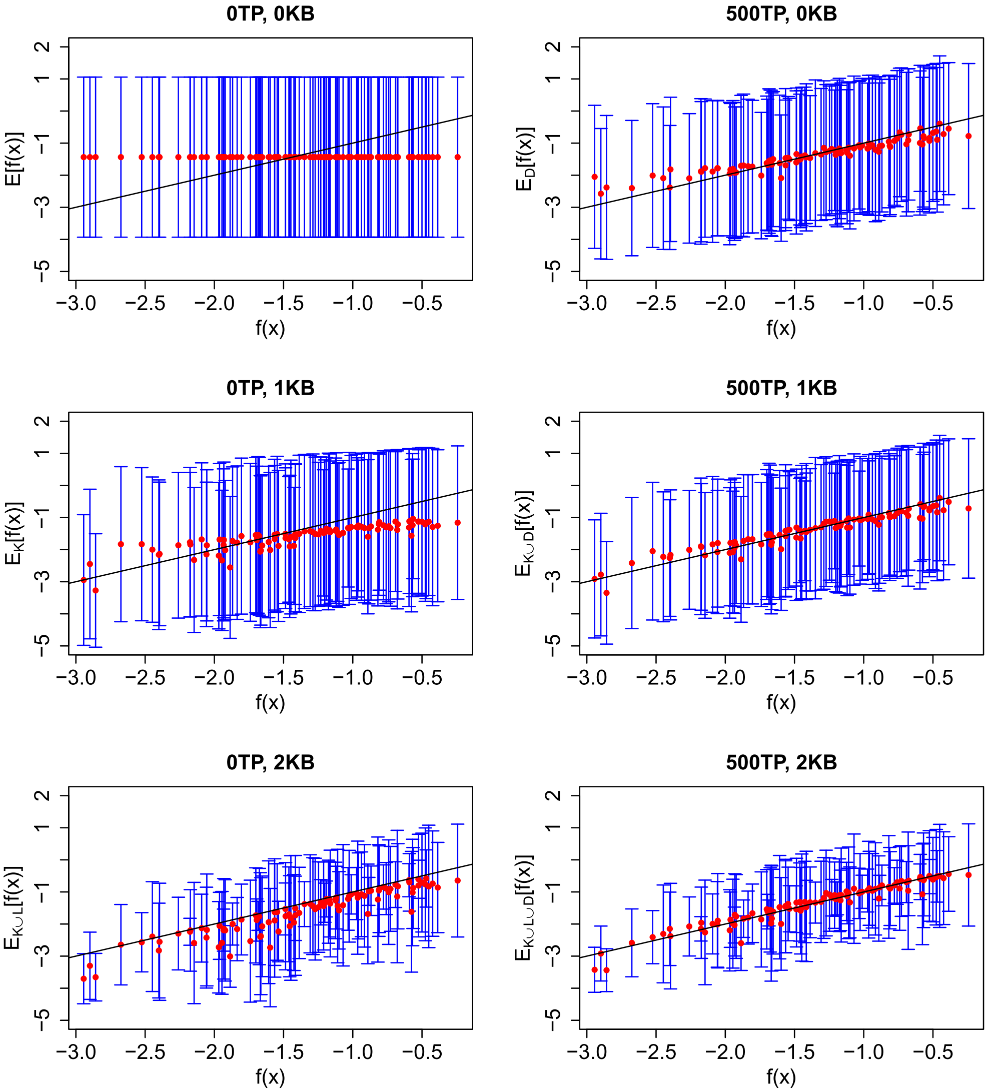

Figure 3: Diagnostic plots for the emulators of the Arabidopsis model output component $[ET]$. These show true model output against emulator expectation plus/minus 3 standard deviations for a set of diagnostic test points given; first row – no known boundary (0KB); second row – single known boundary $\mathcal{K}$ (1KB); third row – two known boundaries $\mathcal{K}$ and $\mathcal{L}$ (2KB). The left column shows diagnostics for emulators without additional training points D in the bulk of the input space (0TP), and the right column shows with 500 additional points (500TP), all for common correlation length parameter $\theta = 3$ for all input parameters.

In terms of the predictions themselves, the middle left panel of Figure 3 shows that the expected values of the points have been marginally influenced in general by $\mathcal{K}$, but to a greater degree for inputs for which the model output is smaller. The bottom left panel shows that the addition of $\mathcal{L}$ results in a much improved effect on the predictions than boundary $\mathcal{K}$ alone. We do note that in addition to increased accuracy, this leads here to slightly underestimated predictions: this is due to the the function values on the boundary $\mathcal{L}$ being typically lower than corresponding orthogonal parts of the input space, in addition to the emulator expectation approximately tracking the form of the correlation component $r_1(a)$ when moving orthogonally away from the boundary, a form which may undershoot the true form of $f(x)$. We note that the results of including both boundaries $\mathcal{K}$ and $\mathcal{L}$ are comparable to using no boundaries and 500 training points across the input space (top right panel), thus highlighting that utilising knowledge of computer model behaviour along known boundaries is worthwhile. Crucially, however, whereas the 500 training points require 500 potentially computationally intensive model evaluations and emulator matrix inversion calculations, the known boundaries involve no model evaluations or matrix inversion calculations. The simultaneous inclusion of both known boundaries and additional training points leads, unsurprisingly, to the emulators with greatest accuracy. In particular, it may be fair to say that out of the six emulators for which diagnostic plots are provided here, only the one with diagnostics provided in the bottom right panel yields sufficient accuracy and predictive capability for practical application.

The substantial and moderate effects of $\mathcal{L}$ and $\mathcal{K}$ respectively on our beliefs in comparison to individual points is largely a result of the dimension of the objects. The known boundaries are $p-1$- and $p-4$-dimensional objects respectively, resulting in significant variance resolution as a consequence of the volume of the input space within their proximity (particularly $\mathcal{L}$ for which the correlation function is effectively over a single dimension). In comparison, individual training runs (which are $0-$dimensional objects) influence far smaller volumes especially in high dimensions.

Since there is little computational cost involved in the incorporation of known boundaries, the most practical solution is to utilise them in conjunction with the regular training points. Looking at the bottom right panel of Figure 3, we notice a substantial improvement in comparison to using either the known boundaries or the training points individually. Were one aware of the known boundaries in advance, one could design the set of 500 runs accordingly, leading to further efficiency gains (see Vernon et al. (2019)). In terms of the biological application to the Arabidopsis model, we see that we are now able to utilise the information from many more known boundaries, now of differing dimensions, for each of the biological outputs of interest, leading to more powerful emulators. This is in contrast to Vernon et al. (2019) where only $p-1$ boundaries could be used, which restricted the number of boundaries available.

4.5 Sensitivity to Emulator Parameter Specification

We now compare emulators constructed using various different emulator parameter specifications. In particular, we explore the effect of changing the common correlation length parameter θ discussed in Section 4.3. We do this as we wish to focus on the

	Sum of Variances					MASPE				
θ	0.1	1	3	6	10	0.1	1	3	6	10
0TP, 0KB	344.23	344.23	344.23	344.23	344.23	0.84	0.83	0.83	0.83	0.83
0TP, 1KB	344.23	344.08	284.03	131.07	55.56	0.84	0.83	0.60	1.50	4.26
0TP, 2KB	344.23	295.29	100.77	15.04	2.44	0.84	0.67	1.77	24.67	182.44
200TP, 0KB	344.23	344.23	277.21	55.00	9.19	0.84	0.83	0.52	1.26	7.35
200TP, 1KB	344.23	344.08	230.32	24.13	2.12	0.84	0.83	0.43	3.61	49.21
200TP, 2KB	344.23	295.29	81.11	2.71	0.09	0.84	0.67	1.00	60.25	2508.44
500TP, 0KB	344.23	344.23	251.15	37.13	5.04	0.84	0.83	0.37	1.59	12.50
500TP, 1KB	344.23	344.08	209.05	16.29	1.10	0.84	0.83	0.35	4.66	82.04
500TP, 2KB	344.23	295.29	73.44	1.82	0.05	0.84	0.67	1.02	83.80	4305.17
1000TP, 0KB	344.23	344.23	229.70	23.87	2.07	0.84	0.83	0.33	2.16	25.09
1000TP, 1KB	344.23	344.08	191.42	10.87	0.56	0.84	0.83	0.33	5.86	129.16
1000TP, 2KB	344.23	295.29	67.08	1.21	0.02	0.84	0.67	1.07	117.86	7764.93

Table 1: Sum of Variances and Mean Absolute Standardised Prediction Error for the set of 500 diagnostic points for different values of common θ and numbers of known boundaries (KB) and training points (TP) in the bulk of the input space.

advantage of utilising known boundaries on emulation without confounding the effect on choice of parameter specification. Whilst we will demonstrate that the effects of known boundaries are substantial regardless of emulator structure and parameter specification, the value of θ does affect the relative size of the contributions of individual points to known boundaries (that is, larger dimensional objects). We compare several emulators with various values of correlation length parameter θ, numbers of training points and numbers of known boundaries using numerical diagnostics for 500 diagnostic points. These diagnostics, shown numerically in Table 1 and visually in Figure 4, are the sum of variances and Mean Absolute Standardised Prediction Error (MASPE), given by:

$$\sum_{w=1}^{500} \nu(x^{(w)}) \qquad \text{and} \qquad \frac{1}{500}\sum_{w=1}^{500} \frac{|f(x^{(w)}) - \mu(x^{(w)})|}{\sqrt{\nu(x^{(w)})}},$$

respectively, where $\mu(x)$ and $\nu(x)$ represent the appropriate emulator mean and variance in each case.

The prior sum of variances is 344.23 (constant for all θ), this being reduced by various degrees depending on the three varying features of our analysis. For small $\theta = 0.1$, neither training points nor known boundaries reduce the variances of the diagnostic points appreciably. With $\theta = 1$, training points are having negligible effect on variance, however, the larger known boundary objects have sufficient diagnostic points within their proximity to reduce uncertainty to some degree. For $\theta = 6$, the reduction in diagnostic variance arising from two known boundaries only (15.04) is greater than that of 1000 training points alone (23.87). As θ gets larger, 1000 training points in $\mathcal{X}$ have greater affect than two known boundaries outside of $\mathcal{X}$, for example, reducing the sum of variances to 2.07 and 2.44 respectively when $\theta = 10$ (and to 0.02 when both are used). These results are as expected from purely geometrical considerations.

It is common for acceptable values of the MASPE to be broadly around 1 (appealing to the properties of a standard half-normal distribution, which has expectation $\sqrt{2/\pi}$), and providing substantial evidence that an emulator is invalid if much greater than 2 or 3 (appealing to Pukelsheim's 3σ rule (Pukelsheim, 1994)). Equivalently, substantial

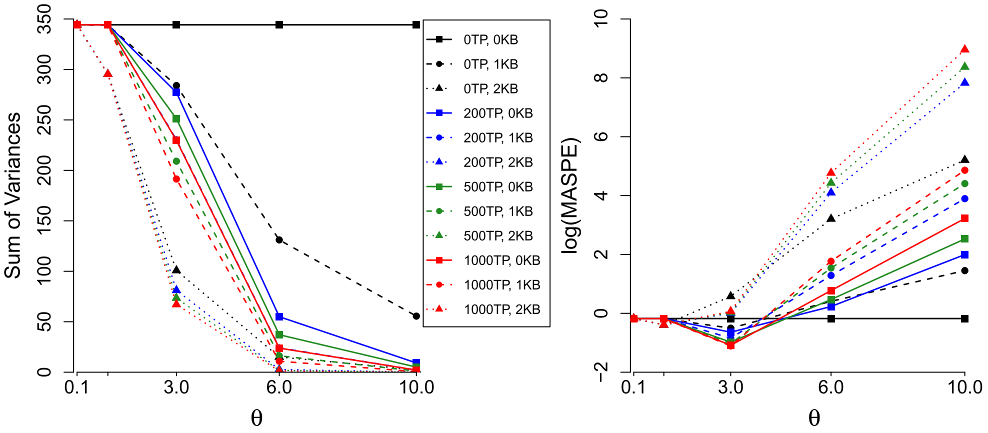

Figure 4: Sum of Variances and log Mean Absolute Standardised Prediction Error for the set of 500 diagnostic points for different values of common θ and numbers of known boundaries (KB) and training points (TP) in the bulk of the input space.

change in MASPE between prior and adjusted beliefs is also cause for concern. Prior MASPE is 0.84, which is suitably close to both 1 and $\sqrt{2/\pi} \approx 0.8$. The MASPE values for emulators with large values of θ are as expected unacceptable, with the value for 1000 training points alone being 25.09 and that for 1000 training points and two known boundaries 7764.93. This excessively larger value is due to the different ways in which known boundaries and training points influence the emulator. The known boundaries affect the input space as a large object with much influence over a particular part of the input space. On the other hand, since the training points are spread out across $\mathcal{X}$, the effect of averaging via interpolation of the points is likely to result in more accurate (and thus with common variance reduction appear more valid) predictions, even if θ is large. The MASPE values for $\theta = 3$, 1000 training points and two known boundaries is 1.07, which is much more acceptable. For the emulators with acceptable diagnostics, we see that the inclusion of known boundaries is clearly beneficial. In addition to sum of variances and MASPEs, we also calculated Root Mean Square Errors (RMSEs) for each emulator, these being displayed and discussed in Appendix I.

5 Conclusion

We have discussed how additional prior insight into the physical structure of a computer model related to known boundaries can be incorporated into emulators leading to substantial increases in accuracy for little additional computational cost.

In particular, here it is shown that if a computer model has boundaries or hyperplanes in its input space where it can either be analytically solved or just evaluated far more efficiently, then these known boundaries can be formally incorporated into the emulation process by analytic Bayesian updating of the emulators with respect to the information contained on the boundaries. Furthermore, we have shown that this is possible for a large class of emulators, and for multiple boundaries of various forms. The progress in this work in comparison to Vernon et al. (2019) is that we presented substantially more general results for arbitrary numbers of boundaries of varying dimension, stating which configurations of known boundaries permit analytical updates. Due to these analytic results and to the ease and substantial benefits of including known boundaries when emulating the Arabidopsis model, we would suggest that future UQ analyses of serious scientific models include a phase of identification and incorporation of known boundaries, if they are found to exist, as standard practice. Whilst the results of this article were with respect to a univariate computer model, the results extend naturally to the multivariate case, as discussed in Appendix H. In addition, many of the examples in this article can be thoroughly investigated via the associated code available at `https://github.com/samjacksonstats/KBE`.

There are many ways in which the work of this article could be developed. For example, extensions to the case of uncertain regression parameters within the emulator are possible, although the formal update would now depend on the specific form of the correlation function $r_j(a)$, which may not be tractable for some choices. Curved boundaries of various geometries could also be incorporated, provided both that suitable transformations were found to convert them to hyperplanes and that we were happy to adopt the induced transformed product correlation structure as our prior beliefs. Finally we note that, for some applications, there may be several hyperplanes in the input space along which model behaviour is known, however, analytical updates incorporating the information given by all of them may not be possible due to the set not satisfying the properties of the proposition in Section 3.3. There is then a possible design problem which involves selecting the best (in some sense) subset of the known boundaries which do permit analytic updating. This choice of boundaries may be in conjunction with design of the training points in the bulk of the input space X_D. Training point design should anyway take the known boundaries into account, as discussed in Vernon et al. (2019). We leave all these considerations to future research.

Supplementary Material

Supplementary Material to the Article: Efficient Emulation of Computer Models Utilising Multiple Known Boundaries of Differing Dimension (DOI: 10.1214/22-BA1304SUPP; .pdf). This file contains proofs for some of the results contained within this article, along with extended discussions relating to black box emulation packages, multivariate emulation, and the application of our methods to the model of *Arabidopsis Thaliana*.

References

Andrianakis, I., Vernon, I., McCreesh, N., McKinley, T. J., Oakley, J. E., Nsubuga, R. N., Goldstein, M., and White, R. G. (2015). "Bayesian History Matching of Complex Infectious Disease Models Using Emulation: A Tutorial and a Case Study on HIV in Uganda." 166

Bastos, T. S. and O'Hagan, A. (2008). "Diagnostics for Gaussian process emulators." *Technometrics*, 51: 425–438. MR2756478. doi: https://doi.org/10.1198/TECH.2009.08019. 183

Bowman, V. E. and Woods, D. C. (2016). "Emulation of Multivariate Simulators Using Thin-Plate Splines with Application to Atmospheric Dispersion." *Uncertainty Quantification*, 4: 1323–1344. MR3572374. doi: https://doi.org/10.1137/140970148. 165

Burg, S. P. (1973). "Ethylene in Plant Growth." *Proceedings of the National Academy of Sciences of the United States of America*, 70(2): 591–597. MR4294060. doi: https://doi.org/10.1073/pnas.2020424118. 181

Castelletti, A., Galelli, S., Ratto, M., Soncini-Sessa, R., and Young, P. C. (2012). "A general framework for Dynamic Emulation Modelling in environmental problems." *Environmental Modelling and Software*, 34: 5–18. 165

de Finetti, B. (1974). *Theory of Probability*, volume 1. Wiley. MR0440640. 167

Du, H., Sun, G., Goldstein, M., and Harrison, G. P. (2021). "Optimization via Statistical Emulation and Uncertainty Quantification: Hosting Capacity Analysis of Distribution Networks." *IEEE Access*, 9: 118472–118483. 166

Edwards, T. L., Nowicki, S., and Marzeion, B. e. a. (2021). "Projected land ice contributions to twenty-first-century sea level rise." *Nature*, 593: 74–82. 165

Goldstein, M. (1999). "Bayes Linear Analysis." In Kotz, S., Read, C. B., Balakrishnan, N., and Vidakovic, B. (eds.), *Encyclopedia of statistical Sciences*, chapter Bayes Linear Analysis, 29–34. New York: Wiley. MR2221272. doi: https://doi.org/10.1214/06-BA116. 167

Goldstein, M. and Wooff, D. (2007). *Bayes Linear Statistics*. Chichester: Wiley. MR2335584. doi: https://doi.org/10.1002/9780470065662. 167, 172

Gu, M. and Berger, J. O. (2016). "Parallel Partial Gaussian Process Emulation for Computer Models with Massive Output." *Annals of Applied Statistics*, 10(3): 1317–1347. MR3553226. doi: https://doi.org/10.1214/16-AOAS934. 166

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004). "Combining field data and computer simulations for calibration and prediction." *SIAM Journal on Scientific Computing*, 26(2): 448–466. MR2116355. doi: https://doi.org/10.1137/S1064827503426693. 165

Initiative, A. G. (2000). "Analysis of the Genome Sequence of the Flowering Plant Arabidopsis Thaliana." *Nature*, 408: 796–815. 181

Jackson, S. E. and Vernon, I. (2022). "Supplementary Material to the Article: Efficient Emulation of Computer Models Utilising Multiple Known Boundaries of Differing Dimension." *Bayesian Analysis*. doi: https://doi.org/10.1214/22-BA1304SUPP. 166

Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. A. (2011). "Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology." *The Annals of Applied Statistics*, 5(4): 2470–2492. MR2907123. doi: https://doi.org/10.1214/11-AOAS489. 165

Kennedy, M. C. and O'Hagan, A. (2001). "Bayesian Calibration of Computer Models." *Journal of the Royal Statistical Society*, 63(3): 425–464. MR1858398. doi: https://doi.org/10.1111/1467-9868.00294. 168

Liu, J., Mehdi, S., Topping, J., Friml, J., and Lindsey, K. (2013). "Interaction of PLS and PIN and Hormonal Crosstalk in Arabidopsis Root Development." *Frontiers in Plant Science*, 4(75). 181

Marshall, L., Johnson, J. S., Mann, G. W., Lee, L., Dhomse, S. S., Regayre, L., Yoshioka, M., Carslaw, K. S., and Schmidt, A. (2019). "Exploring How Eruption Source Parameters Affect Volcanic Radiative Forcing Using Statistical Emulation." *Journal of Geophysical Research: Atmospheres*, 124: 964–985. 166

McKinley, T. J., Vernon, I., Andrianakis, I., McCreesh, N., Oakley, J. E., Nsubuga, R., Goldstein, M., and White, R. G. (2018). "Approximate Bayesian Computation and simulation-based inference for complex stochastic epidemic models." *Statistical Science*, 33(1): 4–18. MR3757500. doi: https://doi.org/10.1214/17-STS618. 166

Pukelsheim, F. (1994). "The Three Sigma Rule." *The American Statistician*, 48(2): 88–91. MR1292524. doi: https://doi.org/10.2307/2684253. 183, 186

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). "Design and analysis of computer experiments." *Statistical Science*, 4(4): 409–435. MR1041765. 165

Swarup, R., Perry, P., Hagenbeek, D., Van Der Straeten, D., Beemster, G. T. S., Sandberg, G., Bhalerao, R., Ljung, K., and Bennett, M. J. (2007). "Ethylene Upregulates Auxin Biosynthesis in Arabidopsis Seedlings to Enhance Inhibition of Root Cell Elongation." *Plant Cell*, 19(7): 2186–2196. 181

Tan, M. H. (2018). "Gaussian Process Modeling of a Functional Output with Information from Boundary and Initial Conditions and Analytical Approximations." *Technometrics*, 60(2): 209–221. MR3804249. doi: https://doi.org/10.1080/00401706.2017.1345702. 166

Vernon, I., Goldstein, M., and Bower, R. G. (2010). "Galaxy Formation: A Bayesian Uncertainty Analysis." *Bayesian Analysis*, 5(4): 619–669. MR2740148. doi: https://doi.org/10.1214/10-BA524. 183

Vernon, I., Goldstein, M., and Bower, R. G. (2014). "Galaxy Formation: Bayesian History Matching for the Observable Universe." *Statistical Science*, 29(1): 81–90. MR3201849. doi: https://doi.org/10.1214/12-STS412. 165

Vernon, I., Goldstein, M., Rowe, J., Topping, J., Liu, J., and Lindsey, K. (2018). "Bayesian uncertainty analysis for complex systems biology models: emulation, global parameter searches and evaluation of gene functions." *BMC Systems Biology*, 12(1). 181

Vernon, I., Jackson, S. E., and Cumming, J. (2019). "Known Boundary Emulation of Complex Computer Models." *Journal of Uncertainty Quantification*, 7(3): 838–876. MR3980268. doi: https://doi.org/10.1137/18M1164457. 166, 167, 168, 170, 173, 176, 185, 188

Williamson, D., Goldstein, M., Allison, L., Blaker, A., Challenor, P., Jackson, L., and Yamazaki, K. (2013). "History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble." *Climate Dynamics*, 41(7): 1703–1729. MR3283898. doi: https://doi.org/10.1137/120900915. 165