Emulating the Thermosphere-Ionosphere Electrodynamics General Circulation Model (TIE-GCM)

Jonathan Rougier, Serge Guillas,
Astrid Maute, Arthur D. Richmond

Technical Report #2007-5
August 14, 2007

This material was based upon work supported by the National Science Foundation under Agreement No. DMS-0112069. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science

Statistical and Applied Mathematical Sciences Institute
PO Box 14006
Research Triangle Park, NC 27709-4006
www.samsi.info

Emulating the Thermosphere-Ionosphere Electrodynamics General Circulation Model (TIE-GCM)

Jonathan Rougier[1]
Department of Mathematics
University of Bristol, UK

Serge Guillas
School of Mathematics
Georgia Institute of Technology

Astrid Maute, Arthur D. Richmond
High Altitude Observatory
National Center for Atmospheric Research

August 14, 2007

[1] Corresponding author: Department of Mathematics, University of Bristol, University Walk, Bristol BS8 1TW, U.K.; email J.C.Rougier@bristol.ac.uk.

Abstract

We analyse the response of the TIE-GCM computer model of the upper atmosphere to atmospheric tides at its lower boundary and to variations of night-time ionization rates. TIE-GCM is an expensive code, and we choose to use our thirty evaluations of the model to construct an emulator, and then use the emulator for the analysis. TIE-GCM has some non-standard features, namely a periodic input, and model-output which is a periodic function of time, and our emulator must account for these. With only a limited number of evaluations our prior choices are important: we model the interaction between two of the inputs explicitly, as well as making informed choices for the emulator's prior R^2 and predictive variance. We validate our choices with detailed predictive diagnostics. Using the updated emulator we are able to quantify TIE-GCM's response to atmospheric tides and night-time ionization rates.

Keywords: Emulator, Gaussian process, Diagnostics, Periodic correlation function

1 Introduction

In a computer experiment, also known as an *in silico* experiment, we investigate the response of a computer code to changes in its parameters. Often, this code represents a complex system, and we refer to the code as a simulator, and the parameters—where they are numeric—as inputs. In many areas, e.g., Environmental Sciences, simulators are large, in the sense that we cannot afford to fill the input-space with evaluations. In this case we cannot avoid what O'Hagan (2006) refers to as 'code uncertainty': that uncertainty about the simulator output that could be reduced by additional evaluations, were we not constrained by resources.

In the Bayesian approach to computer experiments we treat the simulator as an unknown function. Our statistical model for the simulator is termed an emulator: we start with a prior emulator and then condition on evaluations of the simulator, to give us our updated emulator. Typically, the prior emulator is taken to be largely non-informative, which works well when the simulator is small and we have plenty of evaluations, but which will leave us with a lot of code uncertainty when the simulator is large. In this case it is natural to augment our evaluations with an informative prior.

In this paper we show how we have been able to incorporate detailed prior information into an emulator for a large simulator, the Thermosphere-Ionosphere Electrodynamics General Circulation Model (TIE-GCM) of the upper atmosphere. Section 2 describes the simulator, and our approach to emulating it. Section 3 describes the special features of the simulator, and how we represent them, structurally, in our prior emulator, and Section 4 describes how we quantify these choices. There are many uses for our updated emulator: section 5 shows one simple use—examining the simulator's main-effects and interactions, and comparing these with our qualitative understanding of the physics. Section 6 concludes. Additionally, an Appendix describes our framework for specifying periodic correlation functions.

2 Emulating the TIE-GCM simulator

2.1 The experiment

The TIE-GCM simulator (Richmond *et al.*, 1992) is designed to calculate the coupled dynamics, chemistry, energetics, and electrodynamics of the global thermosphere-ionosphere system between about 97 km and 500 km altitude. It has many parameters at the lower and upper boundaries, as well as a number of uncertain internal parameters. For this study we explore the response of the simulated ionospheric $\boldsymbol{E} \times \boldsymbol{B}/B^2$ drift velocity, where $\boldsymbol{E}$ and $\boldsymbol{B}$ are the electric and geomagnetic fields, to variations in just three inputs: two that help describe atmospheric tides at the TIE-GCM lower boundary, and one that constrains the minimum night-time electron density. Radar measurements of the $\boldsymbol{E} \times \boldsymbol{B}/B^2$ drift velocity provide an observational target. The drift varies daily, but also with season, solar cycle and location of the observation. Only averaging over many days for given geophysical conditions will give a regular pattern.

Atmospheric tides are global waves with periods that are harmonics of 24 hours. They are generated at lower atmospheric levels, and they are modulated by variable background winds as they propagate to the upper atmosphere. They are difficult to define since observations are limited and the tides vary not only with geographic location, local time and season, but also in a somewhat irregular manner from one day to the next. Modelling the tidal propagation through the atmosphere, and accurately determining their distribution at the TIE-GCM lower boundary, remains a challenge. For this study, we include fixed diurnal (24 hour period) and semidiurnal (12 hour period) migrating (Sun-synchronous) tidal components at the TIE-GCM lower boundary, taken from the model of Hagan and Forbes (2002a,b), plus an additional variable tidal forcing (migrating (2,2) mode) which is known to be important for the electrodynamics (Fesen *et al.*, 2000). The amplitude of the perturbation in the height of a constant-pressure surface at the TIE-GCM lower boundary, $\texttt{AMP} \in [0, 36]$ dam, and the local time at which this maximises, $\texttt{PHZ} \in [0, 12]$ hr, are two of the three inputs we explore.

At night, the ionospheric electron density below 200 km is small and difficult to measure, but nonetheless has an important influence on the night-time electric field. Our third model-input is the logarithm, base 10, of the minimum night-time electron density in cm^{-3}, $\texttt{EDN} \in [3, 4]$. All other input parameters in the TIE-GCM simulator are held constant for our experiments. The simulations are done for equinox, at low solar and geomagnetic activity. The model is first run for an extensive time period to get a diurnally reproducible state.

One ultimate purpose of our project is to calibrate uncertain TIE-GCM parameters to observational data. An earlier attempt to calibrate `AMP`, `PHZ`, and `EDN` to multiple sources of geomagnetic perturbation data as well as ionospheric drift data revealed that the ionospheric electrical conductivities were underestimated, which caused an under-representation of the simulated geomagnetic perturbation amplitudes, and therefore an inadequate selection of the three inputs. Therefore the project emphasis changed to understanding better the simulator response to the three inputs, in order to construct a second design that was more representative of the observations. This is the analysis we describe here.

Our experimental design was a maximin Latin Hypercube, shown in Figure 1. In retrospect this was not the best choice; this is discussed further in section 5.

The TIE-GCM $\boldsymbol{E}\times\boldsymbol{B}/B^2$ drift velocity outputs (m/s), which we denote by 'drift', comprise periodic functions of magnetic local time in hours on $[0, 24]$, at a large collection of sites across the globe. We analyse the sites marginally, disregarding shared information that might be available from sites that are proximate. Therefore our simulator outputs in each evaluation comprise points on a periodic function of time for some pre-specified site. We concentrate here on the upward drift at the location of the Jicamarca incoherent scatter radar observatory (JRO), Peru, at the geomagnetic equator (11.9° S, 76.0° W geographic). In general, at the geomagnetic equator the upward drift is mostly positive during the day and negative during the night. We write the simulator output as the scalar $f(\boldsymbol{r}, t)$, where $\boldsymbol{r} \equiv (\texttt{AMP}, \texttt{PHZ}, \texttt{EDN})$ and $t \in [0, 24]$ hrs indexes magnetic local time from midnight. We refer to $\boldsymbol{r}$ as the value of the input variables, and t as the value of the index variable. The upward drift for JRO is shown in Figure 2 for the ensemble of evaluations along with the observations. The main features of the simulator-output and the data are a peak around noon, and excursions in the early evening. A previous study showed that the tidal forcing at the lower boundary and the night time electron density influence these two features (Fesen *et al.*, 2000). The peak in the early evening develops for low electron density in the lower ionosphere (E region), when the relative influence

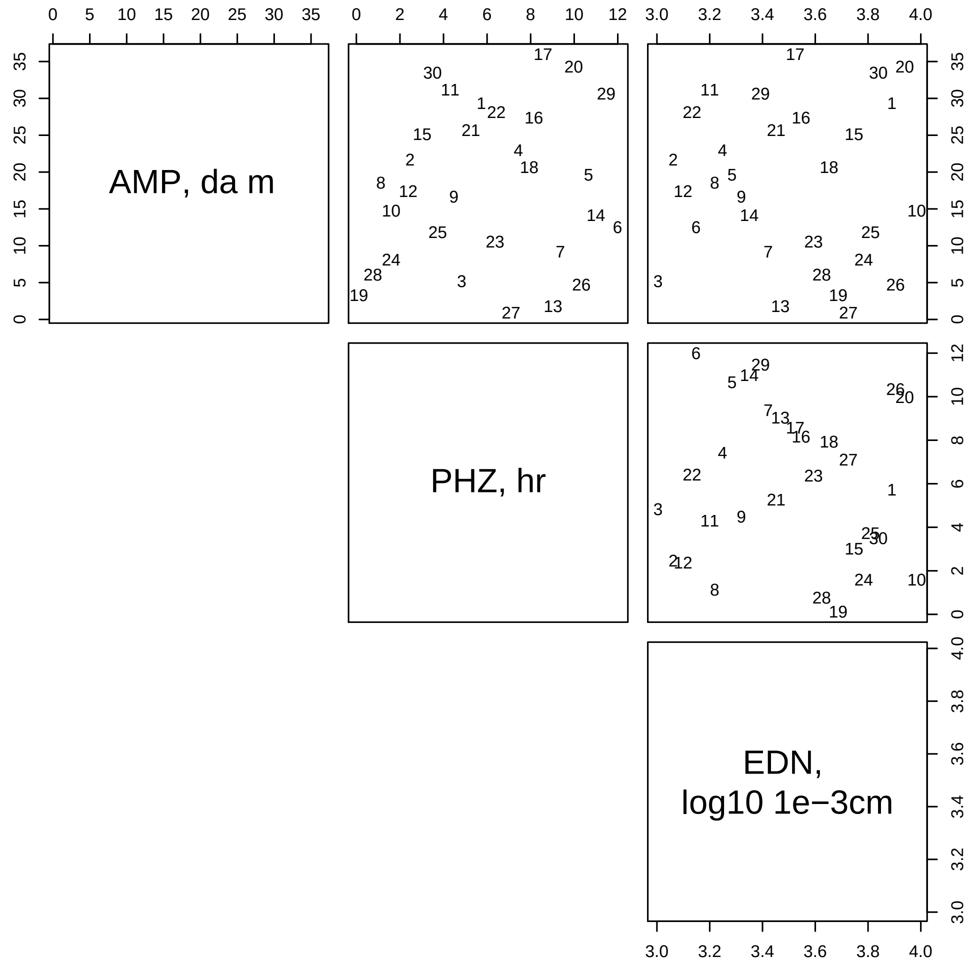

Figure 1: Maximin Latin Hypercube design in the three TIE-GCM inputs. Numbers index the 30 evaluations (see below, Figure 3 and Figure 4).

of the upper ionosphere (F region) dominates. The daytime upward drift is mainly influenced by the tidal winds.

Our intention is to build an emulator for the function $f(\boldsymbol{r}, t)$. An emulator (or, alternatively, 'surrogate') is a stochastic representation of a deterministic function, which provides, for any set of $(\boldsymbol{r}, t)$ values, a point prediction of the simulator outputs and a measure of uncertainty. Where the simulator is deterministic, this uncertainty is reducible by doing further evaluations; O'Hagan (2006) refers to it as 'code uncertainty'. In a fully-probabilistic framework, an emulator is constructed by conditioning prior judgements about the simulator on the ensemble of evaluations, as originally described by Currin *et al.* (1991). This approach has the attractive feature that the emulator interpolates the ensemble with probability one; i.e., every sample path from the emulator passes through every one of the evaluations. This approach, almost invariably implemented using a Gaussian process prior (or, as in this paper, a *conditionally* Gaussian process prior), is gradually maturing as a

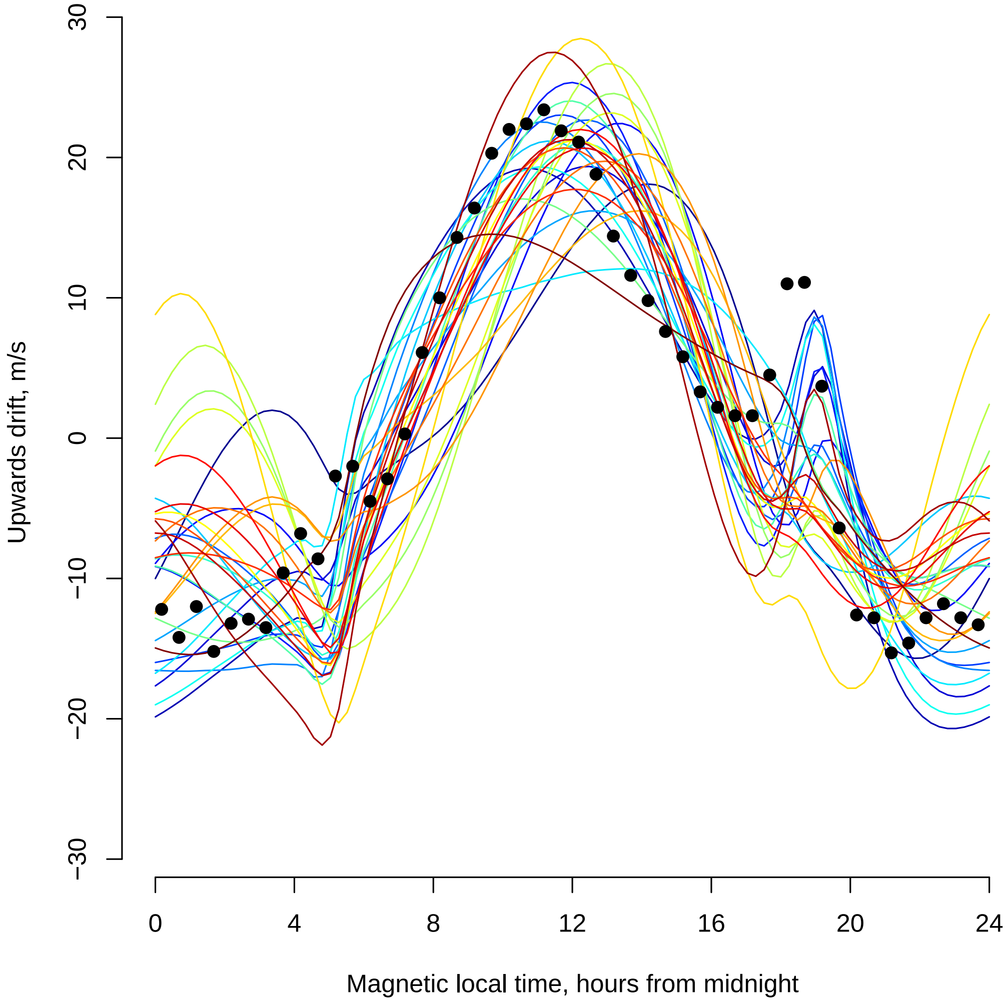

Figure 2: The ensemble of simulator evaluations (coloured lines) and observations (dots), for the Jicamarca (JRO) site

technology (see, e.g., the textbook of Santner *et al.*, 2003). Our emulator for TIE-GCM explores some of the more subtle issues, namely functional rather than scalar outputs, emulator diagnostics, and, perhaps most importantly, highly-structured prior judgements.

2.2 Functional outputs

The recent literature on emulating multivariate (including functional) outputs is reviewed in section 2 of Rougier (2007a), which describes in detail the approached used for the TIE-GCM model. Here follows a summary of the salient features.

There is a natural approach to modelling functional outputs, which leads to what might be termed an 'outer-product' emulator. We think of our emulator as being a linear sum of specified basis functions in the index variables, and we think of the coefficients of this linear combination as being uncertain functions of the input variables, which we also write in terms of specified basis functions. Thus we have

$$f(\boldsymbol{r},t) = \sum\nolimits_i \alpha_i(\boldsymbol{r}) g_i(t), \quad \text{and} \quad \alpha_i(\boldsymbol{r}) = \sum\nolimits_j \beta_{ij} g_j(\boldsymbol{r}) \text{ for each } i, \tag{1a}$$

or, together,

$$f(\boldsymbol{r},t) = \sum\nolimits_{ij} \beta_{ij} g_{ij}(\boldsymbol{r},t) \quad \text{where } g_{ij}(\boldsymbol{r},t) \triangleq g_j(\boldsymbol{r}) g_i(t); \tag{1b}$$

we adopt the convention throughout this paper that functions are identified by both their symbols and their inputs, so that, for example, $g_1(\boldsymbol{r})$ is a different function to $g_1(t)$. We specify the basis functions, and then our distribution for f is inferred from our distribution for $\boldsymbol{\beta} \triangleq \{\beta_{ij}\}$.

As it stands, for practical reasons this formulation requires a finite truncation for the series in i and j, at, say, q_i and q_j terms. This makes the emulator finite-dimensional, since as few as $q_i \times q_j$ output values suffice to identify the $\boldsymbol{\beta}$ coefficients exactly. This seldom corresponds to our judgements about the underlying simulator, and so we add an additional stochastic process that is effectively infinite-dimensional, giving

$$f(\boldsymbol{r},t) = \sum_{j=1}^{q_j} \sum_{i=1}^{q_i} \beta_{ij} g_{ij}(\boldsymbol{r},t) + e(\boldsymbol{r},t) \tag{2}$$

where e is referred to as the residual process. Thus our distribution for f is inferred from our choices for $G \triangleq \{g_{ij}\}$, and, based on these, our distribution for $\{\boldsymbol{\beta}, e\}$. In an outer-product emulator G is the outer-product of functions in $\boldsymbol{r}$ and in t, but in general G can be any finite collection of functions in $(\boldsymbol{r}, t)$.

2.3 Conjugate structure

As is clear from (2), our emulator has much in common with a response surface, e.g., one fitted by Ordinary Least Squares (OLS). The difference is in the residual process: in a standard regression this residual is a stationary nugget, i.e., a process with correlation function $\operatorname{Corr}\big(e(\boldsymbol{r},t), e(\boldsymbol{r}',t')\big) = 1$ if $\boldsymbol{r} = \boldsymbol{r}'$ and $t = t'$, and zero otherwise. Our residual, on the other hand, needs to have a much richer structure, in order that the sample paths of the emulator interpolate the ensemble of evaluations.

But an emulator is sufficiently like a regression that we should prioritise diagnostics in the same way that they are routinely applied in OLS regression, to evaluate the effects of our choices for G, for the distribution of $\{\boldsymbol{\beta}, e\}$, and, if we so choose, non-linear transformations of the simulator-outputs.

Predictive diagnostics are cheap in OLS, but they can be expensive when the residual is not a nugget. For this reason we favour a conjugate approach for our emulator. Our emulator is structured hierarchically, following the standard Normal Inverse Gamma (NIG) approach:

$$\boldsymbol{\beta} \perp\!\!\!\perp e(\cdot) \mid \tau \tag{3a}$$

$$\boldsymbol{\beta} \mid \tau \sim \mathrm{N}_q\left(\boldsymbol{m}, \tau V\right) \tag{3b}$$

$$e(\cdot) \mid \tau \sim \mathrm{GP}\big(0, \tau\, \kappa(\cdot, \cdot)\big) \tag{3c}$$

$$\tau \sim \mathrm{IG}\left(a, d\right) \tag{3d}$$

where $e(\cdot)$ denotes the residual evaluated at any finite collection of $(\boldsymbol{r}, t)$-values. Here GP denotes a Gaussian Process with specified mean function and variance function, N_q a q-dimensional Gaussian distribution with specified mean vector and variance matrix, and IG an Inverse Gamma distribution with specified degrees of freedom and scale. Note that this framework ought to be restricted to model-outputs with a common type, because of the common variance multiplier τ.

This distribution for $\{\boldsymbol{\beta}, e(\cdot)\}$ induces a distribution for $f(\cdot)$ for any finite collection of $(\boldsymbol{r}, t)$-values. Our emulator is specified in terms of the q regressors G, the variance function κ, which is defined on $(\boldsymbol{r}, t) \times (\boldsymbol{r}', t')$, and the NIG hyperparameters $\{\boldsymbol{m}, V, a, d\}$. Conditioning on the evaluations in the ensemble just updates these four hyperparameters. Therefore the predictive distribution of the emulator always has the same form, in this case a multivariate Student-t distribution. The updating equations for the hyperparameters, and the resulting parameters of the predictive distribution for f are standard (see, e.g., Rougier, 2007a).

We appreciate that our choice of the NIG distribution restrictive, in the sense that it is unlikely to reflect fully our judgements about the model (see, e.g., Goldstein, 1983; O'Hagan and Forster, 2004). Our view is that this is an acceptable price to pay for tractability, and a relatively easy quantification of the prior emulator. Ultimately, we will be guided by diagnostic information. If we find that the diagnostics are acceptable then we are fortunate and can proceed with caution. If not, then we have to generalise: the simplest generalisation is to mix over a collection of prior distributions. This is very straightforward within a conjugate analysis, because the predictive density is easy to compute (see, e.g., Rougier, 2007b). We describe our predictive diagnostics in subsection 4.1. More generally, it would be possible to extend the hierarchical approach a level further, with distributions for the hyperparameters, and also for parameters that occur in the variance function κ. This does not give rise to a closed-form updated predictive density, however, so a simulation-based approach to predicting f would be required; the same would be true *a fortiori* for the level beyond that, which would be to have a prior over the collection of regressors (Chipman *et al.*, 1997). Our view, though, is that we have insights about the simulator which are pertinent to our choice of the regressors, the variance function and the hyperparameters, and we do not feel the need for any further formal levels. Therefore we will make explicit choices for these, and tune our choices, informally, after inspecting the emulator diagnostics. While this is technically incoherent, we see it as approximating the inclusion of another level in the hierarchy, while allowing our response to the diagnostic information to be within our control, rather than shaped by aspects of the statistical modelling that we do not really understand.

Note that we are fairly unconcerned about the issue of double-counting the data in the ensemble, when we tune to diagnostics. For one thing, double-counting appears to be an accepted feature of emulator construction, where often the hyperparameters are fitted, e.g., by maximum likelihood or cross-validation, and then plugged-in. Of course this is not in any way exculpatory, but our judgement is that our predictive diagnostics provide a highly-aggregated picture, so that by tuning to these we are unlikely to be over-fitting. Also, since we are hand-tuning, the process is entirely under our control, and we can err on the side of caution.

2.4 Tractability

Much of the literature on emulation, like that of its cousin Spatial Statistics, is concerned with what can be done in practice to handle large datasets. Our ensemble is not particularly large, and so the feasibility of constructing the emulator is not an issue. However, the speed with which we can construct and emulator and predict a response *is* an issue, if we want to perform detailed diagnostic checking, as is the numerical stability of the calculations.

Our first simplifying choice is standard: to treat the variance function κ as separable in the input variables and the index variable, so that we can write

$$\kappa([\boldsymbol{r}, t], [\boldsymbol{r}', t']) = \kappa(\boldsymbol{r}, \boldsymbol{r}')\, \kappa(t, t'). \tag{4}$$

This is a dramatic simplification, that can be thought of in terms of a Markov property for our judgements about $e(\boldsymbol{r}, t)$, as explored in O'Hagan (1998). It is almost invariably invoked in emulators, usually in a much more extreme form, namely that $\kappa(\boldsymbol{r}, \boldsymbol{r}')$ is itself separable in each of the components of $\boldsymbol{r}$ (we will *not* be making this choice). Note that separability is a choice about our prior for f: with sufficient evaluations in our ensemble we would hope that unwanted structure arising from separability is effaced. Likewise, with a reasonable choice of regressors we would hope that separability in the residual variance is relatively unimportant, because the variation in f that is attributed to the residual is small. Note, however, that this is often not the case in practice, when a separable residual variance is combined with just a constant in the regressors (we will not be making this choice either).

Our second choice concerns the treatment of the functional output. In the raw simulator output, the dynamic time-stepping of the solver produces a different time-series for each combination of site and input values. We can regularise this output by interpolating it onto a collection of fixed knots in magnetic local time, say k knots in total. We can go further, though, and restrict ourselves to always predicting the simulator behaviour at the same set of knots. In this case our ensemble data and our predictions have the same regular structure: we can imagine the simulator outputs laid-out with evaluations down the rows and times along the columns, always k in number. This is suggested by denoting the value of the input variables as '$\boldsymbol{r}$', for 'row' (or 'run').

This regular structure has two advantages. First, we can use a vector basis for $g_i(t)$, because now the t values are prescribed and fixed. So instead of specifying q_t functions of t we can specify a $k \times q_t$ matrix, where each row corresponds to a knot. Second, the regressors for the ensemble, if they are the outer-product of regressors in $\boldsymbol{r}$ and regressors in t, have a Kronecker product form, and this is conformable with the Kronecker product form that arises from the residual variance matrix being separable in $\boldsymbol{r}$ and t. The computational advantages of this are substantial (Rougier, 2007a). Therefore our third choice will be to use a full outer-product of regressors in $\boldsymbol{r}$ and t, as described in (1b).

3 Special treatment

In this section we detail the choices we make to incorporate more of our prior information into the emulator. Some of these are obvious: the output from the simulator is periodic, and we would like that to be a feature of our emulator. Likewise, one of the inputs is also periodic (`PHZ`). Others are less so: `AMP` and `PHZ` interact, for example. Our view is that we want to include as much of this information as we can. It is intellectually satisfying if we can ensure that properties that are known or strongly suspected to hold in the simulator also hold in the emulator. But it is also practically important if we have only a limited number of evaluations in our ensemble. In this case we do not want to 'waste' evaluations discovering empirically things that we already know.

The choices we made in subsection 2.4 and we make in this section are 'structural', in that they define our emulator in general terms. In section 4 we quantify our prior choices within the constraints of the structure we have imposed.

3.1 Periodic outputs

At any choice of $\boldsymbol{r}$, the simulator-outputs comprise a periodic function of time, i.e.,

$$f(\boldsymbol{r}, 0) = f(\boldsymbol{r}, 24) \quad \text{for all } \boldsymbol{r}. \tag{5}$$

We judge, further, that there is continuity in at least the first derivative with respect to time. In subsection 2.4 we chose to treat the simulator-outputs as a finite collection of values at a specified collection of knots in $[0, 24]$. One simple approach to periodicity is to make these values the knots of a periodic interpolator. We use a periodic cubic spline. To regularise the data we fit the spline to the raw simulator-outputs at each $\boldsymbol{r}$ and then interpolate this output onto the knot-locations. To predict at new values of $\boldsymbol{r}$, we predict the values at the knot locations, and then fit the spline through the knots. To assess the uncertainty in our prediction, we can interpolate samples from the emulator, or we can interpolate the mean and standard deviation: our preference is for samples because these give a clearer idea of the correlation range. Examples are given in section 5.

Our choice of knot-locations will be deferred until section 4. Here we describe how we construct the vector basis functions for the simulator output. Because the periodicity of the outputs is taken care of by the spline interpolation, we are free to construct these basis vectors without concern for periodicity. Each basis is a vector over the knot-locations; with k knot-locations we are seeking to construct a $k \times q_t$

matrix which, for convenience, should be orthogonal. We use an approach similar to that of Higdon *et al.* (2007). We start by identifying sites similar to the site we want to emulate. Then we stack the evaluations, for all these sites and for all $\boldsymbol{r}$ values, into the rows of a matrix, where each column corresponds to a knot-location. We row-centre this matrix, but do not rescale it. The right eigenvectors from the singular value decomposition (SVD) of the resulting matrix provide an orthonormal basis—known as Empirical Orthogonal Functions (EOFs) in Meteorology—and the squared singular values give some guidance as to the variation explained by each vector in the basis. We do not refer to this as Principal Components, as we cannot consider the rows of the matrix to be exchangeable; nor do we refer to the squared singular values as variances. We are using the SVD as a dimensional-reduction technique, rather than a statistical technique.

We could compute EOFs just from the site we are emulating. However, we are wary of doing this, as we will then be using the ensemble both to choose the time basis functions and then to update the regression coefficients on these basis functions. Our decision to include evaluations from similar sites is mainly to dilute the impact of the data from our particular site. But it would also be useful, in a multi-site analysis, to have a common set of basis vectors. When we decide how many basis functions we need, we will bear in mind the risk of over-fitting, and select the basis vectors more on what they can contribute through their shape, rather than how much extra variation they can explain. This choice, like our other choices below, will be informed by our emulator's predictive diagnostics.

3.2 The relation between amplitude and phase

In our simulator there is a relatively simple relationship between amplitude and phase, which is that amplitude moderates the effect of phase. In particular, it is necessarily true that when amplitude is zero, the setting of phase has no impact, or

$$\boldsymbol{f}(0, \mathtt{PHZ}, \mathtt{EDN}) = \boldsymbol{f}(0, \mathtt{PHZ}', \mathtt{EDN}) \quad \text{for all values of } \mathtt{PHZ},\ \mathtt{PHZ}', \text{ and } \mathtt{EDN}. \tag{6}$$

Generally, as AMP increases, so the impact of PHZ becomes more pronounced. We would like the necessary constraint (6) to hold with probability one in our emulator, and we would also like our prior emulator to have the general property as well. This constrains the choices we can make for the regressors and for the variance function of the residual.

For the regressors, we must ensure that (6) holds for each regressor. The natural formulation for this, which also has the more general property, is always to pair regressors which include PHZ with a carefully-chosen regressor in AMP, in the general form

$$g(\mathtt{AMP}, \mathtt{PHZ}) = g_{\mathtt{AMP}}(\mathtt{AMP}) \times g_{\mathtt{PHZ}}(\mathtt{PHZ}) \tag{7}$$

where $g_{\mathtt{AMP}}(0) = 0$ and $g_{\mathtt{AMP}}$ is monotonic increasing. We choose polynomials for $g_{\mathtt{AMP}}$, namely

$$\sqrt{3}\bar{\mathtt{AMP}}, \quad -3\sqrt{5}\bar{\mathtt{AMP}} + 4\sqrt{5}\bar{\mathtt{AMP}}^2, \quad \ldots \tag{8}$$

where $\bar{\mathtt{AMP}} \triangleq \mathtt{AMP}/36$, i.e., AMP scaled to the unit interval. The coefficients in these polynomials are chosen to make the set of functions orthonormal on $[0, 1]$.

For the residual we use a similar approach, writing the residual in all three inputs as the sum of two parts, one of which drops out when `AMP` is zero:

$$\boldsymbol{e}(\texttt{AMP}, \texttt{PHZ}, \texttt{EDN}) = \bar{\texttt{AMP}} \times \boldsymbol{e}(\texttt{AMP}, \texttt{PHZ}, \texttt{EDN}) + \boldsymbol{e}(\texttt{AMP}, \texttt{EDN}). \tag{9}$$

This residual will be non-stationary in the prior, having greater variability when `AMP` is larger, which is consistent with our judgements. We impose further structure on the prior form of the residual, by separating `EDN` from `AMP` and `PHZ`:

$$\boldsymbol{e}(\texttt{AMP}, \texttt{PHZ}, \texttt{EDN}) = \bar{\texttt{AMP}} \times \boldsymbol{e}(\texttt{AMP}, \texttt{PHZ}) + \boldsymbol{e}(\texttt{EDN}) \tag{10}$$

Thus in our prior emulator, interactions between `EDN` and the other two inputs occur only in the regression terms.

3.3 Periodic input: Phase

Our simulator's response to the `PHZ` input is periodic, in the sense that

$$\boldsymbol{f}(\texttt{AMP}, 0, \texttt{EDN}) = \boldsymbol{f}(\texttt{AMP}, 12, \texttt{EDN}) \quad \text{for all } \texttt{AMP} \text{ and } \texttt{EDN}. \tag{11}$$

If we want this to hold in our emulator, then this constrains the choices we make for the regressors and the residual variance function.

For the regressors, we can separate `PHZ` out from the other inputs, and always use regressors that are periodic. Fourier terms are the natural choice, as they are smooth across the boundary which corresponds to our judgements about the simulator (Bernoulli polynomials, for example, would not have this property); we choose the constant and

$$\sqrt{2}\sin(2\pi k\,\bar{\texttt{PHZ}}), \quad \sqrt{2}\cos(2\pi k\,\bar{\texttt{PHZ}}) \qquad k = 1, 2, \ldots \tag{12}$$

where $\bar{\texttt{PHZ}} \triangleq \texttt{PHZ}/12$, i.e., `PHZ` scaled to the unit interval, and the $\sqrt{2}$ makes these functions orthonormal on the unit interval.

For the residual, we start by treating the variance function of $\boldsymbol{e}(\texttt{AMP}, \texttt{PHZ})$ as separable in `AMP` and `PHZ`, so that we can focus on specifying a variance function in `PHZ` that has periodic sample paths. Thus the variance function for $\boldsymbol{e}(\texttt{AMP}, \texttt{PHZ})$ in (10) can be written as

$$\kappa(\texttt{AMP}, \texttt{AMP}') \times \kappa(\texttt{PHZ}, \texttt{PHZ}') \tag{13}$$

where the first variance function can be any choice, and the second must satisfy, for example, $\kappa(0, 12) = 1$. Our approach is described in the Appendix; it allows us to construct a stationary variance function, which we term the Bernoulli correlation function, and to specify the periodicity, the smoothness and the range of the sample paths.

3.4 Other choices

We make some additional choices to simplify our prior emulator. As noted in subsection 2.1, we will take the regressors to be the outer-product of functions of the input variables $\{\texttt{AMP}, \texttt{PHZ}, \texttt{EDN}\}$ and functions of the index variable $\{t\}$. For the latter we are using EOFs, as described in subsection 3.1. For the former we will choose a monomial structure, so that each basis function is the product of functions of each of the three inputs. As described in subsection 3.3, the functions of PHZ are Fourier functions, given in (12). As described in subsection 3.2, these always occur with one of the functions of AMP given in (8). This leaves the functions of EDN to be specified: we choose Legendre polynomials, shifted onto the interval $[3, 4]$.

For the prior structure of the residual, we have chosen to separate the input variables and the index variable, as described in subsection 2.1. Then, based on our choices in subsection 3.2 and subsection 3.3, our prior variance function is

$$\big\{\mathtt{A\bar{M}P} \times \mathtt{A\bar{M}P}' \times R(\mathtt{AMP} - \mathtt{AMP}')\, R(\mathtt{PHZ} - \mathtt{PHZ}') + \sigma^2_{\mathtt{EDN}}\, R(\mathtt{EDN} - \mathtt{EDN}')\big\} \times R(t - t') \quad (14)$$

where R denotes correlation functions, and the additional parameter $\sigma^2_{\mathtt{EDN}}$ allows us to alter the relative contribution of the two parts to the variance function: setting $\sigma^2_{\mathtt{EDN}} = 1$ would imply that the two parts contributed equally when AMP was at its upper limit. We have made the additional choice to treat all correlation functions as stationary. We will use the Bernoulli correlation function for t (which is also periodic), and we will use the Matérn correlation function for AMP and PHZ (see (15), below).

4 Specifying prior quantities

In the previous sections we have described our structural choices for the emulator. This involves specifying families of functions from which to select the regressors, and parametric forms for the variance function of the residual. In this section we choose which regressors to use (i.e., which subset of each family to retain), and we choose the parameters of the residual variance function. To do this we make informed choices according to our prior judgements about the emulator, and then we tune these choices according to their impact on predictive diagnostics. We tune these choices because our ensemble of evaluations only has $n = 30$ different values for the inputs variables, and so our prior will undoubtedly affect our updated predictions.

Our prior judgements are not site-specific, but the tuning introduces a site-specific element. Here we describe our tuned prior for the JRO site.

4.1 Predictive diagnostics

We use predictive diagnostics, as they are the most relevant for the purpose to which our emulator will be put: predicting the simulator output at specific choice of values for the model-inputs. We use two types of diagnostic, both represented visually as samples from the updated prediction, with the true value superimposed. First, *leave-one-out* (LOO), in which we predict each evaluation in turn using an emulator constructed from our prior choices and the other evaluations. This shows us how much uncertainty we can expect in our predictions ($n - 1$ being close to n) and how this uncertainty varies across the simulator's input-space. Second, *one-step-ahead* (OSA), in which we predict the first evaluation from the prior emulator, predict the second evaluation after updating by just the first, and so on. This shows us how rapidly we learn about the simulator through accumulating evaluations into the emulator. It is also closely linked to the prequential diagnostic approach (Dawid, 1984; Cowell *et al.*, 1999), and yields a scalar summary quantity that could be used to quantify emulator fit, although we do not do this here, as we find a scalar measure to be too crude a summary of the information we can gather from these diagnostics.

Having a specific ordering in the ensemble is useful for LOO, and affects the result for OSA. We order by the value of `EDN`, as we judge this to have the most complicated impact on the simulator output, particularly at extreme values. In this ordering every panel in the OSA figure is an extrapolation from the convex hull of the evaluations used in the emulator, which makes this a stern test.

Examples of these diagnostics are given in Figures 3 and 4. These reflect some tuning of our prior choices: our tuning has been rudimentary (described further below), and undoubtedly we could improve further, but we judge these to be satisfactory. Thus we are content to stay with our NIG conjugate prior, and not have to generalise. When evaluating each panel, we want to be sure that the actual outcome is consistent with our predictive uncertainty, and that where there appears to be an inconsistency it is not large or systematic. We do not show an example of a 'bad' diagnostic, but during the course of analysing the TIE-GCM ensemble, covering many more choices than have been described in this paper, we have generated hundreds of these figures (i.e., we have built thousands of emulators), including many for which we had made mistakes in our choice of prior or in the code, and we have found that it is easy to spot a bad choice. However, we want to go beyond simply eliminating bad choices: we also want to construct emulators that generally have small predictive uncertainties, and that are parsimonious in their use of regressors. While we could not provide an operational specification of the features of the diagnostics that lead us to our particular choice of emulator, we are satisfied with the result. This type of multi-attribute decision is common in regression analysis, but it has received little attention in emulators, where we have heard it expressed, forcibly, that "a Gaussian process can fit anything". While this may be true with sufficient evaluations, our experience is that our choices are important where there is substantial code uncertainty. That is why we favour tractable emulators, detailed diagnostics, and a high level of subjective interpretation, over statistically more sophisticated—but also more monolithic—approaches, such as MCMC.

Looking in more detail at the two sets of diagnostics, Figure 1 shows that the difficult-to-predict jobs (17, 20 and 30) have high `AMP` values. As discussed at the end of section 5, in retrospect our design might have been better-chosen. It is interesting to note that by about job 28 the panels in the OSA and LOO diagnostic are looking similar, suggesting, informally, that the additional twelve evaluations are having only a minor impact on the emulator's predictions. This might be a useful comparison when doing sequential model evaluations, to provide a simple stopping rule.

4.2 Prior choices

All of the choices that follow represent our judgements about the simulator, with some additional input from the diagnostics. The main choices we face are

1. How many knots in magnetic local time for the outputs, and where to put them;

2. How many EOFs to use (the t basis vectors); and

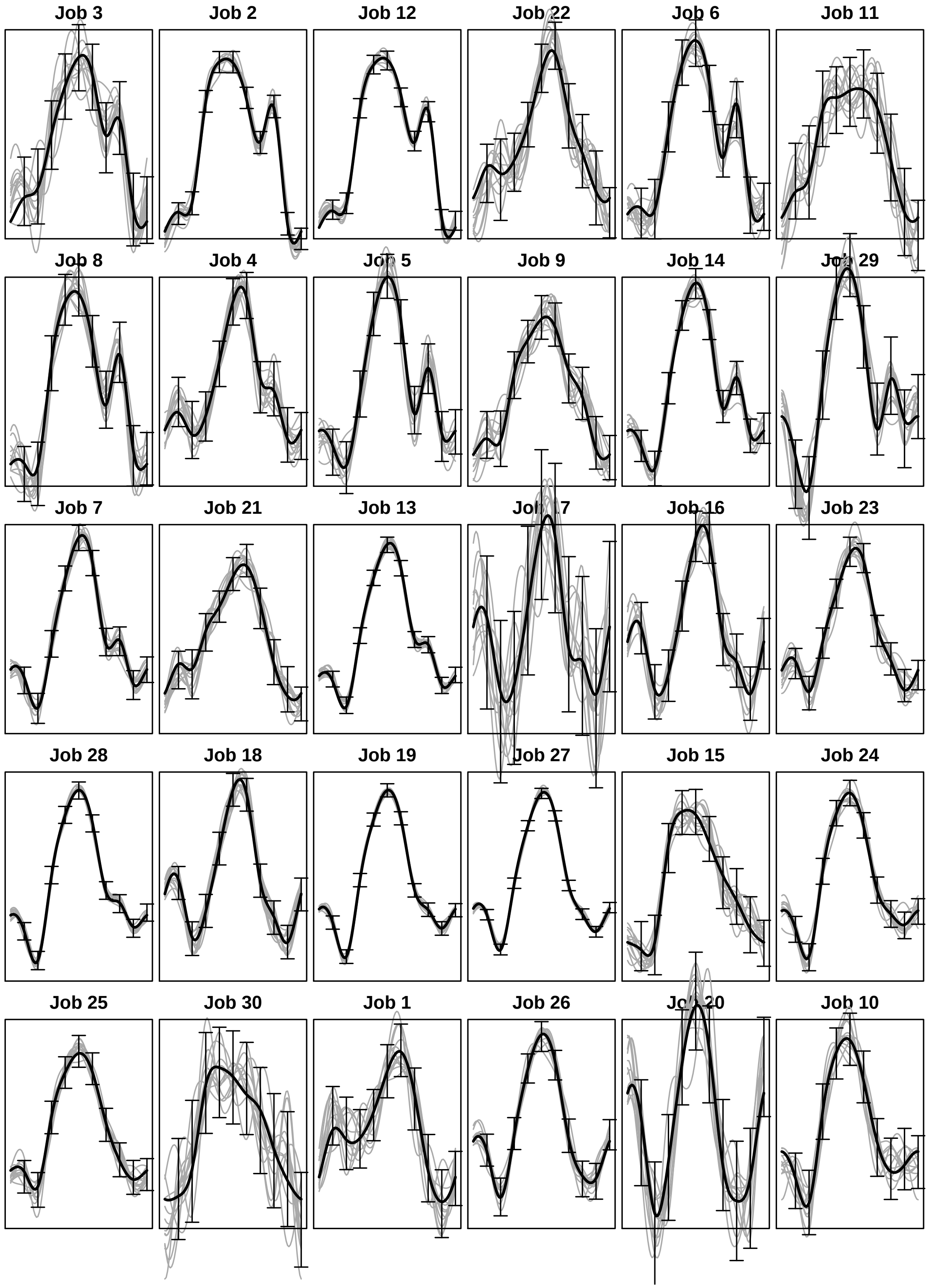

Figure 3: Leave-one-out diagnostic plot, evaluations ordered by EDN (see Figure 1). Each panel shows the simulator output as a periodic function of magnetic local time on the range $[0, 24]$. The error-bars show the predictive mean, plus and minus two standard deviations, the grey lines show samples from the emulator, and the solid black line shows the actual outcome.

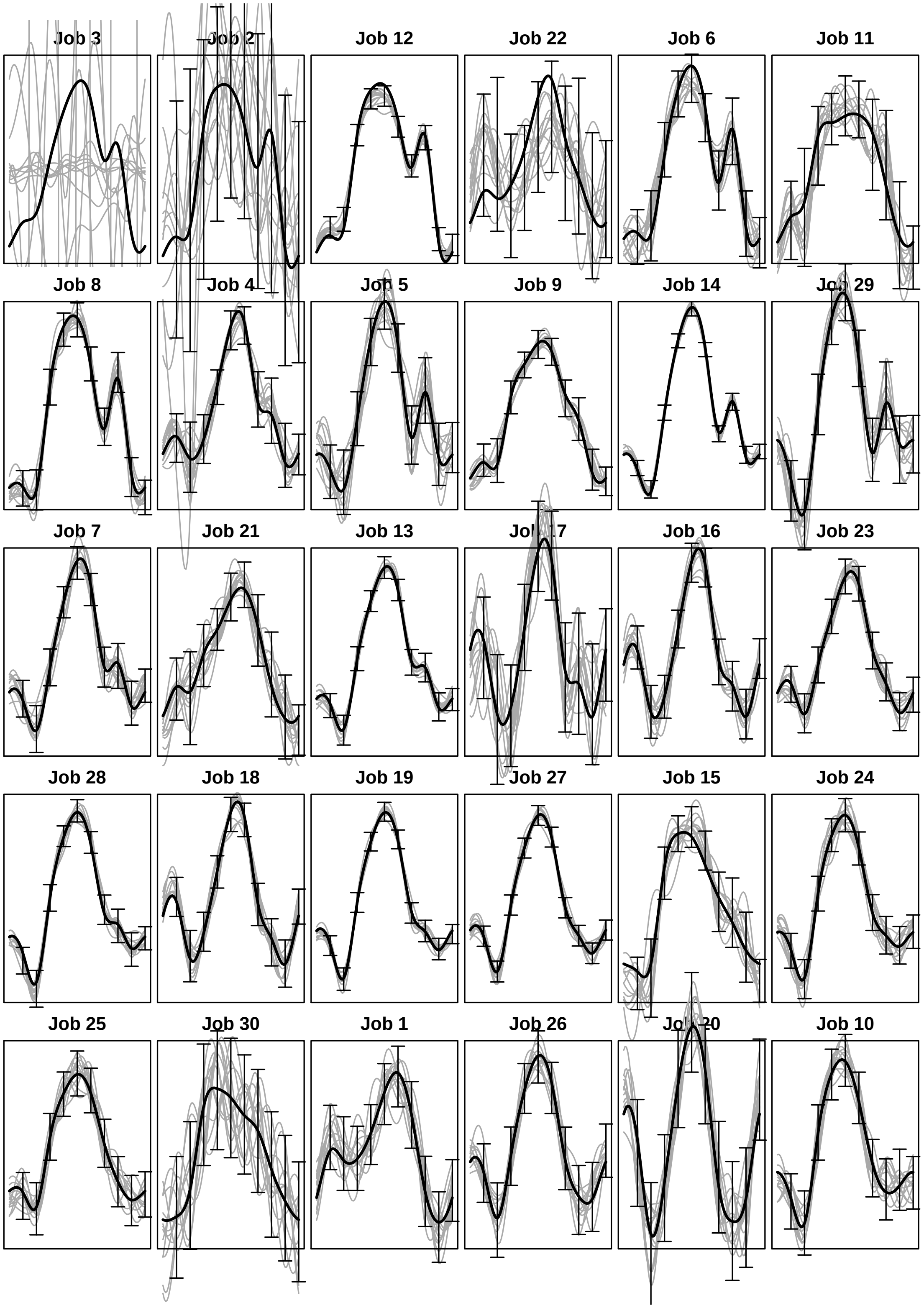

Figure 4: One-step-ahead diagnostic plot, evaluations ordered by EDN. See caption to Figure 3 for details. The samples in the first two panels, Jobs 3 and 2, have been clipped.

3. Which $\boldsymbol{r}$ regressors to use.

Then, following on from this, we need to choose the parameters in the variance function for the residual, eq. (14), and values for the NIG prior hyperparameters, $\{\boldsymbol{m}, V, a, d\}$. Here we describe our choices, with a brief summary of our reasons.

Main choices

We chose 10 knots for the outputs, equally-spaced. We experimented with more and fewer knots, and also with non-equal spacing (we increased the density of knots in the early morning and early evening). The number and location of the knots interacts in a complicated fashion with our other prior choices, due to the highly non-linear operation of computing the EOFs. Ten was enough to capture the main features of the simulator-output with equal spacing.

We chose 6 EOFs, shown in Figure 5. These were computed from the evaluations at the JRO site, and and also the low latitude Arecibo site (ARO, 18.3° N, 66.75° W). We also have simulator output from higher-latitude sites, but we did not include these as the simulator is thought to be less reliable at these sites, due to the high latitude physics which is not captured accurately. While four EOFs appears to capture almost all the variation (measured as the trace of the cross-product), we include two additional EOFs because without them we are not able to capture the simulator's behaviour around the evening excursion.

For the $\boldsymbol{r}$ regressors, we chose a constant and first-order terms only in each of the three inputs, giving six regressors ($q_r = 6$) in total: the outer product of terms of the form $\{1, \texttt{AMP} \times \sin(\texttt{PHZ}), \texttt{AMP} \times \cos(\texttt{PHZ})\}$ and of the form $\{1, \texttt{EDN}\}$. We include interactions between EDN and the other two inputs because we have excluded them from our prior for the residual. We experimented with non-linear terms, but found that they were adequately-handled by the residual. We did not consider excluding any of the linear terms, because, with only 30 evaluations, much of the input-space is an extrapolation from the convex hull of the ensemble, and for this purpose we have more faith in linear regressors than we do in the residual.

Residual variance parameters

The overall form of the variance function is given in (14). We chose $\sigma^2_{\texttt{EDN}} = 1$, so that the variance of the residual is divided equally between the effects of $\{\texttt{AMP}, \texttt{PHZ}\}$ and EDN when AMP is at its upper value (i.e., $\overline{\texttt{AMP}} = 1$): we tried halving and doubling $\sigma_{\texttt{EDN}}$ but this had little impact on the diagnostics.

For AMP and EDN we use a Matérn correlation function, which has the general form

$$R(h; \nu, \theta) = \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{2\sqrt{\nu}\,|h|}{\theta} \right)^{\nu} K_{\nu}\left(\frac{2\sqrt{\nu}\,|h|}{\theta} \right) \tag{15}$$

where $K_\nu(\cdot)$ is the modified Bessel function of order ν; see, e.g., Santner *et al.* (2003, p. 42). The Matérn correlation function allows us to control the smoothness of the sample paths. In both cases we choose $\nu = 2$, which gives sample paths that are differentiable twice, and choose θ to be 1/6 of the range of the input: we assess these choices indirectly by the appearance of the sample paths, shown in the lefthand panel of Figure 6: these are similar to low-order polynomials.

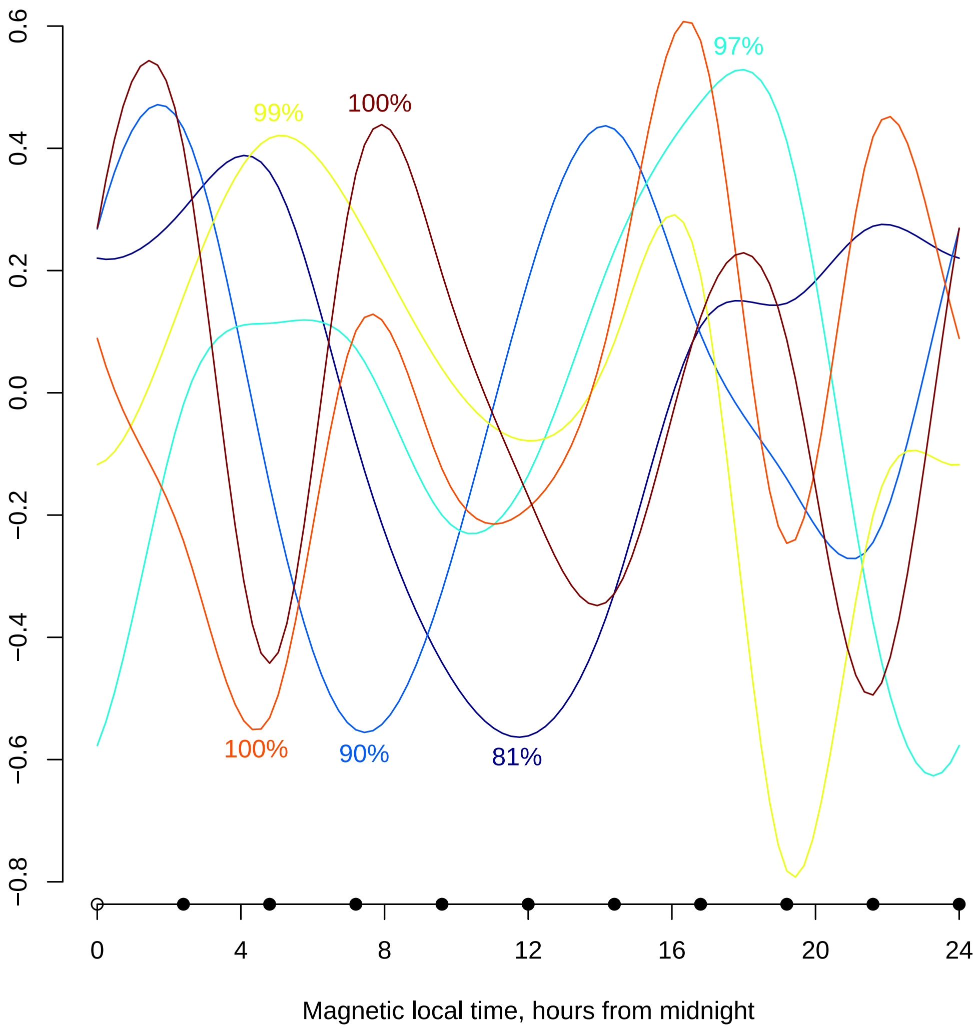

Figure 5: The largest EOFs, computed from model-outputs for the JRO and ARO sites. The percentages represent cumulative explained variation (rounded). The dots on the horizontal axis show the knot locations (equally-spaced, $0 = 24$).

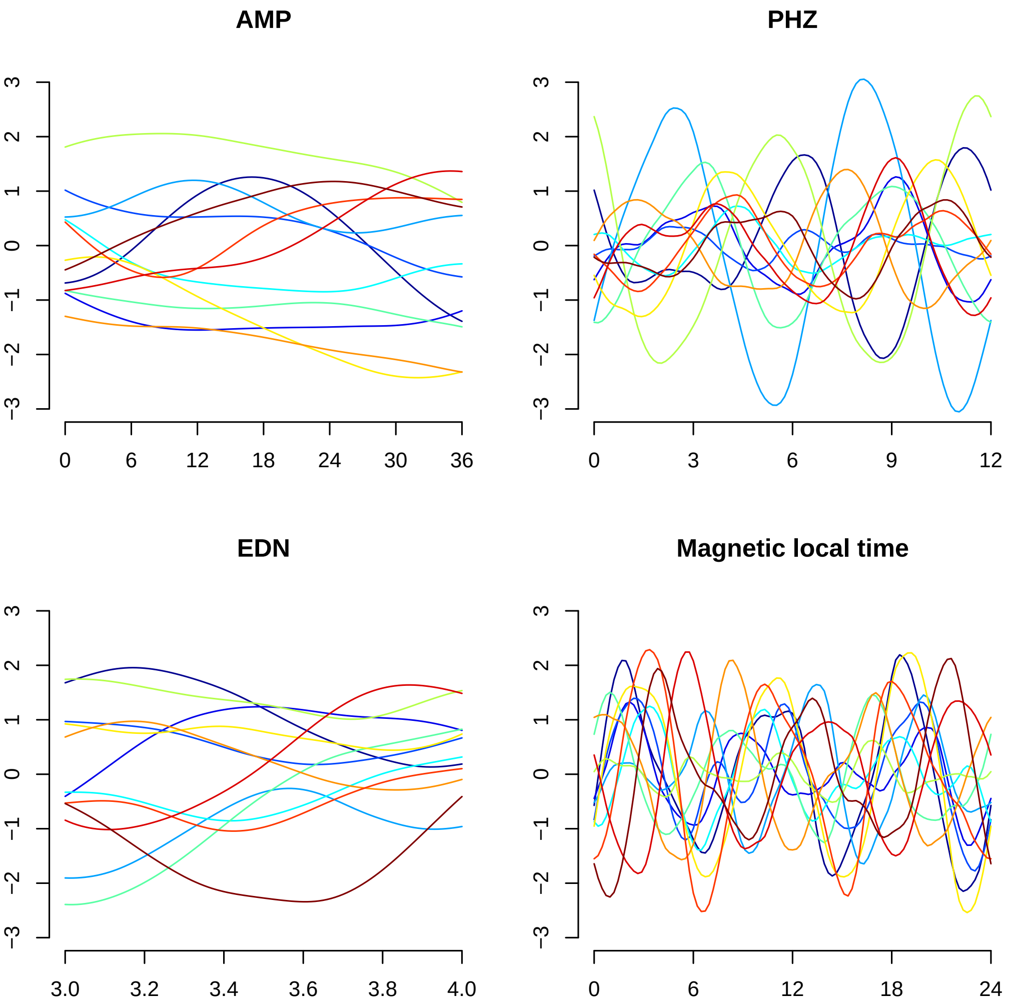

Figure 6: Marginal sample paths for the residual, summarising our choices for the residual variance function. For units see Figure 1; magnetic local time is measured in hours since midnight.

For PHZ and t we use Bernoulli correlation functions (see the Appendix). In both cases we set $m = 3$, controlling differentiability, and $w = 0$, so that the sample paths are orthogonal to the constant in the regressors. For the K parameter, which sets the dominant periodicity, we chose 2 for PHZ and 3 for t. The choice of 2 for PHZ was a natural one, given that the regressors already have first-order effects in PHZ. For t, we judged that deviations from the EOF basis would be fairly local, and so we chose a higher period. Again, we assess our choices indirectly by the appearance of the sample paths, and we performed a sensitivity analysis around our choices.

NIG hyperparameters

Our choices for the NIG hyperparameters $\{\boldsymbol{m}, V, a, d\}$ need to be consistent with our judgements about the simulator output, taking account of the choices we have already made for the regressors and the residual. We choose to express our judgements about $f(\boldsymbol{r}, t)$ in terms of the unconditional distribution of $f(\boldsymbol{r}^*, t^*)$ where $(\boldsymbol{r}^*, t^*)$ has a rectangular distribution. We set $\boldsymbol{m} = \boldsymbol{0}$, so that $\mathrm{E}\big(f(\boldsymbol{r}^*, t^*)\big) = \boldsymbol{0}$. As we have carefully specified our regressors to be orthonormal, $\mathrm{E}\big(g_i(\boldsymbol{r}^*, t^*) g_j(\boldsymbol{r}^*, t^*)\big) = \delta_{ij}$, and

$$\mathrm{Var}\big(f(\boldsymbol{r}^*, t^*) \mid \tau\big) = \tau(\mathrm{tr}\, V + \sigma_e^2) \tag{16}$$

where $\tau\sigma_e^2 \equiv \mathrm{Var}\big(e(\boldsymbol{r}^*, t^*) \mid \tau\big)$. We choose to treat the regression coefficients as exchangeable and conditionally independent, so that $V \equiv \sigma_\beta^2 I_q$, and hence $\mathrm{Var}\big(f(\boldsymbol{r}^*, t^*) \mid \tau\big) = \tau(\sigma_\beta^2\, q + \sigma_e^2)$. We can choose σ_β^2 in terms of a notional prior R-squared, the proportion of the variance attributable to the regressors

$$\text{R-squared} \triangleq \frac{\sigma_\beta^2\, q}{\sigma_\beta^2\, q + \sigma_e^2} \tag{17}$$

where q is determined by our choice of regressors ($q = q_r \times q_t = 6 \times 6$), and σ_e^2 by our choice of parameters for the residual variance function ($\sigma_e^2 = 1.16^2$). We choose an R-squared of 0.75, as we think the regressors will explain about three quarters of the variation in f uniformly over $(\boldsymbol{r}, t)$, which implies that $\sigma_\beta^2 = 0.31^2$.

It remains to choose a and d. For a we choose 1, the smallest value consistent with a proper prior emulator. This implies that the regression coefficients are marginally Cauchy. The very fat tails of these coefficients represents our judgement that only a few of the regressors will be influential. Finally, we choose d according to our judgements about the uncertainty of $f(\boldsymbol{r}^*, t^*)$. We treat $f(\boldsymbol{r}^*, t^*)|\tau$ as approximately Gaussian, in which case

$$\frac{f(\boldsymbol{r}^*, t^*)}{\sqrt{(d/a)(\sigma_\beta^2\, q + \sigma_e^2)}} \tag{18}$$

is approximately Cauchy. Our 95% credible interval for $f(\boldsymbol{r}^*, t^*)$ is $[-30, 30]$, and by equating this to the quantiles of the Cauchy distribution we infer that $d = 1.04$.

To summarise, our choices for the NIG hyperparameters have been determined by our judgements about the explanatory power of the regressors, 'effect sparsity' in the regressors, and uncertainty about the simulator outputs averaged over the inputs.

5 Results: visualising the main-effects and interactions

Here we show one use of our (updated) emulator, visualising the main effects and interactions of the three simulator-inputs, expressed in terms of the periodic output. We show a simple layout in Figure 7, with four values of `EDN`, and, for each value, a low, medium and high value for `AMP` and a low and high value for `PHZ`. By construction, our emulator should generate identical sample paths over different values of `PHZ` when `AMP` $= 0$. We use the values `AMP` $\in \big\{0, 18, 36\big\}$ da m. A second way to check our emulator is to use two values of `PHZ` that are 180 degrees out of phase: in this case we expect to see a reversal of the phase effect; we use `PHZ` $\in \big\{3, 9\big\}$ hr.

In all four panels of Figure 7, which shows the updated mean function for our selected values of the simulator-inputs, we see exactly the relationship between `AMP` and `PHZ` that we anticipated in our prior judgements. The solid and dashed red

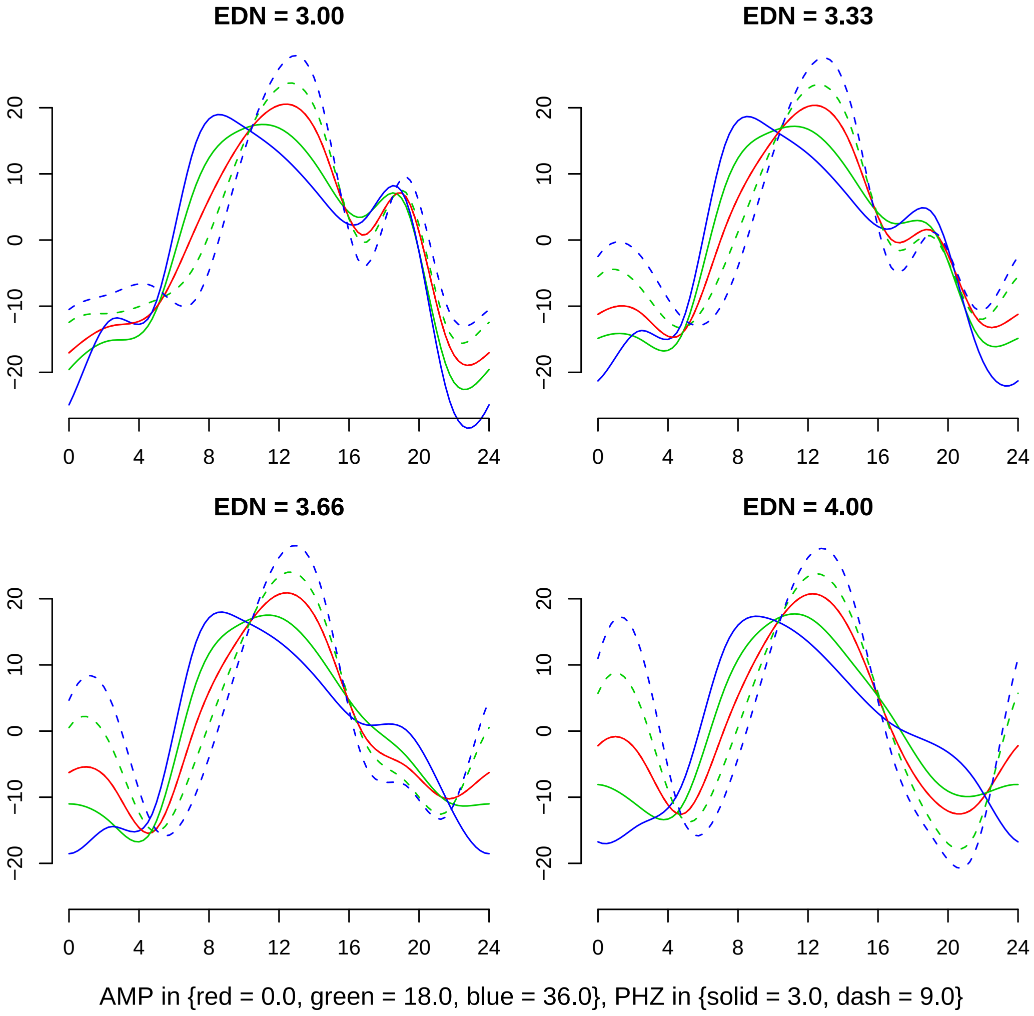

Figure 7: Main effects and interactions among the simulator-inputs for the JRO upwards drift. The horizontal axis shows magnetic local time (hours from midnight), and the vertical axis upwards drift (m/s). For units see Figure 1.

lines coincide, as required, and larger values of AMP are associated with a stronger response to PHZ. We can also see that the two values of PHZ do appear to give outputs that are close to having opposing phases, taking the red AMP $= 0$ line as the control.

Turning to EDN, the relationship uncovered here is entirely driven by the ensemble, as our prior for the effect of EDN was neutral. Our main findings are that higher EDN suppresses the evening excursion, and increases night-time drift. Both these findings are consistent with our qualitative physical understanding. The increase of electron density at night short-circuits the electric field generated in the upper ionosphere (F region), which is responsible for the peak in the early evening, (see, e.g., Eccles, 1998). Therefore the early evening peak disappears with increased night-time electron density. Since the electric-potential drop along the night-time equator from dawn to dusk is basically determined by the day-side electrodynamics, where conductivities are much larger than at night, night-time processes have little effect on the integral of the eastward electric field along the night-time magnetic equator. Therefore, a reduction in the evening upward drift (eastward electric field) must be accompanied by a more-positive (less negative) drift at the other hours of the night. In an earlier experiment we were able to identify a problem with the simulator code because these features were not apparent, highlighting one of the main practical uses of emulators: code verification. We also see that large values of EDN enhance the effect of AMP and PHZ, especially in the night-time; this is an interaction between all four components—the three simulator-inputs and the index variable. The higher variability due to the tides during the night with increased E-region night-time electron density might be due to the fact that the tides are not propagating up to the F region, but are still reaching the E-region ionosphere. Strengthening the E-region electron density and therefore the E-region electrodynamics produces a clearer tidal signal.

The mean function shown in Figure 7 does not tell the whole story, for which we also need the variance function. This is shown in Figure 8, for EDN $= 3.66$. Generally, we are satisfied that the mean function is a good guide to the simulator-response. The most noticeable feature is the greater uncertainty for the AMP $= 36$ predictions. This is a consequence of our emulator structure, and of us not having many evaluations at high AMP values. An important lesson for this type of experiment is that a Latin Hypercube is not necessarily the most effective choice when we have a clear view about how two model-inputs combine. If we were running this experiment again we would arrange to have a higher density of evaluations at higher values of AMP. Likewise, in our follow-up design.

6 Conclusion

This paper has shown that constructing an emulator for a complex simulator involves a large number of choices. This is especially true when our prior emulator is informed by our knowledge of the simulator. The need to validate these choices prioritises the computation of diagnostics, and this in turn constrains the emulator to be tractable. For this reason we have adopted a conjugate Normal Inverse Gamma emulator, and we have made a number of additional structural simplifications. This is really two sides of the same coin: we simplify our statistical modelling to make accessible

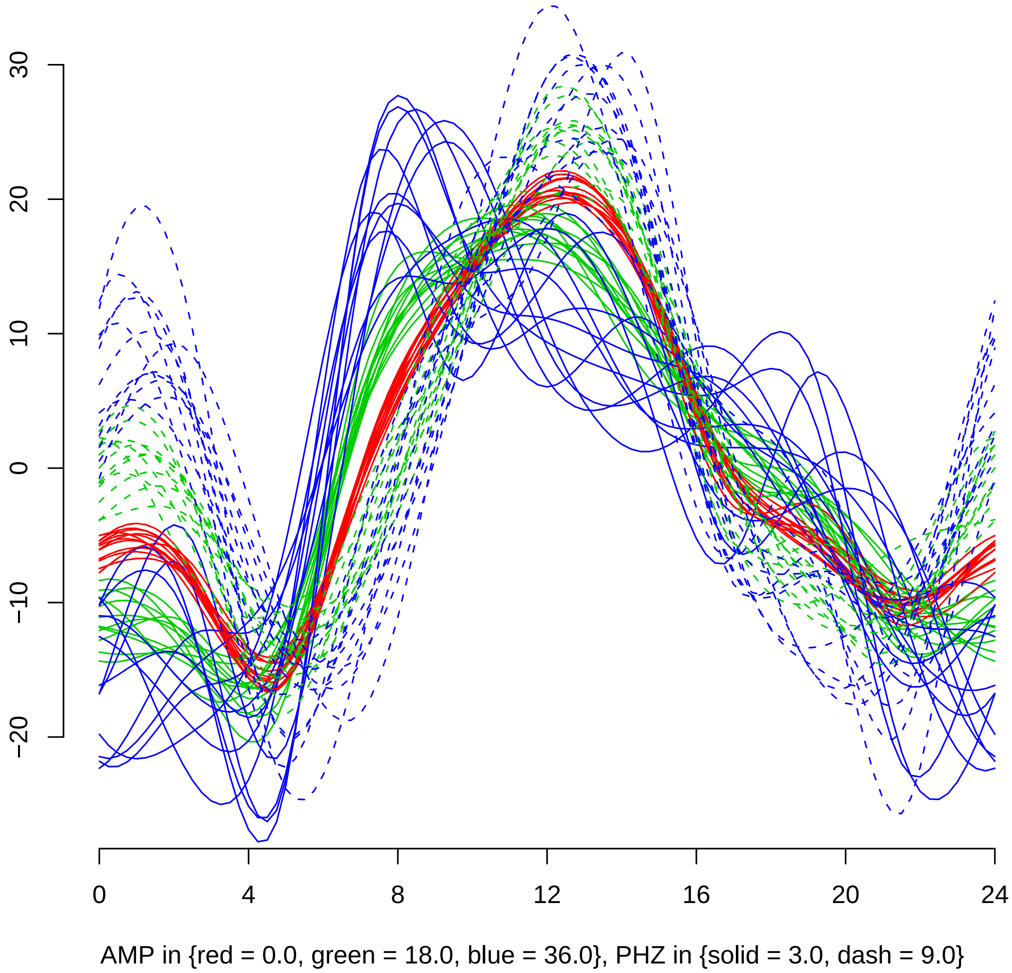

Figure 8: Uncertainty in the updated emulator, for `EDN` = 3.66; *cf.* the bottom lefthand panel in Figure 7.

powerful diagnostic information, and we gain confidence in our simplifications from acceptable diagnostics.

Our emulator has a number of new features. We have shown how periodic functional outputs can be treated, using fixed knots, Empirical Orthogonal Functions, and a periodic interpolator. We have derived a class of periodic correlation functions, which can be specified in terms of smoothness, range and dominant periodicity. We have demonstrated the use of detailed prior information to choose both the regressors and the residual variance function, and to quantify the hyperparameters in our prior emulator. We have presented detailed diagnostic testing of the predictive performance of our emulator. And, finally, we have deployed our emulator to understand better the response of the TIE-GCM simulator to three inputs describing atmospheric tides at the lower boundary.

Acknowledgements

We would like to thank Robert Wolpert and David Steinberg for their very helpful suggestions regarding periodic correlation functions. This work was supported by the Statistical and Applied Mathematical Sciences Institute (SAMSI), and by the National Science Foundation 'Coupling, Energetics, and Dynamics of Atmospheric Regions' and NASA 'Living With a Star' programs.

This material was based upon work supported by the National Science Foundation under Agreement No. DMS-0112069. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

A Periodic correlation functions

We describe a simple construction for a family of periodic correlation functions (i.e., correlation functions with periodic sample paths). We term this the 'Bernoulli' correlation function.

Consider a Gaussian stochastic process with the general form

$$Z(x) = \lambda_0 z_0 + \sum_{k=1}^{\infty} \lambda_k \left\{ z_k \sin(2\pi k x) + z_{-k} \cos(2\pi k x) \right\} \tag{A1}$$

for some scalar x with period 1; the z_i are independent standard Gaussian quantities. Then

$$\begin{aligned} \operatorname{Cov}\bigl(Z(x), Z(y)\bigr) &= (\lambda_0)^2 + \sum_{k=1}^{\infty} (\lambda_k)^2 \bigl\{ \sin(2\pi k x)\sin(2\pi k y) + \cos(2\pi k x)\cos(2\pi k y) \bigr\} \\ &= (\lambda_0)^2 + \sum_{k=1}^{\infty} (\lambda_k)^2 \cos(2\pi k |x-y|) \end{aligned} \tag{A2}$$

where the second line follows the application of standard trigonometric identities. Here it is necessary that the series $\lambda_0, \lambda_1, \ldots$ be squared-summable, in which case Z is a stationary Gaussian process.

If we want Brownian-motion-like sample paths then we require a non-finite number of non-zero λ_k. One attractive choice in this case is to choose the λ_k according to results for Bernoulli polynomials. Fix $m \in \mathbb{N}$. A standard result states (Abramowitz and Stegun, 1965, p. 805)

$$(-1)^{m+1} B_{2m}(|x-y|)/(2m)! = 2/(2\pi)^{2m} \sum_{k=1}^{\infty} k^{-2m} \cos(2\pi k|x-y|) \tag{A3}$$

where B_k is the Bernoulli polynomial of order k. Choose $K \in \mathbb{N}$ and $w \in \mathbb{R}$, and set λ_k according to

$$\lambda_k \triangleq \begin{cases} w & k = 0 \\ 0 & 1 < k < K \\ \sqrt{2}/(2\pi k)^m & k \geq K. \end{cases} \tag{A4}$$

In this case Z is the sum of periodic functions with period not less than K. The variance function is

$$\begin{aligned} V(h) &\triangleq \operatorname{Cov}\big(Z(x), Z(x+h)\big) \\ &= w^2 + \frac{(-1)^{m+1} B_{2m}(|h|)}{(2m)!} - \sum_{i=1}^{K-1} \frac{2}{(2\pi k)^{2m}} \cos(2\pi k|h|) \end{aligned} \tag{A5}$$

and this can be transformed into a correlation function by dividing through by $\sigma^2 \triangleq V(0)$, which will be a function of m, w, and K. From this expression we infer that m determines the *smoothness* of the sample paths, i.e., their differentiability; w affects the range, i.e., the rate at which the correlation falls away in h; and K affects the periodicity.

References

M. Abramowitz and I. Stegun, editors, 1965. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables.* Dover, new edition edition. Available on-line, at `http://www.math.sfu.ca/~cbm/aands/index.htm`.

H. Chipman, M. Hamada, and C.F.J. Wu, 1997. A Bayesian variable-selection approach for analyzing designed experiments with complex aliasing. *Technometrics*, **39**(2), 372–281.

R.G. Cowell, A.P. David, S.L. Lauritzen, and D.J. Spiegelhalter, 1999. *Probabilistic Networks and Expert Systems.* New York: Springer.

C. Currin, T.J. Mitchell, M. Morris, and D. Ylvisaker, 1991. Bayesian prediction of deterministic functions, with application to the design and analysis of computer experiments. *Journal of the American Statistical Association*, **86**, 953–963.

A.P. Dawid, 1984. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A*, **147**(2), 278–290. With discussion, pp. 290–292.

J.V. Eccles, 1998. Modeling investigation of the evening prereversal enhancement of the zonal electric field in the equatorial ionosphere. *J. Geophy. Res.*, **103**(26), 709–26.

C. G. Fesen, G. Crowley, R. G. Roble, A. D. Richmond, and B. G. Fejer, 2000. Simulation of the pre-reversal enhancement in the low latitude vertical ion drifts. *Geophys. Res. Let.*, **27**(13), 1851.

M. Goldstein, 1983. General variance modifications for linear Bayes estimators. *Journal of the American Statistical Association*, **78**, 616–618.

M.E. Hagan and J.M. Forbes, 2002a. Migrating and nonmigrating diurnal tides in the middle and upper atmosphere excited by troposheric latent heat release. *J. Geophys. Res.*, **107**(D24), 4754.

M.E. Hagan and J.M. Forbes, 2002b. Migrating and nonmigrating semidiurnal tides in the middle and upper atmosphere excited by troposheric latent heat release. *J.Geophys. Res.*, **108**(A2), 1062.

D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high dimensional output. Technical Report LA-UR-07-1444, Los Alamos National Laboratory, 2007. To appear in the *Journal of the American Statistical Association*.

A. O'Hagan, 1998. A Markov property for covariance structures. Unpublished, available at http://www.shef.ac.uk/~st1ao/ps/kron.ps.

A. O'Hagan, 2006. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering and System Safety*, **91**, 1290–1300.

A. O'Hagan and J. Forster, 2004. *Bayesian Inference*, volume 2b of *Kendall's Advanced Theory of Statistics*. London: Edward Arnold, 2nd edition.

A.D. Richmond, E.C. Ridley, and R.G. Roble, 1992. A thermosphere/ionosphere general circulation model with coupled electrodynamics. *Geophys. Res. Let.*, **19** (6), 601.

J.C. Rougier, 2007a. Efficient emulators for multivariate deterministic functions. Unpublished, draft available from the author.

J.C. Rougier, 2007b. Lightweight emulators for multivariate deterministic functions. Unpublished, available at http://www.maths.bris.ac.uk/~mazjcr/lightweight1.pdf.

T.J. Santner, B.J. Williams, and W.I. Notz, 2003. *The Design and Analysis of Computer Experiments*. New York: Springer.