University of Kent

School of Economics Discussion Papers

Bayesian Estimation of Large-Scale Simulation Models with Gaussian Process Regression Surrogates

Sylvain Barde

August 2022

KDPE 2203

Bayesian Estimation of Large-Scale Simulation Models with Gaussian Process Regression Surrogates

Sylvain Barde*

August 23, 2022

Abstract

Large scale, computationally expensive simulation models pose a particular challenge when it comes to estimating their parameters from empirical data. Most simulation models do not possess closed form expressions for their likelihood function, requiring the use of simulation-based inference, such as simulated method of moments, indirect inference or approximate Bayesian computation. However, given the high computational requirements of large-scale models, it is often difficult to run these estimation methods, as they require more simulated runs that can feasibly be carried out. This paper aims to address the problem by providing a full Bayesian estimation framework where the true but intractable likelihood function of the simulation model is replaced by one generated by a surrogate model. This is provided by a sparse variational Gaussian process, chosen for its desirable convergence and consistency properties. The effectiveness of the approach is tested using both a Monte Carlo analysis on a known data generating process, and an empirical application in which the free parameters of a computationally demanding agent-based model are estimated on US macroeconomic data.

JEL classification: C14, C15, C52, C63.

Keywords: Bayesian estimation, surrogate methods, Gaussian process, simulation models.

1 Introduction

The increasing availability of computing power has led over time to simulation methods becoming part of the standard toolbox of researchers. Gilbert and Troitzsch (2005) argue that the appeal of these methods within the social sciences, including economics, lie in their enabling a better understanding and formalisation of the non-linearities or emergent phenomena pervasive in social structures. Another motivation is that they allow scenario analysis or simulated experiments to be carried out, conditional on the simulations being a valid representation of the phenomenon of interest. However, establishing this precondition, a process called model validation, is challenging for social science simulations, and particularly for agent-based models (ABMs), which typically simulate the interactions of individual agents. Fagiolo et al. (2007) put this down to the methodological challenges generated by the bottom-up design and emergent properties of ABMs, and their resulting lack of close-form solutions for the evolution of observable variables. Additional challenges include the heterogeneity of practices in their design and conceptual disagreements around how best to validate ABMs.

*School of Economics, Kennedy Building, University of Kent, Park Wood Road, Canterbury CT2 7FS, UK
tel : +44 (0)1 227 824 092, email: s.barde@kent.ac.uk

The author is grateful to participants at the WEHIA 2021, CEF 2021 and CEF 2022 conferences for comments and suggestions. Particular thanks goes to Herbert Dawid, Christophre Georges, Blake LeBaron and Junior Maih for their input on a preliminary version of the manuscript. The author acknowledges the support of the University of Kent for the use of the ICARUS HPC cluster on which the numerical analysis was run. Any errors in the manuscript remain of course the author's.

This paper proposes a Bayesian estimation framework for simulation models that is specifically designed to be tractable on the most computationally demanding models. This will include a demonstration on the Caiani et al. (2016) model, an ABM known for being computationally expensive. The motivation for this focus on ABM validation is that if an estimation methodology works for the hardest models, then it not only allows these specific models to be brought to the data more effectively, but it also improves the state-of-the-art in general. As a result, it is important to emphasise that despite this apparently narrow focus, the method proposed here can have broader application to a wide range of simulation-based models.

Gouriéroux and Monfort (1993, 1996) provide an early overview of simulation-based inference methods, while Fagiolo et al. (2019) review how those approaches have been used, and improved on, over last decade to estimate the parameters of these types of models. Existing methodologies tend to fall into two categories, moment-based and likelihood-based. Both can be considered special cases of the more general indirect inference framework (Gouriéroux et al., 1993; Smith, 1993), differing in the metric chosen for the distance between empirical and simulated auxiliary parameters.[1] Historically, the first method implemented on ABMs is the method of simulated moments (MSM), in Gilli and Winker (2003), which was extended by the simulated minimum distance (SMD) method of Grazzini and Richiardi (2015). Likelihood-based methods involve simulated likelihood, see Kukacka and Barunik (2017) for a recent application, and Bayesian estimation approaches such as Grazzini et al. (2017); Delli Gatti and Grazzini (2020). In the last three cases, kernel density estimation (KDE) is used to obtain a non-parametric estimate of the likelihood function from the model simulations.

Lamperti et al. (2018) make the point that these estimation methods have so far been restricted to relatively simple ABMs, precisely because of the computational cost of running the simulations required to generate either the simulated moments or non-parametric estimates of the likelihood function. Grazzini et al. (2017), whose contribution to Bayesian estimation of simulation models will be discussed further below, consider for instance the computational costs of simulating the data as part of their estimation framework. Their conclusion is that even in the simplest of case of a 1-parameter model, simulation of the model can represent 50% of the overall computational time, with the share increasing sharply with model complexity.

At this point, it is important to provide some working definitions of what will be considered a large-scale model for the purposes of the paper. A first aspect is that it is assumed that there is a limit to the number simulation runs that can be performed, similar to the concept of limited compute budget of Lamperti et al. (2018). This budget can be spent on multiple simulations of the same calibration, as part of a Monte Carlo analysis of model outputs, or on a set of different calibrations, as part of a sensitivity analysis. The rule of thumb used here is that the compute budget allows for 1000 runs, enough to establish the MC frequencies of a model's observables for a typical scenario analysis.[2] The second aspect is the parameter dimensionality of the model, as one should expect a correlation between compute time and number of parameters in the simulation model: a model with a richer set of mechanisms, each requiring dedicated parameters, will typically be computationally more expensive. Furthermore, the curse of dimensionality implies that a higher dimensional parameter space will dilute away this limited compute budget, complicating the exploration of the parameter space. Finally, a high parameter dimensionality will also complicate sampling from the posterior distribution using Bayesian methods such as MCMC, where the threshold for high dimensionality is surprisingly low. Gelman et al. (1997) show that the

[1] See Smith (2016) for a very clear discussion on the three variants of indirect inference based on the choice of metric. The Wald metric is used for moment-based methods, while simulated likelihood methods are linked to the likelihood ratio metric. In fact, the surrogate likelihood method proposed here can also be considered as a form of indirect inference, where the auxiliary model estimated from the simulated data is the GP regression model.

[2] The rationale here is that if a simulation model is so computationally demanding that it is not even possible to establish its statistical properties for a single calibration, then its usefulness is limited.

optimal asymptotic acceptance rate of the random walk Metropolis-Hastings algorithm with respect to the dimensionality of the target distribution is 0.234, however Gelman et al. (1996) show that even in the well-behaved case of a multivariate Gaussian target distribution, this asymptotic behaviour is essentially reached for as few as 6 dimensions.

The strategy proposed in the paper is to generate a surrogate likelihood function for the model, based on Gaussian process (GP) regression, using the limited number of runs in the compute budget to generate training data that spans the parameter space. The use of a surrogate model, or meta-model, to reduce the computational burden of simulation models is not novel in itself. Lamperti et al. (2018), previously mentioned, specifically investigate the use of surrogate models in facilitating ABM calibration. Salle and Yıldızoğlu (2014), Bargigli et al. (2020) and Chen and Desiderio (2022) lie closer to the proposed strategy, all using GP regression to generate a surrogate for an ABM. In these cases, however, the GP surrogate aims to predict simulated moments given a set of model parameters, which can then be used in an MSM estimation. The key efficiency improvement on these approaches, discussed further below, is that by including lagged values of the model variables in the GP regression inputs, one obtains a one-step-ahead predictor of the model outputs themselves. This improves performance by increasing the amount of effective simulation data available for training the surrogate, as well as providing a functional form that can directly be used to generate a surrogate likelihood.

The proposed approach is most closely related to two contributions in the literature. First is the Bayesian estimation framework Grazzini et al. (2017) and Delli Gatti and Grazzini (2020), who use KDE to generate a non-parametric likelihood function from the simulated data. They key departure here lies in choosing a different non-parametric framework for the surrogate likelihood, in order to increase the efficiency of the methodology with respect to the number of simulations required. The second contribution is Platt (2021), who instead generate the surrogate likelihood using a neural network trained on the simulated data. As will be discussed below, this is closest to the methodology proposed here, due to the universal approximation property of neural networks that is shared with GPs.

The remainder of the paper is organised as follows. Section 2 first presents the variational GP regression framework used to generate the surrogate likelihood, section 3 then explains how it integrates within the wider Bayesian estimation framework for simulation models. Section 4 presents the two applications used to test and illustrate the methodology and section 5 concludes.

2 Gaussian process regression and surrogate likelihood

2.1 A one-step-ahead multivariate GP surrogate: setup and notation

The proposed estimation framework requires two distinct Bayesian estimations, the first for estimating the parameters of the GP regression surrogate from the simulated training data, the second for estimating the parameters of the simulation model from the empirical data. Each step will have a prior and a posterior, and because of the risk of confusion this creates, the notation and terminology first needs to be made very clear. An additional concern is that most of the literature on the theoretical properties of GP regression focuses on univariate prediction, while large-scale simulation models will typically be estimated on multivariate data. In general, the notation will follow the GP regression literature wherever possible, in order to facilitate presentation of the theoretical results, but is adapted to deal both with the multivariate nature of the problem as well as the embedding of the GP regression into a wider Bayesian estimation framework.[3] The former involves vectorising over the observable variables and adopting a block-diagonal structure for the covariance matrices, allowing the multivariate GP to be cast as a (larger) univariate process. This makes it straightforward to show that both the derivations of the quantities of

[3]Specifically, the aim is to be consistent with Hensman et al. (2015), Seeger et al. (2008), and Burt et al. (2019).

interest and the properties of the resulting GP regression in the univariate case extend to the multivariate case. To address the second problem, and limit the potential for confusion between the two Bayesian estimation stages, variables and parameters used during the empirical estimation step will be labelled with a superscript $*$.

There are M empirical observable variables, which will be modelled with V latent GP variables, that where necessary will be indexed respectively using a subscript m and v. There are T time series observations for each variable, indexed with subscript t. The simulated data will be generated using S samples $\boldsymbol{\theta}_s$ from the parameter space Θ. Lower case bold letters $\mathbf{x}, \mathbf{y}, \mathbf{u}$, etc. indicate vectors, assumed to be column vectors unless explicitly stated otherwise, while uppercase bold letters $\mathbf{A}, \mathbf{B}, \boldsymbol{\Sigma}$, etc. refer to matrices. $\mathbf{K}$ will specifically refer to the variance-covariance matrix produced by a kernel function, and superscripts attached to kernel matrixes, e.g. $\mathbf{K}^{\mathbf{x},\mathbf{x}}$, will refer to the set of inputs entering the kernel function. Braces are used to refer to the full set of vectors or matrices attached to observable and latent variables, e.g. $\{\mathbf{A}_v\} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_V\}$ is the set of all $\mathbf{A}_v$ matrices attached to the latent variables. Two vectorisation operations are required to convert these sets into single objects. The first is the standard vectorisation operation which stacks multiple column vectors into a single vector, the second constructs a block diagonal matrix from a set of matrices, with $\mathbf{0}$ being an appropriately sized null matrix.

$$\begin{cases} \operatorname{vec}(\{\mathbf{f}_v\}) = \begin{bmatrix} \mathbf{f}_1^T & \mathbf{f}_2^T & \ldots & \mathbf{f}_V^T \end{bmatrix}^T \\ \operatorname{bdiag}(\{\mathbf{A}_v\}) = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{A}_V \end{bmatrix} \end{cases} \tag{1}$$

With these preliminaries out of the way, let $\mathbf{Y}^*$ be a $T \times M$ matrix of empirical data and let $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_S\}$ be a set of S simulated $T \times M$ real-valued matrices, each obtained by simulating the model for T periods using a row vector of d_Θ parameters $\boldsymbol{\theta}_s \in \Theta$.[4] The values in the columns of $\mathbf{Y}_s$ and $\mathbf{Y}^*$ are assumed to be centred. Let $\boldsymbol{\theta}^*$ be a vector of simulation model parameters to be estimated from the empirical data $\mathbf{Y}^*$ using Bayesian methods. Then, ignoring the marginal data density $p(\mathbf{Y}^*)$, the object of interest is the posterior density of the model parameters given the data:

$$p(\boldsymbol{\theta}^* \mid \mathbf{Y}^*) \propto p(\mathbf{Y}^* \mid \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*) \tag{2}$$

The prior probability for the parameter vector, $p(\boldsymbol{\theta}^*)$ is known by definition. Using the chain rule for probabilities, the likelihood term $p(\mathbf{Y}^* \mid \boldsymbol{\theta}^*)$ can be decomposed into a product of individual densities:

$$p(\mathbf{Y}^* \mid \boldsymbol{\theta}^*) = \prod_t p(\mathbf{Y}_t^* \mid \boldsymbol{\Omega}_{t-1}^*, \boldsymbol{\theta}^*) \tag{3}$$

Where $\boldsymbol{\Omega}_t^* = \{\mathbf{Y}_t^*, \mathbf{Y}_{t-1}^*, \ldots \mathbf{Y}_1^*\}$ is the information set at t. Taking logs gives us the log-likelihood, which is the sum of the individual log-likelihood contributions of each observation in the dataset:

$$\ln p(\mathbf{Y}^* \mid \boldsymbol{\theta}^*) = \sum_t \ln p(\mathbf{Y}_t^* \mid \boldsymbol{\Omega}_{t-1}^*, \boldsymbol{\theta}^*) \tag{4}$$

The key challenge facing simulation models is obtaining a predictive density for observation $\mathbf{Y}_t^*$ given knowledge of the previous observations $\boldsymbol{\Omega}_{t-1}^*$ and the model parameters $\boldsymbol{\theta}^*$. As a first simplification we restrict the information set to be $\boldsymbol{\Omega}_t^* = \mathbf{Y}_t$, i.e. we only consider a one-step-ahead prediction, in order to

[4] It is assumed as a simplification that both the simulated and empirical data have T time-series observations, however in general different time-series length can be used for the simulated data relative to the empirical data.

lower the dimensionality of the surrogate model.[5] The second simplification is to approximate the true likelihood (4) with one produced by a surrogate model:

$$\ln p(\mathbf{Y}^* \mid \boldsymbol{\theta}^*) \;\approx\; \ln \hat{p}(\mathbf{Y}^* \mid \boldsymbol{\theta}^*) \;=\; \sum_t \ln p(\mathbf{Y}_t^* \mid \mathbf{Y}_{t-1}^*, \boldsymbol{\theta}^*, \mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\phi}) \tag{5}$$

Where $p(\mathbf{Y}_t^* \mid \mathbf{Y}_{t-1}^*, \boldsymbol{\theta}^*, \mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\phi})$ is the one-step-ahead density for observation $\mathbf{Y}_t^*$, conditional on $\mathbf{Y}_{t-1}^*$ and $\boldsymbol{\theta}^*$ obtained from a surrogate model with internal parameters $\boldsymbol{\phi}$ that has been optimised on a simulated training set $\mathbf{Y}$ obtained with parameter samples $\boldsymbol{\theta}$. An important notational comment is that because any predictive density always depends on $\mathbf{Y}, \boldsymbol{\theta}$ and $\boldsymbol{\phi}$, in line with the GP regression literature, explicit mention of these conditioning variables is dropped. Instead, $\hat{p}$ is used to indicate the use of a surrogate model conditioned on these variables. In order to further simplify the notation and ensure consistency with the GP literature, we transform the conditioning variables $\mathbf{Y}, \mathbf{Y}^*, \boldsymbol{\theta}, \boldsymbol{\theta}^*$ into two sets of inputs $\mathbf{X}, \mathbf{X}^*$. Assuming that $\mathbf{1}_{T-1}$ is a $T-1$ length column vector of ones, letting $\mathbf{Y}_s$ be the simulated observations generated by the model using parameter setting $\boldsymbol{\theta}_s$ and letting $\mathbf{L}$ be the first-order lag operator, the training inputs $\mathbf{X}$ are given by the following block matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{L}\mathbf{Y}_1 & \mathbf{1}_{T-1}\boldsymbol{\theta}_1 \\ \mathbf{L}\mathbf{Y}_2 & \mathbf{1}_{T-1}\boldsymbol{\theta}_2 \\ \vdots & \vdots \\ \mathbf{L}\mathbf{Y}_S & \mathbf{1}_{T-1}\boldsymbol{\theta}_S \end{bmatrix} \tag{6}$$

This results in an $N \times d$ matrix of training inputs, with $N = S(T-1)$ training observations and $d = M + d_\Theta$ dimensions. Similarly, given the empirical dataset $\mathbf{Y}^*$ and a candidate parametrisation $\boldsymbol{\theta}^*$, the set of inputs used for evaluation is obtained as follows:

$$\mathbf{X}^* = \begin{bmatrix} \mathbf{L}\mathbf{Y}^* & \mathbf{1}_{T-1}\boldsymbol{\theta}^* \end{bmatrix} \tag{7}$$

Where necessary, $x_i, x_j, \ldots$ and $x_i^*, x_j^*, \ldots$ and will denote individual rows of $\mathbf{X}$ and $\mathbf{X}^*$ respectively, i.e. individual observations, and $\mathcal{X}$ denotes the d-dimensional input space from which x_i and x_i^* are drawn. It is also convenient to define $\mathbf{y} = \text{vec}(\mathbf{Y} \setminus y_1)$ and $\mathbf{y}^* = \text{vec}(\mathbf{Y}^* \setminus y_1^*)$ as the vectorisations of the simulated and empirical data matrices, with the first row removed to account for the time lag.

The input data structures (6) and (7) make explicit that the unit of observation used in the likelihood (5) is the $\mathbf{Y}_{t-1}^* \to \mathbf{Y}_t^*$ transition and clarify the role of the GP as a one-step ahead surrogate predictor for the simulation model. During the training phase, the GP parameters $\boldsymbol{\phi}$ are optimised to obtain the best fit of the GP predictions to the simulated data $\mathbf{Y}_{s,t}$ given the inputs x, i.e. the lagged observations $\mathbf{Y}_{s,t-1}$ and parameter settings $\boldsymbol{\theta}_s$. Once the GP surrogate has been trained, it can be used in the second stage to approximate the predictive density of the simulation model for $\mathbf{Y}_{t+1}^*$ given an empirical observation $\mathbf{Y}_t^*$ and a parameter vector $\boldsymbol{\theta}^*$. The additional benefit of this input structure is that it maximises the amount of simulated data available for the GP surrogate, as all simulated observations provide a training data point, barring the T lost by taking a time lag. This is an improvement over those existing analyses that use the GP to provide a surrogate for simulated moments, and must therefore average the simulated data $\mathbf{Y}_s$ over the T dimension in order to obtain the simulated moments for $\boldsymbol{\theta}_s$.

In order to handle the multivariate nature of the surrogate as well as help reduce the computational burden of the GP, we use a Linear Model of Coregionalization (LMC) to generate predictions for M observable variables based on V latent variables, using a $M \times V$ weights matrix $\mathbf{B}$. Each of the V latent variables will be modelled using a standard univariate GP.[6] Given $\mathbf{f} = \text{vec}(\{\mathbf{f}_v\})$, a N-length column

[5] In principle, any number of lags can be included, however this will increase the dimensionality of the input space. This trade-off is discussed further in section 3.1

[6] See Alvarez et al. (2012). Because there will typically be $V < M$ latent variables, this has the additional advantage of

vector of predictions from V latent variables $\mathbf{f}_v$ and $\tilde{\mathbf{B}} = \mathbf{B} \otimes \mathbf{I}_N$, the vectorised representation of the LCM on the training data ($\mathbf{X}$, $\mathbf{y}$) is given by:

$$\mathbf{y} = \tilde{\mathbf{f}} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{\Sigma}^2) \tag{8}$$

The measurement error on each variable m is assumed to be I.I.D. Gaussian with standard deviation σ_m and uncorrelated across variables, so that the variance of the vectorised error term ϵ is given by $\mathbf{\Sigma}^2 = \text{bdiag}\left(\{\sigma_m^2 \mathbf{I}_N\}\right)$. In general, assuming a set of arbitrary gaussian distributions $\mathbf{f}_v \sim \mathcal{N}(\boldsymbol{\mu}_v, \mathbf{\Omega}_v)$ for the V latent predictions, the distribution of the LCM prediction is $\tilde{\mathbf{f}} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{\Omega}})$, with the mean and variance given by the following expressions, where $\boldsymbol{\mu} = \text{vec}(\{\boldsymbol{\mu}_v\})$, $\mathbf{\Omega} = \text{bdiag}\left(\{\mathbf{\Omega}_v\}\right)$, and $\tilde{\mathbf{B}} = \mathbf{B} \otimes \mathbf{I}_N$:

$$\begin{cases} \tilde{\boldsymbol{\mu}} = \tilde{\mathbf{B}}\boldsymbol{\mu} \\ \tilde{\mathbf{\Omega}} = \tilde{\mathbf{B}}\mathbf{\Omega}\tilde{\mathbf{B}}^T = \sum_v \left(\mathbf{B}_v \mathbf{B}_v^T \otimes \mathbf{\Omega}_v\right) \end{cases} \tag{9}$$

The second equality for the prediction variance $\tilde{\mathbf{\Omega}}$ corresponds to the more traditional LCM representation provided in Alvarez et al. (2012), and will be useful for proving that existing results on the consistency of univariate GPs readily extend to the multivariate setting. As a result of this, the conditional distribution of the target data $\mathbf{y}$ given the predictions from the LCM $\tilde{\mathbf{f}}$ is simply:

$$p(\mathbf{y} \mid \tilde{\mathbf{f}}) \;=\; \mathcal{N}(\mathbf{y} \mid \tilde{\boldsymbol{\mu}}\,,\; \mathbf{\Sigma}^2) \tag{10}$$

Each latent prediction component $\mathbf{f}_v$ is given by a GP and follows a multivariate Gaussian distribution:

$$\mathbf{f}_v \sim \mathcal{N}(0, \mathbf{K}_v^{\mathbf{x},\mathbf{x}}), \quad [\mathbf{K}_v^{\mathbf{x},\mathbf{x}}]_{i,j} = K_v(x_i, x_j) \tag{11}$$

Where $K_v(.,.)$ is a kernel function used for modelling the covariance of the v^{th} latent variable. The proposed methodology uses the following squared exponential kernel, also know in the literature as the radial basis function (RBF) kernel, where ℓ_v is the kernel length scale specific to each latent variable:

$$K_v(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\ell_v^2}\right) \tag{12}$$

A key feature of the LCM is that in addition to extending GP regression from a univariate to a multivariate setting, the combination of V different kernels, each capturing correlations in the training data at different length scales ℓ_v, enable flexible modelling of the underlying data-generating process. This set of length scales $\{\ell_v\}$ will play an important role in the discussion of the properties of the GP surrogate in section 2.2. With $\mathbf{K}^{\mathbf{x},\mathbf{x}} = \text{bdiag}\left(\{\mathbf{K}_v^{\mathbf{x},\mathbf{x}}\}\right)$, the mean and variance of the LCM predictions on the training data $\mathbf{X}$ are:

$$\begin{cases} \tilde{\boldsymbol{\mu}} = 0 \\ \tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} = \tilde{\mathbf{B}}\mathbf{K}^{\mathbf{x},\mathbf{x}}\tilde{\mathbf{B}}^T \end{cases} \tag{13}$$

It is important to clarify here that the while the LCM kernel, which produces $\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}$, is constructed as a linear combination of RBF kernels (12), it nevertheless is a valid kernel.[7] Given these elements, the likelihood of the evidence for the LCM model is given by:

$$\hat{p}(\mathbf{y}) = \int p(\mathbf{y} \mid \tilde{\mathbf{f}}) p(\tilde{\mathbf{f}}) d\mathbf{f} = \mathcal{N}\left(\mathbf{y} \mid 0, \tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \mathbf{\Sigma}^2\right) \tag{14}$$

reducing the overall number of parameters to estimate, at the cost of introducing an extra hyperparameter, i.e. the number of latent variables to use. The choice of hyperparameter is discussed in section 3.1.

[7] This is because both re-scaled kernels and sums of kernels are themselves valid kernels, as explained for instance in Rasmussen and Williams (2006).

Marginalising out the vectorised GP latents $\mathbf{f}$ results in the following log-likelihood for the training data:

$$\ln \hat{p}(\mathbf{y}) = -\frac{NM}{2}\ln(2\pi) - \frac{1}{2}\mathbf{y}^T\left(\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2\right)^{-1}\mathbf{y} - \frac{1}{2}\ln|\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2| \tag{15}$$

The log-likelihood of the evidence (15) is a function of the LCM and latent GP parameters $\boldsymbol{\phi} = \{\mathbf{B}, \ell_v, \sigma_m\}$, which implies that optimal values for $\boldsymbol{\phi}$ can be obtained by maximisation. A key property of this expression is that it can be interpreted as a penalised regression, as maximising (15) with respect to the GP parameters $\boldsymbol{\phi}$ is equivalent to minimising a combination of the squared deviation of the GP prediction (0 in this case) from the data and a log-determinant penalisation term enforcing smoothness of the GP prediction.[8] This auto-regularisation property forms a crucial motivation for choosing GP regression as the basis for the surrogate likelihood, and is discussed further in section 2.2.

Once the optimal GP parameters $\boldsymbol{\phi}$ have been obtained from the training data, the LCM surrogate can be used to make predictions $\tilde{\mathbf{f}}^*$ for previously unseen inputs $\mathbf{X}^*$. The GP nature of the surrogate implies that these predictions and the training data jointly follow a multivariate gaussian:

$$p(\mathbf{y}, \tilde{\mathbf{f}}^*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{y} \\ \tilde{\mathbf{f}}^* \end{bmatrix} \middle| \, 0, \begin{bmatrix} \tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2 & \tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}^*} \\ \tilde{\mathbf{K}}^{\mathbf{x}^*,\mathbf{x}} & \tilde{\mathbf{K}}^{\mathbf{x}^*,\mathbf{x}^*} \end{bmatrix}\right) \tag{16}$$

The distribution the predictions conditional on the training data $p(\tilde{\mathbf{f}}^* \mid \mathbf{y}) = \mathcal{N}\left(\tilde{\mathbf{f}}^* \mid \tilde{\boldsymbol{\mu}}^*, \tilde{\boldsymbol{\Omega}}^*\right)$ can be obtained using Schur's complement, with the prediction mean and variance functions given by:

$$\begin{cases} \tilde{\boldsymbol{\mu}}^* = \tilde{\mathbf{K}}^{\mathbf{x}^*,\mathbf{x}}\left(\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2\right)^{-1}\mathbf{y} \\ \tilde{\boldsymbol{\Omega}}^* = \tilde{\mathbf{K}}^{\mathbf{x}^*,\mathbf{x}^*} - \tilde{\mathbf{K}}^{\mathbf{x}^*,\mathbf{x}}\left(\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2\right)^{-1}\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}^*} \end{cases} \tag{17}$$

The LCM with GP latent variables outlined above possesses key properties, discussed below, that make it desirable for the computationally constrained setting. It also has the drawback that given N training observations, the covariance matrix $\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2$ is $N \times N$. Because the matrix inversions required to obtain the GP parameters $\boldsymbol{\phi}$ from the likelihood (15) and make predictions (17) from the evaluation inputs $\mathbf{X}^*$ are $\mathcal{O}(N^3)$, exact GP regression infeasible for large N. Section 2.3 will present the variational approximation approach used to get around this problem.

2.2 Theoretical properties of the multivariate GP surrogate

The machine learning literature contains a range of existing results on the theoretical properties of GPs, and the purpose of this section is to discuss how these motivate the choice of GP regression as the basis of the surrogate. One additional result that is provided is an extension of the convergence result of Seeger et al. (2008) from univariate GP regression to the multivariate LCM (9).

The main motivation for this choice is the universal approximation property of GP regression. In broad terms, this means that the mean function of the GP (9) can approximate arbitrarily well any given continuous function within a bounded interval. This property is clearly desirable in our setting, where the quality of the surrogate model's prediction will be a concern. This property stems from the fact that GP regression is a kernel method, which means that before presenting the results of interest it is important to expand on those theoretical properties of kernels that are relevant to universal approximation.[9]

[8] In fact, Bishop (2006, sections 3.1.4 and 6.4) shows using the duality of the Gram matrix that GP regression is equivalent to ridge regression when the kernel is linear, i.e. $K(x_i, x_j) = x_i^T x_j$.

[9] The discussion in this section only covers those concepts required for the properties of interest, for a more complete discussion of the properties of kernels in the context of GP regression see chapter 6 of Rasmussen and Williams (2006) or Kanagawa et al. (2018).

Mercer's theorem establishes that a positive semi-definite kernel such as (12) possesses an eigenfunction expansion:

$$K(x_i, x_j) = \sum_{h=0}^{\infty} \lambda_h \psi_h(x_i) \psi_h(x_j) \tag{18}$$

Where $\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are the eigenvalues associated to a set of orthogonal eigenfunctions $\psi_h(.)$. This implies that the variance-covariance matrices produced by evaluating the kernel (11) on the N input observations in $\mathbf{X}$ can be written as:

$$\mathbf{K}^{\mathbf{x},\mathbf{x}} = \lim_{H \to \infty} \mathbf{\Psi}_H^{\mathbf{x}} \mathbf{\Lambda}_H (\mathbf{\Psi}_H^{\mathbf{x}})^T \tag{19}$$

$\mathbf{\Psi}_H^{\mathbf{x}}$ is an $N \times H$ matrix where the columns are the values produced by the h^{th} eigenfunction on the N data points and $\mathbf{\Lambda}_H$ is an $H \times H$ diagonal matrix containing the corresponding eigenvalues λ_h. The kernel is assumed to be a Hilbert-Schmidt operator, therefore $\sum_h \lambda_h^2 < \infty$, ensuring convergence of (19). Let $\mathcal{H}$ be the Hilbert space of all functions f defined as linear combinations of these eigenfunctions using weights f_h:

$$f(x) = \sum_h f_h \lambda_h^{\frac{1}{2}} \psi_h(x) \tag{20}$$

Given a second function g similarly defined as $g(x) = \sum_h g_h \lambda_h^{\frac{1}{2}} \psi_h(x)$, the inner product of this Hilbert space is defined as $\langle f, g \rangle_{\mathcal{H}} = \sum_h f_h g_h$. From this it is possible to see that $\mathcal{H}$ meets the two criteria required of a reproducing kernel Hilbert space (RKHS):

$$\begin{cases} \langle K(., x_i), K(., x_j) \rangle_{\mathcal{H}} = \sum_h \lambda_h^{\frac{1}{2}} \psi_h(x_i) \lambda_h^{\frac{1}{2}} \psi_h(x_j) = K(x_i, x_j) \\ \langle f(.), K(., x_i) \rangle_{\mathcal{H}} = \sum_h f_h \lambda_h^{\frac{1}{2}} \psi_h(x_i) = f(x_i) \end{cases} \tag{21}$$

The functions $f \in \mathcal{H}$ are linear combinations of the orthogonal basis formed by the eigenfunctions, subject to $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}} < \infty$, i.e. the Hilbert norm of the function f being finite. The relevance of the RKHS to this discussion is that the predicted mean function of a GP regression $\mu(x_i^*)$ on an unseen data point x_i^*, such as (17), is a linear combinations of the eigenfunctions (20) and is therefore in the RKHS. This property underpins both consistency and convergence of GP regression.

Let $\mathcal{Z}$ be a compact subset of $\mathcal{X}$ and $C(\mathcal{Z})$ be the space of continuous functions over $\mathcal{Z}$. Given $f_0 \in C(\mathcal{Z})$ the first important question that the universal approximation property aims to address is whether there exists a function $f \in \mathcal{H}$ that approximates f_0 arbitrarily well, i.e. $\|\mathbf{f} - \mathbf{f}_0\|_\infty < \varepsilon$. This was answered in the affirmative by Micchelli et al. (2006), who prove that this property holds for certain types of kernels, referred to as *universal* kernels, and in fact that $\mathcal{H} = C(\mathcal{Z})$ for universal kernels. With regards to the methodology proposed here, Wynne et al. (2021) first show that a closed bounded interval $\mathcal{Z} \subseteq \mathbb{R}^d$ meets the conditions required for GP regression to be consistent. Second, theorem 17 of Micchelli et al. (2006) proves that the RBF kernel (12) is universal, and theorem 12 of Caponnetto et al. (2008) proves that a multivariate kernel of the form $\mathbf{M} \times K(x_i, x_j)$ is universal if and only if K is itself universal and $\mathbf{M}$ is positive definite. For the LCM model (9), $\mathbf{M} = \mathbf{B}_v \mathbf{B}_v^T$, therefore the resulting LCM kernel, which forms the basis of the surrogate model, possesses the universal approximation property on a bounded subset of $\mathbb{R}^d$.

The next consideration is convergence, i.e. whether the mean GP prediction (17) in fact converges to the true data-generating function f_0 as the size of training data set $\mathbf{X}$ increases. A wide range of asymptotic consistency results in the literature establish that this is the case, for example Choi and

Schervish (2007); Shi and Choi (2011). Recent work by Koepernik and Pfaff (2021) on the extension of the consistency results of GP regression to non-Euclidian domains providing a very good overview of existing results for the Euclidian case, which this works falls under. For the case analysed here, Le Gratiet and Garnier (2015) prove almost sure convergence for Mercer Kernels (18) that are non-degenerate, i.e. $\lambda_h > 0$, and have bounded features $\|\psi_h(x)\|_\infty < \infty$.

However, given the chosen setting of computational constrained large-scale simulation models with a potentially low number of parameter space samples, the bigger concern is the convergence rate of the surrogate predictions to the true process. In addition, because exploration of the parameter space will be carried out using a pseudo-random design of experiment, it is desirable to have some guarantees that the surrogate prediction is not too sensitive to the potentially sparse realisations of the design. Seeger et al. (2008) argue that in such a setting, the concept of interest is information consistency, where one examines the asymptotic behaviour of the expected Kullback-Leibler divergence from the surrogate to the noise distribution (10).[10] The following bound is obtained using the standard formula for the KL divergence between two Gaussians:[11]

$$D_{KL}\big[p(\mathbf{y} \mid \tilde{\mathbf{f}}) \parallel \hat{p}(\mathbf{y})\big] \quad \leq \quad \frac{1}{2}\tilde{\boldsymbol{\mu}}^T \left(\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2\right)^{-1} \tilde{\boldsymbol{\mu}} + \frac{1}{2} \ln \frac{|\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2|}{|\boldsymbol{\Sigma}^2|} \tag{22}$$

As discussed above, the mean predictions of the GP regression $\tilde{\boldsymbol{\mu}}$ are in the RKHS $\tilde{\mathcal{H}}$ of the LCM kernel corresponding to $\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2$. The reproducing property means that the first term is simply the Hilbert norm of the prediction, $\|\tilde{\boldsymbol{\mu}}\|^2_{\tilde{\mathcal{H}}}$, and is therefore bounded. The second term, known as the regret, is equal up to an additive constant to the penalisation term in the log-likelihood (15):

$$R = \frac{1}{2} \ln \left|\mathbf{I}_{MN} + \boldsymbol{\Sigma}^{-2}\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}\right| \tag{23}$$

With $\nu(x)$ the distribution of the input data, the GP prediction is informationally consistent if $N^{-1}E_{\nu(x)}\Big[D_{KL}\big[p(\mathbf{y} \mid \tilde{\mathbf{f}}) \parallel \hat{p}(\mathbf{y})\big]\Big] \to 0$ as $N \to \infty$, i.e. the expected KL divergence (22) per training draw from $\mathcal{X}$ goes to zero as the size of the training set increases. Crucially, because the first term is bounded $\|\tilde{\boldsymbol{\mu}}\|^2_{\tilde{\mathcal{H}}} < \infty$, it vanishes asymptotically and the scaling behaviour of the regret term (23) alone controls the information consistency of GP regression. Several results have established an upper bound on the growth of the regret term of $\mathcal{O}(\ln N)^{d+1}$ for univariate GP regression with a d-dimensional training space $\mathcal{X}$, including Lemma 1 of Seeger et al. (2008). Proposition 1, below, shows that this bound on expected regret $E_{\nu(x)}[R]$ extends in a straightforward way to the multivariate LCM case.

Proposition: *The expected regret of the LCM (23) has the following upper bound, where v^* indicates the latent GP variable possessing the largest regret:*

$$E_{\nu(x)}\Big[\ln\left|\mathbf{I}_{MN} + \boldsymbol{\Sigma}^{-2}\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}\right|\Big] \quad < \quad MV \sum_{h=0}^{\infty} \ln\left(1 + V b^2_{v^*,m^*} \sigma^{-2}_{m^*} \lambda_{v^*,h} N\right)$$

Proof: This is provided in appendix A.

Corollary: *Let the dataset observations x be distributed with a density $\nu(x) = \mathcal{N}(0, 4a^2\mathbf{I}_d)$ for a constant a. Then the expected regret of the LCM (23) with RBF kernels (12) on $\mathbf{X}$ is $\mathcal{O}((\log N)^{d+1})$.*

This is immediate from the fact that the bound on the LCM from the main proposition is the same, up to some multiplicative constants, as the one in Seeger et al. (2008) for the case of a single RBF kernel

[10] The literature on GP regression consistency is divided between work that uses an integrated mean square error (IMSE) loss, such as the previously cited Choi and Schervish (2007); Shi and Choi (2011); Le Gratiet and Garnier (2015); Koepernik and Pfaff (2021), and works that use the Kullback-Leibler loss, such as Seeger et al. (2008); Burt et al. (2019). The two approaches can in fact be reconciled, see Van Der Vaart and Van Zanten (2011).

[11] This is expressed as an inequality as it ignores an additional term containing the trace of $\left(\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} + \boldsymbol{\Sigma}^2\right)^{-1} \boldsymbol{\Sigma}^2$. This trace is strictly negative given that $\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}$ is positive definite.

function. Assuming the same distribution for the inputs, the LCM regret with RBF kernels will therefore will scale with N at the same rate as the regret of an single RBF kernel. A point to note is that as one would expect, the regret of the LCM kernel (23) will be larger than the single kernel case examined in Seeger et al. (2008), due to the fact that V copies of the worst-performing kernel enter the regret term, which itself is multiplied by MV. Nevertheless, given that V and M are fixed for any given problem, the scaling behaviour with respect to the number of training observations N will be the same as that of a single RBF kernel. A key practical implication, discussed in section 3.1, is that the convergence of the LCM learning curve will therefore tend to be determined by the regret of the kernel associated with the smallest length scale ℓ_{v^*}. Intuitively, if the output variable $\mathbf{y}$ changes rapidly over small ranges of $\mathcal{X}$, a small length scale will be required to accurately capture this, and for a given amount of training data N randomly spread over $\mathcal{X}$ the prediction error will be larger than that of a smoother model, where the inputs can be captured with larger length scales.

In addition to ensuring consistency and uniform convergence, the uniform approximation property also enables a comparison with the neural network approach of Platt (2021), mentioned in the introduction. The key justification provided by Platt (2021) for using neural networks as a surrogate is that they also possess the universal approximation property, as proven by Hornik et al. (1989). This similarity is not accidental, as a key result by Neal (1996) establishes that a GP can be considered as an infinitely wide neural network.[12] Specifically, this result establishes that assuming a zero-mean gaussian prior for the network parameters, a single hidden layer neural network will converge under Bayesian learning to a GP as the number of hidden units goes to infinity. This lines up with the RKHS representation of a GP discussed above, where any smooth function can be approximated arbitrarily well by an infinite weighted sum over kernel eigenfunctions. However, even if using a single layer neural network is functionally equivalent to using a GP in terms of the flexibility of the approximation, Neal (1996) points out an important practical difference between the methodologies in terms of over-fitting. The width of a neural network is a hyperparameter that needs to be carefully optimised as part of the design of the network, precisely to avoid over-fitting the training data, whereas as explained in section 2.1, the likelihood (15) used to optimise the GP parameters is self-regularising, which limits over-fitting.[13] What is lost in the trade-off when choosing a GP regression rather than a neural network is the extra flexibility available from using multiple hidden layers in a deep network, for instance being able to model discontinuities.[14] However, given the nature of the setting assumed here, that of a computational constraint on the generation of training data, the design choice is to prefer a simpler, auto-regularising model that generalises well from potentially limited training data.

Finally, there are some additional properties that are worth mentioning. First, similar to Grazzini et al. (2017) and Platt (2021), the methodology is designed to minimise the computationally expensive runs of the simulation model: the two-step design of the methodology fixes the computational budget for the simulation model ex-ante, in line with Lamperti et al. (2018), and the resulting surrogate model can be re-used when estimating the simulation model parameters on different empirical datasets. Another desirable feature is the analytical tractability of GPs, specifically that the Gaussian form of the predictions ensures differentiability of the resulting surrogate likelihood. This specific aspect, which will be covered further in section 2.4, facilitates maximisation of, and sampling from, the posterior through the use of gradient-based methods. Last of all, is the computational tractability of the method. While it is

[12] This argument is referenced for instance in chapter 6.4 of Bishop (2006), the preface of Rasmussen and Williams (2006) and the introduction of Burt et al. (2019)

[13] This over-fitting problem also occurs to some extend with KDE methods, where the bandwidth of the KDE is also a parameter that needs to be set with the aim of avoiding over-fitting. Optimal bandwidth selection rules are known to exist, however.

[14] This trade-off can be alleviated using deep GP regression which essentially builds a network of GP layers, combining the benefits of both approaches. This is beyond scope of this paper, but details can be found in Damianou and Lawrence (2013).

not feasible to obtain exact GP predictions (17) for large N, due to the $\mathcal{O}(N^3)$ nature of the matrix operations required, variational inference can be used to obtain very close approximations to the exact GP predictions that scale much more favourably with N.

2.3 Scaling up: a variational approximation to the LCM

In order to reduce the computational cost of learning the GP surrogate parameters and performing predictions on the empirical data, we use the sparse variational approach of Titsias (2009) and Hensman et al. (2015). In this approach a small number additional observations, known as inducing points, are introduced to augment the training observations. In terms of notation, these inducing points correspond to a set of inputs $\{\mathbf{Z}_v\}$, known as inducing locations, which generate model predictions $\{\mathbf{u}_v\}$, known as inducing values. Importantly, $\{\mathbf{Z}_v\}$ and $\{\mathbf{u}_v\}$ are not known *ex ante*, instead are additional parameters of the GP that need to be estimated during the training stage.

The joint distribution of the training predictions $\mathbf{f}_v$ and inducing values $\mathbf{u}_v$ associated with a given latent variable is now given by the following multivariate gaussian:

$$p(\mathbf{f}_v, \mathbf{u}_v) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f}_v \\ \mathbf{u}_v \end{bmatrix} \middle| \, 0, \begin{bmatrix} \mathbf{K}_v^{\mathbf{x},\mathbf{x}} & \mathbf{K}_v^{\mathbf{x},\mathbf{z}} \\ \mathbf{K}_v^{\mathbf{z},\mathbf{x}} & \mathbf{K}_v^{\mathbf{z},\mathbf{z}} \end{bmatrix}\right) \tag{24}$$

Increasing the number of observations at the training stage might appear to go against the aim of reducing the computational complexity, however the block construction of the joint variance-covariance matrix means that Shur's complement can again be used to provide the latent predictions $\mathbf{f}_v$ conditional on the inducing values $\mathbf{u}_v$:

$$p(\mathbf{f}_v \mid \mathbf{u}_v) \;=\; \mathcal{N}(\mathbf{f}_v \mid \mathbf{A}_v\mathbf{u}_v,\; \mathbf{K}_v^{\mathbf{x},\mathbf{x}} - \mathbf{Q}_v^{\mathbf{x},\mathbf{x}}) \tag{25}$$

With:

$$\begin{cases} \mathbf{A}_v = \mathbf{K}_v^{\mathbf{x},\mathbf{z}} \left(\mathbf{K}_v^{\mathbf{z},\mathbf{z}}\right)^{-1} \\ \mathbf{Q}_v^{\mathbf{x},\mathbf{x}} = \mathbf{K}_v^{\mathbf{x},\mathbf{z}} \left(\mathbf{K}_v^{\mathbf{z},\mathbf{z}}\right)^{-1} \mathbf{K}_v^{\mathbf{z},\mathbf{x}} \end{cases} \tag{26}$$

The key gain in computational terms is that obtaining the conditional distribution for $\mathbf{f}_v$ now only requires inverting $\mathbf{K}_v^{\mathbf{z},\mathbf{z}}$, which by design is much smaller than $\mathbf{K}_v^{\mathbf{x},\mathbf{x}}$. The marginal distribution of the inducing values, which serves as the GP prior on the values themselves is simply:

$$p(\mathbf{u}_v) \;=\; \mathcal{N}(\mathbf{u}_v \mid 0, \mathbf{K}_v^{\mathbf{z},\mathbf{z}}) \tag{27}$$

The conditional (25) and prior distributions (27) for each latent variable v are assumed to be independent of each other, which means that the equivalent joint distributions for the vectorised values $\mathbf{f} = \text{vec}(\{\mathbf{f}_v\})$ and $\mathbf{u} = \text{vec}(\{\mathbf{u}_v\})$ can be expressed as follows:

$$\begin{cases} p(\mathbf{f} \mid \mathbf{u}) \;=\; \mathcal{N}(\mathbf{f} \mid \mathbf{A}\mathbf{u},\; \mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}) \\ p(\mathbf{u}) \;=\; \mathcal{N}(\mathbf{u} \mid 0, \mathbf{K}^{\mathbf{z},\mathbf{z}}) \end{cases} \tag{28}$$

Where $\mathbf{A} = \text{bdiag}\left(\{\mathbf{A}_v\}\right)$, $\mathbf{K}^{\mathbf{x},\mathbf{x}} = \text{bdiag}\left(\{\mathbf{K}_v^{\mathbf{x},\mathbf{x}}\}\right)$ and $\mathbf{Q}^{\mathbf{x},\mathbf{x}} = \text{bdiag}\left(\{\mathbf{Q}_v^{\mathbf{x},\mathbf{x}}\}\right)$. Together with the conditional distribution of the training targets (10), one can define the joint distribution of $\mathbf{f}$, $\mathbf{u}$ and $\mathbf{y}$ as follows.

$$p(\mathbf{f}, \mathbf{u}, \mathbf{y}) = p(\mathbf{y} \mid \mathbf{f})p(\mathbf{f} \mid \mathbf{u})p(\mathbf{u}) \tag{29}$$

As pointed out by Titsias (2009), the problem at this stage is that because of the block structure of

the multivariate gaussian (24), if one attempts to marginalise away the inducing values $\mathbf{u}$ to obtain a likelihood containing $\mathbf{f}$ alone, as was done for (14), one obtains a marginal density which does not depend on the inducing locations $\mathbf{Z}_v$ (via the kernel matrices $\mathbf{K}_v^{\mathbf{x},\mathbf{z}}$ and $\mathbf{K}_v^{\mathbf{z},\mathbf{z}}$). This means that one cannot directly maximise the marginal likelihood with respect to $\mathbf{Z}$ and $\mathbf{u}$. Instead, the true joint distribution (29) is approximated by a variational distribution that, when marginalised, is still a function of $\mathbf{Z}$ and $\mathbf{u}$. Specifically, this joint variational distribution $q(\mathbf{f}, \mathbf{u})$ is obtained by combining (25) with a prior distribution on $\mathbf{u}$, with prior mean $\mathbf{m} = \text{vec}(\{\mathbf{m}_v\})$ and covariance $\mathbf{S} = \text{bdiag}(\{\mathbf{S}_v\})$.

$$q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u})q(\mathbf{u}), \qquad q(\mathbf{u}) = \mathcal{N}(\mathbf{u} \mid \mathbf{m}, \mathbf{S}) \tag{30}$$

The variational prior $q(\mathbf{u})$ depends on an additional set of parameters $\{\mathbf{m}_v, \mathbf{S}_v\}$, which need to be estimated on the training data, on top of the inducing locations $\mathbf{Z}_v$. This is done by ensuring that (30) approximates the following true posterior probability of the model predictions and inducing variables given the training data:

$$p(\mathbf{f}, \mathbf{u} \mid \mathbf{y}) = p(\mathbf{f} \mid \mathbf{u}, \mathbf{y})p(\mathbf{u} \mid \mathbf{y}) \tag{31}$$

Comparing (30) and (31), one can see that $q(\mathbf{f}, \mathbf{u}) \approx p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})$ when $p(\mathbf{f} \mid \mathbf{u}) \approx p(\mathbf{f} \mid \mathbf{u}, \mathbf{y})$ and $q(\mathbf{u}) \approx p(\mathbf{u} \mid \mathbf{y})$. As pointed out in Titsias (2009), this happens when the inducing points defined by $\{\mathbf{Z}, \mathbf{u}\}$ are sufficient statistics for the full set of training observations $\{\mathbf{X}, \mathbf{y}\}$, and conditioning on $\mathbf{y}$ in addition to $\mathbf{u}$ brings no additional information. This observation also encapsulates the intuition behind sparse variational GP regression, which is to reduce the computational burden of training a standard gaussian process on $\{\mathbf{X}, \mathbf{y}\}$ by instead learning a much smaller set of sufficient statistics which parametrise the variational distribution used to approximate the full GP.

Formally, the objective is to minimise the Kullback-Leibler divergence from $q(\mathbf{f}, \mathbf{u})$ to $p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})$.

$$D_{KL}\big[q(\mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})\big] = -\int_{\mathbf{u}} \int_{\mathbf{f}} \ln \frac{p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})}{q(\mathbf{f}, \mathbf{u})} q(\mathbf{f}, \mathbf{u}) \, d\mathbf{f} \, d\mathbf{u} \tag{32}$$

Rearranging this expression provides the evidence lower bound (ELBO), which is the standard objective function used in the variational GP literature.

$$\mathcal{L}(\boldsymbol{\phi}) \;=\; \int_{\mathbf{u}} \int_{\mathbf{f}} \ln \frac{p(\mathbf{f}, \mathbf{u}, \mathbf{y})}{q(\mathbf{f}, \mathbf{u})} q(\mathbf{f}, \mathbf{u}) \, d\mathbf{f} \, d\mathbf{u} \;=\; \ln p(\mathbf{y}) - D_{KL}\big[q(\mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})\big] \tag{33}$$

Given that the log of the evidence $\ln p(\mathbf{y})$ does not depend on the variational parameters $\boldsymbol{\phi}$, maximising the ELBO (33) is equivalent to minimising the KL divergence (32). In addition, because the KL divergence is strictly non-negative, maximising the (33) involves maximising a lower bound on the evidence $\ln p(\mathbf{y})$. Assuming that the variational distribution $q(\mathbf{f}, \mathbf{u})$ successfully approximates (31), the KL divergence will tend to zero, and maximising $\mathcal{L}(\boldsymbol{\phi})$ becomes equivalent to maximising the evidence of the exact GP (15). A more tractable expression for $\mathcal{L}(\boldsymbol{\phi})$ can be obtained by evaluating the integral in (33), thus marginalising out the model predictions on the training data $\mathbf{f}$ and the inducing values $\mathbf{u}$.[15]

$$\mathcal{L}(\phi) \;=\; \ln \mathcal{N}(\mathbf{y} \mid \tilde{\mathbf{B}}\mathbf{A}\mathbf{m}, \boldsymbol{\Sigma}^2) - \frac{1}{2} \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} \tilde{\mathbf{B}} \boldsymbol{\Psi} \tilde{\mathbf{B}}^T \boldsymbol{\Sigma}^{-1}\right) - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u})\big] \tag{34}$$

Where $\boldsymbol{\Psi}$ is the variational variance-covariance matrix of the vectorised latent variables $\mathbf{f}$:

$$\boldsymbol{\Psi} = \mathbf{A}\mathbf{S}\mathbf{A}^T + \mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}} \tag{35}$$

The ELBO $\mathcal{L}(\boldsymbol{\phi})$ is a function of the variational GP's parameters $\boldsymbol{\phi} = \{\mathbf{Z}, \mathbf{m}, \mathbf{S}, \mathbf{B}, \boldsymbol{\ell}, \boldsymbol{\sigma}\}$, and can

[15] This derivation is provided in appendix B.1 for the case of the LCM.

therfore be optimised with gradient search. A key attraction of this formulation, pointed out by Hensman et al. (2015), is that it can be done stochastically using random sub-samples of the training data $\{\mathbf{X}, \mathbf{y}\}$, therefore further improving tractability. The consistency and convergence of this variational approximation is proven by Burt et al. (2019, 2020) by extending the analysis of Seeger et al. (2008). They show that for the RBF kernel, as long as the number of inducing variables scales as $\mathcal{O}((\log N)^d)$, the average KL divergence of the variational approximation, $N^{-1}D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u})\big]$, converges to zero and $E\left[\mathcal{L}(\phi)\right] \rightarrow E\left[\ln p(\mathbf{y})\right]$ This confirms that as long as enough inducing locations are used, the use of a variational approximation instead of an exact GP does not affect the informational consistency of the methodology.

2.4 GP prediction and surrogate likelihood

Once the GP has been trained on the simulated data, i.e. the variational GP parameters $\boldsymbol{\phi}$ have been obtained, the GP prediction for unseen input $\mathbf{X}^*$ can be obtained using the variational distribution of $\mathbf{f}$. This is obtained by marginalising the inducing values $\mathbf{u}$ out of the joint variational distribution (30).[16]

$$q(\mathbf{f}) = \mathcal{N}\left(\mathbf{f} \mid \mathbf{A}\mathbf{m}, \boldsymbol{\Psi}\right) \tag{36}$$

Given the AR(1) time series structure assumed for the surrogate likelihood (5) and the fact that the predictions are already conditioned on lagged observations (7), the contribution of each $(\mathbf{y}_{t-1}, \mathbf{y}_t)$ transition should enter the likelihood independently. As a result, only the main diagonal of the variance-covariance matrix $\boldsymbol{\Psi}$ is required to obtain the variational density of the LCM predictions $\mathbf{f}^*$.

$$q(\tilde{\mathbf{f}}^*) = \mathcal{N}\left(\tilde{\mathbf{f}}^* \mid \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Psi}}\right) \tag{37}$$

Where given $\text{diag}(\boldsymbol{\Psi})$, a diagonal matrix containing the main diagonal of $\boldsymbol{\Psi}$, the mean and variance-covariance are:

$$\begin{cases} \tilde{\boldsymbol{\mu}} = \tilde{\mathbf{B}}\mathbf{A}\mathbf{m} \\ \tilde{\boldsymbol{\Psi}} = \tilde{\mathbf{B}} \times \text{diag}(\boldsymbol{\Psi})\tilde{\mathbf{B}}^T \end{cases} \tag{38}$$

Given (37) the surrogate likelihood of the empirical $\mathbf{y}^*$ data using the LCM is:

$$\hat{p}(\mathbf{y}^* \mid \boldsymbol{\theta}^*) = \int p(\mathbf{y}^* \mid \tilde{\mathbf{f}}^*) q(\tilde{\mathbf{f}}^*) d\mathbf{f}^* = \mathcal{N}\left(\mathbf{y}^* \mid \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Psi}} + \boldsymbol{\Sigma}^2\right) \tag{39}$$

As explained in section 2.2, a key benefit of the approach is that it is straightforward to derive the gradient of the surrogate likelihood with respect to the underlying simulation model parameters $\boldsymbol{\theta}^*$. Because these form part of the inputs $\mathbf{X}^*$ used in the GP prediction, they only enter the surrogate likelihood (39) through the RBF kernel (12) used to compute $\mathbf{K}_v^{\mathbf{x}^*,\mathbf{z}}$.

$$\frac{\partial \mathbf{K}_v^{\mathbf{x}^*,\mathbf{z}}}{\partial \boldsymbol{\theta}_i^*} = \frac{\left(\mathbf{Z}_i - \boldsymbol{\theta}_i^*\right)\left(\mathbf{1}_d\right)^T}{\ell_v^2} \odot \mathbf{K}_v^{\mathbf{x}^*,\mathbf{z}} \tag{40}$$

The gradient of the surrogate likelihood (39) can be obtained by applying the chain rule on the standard derivative of a Gaussian likelihood:

[16] Details are provided in appendix B.2.

$$\begin{aligned}\frac{\partial \ln \hat{p}(\mathbf{y} \mid \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}_i^*} = -\frac{1}{2}\operatorname{tr}\Bigg(&\left(\tilde{\boldsymbol{\Psi}} + \boldsymbol{\Sigma}^2\right)^{-1} \frac{\partial \tilde{\boldsymbol{\Psi}}}{\partial \boldsymbol{\theta}_i^*} + (\mathbf{y} - \tilde{\boldsymbol{\mu}})(\mathbf{y} - \tilde{\boldsymbol{\mu}})^T \left(\tilde{\boldsymbol{\Psi}} + \boldsymbol{\Sigma}^2\right)^{-1} \frac{\partial \tilde{\boldsymbol{\Psi}}}{\partial \boldsymbol{\theta}_i^*} \left(\tilde{\boldsymbol{\Psi}} + \boldsymbol{\Sigma}^2\right)^{-1}\Bigg) \\ &- (\mathbf{y} - \tilde{\boldsymbol{\mu}})^T \left(\tilde{\boldsymbol{\Psi}} + \boldsymbol{\Sigma}^2\right)^{-1} \frac{\partial \tilde{\boldsymbol{\mu}}}{\partial \boldsymbol{\theta}_i^*}\end{aligned} \tag{41}$$

The derivatives of the LCM mean and variance functions (38) are:

$$\begin{cases} \dfrac{\partial \tilde{\boldsymbol{\mu}}}{\partial \boldsymbol{\theta}_i^*} = \tilde{\mathbf{B}} \dfrac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_i^*} \mathbf{m} \\ \dfrac{\partial \tilde{\boldsymbol{\Psi}}}{\partial \boldsymbol{\theta}_i^*} = \tilde{\mathbf{B}} \times \operatorname{diag}\left(\left(\dfrac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_i^*}\right)^T \left(\mathbf{S} - \mathbf{K}^{\mathbf{z},\mathbf{z}}\right) \mathbf{A} + \mathbf{A}^T \left(\mathbf{S} - \mathbf{K}^{\mathbf{z},\mathbf{z}}\right) \left(\dfrac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_i^*}\right)\right) \tilde{\mathbf{B}}^T \end{cases} \tag{42}$$

Given the definitions of $\mathbf{A}$ in (26), we finally have the following expression, which can be calculated using (40):

$$\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_i^*} = \operatorname{bdiag}\left(\left\{\frac{\partial \mathbf{A}_v}{\partial \boldsymbol{\theta}_i^*}\right\}\right) = \operatorname{bdiag}\left(\left\{\frac{\partial \mathbf{K}_v^{\mathbf{x}^*,\mathbf{z}}}{\partial \boldsymbol{\theta}_i^*} \left(\mathbf{K}_v^{\mathbf{z},\mathbf{z}}\right)^{-1}\right\}\right) \tag{43}$$

In practical terms it is more efficient to use automatic differentiation to carry out the gradient calculations, rather than attempt implementing these derivations directly. These do establish, however, that the gradient of the likelihood exists, is well-behaved and ultimately depends on the gradient of the kernel matrix with respect to the parameters in its input.

3 Bayesian estimation with GP regression surrogate (BEGRS)

Having presented the structure of the GP surrogate, its theoretical properties and the variational implementation required to obtaining it for a large training set, we now turn to the practicality of embedding it in a wider Bayesian MCMC estimation framework.

3.1 Design of experiment and GP surrogate training

The first step, which occurs prior to actually running the estimation methodology, is the generation of the training data $\mathbf{X}$ from the simulation model. This needs to be planned carefully, because as established in section 2.2, the consistency and convergence of the GP surrogate depend in part on the distributional properties of the training data. In addition under the assumption that the simulation model is expensive to run, the opportunity cost of running a poor design will be high.

Several key decisions have to be made at this stage, specifically the bounds required on the underlying parameter space Θ, the choice of design used for sampling from Θ, the number of samples to pick in that design and the length of each simulation. It is important to emphasise that the main input in making these decisions will be the researcher's domain knowledge of the simulation model, possibly augmented by any exploratory sensitivity analyses carried out during the model's development. These will help shed light on an important factor in these decisions, which is the variability of the mean one-step ahead behaviour of the model over $\mathcal{Z}$. As discussed in section 2.2 the size of the LCM regret (23), and therefore the convergence of the predictions, is mainly controlled by ℓ_{v^*}, the smallest length scale of the V latent RBF kernels. In practical terms, if the average one-step ahead model predictions are very sensitive to certain parameter values, a finer sample of points will be required to obtain a reliable surrogate. This will affect either the number of samples required, or the chosen design.[17] Similarly, domain knowledge will

[17] For instance, in cases where a sensitivity analysis is able to identify regions of Θ where the model predictions abruptly

generally determine the choice of the bounds $\left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$ on Θ, required by the methodology.[18] In some cases, the bounds will be obvious from the nature of the parameter involved (such as a share, a ratio, etc.), however the general case, determining these bounds will require knowledge of the model's behaviour.

With regards to picking a specific design for drawing training samples θ from Θ, a large literature already details the available options and their relative benefits, for instance Santner et al. (2018). A few important considerations for the proposed methodology merit discussion. As emphasised by Lamperti et al. (2018), the fact that computational costs are a factor means that any chosen design must have good space-filling properties. A popular category in this respect is the Latin hypercube design (LHD), which Santner et al. (2018) show typically provides very good predictive performance. Salle and Yıldızoğlu (2014) also provide evidence that using the near orthogonal Latin hypercube (NOLH) design of Cioppa and Lucas (2007) improves prediction accuracy of GP surrogates, due to the additional orthogonality property of the samples. This is not necessarily applicable here: given that the inputs in the training data (6) contain both parameter samples $\boldsymbol{\theta}$ and lagged time-series data $\mathbf{Y}$, $\mathbf{X}$ will be not orthogonal if general, even if the parameters samples $\boldsymbol{\theta}$ are. The main drawback of LHD and NOLH approaches, however, is that the design is fixed, and additional samples cannot easily be added ex-post if the initial simulation run is found to be insufficient. Lamperti et al. (2018) advocate instead for the use of Sobol sequences, which can easily be extended while maintaining their space-filling properties. While both Santner et al. (2018) and Liefvendahl and Stocki (2006) show that Sobol designs produce less precise predictions than LHD, due to the greater range of inter-point distances, the practical difference between the two is minimal, especially as S increases. All these considerations combine to motivate the use of a Sobol sequence design in the applications of section 4.

In addition to picking a design, one needs to choose the number of samples S to draw from Θ and the length of the simulated time-series T, which together determine the number of training observations $N = S(T-1)$. Two aims should guide this choice, first N should be as large as possible given the computational constraint, in order to improve the convergence of the surrogate. Second, S should also be as large as possible, to ensure good space-filling of the training samples $\boldsymbol{\theta}$. The strategy adopted for the applications in section 4 is therefore to pick a value of T that is in line with the number of observations in the empirical data, and then infer S from the number of T-length simulations that can feasible be run within the computational budget. This is where a Sobol design can be useful, as an initial design that turns out to have insufficient samples S for the purposes of training the surrogate can be augmented ex-post, without the original simulated data having to be discarded.

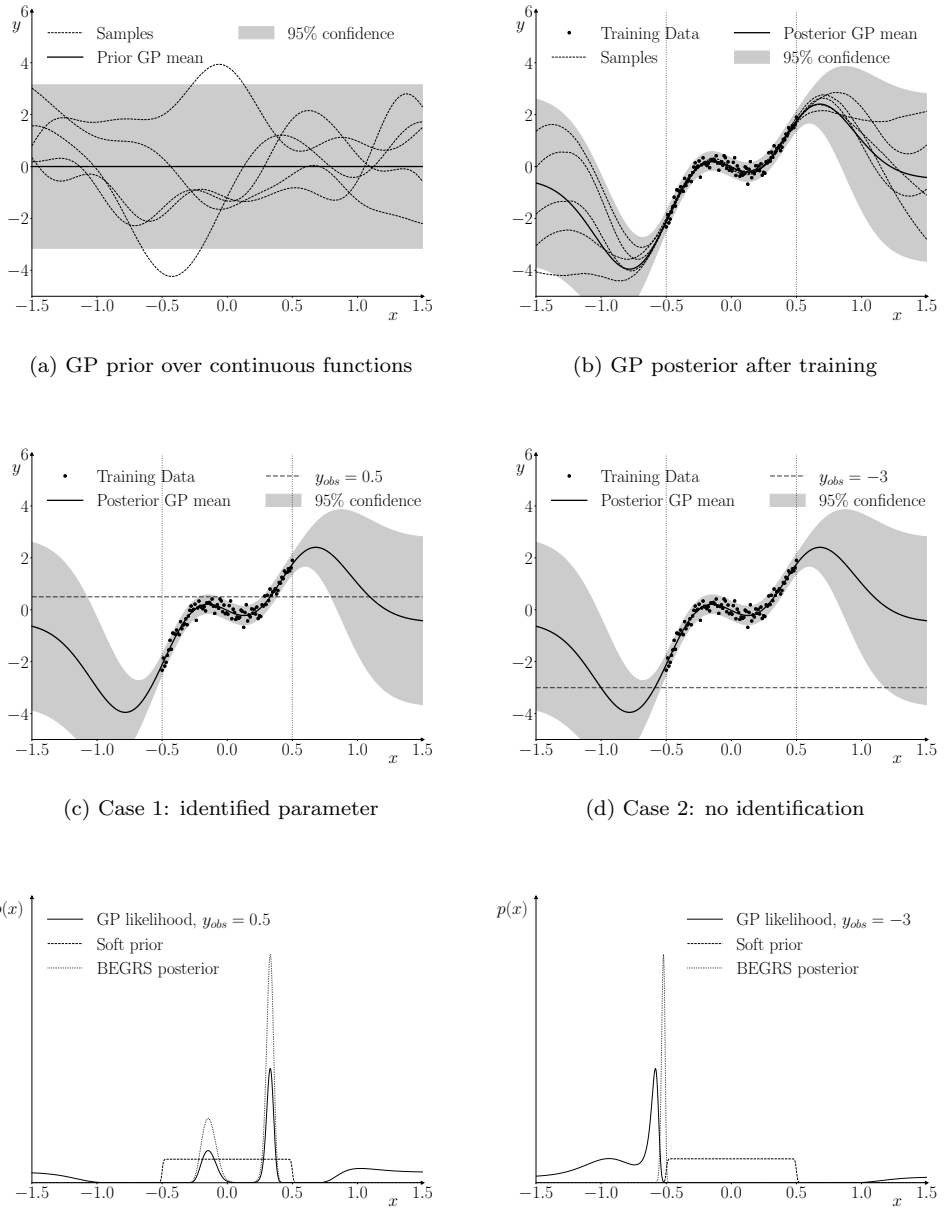
Once the simulated data $\mathbf{Y}$ is generated and has been combined with the parameters samples to form the training data $\mathbf{X}$ (6), the next step is to pick the two main hyperparameters of the variational GP regression framework. These are the number of latent variables V to use as the basis of the multivariate LCM, and the number of inducing points $\mathbf{Z}$ to use in the variational approximation. The number of latent variables can be chosen with the help of a principal component analysis (PCA) or a factor analysis of the simulated data $\mathbf{Y}$. A few clarifying comments are required here, however. Firstly, the structure of the LCM prediction (8) means that $\mathbf{Y}$ is assumed to contain additive, variable-specific, noise on top of the linear combination of latent predictions $\mathbf{f}_v$. This suggests that factor analysis, which allows for this, might be preferable to the strict orthogonal decomposition of $\mathbf{Y}$ provided by PCA. In either case, it is important to note that the loadings obtained via either approach will not, in general match the LCM loadings $\mathbf{B}$. This is because the latent variables of the LCM model are designed to capture the correlations in the training inputs $\mathbf{X}$ at different length scales ℓ_v, a constraint that the latent factors obtained via factor analysis or PCA do not possess. Therefore, these analyses should be used only to

change, it might be beneficial to pick an adaptive design that samples those regions more densely.

[18] As explained in section 2.2, the universal approximation property holds on a compact subset $\mathcal{Z}$ of the full input space $\mathcal{X}$, assumed to be $\mathbb{R}^d$ in our case. Because $\mathcal{X}$ contains Θ, picking $\mathcal{Z} \subseteq \mathcal{X}$ involves picking a bounded interval for Θ.

determine the minimal number of latent variables required to summarize the data.

Less guidance is available regarding the number of inducing points to choose. The $\mathcal{O}((\log N)^d)$ bounds found by Burt et al. (2019, 2020) only offer a scaling guarantee, not a method of calculating the number of inducing points required. In addition, they point out that if the d-dimensional covariates fall on a lower-dimensional manifold embedded in $\mathcal{X}$, then the scaling is driven by the dimensions of the manifold. This is likely to be the case here, given that $\mathbf{X}$ contains lagged values of the simulated data $\mathbf{Y}$, which itself is assumed to be reducible to a smaller number of latent variables V. However, the variational approximation will converge once a sufficient number of inducing points has been included, after which adding further inducing points no longer improves performance. The practical process suggested in the literature is therefore to run the GP regression several times, with an increasing number of inducing points, and identify the threshold at which adding further points no longer improves the ELBO (34).



(a) GP prior over continuous functions

(b) GP posterior after training

(c) Case 1: identified parameter

(d) Case 2: no identification

(e) Case 1: posterior identification

(f) Case 2: no posterior identification

Figure 1: Illustration of surrogate GP likelihood

3.2 Minimal parameter prior, posterior estimation and identification issues

The ultimate purpose of the GP surrogate is to provide a low-cost approximation of the simulation model's likelihood $p(\mathbf{y}^*|\boldsymbol{\theta}^*)$, given empirical data $\mathbf{y}^*$ and a candidate parameter vector $\boldsymbol{\theta}^*$, thus enabling the use of Bayesian methods to determine the posterior $p(\boldsymbol{\theta}^*|\mathbf{y}^*)$. The availability of the gradient of the surrogate likelihood (41) means that as long as the prior for the parameter vector $\boldsymbol{\theta}^*$ is differentiable, one can do so using gradient-based methods. This is illustrated in the section 4 applications, where the BFGS algorithm is used to find the posterior mode, and Hamiltonian Monte-Carlo (HMC) - specifically the NUTS algorithm - is used to determine the posterior density.

While the specific choice of parameter prior $p(\boldsymbol{\theta}^*)$ is up to the researcher, the use of a GP regression surrogate imposes a minimal requirement. As explained previously, GP regression converges to the true f_0 in a compact subset of $\mathcal{Z}$ of the wider input space $\mathcal{X}$, requiring in particular a set of bounds $\left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$ for the parameter space Θ. The kernel, however, is defined over all of $\mathcal{X}$, and serves as the GP's prior over the space of all continuous functions $C(\mathcal{X})$. This is illustrated in figure 1(a), where the GP prior is defined over $\mathbb{R}$, whereas in figure 1(b) the posterior functions only converge to the true DGP on the bounded interval in which training data is available. Intuitively, outside of that interval, no training data was observed and the GP prior entirely determines the GP posterior. This results in a surrogate likelihood that is defined over all of $\mathbb{R}^{d_\Theta}$ but trained only for values of $\boldsymbol{\theta} \in \left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$. This property is the reason why the parameter prior $p(\boldsymbol{\theta}^*)$ in the second stage must restrict the analysis to the $\left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$ bounds, and will also produces an very distinctive identification failure, which is discussed further below.

In principle, a continuous uniform distribution defined over $\left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$ is sufficient to ensure that a Bayesian estimation produces estimates for $\boldsymbol{\theta}^*$ within those bounds. In practice, however, the discontinuity at the boundary will create problems for gradient-based algorithms, which would defeat the purpose of having the gradient of the likelihood. Instead, to ensure that the gradient of the prior is defined at the boundary of the parameter space itself, we recommend using a smooth relaxation of the uniform distribution. This can be carried out using the following double sigmoid function, where $\alpha > 0$ controls the slope of the sigmoid at the boundary:

$$p(\boldsymbol{\theta}^*) = \prod_i \frac{1}{1 + e^{-\alpha\left(\boldsymbol{\theta}_i^* - \underline{\boldsymbol{\theta}}\right)}} \frac{1}{1 + e^{-\alpha\left(\overline{\boldsymbol{\theta}} - \boldsymbol{\theta}_i^*\right)}} \tag{44}$$

The boundary slope parameter α can be chosen to make this soft minimal prior arbitrarily close to being flat and uninformative within the bounds, and arbitrarily close to zero elsewhere. There will nevertheless be a smooth transition across the bounds themselves, ensuring the existence of the gradient at those locations. This generates the following log prior and log prior gradient, which can be used alongside the log-likelihood and its gradient.

$$\begin{cases} \ln p(\boldsymbol{\theta}^*) = -\sum_i \left(\ln\left(1 + e^{-\alpha\left(\boldsymbol{\theta}_i^* - \underline{\boldsymbol{\theta}}\right)}\right) + \ln\left(1 + e^{-\alpha\left(\overline{\boldsymbol{\theta}} - \boldsymbol{\theta}_i^*\right)}\right)\right) \\ \dfrac{\partial \ln p(\boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}_i^*} = \alpha \left(\dfrac{e^{-\alpha\left(\boldsymbol{\theta}_i^* - \underline{\boldsymbol{\theta}}\right)}}{1 + e^{-\alpha\left(\boldsymbol{\theta}_i^* - \underline{\boldsymbol{\theta}}\right)}} + \dfrac{e^{-\alpha\left(\overline{\boldsymbol{\theta}} - \boldsymbol{\theta}_i^*\right)}}{1 + e^{-\alpha\left(\overline{\boldsymbol{\theta}} - \boldsymbol{\theta}_i^*\right)}} \right) \end{cases} \tag{45}$$

The reason why a well-defined gradient on the bounds is a key requirement is because of the potential identification failure mentioned above. Figures 1(c) and (e) illustrate what happens in the well-behaved, identified, case where an empirical observation is consistent with the range of simulated outputs produced in the training data. Even though the resulting surrogate likelihood displays two clear modes corresponding to the most likely parameter values, it takes non-zero values outside of the bounds, due to the GP prior dominating the GP posterior in those locations. Figures 1(d) and (f) illustrate what happens if instead the empirical observation is out of line with the training data. In this case, any con-

tinuous function over $\mathcal{X}$ (as given by the GP prior) is preferable to the GP's surrogate prediction and the mode of the likelihood will lie outside of the bounds $\left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$. In general, therefore, when there are parameters that are not well identified, any gradient-based method will be lead to attempt searches close to or outside of the boundaries of the parameter space. A discontinuous prior at the boundaries would risk stalling these methods, which is not the case of the soft prior (44) which heavily penalises paths outside of the parameter bounds, but does not forbid them. This will result in a very distinctive peak at the parameter boundary itself, visible in figure 1(f), a signal that the surrogate likelihood is pushing the posterior outside of the bounds while the parameter prior is keeping it inside.[19]

The purpose of the uninformative and minimal prior (44) is therefore simply to ensure that Bayesian estimates of $\boldsymbol{\theta}^*$ remain inside the bounds $\left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$, while also ensuring that any identification failures do not stall the BFGS and NUTS algorithms via an undefined gradient at the bounds themselves. It is entirely possible to use a more informative prior, the only requirement, in line with the arguments above, being that it needs to be smooth across the bounds $\left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$ and take arbitrarily low values outside of them. An important warning, however, is that care must be taken when integrating prior information, because the resulting prior can easily overpower the surrogate likelihood. The surrogate will potentially be much flatter than the true, unobserved, likelihood of the model due to the fact that the GP prediction contains prediction error on top of the standard noise term, and in the case where the computational constraint restricts the amount of training data available, this GP prediction error might be sizeable.

4 Applications

Two applications are provided to illustrate the surrogate estimation framework laid out above. The first is a Monte Carlo parameter recovery exercise, demonstrating the consistency of the approach under ideal conditions, while the second is a full estimation exercise on a large simulation model using real data. Both estimations were carried out using the companion BEGRS toolbox developed for the methodology.[20]

4.1 Simulation results on known data generating process

The purpose of this initial exercise is to establish that the proposed BEGRS framework can indeed achieve the theoretical claims outlined above, i.e. that it can produce consistent parameter estimates with a small number of simulation runs relative to the dimensionality of the parameters space. This is done by running a Monte Carlo exercise on the following VAR(1) specification, and attempting to recover a known matrix of parameters B.

$$X_t = BX_{t-1} + \eta_t, \qquad \eta_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \Sigma), \qquad \begin{cases} \Sigma_{i,i} = 1 \\ \Sigma_{i,j} = \rho, \quad i \neq j \end{cases} \tag{46}$$

The number of observable variables is set to $M = 4$, leading to 16 auto-regressive parameters to be estimated in B. It is important to point out that only B is estimated via BEGRS, as the additive noise is captured directly during the training of the GP regression. Instead, the level of correlation between variables ρ forms part of the setting of the Monte Carlo exercise, by verifying that the LCM surrogate can correctly cope with correlated additive noise.

[19] Of course, a likelihood gradient that points outside of the bounds might genuinely be a signal that the bounds were too narrow. However if the problem persists upon re-running the analysis with additional training data on wider bounds, then one can be confident of an identification problem.

[20] The toolbox, which is available from `https://github.com/Sylvain-Barde/begrs`, runs in Python and is based on the GPytorch implementation of Gaussian processes of Gardner et al. (2018). This enables the use of GPU acceleration for the training and likelihood calculation. This is not a requirement, as the GP surrogate can be trained and the mode of the posterior found on a standard desktop PC. However, it is highly recommended for more time-consuming MCMC methods such as NUTS.

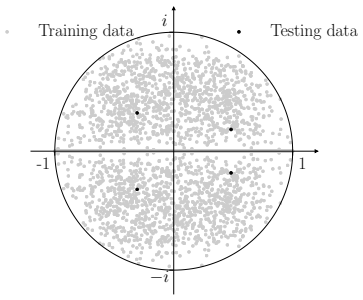


Figure 2: Stability and coverage of VAR parameterisations

Both the training and testing datasets consist of $S = 1000$ series of $T = 200$ observations each. The difference between the two is that the S training series were each generated with a distinct B_s matrix sampled from the 16-dimensional parameter space, while the Monte Carlo replications in the testing data all uses the same B^* matrix. This results in $N = S(T-1) = 199,000$ training observations.[21] All the individual $B_{i,j}$ parameters are drawn from the $[-0.7, 0.7]$ range, allowing the use of a 16-dimensional Sobol sequence to draw training samples for the 1000 training series. Each training sample was checked to ensure that the eigenvalues of the resulting B_s matrices lie within the unit circle.[22] Figure 2 shows a scatter plot of the eigenvalues for all 1000 resulting training samples, showing good space-filling of the unit circle.

For the testing data, an additional stable parameterisation B^* was drawn from further in the Sobol sequence. This ensures that the target parameter vector for the estimation is located in between the training samples and is therefore as distinct as possible from any of them. This is to eliminate the possibility of the methodology appearing to perform well due to luck, should the parameterisation used for testing lie close to one of the samples used in training. Once the testing parameters were picked, 1000 Monte Carlo series of 200 observations each were generated with random draws from η_t. Estimates of B^* were obtained for each series using the mode of the BEGRS posterior $\hat{B}^*_{gp}$ with the minimal prior (44), as well as the standard maximum likelihood VAR estimator $\hat{B}^*_{ml}$, for purposes of comparison.

In the interest of space, the full set of results is provided in appendix C, and only a summary of the key result is shown in table 1, confirming the theoretical discussions from the previous sections. For a large enough number of inducing points, the bias on the BEGRS estimates $\hat{B}_{gp}$ compared to the true value, displayed in the first column, is very small, statistically insignificant and indistinguishable from the bias of the maximum likelihood estimate $\hat{B}_{ml}$. This is in line with the GP regression consistency arguments made above. As expected, performance degrades as the number of inducing points is reduced, following an interesting pattern where consistency fails for entire rows of B^* but is maintained for others, rather than affecting all estimates. This is symptomatic of the GP surrogate being unable to provide good predictions for the corresponding variable, suggesting that the LCM has not converged for that variable. A final comment is that the results are comparable across varying levels of correlations ρ.

Table 1: Summary results for the bias of BEGRS estimates of VAR parameters

Parameter		Number of inducing points		
		250	500	1000
$B^*_{1,1}$	0.154	0.000	−0.022	−0.020
$B^*_{1,2}$	0.113	−0.008	0.015	0.003
$B^*_{1,3}$	0.409	0.001	0.000	−0.015
$B^*_{1,4}$	−0.531	0.020	0.003	0.044
$B^*_{2,1}$	−0.496	0.000	0.039	0.056
$B^*_{2,2}$	−0.196	−0.006	0.039	0.026
$B^*_{2,3}$	0.081	0.022	−0.032	0.014
$B^*_{2,4}$	−0.065	0.007	−0.017	0.010
$B^*_{3,1}$	−0.155	0.058	-0.211^{***}	−0.000
$B^*_{3,2}$	−0.273	0.005	0.012	0.033
$B^*_{3,3}$	−0.068	−0.021	0.452^{***}	0.004
$B^*_{3,4}$	−0.128	−0.032	-0.384^{***}	−0.065
$B^*_{4,1}$	−0.338	0.585^{***}	−0.024	0.036
$B^*_{4,2}$	−0.597	1.037^{***}	0.029	0.045
$B^*_{4,3}$	0.195	-0.425^{***}	0.007	−0.017
$B^*_{4,4}$	0.459	-0.360^{***}	−0.051	−0.044

Note: These results present the bias of the BEGRS estimate $E[\hat{B}^*_{gp}] - B^*$ over 1000 Monte Carlo replications for $\rho = 0.5$. Stars indicate rejection of the null that $\hat{B}^*_{gp}$ follows the sampling distribution of the maximum likelihood estimate $\hat{B}^*_{ml}$, at the 10%, 5% and 1% level respectively.

4.2 Empirical application to a large-scale ABM

The final test of the methodology is a full empirical estimation of the free parameters in the Caiani et al. (2016) model, which offers a good practical illustration of the type of computationally constrained setting to which it can be applied. This model was developed following the financial crisis and great recession of 2008-2009 as a way of improving understanding of the endogenous emergence and contagion of financial shocks in the real economy. In order to achieve this, the model combines a stock-flow consistent approach with fully-fledged commercial and financial networks between households, banks and vertically differentiated firms. This enables both the replication on aggregate of the deep recessions that follow severe financial crises, as well as the analysis at a disaggregated level of the mechanisms that drive them. The framework has been used to analyse policy scenarios such as the effect of fiscal policy design on inequality (Caiani et al., 2019), the effect of fiscal targets in a monetary union (Caiani et al., 2018) and the transmission of monetary policy (Schasfoort et al., 2017).

This model is also useful beyond its applications to policy scenarios, precisely because its computational requirements make it a valuable testbed for validation methodologies aimed at large-scale models, for example the simulation model selection tool of Barde (2020). The model's high computational requirement stems from the stock-flow consistent simulation of credit and deposit networks between banks, consumers and firms, mentioned above. As a result of this, full estimation of the model's free parameters from empirical data has never been carried out. Instead, the existing work using this model and its extensions has typically relied on validation strategies involving replication of stylised facts combined with coarse-grid sensitivity analyses for a small subset of the free parameters. By contrast, we show here that

[21] Note that the order of magnitude for the number of training observations is in line with that of Platt (2021), who use $S = 100$ and $T = 1000$. The flipping of the S and T dimensions compared to their analysis is explained by the aim to maintain a reasonable density of the S parameter space samples in the face of a higher dimensionality (16 vs. 3-7).

[22] 1233 draws were required to obtain 1000 stable parametrisations, meaning that 18.9% of raw samples had to be discarded as unstable.

Table 2: Caiani et al. (2016) free parameter estimates

Parameter		Original	Prior Range	Posterior estimate Mode	Mean
Bank risk aversion (C firms)	ζ_c	3.92245	1 - 10	6.292	6.226
Bank risk aversion (K firms)	ζ_k	21.5133	5 - 40	21.168	21.264
Profit Weight in firm inv.	γ_1	0.01	0.01 - 0.04	0.018	0.018
Capacity Util. Weight in firm inv.	γ_2	0.02	0.01 - 0.04	0.033	0.031
Cons. Firms Precautionary Deposits	σ	1	0.5 - 1.5	0.860	0.863
Intensity of choice - C/K markets	ε^{CK}	0.15	0.05 - 0.3	0.153	0.160
Intensity of choice - credit/deposit	ε^{cd}	0.2	0.05 - 0.3	0.227	0.234
Adaptive expectation parameter	λ	0.25	0.1 - 0.8	0.692	0.666
Labour turnover ratio	ϑ	0.05	0.025 - 0.15	0.033	0.046
Folded normal std. dev.	σ^2_{FN}	0.0094	0.005 - 0.015	0.015	0.015
Haircut param. for defaulted firms	ι	0.5	0.3 - 0.7	0.321	0.357
Unemp. threshold in wage revision	ν	0.08	0.05 - 0.11	0.108	0.105

Table 3: Caiani et al. (2016) MIC goodness-of-fit analysis

	L	r	π	Δy	Δc	Δi	Δw	Aggr.
Original	949.98	2617.81	1903.02	937.05	1027.53	924.44	1554.42	10258.37
Mode	940.50	1313.51	986.31	947.59	907.70	1042.65	1171.27	7862.78
Mean	925.62	1268.94	998.51	952.18	889.40	1049.73	1172.99	7816.96

Note: The aggregate MIC score over the dataset is not the sum of the variable-level scores. This is because there is mutual information between the variables that must be discarded in the aggregate measurement to avoid double-counting. See Barde (2020) for further details.

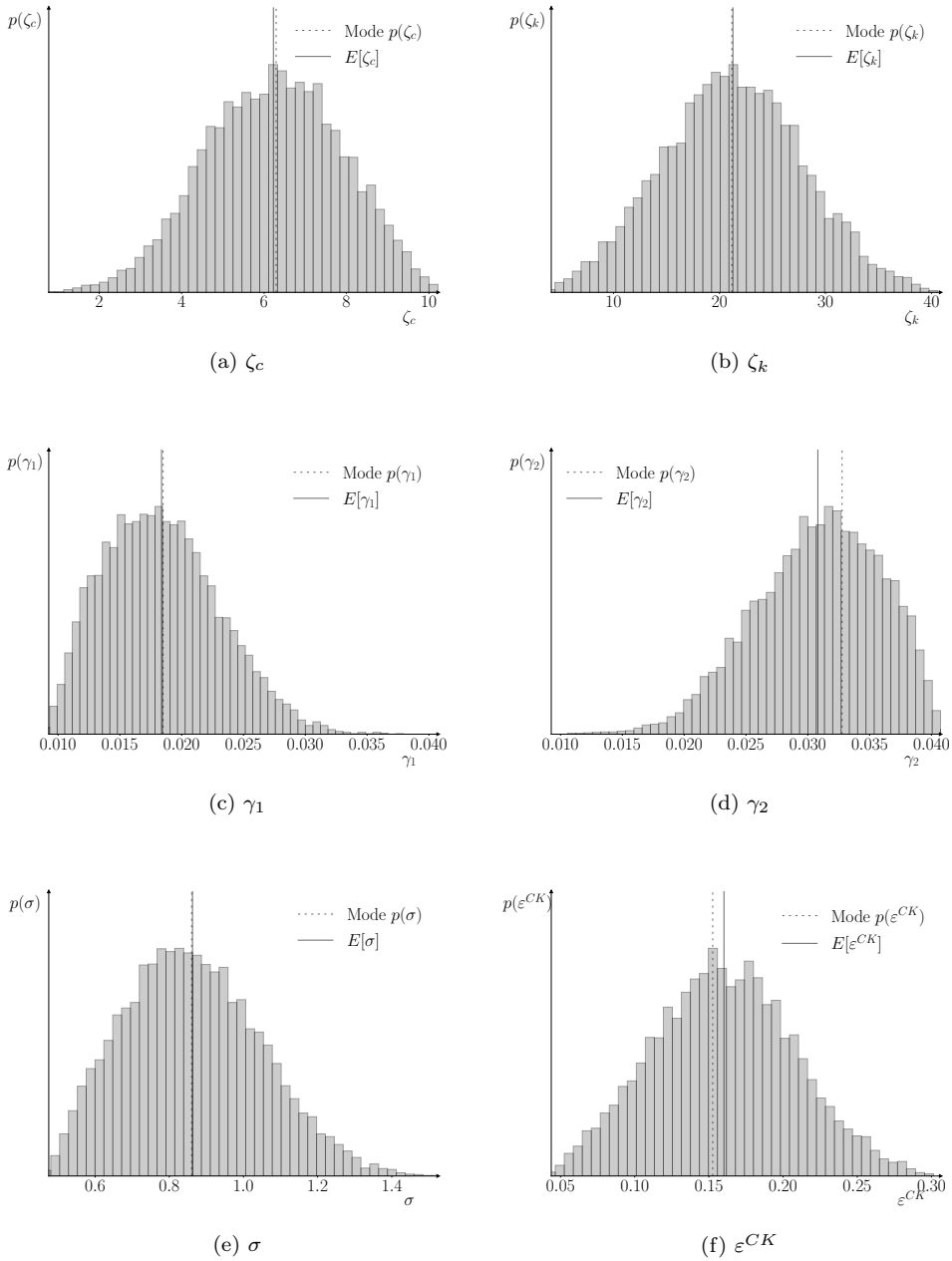
the BEGRS methodology can successfully estimate all the free parameters from standard US macroeconomic time-series data, using only 1000 simulation runs. As was the case in Barde (2020), the analysis uses the Smets and Wouters (2007) data set, with $T = 169$ quarterly observations for the deviation of labour hours from trend L, the real policy rate r, the rate of inflation π, and the real first log difference of output, consumption Δc, investment Δi and wages Δw.

The model has 12 free parameters that are not set from direct observation or inferred from the steady-state constraints. These parameters and their chosen bounds are listed in table 2. The bounds on the first 5 parameters are the ones used in the original Caiani et al. (2016) sensitivity analysis, while the bounds on the remaining 7 parameters are set to allow reasonable variation both above and below the value used in the original paper. In cases where the parameter is naturally bounded in $[0, 1]$ (such as the adaptive expectations parameter λ and the haircut parameter ι), the bounds were set some distance from those natural bounds to avoid pathological behaviour from the simulations. The simulation data used to train the GP surrogate was generated using $S = 1000$ series of $T = 200$ observations, each series using a sample drawn from a 12-dimensional Sobol sequence, similar to the VAR application in the previous section.[23] The prior used in the analysis is again the minimal prior (44) on the parameter bounds, with $\alpha = 20$.

The last two columns of table 2 show two sets of BEGRS estimates: the mode of the posterior, obtained using BFGS, and the mean of the posterior, estimated from 10,000 NUTS iterations. While the estimates for some parameters, such as the second risk aversion, the intensities of choice or labour turnover are close to the values used in by the original authors, others diverge clearly, like the expectation parameter. The marginal frequencies of the NUTS iterations are shown in figures 3 and 4. These show that estimates for several of the parameters lie very close to the boundaries of the parameter space:

[23] A 300 observation burn-in period was included, making each simulation run 500 observations long. The average run time per simulation is 44 minutes, and the use of a 36-worker HPC node reduces the total run time for $S = 1000$ to a more manageable 21 hours.

the standard deviation of the folded normal σ_{FN}^2 which govern the randomness of decision-making, the unemployment threshold in wage revision ν and to a lesser extent the haircut parameter for defaulted firms ι. These three cases, particularly σ_{FN}^2, very probably display the distinctive identification problem discussed in section 3.2.



(a) ζ_c (b) ζ_k

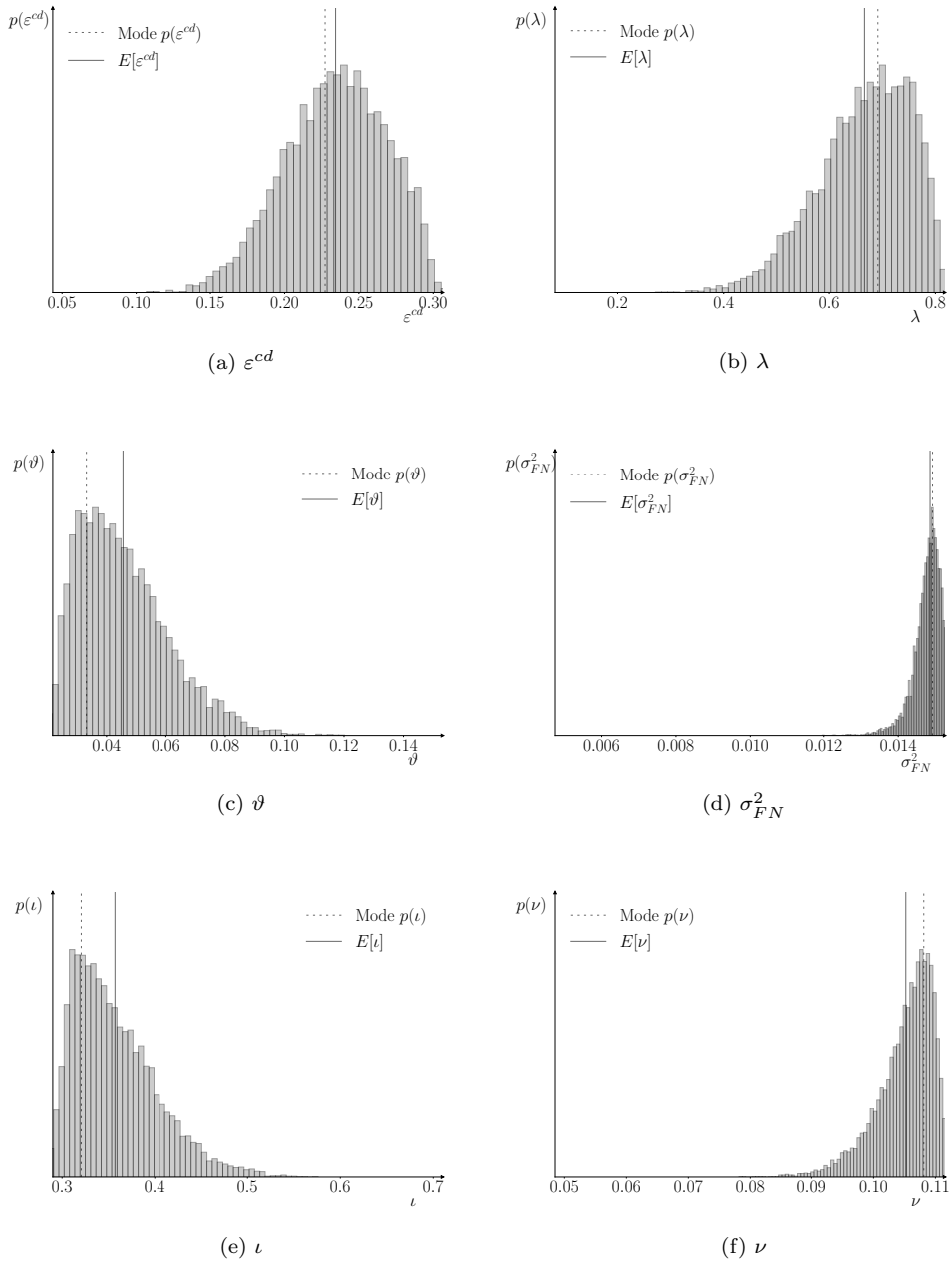
(c) γ_1 (d) γ_2

(e) σ (f) ε^{CK}

Figure 3: NUTS samples for estimated parameters, Part 1

Given these potential identification problems, and also the relatively wide posterior distributions on some of the parameters, one may legitimately question whether the BEGRS estimates improve on the original calibration. Unlike the VAR application, there is no point of reference available to establish, for instance, whether the GP surrogate has converged on the training data. In order to evaluate the improvement in goodness of fit, we replicate the analysis of Barde (2020) and use the multivariate Markov information criterion (MIC) for both the original calibration and the BEGRS estimates.[24] The scores

[24]The MIC is an extension of the Akaike information criterion (AIC), in that it provides an unbiased estimate of the



(a) ε^{cd} (b) λ

(c) ϑ (d) σ^2_{FN}

(e) ι (f) ν

Figure 4: NUTS samples for estimated parameters, Part 2

obtained for each calibration, both at the variable and aggregate levels, are provided in table 3. These confirm that both sets of estimates bring a large and significant improvement to the goodness-of-fit of the model on the Smets and Wouters (2007) dataset compared to the original calibration. Barde (2020) showed that the latter was characterised by reasonable performance on the aggregate quantities (L, Δy, Δc and Δi) but a very poor fit on the aggregate prices (r, π and Δw). Both sets of BEGRS estimates improve the poor fit of the three price variables, at the cost of a slightly worse performance on Δi. This strongly suggests that the training step of the GP regression has run successfully.

cross-entropy between a model and some empirical data. Its main feature is that it can be calculated directly from the simulated data produced by a Markov process. See Barde (2020) for further details.

5 Conclusion

This paper outlines and tests a Bayesian estimation framework specifically aimed at computationally demanding simulating models. This contribution is not the only estimation framework available for simulation models, not is it claimed that is the best in all cases. However, the framework as a whole is designed with computational constraints in mind, in order to make the best use of a potentially limited amount of training data. Core to this are the use of a one-step ahead predictor, which leverages the time-dimension of the simulated data to increase the number of training observations, and the use of GP regression as the surrogate model. This choice is driven by the desirable theoretical properties of the GP surrogate, notably the universal approximation property, proven converge with minimal assumptions on the underlying model, and the self-regularisation property of the GP estimation, which limits overfitting on potentially limited training data. The paper extends a pre-existing result on GP regression to show that the method is consistent, and demonstrates the functionality of the approach by providing the first existing empirical estimation of all the free parameters of a large-scale, computationally intensive ABM.

Several caveats of the proposed methodology do merit discussion. First, the estimation framework detailed here does not offer a panacea: a sizeable amount of simulation data is still required, increasingly so as the dimensionality of the parameter space d_Θ grows beyond the baseline of 10-20 parameters used here. The use of a one-step ahead GP surrogate helps to maximise the amount of training data available from simulation runs, and the two-step nature of the approach ensures that the surrogate can be re-used for different empirical datasets, however it does not eliminate the need for time-consuming simulation in the first place. An important item for future research will be the exploration of the trade-off between the parameter dimensionality of the model and the number of simulation runs required for good performance.

Second, the BEGRS framework outlined above nests two distinct Bayesian estimations, the first estimating the GP parameters from the training data, the second using the GP surrogate to estimate the model parameters from the empirical data. This means that in addition to the standard concerns relating to identification of the simulation model parameters, one additionally needs to verify that the surrogate likelihood generated by the GP is itself valid, i.e. that the first-stage estimation of the GP has converged. The learning curve obtained by the ELBO can assist in evaluation whether this is the case, however given the computational constraint, there may be little that can be done to improve the situation if the cause of any detected issue is a relative lack of training data. A key recommendation, therefore, is to always run a post-estimation goodness-of-fit test, as was done for the empirical application, to assess whether the BEGRS estimates genuinely bring an improvement to the model fit.

Finally, a related concern is that the performance of the methodology does rest on the model satisfying a key assumption, namely that the mean one-step-ahead prediction is a continuous function in the input space $\mathcal{X}$. Discontinuities, or even very rapid changes in the mean function will be captured in the GP surrogate via the use of a kernel with a very short length scale ℓ_v. Because the convergence of the GP surrogate varies inversely with the shortest length scale used, the presence of such discontinuities will degrade the performance of the surrogate for any fixed amount of training data, or equivalently impose a higher training data requirement in order to achieve a given performance. This reinforces the point made above that domain knowledge of the simulation model is an important input in the process.

References

Alvarez, Mauricio A, Lorenzo Rosasco, Neil D Lawrence et al. (2012) "Kernels for vector-valued functions: A review," *Foundations and Trends in Machine Learning*, Vol. 4, pp. 195–266.

Barde, Sylvain (2020) "Macroeconomic simulation comparison with a multivariate extension of the Markov information criterion," *Journal of Economic Dynamics and Control*, Vol. 111.

Bargigli, Leonardo, Luca Riccetti, Alberto Russo, and Mauro Gallegati (2020) "Network calibration and metamodeling of a financial accelerator agent based model," *Journal of Economic Interaction and Coordination*, Vol. 15, pp. 413–440.

Bishop, Christopher M (2006) *Pattern recognition and machine learning*: Springer.

Burt, David R, Carl Edward Rasmussen, and Mark van der Wilk (2020) "Convergence of Sparse Variational Inference in Gaussian Processes Regression," *Journal of Machine Learning Research*, Vol. 21, pp. 1–63.

Burt, David, Carl Edward Rasmussen, and Mark Van Der Wilk (2019) "Rates of convergence for sparse variational Gaussian process regression," in *International Conference on Machine Learning*, pp. 862–871, PMLR.

Caiani, Alessandro, Ermanno Catullo, and Mauro Gallegati (2018) "The effects of fiscal targets in a monetary union: a multi-country agent-based stock flow consistent model," *Industrial and Corporate Change*, Vol. 27, pp. 1123–1154.

Caiani, Alessandro, Antoine Godin, Eugenio Caverzasi, Mauro Gallegati, Stephen Kinsella, and Joseph E Stiglitz (2016) "Agent based-stock flow consistent macroeconomics: Towards a benchmark model," *Journal of Economic Dynamics and Control*, Vol. 69, pp. 375–408.

Caiani, Alessandro, Alberto Russo, and Mauro Gallegati (2019) "Does inequality hamper innovation and growth? An AB-SFC analysis," *Journal of Evolutionary Economics*, Vol. 29, pp. 177–228.

Caponnetto, Andrea, Charles A Micchelli, Massimiliano Pontil, and Yiming Ying (2008) "Universal multi-task kernels," *The Journal of Machine Learning Research*, Vol. 9, pp. 1615–1646.

Chen, Siyan and Saul Desiderio (2022) "A regression-based calibration method for agent-based models," *Computational Economics*, Vol. 59, pp. 687–700.

Choi, Taeryon and Mark J Schervish (2007) "On posterior consistency in nonparametric regression problems," *Journal of Multivariate Analysis*, Vol. 98, pp. 1969–1987.

Cioppa, Thomas M and Thomas W Lucas (2007) "Efficient nearly orthogonal and space-filling Latin hypercubes," *Technometrics*, Vol. 49, pp. 45–55.

Damianou, Andreas and Neil D Lawrence (2013) "Deep gaussian processes," in *Artificial intelligence and statistics*, pp. 207–215, PMLR.

Delli Gatti, Domenico and Jakob Grazzini (2020) "Rising to the challenge: Bayesian estimation and forecasting techniques for macroeconomic Agent Based Models," *Journal of Economic Behavior & Organization*, Vol. 178, pp. 875–902.

Fagiolo, Giorgio, Mattia Guerini, Francesco Lamperti, Alessio Moneta, and Andrea Roventini (2019) "Validation of agent-based models in economics and finance," in *Computer simulation validation*: Springer, pp. 763–787.

Fagiolo, Giorgio, Alessio Moneta, and Paul Windrum (2007) "A critical guide to empirical validation of agent-based models in economics: methodologies, procedures, and open problems," *Computational Economics*, Vol. 30, pp. 195–226.

Gardner, Jacob, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson (2018) "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration," *Advances in neural information processing systems*, Vol. 31.

Gelman, Andrew, Walter R Gilks, and Gareth O Roberts (1997) "Weak convergence and optimal scaling of random walk Metropolis algorithms," *The annals of applied probability*, Vol. 7, pp. 110–120.

Gelman, Andrew, Gareth O Roberts, and Walter R Gilks (1996) "Efficient Metropolis jumping rules," *Bayesian statistics*, Vol. 5, p. 42.

Gilbert, Nigel and Klaus Troitzsch (2005) *Simulation for the social scientist*: McGraw-Hill Education (UK), 2nd edition.

Gilli, Manfred and Peter Winker (2003) "A global optimization heuristic for estimating agent based models," *Computational Statistics & Data Analysis*, Vol. 42, pp. 299–312.

Gouriéroux, Christian and Alain Monfort (1993) "Simulation based inference : a survey with special reference to panel data models," *Journal of Econometrics*, Vol. 59, pp. 5–33.

——— (1996) *Simulation-based Econometric Methods*: Oxford University Press.

Gouriéroux, Christian, Alain Monfort, and Eric Renault (1993) "Indirect Inference," *Journal of Applied Econometrics*, Vol. 8, pp. S85–S118.

Grazzini, Jakob and Matteo Richiardi (2015) "Estimation of ergodic agent-based models by simulated minimum distance," *Journal of Economic Dynamics and Control*, Vol. 51, pp. 148–165.

Grazzini, Jakob, Matteo G Richiardi, and Mike Tsionas (2017) "Bayesian estimation of agent-based models," *Journal of Economic Dynamics and Control*, Vol. 77, pp. 26–47.

Hensman, James, Alexander Matthews, and Zoubin Ghahramani (2015) "Scalable variational Gaussian process classification," in *Artificial Intelligence and Statistics*, pp. 351–360, PMLR.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989) "Multilayer feedforward networks are universal approximators," *Neural networks*, Vol. 2, pp. 359–366.

Kanagawa, Motonobu, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur (2018) "Gaussian processes and kernel methods: A review on connections and equivalences," *arXiv preprint arXiv:1807.02582*.

Koepernik, Peter and Florian Pfaff (2021) "Consistency of Gaussian Process Regression in Metric Spaces," *Journal of Machine Learning Research*, Vol. 22, pp. 1–27.

Kukacka, Jiri and Jozef Barunik (2017) "Estimation of financial agent-based models with simulated maximum likelihood," *Journal of Economic Dynamics and Control*, Vol. 85, pp. 21–45.

Lamperti, Francesco, Andrea Roventini, and Amir Sani (2018) "Agent-based model calibration using machine learning surrogates," *Journal of Economic Dynamics and Control*, Vol. 90, pp. 366–389.

Le Gratiet, Loic and Josselin Garnier (2015) "Asymptotic analysis of the learning curve for Gaussian process regression," *Machine learning*, Vol. 98, pp. 407–433.

Liefvendahl, Mattias and Rafał Stocki (2006) "A study on algorithms for optimization of Latin hypercubes," *Journal of statistical planning and inference*, Vol. 136, pp. 3231–3247.

Micchelli, Charles A, Yuesheng Xu, and Haizhang Zhang (2006) "Universal Kernels.," *Journal of Machine Learning Research*, Vol. 7.

Neal, Radford M (1996) *Bayesian learning for neural networks*, Vol. 118 of Lecture Notes in Statistics: Springer Science & Business Media.

Platt, Donovan (2021) "Bayesian estimation of economic simulation models using neural networks," *Computational Economics*, pp. 1–52.

Rasmussen, Carl Edward and Christopher K. Williams (2006) *Gaussian processes for machine learning*: MIT press Cambridge, MA.

Salle, Isabelle and Murat Yıldızoğlu (2014) "Efficient sampling and meta-modeling for computational economic models," *Computational Economics*, Vol. 44, pp. 507–536.

Santner, Thomas J, Brian J Williams, William I Notz, and Brain J Williams (2018) *The design and analysis of computer experiments*: Springer, 2nd edition.

Schasfoort, Joeri, Antoine Godin, Dirk Bezemer, Alessandro Caiani, and Stephen Kinsella (2017) "Monetary policy transmission in a macroeconomic agent-based model," *Advances in Complex Systems*, Vol. 20.

Seeger, Matthias W, Sham M Kakade, and Dean P Foster (2008) "Information consistency of nonparametric Gaussian process methods," *IEEE Transactions on Information Theory*, Vol. 54, pp. 2376–2382.

Shi, Jian Qing and Taeryon Choi (2011) *Gaussian process regression analysis for functional data*: CRC Press.

Smets, Frank and Rafael Wouters (2007) "Shocks and frictions in US business cycles: A Bayesian DSGE approach," *American Economic Review*, Vol. 97, pp. 586–606.

Smith, Anthony A. (1993) "Estimating Nonlinear Time-Series Models Using Simulated Vector Autoregressions," *Journal of Applied Econometrics*, Vol. 8, pp. S63–S84.

——— (2016) *Indirect Inference*: Palgrave Macmillan UK.

Titsias, Michalis (2009) "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial intelligence and statistics*, pp. 567–574, PMLR.

Van Der Vaart, Aad and Harry Van Zanten (2011) "Information Rates of Nonparametric Gaussian Process Methods.," *Journal of Machine Learning Research*, Vol. 12.

Wynne, George, François-Xavier Briol, and Mark Girolami (2021) "Convergence guarantees for Gaussian process means with misspecified likelihoods and smoothness," *Journal of Machine Learning Research*, Vol. 22.

A Proof for the bound on LCM regret

The main lemma used in Seeger et al. (2008) to derive the bound on the regret term requires that the kernel possess an eigenexpansion (18) and the eigenvalues have a finite Hilbert-Schmidt norm, $\sum_h \lambda_h < \infty$.

Lemma 1 (from Seeger et al., 2008): *Let K be a kernel possessing an eigenexpansion (18) and $\mathbf{K}^{\mathbf{x},\mathbf{x}}$ be the covariance matrix resulting from applying the kernel to a dataset $\mathbf{X}$ containing N observations drawn from density function $\nu(x)$. Then, given a constant $c > 0$, we have:*

$$E_{\nu(x)}\left[R\right] \quad = \quad E_{\nu(x)}\Big[\ln\left|\mathbf{I}_N + c\mathbf{K}^{\mathbf{x},\mathbf{x}}\right|\Big] \quad \leq \quad \sum_{h=0}^{\infty} \ln\left(1 + c\lambda_h N\right)$$

Proof: The expected regret term $E_{\nu(x)}\Big[\ln\left|\mathbf{I}_N + c\mathbf{K}^{\mathbf{x},\mathbf{x}}\right|\Big]$ can be written in terms of the eigenexpansion (18) and rearranged using Sylvester's determinant identity:

$$\begin{aligned} E_{\nu(x)}\Big[\ln\left|\mathbf{I}_N + c\mathbf{K}^{\mathbf{x},\mathbf{x}}\right|\Big] &= \lim_{H\to\infty} E_{\nu(x)}\Big[\ln\left|\mathbf{I}_N + c\mathbf{\Psi}_H^{\mathbf{x}}\mathbf{\Lambda}_H(\mathbf{\Psi}_H^{\mathbf{x}})^T\right|\Big] \\ &= \lim_{H\to\infty} E_{\nu(x)}\Big[\ln\left|\mathbf{I}_H + c\mathbf{\Lambda}_H^{1/2}(\mathbf{\Psi}_H^{\mathbf{x}})^T\mathbf{\Psi}_H^{\mathbf{x}}\mathbf{\Lambda}_H^{1/2}\right|\Big] \end{aligned}$$

The expected regret can now be bounded using Jensen's inequality and the concavity of the logarithm:

$$\begin{aligned} E_{\nu(x)}\Big[\ln\left|\mathbf{I}_N + c\mathbf{K}^{\mathbf{x},\mathbf{x}}\right|\Big] &\leq \lim_{H\to\infty} \ln\Big|\mathbf{I}_H + cE_{\nu(x)}\Big[\mathbf{\Lambda}_H^{1/2}(\mathbf{\Psi}_H^{\mathbf{x}})^T\mathbf{\Psi}_H^{\mathbf{x}}\mathbf{\Lambda}_H^{1/2}\Big]\Big| \\ &= \lim_{H\to\infty} \ln\Big|\mathbf{I}_H + c\mathbf{\Lambda}_H N E_{\nu(x)}\Big[N^{-1}(\mathbf{\Psi}_H^{\mathbf{x}})^T\mathbf{\Psi}_H^{\mathbf{x}}\Big]\Big| \\ &= \sum_{h=0}^{\infty} \ln\left(1 + c\lambda_h N\right) \end{aligned}$$

The final expression for the bound relies on the fact that the orthogonality of the kernel eigenfunctions ensures that $E\left[N^{-1}(\mathbf{\Psi}_H^{\mathbf{x}})^T\mathbf{\Psi}_H^{\mathbf{x}}\right] = \mathbf{I}_H$. The resulting bound is thus the log determinant of a diagonal matrix, which is just the sum of the logarithm of the diagonal entries. ■

This lemma directly provides the proof for the regret bound in Seeger et al. (2008) by making use of Sylvester's determinant identity. By changing the order of summation in the determinant, one can exploit the eigenexpansion of the kernel and obtain a bound that depends on an infinite sum over the eigenvalue spectrum of the kernel. This strategy does not work directly for the LCM kernel (13), which is a linear combination of V different kernels. While linear combinations of kernels are themselves valid kernels and will possess an eigenexpansion (18), their eigenvalue spectrum cannot be determined from the spectrum of the individual kernels in the sum, which complicates the analysis of the bound. This is because the eigenfunctions of the individual kernels depend on each kernel's length scale parameter ℓ_v. The resulting kernel matrices $\mathbf{K}_v^{\mathbf{x},\mathbf{x}}$ are therefore not simultaneously diagonalisable. The strategy used here to obtain a tractable bound for the LCM is instead to show that the regret of the LCM kernel is bounded below a fixed multiple of the regret of worst-performing kernel in the linear combination.

Lemma 2: *Let $\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}} = \sum_{v\in\mathbf{v}} \alpha_v \mathbf{K}_v^{\mathbf{x},\mathbf{x}}$ be a linear combination of V distinct $N \times N$ kernel matrices $\mathbf{K}_v^{\mathbf{x},\mathbf{x}}$ with weights $\alpha_v > 0$, where $\mathbf{v} = \{1, 2, ..., V\}$ is the set of indices identifying each kernel matrix and weight. Given an arbitrary constant $c > 0$, the following bound exists:*

$$\left|\mathbf{I}_N + c\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}\right| \quad < \quad \left|\mathbf{I}_N + Vc\alpha_{v^*}\mathbf{K}_{v^*}^{\mathbf{x},\mathbf{x}}\right|^V$$

Where:

$$v^* = \arg\max_{v\in\mathbf{v}} \left|\mathbf{I}_N + Vc\alpha_v\mathbf{K}_v^{\mathbf{x},\mathbf{x}}\right|$$

Proof: First, use the fact that $\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}$ is a linear combination of V kernels to express the determinant as the following sum:

$$\left|\mathbf{I}_N + c\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}\right| = \left|\sum\nolimits_{v\in\mathbf{v}} \mathbf{M}_v\right|, \qquad \mathbf{M}_v = \frac{1}{V}\mathbf{I}_N + c\alpha_v \mathbf{K}_v^{\mathbf{x},\mathbf{x}}$$

By factorising out the determinants of the individual elements $|\mathbf{M}_v|$, this can be rewritten as the product of the determinants of $\mathbf{M}_v$ and a determinant term containing the inverses of $\mathbf{M}_v$:

$$\begin{aligned}
\left|\mathbf{I}_N + c\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}\right| &= \left(\prod\nolimits_{v\in\mathbf{v}} |\mathbf{M}_v|\right) \left|\sum\nolimits_{v\in\mathbf{v}} \prod\nolimits_{p\in\mathbf{v}\setminus v} (\mathbf{M}_p)^{-1}\right| \\
&< \left(\prod\nolimits_{v\in\mathbf{v}} |\mathbf{M}_v|\right) \left|\sum\nolimits_{v\in\mathbf{v}} \prod\nolimits_{p\in\mathbf{v}\setminus v} \frac{1}{\lambda_p^{min}}\mathbf{I}_N\right| \\
&\leq \left(\prod\nolimits_{v\in\mathbf{v}} |\mathbf{M}_v|\right) |V\mathbf{I}_N|^V \quad = \quad \prod\nolimits_{v\in\mathbf{v}} |V\mathbf{M}_v| \\
&= \prod\nolimits_{v\in\mathbf{v}} \left|\mathbf{I}_N + Vc\alpha_v \mathbf{K}_v^{\mathbf{x},\mathbf{x}}\right|
\end{aligned}$$

The first upper bound in this expression can be obtained by replacing $(\mathbf{M}_v)^{-1}$ by $\mathbf{I}_N/\lambda_v^{min}$, where λ_v^{min} is the smallest eigenvalue of $\mathbf{M}_v$. The second bound comes from the fact that because $\mathbf{K}_v^{\mathbf{x},\mathbf{x}}$ is positive semi-definite and commutes with $\mathbf{I}_N$, the eigenvalues of $\mathbf{M}_v$ are real-valued and $\lambda_v^{min} \geq 1/V$. Finally, if v^* is the index for the kernel $\mathbf{K}_{v^*}^{\mathbf{x},\mathbf{x}}$ such that $|\mathbf{M}_{v^*}| = \max_{v\in\mathbf{v}} |\mathbf{M}_v|$, then we have:

$$\left|\mathbf{I}_N + c\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}\right| \quad < \quad \prod\nolimits_{v\in\mathbf{v}} \left|\mathbf{I}_N + Vc\alpha_v \mathbf{K}_v^{\mathbf{x},\mathbf{x}}\right| \quad < \quad \left|\mathbf{I}_N + Vc\alpha_{v^*} \mathbf{K}_{v^*}^{\mathbf{x},\mathbf{x}}\right|^V$$

■

In our case $\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}$ results from a linear combination of RBF kernels (12), and two factors affect which kernel v^* ends up being performing the worst. The first is the length scale of the kernel ℓ_v, where the lower the length scale, the higher the regret due to the lower smoothness of the resulting prediction. The second factor is the weight, where the higher α_v the higher the regret. However, this bound applies regardless of the type of kernel, which means that in principle it applies to any arbitrary linear combination of different kernel types. This carries across to the proposition below, which combines lemmas 1 and 2 to provide a bound for the regret of the LCM (23).

Proposition (from section 2.2): *The expected regret of the LCM (23) has the following upper bound, where v^* indicates the latent GP variable possessing the largest regret:*

$$E_{\nu(x)}\left[\ln\left|\mathbf{I}_{MN} + \boldsymbol{\Sigma}^{-2}\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}\right|\right] \quad < \quad MV\sum_{h=0}^{\infty} \ln\left(1 + Vb_{v^*,m^*}^2 \sigma_{m^*}^{-2} \lambda_{v^*,h} N\right)$$

Proof: First of all, we rearrange the regret to explicitly reveal the block diagonal structure of the LCM kernel matrix $\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}$, induced by the use of the Kronecker product on the latent kernel matrices $\mathbf{K}_v^{\mathbf{x},\mathbf{x}}$:

$$\begin{aligned}
R &= \ln\left|\mathbf{I}_{MN} + \boldsymbol{\Sigma}^{-2}\tilde{\mathbf{K}}^{\mathbf{x},\mathbf{x}}\right| \\
&= \ln\left|\mathbf{I}_{MN} + \boldsymbol{\Sigma}^{-2}\tilde{\mathbf{B}}\mathbf{K}^{\mathbf{x},\mathbf{x}}\tilde{\mathbf{B}}^T\right| \\
&= \ln\left|\mathbf{I}_{MN} + \boldsymbol{\Sigma}^{-2}\left(\sum\nolimits_v \mathbf{B}_v\mathbf{B}_v^T \otimes \mathbf{K}_v^{\mathbf{x},\mathbf{x}}\right)\right|
\end{aligned}$$

From this, we can determine the sequence of bounds provided below. First, the block diagonal structure of the regret means that Fischer's determinant inequality can be used to bound it below the sum of the log determinants of the main diagonal blocks. This first bound is the sum of M regret terms, each of which is the regret of a linear combination of V kernel matrices. Using lemma 2, each term in this

sum is itself bounded above by a multiple of the regret of the worst-performing kernels for that variable m. Note that in the second inequality, the index v^* identifying the largest regret term might be different across the M variables, due to the variable and kernel-specific weights $b_{v,m}$ and noises σ_m. This can be addressed by identifying the variable m^* with the largest regret term, leading to the third inequality.

$$\begin{aligned} R &\leq \sum_m \ln \left| \mathbf{I}_N + \sigma_m^{-2} \sum\nolimits_v b_{v,m}^2 \mathbf{K}_v^{\mathbf{x},\mathbf{x}} \right| \\ &< \sum_m V \ln \left| \mathbf{I}_N + V \sigma_m^{-2} b_{v^*,m}^2 \mathbf{K}_{v^*}^{\mathbf{x},\mathbf{x}} \right| \\ &< MV \ln \left| \mathbf{I}_N + V \sigma_{m*}^{-2} b_{v^*,m^*}^2 \mathbf{K}_{v^*}^{\mathbf{x},\mathbf{x}} \right| \end{aligned}$$

At this point, the LCM regret term is bounded below an expression containing the regret of a single kernel matrix $\mathbf{K}_{v^*}^{\mathbf{x},\mathbf{x}}$. Using lemma 1, we obtain the following bound for the expectation of that regret:

$$MV \times E_{\nu(x)} \left[\ln \left| \mathbf{I}_N + V \sigma_{m*}^{-2} b_{v^*,m^*}^2 \mathbf{K}_{v^*}^{\mathbf{x},\mathbf{x}} \right| \right] \quad < \quad MV \sum_{h=0}^{\infty} \ln \left(1 + V b_{v^*,m^*}^2 \sigma_{m*}^{-2} \lambda_{v^*,h} N \right)$$

■

B Derivations for the variational approximation of the LCM

B.1 Derivation of the LCM ELBO

Replacing the definition of the joint (29) and variational (30) distributions:

$$\mathcal{L}(\phi) \;=\; \int_{\mathbf{u}} \int_{\mathbf{f}} \ln \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}) p(\mathbf{u})}{p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u})} q(\mathbf{f}, \mathbf{u}) \, d\mathbf{f} \, d\mathbf{u}$$

$$\mathcal{L}(\phi) \;=\; \int_{\mathbf{u}} \int_{\mathbf{f}} \ln p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u}) \, d\mathbf{f} \, d\mathbf{u} + \int_{\mathbf{u}} \int_{\mathbf{f}} \ln \frac{p(\mathbf{u})}{q(\mathbf{u})} q(\mathbf{f}, \mathbf{u}) \, d\mathbf{f} \, d\mathbf{u}$$

$$\mathcal{L}(\phi) \;=\; \int_{\mathbf{u}} \left(\int_{\mathbf{f}} \ln p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}) \, d\mathbf{f} \right) q(\mathbf{u}) \, d\mathbf{u} + \int_{\mathbf{u}} \ln \frac{p(\mathbf{u})}{q(\mathbf{u})} \left(\int_{\mathbf{f}} q(\mathbf{f}, \mathbf{u}) \, d\mathbf{f} \right) d\mathbf{u}$$

$$\mathcal{L}(\phi) \;=\; \int_{\mathbf{u}} \left(\int_{\mathbf{f}} \ln p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}) \, d\mathbf{f} \right) q(\mathbf{u}) \, d\mathbf{u} + \int_{\mathbf{u}} \ln \frac{p(\mathbf{u})}{q(\mathbf{u})} q(\mathbf{u}) \, d\mathbf{u}$$

A first simplification is that the last term of this expression is simply the Kullback-Leibler divergence from $q(\mathbf{u})$ to $p(\mathbf{u})$:

$$\mathcal{L}(\phi) \;=\; \int_{\mathbf{u}} \left(\int_{\mathbf{f}} \ln p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{u}) \, d\mathbf{f} \right) q(\mathbf{u}) \, d\mathbf{u} - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big]$$

Where:

$$D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big] = - \int_{\mathbf{u}} \ln \frac{p(\mathbf{u})}{q(\mathbf{u})} q(\mathbf{u}) \, d\mathbf{u}$$

Given the distributions (27) and (30) for $p(\mathbf{u})$ and $q(\mathbf{u})$, this term can be calculated using the standard formula for the Kullback-Leibler divergence between two multivariate normal distributions. Given the multivariate normal densities (10) and (25) for $p(\mathbf{y} \mid \mathbf{f})$ and $p(\mathbf{f} \mid \mathbf{u})$, one can evaluate the remaining integral:

$$\begin{aligned}\mathcal{L}(\phi) &= \int_{\mathbf{u}} \left(\int_{\mathbf{f}} \left[-\frac{1}{2} \ln \left| 2\pi \boldsymbol{\Sigma}^2 \right| - \frac{1}{2} (\mathbf{y} - \tilde{\boldsymbol{\mu}})^T \boldsymbol{\Sigma}^{-2} (\mathbf{y} - \tilde{\boldsymbol{\mu}}) \right] p(\mathbf{f} \mid \mathbf{u}) \, d\mathbf{f} \right) q(\mathbf{u}) \, d\mathbf{u} \\ &\quad - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big] \\ \mathcal{L}(\phi) &= \frac{1}{2} \ln \left| 2\pi \boldsymbol{\Sigma}^2 \right| - \frac{1}{2} \int_{\mathbf{u}} \left(\int_{\mathbf{f}} \left(\mathbf{y}^T \boldsymbol{\Sigma}^{-2} \mathbf{y} - 2 \tilde{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-2} \mathbf{y} + \tilde{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}^{-2} \tilde{\boldsymbol{\mu}} \right) p(\mathbf{f} \mid \mathbf{u}) \, d\mathbf{f} \right) q(\mathbf{u}) \, d\mathbf{u} \\ &\quad - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big]\end{aligned}$$

Taking advantage of the fact that for a vector $\mathbf{a}$ we have $\mathbf{a}^T \mathbf{a} = \mathrm{tr}(\mathbf{a}\mathbf{a}^T)$, we can write:

$$\begin{aligned}\mathcal{L}(\phi) &= -\frac{1}{2} \ln \left| 2\pi \boldsymbol{\Sigma}^2 \right| - \frac{1}{2} \int_{\mathbf{u}} \left(\int_{\mathbf{f}} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1} \left(\mathbf{y}\mathbf{y}^T - 2\mathbf{y}\tilde{\boldsymbol{\mu}}^T + \tilde{\boldsymbol{\mu}}\tilde{\boldsymbol{\mu}}^T \right) \boldsymbol{\Sigma}^{-1} \right) p(\mathbf{f} \mid \mathbf{u}) \, d\mathbf{f} \right) q(\mathbf{u}) \, d\mathbf{u} \\ &\quad - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big]\end{aligned}$$

Given the LCM prediction $\tilde{\boldsymbol{f}} = \tilde{\mathbf{B}}\mathbf{f}$, one can take the expectation of the trace term with respect to the conditional distribution of the latent variables $p(\mathbf{f} \mid \mathbf{u}) = \mathcal{N}(\mathbf{f} \mid \mathbf{A}\mathbf{u}, \, \mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}})$:

$$\begin{aligned}\mathcal{L}(\phi) &= -\frac{1}{2} \ln \left| 2\pi \boldsymbol{\Sigma}^2 \right| - \frac{1}{2} \int_{\mathbf{u}} \Big(\mathrm{tr} \Big(\boldsymbol{\Sigma}^{-1} \Big(\mathbf{y}\mathbf{y}^T - 2\mathbf{y}\mathbf{u}^T \mathbf{A}^T \tilde{\mathbf{B}}^T + \tilde{\mathbf{B}}\mathbf{A}\mathbf{u}\mathbf{u}^T \mathbf{A}^T \tilde{\mathbf{B}}^T \\ &\quad + \tilde{\mathbf{B}}\mathbf{K}^{\mathbf{x},\mathbf{x}} \tilde{\mathbf{B}}^T - \tilde{\mathbf{B}}\mathbf{Q}^{\mathbf{x},\mathbf{x}} \tilde{\mathbf{B}}^T \Big) \boldsymbol{\Sigma}^{-1} \Big) \Big) q(\mathbf{u}) \, d\mathbf{u} - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big]\end{aligned}$$

$$\begin{aligned}\mathcal{L}(\phi) &= -\frac{1}{2} \ln \left| 2\pi \boldsymbol{\Sigma}^2 \right| - \frac{1}{2} \int_{\mathbf{u}} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1} \left(\mathbf{y}\mathbf{y}^T - 2\mathbf{y}\mathbf{u}^T \mathbf{A}^T \tilde{\mathbf{B}}^T + \tilde{\mathbf{B}}\mathbf{A}\mathbf{u}\mathbf{u}^T \mathbf{A}^T \tilde{\mathbf{B}}^T \right) \boldsymbol{\Sigma}^{-1} \right) q(\mathbf{u}) \, d\mathbf{u} \\ &\quad - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1} \tilde{\mathbf{B}} \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}} \right) \tilde{\mathbf{B}}^T \boldsymbol{\Sigma}^{-1} \right) - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big]\end{aligned}$$

Taking expectation of the trace term with respect to $q(\mathbf{u}) = \mathcal{N}(\mathbf{f} \mid \mathbf{m}, \, \mathbf{S})$:

$$\begin{aligned}\mathcal{L}(\phi) &= -\frac{1}{2} \ln \left| 2\pi \boldsymbol{\Sigma}^2 \right| - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1} \left(\mathbf{y}\mathbf{y}^T - 2\mathbf{y}\mathbf{m}^T \mathbf{A}^T \tilde{\mathbf{B}}^T + \tilde{\mathbf{B}}\mathbf{A}\mathbf{m}\mathbf{m}^T \mathbf{A}^T \tilde{\mathbf{B}}^T + \tilde{\mathbf{B}}\mathbf{A}\mathbf{S}\mathbf{A}^T \tilde{\mathbf{B}}^T \right) \boldsymbol{\Sigma}^{-1} \right) \\ &\quad - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1} \tilde{\mathbf{B}} \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}} \right) \tilde{\mathbf{B}}^T \boldsymbol{\Sigma}^{-1} \right) - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big]\end{aligned}$$

$$\begin{aligned}\mathcal{L}(\phi) &= -\frac{1}{2} \ln \left| 2\pi \boldsymbol{\Sigma}^2 \right| - \frac{1}{2} \left(\mathbf{y} - \tilde{\mathbf{B}}\mathbf{A}\mathbf{m} \right)^T \boldsymbol{\Sigma}^{-2} \left(\mathbf{y} - \tilde{\mathbf{B}}\mathbf{A}\mathbf{m} \right) - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1} \tilde{\mathbf{B}}\mathbf{A}\mathbf{S}\mathbf{A}^T \tilde{\mathbf{B}}^T \boldsymbol{\Sigma}^{-1} \right) \\ &\quad - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1} \tilde{\mathbf{B}} \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}} \right) \tilde{\mathbf{B}}^T \boldsymbol{\Sigma}^{-1} \right) - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big]\end{aligned}$$

The first set of terms are simply the log-likelihood of a multivariate normal distribution, resulting in equation (34), provided in the main section.

$$\mathcal{L}(\phi) = \ln \mathcal{N}(\mathbf{y} \mid \tilde{\mathbf{B}}\mathbf{A}\mathbf{m}, \boldsymbol{\Sigma}^2) - \frac{1}{2} \mathrm{tr} \left(\boldsymbol{\Sigma}^{-1} \tilde{\mathbf{B}} \left(\mathbf{A}\mathbf{S}\mathbf{A}^T + \mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}} \right) \tilde{\mathbf{B}}^T \boldsymbol{\Sigma}^{-1} \right) - D_{KL}\big[q(\mathbf{u}) \parallel p(\mathbf{u}) \big]$$

B.2 Derivation of the LCM predictive density

This is obtained by marginalising the inducing values $\mathbf{u}$ out of the joint variational distribution (30):

$$q(\mathbf{f}) = \int_{\mathbf{u}} p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u}) \, d\mathbf{u}$$

Explicitly writing out the gaussian density functions for $p(\mathbf{f} \mid \mathbf{u})$ and $q(\mathbf{u})$ using (28) and (30):

$$q(\mathbf{f}) = \int_{\mathbf{u}} \exp\left(-\frac{1}{2}\ln\left|2\pi\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| - \frac{1}{2}\left(\mathbf{f} - \mathbf{A}\mathbf{u}\right)^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\left(\mathbf{f} - \mathbf{A}\mathbf{u}\right)\right.$$
$$\left. -\frac{1}{2}\ln\left|2\pi\mathbf{S}\right| - \frac{1}{2}\left(\mathbf{u} - \mathbf{m}\right)^T\mathbf{S}^{-1}\left(\mathbf{u} - \mathbf{m}\right)\right) d\mathbf{u}$$

Leaving the treatment of the determinants aside for the moment, the quadratic forms can be expanded as follows:

$$q(\mathbf{f}) = \int_{\mathbf{u}} \exp\left(-\frac{1}{2}\ln\left|2\pi\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| - \frac{1}{2}\ln\left|2\pi\mathbf{S}\right|\right.$$
$$-\frac{1}{2}\left[\mathbf{f}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{f} + \mathbf{u}^T\mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\mathbf{u} - 2\mathbf{f}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\mathbf{u}\right]$$
$$\left. -\frac{1}{2}\left[\mathbf{u}^T\mathbf{S}^{-1}\mathbf{u} + \mathbf{m}^T\mathbf{S}^{-1}\mathbf{m} - 2\mathbf{m}^T\mathbf{S}^{-1}\mathbf{u}\right]\right) d\mathbf{u}$$

Setting $\mathbf{\Gamma}^{-1} = \mathbf{S}^{-1} + \mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}$ and gathering the terms containing $\mathbf{u}$:

$$q(\mathbf{f}) = \int_{\mathbf{u}} \exp\left(-\frac{1}{2}\ln\left|2\pi\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| - \frac{1}{2}\ln\left|2\pi\mathbf{S}\right|\right.$$
$$-\frac{1}{2}\left[\mathbf{u}^T\mathbf{\Gamma}^{-1}\mathbf{u} - 2\left(\mathbf{f}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A} + \mathbf{m}^T\mathbf{S}^{-1}\right)\mathbf{u}\right]$$
$$\left. -\frac{1}{2}\left[\mathbf{f}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{f} + \mathbf{m}^T\mathbf{S}^{-1}\mathbf{m}\right]\right) d\mathbf{u}$$

Completing the square using $\mathbf{g} = \mathbf{\Gamma}\left(\mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{f} + \mathbf{S}^{-1}\mathbf{m}\right)$ results in:

$$q(\mathbf{f}) = \int_{\mathbf{u}} \exp\left(-\frac{1}{2}\ln\left|2\pi\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| - \frac{1}{2}\ln\left|2\pi\mathbf{S}\right| - \frac{1}{2}\left(\mathbf{u} - \mathbf{g}\right)^T\mathbf{\Gamma}^{-1}\left(\mathbf{u} - \mathbf{g}\right)\right.$$
$$-\frac{1}{2}\left[\mathbf{f}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{f} + \mathbf{m}^T\mathbf{S}^{-1}\mathbf{m}\right]$$
$$\left. -\frac{1}{2}\left[-\left(\mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{f} + \mathbf{S}^{-1}\mathbf{m}\right)^T\mathbf{\Gamma}\left(\mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{f} + \mathbf{S}^{-1}\mathbf{m}\right)\right]\right) d\mathbf{u}$$

Expanding the quadratic form around $\mathbf{\Gamma}$ and factorising like terms produces:

$$q(\mathbf{f}) = \int_{\mathbf{u}} \exp\left(-\frac{1}{2}\ln\left|2\pi\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| - \frac{1}{2}\ln\left|2\pi\mathbf{S}\right| - \frac{1}{2}\left(\mathbf{u} - \mathbf{g}\right)^T\mathbf{\Gamma}^{-1}\left(\mathbf{u} - \mathbf{g}\right)\right.$$
$$\left. -\frac{1}{2}\left[\mathbf{f}^T\mathbf{\Psi}^{-1}\mathbf{f} + \mathbf{m}^T\mathbf{S}^{-1}\left(\mathbf{S} - \mathbf{\Gamma}\right)\mathbf{S}^{-1}\mathbf{m} - 2\mathbf{f}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\mathbf{\Gamma}\mathbf{S}^{-1}\mathbf{m}\right]\right) d\mathbf{u} \tag{A-1}$$

With:

$$\mathbf{\Psi}^{-1} = \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1} - \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}$$

In order to clean up the expression for $q(\mathbf{f})$ further, we need to obtain more tractable expressions for $\mathbf{\Psi}$ and $\mathbf{\Gamma}$. This can be obtained by using the Woodbury matrix identity twice. First, we use it to obtain the expression for $\mathbf{\Psi}$, using the fact that $\mathbf{\Gamma} = \left(\mathbf{S}^{-1} + \mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\right)^{-1}$:

$$\mathbf{\Psi} = \mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}} + \mathbf{A}\mathbf{S}\mathbf{A}^T$$

Second, we apply the Woodbury identity on $\mathbf{\Gamma}^{-1}$ to get:

$$\boldsymbol{\Gamma} \;=\; \mathbf{S} - \mathbf{S}\mathbf{A}^T \left[\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}} + \mathbf{A}\mathbf{S}\mathbf{A}^T\right]^{-1} \mathbf{A}\mathbf{S} \;=\; \mathbf{S} - \mathbf{S}\mathbf{A}^T \boldsymbol{\Psi}^{-1} \mathbf{A}\mathbf{S}$$

Rearranging the expression for $\boldsymbol{\Gamma}$ allows us to simplify the $\mathbf{S}^{-1}(\mathbf{S} - \boldsymbol{\Gamma})\mathbf{S}^{-1}$ term in (A-1):

$$\mathbf{S}^{-1}(\mathbf{S} - \boldsymbol{\Gamma})\mathbf{S}^{-1} = \mathbf{A}^T \boldsymbol{\Psi}^{-1} \mathbf{A}$$

Similarly, rearranging the definition of $\boldsymbol{\Gamma}^{-1}$ gives $\mathbf{S}^{-1} = \boldsymbol{\Gamma}^{-1} - \mathbf{A}^T \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1} \mathbf{A}$. This allows us to simplify the $\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1} \mathbf{A}\boldsymbol{\Gamma}\mathbf{S}^{-1}$ term in (A-1):

$$\begin{aligned}
\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1} \mathbf{A}\boldsymbol{\Gamma}\mathbf{S}^{-1} &= \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1} \mathbf{A}\boldsymbol{\Gamma} \left[\boldsymbol{\Gamma}^{-1} - \mathbf{A}^T \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1} \mathbf{A}\right] \\
&= \left[\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1} - \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1} \mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^T \left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\right] \mathbf{A} \\
&= \boldsymbol{\Psi}^{-1}\mathbf{A}
\end{aligned}$$

Replacing these in the expression for $q(\mathbf{f})$ enables the closing of the second square:

$$\begin{aligned}
q(\mathbf{f}) = \int_{\mathbf{u}} \exp\Bigg(&-\frac{1}{2}\ln\left|2\pi\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| - \frac{1}{2}\ln\left|2\pi\mathbf{S}\right| - \frac{1}{2}\left(\mathbf{u}-\mathbf{g}\right)^T \boldsymbol{\Gamma}^{-1}\left(\mathbf{u}-\mathbf{g}\right) \\
&-\frac{1}{2}\left[\mathbf{f}^T\boldsymbol{\Psi}^{-1}\mathbf{f} + \mathbf{m}^T\mathbf{A}^T\boldsymbol{\Psi}^{-1}\mathbf{A}\mathbf{m} - 2\boldsymbol{\Psi}^{-1}\mathbf{A}\mathbf{m}\right]\Bigg)\, d\mathbf{u} \\
q(\mathbf{f}) = \int_{\mathbf{u}} \exp\Bigg(&-\frac{1}{2}\ln\left|2\pi\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| - \frac{1}{2}\ln\left|2\pi\mathbf{S}\right| - \frac{1}{2}\left(\mathbf{u}-\mathbf{g}\right)^T \boldsymbol{\Gamma}^{-1}\left(\mathbf{u}-\mathbf{g}\right) \\
&-\frac{1}{2}\left(\mathbf{f}-\mathbf{A}\mathbf{m}\right)^T\boldsymbol{\Psi}^{-1}\left(\mathbf{f}-\mathbf{A}\mathbf{m}\right)\Bigg)\, d\mathbf{u}
\end{aligned} \tag{A-2}$$

The last step is to tidy up the determinant terms that we initially set aside. First, the determinant of $\boldsymbol{\Gamma}^{-1}$ can be factorised as follows:

$$\begin{aligned}
\left|\boldsymbol{\Gamma}^{-1}\right| &= \left|\mathbf{S}^{-1} + \mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\right| \\
&= \left|\mathbf{I} + \mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\mathbf{S}\right| \left|\mathbf{S}^{-1}\right|
\end{aligned}$$

The determinant of $\boldsymbol{\Psi}$ can similarly be factorised, using the matrix determinant lemma:

$$\begin{aligned}
\left|\boldsymbol{\Psi}\right| &= \left|\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right) + \mathbf{A}\mathbf{S}\mathbf{A}^T\right| \\
&= \left|\mathbf{S}^{-1} + \mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\right| \left|\mathbf{S}\right| \left|\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| \\
&= \left|\mathbf{I} + \mathbf{A}^T\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)^{-1}\mathbf{A}\mathbf{S}\right| \left|\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right|
\end{aligned}$$

This allows us to establish the following equalities:

$$\left|\boldsymbol{\Psi}\right| \left|\boldsymbol{\Gamma}\right| = \frac{\left|\boldsymbol{\Psi}\right|}{\left|\boldsymbol{\Gamma}^{-1}\right|} = \frac{\left|\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right|}{\left|\mathbf{S}^{-1}\right|} = \left|\left(\mathbf{K}^{\mathbf{x},\mathbf{x}} - \mathbf{Q}^{\mathbf{x},\mathbf{x}}\right)\right| \left|\mathbf{S}\right|$$

The expression (A-2) can now explicitly be written as the product of two multivariate normal distributions, where the distribution of $\mathbf{f}$ no longer depends on $\mathbf{u}$, making the marginalisation straightforward.

$$\begin{aligned}
q(\mathbf{f}) = &\exp\left(-\frac{1}{2}\ln\left|2\pi\boldsymbol{\Psi}\right| - \frac{1}{2}\left(\mathbf{f}-\mathbf{A}\mathbf{m}\right)^T\boldsymbol{\Psi}^{-1}\left(\mathbf{f}-\mathbf{A}\mathbf{m}\right)\right) \\
&\times \int_{\mathbf{u}} \exp\left(-\frac{1}{2}\ln\left|2\pi\boldsymbol{\Gamma}\right| - \frac{1}{2}\left(\mathbf{u}-\mathbf{g}\right)^T\boldsymbol{\Gamma}^{-1}\left(\mathbf{u}-\mathbf{g}\right)\right)\, d\mathbf{u}
\end{aligned}$$

The integral with respect to $\mathbf{u}$ sums to one, resulting in $q(\mathbf{f}) = N\left(\mathbf{f} \mid \mathbf{A}\mathbf{m}, \boldsymbol{\Psi}\right)$, which is equation (36).

C Full VAR Monte Carlo tables

Table A-1: Monte Carlo results for BEGRS estimation of VAR specification, 1000 inducing points

		$\rho = 0$			$\rho = 0.25$			$\rho = 0.5$		
Parameter	value	$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}	$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}	$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}
$B^*_{1,1}$	0.154	−0.007	−0.003	−0.115	0.003	−0.003	0.057	−0.020	−0.006	−0.295
		(0.055)	(0.057)		(0.059)	(0.061)		(0.065)	(0.068)	
$B^*_{1,2}$	0.113	−0.006	0.002	−0.094	−0.024	0.003	−0.328	0.003	0.003	0.035
		(0.062)	(0.063)		(0.070)	(0.074)		(0.077)	(0.081)	
$B^*_{1,3}$	0.409	−0.037	0.002	−0.587	−0.000	0.003	−0.001	−0.015	0.002	−0.209
		(0.061)	(0.063)		(0.077)	(0.079)		(0.071)	(0.073)	
$B^*_{1,4}$	−0.531	0.035	−0.000	0.644	0.007	−0.006	0.090	0.044	−0.002	0.514
		(0.053)	(0.055)		(0.065)	(0.073)		(0.078)	(0.086)	
$B^*_{2,1}$	−0.496	0.035	−0.001	0.626	0.034	−0.001	0.578	0.056	−0.002	0.832
		(0.053)	(0.055)		(0.059)	(0.059)		(0.064)	(0.067)	
$B^*_{2,2}$	−0.196	0.005	−0.004	0.080	−0.028	−0.005	−0.395	0.026	−0.003	0.326
		(0.060)	(0.061)		(0.070)	(0.072)		(0.075)	(0.079)	
$B^*_{2,3}$	0.081	−0.003	−0.001	−0.041	0.010	−0.000	0.137	0.014	0.000	0.199
		(0.064)	(0.067)		(0.072)	(0.076)		(0.069)	(0.070)	
$B^*_{2,4}$	−0.065	0.027	0.007	0.479	−0.008	0.007	−0.116	0.010	0.004	0.123
		(0.053)	(0.056)		(0.065)	(0.069)		(0.083)	(0.085)	
$B^*_{3,1}$	−0.155	−0.013	−0.005	−0.233	−0.012	−0.005	−0.218	−0.000	−0.005	−0.003
		(0.056)	(0.056)		(0.055)	(0.055)		(0.068)	(0.070)	
$B^*_{3,2}$	−0.273	−0.026	0.006	−0.407	−0.026	0.001	−0.381	0.033	0.007	0.378
		(0.063)	(0.063)		(0.068)	(0.070)		(0.085)	(0.087)	
$B^*_{3,3}$	−0.068	−0.029	−0.004	−0.437	−0.034	−0.006	−0.465	0.004	−0.002	0.051
		(0.064)	(0.066)		(0.074)	(0.073)		(0.078)	(0.077)	
$B^*_{3,4}$	−0.128	0.009	−0.002	0.168	0.007	−0.000	0.098	−0.065	−0.010	−0.733
		(0.053)	(0.054)		(0.067)	(0.067)		(0.086)	(0.089)	
$B^*_{4,1}$	−0.338	−0.003	0.007	−0.046	0.026	0.003	0.486	0.036	−0.002	0.572
		(0.056)	(0.057)		(0.052)	(0.053)		(0.058)	(0.062)	
$B^*_{4,2}$	−0.597	0.048	0.003	0.792	0.010	−0.000	0.159	0.045	0.002	0.619
		(0.056)	(0.061)		(0.053)	(0.062)		(0.062)	(0.072)	
$B^*_{4,3}$	0.195	−0.056	−0.003	−0.844	0.020	0.003	0.298	−0.017	0.004	−0.259
		(0.064)	(0.066)		(0.065)	(0.067)		(0.065)	(0.067)	
$B^*_{4,4}$	0.459	−0.011	−0.009	−0.195	0.008	−0.008	0.133	−0.044	−0.009	−0.544
		(0.054)	(0.057)		(0.060)	(0.063)		(0.074)	(0.080)	

- Note: $\hat{B}^*_{gp}$ refers to the BEGRS estimate of the VAR parameters, $\hat{B}^*_{ml}$ refers to standard maximum likelihood estimates, with Monte Carlo standard errors in parenthesis (1000 replications). T-statistics are calculated under the null that $\hat{B}^*_{gp}$ follows the sampling distribution of $\hat{B}^*_{ml}$.

Table A-2: Monte Carlo results for BEGRS estimation of VAR specification, 500 inducing points

Parameter	value	$\rho = 0$			$\rho = 0.25$			$\rho = 0.5$		
		$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}	$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}	$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}
$B^*_{1,1}$	0.154	0.023	−0.003	0.404	0.038	−0.003	0.624	−0.022	−0.006	−0.328
		(0.057)	(0.057)		(0.060)	(0.061)		(0.067)	(0.068)	
$B^*_{1,2}$	0.113	0.016	0.002	0.258	0.006	0.003	0.079	0.015	0.003	0.182
		(0.062)	(0.063)		(0.070)	(0.074)		(0.079)	(0.081)	
$B^*_{1,3}$	0.409	−0.000	0.002	−0.005	−0.002	0.003	−0.030	0.000	0.002	0.004
		(0.064)	(0.063)		(0.077)	(0.079)		(0.071)	(0.073)	
$B^*_{1,4}$	−0.531	0.000	−0.000	0.009	−0.004	−0.006	−0.048	0.003	−0.002	0.040
		(0.054)	(0.055)		(0.066)	(0.073)		(0.080)	(0.086)	
$B^*_{2,1}$	−0.496	−0.016	−0.001	−0.282	0.034	−0.001	0.567	0.039	−0.002	0.588
		(0.056)	(0.055)		(0.059)	(0.059)		(0.066)	(0.067)	
$B^*_{2,2}$	−0.196	−0.034	−0.004	−0.561	0.003	−0.005	0.048	0.039	−0.003	0.490
		(0.062)	(0.061)		(0.071)	(0.072)		(0.088)	(0.079)	
$B^*_{2,3}$	0.081	−0.038	−0.001	−0.569	−0.007	−0.000	−0.097	−0.032	0.000	−0.461
		(0.068)	(0.067)		(0.074)	(0.076)		(0.069)	(0.070)	
$B^*_{2,4}$	−0.065	0.026	0.007	0.472	0.046	0.007	0.662	−0.017	0.004	−0.199
		(0.055)	(0.056)		(0.066)	(0.069)		(0.086)	(0.085)	
$B^*_{3,1}$	−0.155	0.144	−0.005	2.553^{**}	0.025	−0.005	0.447	−0.211	−0.005	-3.039^{***}
		(0.396)	(0.056)		(0.053)	(0.055)		(0.403)	(0.070)	
$B^*_{3,2}$	−0.273	0.140	0.006	2.241^{**}	0.015	0.001	0.217	0.012	0.007	0.135
		(0.420)	(0.063)		(0.069)	(0.070)		(0.313)	(0.087)	
$B^*_{3,3}$	−0.068	0.122	−0.004	1.832^{*}	−0.032	−0.006	−0.431	0.452	−0.002	5.871^{***}
		(0.369)	(0.066)		(0.068)	(0.073)		(0.376)	(0.077)	
$B^*_{3,4}$	−0.128	−0.175	−0.002	-3.262^{***}	0.006	−0.000	0.090	−0.384	−0.010	-4.305^{***}
		(0.340)	(0.054)		(0.067)	(0.067)		(0.241)	(0.089)	
$B^*_{4,1}$	−0.338	0.041	0.007	0.712	0.007	0.003	0.131	−0.024	−0.002	−0.391
		(0.066)	(0.057)		(0.051)	(0.053)		(0.066)	(0.062)	
$B^*_{4,2}$	−0.597	0.006	0.003	0.100	0.047	−0.000	0.761	0.029	0.002	0.400
		(0.053)	(0.061)		(0.056)	(0.062)		(0.062)	(0.072)	
$B^*_{4,3}$	0.195	0.008	−0.003	0.114	−0.002	0.003	−0.034	0.007	0.004	0.109
		(0.071)	(0.066)		(0.064)	(0.067)		(0.068)	(0.067)	
$B^*_{4,4}$	0.459	0.010	−0.009	0.185	−0.005	−0.008	−0.087	−0.051	−0.009	−0.636
		(0.061)	(0.057)		(0.059)	(0.063)		(0.080)	(0.080)	

- Note: $\hat{B}^*_{gp}$ refers to the BEGRS estimate of the VAR parameters, $\hat{B}^*_{ml}$ refers to standard maximum likelihood estimates, with Monte Carlo standard errors in parenthesis (1000 replications). T-statistics are calculated under the null that $\hat{B}^*_{gp}$ follows the sampling distribution of $\hat{B}^*_{ml}$.

Table A-3: Monte Carlo results for BEGRS estimation of VAR specification, 250 inducing points

Parameter	value	$\rho = 0$			$\rho = 0.25$			$\rho = 0.5$		
		$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}	$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}	$E[\hat{B}^*_{gp}] - B^*$	$E[\hat{B}^*_{ml}] - B^*$	t_{val}
$B^*_{1,1}$	0.154	−0.027	−0.003	−0.466	−0.009	−0.003	−0.146	0.000	−0.006	0.006
		(0.057)	(0.057)		(0.063)	(0.061)		(0.067)	(0.068)	
$B^*_{1,2}$	0.113	0.021	0.002	0.332	0.022	0.003	0.295	−0.008	0.003	−0.104
		(0.063)	(0.063)		(0.071)	(0.074)		(0.078)	(0.081)	
$B^*_{1,3}$	0.409	0.011	0.002	0.177	−0.025	0.003	−0.317	0.001	0.002	0.009
		(0.064)	(0.063)		(0.077)	(0.079)		(0.073)	(0.073)	
$B^*_{1,4}$	−0.531	0.031	−0.000	0.575	−0.003	−0.006	−0.035	0.020	−0.002	0.231
		(0.052)	(0.055)		(0.066)	(0.073)		(0.080)	(0.086)	
$B^*_{2,1}$	−0.496	−0.017	−0.001	−0.307	0.027	−0.001	0.448	0.000	−0.002	0.004
		(0.054)	(0.055)		(0.062)	(0.059)		(0.064)	(0.067)	
$B^*_{2,2}$	−0.196	0.001	−0.004	0.017	0.010	−0.005	0.135	−0.006	−0.003	−0.078
		(0.059)	(0.061)		(0.073)	(0.072)		(0.079)	(0.079)	
$B^*_{2,3}$	0.081	−0.015	−0.001	−0.221	−0.010	−0.000	−0.127	0.022	0.000	0.306
		(0.064)	(0.067)		(0.076)	(0.076)		(0.070)	(0.070)	
$B^*_{2,4}$	−0.065	0.045	0.007	0.809	0.021	0.007	0.310	0.007	0.004	0.080
		(0.054)	(0.056)		(0.068)	(0.069)		(0.088)	(0.085)	
$B^*_{3,1}$	−0.155	0.038	−0.005	0.671	0.008	−0.005	0.149	0.058	−0.005	0.840
		(0.054)	(0.056)		(0.068)	(0.055)		(0.072)	(0.070)	
$B^*_{3,2}$	−0.273	0.024	0.006	0.383	0.067	0.001	0.969	0.005	0.007	0.059
		(0.061)	(0.063)		(0.071)	(0.070)		(0.090)	(0.087)	
$B^*_{3,3}$	−0.068	−0.000	−0.004	−0.007	−0.004	−0.006	−0.058	−0.021	−0.002	−0.272
		(0.064)	(0.066)		(0.073)	(0.073)		(0.079)	(0.077)	
$B^*_{3,4}$	−0.128	0.007	−0.002	0.121	0.011	−0.000	0.164	−0.032	−0.010	−0.360
		(0.052)	(0.054)		(0.066)	(0.067)		(0.090)	(0.089)	
$B^*_{4,1}$	−0.338	0.018	0.007	0.308	0.477	0.003	8.988***	0.585	−0.002	9.407***
		(0.056)	(0.057)		(0.478)	(0.053)		(0.351)	(0.062)	
$B^*_{4,2}$	−0.597	0.058	0.003	0.950	0.561	−0.000	9.034***	1.037	0.002	14.336***
		(0.056)	(0.061)		(0.355)	(0.062)		(0.216)	(0.072)	
$B^*_{4,3}$	0.195	−0.014	−0.003	−0.213	−0.157	0.003	−2.339**	−0.425	0.004	−6.338***
		(0.064)	(0.066)		(0.315)	(0.067)		(0.444)	(0.067)	
$B^*_{4,4}$	0.459	−0.038	−0.009	−0.670	0.056	−0.008	0.885	−0.360	−0.009	−4.494***
		(0.054)	(0.057)		(0.189)	(0.063)		(0.458)	(0.080)	

- Note: $\hat{B}^*_{gp}$ refers to the BEGRS estimate of the VAR parameters, $\hat{B}^*_{ml}$ refers to standard maximum likelihood estimates, with Monte Carlo standard errors in parenthesis (1000 replications). T-statistics are calculated under the null that $\hat{B}^*_{gp}$ follows the sampling distribution of $\hat{B}^*_{ml}$.