Taylor & Francis
Taylor & Francis Group

Technometrics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/utch20

Gaussian Process Emulation for High-Dimensional Coupled Systems

Tamara Dolski, Elaine T. Spiller & Susan E. Minkoff

To cite this article: Tamara Dolski, Elaine T. Spiller & Susan E. Minkoff (28 Mar 2024): Gaussian Process Emulation for High-Dimensional Coupled Systems, Technometrics, DOI: 10.1080/00401706.2024.2322651

To link to this article: https://doi.org/10.1080/00401706.2024.2322651

View supplementary material

Published online: 28 Mar 2024.

Submit your article to this journal

Article views: 47

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=utch20

TECHNOMETRICS
2024, VOL. 00, NO. 0, 1–15
https://doi.org/10.1080/00401706.2024.2322651

Taylor & Francis
Taylor & Francis Group

Check for updates

Gaussian Process Emulation for High-Dimensional Coupled Systems

Tamara Dolski[a], Elaine T. Spiller[b], and Susan E. Minkoff[a]

[a]Department of Mathematical Sciences, University of Texas at Dallas, Richardson, TX; [b]Department of Mathematical and Statistical Sciences, Marquette University, Milwaukee, WI

ABSTRACT
Complex coupled multiphysics simulations are ubiquitous in science and engineering. Evaluating these numerical simulators is often costly which limits our ability to run them sufficiently often for forward uncertainty quantification. Furthermore outputs are generally not scalar quantities but depend on space and/or time. Gaussian process emulators are statistical surrogates which can approximate the output of the complex computer models at untested inputs quickly while also providing uncertainty information about the accuracy of evaluating the emulator rather than the full physical model. GP emulators were originally developed in the context of scalar output from a single physical model but have since been extended to vector-valued quantities of interest (parallel partial emulators) and to coupled physics by connecting two independent emulators, one for each type of physics (linked emulation). The parallel partial linked GP emulator (PPLE) developed in this work combines the efficiency of a shared correlation structure with the accuracy of linked emulators to produce a new tool for emulating compositions of functions with vector-valued output. The PPLE applied to two numerical experiments out-performs direct emulation of the output composite function producing results with smaller average prediction error and variance.

ARTICLE HISTORY
Received March 2023
Accepted February 2024

KEYWORDS
Coupled process modeling; High-dimensional systems; Porous media flow; Statistical surrogates

1. Introduction

Coupled simulations of multiphysics systems are ubiquitous across scientific domains (Brown et al. 2008; Keyes et al. 2013), but such coupling poses significant challenges for uncertainty quantification. Gaussian process surrogates of computer models have developed into a powerful class of tools for uncertainty quantification since their introduction as emulators of complex computer models (Currin et al. 1988; Sacks, Schiller, and Welch 1989; Welch et al. 1992). These statistical surrogates allow one to very quickly approximate computationally intensive deterministic simulators via nonparametric interpolation at different input settings. Further, the variance of the statistical surrogate offers a built-in mechanism to quantify the uncertainty of evaluating the surrogate in place of the simulator. (For a thorough overview, see Santner, Williams, and Notz 2018.) Yet challenges remain for constructing Gaussian process (GP) emulators of coupled simulators. A significant obstacle for multiphysics simulators is that, often, the output of the individual simulators are discretized fields rather than scalars. In this work, we introduce a GP-based surrogate suitable for emulating coupled simulators with high-dimensional output.

As one example of a coupled simulator application, when hydrocarbons are extracted from subsurface reservoirs during production, pore pressure depletion may lead to compaction. In many cases, this compaction is small, but in well-known instances, this subsidence has caused well failures, platform collapse, and even loss of human lives. Examples where the subsidence was significant enough to cause facilities damage include the Ekofisk Field in the North Sea (Yudovich and Morgan 1989; Dangerfield 1992) and Belridge in California (Fredrich et al. 1996). Sophisticated numerical simulation of these fields (which may include multiphase flow and nonlinear, inelastic deformation) is important in order to predict and hence prevent such catastrophic situations in the future (Gutierrez, Lewis, and Masters 2001; Minkoff et al. 2003, 2004).

As described in Minkoff and Kridler (2006), simulator coupling generally falls into one of the following categories: full, loose, or iterative coupling. In full coupling, there is a single set of equations which connects flow to mechanics tightly (in other words, the flow equations contain terms for deformation and vice versa, see Lewis and Sukirman 1993b, 1993a; Lewis and Ghafouri 1997; Osorio, Chen, and Teufel 1999; Gutierrez and Lewis 2002). Fully coupled simulators are in some cases considered the most accurate as there is no time step delay in solving one equation and then the other, but they almost always require that modelers make restrictive simplifications of the physics to develop the single system of equations and the numerical discretization to solve this system. Therefore, full coupling has the potential to inflate epistemic uncertainties in its description of reality. A further limitation of this approach is that it usually requires a single spatial domain and a single set of time steps for the different types of physics, neither of which is realistic. Loose coupling provides an attractive alternative to full coupling by linking two independent codes through an interface

CONTACT Elaine Spiller elaine.spiller@marquette.edu Department of Mathematical and Statistical Sciences, Marquette University, Milwaukee, WI.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/TECH.
© 2024 American Statistical Association and the American Society for Quality

(Minkoff et al. 2003). Settari and Mourits (1994), Dean et al. (2003), and Kim, Tchelepi, and Juanes (2011) discuss iterative schemes which attempt to ensure that a loosely-coupled solution converges either to the fully-coupled ("true") solution or to a stable value within a specified tolerance. Iterative coupling is similar to loose coupling in that the two (or more) physical simulators are run sequentially, passing updated information between the simulators. However, in loose coupling one does not repeat the same simulation for a fixed set of time steps until a convergence tolerance is reached as in iterative coupling. Loose-coupling schemes ensure that sophisticated physics in each type of simulator and the decades of work put into these legacy codes can be brought to bear to realistically model coupled phenomena. In this article we develop a framework for investigating UQ based on GP emulation of coupled, high-dimensional multiphysics models and demonstrate GP efficacy in this context on a prototype two-way loosely coupled model for flow and deformation.

The computational science community has developed two general global statistical emulator approaches for computer models with multi-variate output that lend themselves to coupled multiphysics systems. (Continuous partial differential equations (pde's) are discretized using techniques like the finite difference and finite element methods when solved on computers—the idea of a limit can only be approximated on a finite precision machine—and hence numerical solutions of pde's are available only on a discrete grid of points.) The first method involves a dimension reduction to simulator output by fitting basis functions or low-rank representations to the multi-variate (often spatial or temporal) output, and emulating only the associated coefficients with GPs. The second involves treating multi-variate output as independent, but with a shared correlation structure, and emulating the whole field at once. In the former approach, researchers have introduced wavelet (Bayarri et al. 2007), spline (Bowman and Woods 2016), and principal component representations of field output (Higdon et al. 2008). These strategies have been widely adopted in the computer modeling community and have proven to be powerful tools for emulating simulators that yield high-dimensional output. One drawback of emulating coefficients of a reduced-dimension approximation to output fields is that emulator uncertainty now arises both from limited knowledge of the simulator and approximation error from employing a lower-dimensional model that may not preserve the physical properties of the simulator. The other general global strategy of GP emulation of multi-variate output is parallel partial emulation (PPE) (Gu and Berger 2016). PPEs treat each output component independently, and only assume a shared correlation structure with respect to simulator inputs. (Other parameters, like mean regressors, are assumed independent.) Further the predictive mean of the PPE inherits spatial correlations present in the physical simulator without modeling them directly. Thus, it acts as an interpolator for high-dimensional output effectively preserving global structures from the physics in the PPE approximation (Gao and Pitman 2023).

The composition of GP emulators is a natural approach for emulating loosely coupled simulators. It is well established that the composition of two Gaussian processes is not itself a Gaussian process (Damianou and Lawrence 2013; Girard et al. 2003). Yet the composite process emulator can be written as an integral (albeit with no available closed form expression). One can compute Monte Carlo approximations of this composite process (Sanson, Le Maitre, and Congedo 2019), or linearize the composition (Marque-Pucheu, Perrin, and Garnier 2020). The effective result of the linearization is a composition of the GP emulator predictive means of each simulator which may be reasonable if the predictive variance is small. Another approximation to the composition of GP's involves calculating the first two moments of the composite process and considering the so-called Linked GP emulator to be the Normal approximation to the composite process (Kyzyurova, Berger, and Wolpert 2018; Abdelfatah, Bao, and Terejanu 2018; Ming and Guillas 2021).

Here we extend the emulation methodology to the development and application of parallel partial linked GP emulation (PPLE) for coupled vector-valued models. (See Figure 2 for a visualization of linked emulation.) Specifically, we derive the mean and variance for the PPLE which is the main contribution of this work. We consider the first two moments of the random variable that result from the composition of emulators for the component functions under consideration. In this context the "outside" function and emulator are vector-valued, and we update the expectations to derive the mean and variance assuming the "outside" emulator is a parallel partial emulator. The linked emulator setup also enables one to pass fields between simulators via dimension reduction on the output of the "inside" function.

We contrast the accuracy of the PPLE with a composite emulator which directly emulates the final output rather than linking two independent emulators. We consider two main example problems. The first example is a composition of two vector-valued trigonometric functions. In this case we clearly see that the linked emulator has smaller prediction error and credible intervals than the composite emulator. The second example is a more realistic physical model of coupled fluid flow and mechanical deformation in a column of mud with impermeable boundaries everywhere but at the top of the column. At the initial time, a load is dropped on the column, and as the mud compacts, the single-phase fluid (water) is able to exit out of the top of the column. In this coupling we assume that the fluid pressure change is used as a load for the mechanical deformation. The resulting displacement solution can be differentiated to produce a strain that we use to update the porosity of the mud for flow solves at later times. As the porosity along the column passed from the inner function (mechanics) to the outer function (fluid flow) is high-dimensional, we make use of dimension reduction via principal component analysis (PCA) or gradient-based kernel dimension reduction (gKDR) (Fukumizu and Leng 2014) to only send a small subset of modes that represent porosity in the linking step of the PPLE. No dimension reduction is done to represent the final output (pressure) in the column. In this example, we also compare the efficacy and efficiency of the PPLE to four other approaches: many independent emulators fit to scalar-output components (both composite and linked), and local approximate GP (laGP) regression (Gramacy and Apley 2015; Gramacy 2016) (both composite and linked). The PPLE consistently outperforms all of these other methods.

The remainder of the article starts in Section 2.1 with a review of the basic GP methodology for a single type of physics with scalar output quantities. Then in Section 2.2 we explain how emulators of a single physical model can be extended to vector-valued quantities of interest via parallel partial emulation (Gu and Berger 2016). In Section 2.3 we overview the literature for linked emulation (Kyzyurova, Berger, and Wolpert 2018; Ming and Guillas 2021) in which a scalar quantity of interest can be emulated for more than one type of physical simulator coupled together. We then present the new mean and variance theory for the parallel partial linked emulator in Section 3, and we illustrate the success of this emulator over composite emulation in Section 4 with the two sets of numerical experiments described above.

2. Background and Motivation

The task of approximating nonlinear multiphysics systems is very challenging. To demonstrate this point, let us consider the simplest composite function $\eta(x) = g\big(f(x)\big)$ where the "outside" function g, the "inside" function f, and hence the composite function η, are nonlinear 1-D functions of a single variable. Expanding η in a Taylor series about $x = x^{\mathrm{D}}$, we have $\eta(x) \approx L(x) + R(x)$ with the linear Taylor series $L(x) = g\big(f(x^{\mathrm{D}})\big) + (x - x^{\mathrm{D}})g'\big(f(x^{\mathrm{D}})\big)f'(x^{\mathrm{D}})$ and the Taylor Series remainder for η given by

$$R(x) = \frac{1}{2}(x - x^{\mathrm{D}})^2\Big[(f')^2(\xi_{in})g''\big(\xi_{out}\big) + f''(\xi_{in})g'\big(\xi_{out}\big)\Big]. \quad (1)$$

Here ξ_{in} and ξ_{out} are Mean Value Theorem points for the respective derivatives of f and g. The scaling of the quadratic term in the local approximation error given by $R(x)$ can be very large if f and g are strongly nonlinear functions. Instead, if we approximate f and g individually, the equivalent local approximation error terms are scaled by f'' and g''.

We consider an example from Kyzyurova, Berger, and Wolpert (2018) with the functions $f(x) = 3x + 5\cos(5x)$, $g(x) = \cos(7/5x) - x$, and $\eta(x) = g(f(x))$. In Figure 1 we plot the nonlinear portions of $\eta(x)$, $f(x)$, and $g(x)$ along with the approximation error $R(x)$ with $x^{\mathrm{D}} = 0$ for each function. It is clear that the local approximation error for η grows much more quickly as we move away from the interpolation point than it does for f and g.

When we seek to approximate a nonlinear multiphysics system with Gaussian process emulators, there are two primary advantages in taking a linked emulator approach that fits GPs to f and g separately. The first follows from equation 1 where the error bound of η depends on approximating the slope and curvature of the inside function f. The construction of a GP emulator directly on the composite function η is blind to information about f, while a linked emulator has the advantage of modeling the nonlinearities of f directly. The second advantage of linked GP emulation is that the predictive variance estimate provided by the GP emulator of f is used directly in the linked GP estimation of η (Kyzyurova, Berger, and Wolpert 2018; Ming and Guillas 2021).

Below we present a self-consistent treatment of the background on standard Gaussian process emulators, parallel partial emulators for vector-valued output, and linked emulation of composite functions. While this material can be found in some form in the references cited below, the notation varies between references. Thus, here we present a unified overview with consistent notation that leads naturally into our discussion of the new theory we have developed for parallel partial linked GP emulators in Section 3.

2.1. Gaussian Process Emulation of a Single Computer Model with Scalar Output

The standard GP emulator treats computer model output as a draw of a random function of inputs of interest (parameters, initial conditions, boundary conditions, etc.) As computer model evaluations are often computationally intensive, they are only exercised at a limited number of input combinations (referred to collectively as a design). The objective of the emulator is to first learn about a random function that fits the simulator training data (i.e., the input design and corresponding simulator output) and then evaluate the GP in place of running the simulator at untested input combinations. In short, a GP emulator's mean is an interpolator that goes through design-response pairs from the full physics based simulator. Further, the GP's variance offers a mechanism to quantify the uncertainty of replacing simulator runs with emulator evaluations at untested inputs. Detailed background on GP emulators can be found in Rasmussen and Williams (2006) and Santner, Williams, and Notz (2018), with the latter offering additional insights on using GPs to model physics-based simulators.

Consider a p-dimensional vector of inputs $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$ lying in a domain $\chi \subseteq \mathbb{R}^p$ that represents the setting for a single simulator run with scalar output, $f(\mathbf{x})$. Simulator output is assumed to be realizations of a stochastic process. The i^{th} simulator run is then $Y_i = \mathbf{h}^{\mathrm{T}}(\mathbf{x}_i)\boldsymbol{\beta} + Z(\mathbf{x}_i)$, where $\mathbf{h}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\beta}$ is a

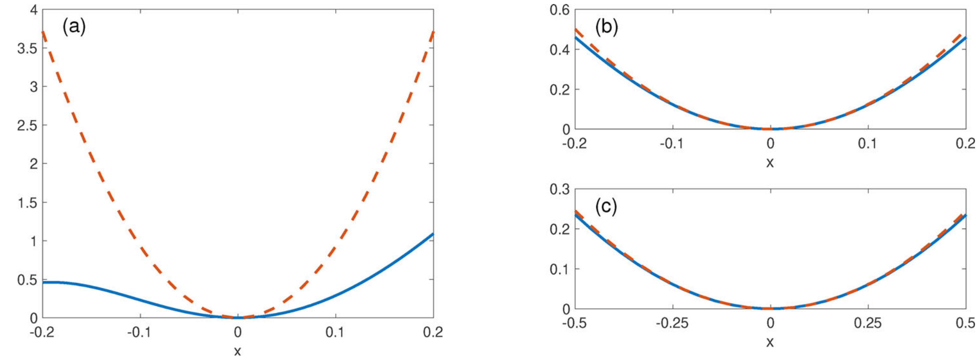

Figure 1. The nonlinear portion of the example functions (solid line) and their respective local error bounds (dashed line) for a) $\eta(x)$, b) $f(x)$, and c) g(x).

trend function and $Z(\mathbf{x})$ is a realization of a zero-mean Gaussian process with covariance $\sigma^2 c(\cdot,\cdot;\boldsymbol{\gamma})$. (Note, in Section 4 we use a power exponential correlation function, but the methodology we develop in Section 3 also holds for some Matérn correlation functions as well.) Further, $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}),\dots,h_q(\mathbf{x}))^{\mathrm{T}}$ is a $q-$dimensional vector of basis functions, and $\boldsymbol{\beta} = (\beta_1,\dots,\beta_q)^{\mathrm{T}}$ are the regression parameters for the trend function.

Consider a design matrix, $\mathbf{x}^{\mathrm{D}}$, of m sets of inputs for m computer model runs (or briefly, $\mathbf{x}^{\mathrm{D}} = [\mathbf{x}_1^{\mathrm{D}},\dots,\mathbf{x}_m^{\mathrm{D}}]^{\mathrm{T}}$). The corresponding simulator outputs are $y_i^{\mathrm{D}} = f(\mathbf{x}_i^{\mathrm{D}})$, and the collected response from exercising the model at each of the m design points is $\mathbf{y}^{\mathrm{D}} = (y_1^{\mathrm{D}},\dots,y_m^{\mathrm{D}})^{\mathrm{T}}$. Restricting the GP model to only GPs that are conditioned to match the design-response pairs, the best linear unbiased predictive mean and standard squared error for the GP emulator evaluated at an untested input, $\mathbf{x}^*$ are respectively given by

$$\mu(\mathbf{x}^*) = \mathbf{h}^{\mathrm{T}}(\mathbf{x}^*)\boldsymbol{\beta} + \mathbf{r}^{\mathrm{T}}(\mathbf{x}^*)R^{-1}(\mathbf{y}^{\mathrm{D}} - H_{\mathbf{x}}^{\mathrm{D}}\boldsymbol{\beta}), \quad \text{and} \tag{2}$$

$$\begin{aligned} \mathrm{v}(\mathbf{x}^*) = \sigma^2 \Big(1 - \mathbf{r}^{\mathrm{T}}(\mathbf{x}^*)R^{-1}\mathbf{r}(\mathbf{x}^*) + \big(\mathbf{h}(\mathbf{x}^*) \\ - (H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}\mathbf{r}(\mathbf{x}^*)\big)^{\mathrm{T}} \\ \times \big((H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}H_{\mathbf{x}}^{\mathrm{D}}\big)^{-1}\big(\mathbf{h}(\mathbf{x}^*) - (H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}\mathbf{r}(\mathbf{x}^*)\big)\Big) \end{aligned} \tag{3}$$

(Sacks, Schiller, and Welch 1989; Welch et al. 1992). Here $H_{\mathbf{x}}^{\mathrm{D}} = [\mathbf{h}(\mathbf{z}_1^{\mathrm{D}}),\dots,\mathbf{h}(\mathbf{z}_m^{\mathrm{D}})]^{\mathrm{T}}$ is an $m \times q$ matrix of regressors evaluated at the design, σ^2 is the scaling of the variance of the process, and $\mathbf{r}(\mathbf{x}^*) = c(\mathbf{x}^*,\mathbf{x}^{\mathrm{D}};\boldsymbol{\gamma})$ is a vector of correlations between the untested and design inputs that depend on the range parameters, $\boldsymbol{\gamma}$. The correlation matrix, R, is an $m \times m$ matrix comparing pairs of design points in $\mathbf{x}^{\mathrm{D}}$. That is, $(R)_{i,j} = c(\mathbf{x}_i^{\mathrm{D}},\mathbf{x}_j^{\mathrm{D}};\boldsymbol{\gamma})$. Coefficients of the mean trend are estimated by $\hat{\boldsymbol{\beta}} := \Big((H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}H_{\mathbf{x}}^{\mathrm{D}}\Big)^{-1}(H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}\mathbf{y}^{\mathrm{D}}$.

The GP parameters σ^2 and $\boldsymbol{\gamma}$ are typically not known, and must be approximated or sampled. Replacing $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}$, the likelihood function for GP parameters is given by

$$\mathcal{L}(\sigma^2,\boldsymbol{\gamma}) = \frac{\exp\left\{-\frac{1}{2\sigma^2}\big(\mathbf{y}^{\mathrm{D}} - H_{\mathbf{x}}^{\mathrm{D}}\hat{\boldsymbol{\beta}}\big)^{\mathrm{T}}R^{-1}\big(\mathbf{y}^{\mathrm{D}} - H_{\mathbf{x}}^{\mathrm{D}}\hat{\boldsymbol{\beta}}\big)\right\}}{|R|^{\frac{1}{2}}\,|(H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}H_{\mathbf{x}}^{\mathrm{D}}|^{\frac{1}{2}}\,(2\pi\sigma^2)^{\frac{m-q}{2}}}. \tag{4}$$

To estimate σ^2 and $\boldsymbol{\gamma}$, one can follow the objective Bayesian approach introduced in Gu, Wang, and Berger (2018) which uses reference priors and further marginalizes the likelihood function over σ^2. One can then obtain maximum a posteriori (MAP) estimates of the range parameters via the `RobustGaSP` package (Gu, Palomo, and Berger 2019). One can also estimate σ^2 with $\hat{\sigma}^2 = \frac{1}{m-q}(\mathbf{y}^{\mathrm{D}} - H_{\mathbf{x}}^{\mathrm{D}}\hat{\boldsymbol{\beta}})^{\mathrm{T}}R^{-1}(\mathbf{y}^{\mathrm{D}} - H_{\mathbf{x}}^{\mathrm{D}}\hat{\boldsymbol{\beta}})$. We use this objective Bayesian approach for fitting all GP emulators in this article—both for the component GPs used in the PPLE derivation in Section 3 and for all other GPs we use for comparison in Section 4 except for the laGP.

2.2. Parallel Partial Emulator Background

For simulators that yield vector-valued outputs (e.g., spatial or space-time fields), we consider the parallel partial emulator (PPE) introduced by Gu and Berger (2016). The PPE assumes that each output component is independent and has its own mean trend and scalar variance, but that all output components share a common correlation structure. The common correlation structure is key to the PPE's relatively low computational cost despite its ability to accurately emulate potentially massive output fields.

To describe the PPE, consider a simulator with s-dimensional vector valued output. Let $y^j(\mathbf{x})$ $(j = 1,\dots,s)$, denote the simulator output at a single coordinate such that $\mathbf{y}(\mathbf{x}) = \big(y^1(\mathbf{x}),\dots,y^s(\mathbf{x})\big)$ is the entire vector-valued response from the simulator evaluated at a single input vector, $\mathbf{x}$. Further, one can concatenate the (length s row) vector valued outputs of the simulator responses of each design point in an $m \times s$ matrix, $Y^{\mathrm{D}} = \big[\mathbf{y}^{\mathrm{T}}(\mathbf{x}_1^{\mathrm{D}}),\dots,\mathbf{y}^{\mathrm{T}}(\mathbf{x}_m^{\mathrm{D}})\big]^{\mathrm{T}}$. The s-dimensional predictive GP mean conditioned on the design/response pairs is then given as the following row vector:

$$\boldsymbol{\mu}(\mathbf{x}^*) = \mathbf{h}^{\mathrm{T}}(\mathbf{x}^*)B_{\mathbf{x}} + \mathbf{r}^{\mathrm{T}}(\mathbf{x}^*)R^{-1}(Y^{\mathrm{D}} - H_{\mathbf{x}}^{\mathrm{D}}B_{\mathbf{x}}), \tag{5}$$

where $H_{\mathbf{x}}^{\mathrm{D}}$ is the $m \times q$ matrix of regressors evaluated at the design and $\mathbf{h}(\mathbf{x}^*)$ is a $q \times 1$ vector of regressors at the untested input, $\mathbf{x}^*$. Further, $B_{\mathbf{x}}$ is a $q \times s$ matrix where each column is the q-dimensional vector of regression coefficients, $\boldsymbol{\beta}_j$ $(j = 1,\dots,s)$, associated with each output component. The predictive variance for each output component is given by

$$\begin{aligned} \mathbf{v}_j(\mathbf{x}^*) = \sigma_j^2\Big(1 - \mathbf{r}^{\mathrm{T}}(\mathbf{x}^*)R^{-1}\mathbf{r}(\mathbf{x}^*) + \big(\mathbf{h}(\mathbf{x}^*) - (H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}r(\mathbf{x}^*)\big)^{\mathrm{T}} \\ \times \big((H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}H_{\mathbf{x}}^{\mathrm{D}}\big)^{-1}\big(\mathbf{h}(\mathbf{x}^*) - (H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}\mathbf{r}(\mathbf{x}^*)\big)\Big), \end{aligned} \tag{6}$$

$j = 1,\dots,s$. Taking $\mathbf{y}_j^{\mathrm{D}}$ to be the jth column of Y^{D}, the output component-wise scalar variance is estimated by $\hat{\sigma}_j^2 = \frac{1}{m-q}(\mathbf{y}_j^{\mathrm{D}} - H_{\mathbf{x}}^{\mathrm{D}}\boldsymbol{\beta}_j)^{\mathrm{T}}R^{-1}(\mathbf{y}_j^{\mathrm{D}} - H_{\mathbf{x}}^{\mathrm{D}}\boldsymbol{\beta}_j)$, and the output component-wise regression coefficients are estimated by $\hat{\boldsymbol{\beta}}_j = \Big((H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}H_{\mathbf{x}}^{\mathrm{D}}\Big)^{-1}(H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}\mathbf{y}_j^{\mathrm{D}}$. We use the `RobustGaSP` package to estimate range parameters for the PPE. In the case of vector-valued or spatial outputs, `RobustGaSP` uses a composite likelihood strategy (Varin, Reid, and Firth 2011) within the same objective Bayesian framework as it does for scalar outputs.

In the above expressions one only needs to calculate two relatively small matrix inverses (even in the case of massive output dimension, for example, $s \gg m, q$), namely of the $m \times m$ matrix R and the $q \times q$ matrix $(H_{\mathbf{x}}^{\mathrm{D}})^{\mathrm{T}}R^{-1}H_{\mathbf{x}}^{\mathrm{D}}$. Perhaps most important from a physical perspective, the PPE mean is a linear combination of the simulator output and thus inherits the smoothness properties of the simulator, such as the spatial correlations, without modeling them directly (Gu and Berger 2016). On the other hand, approaches without shared correlation models such as constructing many independent emulators or local approximate GP regression (Gramacy and Apley 2015) will have different weights corresponding to different output coordinates and thus will not retain the smoothness properties of the simulator. In recent work, Gao and Pitman (2023) demonstrate that the PPE can also inherit conservation properties of the simulator which methods based on localized designs cannot. The PPE does not come without drawbacks. If some spatial output coordinates are responsive and vary with changing inputs while other coordinates are relatively flat, obtaining MAP estimates based on a composite likelihood is a challenging and perhaps futile task given the spatial nonstationarity. Further, due to the independence assumption, spatially correlated samples of the PPE are not available.

2.3. Linked Emulator Background

Consider a multiphysics simulator that results from the composition of two computer models, $\eta(\mathbf{x}, \mathbf{z}) = g(f(\mathbf{x}), \mathbf{z})$. Direct emulation of a composite model, that is with $\mathbf{x}$ and $\mathbf{z}$ as inputs to the emulator and assuming no knowledge of f, is referred to as *composite emulation*. In contrast, Kyzyurova, Berger, and Wolpert (2018) developed an approach that fits a GP emulator to g and a GP emulator to f, and then considers the resulting random variable that is the composition of the two individual GPs. That random variable does not have a closed form description, but for power exponential kernels, the first two moments are calculated in Kyzyurova, Berger, and Wolpert (2018). Those calculations are extended to other common kernels in Ming and Guillas (2021). The emulator using these first two moments is called a *linked GP emulator.*

In their work, Ming and Guillas (2021) assume for the "inside" function f that each dimension of the input space has its own mapping to the output space. (In the following discussion we largely follow their notation.) That is, for $\mathcal{X}_k, \mathcal{Y}_k \subseteq \mathbb{R}$ ($k = 1, \dots, p$), they assume for $x \in \mathcal{X}_k$ that $f_k : \mathcal{X}_k \to \mathcal{Y}_k$ and $f = (f_1, \dots, f_p)^{\mathrm{T}}$. They then construct p emulators of f. At one untested input, $\mathbf{x}^*$, the conditional Normal pdf of each GP emulator, $k = 1, \dots, p$, is defined to be

$$\tilde{f}_k(\mathbf{x}^*) \mid \mathbf{x}^{\mathrm{D}}, \mathbf{y}^{\mathrm{D}}_{f_k} := \mathcal{N}\big(\mu_{f_k}(\mathbf{x}^*), \mathrm{v}_{f_k}(\mathbf{x}^*)\big), \tag{7}$$

where $\mathbf{y}^{\mathrm{D}}_{f_k}$ represents the kth element of the output training data for the vector-valued function f (or the reduced-dimensional representation of the output training data), and where μ_{f_k} and v_{f_k} are the predictive mean and point-wise variance conditioned on the training dataset $(\mathbf{x}^{\mathrm{D}}, \mathbf{y}^{\mathrm{D}}_{f_k})$ as given in (2) and (3).

Assume that the "outside" function g potentially takes as input both the output from $\mathbf{w} = f(\mathbf{x})$ and additional external inputs. That is, for $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^p$ and $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^l$, $g : \mathcal{W}, \mathcal{Z} \to \mathbb{R}$. One can now construct a Gaussian process emulator for the "outside" function $g(\mathbf{w}, \mathbf{z})$. With m design runs, the training data for composite inputs has the form $\mathbf{w}^{\mathrm{D}} = [\mathbf{w}^{\mathrm{D}}_1, \dots, \mathbf{w}^{\mathrm{D}}_m]$, where $\mathbf{w}^{\mathrm{D}}_j = f(\mathbf{x}^{\mathrm{D}}_j)$. Likewise, the training data for the direct input to g is denoted by $\mathbf{z}^{\mathrm{D}} = [\mathbf{z}^{\mathrm{D}}_1, \dots, \mathbf{z}^{\mathrm{D}}_m]$. The $m \times 1$ response vector to that training data is $\mathbf{y}^{\mathrm{D}}_g = g(\mathbf{w}^{\mathrm{D}}, \mathbf{z}^{\mathrm{D}})$. At one untested input, $(\mathbf{w}^*, \mathbf{z}^*)$, the point-wise conditional Normal pdf of the GP emulator for g is

$$\tilde{g}(\mathbf{w}^*, \mathbf{z}^*) \mid \mathbf{w}^{\mathrm{D}}, \mathbf{z}^{\mathrm{D}}, \mathbf{y}^{\mathrm{D}}_g := \mathcal{N}\Big(\mu_g(\mathbf{w}^*, \mathbf{z}^*), \mathrm{v}_g(\mathbf{w}^*, \mathbf{z}^*)\Big). \tag{8}$$

Here μ_g and v_g are the predictive mean and point-wise variance conditioned on the training dataset $(\mathbf{w}^{\mathrm{D}}, \mathbf{z}^{\mathrm{D}}, \mathbf{y}^{\mathrm{D}}_g)$ as given in (2) and (3), respectively, using the shorthand $\mu_g(\mathbf{w}^*, \mathbf{z}^*) \equiv \mu_g([\mathbf{w}^{*\mathrm{T}}, (\mathbf{z}^*)^{\mathrm{T}}]^{\mathrm{T}})$ (and likewise for v_g). The trend function of the mean of the GP for g is assumed to be the form $t(\mathbf{w}, \mathbf{z}; \boldsymbol{\theta}, \boldsymbol{\beta}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\theta} + \mathbf{h}^{\mathrm{T}}(\mathbf{z})\boldsymbol{\beta}$, where $\mathbf{h}(\mathbf{z}) = \big(h_1(\mathbf{z}), \dots, h_q(\mathbf{z})\big)^{\mathrm{T}}$ is a vector of regression basis functions, and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)^{\mathrm{T}}$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_q)^{\mathrm{T}}$ are the regression parameters associated with $\mathbf{w}$ and $\mathbf{z}$, respectively.

The key assumption for linked emulation is that the output from the GP emulator for f_k is a random variable. Specifically, $W_k(\mathbf{x}^*) \overset{ind}{\sim} \tilde{f}_k(\mathbf{x}^*) \mid \mathbf{x}^{\mathrm{D}}, \mathbf{y}^{\mathrm{D}}_{f_k}$.

To construct the linked GP emulator, consider the linked emulator $Y \sim p_\eta \mid \mathbf{x}, \mathbf{z}$, where

$$p_\eta(y \mid \mathbf{x}, \mathbf{z}) = \int_{\mathbf{w}} p_g(y \mid \mathbf{w}, \mathbf{z}) p_f(\mathbf{w} \mid \mathbf{x}) d\mathbf{w}, \tag{9}$$

and where $p_g = \tilde{g} \mid \mathbf{w}^{\mathrm{D}}, \mathbf{z}^{\mathrm{D}}, \mathbf{y}^{\mathrm{D}}_g$ and $p_{\mathbf{f}} = \tilde{\mathbf{f}} \mid \mathbf{x}^{\mathrm{D}}, \mathbf{y}^{\mathrm{D}}_f$ (Kyzyurova, Berger, and Wolpert 2018; Ming and Guillas 2021). The predictive mean of the linked GP emulator at an untested input $(\mathbf{x}^*, \mathbf{z}^*)$ is given by

$$\mu_\eta(\mathbf{x}^*, \mathbf{z}^*) = \boldsymbol{\mu}_f^{\mathrm{T}}\boldsymbol{\theta} + \mathbf{h}^{\mathrm{T}}(\mathbf{z}^*)\boldsymbol{\beta} + I^{\mathrm{T}}A, \tag{10}$$

where each component of $\boldsymbol{\mu}_f$, $\mu_{f_k}(\mathbf{x}^*)$, is the predictive mean of $\tilde{f}_k$ ($k = 1, \dots, p$) evaluated at $\mathbf{x}^*$. The ith component of the vector of correlations between untested and design points is given by

$$I_i = \Big(\prod_{j=1}^{p} \xi_{ij}\Big)\Big(\prod_{j=1}^{l} c(z^*_j, z^{\mathrm{D}}_{ij}; \nu_j)\Big), \tag{11}$$

where $\xi_{ij} = \mathsf{E}\Big[c(W_j, w^{\mathrm{D}}_{ij}; \gamma_j)\Big]$, and $\boldsymbol{\nu}$ and $\boldsymbol{\gamma}$ are range parameters corresponding to $\mathbf{z}$ and $\mathbf{w}$, respectively. Here $A = R_g^{-1}\Big(\mathbf{y}^{\mathrm{D}}_g - \mathbf{w}^{\mathrm{D}}\boldsymbol{\theta} - H^{\mathrm{D}}_{\mathbf{z}}\boldsymbol{\beta}\Big)$, so $I^{\mathrm{T}}A$ is analogous to the last term in (2). Details of analytic expressions for ξ_{ij} can be found in (Ming and Guillas 2021). The variance of the linked emulator at an untested input is $\mathrm{v}_\eta(\mathbf{x}^*, \mathbf{z}^*) = V_1(\mathbf{x}^*, \mathbf{z}^*) + V_2(\mathbf{x}^*, \mathbf{z}^*)$. Using the shorthand $\mathbf{r}(\mathbf{W}, \mathbf{z}^*) \equiv \mathbf{r}([\mathbf{W}^{\mathrm{T}}, (\mathbf{z}^*)^{\mathrm{T}}]^{\mathrm{T}})$, V_1 and V_2 are given by

$$\begin{aligned} V_1(\mathbf{x}^*, \mathbf{z}^*) =& \mathsf{E}\big[\mathbf{W}^{\mathrm{T}}\boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{W} + \mathbf{h}^{\mathrm{T}}(\mathbf{z}^*)\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}\mathbf{h}(\mathbf{z}^*) \\ &+\mathbf{r}^{\mathrm{T}}(\mathbf{W}, \mathbf{z}^*)AA^{\mathrm{T}}\mathbf{r}(\mathbf{W}, \mathbf{z}^*)\big] \\ &+ 2\mathsf{E}\big[\boldsymbol{\theta}^{\mathrm{T}}\mathbf{W}\mathbf{h}^{\mathrm{T}}(\mathbf{z}^*)\boldsymbol{\beta} + \boldsymbol{\theta}^{\mathrm{T}}\mathbf{W}\mathbf{r}^{\mathrm{T}}(\mathbf{W}, \mathbf{z}^*)A \\ &+\mathbf{h}^{\mathrm{T}}(\mathbf{z}^*)\boldsymbol{\beta}\mathbf{r}^{\mathrm{T}}(\mathbf{W}, \mathbf{z}^*)A\big] - \mu^2_\eta(\mathbf{x}^*, \mathbf{z}^*), \end{aligned} \tag{12}$$

and

$$\begin{aligned} V_2(\mathbf{x}^*, \mathbf{z}^*) = \sigma^2\mathsf{E}\Big[&1 - \mathbf{r}^{\mathrm{T}}(\mathbf{W}, \mathbf{z}^*)R_g^{-1}\mathbf{r}(\mathbf{W}, \mathbf{z}^*) \\ &+ \big([\mathbf{W}^{\mathrm{T}}, \mathbf{h}^{\mathrm{T}}(\mathbf{z}^*)]^{\mathrm{T}} - (H^{\mathrm{T}}R^{-1}\mathbf{r}(\mathbf{W}, \mathbf{z}^*)\big)^{\mathrm{T}} \\ &\times (H^{\mathrm{T}}R^{-1}H)^{-1}\big([\mathbf{W}^{\mathrm{T}}, \mathbf{h}^{\mathrm{T}}(\mathbf{z}^*)]^{\mathrm{T}} \\ &- H^{\mathrm{T}}R_g^{-1}\mathbf{r}(\mathbf{W}, \mathbf{z}^*)\big)\Big]. \end{aligned} \tag{13}$$

Analytic expressions for each of the expectations in V_1 and V_2 can be found in Ming and Guillas (2021).

3. Parallel Partial Linked Gaussian Process Emulation

Here we derive the new *parallel partial linked Gaussian process emulator (PPLE)* that combines the efficiency of the parallel partial emulator with the enhanced accuracy of linked emulation for constructing stochastic surrogates in the case of composite functions. We further show that the mean of the parallel partial linked GP emulator inherits the smoothness properties of the composite simulator. As such, the PPLE is a powerful advancement in methodology for faithfully emulating high-dimensional nonlinear multiphysics simulators.

We will still consider the case where the "outside" function $\mathbf{g}$ takes as input both output from an "inside" function, $\mathbf{f}$, and external inputs, but now the output of $\mathbf{g}$ is s-dimensional. We further consider a different case than Ming and Guillas (2021) for the inside function with input, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^p$, and output, $\mathbf{y}_f \in \mathcal{Y} \subseteq \mathbb{R}^n$, so that $\mathbf{f} : \mathcal{X} \to \mathcal{Y}$. We make the assumption that we can approximate f with d independent emulators, with $d \leq n$ and in some cases $d \ll n$. The choice is consistent with Kyzyurova, Berger, and Wolpert (2018) and supported by observations in chap. 3 of Kyzyurova (2017). Further, for coupled computer models with, say $n > 15$, this setup is compatible with employing a dimension-reduction technique to offer a low-dimensional representation of the vector-valued function f as input to $\mathbf{g}$ while constructing linked GP emulators. In summary, for the outside function we then have $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^d$ and $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^l$, $\mathbf{g} : \mathcal{W}, \mathcal{Z} \to \mathbb{R}^s$. The derivation of the PPLE requires carefully identifying terms in the predictive mean and variance of the linked emulator whose dimension changes when considering s-dimensional coupled simulator output instead of scalar coupled simulator output. Figure 2 offers a schematic of the linked parallel partial GP emulator.

We will now derive a parallel partial linked GP emulator, $\tilde{\boldsymbol{\eta}}(\mathbf{x}^*, \mathbf{z}^*) = \text{MVN}\big(\boldsymbol{\mu}_\eta, \text{diag}(\mathbf{v}_\eta)\big)$ for the vector-valued compound function $\boldsymbol{\eta}(\mathbf{x}, \mathbf{z}) = \mathbf{g}(\mathbf{f}(\mathbf{x}), \mathbf{z})$ given the design $(\{\mathbf{x}^{\text{D}}, \mathbf{z}^{\text{D}}\})$, the independent GP emulators $\tilde{f}_k = \mathcal{N}(\mu_{f_k}, \text{v}_{f_k})$ $(k = 1, \ldots, d)$, and the PPE for $\mathbf{g}$, $\tilde{\mathbf{g}} = \text{MVN}(\boldsymbol{\mu}_g, \text{diag}(\mathbf{v}_g))$. For the PPE of $\mathbf{g}$, the design $(\{\mathbf{w}^{\text{D}}, \mathbf{z}^{\text{D}}\})$ remains the same as for the scalar-output case, but now the response, Y_g^{D}, is an $m \times s$ matrix with the ith row given by $\mathbf{g}^{\text{T}}(\mathbf{w}_i^{\text{D}}, \mathbf{z}_i^{\text{D}})$. Recall that the basic premise of parallel partial emulation is that emulators of each component of vector-valued output share a common correlation structure, but each component has its own set of regressor coefficients and its own scalar variance. So fitting the PPE for $\mathbf{g}$ yields estimates for $\boldsymbol{\sigma}^2$ (an $s \times 1$ vector of scalar variances), $B_{\mathbf{z}}$ (a $q \times s$ matrix of regression parameters associated with $\mathbf{z}$), Θ (a $d \times s$ matrix of regression parameters associated with $\mathbf{W}$), and range parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\nu}$ associated with $\mathbf{W}$ and $\mathbf{z}$, respectively. Further, we let $\boldsymbol{\Theta} = [\Theta^{\text{T}}\ B_{\mathbf{z}}^{\text{T}}]^{\text{T}}$ (a $(d+q) \times s$ matrix), and $H = [\mathbf{w}^{\text{D}}\ H_{\mathbf{z}}^{\text{D}}]$ where $H_{\mathbf{z}}^{\text{D}}$ is the $m \times q$ matrix of regressors evaluated at the design. We estimate $\boldsymbol{\Theta}$ with $\hat{\boldsymbol{\Theta}} = \big(H^{\text{T}} R_g^{-1} H\big)^{-1} H^{\text{T}} R_g^{-1} Y_g^{\text{D}}$. For each of the $j = 1, \ldots, d$ scalar emulators, $\tilde{f}_j$ and the PPE emulator, $\tilde{\mathbf{g}}$, we utilize reference priors for the GP range parameters and obtain MAP estimates of the range parameters with the `RobustGaSP` package (Gu, Palomo, and Berger 2019).

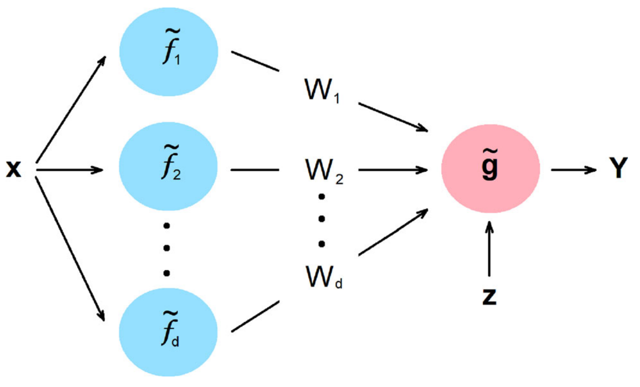

Figure 2. Schematic representation of the linked parallel partial GP emulator.

To construct the PPLE, we need to update the mean and variance of the linked GP emulator as given in (10), (12), and (13), respectively, to account for vector-valued outputs from $\mathbf{g}$. We find the mean by taking the iterated expectation,

$$\begin{aligned}\boldsymbol{\mu}_\eta(\mathbf{x}^*, \mathbf{z}^*) =& \mathsf{E}\left[\boldsymbol{\mu}_g(\mathbf{W}, \mathbf{z}^*)\right] \\ =& \mathsf{E}\big[\mathbf{W}^T\Theta + \mathbf{h}^{\text{T}}(\mathbf{z}^*)B_{\mathbf{z}} \\ &+ \mathbf{r}^{\text{T}}(\mathbf{W}, \mathbf{z}^*)R_g^{-1}\left(Y_g^{\text{D}} - H\boldsymbol{\Theta}\right)\big] \\ =& \boldsymbol{\mu}_f^{\text{T}}(\mathbf{x}^*)\Theta + \mathbf{h}^{\text{T}}(\mathbf{z}^*)B_{\mathbf{z}} + I^{\text{T}}\mathbf{A},\end{aligned} \tag{14}$$

where, due to the shared correlation structure, I is identical to the definition given in (11). Further, $\mathbf{A}$ is the $m \times s$ equivalent of the same term in (10). That is, $\mathbf{A} = R_g^{-1}(Y_g^{\text{D}} - H\boldsymbol{\Theta})$.

To compute the variance of the PPLE, consider the Law of Total variance with $\mathbf{v}_\eta(\mathbf{x}^*, \mathbf{z}^*) = \mathbf{V}_1(\mathbf{x}^*, \mathbf{z}^*) + \mathbf{V}_2(\mathbf{x}^*, \mathbf{z}^*)$, where

$$\mathbf{V}_1(\mathbf{x}^*, \mathbf{z}^*) = \text{var}\left[\mathbf{W}^{\text{T}}\Theta + \mathbf{h}^{\text{T}}(\mathbf{z}^*)B_{\mathbf{z}} + \mathbf{r}^{\text{T}}(\mathbf{W}, \mathbf{z}^*)\mathbf{A}\right], \text{ and} \tag{15}$$

$$\begin{aligned}\mathbf{V}_2(\mathbf{x}^*, \mathbf{z}^*) = \boldsymbol{\sigma}^2\mathsf{E}\Big[&1 - \mathbf{r}^{\text{T}}(\mathbf{W}, \mathbf{z}^*)R^{-1}\mathbf{r}(\mathbf{W}, \mathbf{z}^*) \\ &+ \big([\mathbf{W}^{\text{T}}, \mathbf{h}^{\text{T}}(\mathbf{z}^*)]^{\text{T}} - (H^{\text{T}}R_g^{-1}\mathbf{r}(\mathbf{W}, \mathbf{z}^*)\big)^{\text{T}} \\ &\times (H^{\text{T}}R_g^{-1}H\big)^{-1}\big([\mathbf{W}^{\text{T}}, \mathbf{h}^{\text{T}}(\mathbf{z}^*)]^{\text{T}} \\ &- H^{\text{T}}R_g^{-1}\mathbf{r}(\mathbf{W}, \mathbf{z}^*)\big)\Big].\end{aligned} \tag{16}$$

In $\mathbf{V}_2$, no terms inside the (scalar) expectation differ from V_2 in (13). The only update to this term is that each component of the PPE for $\mathbf{g}$ has its own value of scalar variance, and hence $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_s^2)^{\text{T}}$. Thus, we will focus on updates to $\mathbf{V}_1$. The jth component of $\mathbf{V}_1$, $(j = 1, \ldots, s)$ is given by

$$\left[\mathbf{V}_1(\mathbf{x}^*, \mathbf{z}^*)\right]_j = \text{var}\left[\mathbf{W}^{\text{T}}\Theta_j + \mathbf{h}^{\text{T}}(\mathbf{z}^*)\mathbf{b}_j + \mathbf{r}^{\text{T}}(\mathbf{W}, \mathbf{z}^*)\mathbf{A}_j\right], \tag{17}$$

where Θ_j, $\mathbf{b}_j$ and $\mathbf{A}_j$ are the jth columns of Θ, $B_{\mathbf{z}}$, and $\mathbf{A}$, respectively. Further, $\mathbf{W}, \Theta_j \in \mathbb{R}^d$, while $\mathbf{h}(\mathbf{z}^*), \mathbf{b}_j \in \mathbb{R}^q$, and $\mathbf{r}(\mathbf{W}, \mathbf{z}^*), \mathbf{A}_j \in \mathbb{R}^m$. The jth component of vector $\mathbf{V}_1$ is thus equivalent to (12) and thus given by

$$\begin{aligned}\left[\mathbf{V}_1(\mathbf{x}^*, \mathbf{z}^*)\right]_j =& \mathsf{E}\Big[\mathbf{W}^{\text{T}}\Theta_j\Theta_j^{\text{T}}\mathbf{W} + \mathbf{h}^{\text{T}}(\mathbf{z}^*)\mathbf{b}_j\mathbf{b}_j^{\text{T}}\mathbf{h}(\mathbf{z}^*) \\ &+ \mathbf{r}^{\text{T}}(\mathbf{W}, \mathbf{z}^*)\mathbf{A}_j\mathbf{A}_j^{\text{T}}\mathbf{r}(\mathbf{W}, \mathbf{z}^*)\Big] \\ &+ 2\mathsf{E}\Big[\Theta_j^{\text{T}}\mathbf{W}\mathbf{h}^{\text{T}}(\mathbf{z}^*)\mathbf{b}_j + \Theta_j^{\text{T}}\mathbf{W}\mathbf{r}^{\text{T}}(\mathbf{W}, \mathbf{z}^*)\mathbf{A}_j \\ &+ \mathbf{h}^{\text{T}}(\mathbf{z}^*)\mathbf{b}_j\mathbf{r}^{\text{T}}(\mathbf{W}, \mathbf{z}^*)\mathbf{A}_j\Big] - \left[\boldsymbol{\mu}_\eta^2(\mathbf{x}^*, \mathbf{z}^*)\right]_j^2.\end{aligned} \tag{18}$$

Closed form expressions of the expectations in $[\mathbf{V}_1(\mathbf{x}^*, \mathbf{z}^*)]_j$ are available in Ming and Guillas (2021), and hence we have analytic expressions to evaluate the parallel partial linked GP emulator, $\tilde{\boldsymbol{\eta}}(\mathbf{x}^*, \mathbf{z}^*) = \mathrm{MVN}\Big(\boldsymbol{\mu}_\eta(\mathbf{x}^*, \mathbf{z}^*), \mathrm{diag}\big(\mathbf{v}_\eta(\mathbf{x}^*, \mathbf{z}^*)\big)\Big)$.

We can rewrite the PPLE mean in (14) as $\boldsymbol{\mu}_\eta(\mathbf{x}^*, \mathbf{z}^*) = w^{\mathrm{T}}(\mathbf{x}^*, \mathbf{z}^*) Y_\eta^{\mathrm{D}}$, where $Y_\eta^{\mathrm{D}} = Y_g^{\mathrm{D}}$ and $w(\mathbf{w}^*, \mathbf{x}^*)$ is an $m \times 1$ vector of weights given by

$$w^{\mathrm{T}}(\mathbf{x}^*, \mathbf{z}^*) = [\boldsymbol{\mu}_f^{\mathrm{T}}(\mathbf{x}^*)\ h^{\mathrm{T}}(\mathbf{z}^*)]\big(H^{\mathrm{T}} R_g^{-1} H\big)^{-1} H^{\mathrm{T}} R_g^{-1} + I^{\mathrm{T}} R_g^{-1} - I^{\mathrm{T}} R_g^{-1} H \big(H^{\mathrm{T}} R_g^{-1} H\big)^{-1} H^{\mathrm{T}} R_g^{-1}. \tag{19}$$

The PPLE being the weighted sum of simulator output has several positive implications for emulating nonlinear coupled systems with high-dimensional output. In the case where dimension reduction is not needed for the inside function $\mathbf{f}$, (i.e., $d = n$) the PPLE mean interpolates the composite simulator $\boldsymbol{\eta}$ at the input design. When $d < n$ the PPLE mean is a near interpolator in that any dimension reduction used for $\mathbf{f}$ propagates through the I vector of correlations and linear trend via $\boldsymbol{\mu}_f$, but no approximations are made to the simulator outputs Y_η^{D}. As with the PPE, the PPLE is a linear combination of simulator responses and as such inherits the smoothness and conservation properties of the simulator (Gu and Berger 2016; Gao and Pitman 2023).

4. Numerical Experiments and Results

We illustrate the effectiveness of the parallel partial linked emulator methodology developed in Section 3 with two different sets of numerical experiments involving vector-valued composite functions. In the first set of numerical experiments we consider a combination of trigonometric functions designed to ensure the function is sufficiently nonlinear locally to test out the capabilities of the GP emulator. In the second set of experiments we investigate emulation for a simulator of coupled fluid flow and mechanical deformation of a column of mud saturated with water which compacts when a load is applied to the top of the column. This problem (classically referred to as the Terzaghi consolidation problem (Terzaghi and Peck 1948)) is numerically simulated using two-way loose coupling. This model serves as a prototype for more sophisticated physical simulators. Specifically, in a related study we are using emulation to investigate the effectiveness of hydraulic fracture stimulation. Fluid is injected at high pressures to break the low-permeability shale rock and create pathways for fluid to flow to production wells. Employing a physically realistic hydromechanical modeling code, the Complex Fracture Research Code developed by Mark McClure (see McClure and Horne 2011; McClure 2012), we consider one-dimensional fluid flow in fractures which are embedded in a 2D homogeneous medium undergoing mechanical deformation. Similar to the Terzaghi two-way loose coupling scheme described in this section, flow and deformation are loosely coupled (in this case iteratively coupled).

4.1. Pedagogical Example

We first compare parallel partial linked GP emulation vs parallel partial composite emulation (PPCE) for a “simulator” that is a simple analytic function designed to ensure the function is (a) vector-valued, (b) a composition, and (c) nonlinear enough to showcase the properties of a GP emulator. (Note, PPCE is a PPE fit directly to $\boldsymbol{\eta}$). The general form of this coupled pedagogical simulator is $\boldsymbol{\eta}(\mathbf{x}, \mathbf{z}) = \mathbf{g}\big(\mathbf{f}(\mathbf{x}), \mathbf{z}\big)$ where

$$\mathbf{f}(\mathbf{x}) = \langle \underbrace{\sin(c_1 x_1) + c_2 x_2^2}_{w_1}, \underbrace{\sin(c_3 x_1 \cos(c_4 \pi x_2))}_{w_2} \rangle \tag{20}$$

$$\mathbf{g}(\mathbf{z}, \mathbf{w}) = \langle \underbrace{\cos(c_5 z_2)\sin(c_6 w_1) + \sin(c_7 z_1)\cos(c_8 w_2)}_{g_1}, \underbrace{\cos(c_9 z_2)\sin(c_{10} w_1) + \sin(c_{11} z_1)\cos(c_{12} w_2)}_{g_2} \rangle, \tag{21}$$

with nominal values for $\mathbf{c}$ chosen as $[c_1, \ldots, c_{12}] = [5, 1, 3, 1, 3, 3, 3, 3, 4, 2, 4, 2]$. The functions $\mathbf{f}$, $\mathbf{g}$ and the constants $\mathbf{c}$ were chosen to ensure that the test function exhibits nonlinear behavior. Although $\mathbf{f}$ and $\mathbf{g}$ are periodic, this feature is not necessary to demonstrate the capabilities of the emulator. The design for training the emulators is generated from a Latin Hypercube sampler to ensure the design points are space filling (McKay, Beckman, and Conover 1979; Sacks, Schiller, and Welch 1989; Welch et al. 1992). For this experiment our independent variables are taken from a cube, $(\mathbf{x}, \mathbf{z}) \in [-1, 1] \times [-1, 1]$, and we train both the linked and composite GP emulators using the same 75 design points and test their predictions with 100 out-of-sample test points sampled from the same cube.

In Figure 3, we illustrate the performance of the emulators for this pedagogical problem. The top row in Figure 3 shows the results produced by the PPLE for the two components of $\boldsymbol{\eta}$. The bottom row shows the results produced by the composite emulator for these same two components. We see that for component η_2 (and to a lesser extent for η_1) the prediction errors (distance between the red squares and black dots) are smaller for the PPLE than for the composite emulator. The credible intervals for both components of the PPLE model are also smaller (on average) than the credible intervals for the respective components of the composite emulator, particularly for η_2.

To further test the PPLE we generate a collection of 100 different pairs of functions $\mathbf{f}$ and $\mathbf{g}$ by sampling $\mathbf{c}$ uniformly with values taken from intervals that are ±20% around the nominal values. In Figure 4, we plot the normalized histograms of prediction errors (the difference between the true model and the mean value predicted by the GP) for the 10^4 GP model predictions (100 untested inputs for each of the 100 different composite models). It is evident from this figure that the errors from the parallel partial linked emulator (dashed line shaded brown) are generally smaller than those from the composite emulator (solid line shaded in blue).

4.2. Multiphysics Simulator: Loosely Coupled Fluid Flow and Mechanical Deformation

In this section we provide a second example of coupled emulation, this time for a multiphysics simulator consisting of two-way loose coupling of fluid flow and mechanical deformation (Minkoff et al. 2003, 2004). For this set of coupled physics we consider the Terzaghi consolidation problem (Terzaghi and Peck 1948) which models a column of mud saturated with a single-phase fluid (water). An instantaneous load is dropped on the top

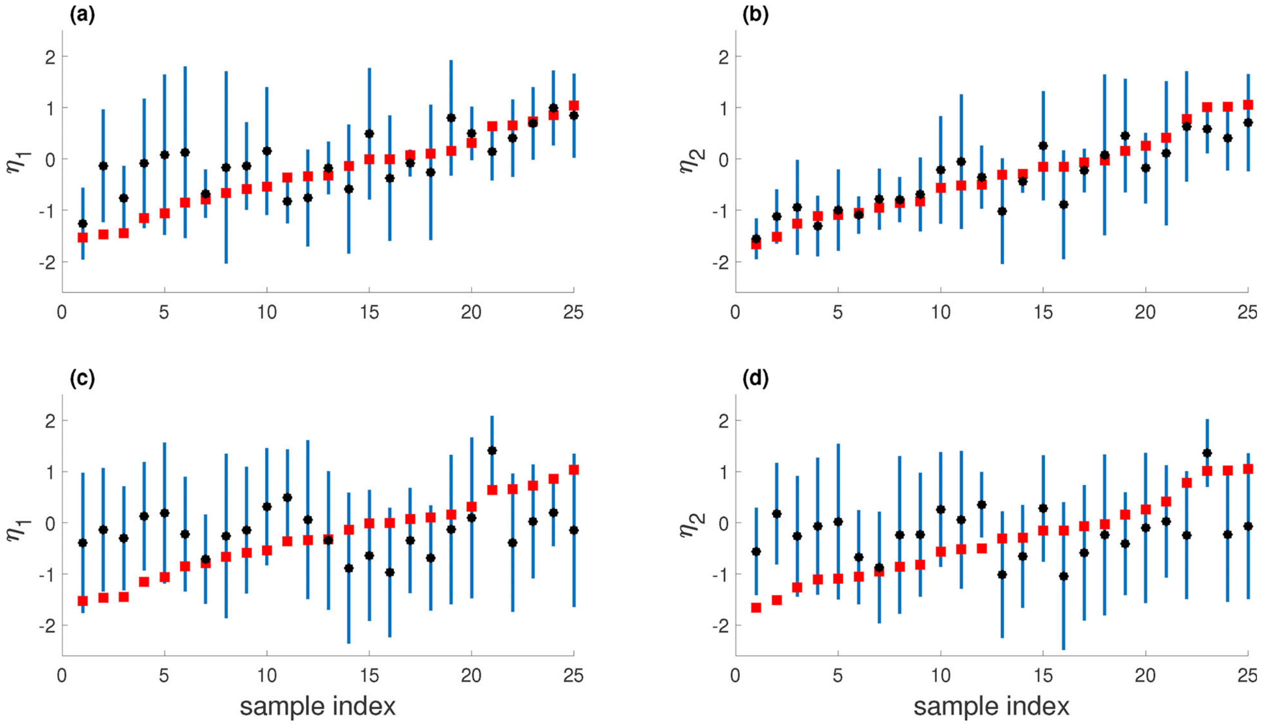

Figure 3. Emulation results for the pedagogical example given by equations 20 and 21 at the nominal value of **c**. Red squares are the true function values. Black dots are the predicted function values given by GP mean evaluations, and the vertical bars are 95% credible intervals. The four plots contain the following: (a) η_1 with PPLE; (b) η_2 with PPLE; (c) η_1 with PPCE; (d) η_2 with PPCE. Note that the indices are sorted in increasing order by the true output function value, so indices correspond for the columns of the figure grid, but not across the rows.

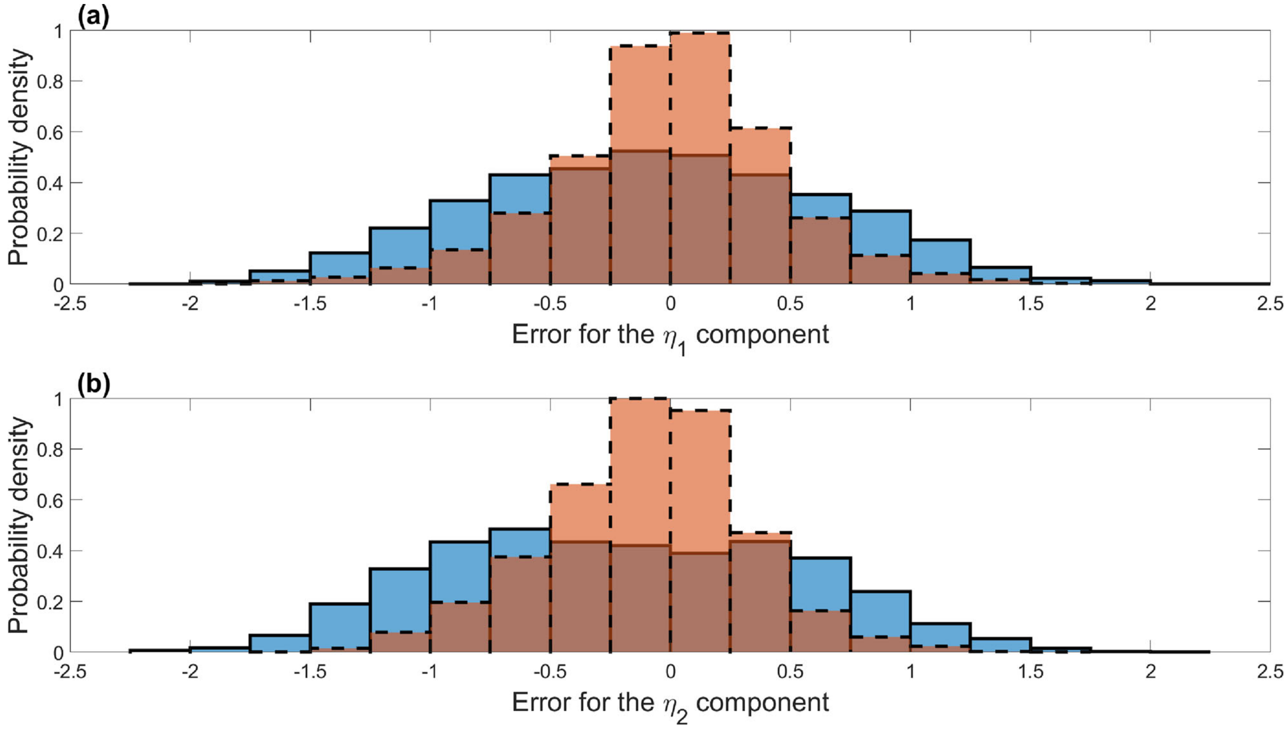

Figure 4. Results from the pedagogical example using 100 different sets of values for **c**. (a) Histogram of the errors for the predictions of η_1 for the PPLE (dashed line) and PPCE (solid line); (b) Histogram of the errors for the predictions of η_2 for the PPLE (dashed line) and PPCE (solid line).

of the column at the start of simulation. The two vertical edges and bottom of the column are assumed impermeable, but when the mud compacts, water is able to escape out of the top of the column. The fluid flow equation is derived from conservation of mass and Darcy's law (see Aziz 1979) where we solve for fluid pressure $p = p(x, t)$ (our quantity of interest). The flow equation is given by

$$\rho_0 \frac{\partial \phi p}{\partial t} = \nabla \left(\frac{k}{\mu c} \nabla p \right) + q. \tag{22}$$

Here ρ_0 is initial fluid density, ϕ is porosity, k is permeability, μ is fluid viscosity, c is fluid compressibility, x is depth, t is time, and q is a source or sink. The quasi-static linear elastic deformation equation (Middleton and Wilcock 1994; Minkoff and Kridler 2006) is given by

$$-(\lambda+\hat{\mu})\frac{d^2u}{dx^2} = f_{el} - \frac{dp}{dx}, \tag{23}$$

where we solve for displacement $u = u(x)$ (our quantity of interest); λ and $\hat{\mu}$ are the Lamé constants, and f_{el} is the external load. For our implementation, both the fluid flow and mechanical deformation equations are solved using the finite element method.

As mentioned in the Introduction, in this work our numerical simulator involves a two-way loose-coupling scheme in which we solve these two equations independently, first for flow and then for deformation over a fixed time period. This two-way loose coupling has been shown to mimic the fully-coupled physics without requiring multiple simplifying assumptions on the model (Minkoff et al. 2003, 2004; Minkoff and Kridler 2006). To allow the flow parameters to be realistically updated as compaction occurs, we periodically pass control from fluid flow to mechanical deformation even though the deformation equation is not time dependent. Specifically, we solve the flow equation for a fixed simulation time period with several smaller flow time steps taken during that period. At the end of this simulation time period, control is passed to the mechanical deformation code, and in particular, we send the change in pressure from flow to mechanics where it becomes a load for deformation (see the right hand side of (23)). We then solve the deformation equation for displacement, and using the fact that strain, ϵ, is the derivative of displacement, we update the porosity of the column as follows:

$$\phi(x,t) = 1 - \frac{1-\phi_0}{e^{\epsilon}}. \tag{24}$$

This updated fluid flow parameter, the porosity field, is sent back to the flow equation which is then solved for a subsequent set of time steps. Therefore, porosity is both a vector-valued input (function of space) for flow and a vector-valued output from deformation. We note that all material parameters for both fluid flow and mechanical deformation are functions of space. Any parameter which is updated during the loose coupling algorithm becomes, by definition, also a function of time.

Because we assume the column of mud contains a single fluid (water), we fix fluid density (ρ_0), fluid viscosity (μ), and fluid compressibility (c) at a single value, focusing instead on three parameters of interest for the two equations ((22) and (23)). Sensitivity analysis of this coupled system indicates that in fact the influential parameters for fluid flow are porosity, ϕ, and permeability, k (see Lee, Spiller, and Minkoff 2019). However, as mentioned above, the porosity field is both an input and an output parameter, and as such is considered a quantity of interest itself. Hence, for flow we only vary permeability as a direct input parameter. For mechanical deformation the sensitivity analysis indicates that both the mechanics and flow parameters are influential, namely, permeability, Young's modulus, and Poisson's ratio $[k, E, \nu]$. (Here that we have made use of the equivalence between the Lamé parameters (which show up in the deformation equation (23)) and Young's modulus and Poisson's ratio which we choose to vary as these values can be found in the literature for the Terzaghi problem.) Nominal values for these three influential parameters as well as the range over which they are varied are given in Table 1. (All inputs are normalized to a unit hypercube before training the emulators. Because permeability can vary over several orders of magnitude in a particular geologic region, we map the logarithm of k to the unit cube.)

Table 1. The nominal, maximum, and minimum values of the input parameters.

Variable	Name	Nominal values	Maximum values	Minimum values
k (in/s)	Permeability	$1.86 \cdot 10^{-11}$	$1.86 \cdot 10^{-10}$	$1.86 \cdot 10^{-12}$
ν	Poisson's ratio	0.3	0.33	0.27
E (lb/in^2)	Young's modulus	10^8	$1.1 \cdot 10^8$	$0.9 \cdot 10^8$
μ (psi · s)	Viscosity	$5.6 \cdot 10^{-5}$	–	–
c (psi)	Fluid compressibility	$1.2 \cdot 10^{-8}$	–	–
ϕ_0	Initial porosity	0.6	–	–
ρ_0 (lb/in^3)	Density	0.37275	–	–
f_{el} (psi/in)	Load	10^8	–	–

In contrast to the two-way loosely coupled simulator, at a single instance in time the composite function we consider for emulation is unidirectional, with mechanics serving as the inside function, **f**, and flow being the outside function, **g**. (In our implementation we discretize the column using 100 spatial grid points.) In summary, flow (**g**), takes in both porosity $\mathbf{w} = \mathbf{f}(\mathbf{x})$, and permeability $\mathbf{z}$, yielding pressure $\boldsymbol{\eta}$. We consider two experiments below: one where the whole column has a single value of permeability (constant permeability), and a second where the column has one value of permeability in the top half of the column and a different permeability in the bottom half (varying permeability).

In addition to applying the vector-valued global GP emulator methods—PPCE and PPLE—to this example, we will also consider methods that fit many independent scalar-output component emulators (or SC), specifically one for each depth. Under each of these two general classes of methods we present results for several sub-methods for comparison. While both PPCE and PPLE emulate a vector-valued function, the PPCE predicts the final output of the composite function $\boldsymbol{\eta}$ directly. In contrast, the PPLE links together separate emulators for **f** and **g**. For the PPLE methodology, when input to the outer function **g** includes high-dimensional output from **f**, dimension reduction is required to construct the emulator for **f**. Here we present results from two dimension reduction techniques for **f**: principal component analysis (PCA) and gradient-based kernel dimension reduction (gKDR) (Fukumizu and Leng 2014). We refer to these two sub-methods as PPLE PCA and PPLE gKDR. With PCA we include enough PCA modes to capture 90% of the variance observed in the training data for porosity (Gu and Shen 2020). In contrast to PCA, gKDR identifies low-dimensional structures between input and output data, leveraging the covariance structure in reproducing kernel Hilbert spaces (Fukumizu and Leng 2014). We note other dimension reduction techniques could also be employed in the PPLE for **f** including ICA and/or non-Euclidean PCA (see Anowar, Sadaoui, and Selim 2021). Note, with PPLE, we do not use dimension reduction on **g** during emulation, only on the output of the linking function **f**.

Conversely for linked GP emulators, if we treat our QoI pressure as a scalar at each depth, dimension reduction is not necessary. However, a notable drawback of using scalar emulators for vector-valued output is that we must construct many independent emulators. We consider four scalar-output sub-methods to compare to our PPLE methodology: scalar-output component linked GP emulators (LESC), scalar-output component composite emulators (CESC), linked GP emulators using a local approximate Gaussian process (LE laGP), and composite emulators using a local approximate Gaussian process (CE laGP). In LESC, both $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$ are a collection of many scalar-output valued emulators and the linked GP emulator framework is used to construct each component of $\tilde{\boldsymbol{\eta}}$ independently (Ming and Guillas 2021). In CESC, we directly predict pressure using many independent scalar emulators of pressure, one at each depth in the column of mud (Gu and Berger 2016). For both the CESC and LESC, we again utilize the objective Bayesian approach with reference priors on the range parameters and obtain MAP estimates of range parameters (for each scalar-output emulator constructed) via the `RobustGaSP` package (Gu, Palomo, and Berger 2019). The laGP effectively considers depth as an input which expands the size of the training data from m to $m \times s$ (m inputs for each depth). It then uses an optimized local subdesign of training data to build a GP emulator at each depth (Gramacy and Apley 2015; Gramacy 2016). Within the scheme to obtain a local subdesign, the laGP uses MLEs to obtain a set of range parameters specific to each depth. For CE laGP we treat pressure as the output, while for LE laGP we treat porosity as the output and then use the subdesign chosen by the laGP to construct an emulator for pressure at each depth. (Note, in our implementation of the laGP methods with m training simulations available, we used sub designs of size $1.5m$. Results with sub designs sized $2m$ and $1.5m$ were effectively equivalent and both were much better than results from sub designs sized m.) Attributes of the seven methods whose results we compare are shown in Table 2.

The efficacy of each emulator is compared using average root mean squared error (RMSE), average length of the 95% credible interval (L_{CI}), and the coverage, or percent of true values that fall within the 95% credible interval. The RMSE and L_{CI} are given by

$$\text{RMSE} = \sqrt{\frac{1}{sN}\sum_{j=1}^{s}\sum_{i=1}^{N}\left(\boldsymbol{\eta}_j(\mathbf{x}_i^*, \mathbf{z}_i^*) - \boldsymbol{\mu}_j(\mathbf{x}_i^*, \mathbf{z}_i^*)\right)^2}, \quad \text{and} \quad (25)$$

$$L_{CI} = \frac{1}{sN}\sum_{j=1}^{s}\sum_{i=1}^{N} CI_{ij}, \quad (26)$$

where N is the number of testing inputs, $\boldsymbol{\mu}_j(\mathbf{x}_i^*, \mathbf{z}_i^*)$ is the predictive emulator mean evaluation for the ith testing input, and $\boldsymbol{\eta}_j(\mathbf{x}_i^*, \mathbf{z}_i^*)$ is the evaluation of the composite simulator for the ith testing input at the jth spatial coordinate. CI_{ij} is the 95% posterior credible interval at each depth for each GP model under consideration. Note, when results are reported spatially the averages over s are omitted in (25) and (26). We also report on the efficiency of each emulator by reporting the *timing* which we define to be the time to train each emulator plus the time to evaluate each emulator once over the whole column.

Table 2. Comparison of emulation strategies for the Terzaghi consolidation multi-physics simulator.

	PPLE PCA	PPLE gKDR	PPCE	LESC	CESC	LE laGP	CE laGP
Vector output?	Y	Y	Y	N	N	N	N
Linked? If Y, dim. red. for $\tilde{f}$?	Y/Y	Y/Y	N	Y/N	N	Y/N	N

In our experiments we ran the simulator 200 times, each time varying the three influential input parameters. (We used half the simulations for training, $m = 100$, and the other half for out-of-sample testing, $N = 100$.) In particular, we varied Poisson's ratio, ν, and Young's modulus, E, within ±10% of their nominal values using a Latin hypercube sampler (see Table 1). As noted earlier, we varied permeability over a log rather than a linear scale. For linked GP emulators, we do not attempt to ensure the porosity which is the input to the outer function is space filling. However, an adaptive strategy could be used to improve on this design as is done in Ming and Guillas (2021).

Constant permeability experiment. Figure 5 shows the simulated pressure fields along with the RMSE and L_{CI} all plotted over the column for the both of the PPLEs and the PPCE. All of these emulators do an excellent job of predicting the simulated pressure curves used for testing, but the PPLE methods outperform the PPCE. (Note the magnitude of the errors is on the order of one tenth of one percent of the size of the data as shown in Figure 5(a)–(c).) At the top of the column, the average credible intervals are a bit larger for the PPLE methods as compare to PPCE, but all L_{CI}s are small compared to the magnitude of the data. Figure 6 shows the same content as Figure 5, but these results come from applying the scalar-output independent emulator methods. We see that the independent scalar-output emulators do worse than the PPE-based emulators. (Note that the error and credible interval scales are 1–2 orders of magnitude larger in Figures 6(b)&(c) than in Figure 5(b) and (c), respectively.) Amongst scalar-output component emulators, the linked GP models (LESC and LE laGP) significantly outperform the composite GP models (CESC and CE laGP) both in terms of RMSE and L_{CI}. However, the linked GP models are noisier in their predictive efficacy along the column. We further summarize these results in Table 3 by reporting values of RMSE and L_{CI} averaged over the column depth. We also report the average coverage of each method which is reasonable for each of the seven methods explored. Finally, we report the timing for each method. Amongst all methods, PPLE PCA is the fastest. For the PPLE methods, the difference in timings is entirely driven by computational intensity of applying dimension reduction to $\mathbf{f}$. Dimension reduction method gKDR is relatively expensive as compared to PCA. It is further worth noting that all four of the scalar-output component emulators could be sped up by performing the computations for each output component in parallel. For the laGP methods, each emulator evaluation amounts to constructing a new local emulator which would be an added computational burden if one were to use LE laGP and CE laGP in a forward or inverse UQ problem that requires many emulator evaluations across each output component.

We note that while it makes physical sense to look at a homogeneous column with a single value for permeability since

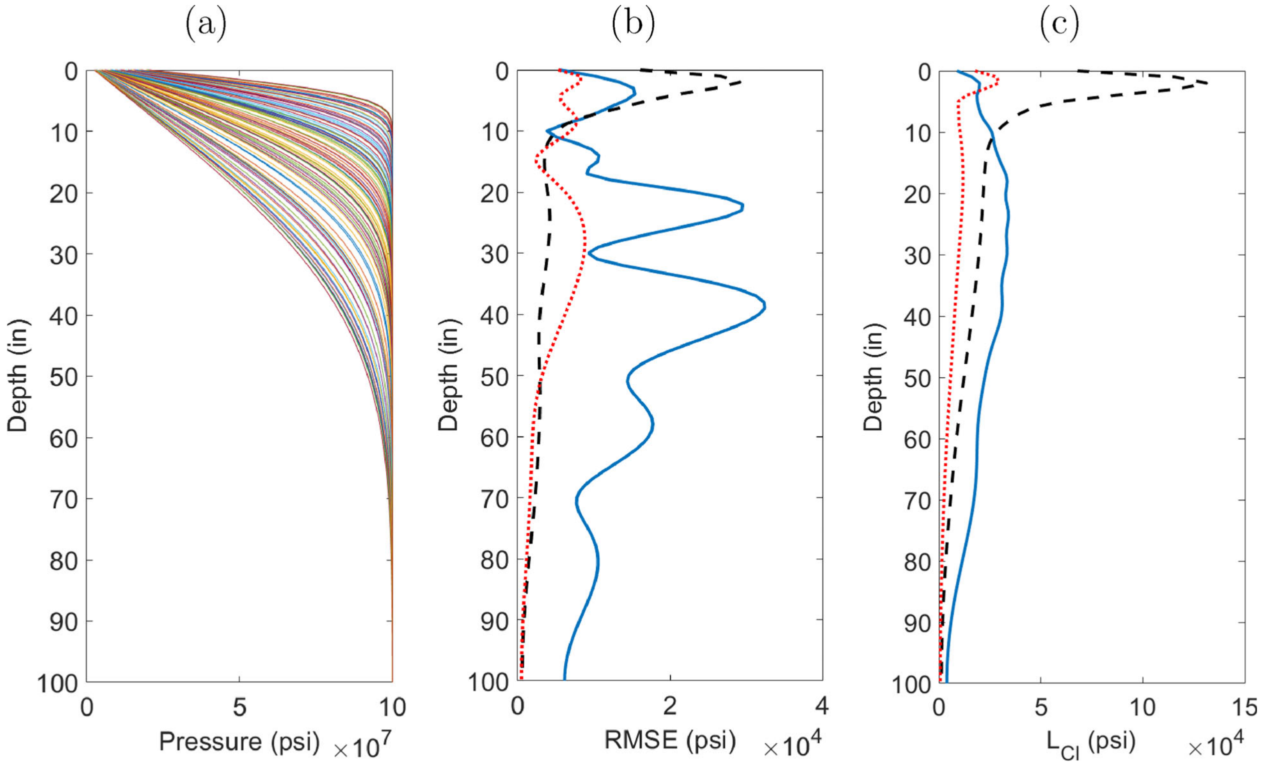

Figure 5. Global GP emulator results for the test case with constant permeability throughout the column. (a) Simulated pressure values along the column of mud used for testing the emulators. (b) RMSE over the column for PPCE (blue solid line), PPLE PCA (long dashed black line), and PPLE gKDR (short dashed red line). (c) L_{CI} over the column for each method with line styles corresponding to those shown in (b).

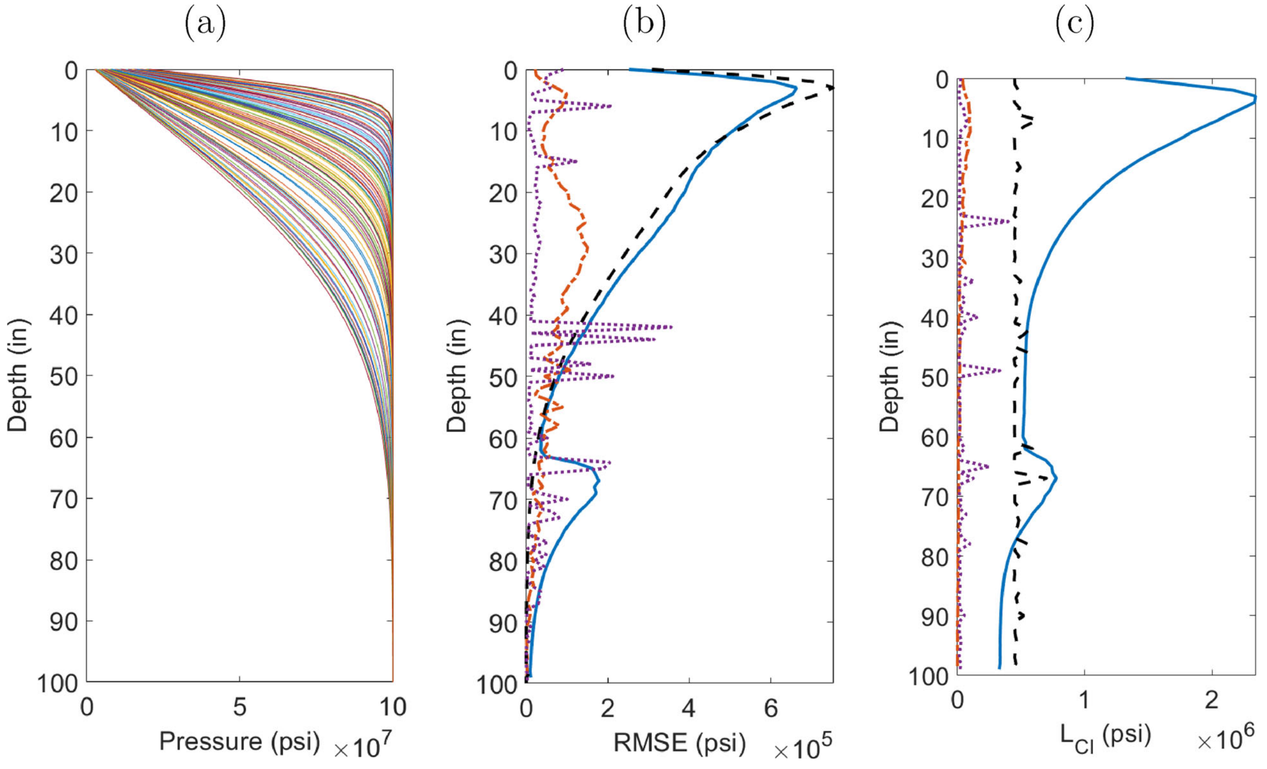

Figure 6. Single depth results for the test case with constant permeability throughout the column. (a) plot of the simulated pressure values along the column of mud used for testing the emulators; (b) the root mean squared error (RMSE) plots for CE laGP (solid blue line), LE laGP (dash-dotted orange line), CESC (long dashed black line), and LESC (short dashed purple line); (c) the average 95% credible interval (L_{CI}) for these methods.

we have held the other input parameters constant, this particular case is simple enough that all emulators give relatively small errors. Therefore, we extend our experiment to a variable permeability column in the next section. This case is both more

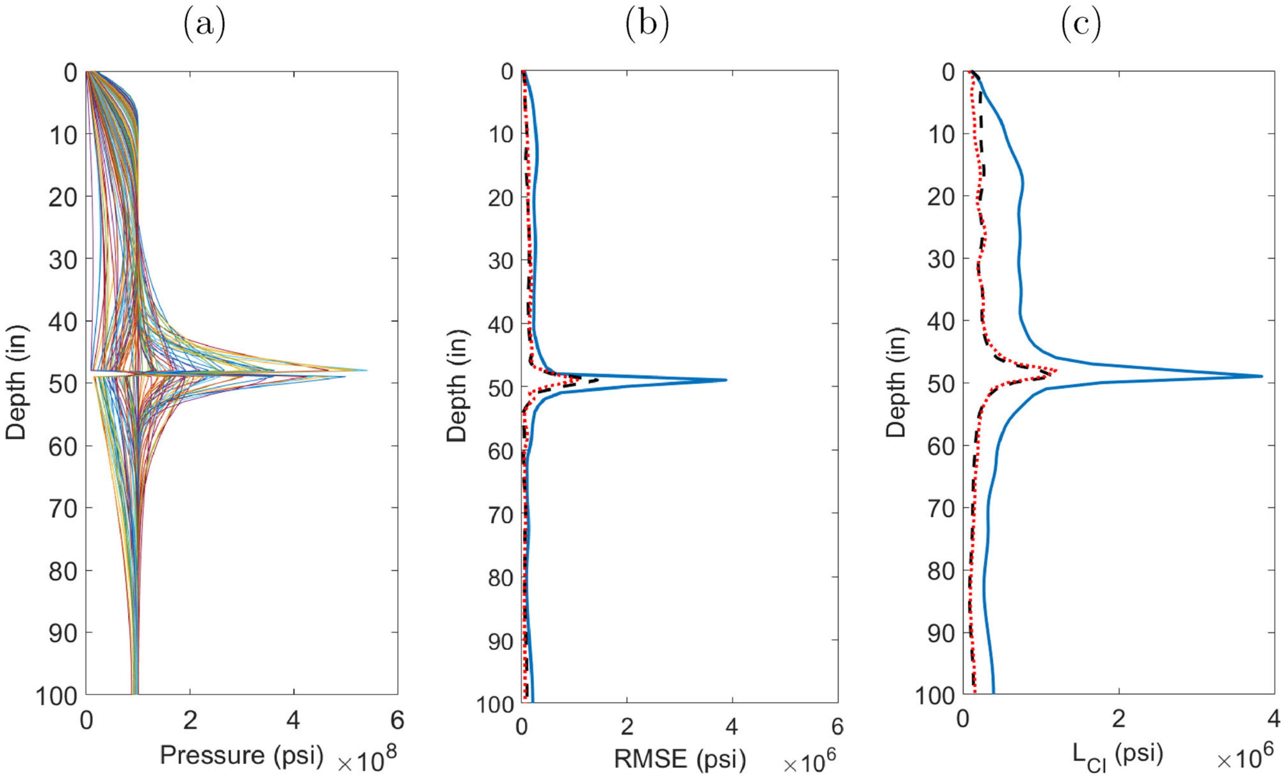

Figure 7. Results for the experiment with two permeability values in the column of mud. (a) The true pressure values along the mud column used for testing. (b) The root mean squared error (RMSE) plots for PPCE (blue solid line), PPLE PCA (long dashed black line), and LE gKDR (short dashed red line). (c) The average 95% credible interval (L_{CI}) for the three methods.

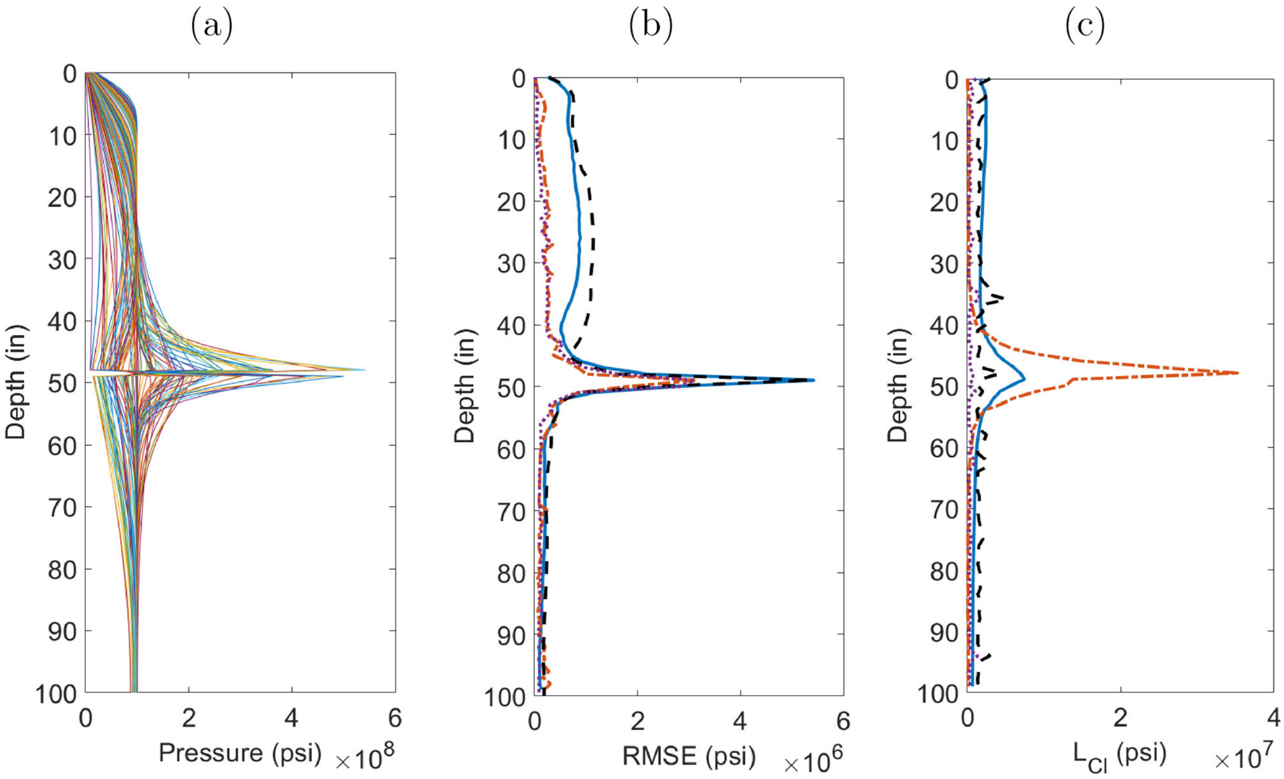

Figure 8. Results for the experiment with two permeability values in the column of mud. (a) the true pressure values along the mud column used for testing; (b) the root mean squared error (RMSE) plots for CE laGP (solid blue line), LE laGP (dash-dotted orange line), CESC (long dashed black line), and LESC (short dashed purple line); (c) the average 95% confidence interval (L_{CI}) for the four methods.

realistic and results in a discontinuity in the pressure field which poses a greater challenge for emulation.

Varying permeability experiment. We now repeat the previous experiment for a column of mud with two values for

Table 3. Comparison of the different emulation methods applied to the constant permeability Terzaghi experiment.

	PPLE PCA	PPLE gKDR	PPCE	LESC	CESC	LE laGP	CE laGP
Timing (s)	2	377	2	143	69	342	135
RMSE (psi)	4.1e3	3.9e3	1.4e4	3.6e4	1.7e5	5.7e4	2.0e5
L_{CI} (psi)	1.7e4	6.5e3	2.1e4	3.9e4	4.7e5	2.7e4	7.9e5
Coverage (%)	97	93	92	93	99	80	94

Table 4. Comparison of the different emulation methods applied to the two-permeability Terzaghi experiment.

	PPLE PCA	PPLE gKDR	PPCE	LESC	CESC	LE laGP	CE laGP
Timing (s)	2	383	1.8	176	62	291	140
RMSE (psi)	1.2e5	1.1e5	2.6e5	2.4e5	6.6e5	2.6e5	5.6e5
L_{CI} (psi)	2.2e5	2.4e5	6.0e5	4.1e5	1.7e6	1.7e6	1.8e6
Coverage (%)	93	92	94	91	100	86	95

permeability: one in the top half of the column, and a second distinct value in the bottom half. Figures 7, 8, and Table 4 summarize the results for the two-permeability simulation experiment, analogous to Figures 5, 6, and Table 3 for the constant permeability experiment. In Figure 7 we see that the results from the PPLE methods are nearly identical, both for RMSE and L_{CI}. Further, both PPLE's L_{CI} values are small throughout the column, and significantly smaller that the PPCE L_{CI} near the discontinuity in the pressure field halfway down the column. In Figure 8 we see that the two scalar-output linked methods (LESC and LE laGP) are comparable to each other with smaller RMSEs than the scalar-output composite methods (CESC and CE laGP). The LE laGP has the largest credible interval of all seven methods, particularly near the discontinuity in the pressure field. Note that while the scale of the errors is the same for Figures 7(b) and 8(b), the scales of the credible intervals in Figures 7(c) and 8(c) differ by an order of magnitude. Table 4 confirms that the two PPLE methods again outperform the other five methods while PPLE PCA is faster than PPLE gKDR. The timings in Tables 3 and 4 include fitting the emulators and a single predictive mean evaluation for a design with size $m = 100$ and $s = 100$ outputs locations. Evaluating a PPE predictive mean scales like $O(m^2) + O(ms)$ (Gu and Berger 2016) which the PPLE inherits. Additionally, the PPLE requires evaluations of the predictive means and variances for each inside emulator, each $O(m^2)$. In applications with $s \gg m$, this PPLE scaling which is dominated by $O(ms)$ is advantageous over evaluating many scalar component emulators which would scale as $O(m^2 s)$. Further, the PPLE also scales to large-dimensional output better than laGP methods where each new emulator mean evaluation for each output component effectively requires fitting a new laGP.

5. Discussion

There are several outstanding challenges and opportunities that would make the PPLE an even more effective tool for uncertainty quantification of multiphysics models. One common challenge for high-dimensional coupled simulators, which the PPLE would inherit, is a mismatch of spatial domains. For example, in more realistic flow and deformation simulation of a region of the subsurface undergoing compaction during pore pressure depletion from oil and gas production, the fluid flow domain would generally only consist of the reservoir itself. However, the mechanics domain would extend up to the Earth's surface to include the overburden, below the reservoir (underburden) and also further out laterally than just the reservoir domain (see Minkoff et al. 2004). In such situations, it might be possible for the PPLE to play a role in quantifying uncertainties introduced by such domain mismatches. A further challenge for the PPLE itself (which it inherits from the PPE) is that, although the mean and variance are readily available, there is no mechanism to draw spatially correlated samples from the PPLE.

We also note that while the examples in this article are to loosely coupled multiphysics simulators, the theory we develop would apply to any type of coupling (including full coupling) as long as the quantities of interest can be extracted from the simulator. For example, a fully coupled model would likely output pressure and displacement. One can estimate porosity from strain and hence from displacement. In our formulation here, porosity is the QoI passed from the inner to the outer simulator in our loosely coupled model, but it could be directly determined from a fully coupled multiphysics model as well and hence emulated. Further, perhaps only one simulator in a loosely coupled system is computationally expensive to exercise. Perhaps the whole space-time field of that expensive simulator could be emulated and the full multiphysics system then emulated with the PPLE. In such a case, it would be interesting to study how the emulator error propagates in this system and to determine if that error could be controlled. Another use of the PPLE might be for existing tightly coupled multiphysics systems such as biomechanics applications for cartilage of the knee (Shim et al. 2021). Conceptually, one may be able to glean knowledge of the simulator itself and build more effective emulators by emulating the interacting fields individually and combining them with the PPLE. This list of outstanding problems is just a subset of future extensions for the PPLE methodology described in this work.

6. Conclusions

In this work, we derive the mean and variance for the parallel partial linked Gaussian process emulator, which is a new and useful framework for investigating uncertainty quantification based on GP emulation of coupled, high-dimensional multiphysics models. Specifically we build on methodology for the linked emulator (Kyzyurova, Berger, and Wolpert 2018; Ming and Guillas 2021) and the parallel partial emulator (Gu and Berger 2016). The linked emulator is the Normal approximation to a composition of two Gaussian Processes, one for each coupled computer model under consideration. The parallel partial emulator is a computationally efficient approach for emulating computer models with high output dimensions because although it treats output components independently, it utilizes a shared correlation structure for all output components. The PPLE exploits the ease and efficiency of the PPE and the Linked emulator for high-dimensional, coupled simulators.

We demonstrate the effectiveness of the PPLE by applying it to two coupled systems with vector valued output. The first application of the PPLE is to a trigonometric composite function, and when compared to direct emulation of the composite function, the PPLE offers a more accurate predictive mean and tighter credible intervals. For the second application of the PPLE, we consider solving the Terzaghi consolidation problem for compaction of a column of mud saturated with a single fluid, which we solve via a two-way loose coupling of fluid flow and mechanical deformation. This loose coupling allows us to consider porosity changes in time. Specifically, we consider the porosity field output from the mechanics as an input to the flow simulator, and we emulate the resulting pressure field. Because this coupling parameter is high dimensional, we use dimension reduction, namely PCA or gKDR, in the emulation of the porosity field to construct the linked emulator for pressure without utilizing dimension reduction on the pressure field itself. We compare the PPLE to the parallel partial composite emulator as well as to 4 strategies that fit emulators to each scalar component of the output pressure vector. The PPLE produces a more accurate predictive mean and tighter credible intervals than either direct composite emulation or scalar-output component emulation. Further, the PPLE is very computationally efficient, and inherits smoothness properties from the output of the multiphysics simulator.

Supplementary Materials

The zip file contains MATLAB code and data to reproduce Figure 7. These files can also be found on GitHub Dolski, Spiller, and Minkoff (2024). This code employs two available MATLAB packages: `ppgasp()` and `gkDR()`. The function `gKDR.m` and associated function files are included in the zip file, and can also be found here Fukumizu and Leng (2013). The `RobustGaSP-in-Matlab` package can be found here: Gu (2019).

Acknowledgments

The authors would also like to thank the two anonymous referees and the associate editor for comments and suggestions that greatly improved this work.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

The authors gratefully acknowledge support by NSF grant # DMS-2053858 and # DMS-2053872, *CDSE: Collaborative Research: Surrogates and Reduced Order Modeling for High Dimensional Coupled Systems*, and by the sponsors of the UT Dallas *3D+4D Seismic FWI* research consortium. Spiller and Minkoff would also like to acknowledge support of the Reduced Order Modeling working group from the SAMSI Model Uncertainty Mathematical and Statistical program in 2018-2019 (NSF grant # DMS-1638521).

ORCID

Elaine T. Spiller https://orcid.org/0000-0001-7070-7800

References

Abdelfatah, K., Bao, J., and Terejanu, G. (2018), "Geospatial Uncertainty Modeling Using Stacked Gaussian Processes," *Environmental Modelling & Software*, 109, 293–305. [2]

Anowar, F., Sadaoui, S., and Selim, B. (2021), "Conceptual and Empirical Comparison of Dimensionality Reduction Algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE)," *Computer Science Review*, 40, 100378. [9]

Aziz, K. (1979), "Petroleum Reservoir Simulation," Basel: Applied Science Publishers, 476. [8]

Bayarri, M. J., Walsh, D., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., and Sacks, J. (2007), "Computer Model Validation with Functional Output," *The Annals of Statistics*, 35, 1874–1906. [2]

Bowman, V. E., and Woods, D. C. (2016), "Emulation of Multivariate Simulators Using Thin-Plate Splines with Application to Atmospheric Dispersion," *SIAM/ASA Journal on Uncertainty Quantification*, 4, 1323–1344. [2]

Brown, D. L., Bell, J., Estep, D., Gropp, W., Hendrickson, B., Keller-McNulty, S., Keyes, D., Oden, J. T., Petzold, L., and Wright, M. (2008), "Applied Mathematics at the US Department of Energy: Past, Present and a View to the Future," Technical Report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States). [1]

Currin, C., Mitchell, T., Morris, M. D., and Ylvisaker, D. (1988), "A Bayesian Approach to the Design and Analysis of Computer Experiments," Technical Report, Oak Ridge National Laboratory, Oak Ridge, TN (USA). [1]

Damianou, A., and Lawrence, N. D. (2013), "Deep Gaussian Processes," in *Artificial Intelligence and Statistics*, pp. 207–215, PMLR. [2]

Dangerfield, J. (1992), "Ekofisk Field Subsidence Fault Analysis Using 3-D Seismic,," in *Reservoir Geophysics*, ed. R. E. Sheriff, pp. 110–121, Tulsa, OK: SEG. [1]

Dean, R., Gai, X., Stone, C., and Minkoff, S. (2003), "A Comparison of Techniques for Coupling Porous Flow and Geomechanics," in *Proceedings of the 17th Reservoir Simulation Symposium*, Houston, TX.: SPE 79709. [2]

Dolski, T., Spiller, E. T., and Minkoff, S. M. (2024), "MATLAB pacakge for Parallel Partial Linked GP Emulators," available at *https://github.com/TamaraKozareva/Gaussian-Process-Emulation-for-High-Dimensional-Coupled-Systems*. [14]

Fredrich, J., Arguello, J., Thorne, B., Wawersik, W., Deitrick, G., de Rouffignac, E., Myer, L., and Bruno, M. (1996), "Three-dimensional Geomechanical Simulation of Reservoir Compaction and Implications for Well Failures in the Belridge Diatomite," in *Proceedings of the SPE Annual Technical Conference and Exhibition*, SPE, no. 36698, pp. 195–210. [1]

Fukumizu, K., and Leng, C. (2013), "gKDR MATLAB Package," available at *https://www.ism.ac.jp/~fukumizu/software.html*. [14]

——— (2014), "Gradient-based Kernel Dimension Reduction for Regression," *Journal of the American Statistical Association*, 109, 359–370. [2,9]

Gao, Y., and Pitman, E. B. (2023), "Parallel Partitial Emulation in Applications," Manuscript under review. [2,4,7]

Girard, A., Rasmussen, C., Candela, J. Q., and Murray-Smith, R. (2003), "Gaussian Process Priors with Uncertain Inputs Application to Multiple-Step Ahead Time Series Forecasting," in *Advances in Neural Information Processing Systems* (Vol. 15). [2]

Gramacy, R. B. (2016), "laGP: Large-Scale Spatial Modeling via Local Approximate Gaussian Processes in R," *Journal of Statistical Software*, 72, 1–46. [2,10]

Gramacy, R. B., and Apley, D. W. (2015), "Local Gaussian Process Approximation for Large Computer Experiments," *Journal of Computational and Graphical Statistics*, 24, 561–578. [2,4,10]

Gu, M. (2019), "RobustGaSP-in-Matlab," available at *https://github.com/MengyangGu/RobustGaSP-in-Matlab*. [14]

Gu, M., and Berger, J. O. (2016), "Parallel Partial Gaussian Process Emulation for Computer Models with Massive Output," *The Annals of Applied Statistics*, 10, 1317–1347. [2,3,4,7,10,13]

Gu, M., Palomo, J., and Berger, J. O. (2019), "RobustGaSP: Robust Gaussian Stochastic Process Emulation in R," *The R Journal*, 11, 112–136. [4,6,10]

Gu, M., and Shen, W. (2020), "Generalized Probabilistic Principal Component Analysis of Correlated Data," *Journal of Machine Learning Research*, 21, 1–41. [9]

Gu, M., Wang, X., and Berger, J. O. (2018), "Robust Gaussian Stochastic Process Emulation," *The Annals of Statistics*, 46, 3038–3066. [4]

Gutierrez, M., and Lewis, R. (2002), "Coupling of Fluid Flow and Deformation in Underground Formations," *Journal of Engineering Mechanics*, 128, 779–787. [1]

Gutierrez, M., Lewis, R., and Masters, I. (2001), "Petroleum Reservoir Simulation Coupling Fluid Flow and Geomechanics," *SPE Reservoir Evaluation and Engineering*, 4, 164–172. [1]

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, 103, 570–583. [2]

Keyes, D. E., McInnes, L. C., Woodward, C., Gropp, W., Myra, E., Pernice, M., Bell, J., Brown, J., Clo, A.and Connors, J. et al. (2013), "Multiphysics simulations: Challenges and opportunities," *The International Journal of High Performance Computing Applications*, 27, 4–83. [1]

Kim, J., Tchelepi, H. A., and Juanes, R. (2011), "Stability, Accuracy, and Efficiency of Sequential Methods for Coupled Flow and Geomechanics," *SPE Journal*, 16, 249–262. [2]

Kyzyurova, K. N. (2017), "On Uncertainty Quantification for Systems of Computer Models," Ph.D. thesis, Duke University. [6]

Kyzyurova, K. N., Berger, J. O., and Wolpert, R. L. (2018), "Coupling Computer Models through Linking their Statistical Emulators," *SIAM/ASA Journal on Uncertainty Quantification*, 6, 1151–1171. [2,3,5,6,13]

Lee, H., Spiller, E. T., and Minkoff, S. E. (2019), "Dimension Reduction and Global Sensitivity Metrics Using Active Subspaces for Coupled Flow and Deformation Modeling," in *SEG Technical Program Expanded Abstracts 2019*, Society of Exploration Geophysicists, pp. 3240–3244. [9]

Lewis, R., and Ghafouri, H. (1997), "A Novel Finite Element Double Porosity Model for Multiphase Flow through Deformable Fractured Porous Media," *International Journal for Numerical and Analytical Methods in Geomechanics*, 21, 789–816. [1]

Lewis, R., and Sukirman, Y. (1993a), "Finite Element Modelling for Simulating the Surface Subsidence Above a Compacting Hydrocarbon Reservoir," *International Journal for Numerical and Analytical Methods in Geomechanics*, 18, 619–639. [1]

——— (1993b), "Finite Element Modelling of Three-Phase Flow in Deforming Saturated Oil Reservoirs," *International Journal for Numerical and Analytical Methods in Geomechanics*, 17, 577–598. [1]

Marque-Pucheu, S., Perrin, G., and Garnier, J. (2020), "An Efficient Dimension Reduction for the Gaussian Process Emulation of Two Nested Codes with Functional Outputs," *Computational Statistics*, 35, 1059–1099. [2]

McClure, M. (2012), "Modeling and Characterization of Hydraulic Stimulation and Induced Seismicity in Geothermal and Shale Gas Reservoirs," Ph.D. thesis, Stanford University Stanford, California. [7]

McClure, M., and Horne, R. N. (2011), "Investigation of Injection-Induced Seismicity Using a Coupled Fluid Flow and Rate/State Friction Model," *Geophysics*, 76, WC181–WC198. [7]

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 21, 239–245. [7]

Middleton, G. V., and Wilcock, P. R. (1994), *Mechanics in the Earth and Environmental Sciences*, Cambridge: Cambridge University Press. [9]

Ming, D., and Guillas, S. (2021), "Linked Gaussian Process Emulation for Systems of Computer Models Using Matérn Kernels and Adaptive Design," *SIAM/ASA Journal on Uncertainty Quantification*, 9, 1615–1642. [2,3,5,6,7,10,13]

Minkoff, S. E., and Kridler, N. M. (2006), "A Comparison of Adaptive Time Stepping Methods for Coupled Flow and Deformation Modeling," *Applied Mathematical Modelling*, 30, 993–1009. [1,9]

Minkoff, S. E., Stone, C. M., Bryant, S., and Peszynska, M. (2004), "Coupled Geomechanics and Flow Simulation for Time-Lapse Seismic Modeling," *Geophysics*, 69, 200–211. [1,7,9,13]

Minkoff, S. E., Stone, C. M., Bryant, S., Peszynska, M., and Wheeler, M. F. (2003), "Coupled Fluid Flow and Geomechanical Deformation Modeling," *Journal of Petroleum Science and Engineering*, 38, 37–56. [1,2,7,9]

Osorio, J., Chen, H., and Teufel, L. (1999), "Numerical Simulation of the Impact of Flow-Induced Geomechanical Response on the Productivity of Stress-Sensitive Reservoirs," in *Proceedings of the 15th Reservoir Simulation Symposium*, SPE, no. 51929, pp. 373–387. [1]

Rasmussen, C. E., and Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, Adaptative Computation and Machine Learning Series, Boston, MA: The MIT Press. [3]

Sacks, J., Schiller, S. B., and Welch, W. J. (1989), "Designs for Computer Experiments," *Technometrics*, 31, 41–47. [1,4,7]

Sanson, F., Le Maitre, O., and Congedo, P. M. (2019), "Systems of Gaussian Process Models for Directed Chains of Solvers," *Computer Methods in Applied Mechanics and Engineering*, 352, 32–55. [2]

Santner, T. J., Williams, B. J., and Notz, W. I. (2018), *The Design and Analysis of Computer Experiments* (2nd ed.), Springer Series in Statistics, New York: Springer-Verlag. [1,3]

Settari, A., and Mourits, F. (1994), "Coupling of Geomechanics and Reservoir Simulation Models,"*Computer Methods and Advances in Geomechanics*, eds. H. J. Siriwardane and M. Zaman, pp. 2151–2158, Rotterdam: Balkema. [2]

Shim, J. J., Maas, S. A., Weiss, J. A., and Ateshian, G. A. (2021), "Finite Element Implementation of Biphasic-Fluid Structure Interactions in febio," *ASME Journal of Biomechanical Engineering*, 143, 091005. [13]

Terzaghi, K., and Peck, R. B. (1948), *Soil Mechanics in Engineering Practice*, New York: Wiley. [7]

Varin, C., Reid, N., and Firth, D. (2011), "An Overview of Composite Likelihood Methods," *Statistica Sinica*, 21, 5–42. [4]

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15–25. [1,4,7]

Yudovich, A., and Morgan, D. (1989), "Casing Deformation in Ekofisk," *Journal of Petroleum Technology*, 41, 729–734. [1]