Mathematical Biology manuscript No.
(will be inserted by the editor)

Gaussian process emulator as a tool for upscaling complex multi-scale stochastic biological models

Oluwole K. Oyebamiji · Darren J. Wilkinson · Prashant Gupta · Jayathilake Gedara

Received: date / Accepted: date

Abstract The performance of credible and reliable simulations in open engineered biological frameworks could offer practical application of the scientific knowledge to solve real-world problems and enhance our ability to make novel discoveries. Therefore, maximizing potential to explore the range of solutions at this frontier level could reduce the potential risk of failure at a large scale. The simulation parameters at this micro level could then be derived from first principles of biological life, combined with principle of evolution and ecology. One major application of this type of knowledge is in management of wastewater treatment system. Wastewater treatment plant optimization focuses on aggregate outcomes of individual particle-level processes and behavioural rules.

One of the crucial aspects of engineering biology approach in wastewater treatment study is to run a high complex simulation of biological particles. The model has the ability to scale from one level to another to better understand how to effectively manage real systems with minimal physical experimentation. Nevertheless, simulation of open biological systems is difficult because they often involve a large number of bacteria that ranges from order 10^{12} to 10^{18} individual particles and are physically and genetically complex. The models are computationally expensive and due to computing constraints, limited set of scenarios are often possible.

This problem can be eradicated by using a statistical approximation of the complex models which will help in reducing the computational burden. Our aim in this paper is to build a cheaper surrogate models (called metamodels) from simulations of the LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator), a classical molecular dynamical model for biological particle simulation. The paper focuses on how to use an emulator as an effective tool for studying and incorporating microscale processes in a computationally efficient way into macroscale models. The main issue we address is to highlight the strategy for upscaling high-level summary from LAMMPS simulation data which involves transferring of information from one spatial or temporal scale to another one. We propose a Gaussian process regression in form of kriging metamodel.

Newcastle University, School of Mathematics & Statistics
Newcastle upon Tyne, NE1 7RU, UK.
Tel.: (+44) 7411875750
E-mail: wolemi2@yahoo.com
Darren J. Wilkinson
Newcastle University, School of Mathematics & Statistics
Newcastle upon Tyne, NE1 7RU, UK.
Jayathilake Gedara
Newcastle University, Department of Mechanical & Systems Engineering
Newcastle upon Tyne, NE1 7RU, UK.
Prashant Gupta
Newcastle University, Department of Biology,
Newcastle upon Tyne, NE1 7RU, UK.

Keywords Bayesian techniques · GP emulator · biofilms · floc · kriging

1 Introduction

There is a common assumption that to identify crucial features and model water treatment plant on a large scale, there is a need to understand the interactions of microbes at fine resolution based models that could provide the best available representation of micro scale responses. The challenge then becomes how we can transfer this small-scale information to the macroscale process via a mesoscale in a computationally efficient and sufficiently accurate way, and to also probably quantified the associated risk or error in the process.

The macro scale characteristics of wastewater treatment plants are the consequences of microscale features of a vast number of individual particles that produce the floc (Ofiteru *et al*., 2014). In other words, the properties of cells or particles at a micro level is used for characterising the behaviour of wastewater treatment plant at a macro scale, even though there is a wide separation in their spatial and temporal dimensions at which their biological and physical processes take place. Flocs are aggregation of microbes mixed with an adhesive material called EPS. They are often difficult to measure or quantify because of their irregular size and shape. A wide range of different equivalent diameters is often used to characterize the floc size, see Jarvis *et al*. (2005) for further details. The individual particle that makes up the flocs are simulated as a sphere of a variable volume. The floc plays a strategic role in understanding the process involves in wastewater treatment plant.

The complex nature of the transitions from cellular level (microscale) to floc (mesoscale) and to wastewater treatment plant (macro-scale) introduce a scaling problem and a robust and coherent strategy is required to efficiently handle this multi-scale problem. One useful approach to this challenge is the use of statistical emulators called metamodels. Emulation is a statistical technique for simplifying models that leads to reduced-form representations of complex models that are computationally much faster to run.

Statistical emulation can help to overcome this problem. Emulation integrates both the process-based model and statistical techniques. Besides, emulators offer rapid and relatively quick alternatives for projection of climate change impacts on agricultural productivity for diverse climate scenarios (Oyebamiji *et al*., 2015). Another benefit of emulation is the provision of a measure of uncertainties associated with the projections.

The aim of this paper is to describe how to use an emulator as an effective tool for understanding and incorporating microscale processes in a computationally efficient way into macroscale models. The main issue we address here is the upscaling problem that involves transferring of information from one spatial or temporal scale to another one. The focus is to apply emulator to adapt and relate LAMMPS model output predictions from an individual particle levels (microscale) to make predictions of an aggregate of particles of varying species called floc/biofilm at mesoscale levels. Subsequently, to further transfer the information to macro-level processes of wastewater treatment plants. Van *et al*. (2009) earlier reviewed some of the popular techniques for upscaling complex problems while Frazer *et al*. (2013) and Wheater *et al*. (2008) specifically focused their attentions on how to use emulators for upscaling hydrological processes and land use management properties.

Young *et al*. (2011) described the behaviour of large linear dynamic models using statistical principles of dynamic emulation. Their approach identifies a low-order model that approximates the behaviour of the high-order dynamic simulator that is much cheaper. Oakley & O'Hagan (2004) described a Bayesian method for quantification of uncertainty in complex computer models. Kennedy *et al*. (2006) presented some notable examples where GP modelling applications have been implemented.

Similarly, Higdon *et al*. (2008) applied the Oakley & O'Hagan (2002) approach in conjunction with a PCA for basis representations of high-dimensional output. Apart from reducing the dimensionality of the problem this PCA technique also reduces the computation time required for obtaining the posterior distributions. A closely related study of Wilkinson (2010) performed a calibration of multivariate experiments by extending the approach of Kennedy *et al*. (2001) to multivariate models.

Due to the spatio-temporal nature of LAMMPS outputs, our approach is to condense the massive, long time series outputs of particles of various species from LAMMPS models by spatially aggregating to produce the most relevant outputs in the form of floc aggregates or biofilm. The data compression has the benefit of suppressing or reducing some of the nonlinear response features, simplifying the construction of the emulator. Some of highly interested properties at the mesoscale level like the size, shape and structure of biofilm/floc are characterized. For instance, we approximate the floc size using an equivalent diameter. This strategy will enable us to treat the floc as a ball of a sphere, and we can emulate the diameter of a sphere that circumscribes its boundary/outline. The center of the sphere will be equivalent to the center of mass of the component particles. See Figure (1).

We use the Gaussian process emulation in the form of kriging metamodels where output data can be decomposed into a mixture of deterministic (non-random trend) and a residual random variation. Our approach combines the two stage technique proposed in O'Hagan (2006) and Oyebamiji *et al*. (2015) as a single step and is also related to Higdon *et al*. (2008) who combine GP emulation with a basis representation for calibration of computer models with high dimensional outputs.

We describe the models and simulation data used for the analysis in Section 2. In Section 3 we describe the methods and emulation procedures. Section 4 provides the results of the analysis. Sections 5 and 6 present the discussion and concluding comments respectively.

2 Simulation model

2.1 LAMMPS model description

2.2 Experimental design

This section describes the procedure for generating the parameter combinations and variables at which the LAMMPS model is run. We run the LAMMPS code for a small sample of inputs using a Latin Hypercube Design (LHD). This produces data for training the LAMMPS emulator to approximate the major LAMMPS outputs. LHD provides a good coverage of the input space with a relatively small number of design points. We use maximin LHS technique that optimises samples by maximizing the minimum distance between design points Santner et al. (2003). Suppose we want to sample a function of p variables, the range of each variable is divided into n probable intervals, n sample points are then drawn such that a Latin Hypercube is created. We generate an $n \times p$ variables Latin Hypercube sample matrix with values uniformly distributed on interval [0,1]. We then transform the generated sample to the quantile of a uniform distribution using the range of the parameters given in Table 1.

2.3 Simulation data

Let the design matrix which contain the input to the LAMMPS model be denoted by $\mathbf{X} = (\theta_p^i, t, p = 1, \ldots, 25; i = 1, \ldots, 1000)$, where the subscript $p = 25$ represents 20 callibrated model parameters and 5 variables that represent the model initial conditions (see Table 1), superscript i denote the 1000 different realisations (design points) and t is the time slice in seconds at which the output data is recorded $t = 1, \ldots, T$. The design matrix $\mathbf{X}_{1000\times26}$ denotes the input values at which the LAMMPS model is run for every combinations of x_i which is a point in $\mathbf{X}$, where x_i represents i^{th} row of $\mathbf{X}$. The LAMMPS code is run for two days (172800 s simulation time).

The model is computationally demanding, we limit our simulations to just 100 replicates at each design point i. The essence of repeated runs is to incorporate stochastic variations in our outputs. The output results are recorded at a time-step of 2000 seconds which gives about 86 different time slices. The data will provide initial estimates of the mean and variance of the LAMMPS model.

The current LAMMPS code is set up to produce the following outputs namely particle diameter, mass, position (3-dimensional) for each time step t. The time series output at each design point is denoted as a matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T]$, such that $\mathbf{y} = [y_1, \ldots, y_n]$, where $T = 86$ in this simulation and n is the total number of particles at each time step. The number of particles *n* at each time slice varies across the design points and, in particular, increasing with time as it expected for the microbes to be growing. An independent simulation of 100 runs with ten replicates is performed for cross-validation purpose. Here, the simulation is run for a longer period than the previous simulations (4 days simulation). We consider emulation of floc which is summarized by aggregating all the individual microbe at each time step.

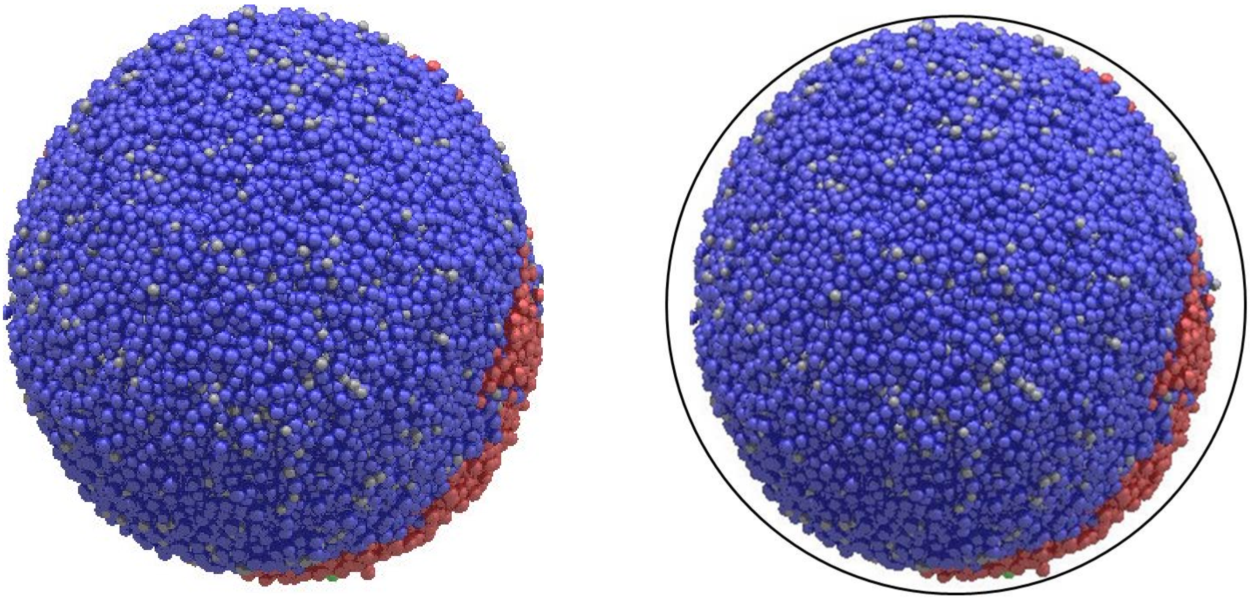


Fig. 1: Transformation of microscale particles to floc at the mesoscale for a particular time. Floc equivalent diameter is the diameter of the smallest sphere that circumscribes the outline of the projected floc.

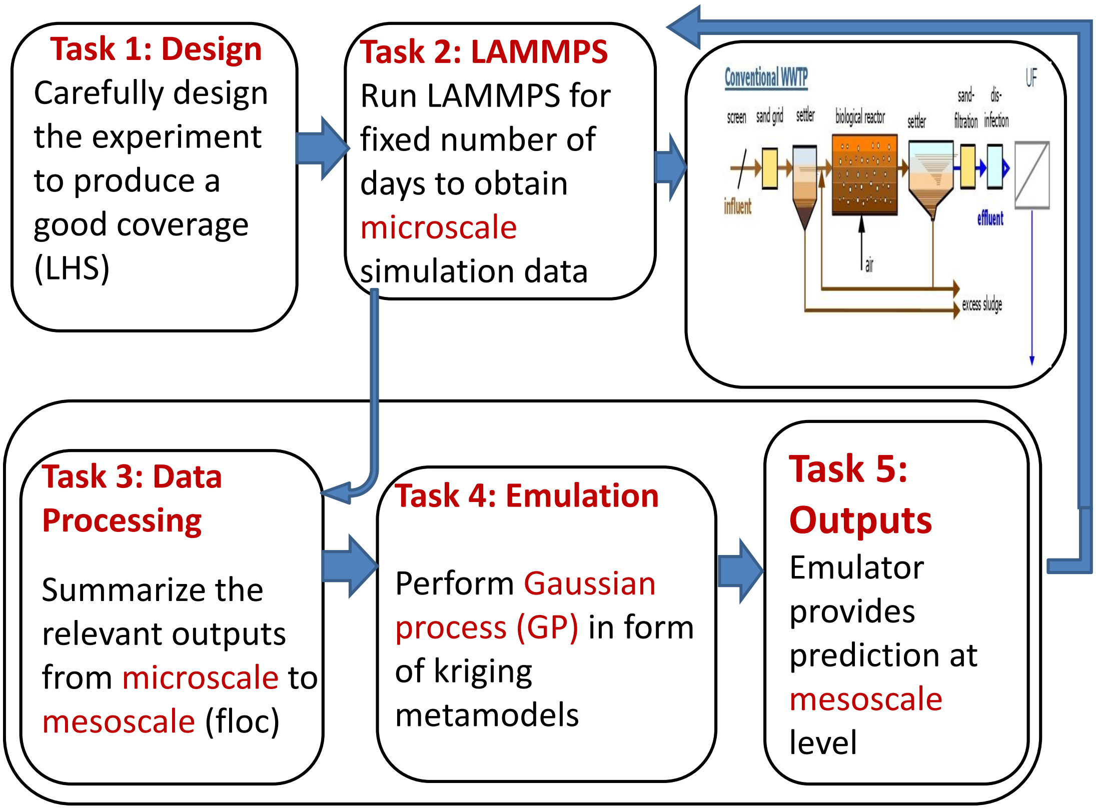

Task 1: Design
Carefully design the experiment to produce a good coverage (LHS)
Task 2: LAMMPS
Run LAMMPS for fixed number of days to obtain microscale simulation data
Conventional WWTP
UF
screen
sand grid
settler
biological reactor
settler
sand-filtration
dis-infection
influent
effluent
air
excess sludge
Task 3: Data Processing
Summarize the relevant outputs from microscale to mesoscale (floc)
Task 4: Emulation
Perform Gaussian process (GP) in form of kriging metamodels
Task 5: Outputs
Emulator provides prediction at mesoscale level

Fig. 2: Schematic diagram showing key emulation stages

3 Methods

A Bayesian framework for emulation is almost always based on the assumption that a Gaussian process prior distribution can be specified for unknown parameters and hyperparameters. Under a Bayesian perspective unknown parameters are treated as random variables. The given prior distribution can be updated from training data. Applying Bayes rule to this setting, a posterior distribution can be obtained. The posterior distribution is also a Gaussian process. We are implementing GP technique in form of kriging because of its wide applicability and flexibility. Another additional benefit of GP modelling is the quantification of the model uncertainty.

A major difficulty with GP modelling is the computational effort associated with dealing with large data, as computer time scales are of order $O(n^3)$ where n is the number of observations. Several techniques have been adopted to overcome this computational problem. Earlier techniques are documented in Rasmussen & Williams (2006) and Quinonero-Candela & Rasmussen (2005). GP emulation is based on the Bayesian technique and experimental design of computer experiments for predicting model outputs at test input point Sacks *et al.* (1989) and Santner *et al.* (2003). A GP emulator assumes that a simulator output is an unknown function $g(.)$. We can then choose a prior distribution for $g(.)$ using the Bayesian approach and update this distribution, with some data obtained from the simulator runs.

3.1 Kriging

Kriging is a geostatistical technique for interpolating the value of an unknown random observation from data $\mathbf{y}(\mathbf{x})$ observed at known locations. Kriging models are also commonly used for building cheaper surrogate model of expensive computer codes Currin et al. (1991), Martin & Simpson (2004), Osio et al. (1996), Li & Sudjianto (2005). The two stage techniques describes in O'Hagan (2006) are combined as a single step. Here, $\mathbf{y}(\mathbf{x})$ can be decomposed into a mixture of deterministic (non-random trend) and a residual random variation. The trend could be modelled as a constant in ordinary/simple kriging or as an n^{th} order polynomial in universal kriging. In this section, we shall discuss the universal kriging technique. The model formulation is given as

$$\mathbf{y}(\mathbf{x}) = f(\mathbf{x}) + \varepsilon(\mathbf{x}) \tag{1}$$

where $\mathbf{y}(\mathbf{x})$ is the output of interest (eg floc equivalent diameter). The deterministic function $f(\mathbf{x})$ is the mean approximation of the expensive computer simulator (eg LAMMPS) and f is a polynomial function. Under this assumption, $f(x)$ can be modelled as

$$f(\mathbf{x}) = \sum_{j=1}^{p} \beta_j h_j(x) = \mathbf{H}(x)\beta \tag{2}$$

$\beta = [\beta_1, \ldots, \beta_p]$ is a $(p \times 1)$ vector of unknown regression coefficients and $\mathbf{H}(x) = \Big[h_1(x), ..., h_p(x)\Big]^T$ is a $(n \times p)$ matrix of regression functions, $\varepsilon(\mathbf{x})$ is a stochastic Gaussian process with mean zero and characterize by its covariance function $K = Cov(\varepsilon(\mathbf{x}), \varepsilon(\mathbf{x})) = \sigma^2 \mathbf{C}(\mathbf{x}, \mathbf{x}')$, where σ^2 denotes the variance of $\varepsilon(\mathbf{x})$ also called process variance and $\mathbf{C}$ is a $(n \times n)$ positive definite matrix of correlation between $\varepsilon(\mathbf{x})$'s at the experimental design points. We are assuming a univariate output and a deterministic computer model.

Similarly, $\mathbf{t}(x^{new}) = \Big[Cor(x_1, x^{new}), \ldots, Cor(x_n, x^{new})\Big]^T$ for the $(n \times 1)$ vector of correlations between the $\varepsilon(\mathbf{x})$'s at the design points and new input points x^{new}. We use both Gaussian equation and exponential correlation functions, equations (3,4) Sacks et al. (1989), Kleijnen (2009), Kleijnen & Simpson (2005).

$$\mathbf{C} = \Big\{\exp(-(x - x')^T \alpha (x - x'))\Big\}, \tag{3}$$

$$\mathbf{C} = \exp(-(x - x')/\alpha) \tag{4}$$

where α is the correlation hyperparameters to be estimated from the data.

The universal kriging predictor $\mu^{\bullet}(x^{new})$ of the value of $\mathbf{y}(x)$ at the new target point x^{new} is the linear predictor

$$\mu^{\bullet}(x^{new}) = \sum_{j=1}^{n} \lambda_j \mathbf{y}(x_j) = \lambda(x)^T \mathbf{y}, \tag{5}$$

for the sample points $x_1, \ldots, x_n$, where the coefficients or weights $\lambda = (\lambda_1, \ldots, \lambda_n)^T$ are estimated by minimizing the variance of prediction error for each realizations of the random function $\mathbf{y}(x)$. The best linear unbiased predictor (BLUP) is computed by minimizing the mean squared error

$$MSE[\mu^{\bullet}(x^{new})] = E\Big[\lambda^T \mathbf{y} - \mathbf{y}(x^{new})\Big]^2, \tag{6}$$

subject to the unbiasedness constraint $E\Big[\lambda^T(x)\mathbf{y}\Big] = E\Big[\mathbf{y}(x^{new})\Big]$. The mean squared error in equation (6) can be rewritten by substituting the value of $\mathbf{y}$ in equation (1), we thus have

$$\sigma^2\Big[1 + c\lambda^T(x)\mathbf{C}\lambda(x) - 2\lambda^T(x)\mathbf{t}(x^{new})\Big], \tag{7}$$

The unbiasedness constraint is now denoted as $H^T\lambda(x) = h(x)$. The optimal weights λ_j^* in $\mathbf{y}(x^{new})$ is estimated by using Lagrange multipliers to constraining MSE minimization. The Lagrange multiplier is also use to solve system of equations in order to compute p coefficients of λ. Given the inverse of matrix $\mathbf{C}$, then the best linear unbiased predictor for kriging model is given as

$$\mu_{uk}^{\bullet}(x) = h^T(x)\hat{\beta} + \mathbf{t}^T(x)\mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\hat{\beta}) \tag{8}$$

Similarly, the variance follows by substituting in the optimal value of $\lambda^*(x)$ in the MSE equation, we have

$$\mathbf{K}^{\bullet}_{uk} = \hat{\sigma}^2 \Big\{ C(\mathbf{x},\mathbf{x}') - \mathbf{t}(\mathbf{x})^T \mathbf{C}^{-1}\mathbf{t}(\mathbf{x}) + \Big(h(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T\mathbf{C}^{-1}\mathbf{t}(\mathbf{x})\Big)(\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\Big(h(\mathbf{x}')^T - \mathbf{t}(\mathbf{x}')^T\mathbf{C}^{-1}\mathbf{t}(\mathbf{x}')\Big)^T\Big\}. \tag{9}$$

See more details in (Sacks et al., 1989, Santner et al., 2003, Kleijnen & Mehdad, 2014).

The next problem is how to estimate the unknown parameters. We use a Maximum Likelihood Estimation (MLE) technique like many other authors (Santner et al., 2003, Kleijnen & Mehdad, 2012), are being used as an estimator of the kriging model parameters because of its computational efficiency, $\theta = (\beta, \sigma^2, \alpha)$. MLE is based on the assumption of Gaussian probability distribution. The likelihood of the model parameters is defined as the probability of the n observations $\mathbf{y} = y_1, , y_n$, given the model parameters such that

$$L(\theta|\mathbf{y}) = \Pi^n_{j=1} p(\mathbf{y}_j|\theta). \tag{10}$$

The expression for $L(\theta|\mathbf{y})$ is given by equation below as

$$L(\beta, \sigma^2, \alpha; \mathbf{y}) \propto \frac{|\mathbf{C}|)^{-\frac{1}{2}}}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{\frac{(\mathbf{y}-H\beta)^T\mathbf{C}^{-1}(\mathbf{y}-H\beta)}{2\sigma^2}\right\} \tag{11}$$

where $\alpha = [\alpha_1, \ldots, \alpha_n]$ is a vector of correlation lengths and $|\mathbf{C}|$ is the determinant of correlation matrix $\mathbf{C}$ and $K = \sigma^2\mathbf{C}$. By taking the derivative of the log-likelihood of equation (27) with respect to β and σ^2 and solving for zero, the estimates $\hat{\beta}$ and $\hat{\sigma}^2$ are given respectively as $\hat{\beta} = (\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{C}^{-1}\mathbf{y}$ and $\hat{\sigma^2} = \frac{1}{n}\Big[(\mathbf{y}-\mathbf{H}\hat{\beta})^T\mathbf{C^{-1}}(\mathbf{y}-\mathbf{H}\hat{\beta})\Big]$. The alternative way of performing this computation under fully-Bayesian technique is to marginalise the conditional $\Big(f(.)|\mathbf{y},\beta,\sigma^2,\alpha\Big)$ with respect to posteriors of β and σ^2. To estimate α, we maximize over α and σ^2, the concentrated likelihood given below after plugging the values of $\hat{\beta}$

$$-2logL(\hat{\beta}, \sigma^2, \alpha; \mathbf{y}) = nlog(2\pi) + log(|\mathbf{C}|) + \Big\{(\mathbf{y}-\mathbf{H}\hat{\beta})^T\mathbf{C}^{-1}(\mathbf{y}-\mathbf{H}\hat{\beta})\Big\} \tag{12}$$

We observe that that both $\hat{\sigma}^2$ and $\mathbf{C}$ depend upon the correlation parameter α. The trend and covariance parameters θ can be computed quickly and very efficiently by using a global optimiser which is based on the extension of the efficient algorithm proposed in Park & Baek (2001) for likelihood maximization. Further details are provided in (Roustant et al., 2012).

Because of the stochastic nature of the data we analyse in this paper, we briefly describe the extension of above derivation for the noisy observations, covariance $K = \sigma^2\mathbf{C}$ is replaced by $\sigma^2\mathbf{C} + diag(\tau_1^2, \ldots, \tau_n^2)$ in equations (27,29,28) respectively, where $\tau^2 = \tau_1^2, \ldots, \tau_n^2$ are the noise variances. Kriging technique is equivalent to the Bayesian method describes in Hankin (2005) and Oakley (1999), if we assign improper uniform priors on the β. In other words, non-informative Bayesian analysis often leads to kriging predictor and variance and these estimators appear respectively as conditional mean and variance in equations (28 and 29). The following is a summary of the calculations above.

(i) Perform the kriging emulation of data $\mathbf{y}$.

(ii) Given the posterior density of α as defined in equation (**??**) of Appendix 1, compute the MLE of α, such that equation (**??**) is maximized.

(iii) Set $\mathbf{C} = \mathbf{C}(\hat{\alpha})$.

(iv) Compute estimate $\hat{\beta} = (\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{C}^{-1}\mathbf{y}$

(v) Compute estimate $\widehat{\sigma^2} = \frac{1}{n}\Big[(\mathbf{y}-\mathbf{H}\hat{\beta})^T\mathbf{C^{-1}}(\mathbf{y}-\mathbf{H}\hat{\beta})\Big]$.

(vi) Compute posterior mean estimate $\mu^{\bullet}_{uk}(x)$ given in equation (28).

(vii) Compute posterior variance estimate $\mathbf{K}^{\bullet}_{uk}$ given in equation (29) above.

3.2 Multivariate kriging

Multivariate kriging extends kriging to m observation types popularly called 'cokriging'. Cokriging has been widely applied in various area especially in multifidelity surrogate models (Forrester et al., 2007, Kuya et al., 2011) where there are array of m levels of code usually from the expensive (accurate) to the less expensive (crude) simulators.

Suppose we have m output levels of code $\mathbf{Y}(x) = (Y_1(x), \dots, Y_m(x))$, The k^{th} output $y_k(x)$ is modelled as a Gaussian process $y_k(x) = Y_k(x)$. We use an autoregreesive (AR) model earlier proposed by Kennedy & O'Hagan (2000) which is based on Markov property such that $Cov(Y_t(x), Y_{t-1}(x)|Y_{t-1}(x)) = 0, x \neq x'$ and recently applied by Le Gratiet (2013). The model formulation assumes

$$Y_k(x) = \rho_{k-1} Y_{k-1}(x) + \delta_k(x) \tag{13}$$

for $k \in (2, \dots, m)$, $\delta_k(x)$ is a Gaussian process that models the bias between the output k and the output $k-1$ adjusted and ρ_{k-1} is the scaling factor between Y_k and Y_{k-1}. The ρ_{k-1} can be further treated as a linear regression function such that

$$\rho_{k-1}(x) = g_{k-1}^T(x)\gamma_{k-1} \tag{14}$$

and $g_{k-1}^T(x)$ is a vector of regression functions with covariance function of the form $c_k(x,x) = \sigma_k^2 r_k(x-x;\alpha_k)$, where σ_k^2 is the variance of the Gaussian process and α_k are the correlation hyper parameters of correlation function r_k. In addition, since each of $Y_k(x)$ is a GP then the joint process $(Y_1(x), \dots, Y_m(x))$ is a multivariate GP with mean

$$E[Y_k(x)|\sigma^2, \alpha, \beta, \gamma] = h_k(x)^T \beta \tag{15}$$

and covariance function

$$cov\Big[Y_k(x), Y_k(x')|\sigma^2, \alpha, \mathbf{B}, \gamma_k\Big] = \sum_{k=1}^{m} \sigma_k^2 \Big(\prod_{i=k}^{k-1} \rho_i^2(x)\Big) r_k(x-x;\alpha_k) \tag{16}$$

where $\sigma^2 = (\sigma_1^2, \dots, \sigma_k^2)$, $\alpha = (\alpha_1, \dots, \alpha_k)$, $\mathbf{B} = (\beta_1, \dots, \beta_k)$ and $\gamma = (\gamma_2, \dots, \gamma_k)$,

$$h_k'(x)^T = \left(\Big(\prod_{i=1}^{k-1} \rho_i(x)\Big) g_1^T(x), \Big(\prod_{i=2}^{k-1} \rho_i(x)\Big) g_2^T(x), \dots, \rho_{k-1} g_{k-1}^T(x), g_k^T(x)\right).$$

$\mathbf{X}_k$ is a design matrix and $\Psi_k(\mathbf{X}_k, \mathbf{X}_{k'})$ is a $(n_k \times n_{k'})$ correlation matrix. Unlike Le Gratiet (2013) and Le Gratiet & Garnier (2014) that uses the Bayesian estimation technique, we introduce a likelihood maximization approach of Forrester et al. (2007) in this analysis. Under $k=2$ setting, then equation 13 can be rewritten as

$$Y_2(x) = \rho Y_1(x) + \delta(x) \tag{17}$$

where the design matrix is now defined as

$$\mathbf{X} = \Big(\mathbf{X}_1, \mathbf{X}_2\Big)^T = (\mathbf{x}_1^{(1)}, \dots, \mathbf{x}_1^{(n_1)}, \mathbf{x}_2^{(1)}, \dots, \mathbf{x}_2^{(n_2)})^T \tag{18}$$

such that

$$\mathbf{Y} = \Big(\mathbf{Y}_1(\mathbf{X}_1), \mathbf{Y}_2(\mathbf{X}_2)\Big)^T = (Y_1(\mathbf{x}_1^{(1)}), \dots, Y_1(\mathbf{x}_1^{(n_1)}), Y_1(\mathbf{x}_2^{(1)}), \dots, Y_1(\mathbf{x}_2^{(n_2)}))^T. \tag{19}$$

The conditional distribution of the output at a new target point $\mathbf{x}^{new}$ under a universal cokriging formulation is given as

$$\Big[\mathbf{y}_2(\mathbf{x}^{new})|\mathbf{y} = \mathbf{y}_1, (\beta_1, \beta_2, \rho), (\sigma_1^2, \sigma_2^2), (\alpha_1, \alpha_2)\Big] \sim N(\mu_{Y_2}(\mathbf{x}^{new}), \mathbf{K}(\mathbf{x}^{new})) \tag{20}$$

the mean and variance functions are given respectively as

$$\widehat{\mu}_{\mathbf{y}_2}(\mathbf{x}) = h^T(\mathbf{x})\hat{B} + \mathbf{t}^T(\mathbf{x})\Sigma^{-1}(\mathbf{y} - \mathbf{H}\hat{B}) \tag{21}$$

$$\widehat{\mathbf{K}}_{\mathbf{y}_2}(x) = \hat{\rho}^2\hat{\sigma}_1^2 + \hat{\sigma}_r^2 - \mathbf{t}^T(\mathbf{x})\Sigma^{-1}\mathbf{t}(\mathbf{x}), \tag{22}$$

where

$$\mathbf{B} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \quad , \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad h' = (\rho g_1^T(\mathbf{x}), g_2^T(\mathbf{x})),$$

$$t(\mathbf{x}) = \rho\sigma_1^2\Psi_1(\mathbf{x},\mathbf{X}_1), \rho^2\sigma_1^2\Psi_1(\mathbf{x},\mathbf{X}_2) + \sigma_r^2\Psi_r(\mathbf{x},\mathbf{X}_2))^T, \tag{23}$$

and covariance matrix given as

$$\Sigma = \begin{pmatrix} \sigma_1^2\Psi_1(\mathbf{X}_1,\mathbf{X}_1) & \rho\sigma_1^2\Psi_1(\mathbf{X}_1,\mathbf{X}_2) \\ \rho\sigma_1^2\Psi_1(\mathbf{X}_2,\mathbf{X}_1) & \left(\rho^2\sigma_1^2\Psi_1(\mathbf{X}_2,\mathbf{X}_2) + \sigma_r^2\Psi_r(\mathbf{X}_2,\mathbf{X}_2)\right) \end{pmatrix}$$

$$\mathbf{H} = \begin{pmatrix} g_1^T(x_1^{(1)}) & 0 \\ \vdots & \vdots \\ g_1^T(x_{n_1}^{(1)}) & 0 \\ \rho g_1^T(x_1^{(2)}) & g_2^T(x_1^{(2)}) \\ \vdots & \vdots \\ \rho g_1^T(x_{n_2}^{(2)}) & g_2^T(x_{n_2}^{(2)}) \end{pmatrix} = \begin{pmatrix} F_1(\mathbf{X}_1) & 0 \\ \rho F_1(\mathbf{X}_2) & F_2(\mathbf{X}_2) \end{pmatrix}.$$

The next problem is how to estimate the unknown parameters $(\beta_1, \beta_r, \rho, \sigma_1^2, \sigma_r^2, \alpha_1, \alpha_r)$. We use a maximum likelihood approach of Forrester et al. (2007) and Kennedy & O'Hagan (2000). Firstly, we consider estimating the parameters $\theta_1 = (\beta_1, \sigma_1^2, \alpha_1)$ and $\theta_2 = (\beta_r, \rho, \sigma_r^2, \alpha_r)$ differently because of the conditional independence that exists between the data $Y_1(x)$ and $Y_2(x)$. Therefore, we can maximize the log-likelihood given below to estimate θ_1

$$-\frac{n_1 log(\sigma_1^2)}{2} - \frac{1}{2}log(|\Psi_1(\mathbf{X}_1,\mathbf{X}_1)|) - \frac{\left\{(\mathbf{y}_1 - F_1\beta_1)^T\Psi_1(\mathbf{X}_1,\mathbf{X}_1)^{-1}(\mathbf{y}_1 - F_1\beta_1)\right\}}{2\sigma_1^2} \tag{24}$$

where $|\Psi_1(\mathbf{X}_1,\mathbf{X}_1)|$ is the determinant of correlation matrix $\Psi_1(\mathbf{X}_1,\mathbf{X}_1)$, by taking the derivative of the equation (24) with respect to β_1 and σ_1^2 and solving for zero, the estimates $\hat{\beta}_1$ and $\hat{\sigma}_1^2$ are given respectively as

$$\hat{\beta}_1 = (F_1^T\Psi_1(\mathbf{X}_1,\mathbf{X}_1)F_1)^{-1}F_1^T\Psi_1(\mathbf{X}_1,\mathbf{X}_1)^{-1}\mathbf{y}_1$$

and $\hat{\sigma_1^2} = \frac{1}{n_1}\left[(\mathbf{y}_1 - F_1\hat{\beta}_1)^T\Psi_1(\mathbf{X}_1,\mathbf{X}_1)(\mathbf{y}_1 - F_1\hat{\beta}_1)\right]$. The alternative way of performing this computation under fully-Bayesian technique of Le Gratiet (2013) and Le Gratiet & Garnier (2014) is to marginalise the conditional $\left(f(.)|\mathbf{y}_1, \beta_1, \sigma_1^2, \alpha_1\right)$ with respect to β_1 and σ_1^2. To estimate α_1, we maximize over α_1 the concentrated likelihood given below after plugging the values of $\hat{\beta}$ and $\hat{\sigma_1^2}$ in equation (24) to give

$$-\frac{n_1 log(\hat{\sigma}_1^2)}{2} - \frac{1}{2}log(|\Psi_1(\mathbf{X}_1,\mathbf{X}_1)|) \tag{25}$$

Secondly, we describe estimation of $\theta_2 = (\beta_r, \rho, \sigma_r^2, \alpha_r)$. Let $\mathbf{r} = \mathbf{y}_2 - \rho\mathbf{y}_1(\mathbf{X}_2)$ and $F = [F_2 \quad \rho\mathbf{y}_1(\mathbf{X}_2)]$, where $\mathbf{y}_1(\mathbf{X}_2)$ are the collocated points of $\mathbf{y}_1$ and $\mathbf{y}_2$. Therefore, the log-likelihood of $\mathbf{r}|\mathbf{y}_2$ is given as

$$-\frac{n_2}{2}log(\sigma_r^2) - \frac{1}{2}log(|\Psi_r(\mathbf{X}_2,\mathbf{X}_2)|) - \frac{\left\{(\mathbf{r} - F\beta_r)^T\Psi_r(\mathbf{X}_2,\mathbf{X}_2)^{-1}(\mathbf{r} - F\beta_r)\right\}}{2\sigma_r^2} \tag{26}$$

$\hat{\beta}_r = (F^T\Psi_r(\mathbf{X}_2,\mathbf{X}_2)F)^{-1}F^T\Psi_r(\mathbf{X}_2,\mathbf{X}_2)^{-1}\mathbf{r}, \quad \hat{\sigma_r^2} = \frac{1}{n_2}\left[(\mathbf{r} - F\hat{\beta}_r)^T\Psi_r(\mathbf{X}_2,\mathbf{X}_2)(\mathbf{r} - F\hat{\beta}_r)\right]$. Again, α_r and ρ are estimated by maximized the restricted log-likelihood derived by substituting values $\hat{\beta}_r$ and $\hat{\sigma_r^2}$ in equation (26). The trend and covariance parameters α_r and ρ is computed quickly and very efficiently by using a global optimiser which is based on the extension of the efficient algorithm proposed in Park & Baek (2001) for likelihood maximization. Further details are provided in (Roustant et al., 2012).

For $k = 1$, our results is equivalent to universal kriging estimate. The derivation above can be extended to a case where $k > 2$. The mean and variance can be estimated by using equation (13). See futher detail in (Kennedy & O'Hagan, 2000).

Because of the stochastic nature of the data we analyse in this paper, we briefly describe the extension of above derivation for the noisy observations, covariance $K = \sigma^2\mathbf{C}$ is replaced by $\sigma^2\mathbf{C} + diag(\tau_1^2, \ldots, \tau_n^2)$ in equations (27,29,28) respectively, where $\tau^2 = \tau_1^2, \ldots, \tau_n^2$ are the noise variances. Kriging technique is equivalent to the Bayesian method describes in Hankin (2005) and Oakley (1999), if we assign improper uniform priors on the β. In other words, non-informative Bayesian analysis often leads to kriging predictor and variance and these estimators appear respectively as conditional mean and variance in equations (28 and 29). The following is a summary of the calculations above.

$$\Sigma = \begin{pmatrix} \sigma_1^2(\Psi_1 + I_{n_1\times n_1}\lambda_1) & \rho\sigma_1^2\left[\Psi_{12} + \begin{pmatrix} 0_{((n_1-n_2)\times n_2)} & I_{(n_1\times n_1)} \end{pmatrix}^T \lambda_1\right] \\ \rho\sigma_1^2\left[\Psi_{21} + \begin{pmatrix} 0_{(n_2\times(n_1-n_2))} & I_{(n_1\times n_1)} \end{pmatrix}\lambda_1\right] & \rho^2\sigma_1^2\left(\Psi_{1'} + I_{n_2\times n_2}\lambda_1\right) + \sigma_r^2\left(\Psi_r + I_{n_2\times n_2}\lambda_2\right) \end{pmatrix}$$

where $\Psi_1 = \Psi(\mathbf{X}_1, \mathbf{X}_1)$, $\quad \Psi_{12} = \Psi(\mathbf{X}_1, \mathbf{X}_2) = \Psi(\mathbf{X}_2, \mathbf{X}_1) \quad \Psi_{1'} = \Psi(\mathbf{X}_2, \mathbf{X}_2) \quad$ and $\Psi_r = \Psi(\mathbf{X}_2, \mathbf{X}_2)$.

$$f(\mathbf{Y}) = \frac{|\Sigma_Y|^{-\frac{1}{2}}}{(2\pi)^{\frac{mn}{2}}} \exp\left\{\frac{(\mathbf{Y} - \mathbf{HB})^T \Sigma_Y^{-1} (\mathbf{Y} - \mathbf{HB})}{2}\right\} \tag{27}$$

$$\Sigma_Y = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \ldots & \Sigma_{1m} \\ \Sigma_{21} & \Sigma_{22} & \ldots & \Sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m1} & \Sigma_{m2} & \ldots & \Sigma_{mm} \end{pmatrix}$$

where Σ_{12} is the covariance between observations of type 1 and 2, Σ_{11} is the univariate variance matrix.

$$\hat{B} = (\mathbf{H}^T \Sigma_Y^{-1} \mathbf{H})^{-1} \mathbf{H}^T \Sigma_Y^{-1} \mathbf{Y}$$

$$\mu(\mathbf{x}) = \mathbf{H}^T(x)\hat{B} + \mathbf{t}^T(\mathbf{x})\hat{\Sigma}_Y^{-1}(\mathbf{Y} - \mathbf{H}\hat{B}) \tag{28}$$

$$\mathbf{K} = \left\{\hat{\Sigma}_0 - \hat{\Sigma}_{0,mn}\hat{\Sigma}_Y^{-1}\hat{\Sigma}_{0,mn} + \mathbf{U}(\mathbf{H}^T\hat{\Sigma}_Y^{-1}\mathbf{H})^{-1}\mathbf{U}^T\right\} \quad (29)$$

$$\mathbf{U} = I_n - \hat{\Sigma}_{0,mn}\hat{\Sigma}_Y^{-1}\mathbf{H}$$

To estimate Σ_Y which must be positive definite, we assume separable covariance model of the form $\Sigma_Y = \Sigma_0 \mathbf{C}$ Σ_0 is the covariance matrix between different observation types and matrix $\mathbf{C}$ is the input space covariance function. The joint likelihood of Σ_Y and $\mathbf{B}$ is obtained and maximized

3.3 Procedure for emulating LAMMPS outputs

Here, we shall focus on the floc emulation and a single ouput from LAMMPS model. A single run of LAMMPS model consists of a simulation over many time steps which requires much computer workload and time taken. There are two different approaches to this problem. Firstly, we could emulate the simulator outputs (e.g., particles at the micro level) and use the emulator to link to the simulator at a mesoscale level for a floc. This approach is currently not practicable owing to a large number of simulation data involves, although it could be possible to perform some forms of data reduction. It is kikely that pattern decomposition might even complicate the upscaling problem.

The second approach that we adopt is to focus on the cluster of particles as a floc because of a large number of data involve and emulate their interested properties like floc size. The floc is treated as a ball of a sphere, and we estimate the diameter of a sphere that circumscribes its boundary/outline. The center of the sphere will be equivalent to the center of mass of the component particles as shown Figure (1). The detailed procedure of emulating the floc diameter will be described in this section and for biofilm in next section. In the further analysis, we will also emulate the center of the sphere to give spatial attributes to the floc characterization.

The LAMMPS model is a bit more complicated to emulate easily thus require a careful strategy. Some of the major challenges of LAMMPS emulation are the nature of the outputs produced from the model itself that make it much difficult to emulate. The LAMMPS model is expensive to evaluate i.e slow and difficult to run for a large parameter space of interest, which limits the amount of information available for emulation. The model is stochastic in nature, this introduces much randomness in the data. The model is also dynamic in nature, the data are arraged as a sequence of outputs at different time points. Finally, the model produces high-dimensional and multiple outputs which make the emulation more computationally demanding than usual Despite all these caveats, the good news is that there is a large knowledge base addressing these problems.

There is a limited number of literature that treat emulation of stochastic simulators. Earlier work of Kleijnen & Beers (2005) performs ordinary kriging emulation of detrended and standardized response $\mathbf{y}'$ from stochastic outputs. The scale response is derived by repeating the simulation k times at each design point such that $\mathbf{y}' = \frac{\bar{\mathbf{y}} - \hat{f}}{\sigma^2/\sqrt{k}}$, where $\bar{\mathbf{y}}(x_i) = \frac{\sum_{j=1}^{k} \mathbf{y}_{ij}}{k}$, $\sigma^2(x_i) = \frac{\sum_{j=1}^{k} (\mathbf{y}_{ij} - \bar{\mathbf{y}})^2}{k-1}$ and $\hat{f}$ is estimate of main signal function. This approach was extended by Bates *et al.* (2006) where an independent GP emulator is developed for both the mean response and stochastic (noise) variance. A related approach was documented in Kersting *et al.* (2007) and Bates *et al.* (1997) where an additional GP model is built to estimate the noise variance of the noise-free dataset.

In addition to develop a model for the mean response, Boukouvalas *et al.* (2009) also fit empirical log-transformed noise variance in an heteroscedastic GP modelling of a stochastic simulator. Similar to Bates *et al.* (2006) Boukouvalas *et al.* (2009); Henderson *et al.* (2012) focuses on the emulation and calibration of a stochastic computer model, implementing two independent Gaussian processes on the sample mean and log-transformed standard deviation of simulation outputs. The independent GP models use nugget parameters to account for sampling error in the data.

Our initial treatment of the stochasticity in the model is to perform multiple runs and average the key outputs which are then taken as deterministic in nature. The second approach we use is to incorporate nugget terms in form empirical variance derived from the repeated simulation data.

3.4 Dynamic emulation

Due to the nature of output data from LAMMPS model, we consider using a dynamic emulation technique. Dynamic emulation models the evolution or trajectory of random variables over some time-steps (Conti et al., 2009). Emulation of time-series data or physical processes that evolve with time which implies that model output at time t becomes an input to the model at time $t+1$. The model can be written as

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{y}_{t-1}) \tag{30}$$

Where $\mathbf{y}_{t-1}$ is the state vector at the previous time step for $t = 1, \ldots, T$, and $\mathbf{x}_t$ are the inputs at time t which includes the model parameters, forcing and initial conditions. There are two fundamental techniques for addressing dynamic emulation as discussed in (Conti et al., 2009). The multi-step and single-step emulations and there are about three different variants of the multi-step technique.

Firstly, under the multi-step emulation procedure, we can emulate a complete multi-step run of the computer model. One of the ways to proceed with this according to Conti *et al.* (2010) is to treat the problem as a multivariate output simulator and develop a multi-output emulator where the dimension of the output space is given as T. Closely related to this approach, is to build one single-output emulator that incorporates time as an additional input to the emulator such that $\mathbf{y}_t = f(\mathbf{x},t)$, where the training data for building emulator consists nT data points. The limitation of this approach is that it is inefficient in practice because the dimension of the data becomes vast which introduces additional computational difficulty.

The third variant is to emulate each time step, which produces an emulator that is specific to a particular time step, an approach that assumes independence between the time steps. This method was used in Boukouvalas *et al.* (2014) but is not suitable for our present data. Here, where are interested in the temporal correlations across the time steps, and specifically for using an emulator to scale up LAMMPS model outputs from an order of O(10^6) particles to O(10^{13}) particles, i.e., for making multiple-step ahead predictions.

Under the single-step procedure, the method uses the simpler, single step simulator and use the emulator repeatedly to generate the full-time series of the resulting predictions up to the number of desired time points. This framework reduces the dimension of the problem. We implement both methods for emulation floc equivalent diameter. Although, the single-step emulation seems much appealing for our upscaling problem considering fact that we want to capture the complete behaviour of floc diameter over a number of time steps. We describe the single-step procedure below.

3.4.1 Single-step emulation

We describe the single step function emulation here. We follow a similar procedure described in Conti et al. (2009), Bhattacharya (2007). Starting from initial run of the model at time t_0, we construct the single step emulator $\mathbf{y}_1 = f(\mathbf{x}_1,\mathbf{y}_0)$ using a GP regression in form of kriging. One of the usefulness of dynamic emulation is to make a multiple step ahead predictions using iterative technique to repeat one-step-ahead predictions until the desired number of points. We proceed sequentially, feeding back the entire output distribution from the GP model, such that at time step $t = 1$, for input $(\mathbf{x}_1,\mathbf{y}_0)$, we sample from the distribution of $f(\mathbf{y}_0,\mathbf{x}_1)$, the model output is given as

$$\tilde{\mathbf{y}}_1^{(s)} \sim N\Big(\mu^{\bullet}(\mathbf{x}_1,\mathbf{y}_0),\mathbf{K}^{\bullet}(\mathbf{x}_1,\mathbf{y}_0)\Big).$$

For the next prediction at time $t = 2$, the input data $\mathbf{x}_2$ is augmented by complete distribution $\mathbf{y}_1^{(s)}$ such that $\mathbf{X}_2 = [(\mathbf{x}_2,\tilde{\mathbf{y}}_1)]^T$, then we generate sample from the distribution of $f(\tilde{\mathbf{y}}_1^{(s)},\mathbf{x}_2)$ and denote as $\tilde{\mathbf{y}}_2^{(s)}$, note that distribution of $\tilde{\mathbf{y}}_2^{(s)}$ is no longer normally distributed. This procedure is repeated until $T-1$ steps is reached. The construction of single-step emulator is summarized below:

(i) Subsample 300 points randomly from original 1000 points and formulate a single step emulator using equation (30) such that $\mathbf{y}_1 = f(\mathbf{x},\mathbf{y}_0)$, where $\mathbf{x}$ is the new design matrix for running the LAMMPS model for the single step function, $\mathbf{x}$, as usual, include initial conditions and calibrated (constant) parameters while the corresponding output is the value of current state variable $\mathbf{y}_t$.

(ii) Perform the GP emulation in the form of kriging as described in previous chapter 2, where we use a quadratic mean and Matern covariance functions. Parameters $\theta = [\hat{\beta},\hat{\sigma^2},\hat{\alpha}]$ are estimated by MLE technique.

(iii) Compute the posterior distribution of $\Big(f(.)|\mathbf{y},\hat{\theta}\Big) \sim N\Big(\mu^{\bullet}(x_0),\mathbf{K}^{\bullet}(x_0)\Big)$ where $\mu^{\bullet}(x)$ and $\mathbf{K}^{\bullet}(x)$ are defined in equations (28, 29) respectively.

(iv) Use the emulator to simulate from $\Big(f(.)|\mathbf{y}_1,\hat{\theta}\Big)$ to obtain $\tilde{\mathbf{y}}_1^{(s)}$ and then iterate the next steps for $t = 1,\dots,T-1$ to give a full time series $\Big[\tilde{\mathbf{y}}_1^{(s)},\dots,\tilde{\mathbf{y}}_{T-1}^{(s)}\Big]$.

(v) Derive a new training data by augmenting the original data with simulated time series and rebuild the single-step emulator with the new training data given below.

$$\begin{pmatrix} \text{Original inputs} \\ \vdots \\ (\mathbf{y}_0, \mathbf{x}_1) \\ (\tilde{\mathbf{y}}_1, \mathbf{x}_2) \\ \vdots \\ (\tilde{\mathbf{y}}_{T-1}, \mathbf{x}_T) \end{pmatrix} = \begin{pmatrix} \text{Original outputs} \\ \vdots \\ \tilde{\mathbf{y}}_1^{(s)} \\ \tilde{\mathbf{y}}_2^{(s)} \\ \vdots \\ \tilde{\mathbf{y}}_T^{(s)} \end{pmatrix}.$$

(vi) Simulate $\tilde{\mathbf{y}}_{t+1}^{(s)}$ from conditional distribution $\left(f(.)|\mathbf{y}_t, \hat{\theta}\right)$.

(vii) Repeat the entire process many times to obtain $\tilde{\mathbf{Y}}^N = \left[\tilde{\mathbf{y}}_1^{(s)}, \dots, \tilde{\mathbf{y}}_{T-1}^{(s)}\right]^N$, for $s = 1, \dots, N$, where $\tilde{\mathbf{Y}}^N$ is a sample from the joint distribution of $[\mathbf{y}_1, \dots, \mathbf{y}_{T-1}]$ given the emulator training data and initial conditions and N is the number of Monte Carlo (MC) sample.

3.4.2 Normal approximations to recursive iterations

One of the limitations of above procedure is that it is highly prone to numerical problems associated with the inversion of the covariance matrix as training data is augmented. Moreover, an additional computational cost is often involved. There is a simple normal approximation to the above procedure that can prevent repeated use of single emulator thus avoiding the computational difficulties. Here, we summarize the approximation technique according to Conti et al. (2009). This new procedue is based on the assumption that augmentation of training data at each iteration step will have a relatively minimal effect provided that we use a large sample size for building our single-step emulator, in other words, additional data at each step could be discarded. In addition, since our training data for the single step emulator $\mathbf{y}_t = f(\mathbf{x}, t)$ is modelled as a GP, thus makes it difficult to derive a joint distribution for $\mathbf{y}_1, \dots, \mathbf{y}_T$ in a closed form, rather a normal approximation is proposed to estimate the marginal distribution of each $\mathbf{y}_t$ for $t = 1, \dots, T$. Suppose, the marginal distribution of $\mathbf{y}_t$ can be approximated as $\mathbf{y}_t \sim N\Big(\mu_t(.), \mathbf{K}_t(.)\Big)$, where $\mu_1 = \mu^\bullet(x_1, y_0)$ and $\mathbf{K}_1 = \mathbf{K}^\bullet\Big((x_1, y_0), (x_1, y_0)\Big)$, $\mu^\bullet(.)$ and $\mathbf{K}^\bullet(., .)$ are already defined in equations (28 and 29) respecctively.

Then we have

$$\mu_{t+1} = E\Big(\mu^\bullet(\mathbf{y}_t, x_{t+1})|f(\mathbf{y})\Big), \tag{31}$$

$$\mathbf{K}_{t+1} = E\Big(\mathbf{K}^\bullet(x_{t+1}, \mathbf{y}_t), (x_{t+1}, \mathbf{y}_t)|f(\mathbf{y})\Big) + var\Big(\mu^\bullet(\mathbf{y}_t, x_{t+1})|f(\mathbf{y})\Big). \tag{32}$$

Now, we can then estimate the two quantities above using simulation from Monte Carlo sampling to repeatedly revise the mean and variance of the single step emulator.

$$\hat{\mu}_{t+1} = \frac{1}{N}\sum_{s=1}^{N}\Big(\mu^\bullet(\tilde{\mathbf{y}}_t^{(s)}, x_{t+1})|f(\mathbf{y})\Big), \tag{33}$$

$$\widehat{\mathbf{K}}_{t+1} = \frac{1}{N}\sum_{s=1}^{N}\Big(\mathbf{K}^\bullet(x_{t+1}, \tilde{\mathbf{y}}_t^{(s)}), (x_{t+1}, y_t)|f(\mathbf{y})\Big) + \frac{1}{N}\sum_{s=1}^{N}\Big(\mu^\bullet(\tilde{\mathbf{y}}_t^{(s)}, x_{t+1})|f(\mathbf{y})\Big)^2, \tag{34}$$

where $\tilde{\mathbf{y}}_t^{(s)}$ is a sample from $N\Big(\mu_t(.), \mathbf{K}_t(.)\Big)$. This approximation technique is also related to procedure earlier described in Azman & Kocijan (2005), **?**), **?** where GP is applied to a nonlinear dynamic systems to propagate uncertainty in an iterative multiple-step-ahead predictions.

4 Results

4.1 Fractals

Fractals are of rough or fragmented geometric shape that can be subdivided in parts, each of which is (at least approximately) a reduced copy of the whole. The number, very often non-integer, often the only one measure of fractals. It measures the degree of fractal boundary fragmentation or irregularity over multiple scales Zmeskal et al. (2001). Fractal dimension of a floc is a measure of the complexity of its external shape (de Boer et al., 2000). It reflects the hydrodynamic environment that produces microbial aggregates. The fractal dimension can also be used to study the process of aggregation in wastewater treatment where the characteristics of the aggregates play a crucial role to the performance, and operational stability Amaral et al. (1997). It quantifies the particle morphology and particle density and settling velocity are a function of the fractal dimensions.

Unlike de Boer et al. (2000) that uses the relationship between the object area and perimeter to calculate the fractal dimension, we used the ration of radius of agglomerates to the mean radius of the particles as given by the formula below

$$R_a = \sqrt{\frac{\sum_{i=1}^{n} m_i d_i^2}{\sum_{i=1}^{n} m_i}} \tag{35}$$

$$R_m = \sqrt{\frac{\sum_{i=1}^{n} r_i}{n}} \tag{36}$$

$$F_D = \frac{log(R_a/R_m}{log(n)} \tag{37}$$

where F_D is a fractal dimension. For Euclidean objects such as circles or squares, $F_D = 1$, and for irregular particles such as flocs, $F_D < 1$.

Span is another important property. The span factor gives an indication of the distribution width Suppose at time step t, the LAMMPS output is written in the form

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{y}_{t-1}) \tag{38}$$

Where $\mathbf{y}_{t-1}$ the state vector at the previous time step, $\mathbf{x}_t$ are the input at time t which includes the model parameters, forcing and initial conditions as described earlier. We summarize the individual particle at microscale to a large (mesoscale) as a floc. We consider emulation of floc which is summarized by aggregating all the individual microbe at each time step. The number of particles n at each time slice varies across the design points as stated before. The number of design points at each time step is 1000 and $T = 172$ in our simulation. The total floc mass at time t is given as

$$M_t = \sum_{k=1}^{n} m_{kt}, \tag{39}$$

and center of mass $\tilde{C}$ for the floc aggregate in 3-dimension, for X direction using equation

$$\tilde{P}_{x_t} = \frac{\sum_{k=1}^{n} m_{kt} X_{kt}}{M_t}, \tag{40}$$

where M_t is the total floc mass at time t for all the species and m_{kt}'s are individual particle level mass. Replace X_{kt} in equation (40) with Y_{kt} and X_{kt} respectively to derive for other directions.

There are two different ways to derive the floc equivalent diameter namely the volume and distance techniques. Under the distance approach, the diameter of the smallest circle that circumscribes the outer edge or sketch of the floc can be obtained by computing relative distances in $X-Y-Z$ positions of each of the particle from the center of mass of the floc aggregate. The sum of the maximum of this distances and radius of the particle with the largest distance will form the radius of the outer sphere as shown in Figure (1).

Suppose at time t, the distance in euclidean three-space between any two positions, say particle p at position $P=(x_k,y_k,z_k)$ and floc center of mass at point $\tilde{P}=(x_0,y_0,z_0)$ is given as

$$d_k=\sqrt{(x_{kt}-x_0)^2+(y_{kt}-y_0)^2+(z_{kt}-z_0)^2}, \tag{41}$$

$$d_{eqv}=2(\max(d_k)+r_{k'}), \tag{42}$$

where $r_{k'}$ is the radius of particle with largest distance and x, y and z are respective directions, $k=1,\dots,n$. The second approach is to compute the total volume of the floc using the volume of each individual particle (particle is taken as a sphere).

$$d_{eqv}=\sum_{k=1}^{n}\sqrt[3]{\frac{6V_{kt}}{\pi}} \tag{43}$$

where V_{kt} volume of individual spherical particle k at time t, π is a constant and d_{eqv} is the floc equivalent diameter. The volume technique under-estimates the value of equivalent diameter.

We apply the kriging model to train the data for floc equivalent diameter d_{eqv} using a normal approximation technique of dynamic emulation. The output data for training the emulator are denoted as $\mathbf{y}=d_{eqv}$ and the corresponding input data are stored as a matrix . Some results are given below.

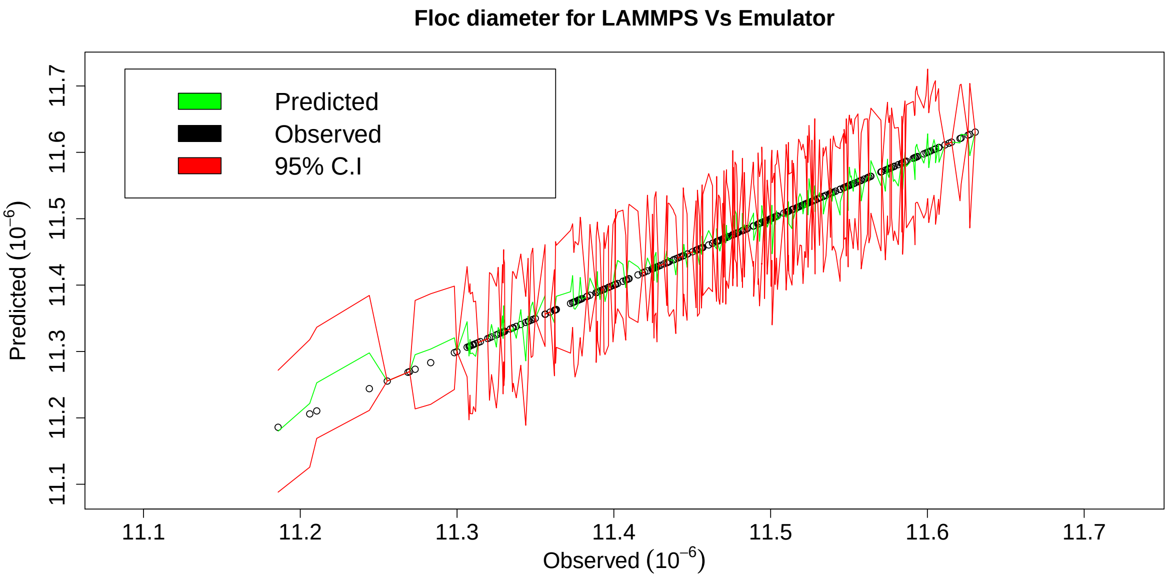


Fig. 3: Comparison between floc diameter for LAMMPS model and emulator with 95% C.I

4.2 Emulator Performance

(i) It takes LAMMPS model between (13-24 hours) to obtain 2 days simulation ensembles on 8G ram, 4-cores Linux machine.

(ii) Emulator gives results almost instantaneously (2 minute).

(iii) This is 1100-fold increase in computational efficiency.

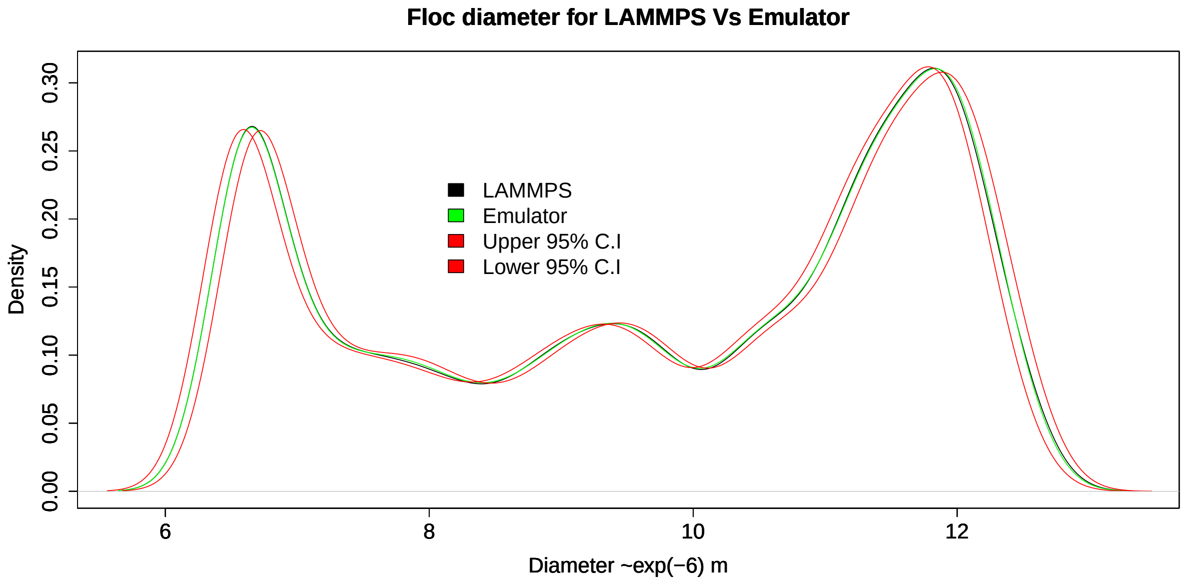
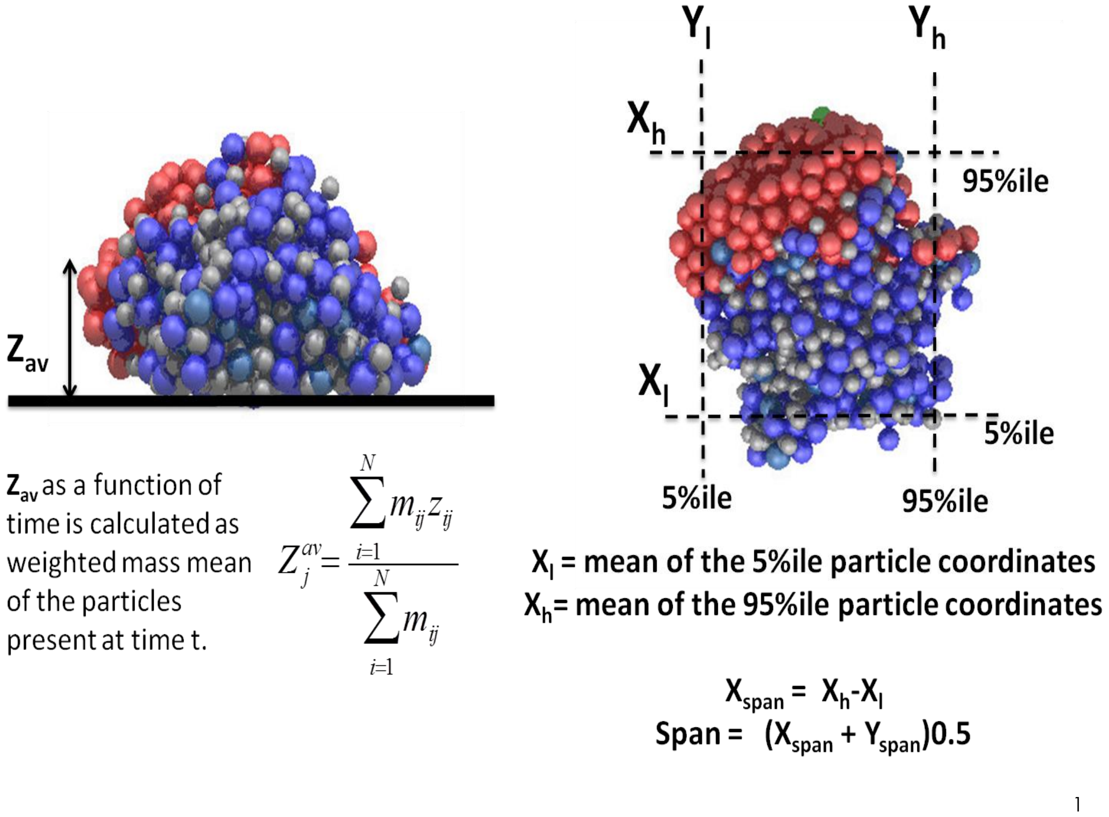


Fig. 4: Probability density function for LAMMPS model and emulator with 95% C.I

5 Biofilm emulation

We describe emulation of biofilm in this section, we characterize the biofilm as shown in the Figure 5 below. We apply the same procedure for emulating floc to the biofilm modelling.



Z_{av} as a function of time is calculated as weighted mass mean of the particles present at time t.

$$Z_j^{av} = \frac{\sum_{i=1}^{N} m_{ij} z_{ij}}{\sum_{i=1}^{N} m_{ij}}$$

X_l = mean of the 5%ile particle coordinates

X_h= mean of the 95%ile particle coordinates

X_{span} = X_h-X_l

Span = $(X_{span} + Y_{span})0.5$

Fig. 5: Biofilm characterization

6 Discussion

7 Conclusion

In this paper, we have demonstrated a means of making inference about the parameters of the emulator using a GP regression that is based upon kriging. We have presented a simple statistical method for emulating the underlying physical dynamics of response of potential floc diameter.

In modelling the residual micro scale particle, we reduced the complexity of the computation by aggregating spatially from a fine to a more coarse resolution as a floc. We assume that the aggregation will reduce the complexity and structure of the global trend component of the emulator.

References

Currin, C., Mitchell, T.J., Morris, M.D., and Ylvisaker, D. (1991). Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments. *Journal of the American Statistical Association*, $86(416), 953-963$. 5

Martin, J. D., & Simpson, T. W. (2004). On the use of kriging models to approximate deterministic computer models. *In ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, $481-492$. 5

Osio, I.G. and Amon, C.H. (1996). An Engineering Design Methodology with Multistage Bayesian Surrogate and Optimal Sampling. *Research in Engineering Design*, $8(4), 189-206$. 5

Sacks, J., Welch, W., Mitchell, T., Wynn, H. (1998). Design and analysis of computer experiments. *Statistical Science*, $4(4), 409-435$. 6, 7

Santner, T., Williams, B., Notz, W. (2003). The Design and Analysis of Computer Experiments. Springer. 3, 7

Li, R., & Sudjianto, A. (2005). Analysis of computer experiments using penalized likelihood in gaussian kriging models. *Technometrics*, $47(2), 111-120$. 5

Andrianakis, Y., & Challenor, P. G. (2009). Parameter estimation and prediction using gaussian processes. *Technical report*, MUCM Technical report 09/05, University of Southampton.

Roustant, O., Ginsbourger, D., & Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. 7, 10

Park J.S., & Baek, J. (2001). Efficient Computation of Maximum Likelihood Estimators in a Spatial Linear Model with Power Exponential Covariogram. *Computer Geosciences*, $27, 1-7$. 7, 10

Hankin, R. K. (2005). Introducing BACCO, an R package for Bayesian analysis of computer code output. *Journal of Statistical Software*, $14(16), 1-21$. 7, 10

O'Hagan, A. (2006). Bayesian Analysis of Computer Code Outputs: A Tutorial. *Reliability Engineering and System Safety*, $91, 1290-1300$. 3, 5

Conti, S., Gosling, J. P., Oakley, J. E., & O'hagan, A. (2009). Gaussian process emulation of dynamic computer codes. *Biometrika*, asp028. 11, 12, 13

Conti, S., & OHagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of statistical planning and inference*, $140(3), 640-651$. 12

Bhattacharya, S. (2007). A simulation approach to Bayesian emulation of complex dynamic computer models. iB̑ayesian Analysis, $2(4), 783-815$. 12

Azman, K., & Kocijan, J. (2005). Comprising prior knowledge in dynamic gaussian process models. *In Proceedings of the International Conference on Computer Systems and Technologies-CompSysTech*, Vol. 16(17.6). 13

Kleijnen, J. P. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, $192(3), 707-716$. 6

Martin, J. D., & Simpson, T. W. (2005). Use of kriging models to approximate deterministic computer models. *AIAA journal*, $43(4), 853-863$. 6

Kleijnen, J. P., & Mehdad, E. (2014). Multivariate versus univariate kriging metamodels for multi-response simulation models. *European Journal of Operational Research*, $236(2), 573-582$. 7

Kleijnen, J. P. (2009) Boukouvalas, A., Cornford, D., & Singer, A. (2009). Managing uncertainty in complex stochastic models: *Design and emulation of a rabies model. In 6th St. Petersburg Workshop on Simulation*, (pp. 839-841). 11

Kersting, K., Plagemann, C., Pfaff, P., & Burgard, W. (2007). Most likely heteroscedastic Gaussian process regression. *In Proceedings of the 24th international conference on Machine learning*, (pp. 393-400). ACM. 11

Kleijnen, J.P., & Van Beers, W.C. (2005). Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *European Journal of Operational Research*, $165(3), 826-$

834. 11

Bates, R. A., Kenett, R. S., Steinberg, D. M., & Wynn, H. P. (2006). Achieving robust design from computer simulations. *Quality Technology and Quantitative Management*, $3(2), 161-177$. 11

Goldberg, P. W., Williams, C. K., & Bishop, C. M. (1997). Regression with input-dependent noise: A Gaussian process treatment. *Advances in neural information processing systems*, $10, 493-499$. 11

Henderson, D. A., Boys, R. J., Krishnan, K. J., Lawless, C., & Wilkinson, D. J. (2012). Bayesian emulation and calibration of a stochastic computer model of mitochondrial DNA deletions in substantia nigra neurons. *Journal of the American Statistical Association*. 11

Boukouvalas, A., Sykes, P., Cornford, D., & Maruri-Aguilar, H. (2014). Bayesian precalibration of a large stochastic microsimulation model. *Intelligent Transportation Systems, IEEE Transactions on*, $15(3), 1337-1347$. 12

Kleijnen, J., & Mehdad, E. (2012). Kriging in multi-response simulation, including a Monte Carlo laboratory. CentER Discussion Papers Series, (2012-039). 7

Jarvis, P., Jefferson, B., & Parsons, S. A. (2005). Measuring floc structural characteristics. *Reviews in Environmental Science and Bio/Technology*, $4(1-2), 1-18$. 2

Fraser, C. E., McIntyre, N., Jackson, B. M., & Wheater, H. S. (2013). Upscaling hydrological processes and land management change impacts using a metamodeling procedure. *Water Resources Research*, $49(9), 5817-5833$. 2

Wheater, H.S., B. Reynolds, N. McIntyre, M. Marshall, B. Jackson, Z. Frogbrook, I. Solloway, O. J. Francis, and J. Chell (2008). Impacts of upland land management on flood risk: Multi-scale modelling methodology and results from the Pontbren experiment, *FRMRC Res. Rep. UR*, 16, 163 pp., Imp. Coll. & CEH Bangor, London, U.K. 2

Van Oijen, M., Thomson, A., & Ewert, F. (2009). Spatial upscaling of process-based vegetation models: An overview of common methods and a case-study for the UK. Methods, 1(3). 2

Ofiteru, I. D., Bellucci, M., Picioreanu, C., Lavric, V., & Curtis, T. P. (2014). Multi-scale modelling of bioreactorseparator system for wastewater treatment with two-dimensional activated sludge floc dynamics. *Water research*, $50, 382-395$. 2

Higdon, D., Gattiker, J., Williams, B. & Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, $103, 570-583$. 2

Kennedy, M.C. & O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of Royal Statistical Society*, series B, $63(3), 425-464$. 2

Kennedy, M. C., Anderson, C. W., Conti, S., and O'Hagan, A. (2006). Case studies in Gaussian process modelling of computer codes. *Reliability Engineering & System Safety*, $91, 1301-1309$. 2

Oakley, J. (1999). Bayesian Uncertainty Analysis For Complex Computer Codes. Ph.D. thesis, University of Sheffield. 7, 10

Oakley, J. and O'Hagan, A. (2002). Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, $89, 769-784$. 2

Oakley, J. E. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach.*Journal of Royal Statistical Society*, $66B, 751-769$. 2

Oyebamiji, O.K., Edwards, N.R., Holden, P.B., Garthwaite, P.B., Schaphoff, S., and Gerten, D. (2015). Emulating global climate change impacts on crop yields. *Statistical Modelling*, 1471082X14568248. 2, 3

Higdon, D., Gattiker, J., Williams, B. & Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, $103, 570-583$. 3

Quinonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, $6, 1939-1959$. 5

Rasmussen, C.E. and Williams, C.K.I. (2006). Gaussian Processes for Machine Learning, the MIT Press. 5

Sacks, J., Welch, W., Mitchell, T., Wynn, H. (1998). Design and analysis of computer experiments. *Statistical Science*, $4(4), 409-435$. 5

Santner, T., Williams, B., Notz, W. (2003). The Design and Analysis of Computer Experiments. Springer. 5

Wilkinson, R.D., in: Biegler *et al*. (Eds.) (2010). Large-Scale Inverse Problems and Quantification of Uncertainty. *John Wiley & Sons*, New York. 2

Young, P.C. and Ratto, M. (2011). Statistical emulation of large linear dynamic models. *Technometrics*, $53(1), 29-43$.

Le Gratiet, L. (2013). Bayesian analysis of hierarchical multifidelity codes. SIAM/ASA *Journal on Uncertainty Quantification*, $1(1), 244-269$. 2

Le Gratiet, L., & Garnier, J. (2014). Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, $4(5)$. 8, 9

Forrester, A. I., Sobester, A., & Keane, A. J. (2007). Multi-fidelity optimization via surrogate modelling. *In Proceedings of the royal society of london a: mathematical, physical and engineering sciences*, $463(2088), 3251-3269$. The Royal Society. 8, 9

Kennedy, M. C., & O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, $87(1), 1-13$. 8, 9

Kuya, Y., Takeda, K., Zhang, X., & J. Forrester, A. I. (2011). Multifidelity surrogate modeling of experimental and computational aerodynamic data sets. *AIAA journal*, $49(2), 289-298$. 8, 9, 10
8

Amaral, A. L., Alves, M. M., Mota, M., & Ferreira, E. C. (1997). Morphological characterisation of microbial aggregates by image analysis. *Proceedings of the* $9^t h$ *Pattern Recorgnition Conference*, $95-100$, Coimbra, 1997. 14

de Boer, D. H., Stone, M., & Levesque, L. M. (2000). Fractal dimensions of individual flocs and floc populations in streams. *Hydrological Processes*, $14(4), 653-667$. 14

Zmeskal, O., Vesely, M., Nezadal, M., & Buchnicek, M. (2001). Fractal analysis of image structures. *Harmonic and Fractal Image Analysis*, $3-5$. 14

Table 1: List of LAMMPS model parameters

Index	List of parameters	Value
1	KsHET	0.01
2	Ko2HET	0.81
3	Kno2HET	0.0003
4	Kno3HET	0.0003
5	Knh4AOB	0.001
6	Ko2AOB	0.0005
7	Kno2NOB	0.0013
8	Ko2NOB	0.00068
	Defining maximum growth variables	
9	MumHET	0.00006944444
10	MumAOB	0.00003472222
11	MumNOB	0.00003472222
12	etaHET	0.6
	Defining decay rates variables	
13	bHET	0.00000462962
14	bAOB	0.00000127314
15	bNOB	0.00000127314
16	bEPS	0.00000196759
17	YEPS	0.18
18	YHET	0.61
19	EPSratio	1.25
20	factor	1.5
	Initial conditions (nutrients)	
21	sub	0.08
22	no2	0.008
23	no3	1e-05
24	o2	0.01
25	nh4	0.09

Appendix 1: Model Parameters

Appendix 2: Model performance

We compute the squared differences between the actual floc equivalent diameter d_{eqv} as $\mathbf{y}$ and $\bar{\mathbf{y}}$ and also compute the squared differences between the LAMMPS values and the emulator predictions. The proportion of the variance in the LAMMPS values that is explained by the emulator is

$$\rho = 1 - \left[\frac{\sum_{t=1}^{T} \sum_{n=1}^{N} (\mathbf{y}_{tn} - \bar{\mathbf{y}}_{tn})^2}{\sum_{t=1}^{T} \sum_{n=1}^{N} (y_{tn} - \bar{y})^2} \right] \tag{44}$$

and the overall cross-validation root mean squared error (RMSE$_{CV}$) is

$$\text{RMSE}_{CV} = \left(\sum_{t=1}^{8} \sum_{n=1}^{N} \frac{(\mathbf{y}_{tn} - \bar{\mathbf{y}}^{\star}_{tn})^2}{(T \times N)} \right)^{1/2}. \tag{45}$$