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

Statistical emulation of Individual-based model simulation of microbial communities using cokriging

Oluwole K. Oyebamiji · Darren J. Wilkinson · Jayathilake Gedara et al.

Received: date / Accepted: date

Abstract The performance of credible simulations in open engineered biological frameworks is an important step for practical application of scientific knowledge to solve real-world problems and enhance our ability to make novel discoveries. Therefore, maximising our potential to explore the range of solutions at frontier level could reduce the potential risk of failure on a large scale. One primary application of this type of knowledge is in the management of wastewater treatment systems. Efficient optimisation of wastewater treatment plant focuses on aggregate outcomes of individual particle-level processes. One of the crucial aspects of engineering biology approach in wastewater treatment study is to run a high complex simulation of biological particles. This type of model can scale from one level to another and can also be used to study how to manage real systems effectively with minimal physical experimentation.

Nevertheless, simulation of open biological systems is challenging because they often involve a large number of bacteria that ranges from order 10^{12} to 10^{18} individual particles and are physically complex. The models are computationally expensive and due to computing constraints, limited set of scenarios are often possible. A simplified approach to this problem is to use a statistical approximation of the simulation ensembles derived from the complex models which will help in reducing the computational burden. Our aim in this paper is to build a cheaper surrogate of the Individual-based (IB) model simulation of biological particle. 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 emulating high-level summaries from the IB model simulation data. We use a Gaussian process regression in the form of cokriging metamodel for the emulation.

Keywords IB models · GP emulator · biofilms · floc · cokriging

1 Introduction

To identify crucial features and model water treatment plants on a large scale, there is a need to understand the interactions of microbes at fine resolution using models that could provide the best available representation of

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.

micro scale responses. The challenge then becomes how we can transfer this small-scale information to the macroscale process in a computationally efficient and sufficiently accurate way. It has been established that the macro scale characteristics of wastewater treatment plants are the consequences of microscale features of a vast number of individual particles that produce the community of such bacterial (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.

We know there is a wide separation in the spatial and temporal dimensions at which biological and physical processes occur which complicates the complete understanding of the emerging behaviour of the particle. The complex nature of the transitions from cellular level (microscale) to a group of bacteria (floc/biofilm) at mesoscale introduce a scaling problem in addition to model complexity, thus making the simulation from the IB model a computationally expensive task and a robust strategy is required to handle this issue efficiently. One useful approach for addressing this problem is the use of statistical emulators sometimes 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. Emulators offer rapid and relatively quick alternatives for projection of model outputs (Oyebamiji *et al.*, 2015). Another benefit of emulation is the provision of a measure of uncertainty associated with the projections.

There has been a significant number of research applications dealing with statistical emulation of expensive computer models. Much of the work ranges from a univariate Gaussian emulation to multi-output predictions in Conti *et al.* (2010). Similarly, Higdon *et al.* (2008) applied the Oakley & O'Hagan (2002) approach in conjunction with a PCA for basic 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.

On the other hand, the procedure for handling stochastic noise in emulation was described in Henderson *et al.* (2012) and Boukouvalas *et al.* (2014). Beside this, there is a limited amount of literature that treats emulation of stochastic simulators. Earlier work of Kleijnen & Beers (2005) performed ordinary kriging emulation of detrended and standardised response $\mathbf{y}'$ from stochastic outputs where the scale response was derived by repeating the simulation several times at each design point. 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.

On a different note, 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 while Kennedy *et al.* (2006) presented some notable examples where GP modelling applications have been implemented.

The aim of this paper is to describe how to use an emulator as an effective tool for incorporating microscale processes in a computationally efficient way into macroscale models. The focus is to train the dynamic emulator with a micro-level simulation data from IB model for the predictions of an aggregate of particles of varying species called floc and biofilms. Flocs and biofilms are aggregations of microbes mixed with an adhesive material called Extracellular Polymeric Substance (EPS). They are often difficult to measure or quantify because of their irregular size and shape. For instance, a wide range of different equivalent diameters has been used to characterise the floc size, see Jarvis *et al.* (2005) for further details. The floc plays a strategic role in understanding the processes involved in wastewater treatment plants.

In this study, we describe the procedure for emulating high summary outputs from an IB model simulation of microbial organisms based on LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator), a classical dynamical model for biological particle simulation. The emulator constructed will be further used to transfer 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 attention on how to use emulators for upscaling hydrological processes and land use management properties.

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 by spatially aggregating to produce the most relevant outputs in the form of floc and biofilms aggregates. The data compression has the benefit of suppressing or reducing some of the nonlinear response features, simplifying the construction of the emulator. Some of the most interesting properties at the mesoscale level like the size, shape, and structure of biofilms and floc are characterised, see Figure (2).

We use Gaussian process emulation in the form of multivariate kriging metamodels where output data can be decomposed into a mixture of deterministic (non-random trend) and a residual random variation. In particular, we develop dynamic emulators for the multi-outputs simulation data using cokriging. The cokriging model is formulated appropriately to filter the noise derived from replicate simulations. We describe the models and simulation data utilised for the analysis in Section 2. In Section 3, we describe the methods and emulation procedures. Section 4 provides the results of the analysis. Section 5 and 6 present the discussion and concluding comments respectively.

2 Simulation model

2.1 Individual-based Modelling of Microbial Communities

The present study attempts to model the activated sludge process (ASP) at the individual microbe level since pilot scale plants and laboratory scale experiments of WWTP are expensive, cumbersome, non-invasive and often cannot provide information at the micro-scale, which is required for operational optimisation of WWTP. The mathematical models used for ASP can be mainly divided into two general classes according to the way the biomass is represented: Continuous and discrete models. In the present work, an IB Model is developed. Figure 1 shows the typical computation domain associated with IbM of biofilms/flocs. It has three sub-domains each for biofilm/floc, mass transfer boundary layer, and bulk fluid. In the present model, three functional groups of microorganism and two inert states are considered as soft agents within the model. These are Heterotrophs-HET, Ammonia oxidizers-AOB, and Nitrite oxidizers-NOB. For the inert states, Extracellular Polymeric Substance (EPS), secreted by some heterotrophs and dead agents are also represented by soft spheres. Agents have four state variables as position, mass, radius, and type. The IB model consists of two sub-models: one deals with the growth and behaviour of individual bacteria as autonomous agents (i.e., biological processes); the other deals with the substrate and product diffusion and reaction and fluid flow (i.e., physical processes). Each cell grows by consuming the substrate and divides when a certain mass is reached. When agents grow and divide the system deviates from its mechanical equilibrium due to some residual pressure built-up in the biomass.

Depending on the net force acting on each agent, resulting from its spatial interaction with other local agents, the position of each agent is updated until the mechanical equilibrium is obtained using the Discrete Element Method (DEM). In DEM, contact, EPS adhesion, shear, and gravitational forces are considered, and the position of agents are updated by solving Newtons second law equation. For the substrates, COD, oxygen, ammonia, nitrite, and nitrate are considered. The diffusion-reaction equation governs the substrate concentrations and this transport equation is solved in a fixed Cartesian grid using a Finite Difference Method. In our model (NUFEB1.0), the traditional IB model is extended to incorporate mechanical interactions between agents. The model is implemented in LAMMPS, an open-source C^{++} molecular dynamics code (http://lammps.sandia.gov/). More details about NUFEB1.0 can be found at://github.com/nufeb/NUFEB.

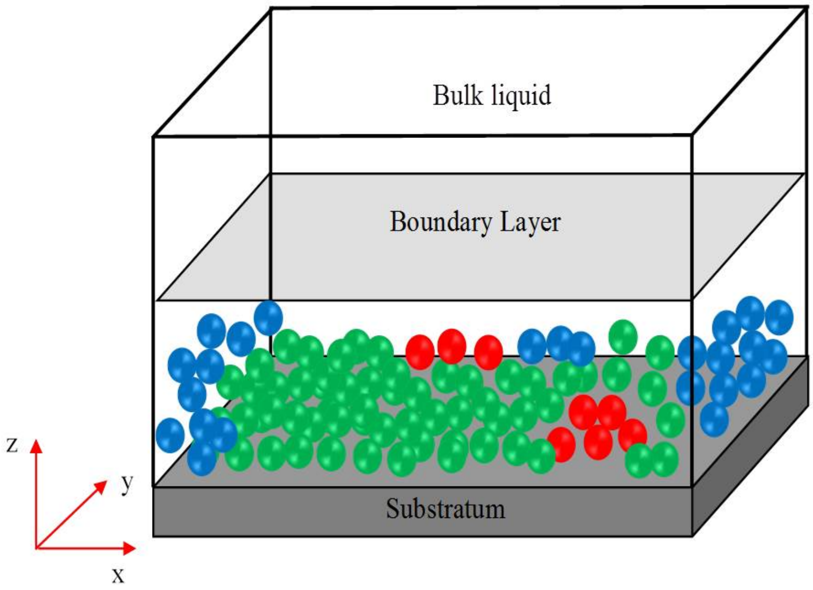


Fig. 1: Computational domain for IB model of biofilms and floc

2.2 Simulation data

This section describes the procedure for generating the parameter combinations and variables at which the LAMMPS IB model is run. We run the IB model for a small sample of input parameters which are generated using a Latin Hypercube Design (LHD). This procedure provides data for training our emulator to approximate the major outputs. The LHD technique provides a good coverage of the input space with a relatively small number of design points. We use the ”maximin” version of LHD technique that optimises samples by maximising 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, and n sample points are then drawn such that a Latin Hypercube is created.

In this study, we generate an $n \times p$ variables Latin Hypercube sample matrix with values uniformly distributed on the interval [0,1]. We then transform the generated sample to the quantile of a uniform distribution. The parameters are varied within the range of $\pm100\%$ of the standard values given in Table 2 to cover a wide variation of the computer model outputs behaviour. We limit our analysis to just $n = 300$ training points and ten replicates at each design point because of the expense of this computer model. The essence of repeated runs is to incorporate stochastic variations in our outputs.

Let the design matrix which contains the input to the LAMMPS model be denoted by $\mathbf{X} = (\theta_p^i, t, p = 1, \ldots, 27; i = 1, \ldots, 300)$, where the subscript $p = 27$ represents 27 model parameters that are varied and 5 inlet concentration variables that represent the model initial conditions (see Table 2), superscript i denote the 300 different realisations (design points) and t is the time slice in seconds at which the output data is recorded $t = 1, \ldots, 72$. The design matrix $\mathbf{X}_{300\times32}$ 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 simulator is run for about eight days (720000s simulation time). The output results are recorded at a time-step of 10000 seconds which gives about 72 different time slices.

The simulator is run for both the floc and biofilms simulations. The following outputs are produced from the simulator particle diameter, mass, position (3-dimensional) and nutrient consumption at each time step. 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 $T = 72$ 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.

2.3 Outputs for emulation

Figures (2) show a spatially distributed nature of the the floc and biofilms making the emulation of these data a large dimensional problem. We preprocess the data by measuring the aggregated characteristic on them to reduce the dimensionality of the problem. Suppose at time step t, we summarize the individual particle at microscale (particle level) to a larger scale of biofilm/floc where we measure the following characteristics.

(1) Biofilms/floc total number of particles

(2) Biofilms/floc particle composition - calculate the proportion of each species, HET, AOB,NOB, EPS and DEAD

(3) Biofilms /floc total mass - $M_t = \sum_{k=1}^{n} m_{kt}$, where M_t is the total floc mass at time t for all the species and m_{kt}'s are the individual particle level mass.

(4) Floc equivalent diameter/ biofilm height - 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 (2). 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 $d_k = \sqrt{(x_{kt} - x_0)^2 + (y_{kt} - y_0)^2 + (z_{kt} - z_0)^2}$, and $d_{t\ eqv} = 2(\max(d_k) + r_{k'})$, where $r_{k'}$ is the radius of particle with largest distance and x, y and z are respective directions, $k = 1, \ldots, n$. The second approach is to compute the total volume of the floc from the volume of each individual particle (each particle is taken as a sphere).

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

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.

(5) Biofilms/floc segregation index - The index measures the degree to which colocalized particles are genetically related to each other. Consider a particle c_{ij} in a given a population of M particles such that $i = 1, \ldots, M$, and identify related particles within a distance of 10 diameter length with the same phenotype as c_{ij}, see further details in Mitri et al. (2011). The index is given as $\sigma_t = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{N}\sum_{j=1}^{N}\rho(c_i, c_j)\right)$, where

$$\rho(c_i, c_j) = \begin{cases} 0, c_j \text{ is not the same phenotype as } c_i \\ 1, c_j \text{ is the same phenotype as } c_i \end{cases} \tag{2}$$

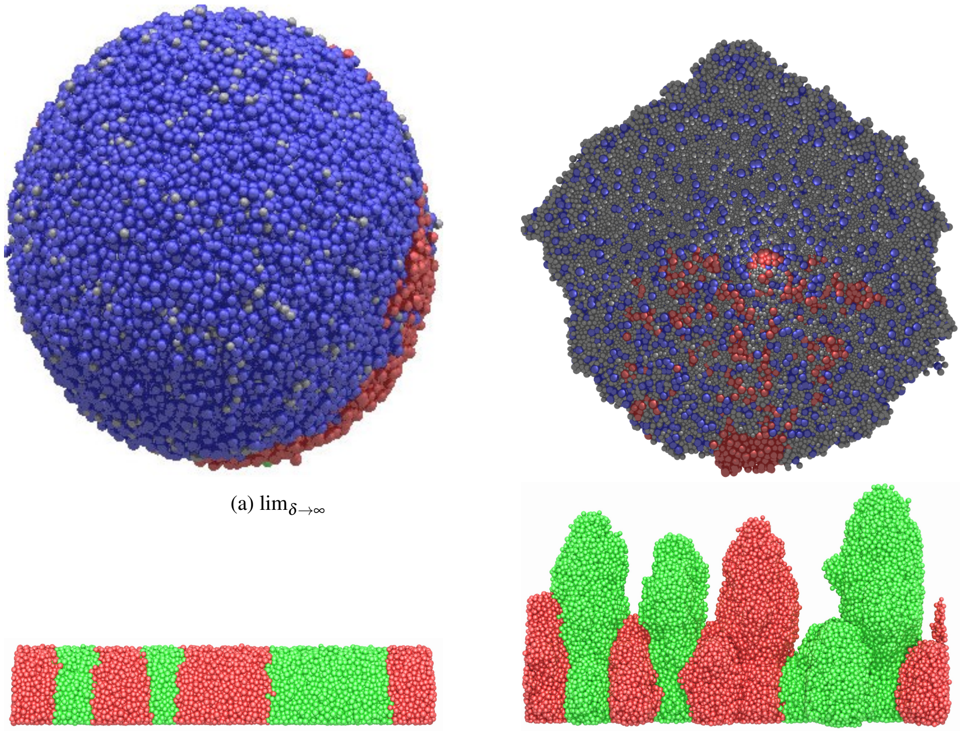
(6) Biofilm/floc fractal dimension - Fractals are of rough or fragmented geometric shape that can be subdivided in parts. Fractal dimension of a biofilm or floc is a measure of the complexity of its external shape (de Boer et al., 2000). It reflects the hydrodynamic environment that produces microbial 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 in the performance, and operational stability Amaral et al. (1997). Unlike de Boer et al. (2000) that uses the relationship between the object area and perimeter to calculate the fractal dimension, we used the ratio of radius of agglomerates to the mean radius of the particles as given by the formula below.

$$F_{Dt} = \frac{log(R_a/R_m}{log(n)} \tag{3}$$

where F_{Dt} is a fractal dimension, $R_a = \sqrt{\frac{\sum_{k=1}^{n} m_{kt} d_{kt}^2}{\sum_{k=1}^{n} m_{kt}}}$ and $R_m = \frac{\sum_{k=1}^{n} r_{kt}}{n}$, d_{kt}, r_{kt} and m_k are the particle diameter, radius and mass respectively. The parameter δ is a measure of active layer thickness of the floc and/or biofilms and is given as the ratio between the nutrient transport to the biomass and the nutrient consumption by the bacteria. In addition, δ determines the resulting shape of the floc to a certain extent, large δ values signify a large nutrient availability for the growing floc/biofilms thus decreases the heterogeneity within the floc and smooth surface floc or biofilm is formed as in Figure 2(a) while a low δ value gives a more irregularly shaped (fractal) floc is formed in Figure 2(b).

(7) Simpson diversity index - a measure of diversity of a biofilm/floc, $D_t = 1 - \frac{\sum n(n-1)}{N(N-1)}$ where n is the total number of organisms of a particular species and N is the total number of organisms of all species.

(8) Biofilm surface roughness -



(a) $\lim_{\delta \to \infty}$

(c) $\lim_{\delta \to \infty}$

(d) $\lim_{\delta \to 0}$

Fig. 2: 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. $\delta = \sqrt{\frac{S_{bulk} D Y_s}{\mu_{max} \rho L^2}}$, S_{bulk},D,Y_s, μ_{max} ,ρ and L are the bulk nutrient concentration, diffusion coefficient, yield coefficient, maximum specific growth rate, biomass density, and boundary layer thickness respectively. The top-left - smooth floc: top-right - rough floc: bottom-left - smooth biofilms: bottom-right - rough biofilms.

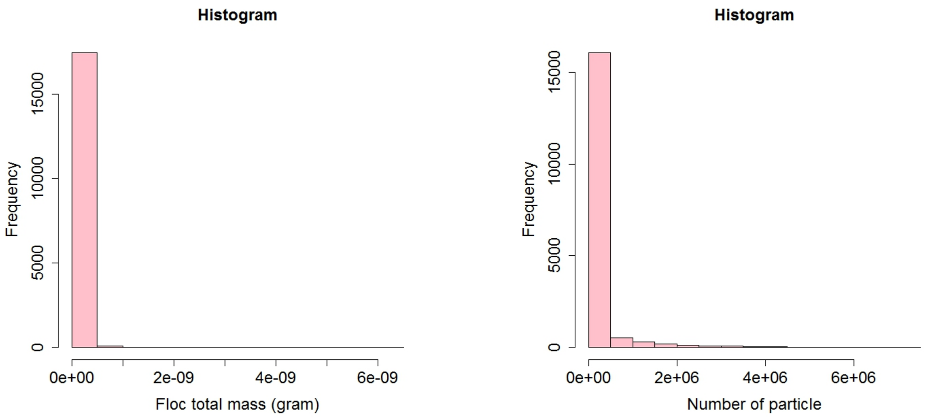


Fig. 3: Histogram of floc total mass and particle growth showing a large variation in these datasets

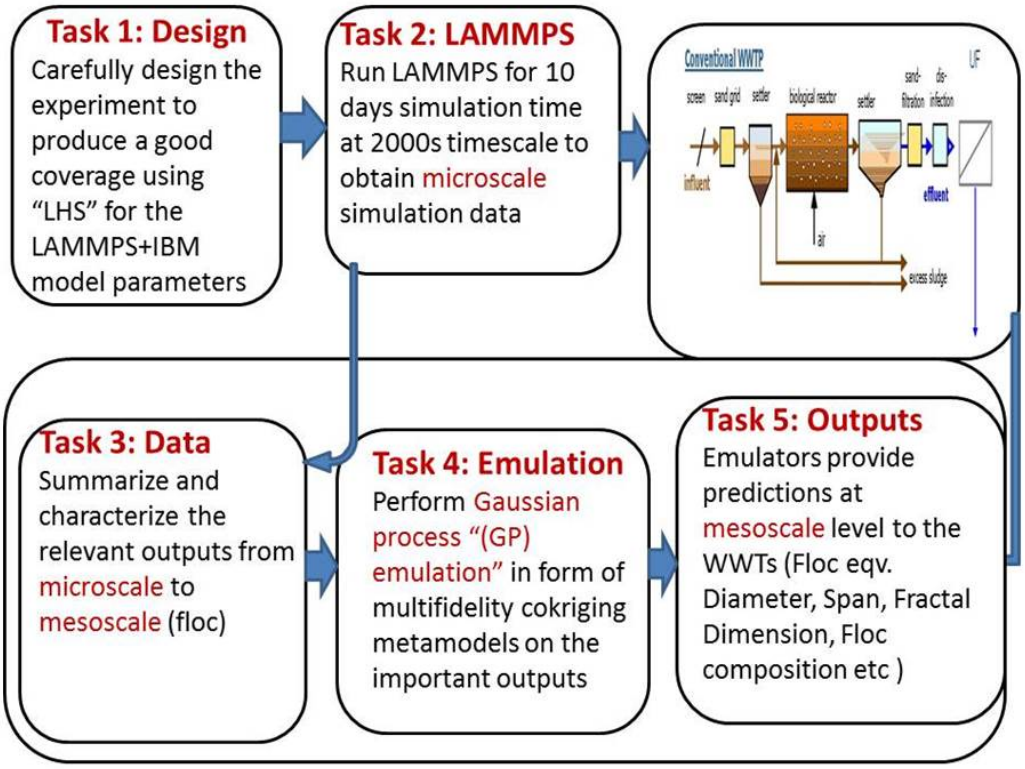

Task 1: Design
Carefully design the experiment to produce a good coverage using "LHS" for the LAMMPS+IBM model parameters
Task 2: LAMMPS
Run LAMMPS for 10 days simulation time at 2000s timescale to obtain microscale simulation data
Conventional WWTP
screen
sand grid
settler
biological reactor
settler
sand-filtration
dis-infection
UF
influent
effluent
air
excess sludge
Task 3: Data
Summarize and characterize the relevant outputs from microscale to mesoscale (floc)
Task 4: Emulation
Perform Gaussian process "(GP) emulation" in form of multifidelity cokriging metamodels on the important outputs
Task 5: Outputs
Emulators provide predictions at mesoscale level to the WWTs (Floc eqv. Diameter, Span, Fractal Dimension, Floc composition etc)

Fig. 4: 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, and a posterior distribution can be obtained. The posterior distribution is also a Gaussian process. A popular method for constructing a metamodel is the Gaussian process regression also called kriging. A major difficulty with GP modelling is the computational effort associated with dealing with huge amount of 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 $g(.)$ with a given prior distribution for $g(.)$, using the Bayesian approach and update this distribution with some data obtained from the simulator runs. We are implementing GP technique in the form of cokriging because of its wide applicability and flexibility.

3.1 Cokriging

We use cokriging which is a multivariate extension of kriging to m observation types. Cokriging has been widely applied in a various area especially in multifidelity surrogate models where there are an array of m levels of code usually from the expensive (accurate) to the less expensive (crude) simulators which are modelled jointly. It involves emulation of a function that is costly to evaluate which is enhanced by data from a cheaper simulation of the function (Forrester et al., 2007, Kuya et al., 2011). In this paper, we are assuming that the characterised outputs from the IB models have m code levels. We shall briefly describe what the kriging technique entails providing a basis for the theory of cokriging.

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., where a given scalar output $\mathbf{y}(\mathbf{x})$ 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. We discuss the universal kriging technique that we use in this paper. The model formulation is given as

$$\mathbf{y}(\mathbf{x}) = f(\mathbf{x}) + \varepsilon(\mathbf{x}) \tag{4}$$

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

$$f(\mathbf{x}) = \sum_{j=1}^{p} \beta_j h_j(x) = \mathbf{H}(x)\beta \tag{5}$$

$\beta = [\beta_1, \ldots, \beta_p]$ 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 an exponential correlation function $\mathbf{C} = \Big\{\exp(-(x-x')^T \alpha (x-x'))\Big\}$, where α is the correlation hyperparameters to be estimated from the data Sacks et al. (1989), Kleijnen (2009), Kleijnen & Simpson (2005). The best linear unbiased predictor for universal kriging model is given as

$$\mu^{\bullet}_{uk}(x) = h^T(x)\hat{\beta} + \mathbf{t}^T(x)\mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\hat{\beta}) \tag{6}$$

and variance

$$\mathbf{K}^{\bullet}_{uk} = \hat{\sigma}^2 \Big\{ C(\mathbf{x}, \mathbf{x}') - \mathbf{t}(\mathbf{x})^T \mathbf{C}^{-1} \mathbf{t}(\mathbf{x}) + \Big(h(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T \mathbf{C}^{-1} \mathbf{t}(\mathbf{x})\Big) (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \Big(h(\mathbf{x}')^T - \mathbf{t}(\mathbf{x}')^T \mathbf{C}^{-1} \mathbf{t}(\mathbf{x}')\Big)^T \Big\}. \tag{7}$$

See more details in (Sacks et al., 1989, Santner et al., 2003, Kleijnen & Mehdad, 2014). Suppose we now have m output levels of code $\mathbf{Y}(x) = (Y_1(x), \ldots, Y_m(x))$, The k^{th} output $y_k(x)$ is modelled as a Gaussian process $y_k(x) = Y_k(x)$. We use an autoregressive (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

$$Y_k(x) = \rho_{k-1}Y_{k-1}(x) + \delta_k(x) \tag{8}$$

for $k \in (2, \ldots, m)$, where $\delta_k(x)$ 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

$$\rho_{k-1}(x) = g_{k-1}^T(x)\gamma_{k-1} \tag{9}$$

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 $(Y_1(x), \ldots, Y_m(x))$ is a multivariate GP with mean

$$E[Y_k(x)|\sigma^2, \alpha, \beta, \gamma] = h_k(x)^T\beta \tag{10}$$

and covariance function

$$cov\Big[Y_k(x), Y_k(x')|\sigma^2, \alpha, \mathbf{B}, \gamma_k\Big] = \sum_{k=1}^{m} \sigma_k^2 \Big(\prod_{i=k}^{k-1} \rho_i^2(x)\Big) r_k(x-x;\alpha_k), \tag{11}$$

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

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

$\mathbf{X}_k$ 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, in this paper we follow a likelihood maximization approach of Forrester et al. (2007) and in order to simplify our approach, we assume $k = 2$ so that equation 8 can be rewritten as

$$Y_2(x) = \rho Y_1(x) + \delta(x) \tag{12}$$

where the design matrix is now re-defined as

$$\mathbf{X} = \Big(\mathbf{X}_1, \mathbf{X}_2\Big)^T = (\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_1^{(n_1)}, \mathbf{x}_2^{(1)}, \ldots, \mathbf{x}_2^{(n_2)})^T \tag{13}$$

such that

$$\mathbf{Y} = \Big(\mathbf{Y}_1(\mathbf{X}_1), \mathbf{Y}_2(\mathbf{X}_2)\Big)^T = (Y_1(\mathbf{x}_1^{(1)}), \ldots, Y_1(\mathbf{x}_1^{(n_1)}), Y_1(\mathbf{x}_2^{(1)}), \ldots, Y_1(\mathbf{x}_2^{(n_2)}))^T. \tag{14}$$

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

$$\Big[\mathbf{y}_2(\mathbf{x}^{new})|\mathbf{y} = \mathbf{y}_1, (\beta_1, \beta_2, \rho), (\sigma_1^2, \sigma_2^2), (\alpha_1, \alpha_2)\Big] \sim N(\mu_{Y_2}(\mathbf{x}^{new}), \mathbf{K}(\mathbf{x}^{new})) \tag{15}$$

the mean and variance functions are given respectively as

$$\widehat{\mu}_{\mathbf{y}_2}(\mathbf{x}) = h^T(\mathbf{x})\hat{B} + \mathbf{t}^T(\mathbf{x})\Sigma^{-1}(\mathbf{y} - \mathbf{H}\hat{B}) \tag{16}$$

$$\widehat{\mathbf{K}}_{\mathbf{y}_2}(\mathbf{x}) = \hat{\rho}^2\hat{\sigma}_1^2 + \hat{\sigma}_r^2 - \mathbf{t}^T(\mathbf{x})\Sigma^{-1}\mathbf{t}(\mathbf{x}), \tag{17}$$

where

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

$$t(\mathbf{x}) = \rho\sigma_1^2\Psi_1(\mathbf{x},\mathbf{X}_1), \rho^2\sigma_1^2\Psi_1(\mathbf{x},\mathbf{X}_2) + \sigma_r^2\Psi_r(\mathbf{x},\mathbf{X}_2))^T, \tag{18}$$

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 $(\beta_1, \beta_r, \rho, \sigma_1^2, \sigma_r^2, \alpha_1, \alpha_r)$ and incorporate stochasticity in our model formulation which we describe in the Appendix 3.

4 Procedure for emulating IB model outputs

Our emulation can be categorised into two broad groups. Emulation of the floc and biofilms. There are two different approaches to each of the problem. Firstly, we could emulate the individual particle at the micro levels and use the emulator to link the simulator output at a mesoscale level for as a floc or biofilm. 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 likely that pattern decomposition might even complicate our problem.

The second approach that we adopt in this study is to focus on the cluster of particles as flocs and biofilms because of a vast number of data involve and emulate their interested properties described in subsection 2.3. A single run of LAMMPS model consists of a simulation over many time steps which requires much computer workload and time taken. Here, we shall focus on the floc emulation and in particular we shall describe the emulation of Simpson species diversity, total mass, the number of particles, fractal dimension and equivalent diameter of the floc to simplify our approach. Emulation of other outputs will follow a similar procedure. 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 in Figure (2). The detailed procedure of emulating the floc parameters will be described in this section and for the biofilm is deferred to next section.

Some of the challenges of LAMMPS emulation are the nature of the outputs produced from the model which 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 because the data are arranged 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. The stochasticity in the model is handled by performing multiple runs and average the key outputs which are then taken as deterministic in nature. Secondly, we fit a heterogeneous cokriging model that incorporate noise in the form of empirical variance derived from the repeated simulation data.

4.1 Dynamic emulation

Due to the dynamic nature of output data from LAMMPS model, we apply a novel dynamic emulation strategy within a cokriging framework. Dynamic emulation models the evolution or trajectory of random variables over some time-steps. 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

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{y}_{t-1}), \tag{19}$$

where $\mathbf{y}_{t-1}$ is the state vector at the previous time step for $t = 1, \ldots, T$, and $\mathbf{x}_t$ (each $\mathbf{x}_t$ corresponds to design matrix $X_{300\times32}$) are the inputs at time t which includes the model parameters, forcing and initial conditions (see Table 2). We use a single-step emulation technique proposed in Conti et al. (2009). Under the single-step procedure, the method assumes that a simpler, single step emulator can be built from a dynamic computer model, and the resulting emulator can be used repeatedly to generate the full-time series of the predictions up to the number of desired time points. This framework reduces the dimension of the problem and enables us to capture the complete behaviour of characterised outputs over a number of time steps.

4.1.1 Single-step emulation

We follow a similar procedure described in Conti et al. (2009) and 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 cokriging on the five chosen outputs. 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 cokriging model, such that at time step $t = 1$, and 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 $\tilde{\mathbf{y}}_2^{(s)}$.This procedure is repeated until $T-1$ steps is reached. The construction of single-step emulator is summarized below:

(i) Sample 280 points randomly from original 300 points and formulate a single step emulator using equation (19) 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 cokriging as described in subsection 3.1, where we use linear mean and exponential covariance functions. The parameters $\hat{\theta} = (\beta_1, \beta_r, \rho, \sigma_1^2, \sigma_r^2, \alpha_1, \alpha_r)$ 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 $\mu^\bullet_{y2}(x)$ and $\mathbf{K}^\bullet_{y2}(x)$ are the cokriging predictor and variance defined in equations (A.6, A.7) 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 $t = 1, \ldots, T-1$ to give a full time series $\Big[\tilde{\mathbf{y}}_1^{(s)}, \ldots, \tilde{\mathbf{y}}_{T-1}^{(s)}\Big]$.

4.1.2 Normal approximations

One of the limitations of the single-step emulation procedure is that it is highly prone to numerical problems associated with ill-conditioned covariance matrix as training data is augmented. Moreover, an additional computational cost is often involved. Conti et al. (2009) proposed a simple normal approximation to the above procedure that we applied in this study. This approach is comparable to Azman & Kocijan (2005) technique applied on a nonlinear dynamic systems to propagate uncertainty in an iterative multiple-step-ahead predictions. Now, we can estimate the two quantities in equations A.6 and A.7 in Appendix 3 using simulation from Monte Carlo sampling to repeatedly revise the mean and variance of the single step emulator such that

$$\hat{\mu}_{t+1} = \frac{1}{N}\sum_{s=1}^{N}\Big(\mu^{\bullet}(\tilde{\mathbf{y}}_t^{(s)}, x_{t+1})|f(\mathbf{y})\Big), \tag{20}$$

$$\widehat{\mathbf{K}}_{t+1} = \frac{1}{N}\sum_{s=1}^{N}\Big(\mathbf{K}^{\bullet}(x_{t+1}, \tilde{\mathbf{y}}_t^{(s)}), (x_{t+1}, y_t)|f(\mathbf{y})\Big) + \frac{1}{N}\sum_{s=1}^{N}\Big(\mu^{\bullet}(\tilde{\mathbf{y}}_t^{(s)}, x_{t+1})|f(\mathbf{y})\Big)^2, \tag{21}$$

where $\tilde{\mathbf{y}}_t^{(s)}$ is a sample from $N\Big(\mu_t(.), \mathbf{K}_t(.)\Big)$.

5 Results

Suppose at time step t, the LAMMPS output is written in the form

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{y}_{t-1}), \tag{22}$$

where $\mathbf{y}_{t-1}$ is 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 summarise the individual particle at the microscale to a large (mesoscale) as a floc or biofilm. We consider emulation of interested properties of floc and biofilms which are summarised by aggregating the entire individual microbe at each time step. The number of particles n at each time slice varies across the design points as stated earlier. The data is subdivided into two groups. We use 280 data points to build the emulator and use remaining left-out 20 data points to test the performance of the emulator. In our training set, we are expected to have about 20160 (280 × 72) observations for each characterised output. We note that there are some points in our training set where we have incomplete time series of the simulation. The incomplete simulation occurs as a result of the large value of growth rate parameter that is responsible for making the particles grow too rapidly and filled the entire simulation box. This often generates segmentation error and makes the simulation stop earlier.

A cokriging model could not be applied directly to the entire data because of the computational difficulty of this large sample size coupled with a large number of parameters (33, including initial state variable) to be estimated. We know kriging algorithm scales cubically with the number of observations $O(N^3)$, which is not feasible for our present data. To proceed, we subsample 4500 observations randomly for each output to reduce the computational load.

Gaussian processes regression (kriging) is prone to numerical problems because of an ill-conditioned covariance matrix. This problem is more pronounced in the cokriging algorithm we used in this study. To reduce the severity of this issue, we use an exponential covariance function for each output and standardize our input data to range in $[0,1]$. This transformation will also eliminate the unit of measurement and enable us to get better parameter estimates of covariance functions.

The histograms of total mass and number of particle in a given floc are displayed in Figure 2.3 of the Introductory section. There is a significant degree of skewness and high variance in their patterns, so we log-transformed these two outputs to reduce their skewness and make the data more interpretable to meet our model assumptions. The scatter plots of some of the output data sets (not shown) show a few extreme data points which we exclude from the training data before we fit the model.

We apply the cokriging model to train the resulting data for the characterised outputs (see section 2.3) to produce a single step emulator. We then implement the single-step emulator repeatedly using equations A.6 and A.7 derived from the normal approximation until time $t = 72$ is reached. Because of the stochasticity in the simulation, we fit a heteroscedastic cokriging straightaway by incorporating the empirical variance derived from the replicates as noise in the mean response emulator. This approach will ensure joint predictions of both

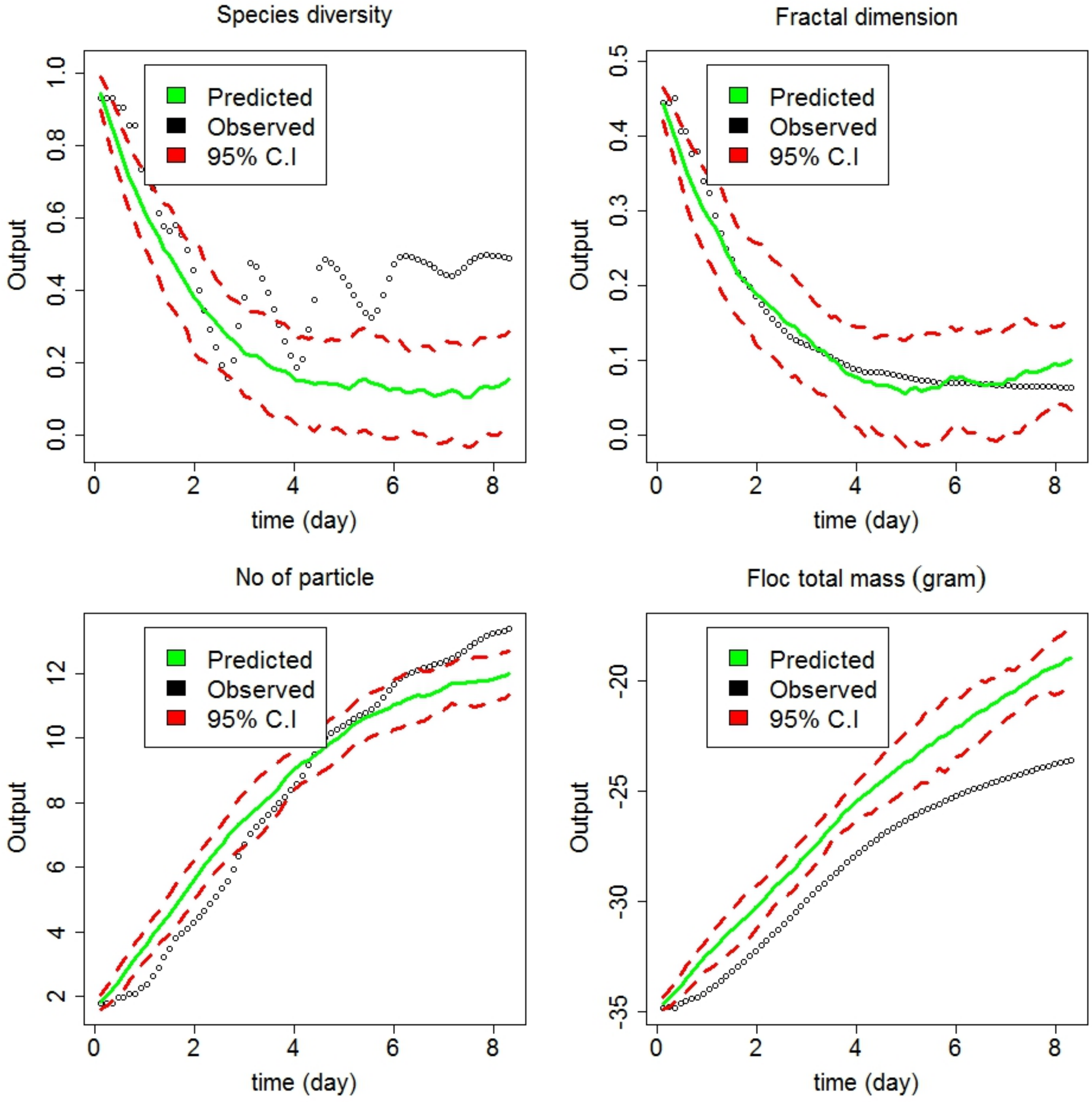


Fig. 5: Comparison of the emulator performance with simulation data for 4 major outputs from LAMMPS floc simulation (black) and their emulator predictions (green) with 95% C.I (red). Note: the bottom plots of number of particle and floc total mass are in logarithm scale

the mean and variance thus making the data estimation more robust, unlike in Henderson *et al.* (2012) and Kersting *et al.* (2007) where an independent GPs is performed on the mean and variance.

We also fit a separate kriging model for each of the outputs to compare their performance with the cokriging. To test the overall performance of the single-step emulator, we run the emulator for the entire 20 sets and compute the proportion of variance explained by each model outputs. Some results are given below.

Having built the dynamic emulator, we assess the performance of the dynamic emulator by comparing emulator predictions with the simulation data for four different characterised outputs from floc. The cross-validation results are reported in Figures (5,6,7), they are based on some of the left out 20 design points that were not used to build the emulator. Figure 5 gives time series plots showing the pattern of change in the

Table 1: Cross-validated proportion of variance ρ and root mean squared error RMSE_{CV} showing the performance of the emulators for randomly chosen design points for flocs and biofilms

	ρ	RMSE_{CV}
Floc species diversity	-0.81	
Floc eqv. diameter	0.44	
Floc fractal dimension	0.98	
Floc total number of particle	0.97	
Biofilm mean height	0.95	
Biofilm surface roughness	0.97	
Biofilm segregation index	0.82	

outputs over some days. The plots demonstrate the ability of our dynamic emulator to propagate the chosen outputs forward by applying the emulator iteratively to desired time point.

Each of the output is plotted against time. The top-left corner is the plot for the species diversity, an index which represents the abundance of different microbial species in a population (floc aggregates). We can see a gradual decreasing in trend over time for simulation data, and this is expected because the species interacts with its environment and other particles within the floc and also compete together for available nutrients. In this study, we know that the HET microbes compete with both AOB and NOB for resources and easily outcompeted and dominate at the end due to higher growth rates and anoxic reactions of HET as compared to AOB and NOB, this often result in loss of some species.

The species diversity emulator produces similar pattern to the simulation data although after day ”3” the emulator deviates from the usual trend but not significantly. This inability of the emulator to capture the entire trend could be further explained by the other plots in Figure 9 in Appendix 4. The plots indicate different temporal behaviours for the species diversity; this could potentially complicate the modelling and ability of the emulator to capture its entire emerging patterns that could degrade the emulator performance over time which suggests that it has not learnt the complete behaviour.

However, the plots for other outputs are relatively well predicted. The plot of the fractal dimension is shown in the top-right corner shows similar decreasing pattern like species diversity. The values of fractal dimension can be used as a standard for comparing the biological experiments against the theories. The emulator for the fractal dimension predicts the temporal behaviour relatively well, almost all the points lie within the 95% C.I. The predicted confidence bands remain very small. There is a general increasing trend in the bottom plots for a log-transformed number of particle and the total mass of the floc.

We note that the shape, size and structure of biofilms and floc are essential operation parameters in the management of wastewater. These characterised physical properties are quite significant in the removal efficiency of solid particles as impurities once they are formed from in wastewater treatment processes.

Figure 6 shows a comparison between the probability density function of the simulated and emulated species diversity, fractal dimension, number of particles and the total mass of the floc. The predicted density by the emulator (green) for these outputs are relatively close to the simulation data. The degree of similarity of the distributions reflects the accuracy of the dynamic emulation. The exception is species diversity, which mainly shows over predictions in some area. The distributions for each output have a single major peak. Moreover, the density of total mass and number of particle have a further minor peak making it bimodal in nature. The skewness of the density function also varies markedly with each output. The density plots for species diversity and fractal dimensions are related.

5.1 Biofilm emulation

We describe emulation of bacterial biofilms in this section where we apply the same procedure as in the flocs modelling. The plots show the assessment of our dynamic emulation approach on the floc outputs where the emulators have been used iteratively to capture the evolution of each of the characterised outputs with time. Figure (5) in the top-left is the species diversity indices for the particle, and top-right is the plot of fractal

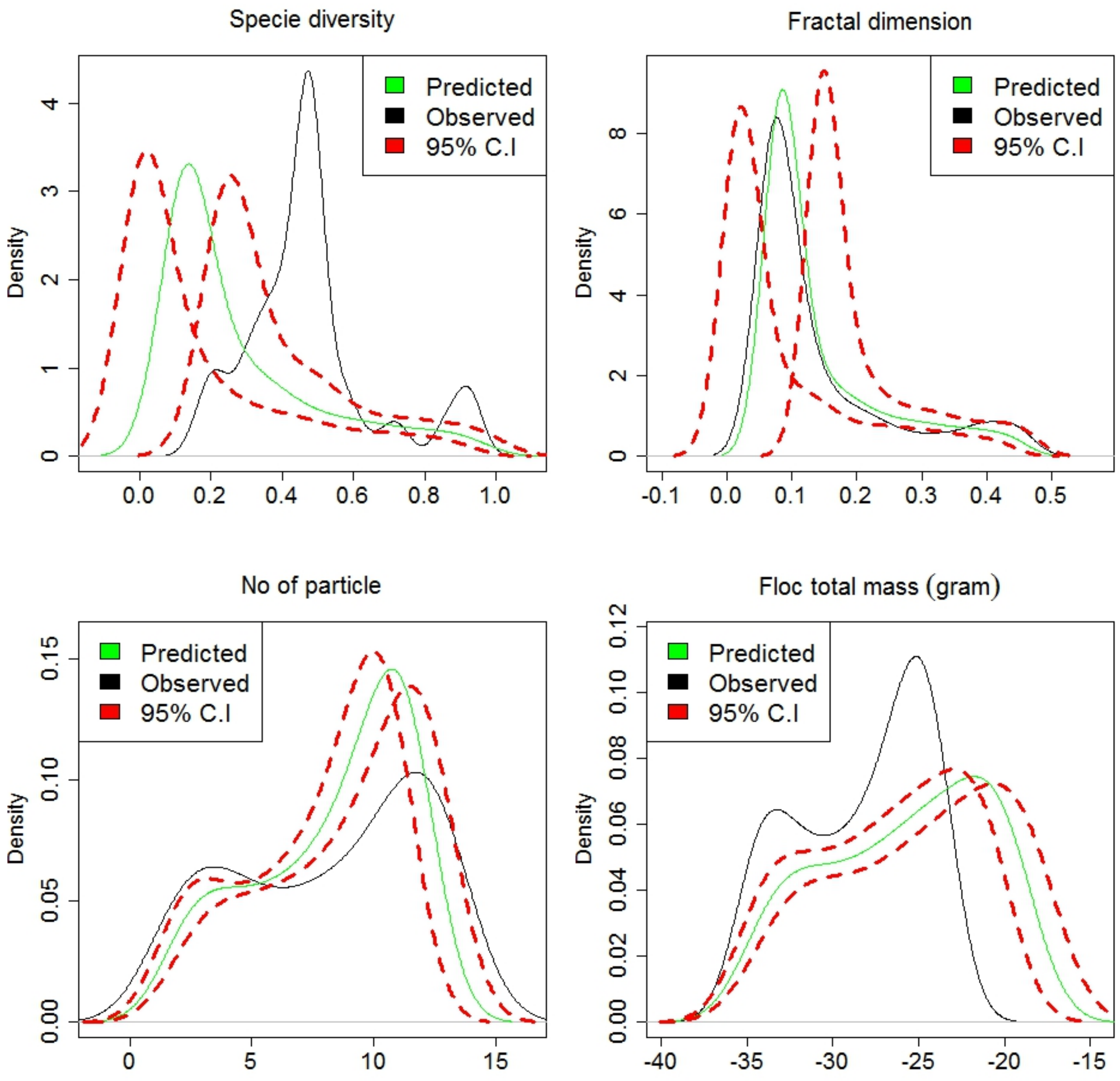


Fig. 6: Emulator crossvalidation: probability density function for 4 major outputs from LAMMPS floc simulation and their emulator predictions with 95% C.I. Note: the bottom plots of number of particle and floc total mass are in logarithm scale

dimensions often used to quantify the particle morphology. We can see that the fractal dimensions produced by the simulator and emulator have similar temporal patterns

Computing the surface roughness, average height and segregation index for biofilms of a set of large particles for a longer period as in this study is a very time-consuming. We, therefore, limit our computation for these two critical biofilm parameters to a few time point (less than two days) as shown in Figure 7.

5.1.1 Sensitivity results

To have a better understanding of the model parameters, including how their values impact the IB model outputs, a sensitivity analysis of the given parameters to measure their relative importance is performed. The results from this analysis will enhance our understanding of the most contributing input variables to output

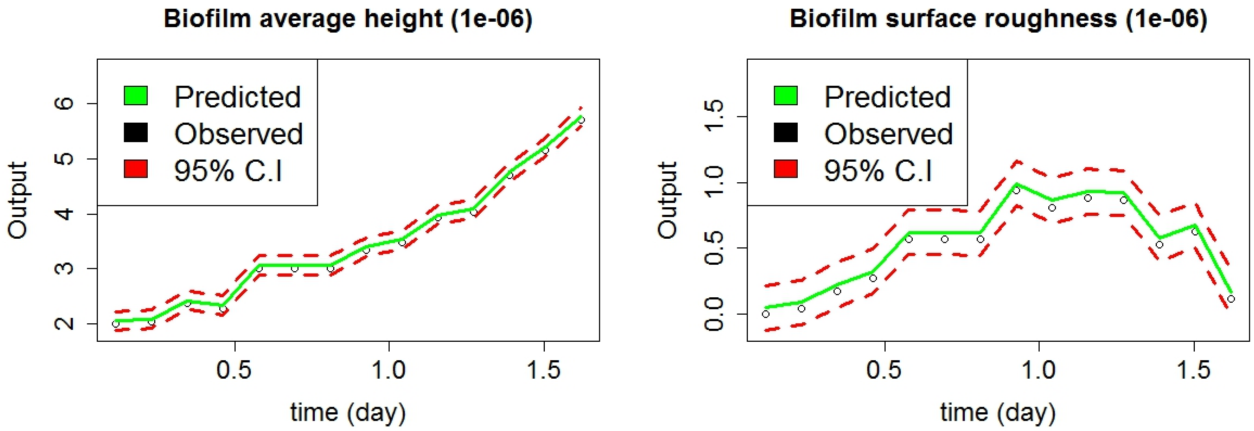


Fig. 7: Comparison of biofilm average height and surface roughness for LAMMPS model and emulator with 95% C.I

behaviour. Therefore, we can identify the most influential parameters (out of the 32 parameters we use in this study). Several techniques have been documented in the literature for performing sensitivity analysis Iooss & Lemaitre (2014), a popular technique is the Sobol global sensitivity method. This method computes the indices by decomposing the variance up to a specified order Saltelli et al. (2008). We use a Bayesian approach for global sensitivity described in Le Gratie et al. (2012) where a Sobol index is estimated based on kriging metamodels. The idea is to incorporate both the uncertainties due to the surrogate modelling and the one due to the numerical evaluations of the variances and covariances involved in the Sobol estimation.

We sample 1000 observations randomly from a uniform distribution with a range within $[0,1]$, for each of the 32 input variables in Table 1. We computed both the first order sensitivity since we have no quadratic and interaction terms in our model. We then apply a Bootstrapping to compute 95% confidence intervals on the estimated indices. This procedure was applied to all the eight outputs we consider in this study.

The combined results for both the flocs and biofilms are given below in Figure 8 for the eight outputs we examine in this paper. The sensitivity analysis clearly indicates that nutrient boundary conditions are the most critical parameters for predictions of most of the outputs. The outcome result is not surprising because these parameters regulate the distribution and transport of nutrients across the computational domain thus determine the particle growth and division. Also, particle grows when the food is readily available. The substrate boundary concentration ”s” with corresponding values 0.26, 0.024, 0.093, 0.043, 0.84, 0.28, 0.88 and 0.40 for the floc total mass, floc equivalent diameter, floc fractal dimension, floc total number of particle, floc species diversity, biofilm average height, biofilm surface roughness and biofilm segregation index respectively, is the dominant variable. The rest are oxygen boundary concentration ”o2”, nitrate boundary concentration ”no3” and ammonia boundary concentration ”nh4”.The nitrite boundary concentration ”no2” is the least sensitive among the nutrient boundary conditions. The next three most sensitive parameters apart from the nutrient concentrations are the HET growth related parameters namely maximum specific growth rate for HET ”MumHET”, reduction factor in anoxic conditions ”etaHET” and carbon source affinity for HET ”KsHET”

6 Discussion

This paper has shown that cokriging technique can be effectively applied to model a dynamic simulation model while incorporating the noise in the form of empirical variances derived from the replicate observations. We observe that the performance of the single step emulators using it iteratively to capture the underlying dynamics of output behaviours of IB model depend hugely on the initial conditions (initial state variables at time t_0) and sample size we take when applying the normal approximation. For the results presented in

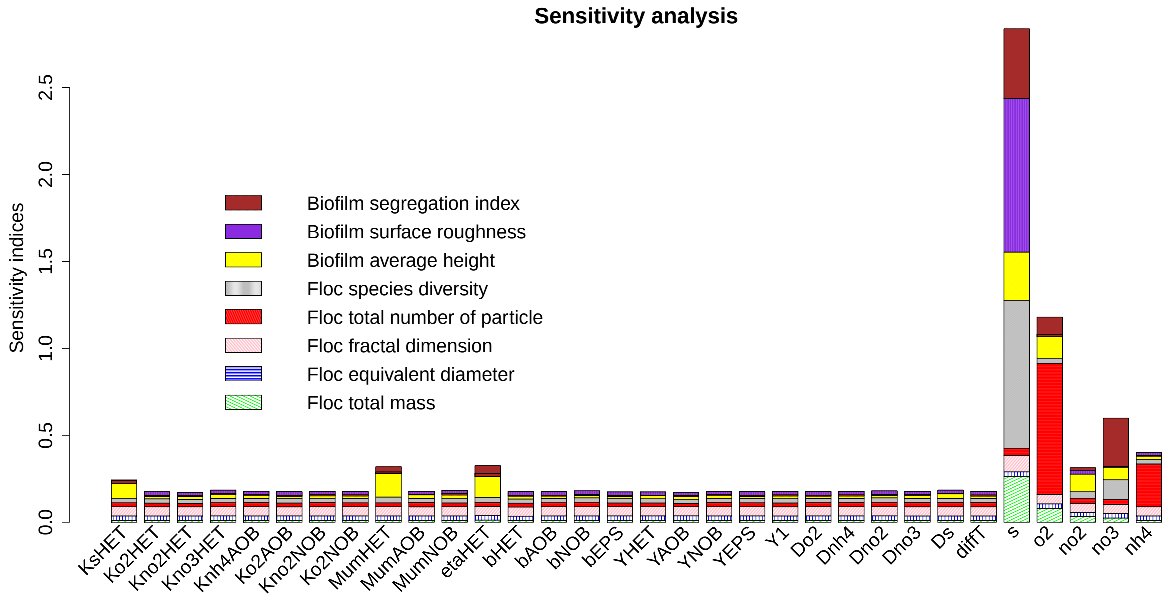


Fig. 8: Barplots showing the sensitivity indices for the eight characterized outputs from LAMMPS model.

this paper, we restrict to just 100 samples because of the expense of running the emulator repeatedly. Another critical issue is the evaluation of ill-conditioned covariance parameters which we encounter so often in this analysis because of closely spaced design points. We have reduced the severity of the problem by using an exponential covariance function. We also standardise our input data to range in $[0,1]$. This transformation also helps to eliminate the effect of measurement units and enable us to get better parameter estimates of covariance functions. Other related challenges we encounter during the emulation are. Overall, the performance of these single-step emulators applying to propagate the state variable forward with time is relatively good.

However, we can also 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 and thus not appropriate for the simulation study in this paper. Another alternative is to emulate each time step, which produces an emulator that is peculiar 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, we are interested in the dynamic behaviours over temporal space, and specifically for using an emulator for making multiple-step ahead predictions.

7 Conclusion

In this paper, we demonstrate a new concept of making inference about the parameters of the emulator using a GP regression that is based upon multifidelity cokriging technique. This concept relies on the assumption that characterised outputs from the IB model can form a different level of codes. Stochasticity in the simulation is treated by incorporating noise in the form of empirical variances derived from the replicates. Our approach combines the two-stage technique proposed in O'Hagan (2006) and Oyebamiji *et al.* (2015) as a single step. We have presented a simple statistical method for emulating the underlying physical dynamics of the major characterised outputs of the IB model simulations of microbial growth. In modelling our microscale simulation data as flocs and bacteria biofilms, we reduced the complexity of the computation by aggregating spatially from a fine (individual microbes) to a more coarse resolution as flocs and biofilms. We assume that the aggregation will reduce the complexity and structure of the global trend component of the emulator.

These emulators are much faster to run compared to the simulation model. The IB model simulation implemented within LAMMPS is computationally expensive to run, while these emulators give results almost instantaneously. Under different parameter combinations, it takes an average of between $8-10$ hours to simulate the growth of the particles for about eight days at 20s timestep on Linux cluster machine. Apart from the computation time requires for fitting the single-step emulator, it takes < 2 min to apply the emulator iteratively to generate the corresponding trajectories of the characterised outputs. This shows approximately 240-fold increase in computational efficient.

Acknowledgements The work has been supported by Newcastle University under the Frontiers in Engineering Biology (NUFEB) project.

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*, $86(416), 953-963$. 8

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*, $481-492$. 8

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

Sacks, J., Welch, W., Mitchell, T., Wynn, H. (1998). Design and analysis of computer experiments. *Statistical Science*, $4(4), 409-435$. 8, 9

Santner, T., Williams, B., Notz, W. (2003). The Design and Analysis of Computer Experiments. Springer. 4, 9

Li, R., & Sudjianto, A. (2005). Analysis of computer experiments using penalized likelihood in gaussian kriging models. *Technometrics*, $47(2), 111-120$. 8

Roustant, O., Ginsbourger, D., & Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. 22

Park J.S., & Baek, J. (2001). Efficient Computation of Maximum Likelihood Estimators in a Spatial Linear Model with Power Exponential Covariogram. *Computer Geosciences*, $27, 1-7$. 22

O'Hagan, A. (2006). Bayesian Analysis of Computer Code Outputs: A Tutorial. *Reliability Engineering and System Safety*, $91, 1290-1300$. 8, 17

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

Conti, S., & OHagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of statistical planning and inference*, $140(3), 640-651$. 2, 17

Bhattacharya, S. (2007). A simulation approach to Bayesian emulation of complex dynamic computer models. i$\widehat{\text{B}}$ayesian Analysis, $2(4), 783-815$. 11

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). 11

Kleijnen, J. P. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, $192(3), 707-716$. 8

Martin, J. D., & Simpson, T. W. (2005). Use of kriging models to approximate deterministic computer models. *AIAA journal*, $43(4), 853-863$. 8

Kleijnen, J. P., & Mehdad, E. (2014). Multivariate versus univariate kriging metamodels for multi-response simulation models. *European Journal of Operational Research*, $236(2), 573-582$. 9

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. 2, 13

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*, $165(3), 826-834$. 2

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

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*, $10, 493-499$. 2

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*. 2, 13

Boukouvalas, A., Sykes, P., Cornford, D., & Maruri-Aguilar, H. (2014). Bayesian precalibration of a large stochastic microsimulation model. *Intelligent Transportation Systems, IEEE Transactions on*, $15(3), 1337-1347$. 2, 17

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. 3

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., 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. 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, 17

Quinonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, $6, 1939-1959$. 7

Rasmussen, C.E. and Williams, C.K.I. (2006). Gaussian Processes for Machine Learning, the MIT Press. 7

Sacks, J., Welch, W., Mitchell, T., Wynn, H. (1998). Design and analysis of computer experiments. *Statistical Science*, $4(4), 409-435$. 8

Santner, T., Williams, B., Notz, W. (2003). The Design and Analysis of Computer Experiments. Springer. 8

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*, $4(5)$. 9, 22

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. 9, 22

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, 22

Kuya, Y., Takeda, K., Zhang, X., & J. Forrester, A. I. (2011). Multifidelity surrogate modeling of experimental and computational aerodynamic data sets. *AIAA journal*, $49(2), 289-298$. 9, 22
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. 5

de Boer, D. H., Stone, M., & Levesque, L. M. (2000). Fractal dimensions of individual flocs and floc populations in streams. *Hydrological Processes*, $14(4), 653-667$. 5

Mitri, S., Xavier, J. B., & Foster, K. R. (2011). Social evolution in multispecies biofilms. *Proceedings of the National Academy of Sciences*, $108(2), 10839-10846$. 5

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Salsana, M. & Tarantola, S. (2008). Global sensitivity analysis - *The primer. Wiley*. 16

Iooss, B. & Lemaitre, P. (2014). A review on global sensitivity analysis methods. 16

Le Gratiet, L., Cannamela, C. & Iooss, B. (2013). A Bayesian approach for global sensitivity analysis of (multi-fidelity) computer codes. 16

Appendix 1: Model parameters

Table 2: List of IB model parameters

Index	Parameters	Values
	Affinity variables	
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
	Maximum growth variables	
9	MumHET	0.00006944444
10	MumAOB	0.0000088
11	MumNOB	0.000009375
12	etaHET	0.6
	Decay rates variables	
13	bHET	0.00000462962
14	bAOB	0.00000127314
15	bNOB	0.00000127314
16	bEPS	0.00000196759
	Yield coefficient variables	
17	YHET	0.61
18	YAOB	0.33
19	YNOB	0.083
20	YEPS	0.18
	Diffusion coefficient variables	
21	Do2	0.000000002
22	Dnh4	0.0000000014
23	Dno2	0.0000000012
24	Dno3	0.0000000012
25	Ds	0.0000000005
	Critical diameter of death	
26	deadDia	0.0000008
27	factor	1.5
	Inlet concentrations (nutrients)	
1	sub	0.008
2	no2	0.0001
3	no3	0.0008
4	o2	0.0008
5	nh4	0.0009

Appendix 2: Model performance

We compute the squared differences between the actual floc and biofilms characterized outputs. For instance, for 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 model that is explained by the emulator is given as

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

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

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

Appendix 3: MLE of cokriging parameters

We use a maximum likelihood approach of Forrester et al. (2007) and Kennedy & O'Hagan (2000) because of its computational efficiency. The MLE procedure is divided into two categories. 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

$$-\frac{n_1 log(\sigma_1^2)}{2} - \frac{1}{2} log(|\Psi_1(\mathbf{X}_1, \mathbf{X}_1)|) - \frac{\left\{ (\mathbf{y}_1 - F_1\beta_1)^T \Psi_1(\mathbf{X}_1, \mathbf{X}_1)^{-1} (\mathbf{y}_1 - F_1\beta_1) \right\}}{2\sigma_1^2} \tag{A.3}$$

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 (A.3) 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 (A.3) to give

$$-\frac{n_1 log(\hat{\sigma}_1^2)}{2} - \frac{1}{2} log(|\Psi_1(\mathbf{X}_1, \mathbf{X}_1)|) \tag{A.4}$$

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

$$-\frac{n_2}{2} log(\sigma_r^2) - \frac{1}{2} log(|\Psi_r(\mathbf{X}_2, \mathbf{X}_2)|) - \frac{\left\{ (\mathbf{r} - F\beta_r)^T \Psi_r(\mathbf{X}_2, \mathbf{X}_2)^{-1} (\mathbf{r} - F\beta_r) \right\}}{2\sigma_r^2} \tag{A.5}$$

$\hat{\beta}_r = (F^T \Psi_r(\mathbf{X}_2, \mathbf{X}_2) F)^{-1} F^T \Psi_r(\mathbf{X}_2, \mathbf{X}_2)^{-1} \mathbf{r}$, $\quad \hat{\sigma_r^2} = \frac{1}{n_2} \left[(\mathbf{r} - F\hat{\beta}_r)^T \Psi_r(\mathbf{X}_2, \mathbf{X}_2)(\mathbf{r} - F\hat{\beta}_r) \right]$. 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 (A.5). The trend and covariance parameters α_r and ρ is computed 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). The derivation above can be extended to a case where $k > 2$ and for $k = 1$, our results is equivalent to universal kriging estimate.

Due to 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 (6, 7) respectively for the kriging predictor, where $\tau^2 = \tau_1^2, \ldots, \tau_n^2$ are the noise variances. And for the cokriging model, the covariance matrix Σ in subsection 3.1 is rewritten as

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

where $\mathbf{I}$ and $\underline{0}$ are matrices of ones and zeroes respectively, $\Psi_1 = \Psi(\mathbf{X}_1, \mathbf{X}_1)$, $\Psi_{12} = \Psi(\mathbf{X}_1, \mathbf{X}_2) = \Psi(\mathbf{X}_2, \mathbf{X}_1)$, $\Psi_{1'} = \Psi(\mathbf{X}_2, \mathbf{X}_2)$ and $\Psi_r = \Psi(\mathbf{X}_2, \mathbf{X}_2)$ and parameters $\lambda = (\lambda_1, \lambda_2)$ are estimated along with other parameters using modified likelihood function. The modified cokriging predictor with the variance are given respectively as

$$\widehat{\mu}_{\mathbf{y}_2}(\mathbf{x}) = h^T(\mathbf{x})\hat{B} + \mathbf{t}^T(\mathbf{x})(\Sigma + \lambda)^{-1}(\mathbf{y} - \mathbf{H}\hat{B}) \tag{A.6}$$

$$\widehat{\mathbf{K}}_{\mathbf{y}_2}(x) = \hat{\rho}^2 \hat{\sigma}_1^2 + \hat{\sigma}_r^2 - \mathbf{t}^T(\mathbf{x})(\Sigma + \lambda)^{-1} \mathbf{t}(\mathbf{x}), \tag{A.7}$$

Appendix 4: Additional figures

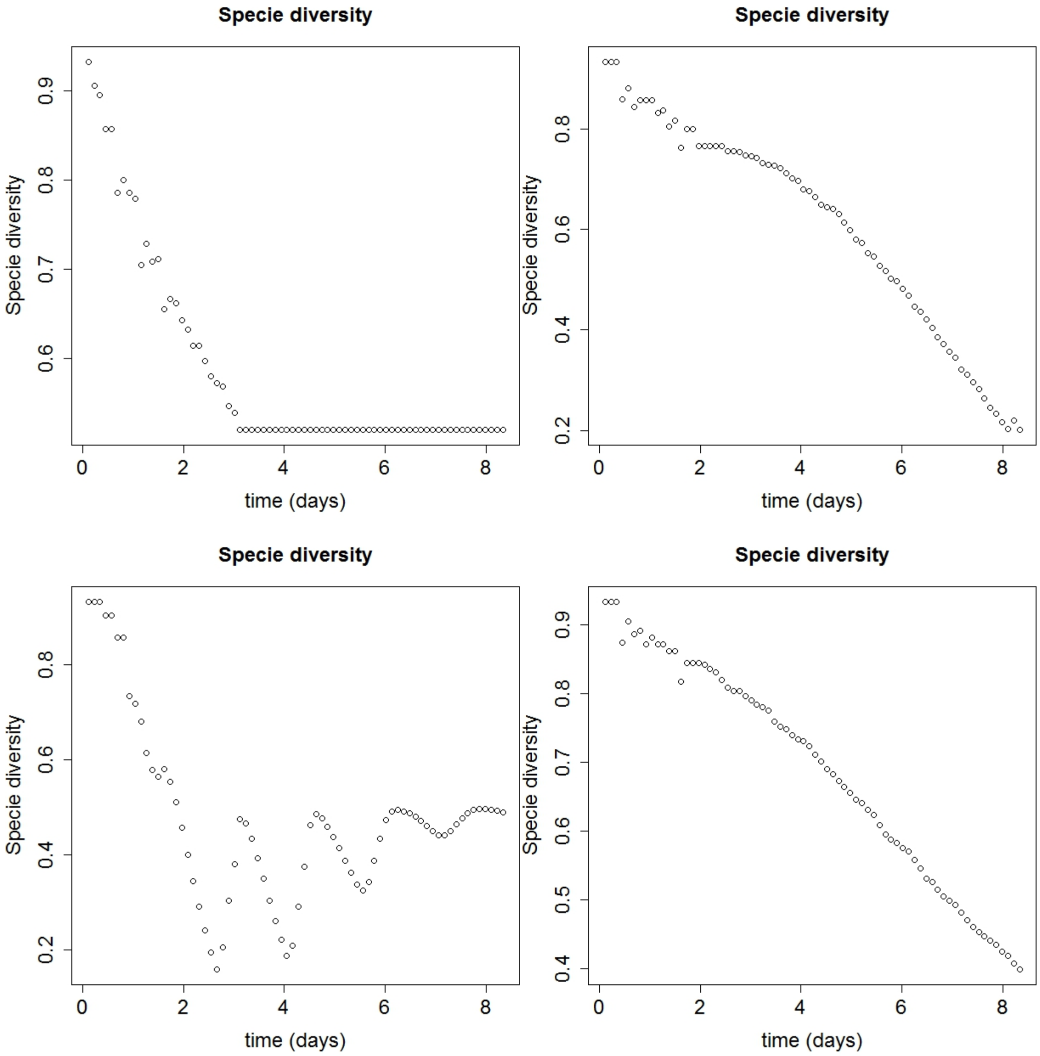


Fig. 9: Diverging behaviour of floc species diversity indices over time for some randomly selected design points

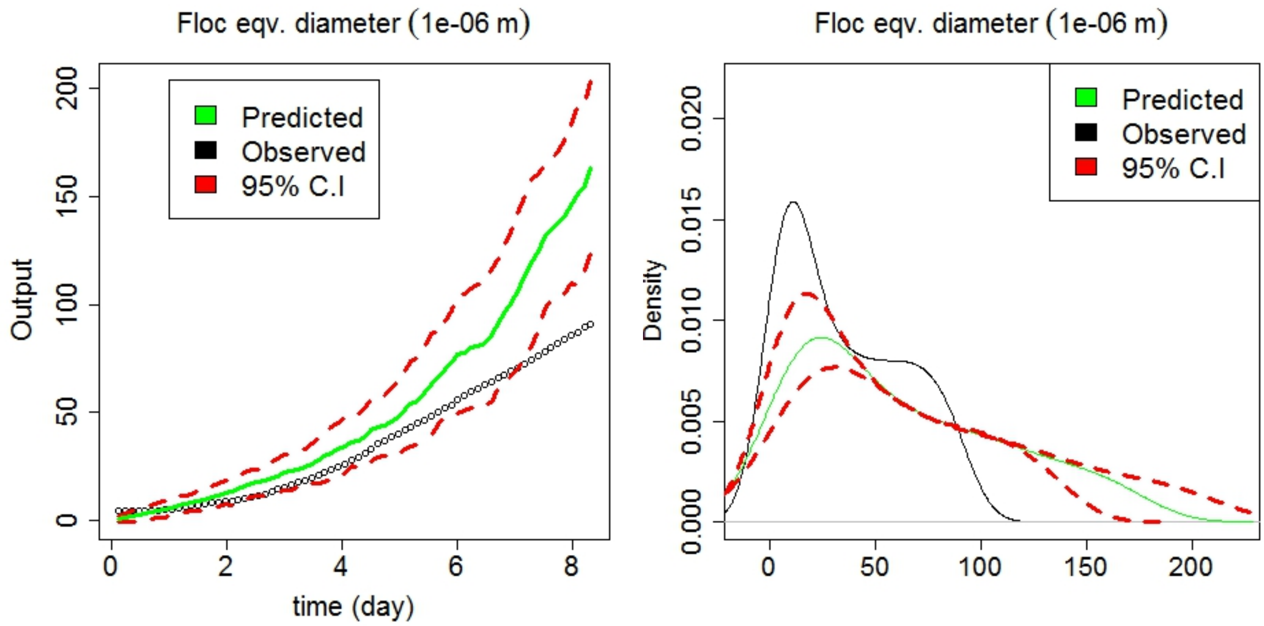


Fig. 10: Comparison of time series plot and density function of floc equivalent diameter for LAMMPS model and emulator with 95% C.I