Journal of Computational Science 22 (2017) 69–84

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Gaussian process emulation of an individual-based model simulation of microbial communities

O.K. Oyebamiji[a,*], D.J. Wilkinson[a], P.G. Jayathilake[b], T.P. Curtis[c], S.P. Rushton[d], B. Li[e], P. Gupta[d]

[a] School of Mathematics & Statistics, Newcastle University, United Kingdom
[b] Department of Mechanical & Systems Engineering, Newcastle University, United Kingdom
[c] School of Civil Engineering and Geosciences, Newcastle University, United Kingdom
[d] School of Biology, Newcastle University, United Kingdom
[e] School of Computing Science, Newcastle University, United Kingdom

ARTICLE INFO

Article history:
Received 20 January 2017
Received in revised form 3 August 2017
Accepted 8 August 2017
Available online 14 August 2017

Keywords:
Individual-based models
Multivariate Gaussian process
Biofilms
Flocs
Emulator

ABSTRACT

The ability to make credible simulations of open engineered biological systems is an important step towards the application of scientific knowledge to solve real-world problems in this challenging, complex engineering domain. An important application of this type of knowledge is in the design and management of wastewater treatment systems. One of the crucial aspects of an engineering biology approach to wastewater treatment study is the ability to run a simulation of complex biological communities. However, the simulation of open biological systems is challenging because they often involve a large number of bacteria that ranges from order 10^{12} (a baby's microbiome) to 10^{18} (a wastewater treatment plant) individual particles, and are physically complex. Since the models are computationally expensive, and due to computing constraints, the consideration of only a limited set of scenarios is often possible. A simplified approach to this problem is to use a statistical approximation of the simulation ensembles derived from the complex models at a fine scale which will help in reducing the computational burden. Our aim in this paper is to build a cheaper surrogate of an individual-based (IB) model simulation of microbial communities. 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 a strategy for emulating high-level summaries from the IB model simulation data. We use a Gaussian process regression model for the emulation. Under cross-validation, the percentage of variance explained for the univariate emulator ranges from 83–99% and 87–99% for the multivariate emulators, and for both biofilms and floc. Our emulators show an approximately 220-fold increase in computational efficiency. The sensitivity analyses indicated that substrate nutrient concentration for nitrate, carbon, nitrite and oxygen as well as the maximum growth rate for heterotrophic bacteria are the most important parameters for the predictions. We observe that the performance of the single step emulator depends hugely on the initial conditions and sample size taken for the normal approximation. We believe that the development of an emulator for an IB model is of strategic importance for using microscale understanding to enable macroscale problem solving.

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 provide the best possible representation of micro-scale responses. The challenge then becomes how we can transfer this small-scale information to the engineered 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 populations [37]. In other words, the properties of cells or particles at a micro level dictate the behaviour of a wastewater treatment plant at a macro scale.

* Corresponding author.
E-mail address: Oluwole.Oyebamiji@newcastle.ac.uk (O.K. Oyebamiji).

http://dx.doi.org/10.1016/j.jocs.2017.08.006

We know that 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 emergent behaviour of the system. The scale transition for modelling biofilms and flocs in this study ranges from micro- to meso- to macro-scales (although, we only consider micro-meso-scales in this study) (see Fig. 1) for details. This multiscale approach was used in this study for passing aggregate information from one level to the other. The complex nature of the transitions from cellular level (microscale) to a group of bacteria (floc/biofilm) at mesoscale introduces a scaling problem in addition to model complexity, thus making the simulation from the micro model a computationally expensive task. A robust strategy is required to handle this issue efficiently.

One useful approach for addressing this problem is via 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 which are computationally much faster to run. Emulators offer rapid and relatively quick alternatives for projection of model outputs [41,42]. A further benefit of emulation is the provision of a measure of uncertainty associated with the projections.

There have been a significant number of research applications dealing with the statistical emulation of expensive computer models. This ranges from a univariate Gaussian process emulation to multi-output predictions [8]. Similarly, [35] developed a Bayesian framework for the uncertainty analysis for the distribution of unknown input. In particular, [35] used a univariate Gaussian process for emulating computationally expensive simulator outputs with uncertain inputs. [19] extended the univariate GP approach in [35] to a multivariate GP and combined this with a principal component analysis (PCA) for calibrating high dimensional outputs from a computationally demanding computer model against the field data from an experiment. The experimental data was used to constrain uncertainty in the calibration parameters. The PCA reduces the dimension of the problem and computation time required for obtaining posterior distributions from Bayesian inference.

Another application of this sort of modelling is to separate stochastic from deterministic variations, the procedure for handling stochastic noise in emulation was described in [17] and [6]. However, there is a limited amount of literature that treats the emulation of stochastic simulators. Earlier work of [26] 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 [4] where an independent GP emulator is developed for both the mean response and stochastic (noise) variance. A related approach was documented in [24] and [15] where an additional GP model was built to estimate the noise variance of the noise-free dataset.

On a different note, [58] described the behaviour of large linear dynamic models that used 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. [36] described a Bayesian method for quantification of uncertainty in complex computer models while [23] 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 micro-level simulation data from an individual-based (IB) model for the predictions of an aggregate of particles, of varying species, called floc and biofilms. Biofilms are the aggregated microbial communities attached to surfaces. Flocs are aggregated microbial communities suspended in water. Their characteristic size is around 500 μm. The morphological features depend on the growth conditions. For example, high nutrient conditions may promote a smooth surface while a rough surface structure is more likely to emerge at low nutrient concentration. We have modelled their biological and chemical functions as listed in the supporting document.

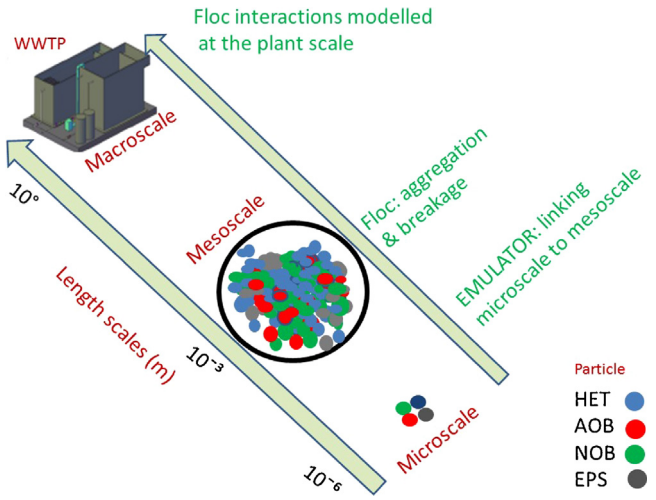


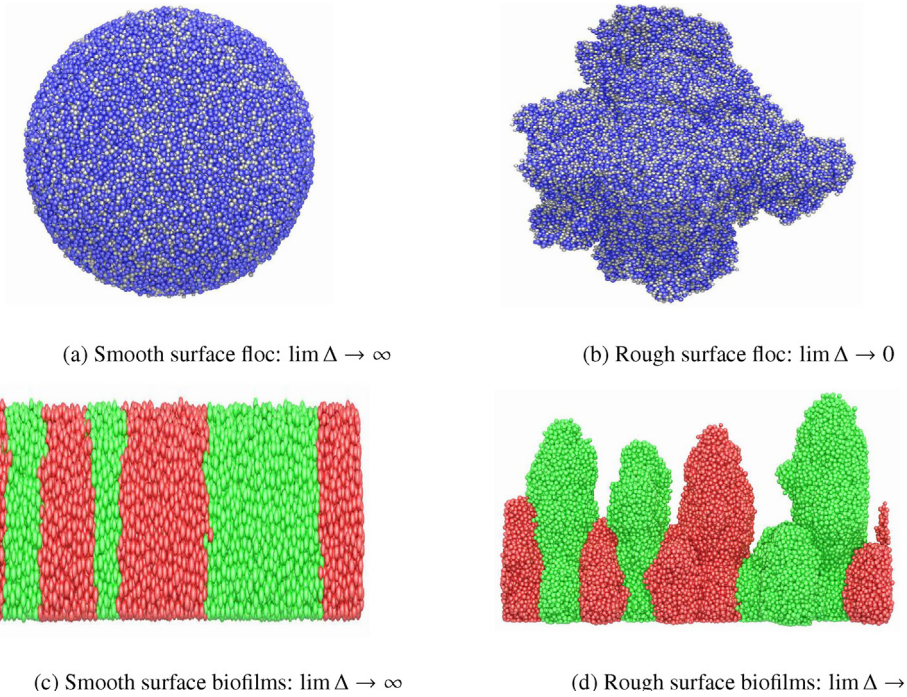
Fig. 1. Schematic of different length scales for multiscale modelling of an activated sludge process based WWTP. The scale transition from a bacterium or cellular level (microscale – <micrometer size) to the floc and biofilm aggregates (mesoscale – millimetre size) to the macroscopic bulk WWTP operation as well as floc and biofilm interactions (macroscale – metre size). The emulator is linking the microscopic (bacterium/cell) to the mesoscopic (biofilm/floc) and, ultimately, to the macroscopic bulk operational parameters.

The flocs and biofilms are mixed with an adhesive material called extracellular polymeric substance (EPS). The EPS is a class of organic macromolecules such as polysaccharide, proteins, nucleic acids, lipids and other polymeric compounds which are found in the intracellular space of organic aggregates [57]. We do not model each component of EPS. In our microscale simulations, EPS particles represent the collection of different substances of EPS. The flocs and biofilms 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 [21] 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 summary outputs from an IB model simulation of microbial organisms based on large-scale atomic/molecular massively parallel simulator (LAMMPS), a classical dynamical model for particle simulation [46]. The emulator constructed will be further used to transfer information to macro-level processes of wastewater treatment plants. [54] earlier reviewed some of the popular techniques for upscaling complex problems while [13] and [56] 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 flocs and biofilm 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 flocs are characterised, see Fig. 2.

We use Gaussian process emulation (or kriging metamodels) where output data can be decomposed into a mixture of deterministic (non-random trend) and a residual random variation. In



(a) Smooth surface floc: $\lim \Delta \to \infty$ (b) Rough surface floc: $\lim \Delta \to 0$

(c) Smooth surface biofilms: $\lim \Delta \to \infty$ (d) Rough surface biofilms: $\lim \Delta \to 0$

Fig. 2. Transformation of microscale particles to floc at the mesoscale for a particular time. There are five different particle species namely HET (blue), AOB (red), NOB (green), DEAD (black) and EPS (grey) each represented by different colour. Different shapes occur as result of the effect of microbial growth rate and nutrient concentration on the biofilms and flocs. The non-dimensional parameter $\Delta = \sqrt{(S_{bulk}DY_s)/(\mu_{\max}\rho L^2)}$ represents the ratio between the maximal nutrient transport to the biofilm and the maximal nutrient consumption by the bacteria. See Section 2.2 for further details.

particular, we develop dynamic emulators for the multi-outputs simulation data. The GP model is formulated appropriately to include a non-zero nugget term 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 study. Sections 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 wastewater treatment plants (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 (discrete). Fig. 3 shows the typical computation domain associated with IB models of biofilms. It has three sub-domains: 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. The microorganisms are heterotrophs (HET) which consume organic carbon source and oxygen, ammonia oxidizing bacteria (AOB) which convert ammonia and oxygen to nitrite, and nitrite oxidizing bacteria (NOB) which use nitrite and oxygen to produce nitrate. For the inert states, extracellular polymeric substance (EPS), secreted by some heterotrophs and dead agents are also represented by soft spheres (labelled DEAD). Agents have four state variables: position, mass, radius, and type. The IB model consists of two sub-models: one deals with the

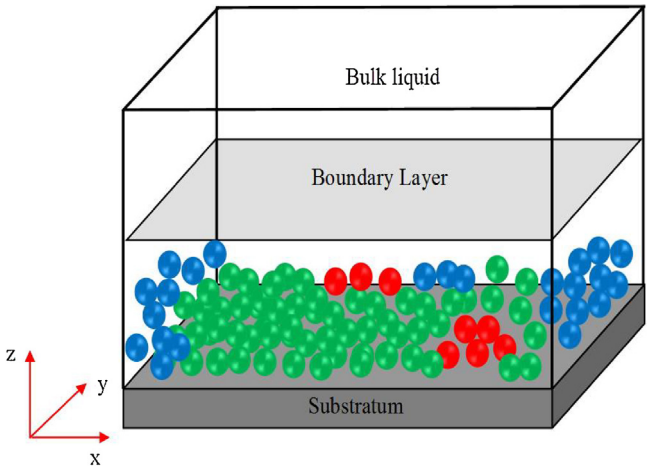


Fig. 3. A typical computational domain for IB model of biofilms. The microbes functional groups are HET (blue), AOB (red), NOB (green). (For interpretation of the references to color in this legend, the reader is referred to the web version of the article.)

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 split, 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 the DEM, contact, EPS adhesion, shear, and gravitational forces are considered, and the position of agents are updated by solving Newton's second law equation. For

the substrates, chemical oxygen demand (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 work, the traditional IB model is extended to incorporate mechanical interactions between agents. See more details about the biological and chemical kinetics in the supporting documents and [22]. The model is implemented in LAMMPS, an open-source C^{++} molecular dynamics code (http://lammps.sandia.gov/) [46]. More details about the NUFEB 1.1 version of the model that we emulated can be found at https://github.com/nufeb/NUFEB/releases.

2.2. Simulation data

We ran 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 the LHD technique that optimises samples by maximising the minimum distance between design points [52]. 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.

We generated 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 ±50% 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 five 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 $\mathbf{X} = (\theta_p^i, p = 1, \ldots, 32; i = 1, \ldots, 300)$; where the subscript p represents 27 model parameters that are varied and 5 nutrient concentration variables fixed at their nominal values stated in Table 2. The superscript i denotes the 300 different realisations (design points) and t is the time slice in seconds at which the output data is recorded, $t = 1, \ldots, 62$. The design matrix $\mathbf{X}_{300\times32}$ denotes the input values at which the LAMMPS model is run for every combination of x_i (which is a point in $\mathbf{X}$, where x_i represents ith row of $\mathbf{X}$). The simulator is run for six days to capture sufficient emergent behaviour. The spatial outputs from the simulation are recorded at a time-step of 8250 s to reduce the size of the data, which gives about 62 different time slices.

The simulator was run for both the flocs and biofilms simulations. The following ten outputs are produced from the simulator at each time step: particle diameter, mass, position (3-dimensional) and nutrient consumption variables for S, C, NH_4, NO_3 and NO_2. The spatial outputs at each time point are denoted as a matrix $\mathbf{Y}_{k\times10}$ and k is the total number of particles at each time step. The number of particles k at each time slice varied over time and, in particular, increased with time, as is expected.

2.3. Outputs for emulation

Fig. 2 illustrates the spatially distributed nature of flocs and biofilms, making the emulation of these data a high dimensional problem. We preprocess the data by measuring aggregated characteristics on them to reduce the dimensionality of the problem. Suppose at time step t, we summarize the individual particles at the microscale (particle level) to a larger scale of biofilms and flocs where we measure the following characteristics. These morphological characteristics are essential factors in the design and performance of wastewater reactors.

(1) Floc equivalent diameter (metre) – The diameter of the smallest circle that circumscribes the outer edge or sketch of the floc can be obtained by computing the total volume of the floc from the volume of each particle (the individual 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 $d_{t,eqv}$ is the floc equivalent diameter at time t.

(2) Floc fractal dimension (dimensionless) – Fractals are of rough or fragmented geometric shape that can be subdivided into self-similar parts. The fractal dimension of a floc is a measure of the complexity of its external shape [11]. 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 [3]. Unlike [11], which 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

$$F_{Dt} = \frac{\log(R_a/R_m)}{\log(n)}, \tag{2}$$

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.

(3) Floc total number of particles (dimensionless) – $N_t = \sum_{k=1}^{n} N_{kt}$, where N_{kt} represents number of each species, HET, AOB, NOB, EPS and DEAD that are present.

(4) Floc total mass (kg) – $M_t = \sum_{k=1}^{n} m_{kt}$, 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.

(5) Biofilm average height (metre) – The biofilms are partitioned into several smaller blocks. Each sub-block has dimension $d^{max} \times d^{max} \times d^{max}$. We compute the Euclidean distances between the center of each particle and the lattice blocks along the baseline (plane $z = 0$) to identify the occupied blocks. We, therefore, marked as "occupied" every block with one or more particle centers contained within it while the others are marked as "vacant". The height $h_t(x, y)$ of the biofilm above each base block is defined as the maximum of the particle z-values of the occupied blocks. The biofilm mean height at time t is then given as

$$\bar{h}(t) = \frac{1}{L_x L_y} \int_i \int_j h_t(x, y) dx dy,$$

where $L_x = L_y = 10$ are the number of blocks.

(6) Biofilms surface roughness (metre) – It is one of the key quantitative descriptors of biofilms structure. It measures the magnitude of variability in height over the surface structure, i.e. the depth of biofilm irregularities. It determines the rate of diffusion of nutrients into the biofilms. The smaller the values of these indices, the smoother the biofilm surface while large val-

ues indicate very rough surface [16,44,45]. The biofilm surface roughness at time t is given as

$$s(t) = \left(\frac{1}{L_x L_y} \int_i \int_j [h_t(x,y) - \bar{h}(t)]^2 dxdy \right)^{1/2} \quad (3)$$

(7) Biofilms segregation indices (dimensionless) – These indices measure 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 [33]. 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

$$\rho(c_i, c_j) = \begin{cases} 0, c_j \text{is not the same phenotype as} c_i \\ 1, c_j \text{is the same phenotype as} c_i \end{cases} \quad (4)$$

(8) Biofilms Simpson diversity indices (dimensionless). These indices measure diversity of biofilms and flocs, $D_t = 1 - \frac{\sum n(n-1)}{N(N-1)}$ where n is the total number of organisms of a particular species and N is the total number of organisms of all species.

Let $\Delta = \sqrt{(S_{bulk} D Y_s)/(\mu_{\max} \rho L^2)}$, where 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. Fig. 2 shows different shapes and structures of simulated microbes that form the biofilms and flocs based on various parameter settings. The resulting biofilms and flocs shapes are regulated by the potential value of the Δ parameter which in turn depends on six different parameters as defined above. The parameter Δ is a measure of an active layer thickness of the floc and biofilms and is given as the ratio between the nutrient transport to the biomass and the nutrient consumption by the bacteria. The value of Δ determines the resulting shapes of the flocs and biofilms, large Δ values signify a high nutrient availability for the growing flocs and biofilms thus decreases the heterogeneity within the floc or biofilm which give rise to flocs and biofilms that are compact and smooth in structure; see Fig. 2(a). A high substrate transfer rate will cause the nutrient concentration to penetrate more deeply into the biofilms. This would allow a high uniform microbial growth rate within the flocs and biofilms while a low Δ value or a decrease in nutrient transport rate produces rough and irregularly shaped flocs and biofilms; see Fig. 2(b). The varying structural patterns clearly show the diversity of the effects we are emulating. It is important for us to capture and incorporate these various emergent behaviours into our emulator formulation.

The histograms of the log-transformed outputs we consider are illustrated in Figs. 4 and 5 showing the data structure. The essence of the transformation is to reduce their skewness and make the data more interpretable to meet our emulator model assumptions.

The histograms of floc equivalent diameter and number of particles in Fig. 4 are relatively similar, symmetric and bimodal in nature with the presence of two major peaks except for the fractal dimension which is roughly right-skewed. The histogram of the total mass also has a single major peak and a minor peak making it a bimodal distribution. Fig. 5, unlike the floc, the histograms of biofilms average height and species diversity indices are left- and right-skewed respectively with a single major peak while that of surface roughness and segregation indices are non-symmetric with no distinct shape. Most data points in species diversity indices lie between -1 and 0. Extreme data points are present in this dataset as it shows a significant degree of skewness and high variance.

Fig. 6 is the schematic diagram summarizing the key emulation stages. It shows the procedures involved in the emulation of the characterized outputs from the data preparation to the mesoscale data modelling.

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 (the parameters of the prior). 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 Gaussian process regression, also called kriging. A major difficulty with GP modelling is the computational effort associated with dealing with a 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 [49] and [48]. GP emulation is based on Bayesian updating and experimental design of computer experiments for predicting model outputs at test input points [50,52]. A GP emulator assumes that a simulator output is an unknown function $g(.)$ with a given prior distribution for $g(.)$, updated using data obtained from the simulator runs. We are implementing GP emulation for predictions in this paper because of its wide applicability and flexibility.

3.1. Gaussian process (GP)

Multivariate GP or kriging has been widely applied in various areas, especially in multifidelity surrogate models where there are an array of k 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 [12,28]. We shall briefly describe what the univariate GP or kriging technique entails to introduce the theory of multivariate GP. 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 models of expensive computer codes [10,31,39,30]. The two stage techniques described in [38] 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 mean function $f(\mathbf{x})$ of a Gaussian process usually denotes its trend. The trend could be modelled as a constant in ordinary/simple kriging or as an nth order polynomial in universal kriging. Here, we use the universal kriging technique. The model formulation is given as

$$\mathbf{y}(\mathbf{x}) = f(\mathbf{x}) + \boldsymbol{\varepsilon}(\mathbf{x}), \quad (5)$$

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

$$f(\mathbf{x}) = \sum_{j=1}^{m} \beta_j h_j(x) = \mathbf{H}(x)\boldsymbol{\beta}, \quad (6)$$

$\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]$ is a $(m \times 1)$ vector of unknown regression coefficients and $\mathbf{H}(x) = [h_1(x), \ldots, h_m(x)]^T$ is a $(n \times m)$ matrix of regression functions, $\varepsilon(\mathbf{x})$ is a stochastic Gaussian process with mean zero and characterized by its covariance function $cov(\varepsilon(\mathbf{x}), \varepsilon(\mathbf{x}')) = \sigma^2 cor(\mathbf{x}, \mathbf{x}')$, where σ^2 denotes the variance of $\varepsilon(\mathbf{x})$ also called process variance and $\mathbf{A}$ is a $(n \times n)$ positive definite matrix of correlations at the

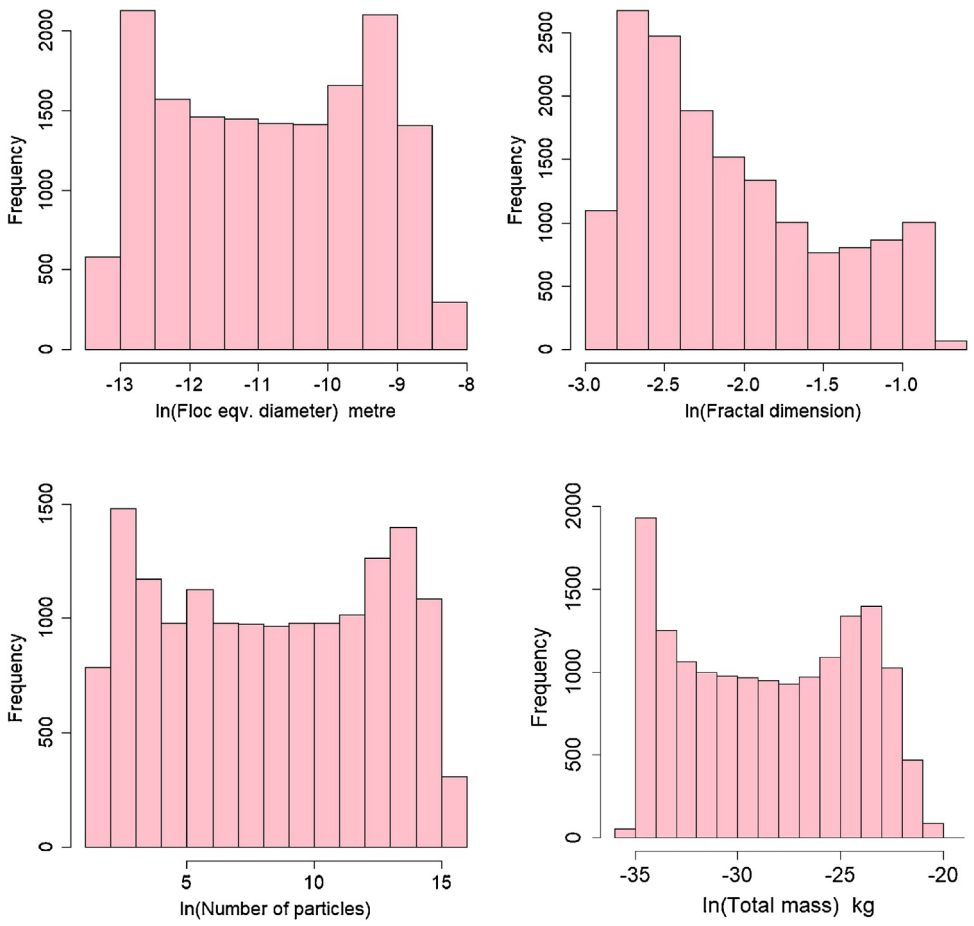


Fig. 4. Histogram of characterized floc outputs: floc equivalent diameter, fractal dimension, total mass and particle growth showing a large variation in these datasets. All plots are on a natural logarithmic scale.

experimental design points (i.e. $\mathbf{A} = cor(\mathbf{x}, \mathbf{x}')$). We are assuming a univariate output and a non-deterministic computer model.

Similarly, $t(x^{new}) = [cor(x_1, x^{new}), \ldots, cor(x_n, x^{new})]^T$ denotes the $(n \times 1)$ vector of correlations between the $\mathbf{x}$'s at the design points and new input points x^{new}. We use an exponential covariance function of the form

$$c(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\{-(\mathbf{x} - \mathbf{x}')^T \mathbf{R}(\mathbf{x} - \mathbf{x}')\} + \delta I, \tag{7}$$

where $\mathbf{R}$ is a diagonal matrix of correlation or scale hyperparameters. It determines how fast the spatial correlation decays throughout the input space to be estimated from the data. $\delta \geq 0$ is the nugget parameter and I is an indicator function which is 1 if $x = x'$ and 0 otherwise. The nugget is often considered as stochastic noise and typically represents measurement error. The nugget provides a mechanism for incorporating measurement error into the Gaussian process. It improves the stability of the computations and the predictive accuracy of the model [47,50,27,32].

The best linear unbiased predictor (BLUP) and mean squared prediction error (MSPE) for the universal kriging model are given as

$$\boldsymbol{\mu}(\mathbf{x}) = h^T(\mathbf{x})\hat{\boldsymbol{\beta}} + t^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}}), \tag{8}$$

$$\mathbf{c}^{**}(\mathbf{x}, \mathbf{x}') = \{cor(\mathbf{x}, \mathbf{x}') - t(\mathbf{x})^T\mathbf{A}^{-1}t(\mathbf{x}') + [h(\mathbf{x})^T - t(\mathbf{x})^T \mathbf{A}^{-1}t(\mathbf{x})](\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}[h(\mathbf{x}')^T - t(\mathbf{x}')^T\mathbf{A}^{-1}t(\mathbf{x}')]^T\}. \tag{9}$$

See more details in [25]. One limitation of the separate univariate GPs for modelling multiple output data is that it neglects the correlation between the outputs. We will address this problem by using a multivariate GP. The multivariate extension is based upon the Bayesian perspective (using non-informative priors). This framework will enable us to derive closed form expressions for the estimates of the parameters. The model specification looks similar to the univariate case earlier defined, except that we will be placing prior distributions on the unknown parameters. The multivariate normal distribution generalizes the univariate normal distribution where the k-dimensional density is given by

$$f(\mathbf{Y}) = \frac{1}{(2\pi)^{k/2}(|\boldsymbol{\Sigma}_\mathbf{Y}|)^{1/2}} \exp\frac{-1}{2}(\mathbf{Y} - \mathbf{HB})^T \boldsymbol{\Sigma}_\mathbf{Y}^{-1}(\mathbf{Y} - \mathbf{HB}), \tag{10}$$

where $|\boldsymbol{\Sigma}_\mathbf{Y}|$ is the determinant of covariance matrix. Suppose we now have k outputs $\mathbf{Y}(x) = (Y_1(x), \ldots, Y_k(x))$, which has a joint matrix normal distribution for n simulator runs, such that for any matrix $\mathbf{Y}_{n\times k}$ of outputs, we have

$$\mathbf{Y}|\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{R}, \delta \sim MN_{n,k}(\mathbf{HB}, \boldsymbol{\Sigma} \otimes \mathbf{A}), \tag{11}$$

where $\mathbf{H}_{n\times m}$ is the model matrix with ith row denoted as $h(.)^T$ and defined previously as a vector of regression functions. The matrix $\mathbf{B}_{m\times k}$ is a matrix of unknown regression coefficients and $\mathbf{R}_{n\times n}$ is a diagonal matrix of scale parameters and δ is the nugget parameter. To simplify our approach, we have assumed a separable covariance structure because it is relatively easy to perform and our outputs are also correlated. The assumption of separability of covariance function implies that output variance can be decomposed such that $cov(f(\mathbf{x}), f(\mathbf{x}')) = \boldsymbol{\Sigma}_Y = \boldsymbol{\Sigma} \otimes \mathbf{A}$, where $\boldsymbol{\Sigma}$ is an $k \times k$ positive definite matrix of cross-covariance between the outputs at any input and $\mathbf{A}_{n\times n}$ is a correlation matrix across the input space and $\otimes$ is a kronecker product operator. We use linear mean and exponential

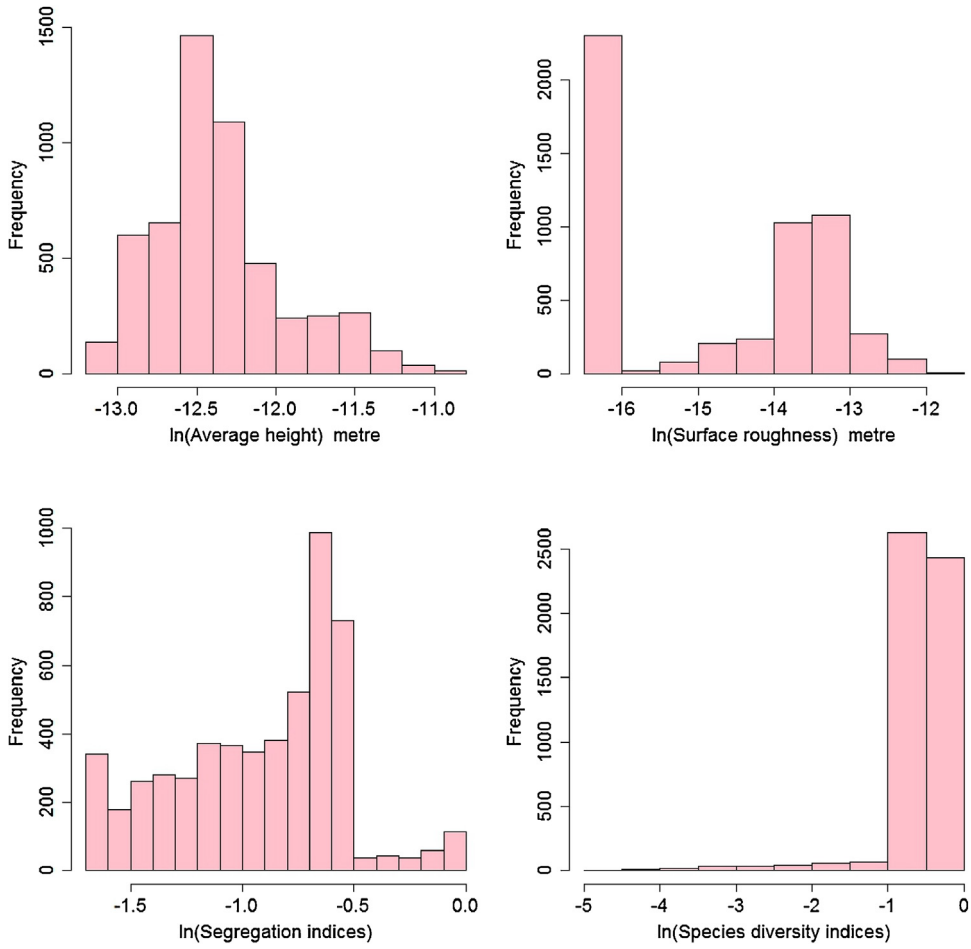


Fig. 5. Histogram of characterized biofilms outputs: biofilms height, surface roughness, segregation indices and Simpson diversity indices showing a large variation in these datasets. All plots are on a natural logarithmic scale.

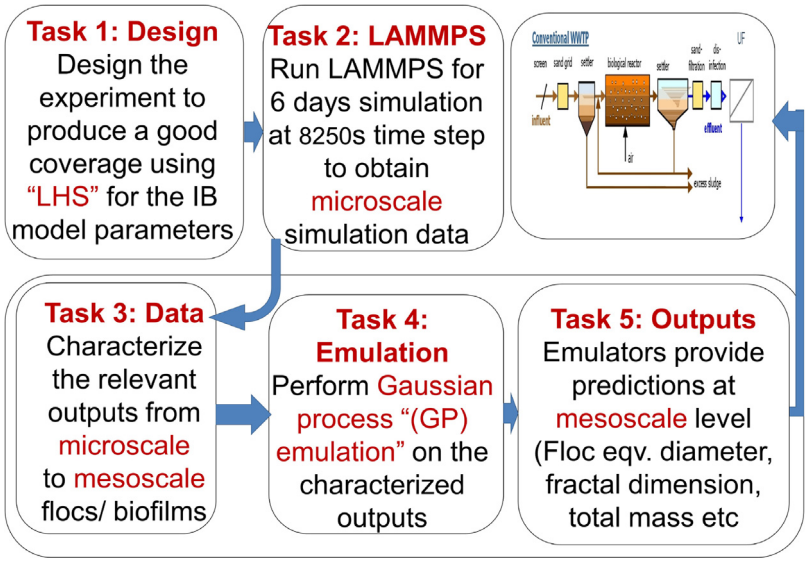


Fig. 6. Schematic diagram showing key emulation stages.

correlation functions as defined earlier under a univariate GP or kriging for an independent emulator.

The mean and covariance functions can be modelled in term of matrices **B**, $\boldsymbol{\Sigma}$ and **R** of hyperparameters and nugget parameter δ. In reality, these parameters are unknown, and major problem under the multivariate GP or kriging is their estimation. The next problem is how to estimate these unknown parameters. A popular approach is to use the maximum likelihood estimation (MLE) framework of [52] by maximizing the likelihood function for statistical parameter estimation. The MLE technique is related to the

maximum a posteriori estimation (MAP) that assumes a uniform prior distribution of the parameters. One of the limitations of the MLE is that the likelihood function could become flat near the optimum value [30]. A popular alternative estimation method is to use restricted or residual maximum likelihood (REML) of [43,53] which produces less-biased variance and covariance components. The REML approach can be derived from the likelihood function by accounting for the uncertainty in $\mathbf{B}$ and $\boldsymbol{\Sigma}$, i.e., by integrating them out.

To marginalize out $\mathbf{B}$ and $\boldsymbol{\Sigma}$, in practice, it is often appealing to consider a Bayesian perspective when building metamodels or surrogate models [59,5,8]. Bayesian emulation assigns a Gaussian process prior distribution to the function $f(.)$, conditional on unknown parameters. This prior distribution is updated using training data. In our case, unknown parameters $\mathbf{B}$, $\boldsymbol{\Sigma}$, $\mathbf{R}$, δ of the multivariate Gaussian process or kriging are treated as random variables which produce the posterior distribution of $f(.)$ as the emulator.

Because we have little or no prior information about the mean and covariance parameters $\mathbf{B}$ and $\boldsymbol{\Sigma}$, we also follow [9] and [8] approaches, where we use a non-informative prior $\pi(\mathbf{B}, \boldsymbol{\Sigma}, \tilde{\mathbf{R}}) \propto \pi_{\mathbf{R}}(\tilde{\mathbf{R}})|\boldsymbol{\Sigma}|^{-(k+1)/2}$ which are both related to recent approaches of [40] and [47]. We can obtain the conditional posterior distribution of the computer model $f(.)|\mathbf{B}, \boldsymbol{\Sigma}, \tilde{\mathbf{R}}, \mathbf{Y}$ by combining Eq. (11) with standard multivariate normal theory, using some algebraic manipulations that produces

$$f(.)|\mathbf{B}, \boldsymbol{\Sigma}, \tilde{\mathbf{R}}, \mathbf{Y} \propto MN_m(m^*, c^*(.,.)\boldsymbol{\Sigma}), \tag{12}$$

where $\tilde{\mathbf{R}} = [\mathbf{R}, \delta]$ and m^* and c^* are given respectively by Eqs. (A.3) and (A.4) in Appendix 3. We can further derive the conditional posterior distribution of $f(.)|\tilde{\mathbf{R}}$ by integrating Eq. (12) with respect to $\mathbf{B}$ and $\boldsymbol{\Sigma}$ such that

$$f(.)|\tilde{\mathbf{R}}, \mathbf{Y} \propto T_k(\boldsymbol{\mu}(.), \mathbf{c}^{**}(.,.)\hat{\boldsymbol{\Sigma}}, n-m), \tag{13}$$

where T_k is the conditional students' t-process with $(n-m)$ degrees of freedom where $\boldsymbol{\mu}(.)$ and $\mathbf{c}^{**}(.,.)$ are given respectively as

$$\boldsymbol{\mu}(x) = h(x)^T\hat{\boldsymbol{B}} + t^T(x)\mathbf{A}^{-1}(\mathbf{Y} - \mathbf{H}\hat{\boldsymbol{B}}), \tag{14}$$

$$\mathbf{c}^{**}(\mathbf{x}, \mathbf{x}') = c^*(\mathbf{x}, \mathbf{x}') + [h(\mathbf{x}) - \mathbf{H}^T\mathbf{A}^{-1}t(\mathbf{x})]^T(\mathbf{H}\mathbf{A}^{-1}\mathbf{H})^{-1}[h(\mathbf{x}') - \mathbf{H}^T\mathbf{A}^{-1}t(\mathbf{x}')]. \tag{15}$$

We note that these estimated parameters are also equivalent to the best linear unbiased predictor (BLUP) and mean squared prediction error (MSPE) derived under a frequentist perspective with $\hat{\boldsymbol{B}} = (\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{A}^{-1}\mathbf{Y}$ and $\hat{\boldsymbol{\Sigma}} = (\mathbf{Y} - \mathbf{H}\hat{\boldsymbol{B}})^T\mathbf{A}^{-1}(\mathbf{Y} - \mathbf{H}\hat{\boldsymbol{B}})(n-m)^{-1}$ corresponding to generalized least square estimates. We can obtain the full posterior $\pi_{\tilde{\mathbf{R}}}(\mathbf{B}, \boldsymbol{\Sigma}, \tilde{\mathbf{R}}|\mathbf{Y})$ (see Eq. (A.5) in Appendix 3) of the hyperparameters by combining Eq. (11) and full prior $\pi(\mathbf{B}, \boldsymbol{\Sigma}, \tilde{\mathbf{R}})$ using Bayes theorem which after integrating out both $\mathbf{B}$ and $\boldsymbol{\Sigma}$ produces

$$\pi_{\tilde{\mathbf{R}}}(\tilde{\mathbf{R}}|\mathbf{Y}) \propto \pi_{\tilde{\mathbf{R}}}(\tilde{\mathbf{R}})|\mathbf{A}|^{-k/2}|\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H}|^{-k/2}|\mathbf{Y}^T G\mathbf{Y}|^{-(n-m)/2}, \tag{16}$$

where G is defined by Eq. (A.6) in Appendix 3. We know it is relatively hard to elicit a prior distribution for the scale parameter $\tilde{\mathbf{R}}$ that can produce an analytical expression for the posterior distribution in Eq. (16). Also, the fully Bayesian approach using Markov chain Monte Carlo for removing the dependence of the scale parameter $\mathbf{R}$ will lead to a significant increase in computational time. We use a plug in approach based on a posterior mode of $\tilde{\mathbf{R}}$, derived by maximizing Eq. (16). We set the prior distribution $\pi_{\tilde{\mathbf{R}}}(\tilde{\mathbf{R}})$ to a constant since we have no prior information about the scale parameter $\tilde{\mathbf{R}}$, therefore we neglect the $\pi_{\tilde{\mathbf{R}}}(\tilde{\mathbf{R}})$ term in our optimisation routine. See further details in [9], [8] and [47].

4. Procedure for emulating IB model outputs

We consider separately the problems of emulating flocs and biofilms. There are two different potential approaches to each of these problems. Firstly, we could emulate the individual bacterium at the micro level and use the emulator to link the simulator output at a mesoscale level as a floc or biofilm. This approach is currently not practicable owing to the large amount of simulation data involved, although it could be possible to perform some form of data reduction.

The second approach, which we adopt in this study, is to focus on clusters of particles as flocs and biofilms and emulate their interesting properties, as described in Section 2.3. A single run of the LAMMPS model consists of a simulation over many time steps which requires considerable computer time. Here, we shall focus on floc emulation, and in particular, we shall describe the emulation of floc equivalent diameter, fractal dimension, the total number of particles and floc total mass, to simplify our approach. Emulation of other outputs will follow a similar procedure. The floc is treated as an approximate 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 Fig. 2(a) and (b). The detailed procedure of emulating the floc parameters will be described in this section and for the biofilm is deferred to the next section.

Some of the challenges of emulating the LAMMPS model are the nature of the outputs produced from the model which make it 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 randomness in the output. 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 this, there is a vast literature addressing these problems. The stochasticity in the model is first handled by performing multiple runs and averaging the key outputs which are then taken as deterministic in nature. Secondly, we fit non-zero nugget univariate and multivariate GP models to incorporate stochastic noise in our model formulation.

4.1. Dynamic emulation

Due to the dynamic nature of output data from the LAMMPS model, we apply a dynamic emulation strategy within a multivariate GP 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 implies that model output at time t becomes an input to the model at time $t+1$. The model can be written as

$$\mathbf{Y}_t = f(\mathbf{x}_t, \mathbf{Y}_{t-1}), \tag{17}$$

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 a design matrix) are the inputs at time t which includes the model parameters, forcing and initial conditions (see Table 2). We use the single-step emulation technique proposed in [9]. 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.

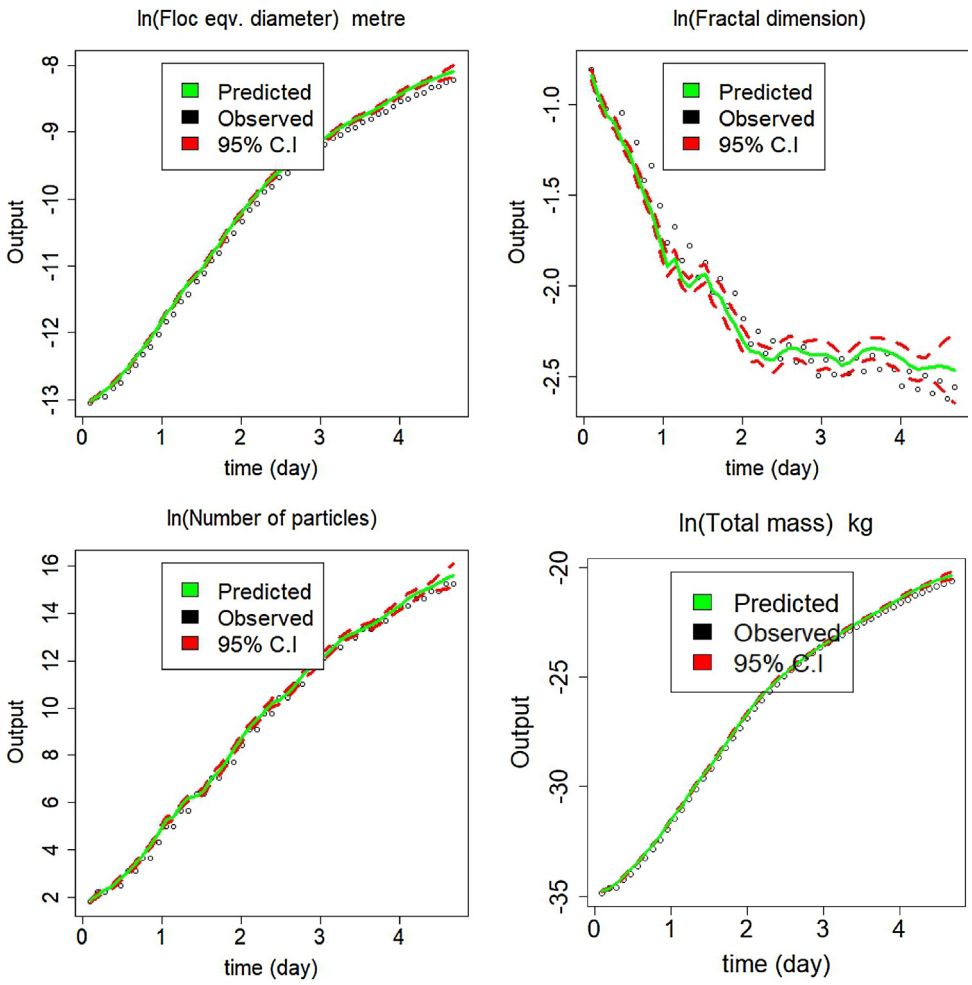


Fig. 7. Comparison of the multivariate GP emulator performance with simulation data for 4 major outputs from LAMMPS floc simulation (black) and their emulator predictions (green) with 95% C.I. (red). Note: the outputs are plotted on a natural logarithmic scale. (For interpretation of the references to color in this legend, the reader is referred to the web version of the article.)

4.1.1. Single-step emulation

Starting from initial runs 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 on the characterized outputs from LAMMPS model. We can use dynamic emulation to make multiple step ahead predictions using an iterative technique to repeat one-step-ahead predictions until the desired number of points are obtained. We proceed sequentially, feeding back the entire output distribution from the GP model, such that at time step $t = 1$, and for input $(\mathbf{x}_1, \mathbf{Y}_0)$, we sample from the distribution of $f(\mathbf{Y}_0, \mathbf{x}_1)$, with the model output given as $\tilde{\mathbf{Y}}_1^{(s)} \sim N[\boldsymbol{\mu}(\mathbf{x}, \mathbf{Y}_0), \mathbf{c}^{**}(\mathbf{x}, \mathbf{Y}_0; \mathbf{x}', \mathbf{Y}_0)]$. 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 a 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 are completed. The construction of single-step emulator is summarized below:

(i) Sample 280 points randomly from original 300 points as training datasets and leave the 20 data points for the cross validation.
(ii) Formulate a single step emulator using Eq. (17) such that $\mathbf{y}_1 = f(\mathbf{x}, \mathbf{y}_0)$, where $\mathbf{x}$ is the design matrix that is uniformly sampled to cover the entire space of interest for running the LAMMPS model for the single step function. The corresponding output is the value of current state variable y_t, e.g. fractal dimension.
(iii) Perform GP emulation as described in Section 3.1, where we use linear mean and exponential covariance functions. The multivariate GP parameters $\Theta = (\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{R}, \delta)$ are estimated using Bayesian inference with non informative priors.
(iv) Compute the posterior distribution of multivariate GP $(f(.)|\mathbf{Y}, \hat{\Theta}) \sim N(\boldsymbol{\mu}(x), \mathbf{c}^*(\mathbf{x}, \mathbf{x}'))$ where $\boldsymbol{\mu}(x)$ and $\mathbf{c}^{**}(\mathbf{x}, \mathbf{x}')$ are defined in Eqs. (14) and (15) respectively and Eqs. (8) and (9), for univariate case.
(v) Use the emulator to simulate from $(f(.)|\mathbf{Y}_1, \hat{\Theta})$ 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 $[\tilde{\mathbf{Y}}_1^{(s)}, \ldots, \tilde{\mathbf{Y}}_{T-1}^{(s)}]$.

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 an ill-conditioned covariance matrix as training data are augmented. Moreover, an additional computational cost is often involved. [9] proposed a simple normal approximation to the above procedure that we applied in this study. This approach is comparable to a technique due to [1] applied on a nonlinear dynamic system to propagate uncertainty in iterative multiple-step-ahead predictions. Now, we can estimate the two quantities in Eqs. (14) and (15)

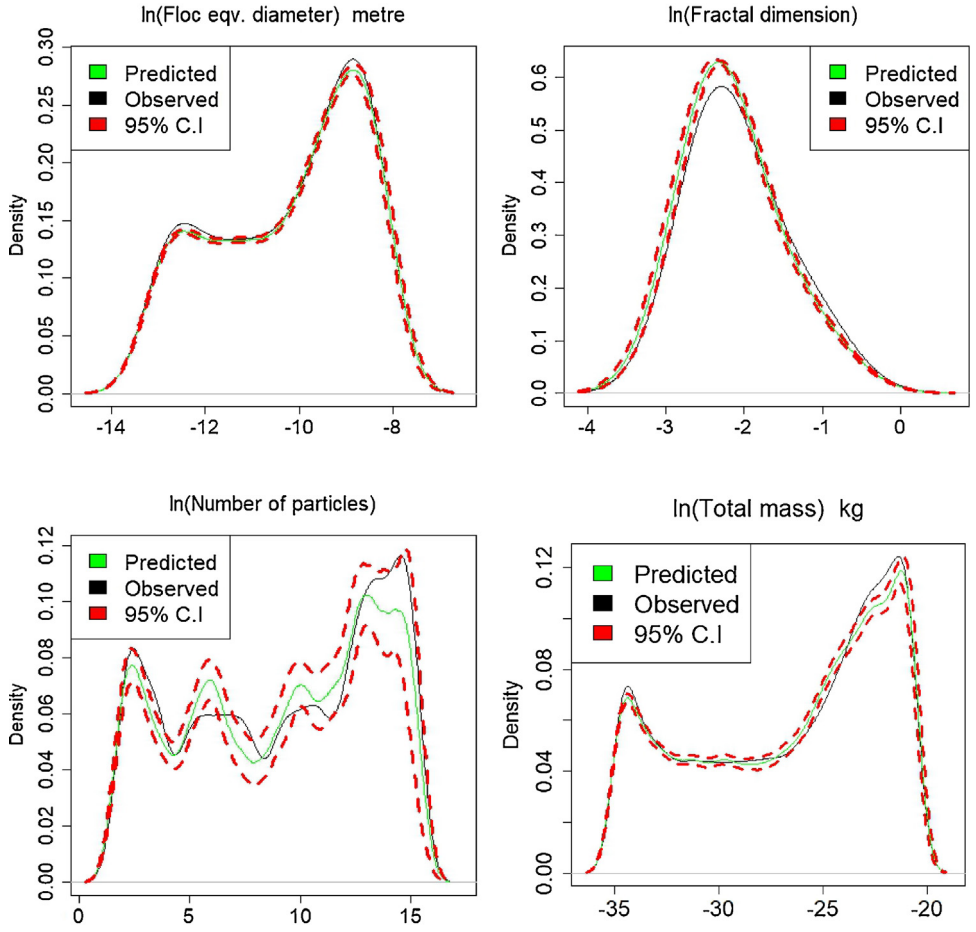


Fig. 8. Multivariate GP emulator using a normal approximation procedure for cross-validation of a randomly chosen design point: probability density function for 4 major outputs from LAMMPS floc simulation and their emulator predictions with 95% C.I. Note: the outputs are plotted on a logarithmic scale.

using simulation from Monte Carlo sampling to repeatedly revise the mean and variance of the single step emulator such that

$$\hat{\boldsymbol{\mu}}_{t+1} = \frac{1}{N}\sum_{s=1}^{N}(\boldsymbol{\mu}(\tilde{\mathbf{Y}}_t^{(s)}, \mathbf{x}_{t+1})|f(\mathbf{Y})), \tag{18}$$

$$\hat{\mathbf{c}}_{t+1}^{**} = \frac{1}{N}\sum_{s=1}^{N}[\mathbf{c}^{**}(\mathbf{x}_{t+1}, \tilde{\mathbf{Y}}_t^{(s)}), (\mathbf{x}_{t+1}, \mathbf{Y}_t)|f(\mathbf{Y})] + \frac{1}{N}\sum_{s=1}^{N}[\mu^{\bullet}(\tilde{\mathbf{Y}}_t^{(s)}, \mathbf{x}_{t+1})|f(\mathbf{Y})]^2, \tag{19}$$

where $\tilde{\mathbf{Y}}_t^{(s)}$ is a sample from $MN[\boldsymbol{\mu}_t(.), \mathbf{c}_t^{**}(.)]$ and N is the number of Monte Carlo samples. This new approximation is based on the assumption that augmentation of training data at each iteration step will have a relatively minimal effect provided that a large sample size is used 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, it is difficult to derive a joint distribution for $\mathbf{Y}_1, \ldots, \mathbf{Y}_T$ in a closed form, rather a normal approximation is proposed to estimate the marginal density of each $\mathbf{Y}_t$ for $t = 1, \ldots, T$. Note that this procedure is also applicable to the univariate GP model.

5. Results

5.1. Floc emulation

Having summarised the individual particles at the microscale to a large ensemble (mesoscale) as flocs or biofilms at each time step, we consider emulation of interesting properties of flocs and biofilms. To test the adequacy of our proposed dynamic emulator using a multivariate GP, we apply the normal approximation scheme to model the eight characterized outputs from the LAMMPS model. In this section, we shall focus our attention on the floc emulation (floc equivalent diameter, fractal dimension, the total number of particles and total mass).

Suppose at time step t, the LAMMPS output is written in the form $\mathbf{Y}_t = f(\mathbf{x}_t, \mathbf{Y}_{t-1})$, where $\mathbf{Y}_{t-1}$ is the state vector at the previous time step, $\mathbf{x}_t$ are the input at time t as defined in the previous section. The data are subdivided into two groups. We use 280 data points as training data to build the single-step emulator and use the remaining 20 data points to test the performance of the emulator.

The multivariate GP could not be applied directly to the entire data because of the computational difficulty of large sample size ($17{,}360 = 280 \times 62$) in our training set, coupled with a significant number (36) of parameters (32 model parameters and four state variables) to be estimated. We know GP algorithms scale cubically with the number of observations $O(N^3)$, and so these are not feasible for our present data. Although it is possible to perform dimension reduction, we will not follow this approach in this paper. GP regression is prone to numerical problems because of an ill-conditioned covariance matrix. This issue is more pronounced in the multivariate GP algorithm we use in this study.

To proceed, we subsample 2500 observations randomly to cover the space of interest from each output. We fit a separate univariate GP model assuming independence between the characterized outputs. The advantage of first fitting an independent univariate GP is that one can easily perform surrogate based sensitivity analysis to

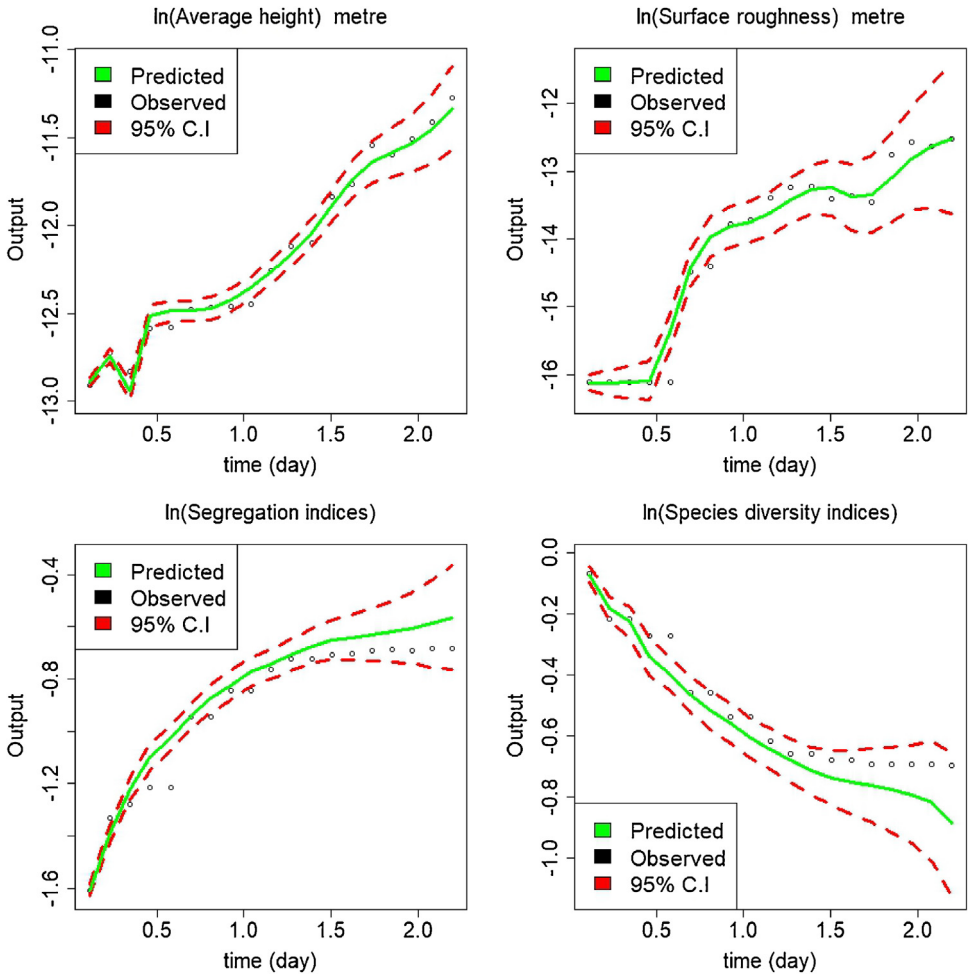


Fig. 9. Comparison of the multivariate GP emulator performance with simulation data for 4 major outputs from LAMMPS biofilms simulation (black) and their emulator predictions (green) with 95% C.I. (red). Note: the outputs are plotted on a natural logarithmic scale. (For interpretation of the references to color in this legend, the reader is referred to the web version of the article.)

identify relevant parameters. This will reduce the high dimensionality of the parameter space saving considerable computation time during joint output emulation.

The Bayesian sensitivity analysis was carried out to identify the most relevant variables, see further details of sensitivity analysis in Section 5.3. Based on the outcome of the sensitivity analysis, we proceed with multivariate GP emulation using only nine selected input variables (five most important model parameters and four state variables). To further reduce the effect of an ill-conditioned covariance matrix, we use an exponential covariance function for each output and standardize our input data to range over [0, 1]. This transformation will also eliminate the unit of measurement and enable us to get better parameter estimates for the covariance functions.

Having built the single-step emulator for the characterised outputs, we execute the single-step emulator repeatedly using Eqs. (18) and (19) derived from the normal approximation until time $t = 62$ is reached. Because of the stochasticity in the simulation, we fit a non-zero nugget multivariate GP model by incorporating the nugget as measurement noise in the mean response emulator. Apart from improving the stability of the computations, the presence of the nugget also encourages in more robust parameter estimation leading to better predictive accuracy.

This approach will ensure joint predictions of both the mean and variance, unlike in [17] and [24] where an independent GP model is performed on the mean and variance. To test the overall performance of the single-step emulator, we run the emulator for the complete 20 test data points and compute the proportion of variance explained (ρ) and root mean squared error of cross-validation (RMSE_{CV}) by each of the model outputs.

We assess the performance of the dynamic emulator by comparing emulator predictions with the simulation data for four different characterised outputs from flocs. The cross-validation results are reported in Fig. 7 which also gives time series plots showing the patterns of change in the 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 the desired time point.

The plots for the four log-transformed floc outputs shows that they are relatively well predicted. The emulator predictions are similar to the simulation data. The top-left corner is the floc equivalent diameter which is an important morphological property of flocs. There is a nonlinear increase in trend over time indicating that emulator captures the growing patterns relatively well except in few places where the simulation data lie slightly outside the confidence bands. The predicted confidence bands remain small.

The plot of the fractal dimension is shown in the top-right corner. The value of the fractal dimension is an indication of the structural complexity of the biological particle allowing the fractal dimension to be used as a standard for comparing the biological experiments against theories. The fractal dimension indicates a decreasing trend pattern because the irregular shape at the begin-

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 for both univariate and multivariate GP for the 20 left-out data points.

Outputs	Univariate		Multivariate	
	ρ	$RMSE_{CV}$	ρ	$RMSE_{CV}$
Floc eqv. diameter (m)	0.94	0.35	0.91	0.44
Floc fractal dimension	0.99	0.22	0.99	0.16
Floc total number of particles	0.99	1.33	0.99	1.11
Floc total mass (kg)	0.99	1.04	0.99	1.29
Biofilm mean height (m)	0.84	0.18	0.87	0.16
Biofilm surface roughness (m)	0.83	1.11	0.93	0.71
Biofilm segregation indices	0.99	0.22	0.99	0.16
Simpson species diversity indices	0.99	0.61	0.99	0.47

ning of the simulation becomes more uniform in nature, unlike floc equivalent diameter. Similar to the floc equivalent diameter, the emulator for fractal dimension predicts the temporal behaviour relatively well; almost all the points lie close to the 95% C.I. The growth curve (total number of particles) and the total mass of the floc grow non-linearly with time, and both have similar patterns.

Fig. 8 shows a comparison between the probability density function of the simulated and emulated floc equivalent diameter, fractal dimension, the total number of particles and total mass. The predicted densities 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 emulator. The distributions are relatively bimodal in nature having both major and minor peaks except for number of particles that has a single major peak and three minor peaks (multimodal). Moreover, the four density plots are quite dissimilar as earlier observed under their time series plots in Fig. 7. Overall, the multivariate GP emulator reproduces the temporal patterns quite well.

Table 1 compares the overall performance of the emulators using ρ and $RMSE_{CV}$ metrics. The closer the values of ρ to 1, the better the emulator and vice versa. Therefore, regarding the proportion of variance ρ, the multivariate GP seems to outperform the individual univariate emulator for all of the outputs except for floc equivalent diameter where there is a reduction in the value of ρ from 0.94 to 0.91. On the other hand, lower values of root mean squared error $RMSE_{CV}$ indicates more accurate predictions. Similarly, we observe that the corresponding values of $RMSE_{CV}$ in the multivariate GP model are lower than that of the univariate GP emulator with the exceptions of floc equivalent diameter and total mass. This suggests that multivariate GP emulators are much better than independent univariate emulators. This is due to the incorporation of output correlation in the multivariate models.

5.2. Biofilm emulation

We now consider the emulation of bacterial biofilms, where we apply the same procedure as in the floc modelling. The plots show the assessment of our dynamic emulation approach on the biofilm outputs where the emulators have been used iteratively to capture the evolution of each of the characterised outputs with time. Computing the biofilm surface roughness, average height and segregation indices for a set of large particles for a longer period as in this study is very time-consuming. We therefore limit our computation for these critical biofilm parameters to a few time points (about two days) as shown in Fig. 9. In Fig. 9, the top-left plot is the biofilm average height, and the top-right plot is the biofilm surface roughness. The biofilm height is irregular in shape (<0.5 day) before a gentle increment and a further non-linear increase, most of the simulation points fall within the 95% C.I. The surface roughness is approximately constant at the earlier time before rapidly increased over time until a saturated point (say 0.6 days), after which there is a fairly steady increase in growth. We see that predictions produced by the simulator and emulator are close and have similar temporal patterns with only a few points falling outside of the confidence bands.

The bottom-right corner is the plot for the biofilm species diversity. This index represents the abundance of different microbial species in a population and usually ranges between 0 and 1. A large value signifies greater diversity, while the plots for the average height, surface roughness and segregation indices clearly show a non-linear increasing pattern, a gradual decreasing trend over time is expected because the species interact with the environment and other particles within the biofilms and also compete together for available nutrients. In this study, we know that the HET microbes compete with both AOB and NOB for oxygen. The AOB and NOB are easily outcompeted and then HET dominate at the end due to their higher growth rates and the insensitivities of HET to oxygen conditions. The species diversity emulator produces a fairly similar pattern to the simulation data. The effect of random variation is more apparent in the histogram in Fig. 5. Besides, there is a significant presence of extreme data which could potentially complicate the modelling and ability of the emulator to capture all the emergent behaviour. The biofilm density plots in Fig. 10 are also well predicted.

Overall, the 95% C.I. for the biofilms are large compared to the floc emulation with narrow bands. The uncertainty levels are generally increasing with time as expected, an indication of degradation of emulator performance. We note that the shape, size and structure of biofilms and flocs are essential operational parameters in the management of wastewater. These characterised physical properties are significant in the removal efficiency of flocs in the wastewater treatment processes.

5.3. 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 was performed. The results from this analysis will enhance our understanding of the most influential input variables to output behaviour (out of the 32 parameters we used in this study). Several techniques have been documented in the literature for performing sensitivity analysis [20], but one popular technique is the Sobol global sensitivity method. This method computes the indices by decomposing the variance up to a specified order [51]. We use a Bayesian approach for global sensitivity described in [29] where a Sobol index is estimated based on GP 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 only the first order sensitivity since we have no quadratic and interaction terms in our model. We then apply bootstrapping to compute 95% confidence intervals on the estimated indices. This procedure was applied to all eight outputs we consider in this study. The combined results for both the flocs and biofilms are given in Fig. 11 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, particles grow when the food is readily available. The substrate boundary concentration for nitrate "no_3" with sensitivity indices 1.0, 0.90, 0.39 and 0.38 for floc fractal dimension, biofilm segregation indices, average height and species

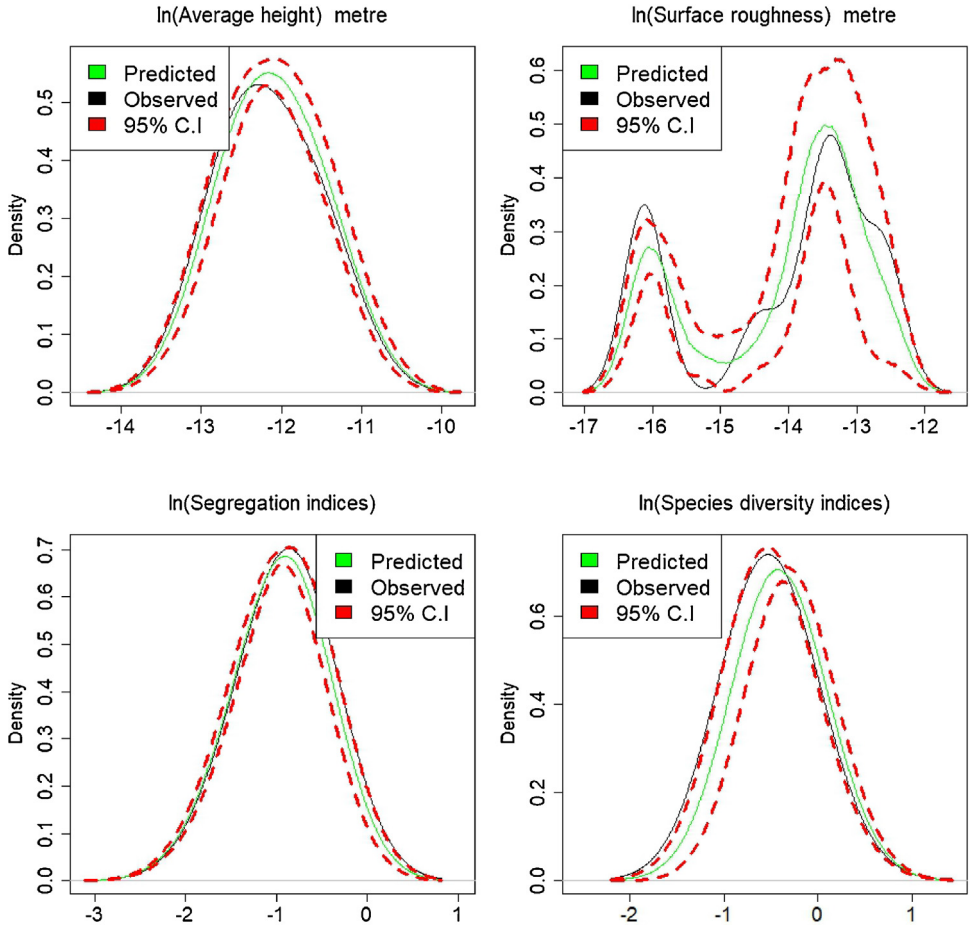


Fig. 10. Multivariate GP emulator cross-validation: probability density function for 4 major outputs from LAMMPS biofilms simulation and their emulator predictions with 95% C.I. Note: the outputs are plotted on a natural logarithmic scale.

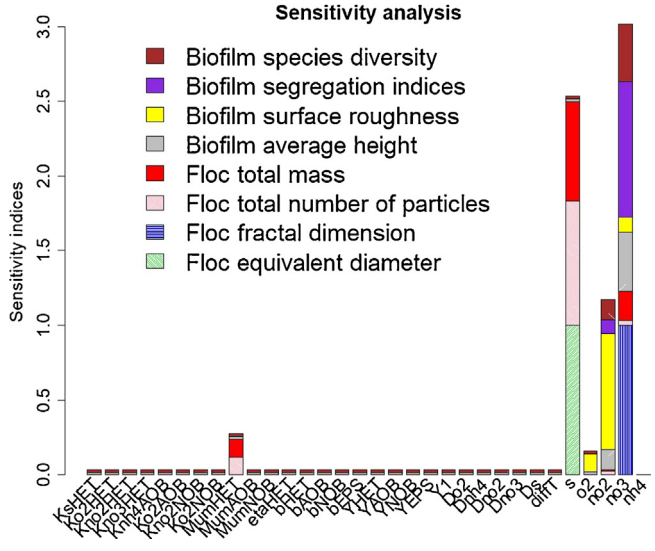


Fig. 11. Barplots showing the sensitivity indices for the eight characterized outputs from LAMMPS model.

diversity respectively. However, other nutrient boundary conditions influence the biofilms greatly.

The carbon substrate "s" is the second most sensitive parameter for the floc equivalent diameter, the total number of particles and total mass and are less important for biofilm properties. Apart from nutrient concentrations, the only sensitive parameter is the maximum specific growth rate for HET "MumHET". It is apparent that the majority of the variables are less important or non-essential as indicated by their low sensitivity indices. Having identified these most relevant parameters for flocs and biofilms, we then based our multivariate GP modelling on these few selected parameters. This reduction in the dimension of input parameters significantly speeds up computation.

6. Discussion

This paper has shown that the multivariate GP technique can be effectively applied to model a dynamic simulation model while incorporating the non-zero nugget as a random noise within the framework. We observe that the performance of the single-step emulator used iteratively to capture the underlying dynamics of output behaviours of the IB model depends 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 this paper, we were restricted to just 1000 samples to speed up the algorithm because of the expense of running the emulator repeatedly. Another critical issue is the evaluation of ill-conditioned covariance parameters which we often encountered in this analysis because of closely spaced design points. The introduction of a nugget parameter and the use of an exponential covariance function reduced the severity of the problem. We also standardise our input data to range in [0, 1]. This transformation also helps to eliminate the effect of measurement units and enables us to get better parameter estimates for covariance functions. Overall, the

Table 2
List of IB model parameters.

Index	Parameters	Values	Units	References
Affinity variables				
1	KsHET	0.004	$kg\,m^{-3}$	[55]
2	Ko2HET	0.0002	$kg\,m^{-3}$	[55]
3	Kno2HET	0.0003	$kg\,m^{-3}$	[37]
4	Kno3HET	0.0003	$kg\,m^{-3}$	[37]
5	Knh4AOB	0.001	$kg\,m^{-3}$	[7]
6	Ko2AOB	0.0005	$kg\,m^{-3}$	[7]
7	Kno2NOB	0.0013	$kg\,m^{-3}$	[7]
8	Ko2NOB	0.00068	$kg\,m^{-3}$	[7]
Maximum growth variables				
9	MumHET	0.00006944444	s^{-1}	[18]
10	MumAOB	0.0000088	s^{-1}	[7]
11	MumNOB	0.000009375	s^{-1}	[7]
12	etaHET	0.6	–	[18]
Decay rates variables				
13	bHET	0.00000462962	s^{-1}	[37]
14	bAOB	0.00000127314	s^{-1}	[37]
15	bNOB	0.00000127314	s^{-1}	[37]
16	bEPS	0.00000196759	s^{-1}	[37]
Yield coefficient variables				
17	YHET	0.61	gCOD/gCOD	[34]
18	YAOB	0.33	gCOD/gN	[7]
19	YNOB	0.083	gCOD/gN	[7]
20	YEPS	0.18	gCOD/gN	[34]
Diffusion coefficient variables				
21	D_{o2}	0.000000002	$m^2\,s^{-1}$	[2]
22	D_{nh4}	0.0000000014	$m^2\,s^{-1}$	[2]
23	D_{no2}	0.0000000012	$m^2\,s^{-1}$	[2]
24	D_{no3}	0.0000000012	$m^2\,s^{-1}$	[2]
25	D_s	0.0000000005	$m^2\,s^{-1}$	[2]
Critical diameter of death				
26	deadDia	0.0000008	–	–
27	factor	1.5	–	–
Boundary concentrations (nutrients)				
28	sub	0.008	$kg\,COD\,m^{-3}$	–
29	no2	0.0001	$kg\,N\,m^{-3}$	–
30	no3	0.0008	$kg\,N\,m^{-3}$	–
31	o2	0.0008	$kg\,m^{-3}$	–
32	nh4	0.0009	$kg\,N\,m^{-3}$	–

performance of these single-step emulators 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 [8] 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 of 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 is 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 [6] but is not suitable for our present data. Here, we are interested in the dynamic behaviour over time, and specifically for using an emulator for making multiple-step ahead predictions.

We have employed a separable covariance model because it is mathematically tractable and simplifies our estimation procedure. We believe the assumption of separability of covariance function to model the multiple outputs is not too restrictive because the emulators performed quite well for most of the outputs. We could have tried either of the convolution techniques which involves a mixture of a Gaussian white noise process with some smoothing kernel. Another alternative is to consider a linear model of coregionalization where outputs are a linear combination of independent univariate GPs, but these methods are too computational demanding [14]. They require estimation of more parameters and full inversion of a nonseparable covariance matrix, an infeasible task in practice. A possible extension of this work would be to extend to nonseparable covariance functions. It would be useful to examine whether we can further improve our results by using nonseparable covariance functions.

7. Conclusion

In this paper, we demonstrate a new method of making inference about the parameters of an emulator using a GP regression model that is based upon a multivariate GP technique. The technique described in this paper could be seen as an extension of [9] and [8] which focus only on deterministic simulators. Stochasticity in the simulation is treated by incorporating a non-zero nugget parameter as a measure of random noise in the simulation. Our approach combines the two-stage technique proposed in 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 than the simulation model. The IB model simulation implemented within LAMMPS is computationally expensive, while these emulators give results almost instantaneously. Under different parameter combinations, it takes an average of between 5 and 6 h to simulate the growth of the particles for about six days at 8250 s timestep on a Linux cluster machine. Apart from the computational time required 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 an approximately 220-fold increase in computational efficiency. It is not plausible that any IB models (no matter how efficient) will be able to reproduce an entire real world microbial community such as a wastewater treatment plant or human microbiome. Therefore, an emulator, though this can be undoubtedly be improved, represents an important step forward in the simulation of complete microbial systems.

Acknowledgements

The work has been supported by the EPSRC (grant No. EP/K039083/1), by the Newcastle University Frontiers in Engineering Biology (NUFEB) project. We thank the NUFEB modelling team for their useful comments that have helped improve this paper.

Appendix 1. Model parameters

See Table 2.

Appendix 2. Model performance

Let $\mathbf{y}$ denote the LAMMPS values (e.g. floc or biofilm) where $\bar{\mathbf{y}}$ is the grand mean of each LAMMPS output and $\hat{\mathbf{y}}$ as the emulator predictions. We compute the squared differences between the actual LAMMPS outputs and their grand mean, also compute the squared differences between the LAMMPS values and the emulator predic-

tions. The proportion of the variance in the LAMMPS model that is explained by the emulator is given as

$$\rho = 1 - \left[\frac{\sum_{t=1}^{62}\sum_{n=1}^{20}(\mathbf{y}_{tn} - \hat{\mathbf{y}}_{tn})^2}{\sum_{t=1}^{62}\sum_{n=1}^{20}(\mathbf{y}_{tn} - \bar{\mathbf{y}})^2} \right] \tag{A.1}$$

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

$$\mathrm{RMSE}_{CV} = \left(\sum_{t=1}^{62}\sum_{n=1}^{20} \frac{(\mathbf{y}_{tn} - \bar{\hat{\mathbf{y}}}_{tn})^2}{(72 \times 20)} \right)^{1/2}. \tag{A.2}$$

Appendix 3. Parameter estimation

The following given equations are further estimates from the parameter estimation of multivariate GP.

$$m^*(\mathbf{x}) = h(\mathbf{x})\mathbf{B}^T + t(\mathbf{x})\mathbf{A}^{-1}(\mathbf{Y} - \mathbf{HB})^T \tag{A.3}$$

$$c^*(\mathbf{x}, \mathbf{x}') = cor(\mathbf{x}, \mathbf{x}') - t^T(\mathbf{x})\mathbf{A}^{-1}t(\mathbf{x}') \tag{A.4}$$

$$\pi(\mathbf{B}, \boldsymbol{\Sigma}, \tilde{\mathbf{R}}|\mathbf{Y}) \propto \pi_{\tilde{\mathbf{R}}}(r)|\mathbf{A}|^{-k/2}|\boldsymbol{\Sigma}|^{-\frac{n-m+k+1}{2}} \exp\left[-\frac{1}{2} tr((\mathbf{Y}^T G \mathbf{Y} \boldsymbol{\Sigma}^{-1}) + tr(\mathbf{B} - \hat{\boldsymbol{B}})^T \mathbf{H}^T \mathbf{A}^{-1} \mathbf{H} (\mathbf{B} - \hat{\boldsymbol{B}}) \boldsymbol{\Sigma}^{-1}) \right] \tag{A.5}$$

$$G = \tilde{\mathbf{R}}^{-1} - \tilde{\mathbf{R}}^{-1}\mathbf{H}(\mathbf{H}^T\tilde{\mathbf{R}}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\tilde{\mathbf{R}}^{-1} \tag{A.6}$$

Data reference

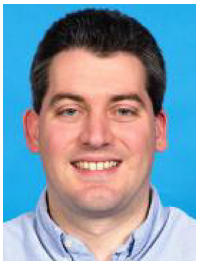
O.K. Oyebamiji, D.J. Wilkinson, P.G. Jayathilake, T.P. Curtis, S.P. Rushton, B. Li, P. Gupta (2017). Dataset: Gaussian process emulation of an individual-based model simulation of microbial communities. Data repository.

References

[1] K. Ažman, J. Kocijan, Comprising prior knowledge in dynamic Gaussian process models, Proceedings of the International Conference on Computer Systems and Technologies – CompSysTech, vol. 16 (2005).
[2] E. Alpkvist, C. Picioreanu, M. van Loosdrecht, A. Heyden, Three-dimensional biofilm model with individual cells and continuum EPS matrix, Biotechnol. Bioeng. 94 (5) (2006) 961–979.
[3] A.L. Amaral, M.M. Alves, M. Mota, E.C. Ferreira, Morphological characterisation of microbial aggregates by image analysis, Proceedings of the 9th Portuguese Conference on Pattern Recognition (RecPad'97) (1997) 95–100.
[4] R.A. Bates, R.S. Kenett, D.M. Steinberg, H.P. Wynn, Achieving robust design from computer simulations, Qual. Technol. Quant. Manag. 3 (2) (2006) 161–177.
[5] I. Bilionis, N. Zabaras, B.A. Konomi, G. Lin, Multi-output separable Gaussian process: towards an efficient, fully Bayesian paradigm for uncertainty quantification, J. Comput. Phys. 241 (2013) 212–239.
[6] A. Boukouvalas, P. Sykes, D. Cornford, H. Maruri-Aguilar, Bayesian precalibration of a large stochastic microsimulation model, IEEE Trans. Intell. Transp. Syst. 15 (3) (2014) 1337–1347.
[7] E.R. Bruce, L.M. Perry, Environmental Biotechnology: Principles and Applications, McGraw Hill, New York, 2001, pp. 400.
[8] S. Conti, A. O'Hagan, Bayesian emulation of complex multi-output and dynamic computer models, J. Stat. Plan. Inference 140 (3) (2010) 640–651.
[9] S. Conti, J.P. Gosling, J.E. Oakley, A. O'Hagan, Gaussian process emulation of dynamic computer codes, Biometrika 96 (2009) 663–676.
[10] C. Currin, T. Mitchell, M. Morris, D. Ylvisaker, Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments, J. Am. Stat. Assoc. 86 (416) (1991) 953–963.
[11] D.H. de Boer, M. Stone, L.M. Levesque, Fractal dimensions of individual flocs and floc populations in streams, Hydrol. Process. 14 (4) (2000) 653–667.
[12] A.I. Forrester, A. Sóbester, A.J. Keane, Multi-fidelity optimization via surrogate modelling, Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci. 463 (2007) 3251–3269.
[13] C. Fraser, N. McIntyre, B. Jackson, H. Wheater, Upscaling hydrological processes and land management change impacts using a metamodeling procedure, Water Resour. Res. 49 (9) (2013) 5817–5833.
[14] T.E. Fricker, J.E. Oakley, N.M. Urban, Multivariate Gaussian process emulators with nonseparable covariance structures, Technometrics 55 (1) (2013) 47–56.
[15] P.W. Goldberg, C.K. Williams, C.M. Bishop, Regression with input-dependent noise: a Gaussian process treatment, Adv. Neural Inf. Process. Syst. 10 (1997) 493–499.
[16] D. Head, Linear surface roughness growth and flow smoothening in a three-dimensional biofilm model, Phys. Rev. E 88 (3) (2013) 032702.
[17] D.A. Henderson, R.J. Boys, K.J. Krishnan, C. Lawless, D.J. Wilkinson, Bayesian emulation and calibration of a stochastic computer model of mitochondrial DNA deletions in substantia nigra neurons, J. Am. Stat. Assoc. 104 (485) (2009) 76–87.
[18] M. Henze, W. Gujer, T. Mino, T. Matsuo, M.C. Wentzel, G.v.R. Marais, M.C. Van Loosdrecht, Activated sludge model no. 2D, ASM2D, Water Sci. Technol. 39 (1) (1999) 165–182.
[19] D. Higdon, J. Gattiker, B. Williams, M. Rightley, Computer model calibration using high-dimensional output, J. Am. Stat. Assoc. 103 (482) (2008) 570–583.
[20] B. Iooss, P. Lemaître, A review on global sensitivity analysis methods, in: Uncertainty Management in Simulation – Optimization of Complex Systems, Springer, 2015, pp. 101–122.
[21] P. Jarvis, B. Jefferson, S.A. Parsons, Measuring floc structural characteristics, Rev. Environ. Sci. Biotechnol. 4 (1–2) (2005) 1–18.
[22] P.G. Jayathilake, P. Gupta, B. Li, C. Masden, O. Oyebamiji, R. González-Cabaleiro, S. Rushton, B. Bridgens, D. Swailes, B. Allen, S. McGough, P. Zuliani, I.D. Ofiteru, D. Wilkinson, J. Chen, T. Curtis, A mechanistic individual-based model of microbial communities, PLOS One 12 (8) (2017) p.e0181965, http://dx.doi.org/10.1371/journal.pone.0181965.
[23] M.C. Kennedy, C.W. Anderson, S. Conti, A. O'Hagan, Case studies in Gaussian process modelling of computer codes, Reliab. Eng. Syst. Saf. 91 (10) (2006) 1301–1309.
[24] K. Kersting, C. Plagemann, P. Pfaff, W. Burgard, Most likely heteroscedastic Gaussian process regression, Proceedings of the 24th International Conference on Machine Learning ACM (2007) 393–400.
[25] J.P. Kleijnen, E. Mehdad, Multivariate versus univariate Kriging metamodels for multi-response simulation models, Eur. J. Oper. Res. 236 (2) (2014) 573–582.
[26] J.P. Kleijnen, W.C. Van Beers, Robustness of Kriging when interpolating in random simulation with heterogeneous variances: some experiments, Eur. J. Oper. Res. 165 (3) (2005) 826–834.
[27] J.P. Kleijnen, Kriging metamodeling in simulation: a review, Eur. J. Oper. Res. 192 (3) (2009) 707–716.
[28] Y. Kuya, K. Takeda, X. Zhang, A.I.J. Forrester, Multifidelity surrogate modeling of experimental and computational aerodynamic data sets, AIAA J. 49 (2) (2011) 289–298.
[29] L. Le Gratiet, C. Cannamela, B. Iooss, Bayesian approach for global sensitivity analysis of (multifidelity) computer codes, SIAM/ASA J. Uncertain. Quantif. 2 (1) (2014) 336–363.
[30] R. Li, A. Sudjianto, Analysis of computer experiments using penalized likelihood in Gaussian Kriging models, Technometrics 47 (2) (2005) 111–120.
[31] J.D. Martin, T.W. Simpson, Use of kriging models to approximate deterministic computer models, AIAA J. 43 (4) (2005) 853–863.
[32] J.D. Martin, T.W. Simpson, Use of kriging models to approximate deterministic computer models, AIAA J. 43 (4) (2005) 853–863.
[33] S. Mitri, J.B. Xavier, K.R. Foster, Social evolution in multispecies biofilms, Proc. Natl. Acad. Sci. U. S. A. 108 (Suppl. 2) (2011) 10839–10846.
[34] B.-J. Ni, F. Fang, W.-M. Xie, M. Sun, G.-P. Sheng, W.-H. Li, H.-Q. Yu, Characterization of extracellular polymeric substances produced by mixed microorganisms in activated sludge with gel-permeating chromatography, excitation–emission matrix fluorescence spectroscopy measurement and kinetic modeling, Water Res. 43 (5) (2009) 1350–1358.
[35] J. Oakley, A. O'Hagan, Bayesian inference for the uncertainty distribution of computer model outputs, Biometrika 89 (4) (2002) 769–784.
[36] J.E. Oakley, A. O'Hagan, Probabilistic sensitivity analysis of complex models: a Bayesian approach, J. R. Stat. Soc. Ser. B: Stat. Methodol. 66 (3) (2004) 751–769.
[37] I.D. Ofiţeru, M. Bellucci, C. Picioreanu, V. Lavric, T.P. Curtis, Multi-scale modelling of bioreactor–separator system for wastewater treatment with two-dimensional activated sludge floc dynamics, Water Res. 50 (2014) 382–395.
[38] A. O'Hagan, Bayesian analysis of computer code outputs: a tutorial, Reliab. Eng. Syst. Saf. 91 (10) (2006) 1290–1300.
[39] I.G. Osio, C.H. Amon, An engineering design methodology with multistage Bayesian surrogates and optimal sampling, Res. Eng. Des. 8 (4) (1996) 189–206.
[40] A.M. Overstall, D.C. Woods, Multivariate emulation of computer simulators: model selection and diagnostics with application to a humanitarian relief model, J. R. Stat. Soc. Ser. C: Appl. Stat. 65 (2016) 483–505, http://dx.doi.org/10.1111/rssc.12141.
[41] O.K. Oyebamiji, N.R. Edwards, P.B. Holden, P.H. Garthwaite, S. Schaphoff, D. Gerten, Emulating global climate change impacts on crop yields, Stat. Model. 15 (6) (2015) 499–525, http://dx.doi.org/10.1177/1471082X14568248.
[42] O.K. Oyebamiji, Statistical Emulation for Environmental Sustainability Analysis (Ph.D. thesis), The Open University, 2014.
[43] H.D. Patterson, R. Thompson, Recovery of inter-block information when block sizes are unequal, Biometrika 58 (3) (1971) 545–554.
[44] C. Picioreanu, M.C. Van Loosdrecht, J.J. Heijnen, et al., Mathematical modeling of biofilm structure with a hybrid differential-discrete cellular automaton approach, Biotechnol. Bioeng. 58 (1) (1998) 101–116.

[45] C. Picioreanu, M.C. van Loosdrecht, J.J. Heijnen, et al., A theoretical study on the effect of surface roughness on mass transport and transformation in biofilms, Biotechnol. Bioeng. 68 (4) (2000) 355–369.
[46] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, J. Comput. Phys. 117 (1) (1995) 1–19.
[47] T. Pourmohamad, H.K. Lee, et al., Multivariate stochastic process models for correlated responses of mixed type, Bayesian Anal. 11 (3) (2016) 797–820.
[48] J. Qui nonero-Candela, C.E. Rasmussen, A unifying view of sparse approximate Gaussian process regression, J. Mach. Learn. Res. 6 (December) (2005) 1939–1959.
[49] C.E. Rasmussen, Gaussian Processes for Machine Learning, MIT Press, Cambridge, 2006.
[50] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and analysis of computer experiments, Stat. Sci. (1989) 409–423.
[51] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis: The Primer, John Wiley & Sons, 2008.
[52] T.J. Santner, B.J. Williams, W.I. Notz, The Design and Analysis of Computer Experiments, Springer Science & Business Media, 2013.
[53] J.D. Svenson, T.J. Santner, Multiobjective Optimization of Expensive Black-Box Functions Via Expected Maximin Improvement, The Ohio State University, Columbus, Ohio, 2016, pp. 32.
[54] M. Van Oijen, A. Thomson, F. Ewert, Spatial upscaling of process-based vegetation models: an overview of common methods and a case-study for the UK, Methods 1 (2009) 3.
[55] O. Wanner, Mathematical Modeling of Biofilms, IWA Pub., 2006.
[56] H. Wheater, B. Reynolds, N. McIntyre, M. Marshall, B. Jackson, Z. Frogbrook, I. Solloway, O. Francis, J. Chell, Impacts of Upland Land Management on Flood Risk: Multi-Scale Modeling Methodology and Results from the Pontbren Experiment, in: FRMRC Research Report UR 720 16, Imperial College & CEH Bangor, 2008.
[57] J. Wingender, T.R. Neu, H.-C. Flemming, What are bacterial extracellular polymeric substances? in: Microbial Extracellular Polymeric Substances, Springer, 1999, pp. 1–19.
[58] P.C. Young, M. Ratto, Statistical emulation of large linear dynamic models Technometrics 53 (1) (2011) 29–43, http://dx.doi.org/10.1198/TECH.2010.07151.
[59] B. Zhang, B.A. Konomi, H. Sang, G. Karagiannis, G. Lin, Full scale multi-output Gaussian process emulator with nonseparable auto-covariance functions, J. Comput. Phys. 300 (2015) 623–642.

Dr. Oluwole Oyebamiji received his PhD in Statistical modelling from The Open University, UK in 2014. He also received Master of Philosophy in Statistics & Modelling Science from Strathclyde University, UK in 2011. He is currently a Research Associate in the School of Mathematics & Statistics, Newcastle University, UK. His research interests include Bayesian methods, multivariate analysis, experimental design, statistical modelling and big data analytics.

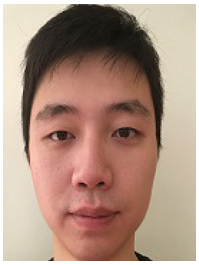


Darren Wilkinson is Professor of Stochastic Modelling at Newcastle University and co-Director of the EPSRC Centre for Doctoral Training in Cloud computing for big data. He is a Bayesian statistician with research interests in computational inference for complex stochastic models with applications to systems biology.

Dr. Jayathilake Pahala Gedara is currently employed as a research associate at School of Mechanical \& Systems Engineering, Newcastle University, UK. His main area of expertise is numerical modelling and simulation of biological systems. He is currently working on multi-scale modelling of biofilms. Dr. Jayathilake obtained his PhD in numerical modelling of permeable capsules in Stokes flows from National University of Singapore (NUS) in 2010.

Thomas Curtis is Professor of Environmental Engineering at Newcastle University and EPSRC Dream Fellow. He joined the University of Newcastle in 1994 after a Master and PhD in Public Health Engineering from Leeds University. He has done significant work in the interaction between water, waste, the environment and health. His current research works cut across the need to harness the new generation of biology and microbial ecology to the development of science and technological tools required for the treatment of water and waste.

Stephen Rushton graduated from Oxford in 1977 before moving onto undertake a PhD at the University of East Anglia where he studied the population dynamics of invertebrates, developing population models to investigate regulation. He then became a Post Doc at Newcastle modelling plant animal and then eventually microbial populations and communities. His key expertise in developing spatio-dynamic models for living organisms. He has published approximately 200 refereed papers, has a H-index of 50 and interests that span biomedical research to natural history.



Bowen Li is a PhD candidate in the School of Computing Science at Newcastle University. He received his Master's Degree in Computer Security and Resilience from Newcastle University in 2011. He worked as a Research Assistant on the EPSRC-funded UNCOVER project (2013-2016). Currently, he is continuing his research as a Research Assistant on EPSRC-funded NUFEB project. His relevant research interests include system modelling, model checking for systems biology and partial order methods of concurrent systems.

Dr. Prashant Gupta is currently employed as a scientist at Procter \& Gamble Technical Centres Ltd, Newcastle Upon Tyne. His main area of expertise is modelling and simulation of motion/evolution of assembly of discrete particles/organisms. During the writing of this manuscript, Dr. Gupta worked at School of Biology as a postdoctoral researcher working on project NUFEB. He worked on multi-scale modelling of biofilm modelling: building a modelling framework describing physics and biology of the bacterial colonies at different length and temporal scales. Dr. Gupta obtained his PhD in modelling hydrodynamics of multiphase flows from the University of Edinburgh in 2014.