Statistical emulation as a tool for analysing complex computer model outputs & upscaling

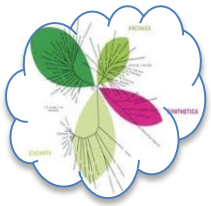
Oluwole Oyebamiji
School of Mathematics & Statistics,
Newcastle University, UK

NUFEB
Newcastle University
9th May, 2016

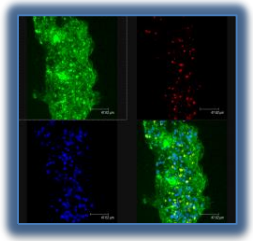
Aims of talk

- Objectives: NUFEB project
- Simulator and emulator
- Key attributes of LAMMPS model
- Procedure for building an emulator
- Gaussian process (Kriging)
- Relevant LAMMPS model outputs for emulation
- Results and conlusion

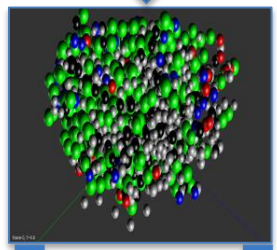
NUFEB groups



Synbio

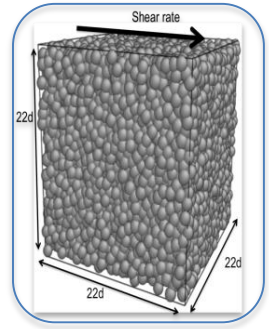


Ecology and Evolution

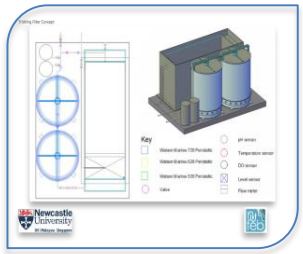


IB Model

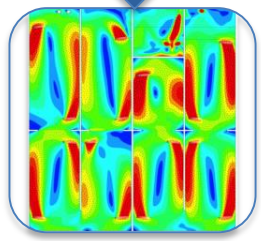
Emulator



Flow and Mech



Pilot Plant



CFD

Flow and Mech

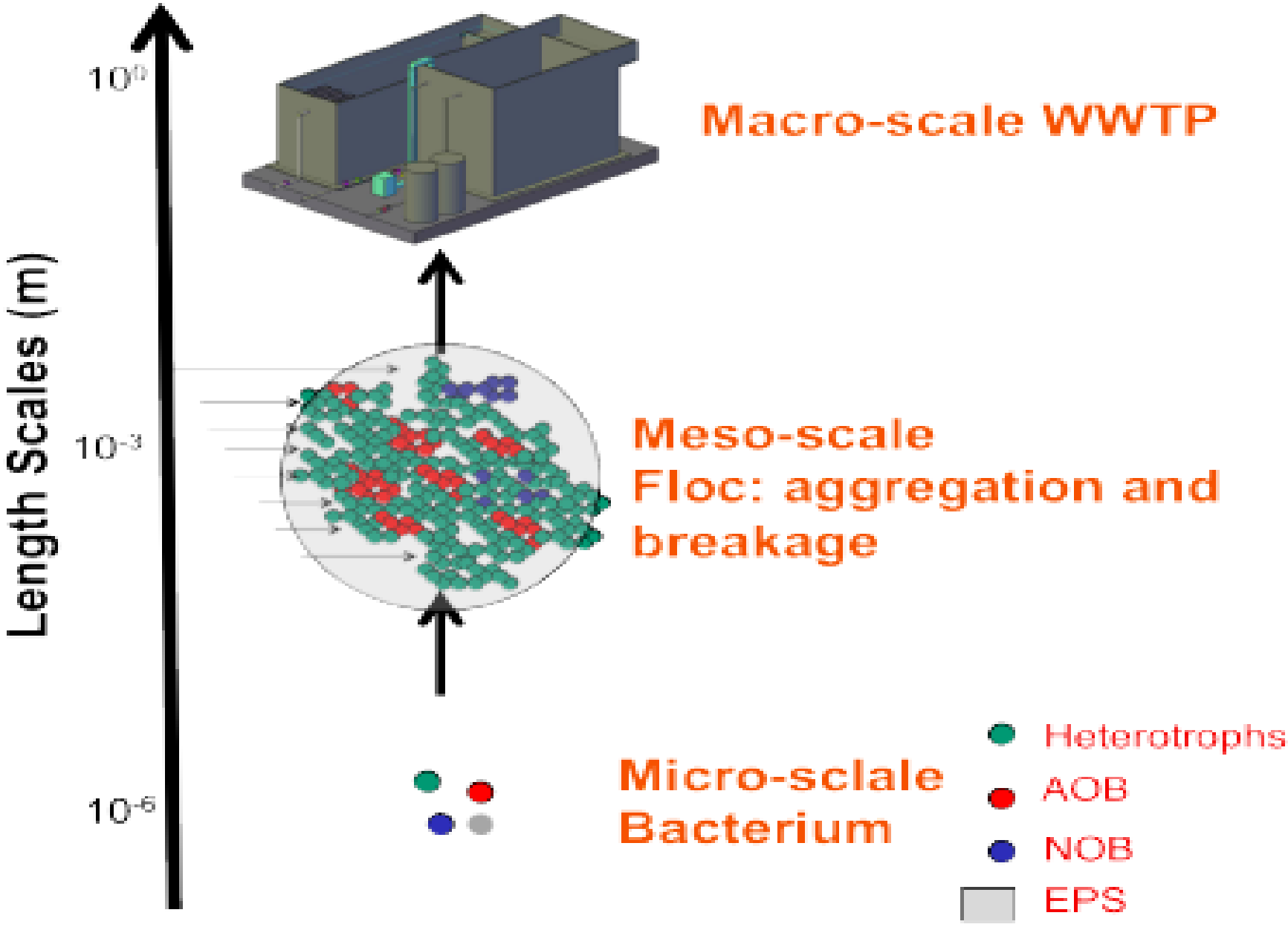
Overview

- The ultimate goal of NUFEB is to develop techniques for building computer models of real, complex physical systems, and for using such models to understand better how to control and manipulate real systems with minimal physical experimentation
- One of the crucial aspects of the NUFEB project is the simulation of biological floc/biofilm models that scales from one level to another to study how to effectively manage real systems.
- A large number of bacteria that ranges from order 10^{12} to 10^{18} individual particles and are physically and genetically complex
- The models are computationally expensive and can involve a computer run of several days.
- Computationally cheaper models can be obtained from large ensembles of simulations using statistical emulation

Conceptual framework

- The macroscale characteristics of wastewater treatment plants are the consequences of microscale features of a vast number of individual particles that produce the biofilm/floc
- There is a need to understand the interactions of microbes at fine resolution based models that could provide the best available representation of micro scale responses
- The challenge then becomes how we can transfer this small scale information to the macroscale process in a computationally efficient way

Scaling up diagram



Simulator and emulator

- A simulator is a model of a real process. Typically implemented as a computer code
- A function taking inputs $\mathbf{x}$ and giving outputs $\mathbf{y}$

$$\mathbf{y} = f(\mathbf{x})$$

- An emulator statistical approximation to the simulator
- It is a fast and cheaper surrogate for a computationally expensive computer model
- Expressing knowledge/beliefs about what the output will be at any given input(s)
- Built using prior information and a training set of model runs
- The GP emulator expresses f as a GP conditional on Hyper-parameters
- Use mainly for interpolation; sensitivity analysis; uncertainty analysis; calibration purpose etc

Key attributes of LAMMPS model

- Expensive to evaluate - we can not run them at every parameter combination of interest, which limits the amount of information
- The LAMMPS model is stochastic
- The LAMMPS model is dynamic
- The model produces high-dimensional inputs and multiple outputs
- Despite all these caveats, the good news is that there is a large knowledge base addressing these problems

Building an emulator

- Screening: which simulator inputs matter; what are plausible input range; constraints in the input combinations; elicit beliefs about input distributions
- Experimental Design: where to run the simulator.
- Output behaviour: stochastic/ deterministic; multiple outputs; multi-scale emulator
- Model structure: mean and covariance functions
- Inference: Estimating parameters; building emulator
- Cross-validation: check validity of emulator

Emulator choice

- A Bayesian framework for emulation is based on the assumption that a Gaussian process prior distribution can be specified for unknown parameters and hyperparameters
- Gaussian processes (GPs) extend multivariate Gaussian distributions to infinite dimensionality. Statistical distribution Y(t), t T, for which any finite linear combination of samples has a joint Gaussian distribution. $Y \sim GP(m, K)$, where m $=$ mean, $K =$ covariance function
- The parameters are treated as random variables
- The given prior distribution can be updated from training data. Applying Bayes rule to this setting, a posterior distribution can be obtained.
- We are implementing GP technique in form of kriging because of its wide applicability and flexibility.

Kriging

- Kriging is a geostatistical technique for interpolating the value of an unknown random observation from data $\mathbf{y}(\mathbf{x})$ observed at known locations.
- Let $X = x^1, \ldots, x^n$ denote the points where $\mathbf{y}$ has already been evaluated, and let alertunivariate output $\mathbf{y}$ be denoted as $y = (y(x^1), \ldots, y(x^n))$
- The model formulation universal kriging is given as

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

where $f(x) = H^T\boldsymbol{\beta}$; $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]$ is a $(p \times 1)$ vector of regression coefficients and H is a 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 $\varepsilon(\mathbf{x})$

Kriging continuation....

The best linear unbiased predictor for kriging model is given as

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

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

$$\Big(h(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T\mathbf{C}^{-1}\mathbf{t}(\mathbf{x})\Big)(\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\Big(h(\mathbf{x}')^T - \mathbf{t}(\mathbf{x}')^T\mathbf{C}^{-1}\mathbf{t}(\mathbf{x}')\Big)^T\Big\}$$

Parameter estimation

Using MLE for parameter estimation, we have

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

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]$ By taking the derivative of the LLK with respect to $\boldsymbol{\beta}$ and σ^2 and solving for zero, the estimates $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ are given respectively as $\hat{\boldsymbol{\beta}} = (\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{C}^{-1}\mathbf{y}$ and $\hat{\sigma^2} = \frac{1}{n}\left[(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}})^T\mathbf{C}^{-\mathbf{1}}(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}})\right]$

To estimate $\boldsymbol{\alpha}$, we maximize over $\boldsymbol{\alpha}$ and σ^2 the concentrated likelihood given below after plugging the values of $\hat{\boldsymbol{\beta}}$

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

Multivariate kriging

Multivariate case has m observation types

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

$$\mathbf{\Sigma}_Y = \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} & \cdots & \mathbf{\Sigma}_{1m} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} & \cdots & \mathbf{\Sigma}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Sigma}_{m1} & \mathbf{\Sigma}_{m2} & \cdots & \mathbf{\Sigma}_{mm} \end{pmatrix}$$

where $\mathbf{\Sigma}_{12}$ is the covariance between observations of type 1 and 2, $\mathbf{\Sigma}_{11}$ is the univariate variance matrix.

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

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

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

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

Multivariate.....

To estimate $\mathbf{\Sigma}_Y$ which must be positive definite, we assume separable covariance model of the form $\mathbf{\Sigma}_Y = \mathbf{\Sigma}_0 \mathbf{C}$, where $\mathbf{\Sigma}_0$ is the covariance matrix between observation types and $\mathbf{C}$ input space covariance function

The joint likelihood of $L(\mathbf{\Sigma}_Y, \mathbf{B})$ is obtained and maximized

Stochasticity

- Univariate - incorporate nugget terms in the form empirical variance derived from the repeated simulation data. The extension of above derivation for the noisy observations, covariance $K = \sigma^2 \mathbf{C}$ is replaced by $\sigma^2 \mathbf{C} + diag(\boldsymbol{\tau}_1^2, \ldots, \boldsymbol{\tau}_n^2)$, where $\boldsymbol{\tau}^2 = \tau_1^2, \ldots, \tau_n^2$ are the noise variances
- Multivariate - Follow **?** approach of emulating a scale response
- For a single output, the scale response is derived by repeating the simulation k times at each design point such that $\mathbf{y}' = \frac{\bar{\mathbf{y}} - \hat{f}}{\sigma^2/\sqrt{k}}$, where $\bar{\mathbf{y}}(x_i) = \frac{\sum_{j=1}^{k} \mathbf{y}_{ij}}{k}$, $\sigma^2(x_i) = \frac{\sum_{j=1}^{k} (\mathbf{y}_{ij} - \bar{\mathbf{y}})^2}{k-1}$ and $\hat{f}$ is estimate of main signal function.
- Apply kriging model to $\mathbf{Y} = [\mathbf{y}'_1, \ldots, \mathbf{y}'_m]$

Dynamic emulation

The model can be written as

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

Where $\mathbf{y}_{t-1}$ is the state vector at the previous time step for $t = 1, \ldots, T$, and $\mathbf{x}_t$ are the inputs at time t which includes the model parameters, forcing and initial conditions.

- Multi-step -
  - Fit multi-output emulation
  - Use time as an additional variable (single output emulator)
  - Emulate each time step
- Single-step - Develop a single step emulator and use the emulator repeatedly to generate the full-time series of the resulting predictions up to the number of desired time points
- Follow **?** approach, starting from initial run of the model at time t_0, we construct the single step emulator $\mathbf{y}_1 = f(\mathbf{x}_1, \mathbf{y}_0)$ using a GP regression in form of kriging

Dynamic emulation

- Proceed sequentially, feeding back the entire output distribution from the GP model, such that at time step $t = 1$, for input $(\mathbf{x}_1, \mathbf{y}_0)$, we sample from the distribution of $f(\mathbf{y}_0, \mathbf{x}_1)$, the model output is given as

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

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

Dynamic emulation.....

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

- Simulate $\tilde{\mathbf{y}}_{t+1}^{(s)}$ from conditional distribution $\left(f(.)|\mathbf{y}_t, \hat{\theta}\right)$.
- Repeat the entire process many times to obtain $\tilde{\mathbf{Y}}^N = \left[\tilde{\mathbf{y}}_1^{(s)}, \ldots, \tilde{\mathbf{y}}_{T-1}^{(s)}\right]^N$, for $s = 1, \ldots, N$, where $\tilde{\mathbf{Y}}^N$ is a sample from the joint distribution of $[\mathbf{y}_1, \ldots, \mathbf{y}_{T-1}]$ given the emulator training data and initial conditions and N is the number of Monte Carlo (MC) sample
- The technique is prone to numerical problems associated with the inversion of the covariance matrix

Normal approximation

There is a simple normal approximation to the above procedure according to **?**, the marginal distribution of $\mathbf{y}_t$ can be approximated as $\mathbf{y}_t \sim N\Big(\mu_t(.), \mathbf{K}_t(.)\Big)$ (defined earlier)

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

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

MC sampling to repeatedly revise the mean and variance

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

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

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

Simulation

- Design - LHS, with uniform distribution on 25 parameters
- Generate 1000 design points with 100 replicates
- LAMMPS run for 2days 172800s (results are recorded at every 2000s)
- Characterize the emerging floc and biofilm shapes

Parameters

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

Floc characterization

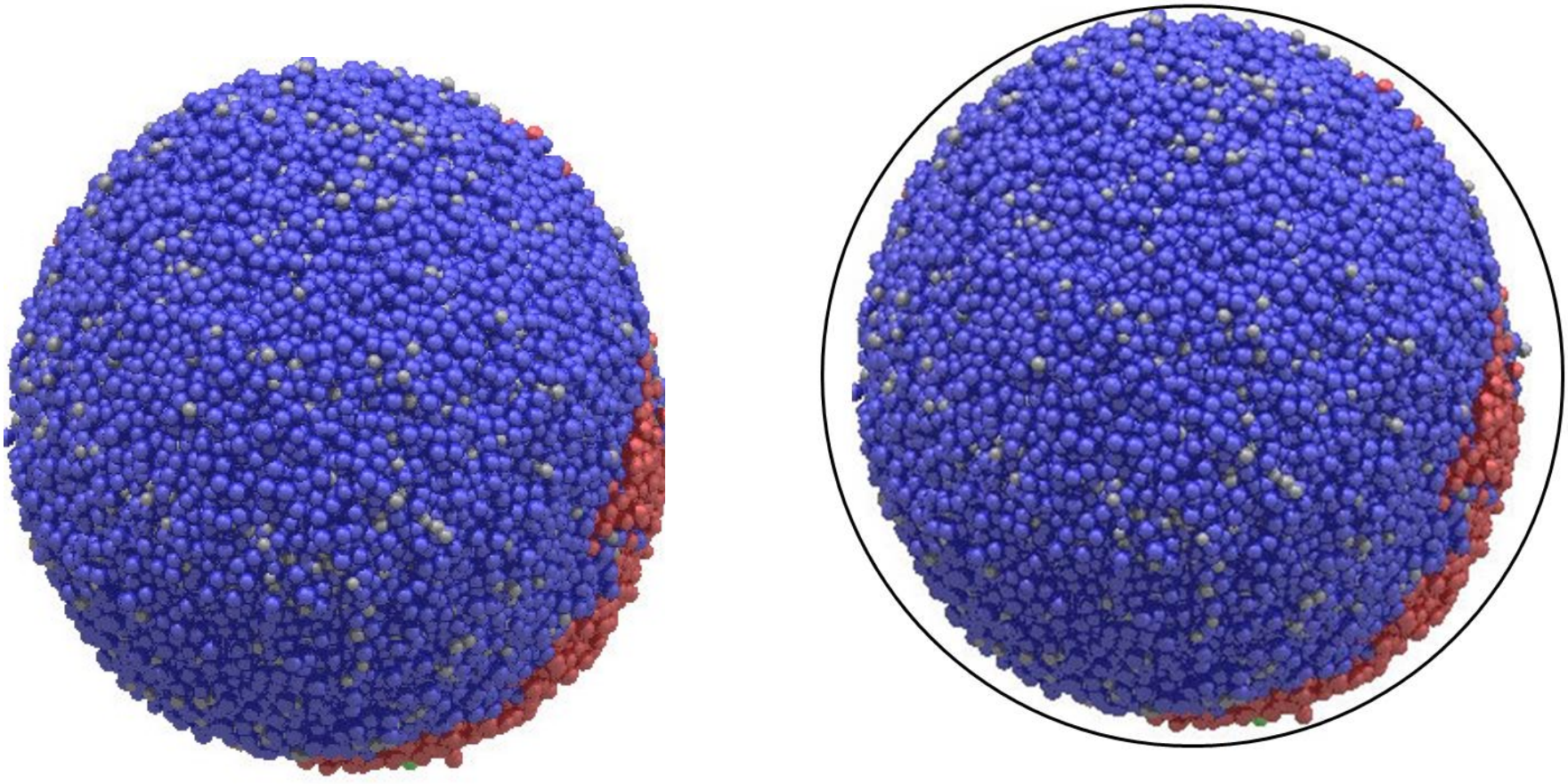
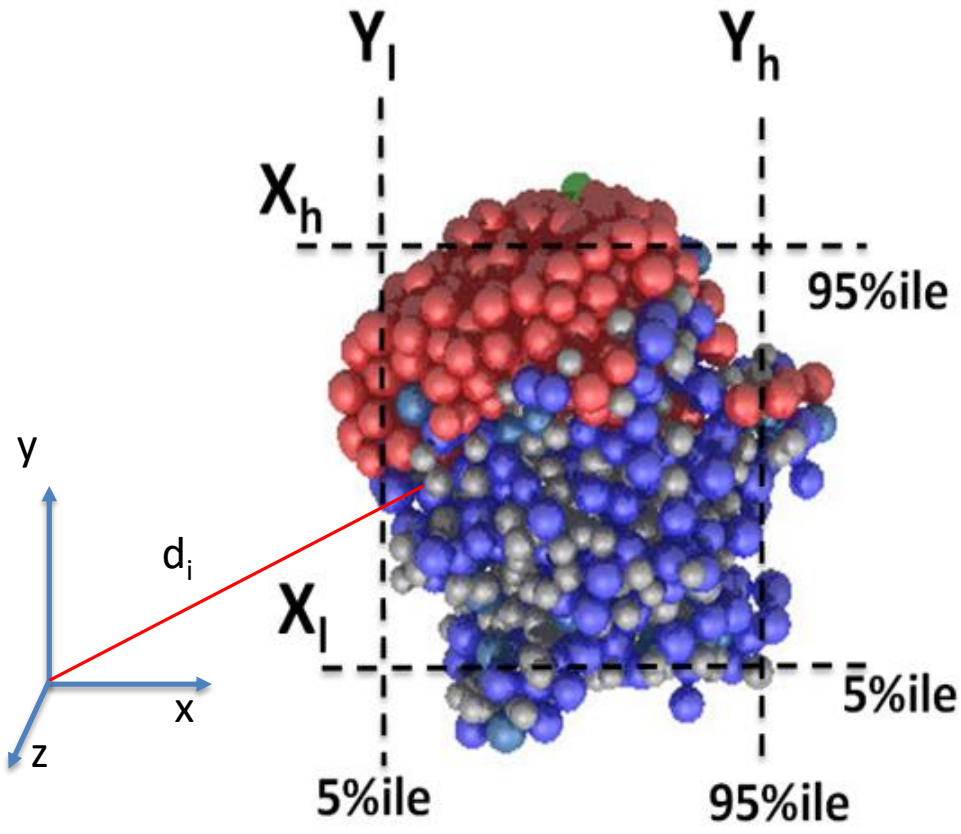


Figure: Transformation of microscale particles to floc at the mesoscale for a particular time. Floc equivalent diameter is the diameter of the smallest sphere that circumscribes the outline of the projected floc.

Fractal characterization



Radius of agglomerate = $R_a = \sqrt{\frac{\sum_{i=1}^{n} m_i d_i^2}{\sum_{i=1}^{n} m_i}}$

Mean radius of particles = $R_m = \frac{\sum_{i=1}^{n} r_i}{n}$

Fractal dimension = $\frac{\ln(R_a / R_m)}{\ln(n)}$

X_l=mean of the 5%ile particle coordinates

X_h=mean of the 95%ile particle coordinates

$X_{span} = X_h - X_l$

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

Relevant LAMMPS outputs for emulation

Flocs are aggregation of microbes mixed with an adhesive material called EPS

- Biofilm / floc total mass at each time step
- Biofilm / floc equivalent diameter at each time step
- EPS total mass at each time step
- Total number of particles at each time step
- The mass ratio of individual particle to the total biofilm / floc mass
- The distribution of floc / biofilm diameter

Result 1

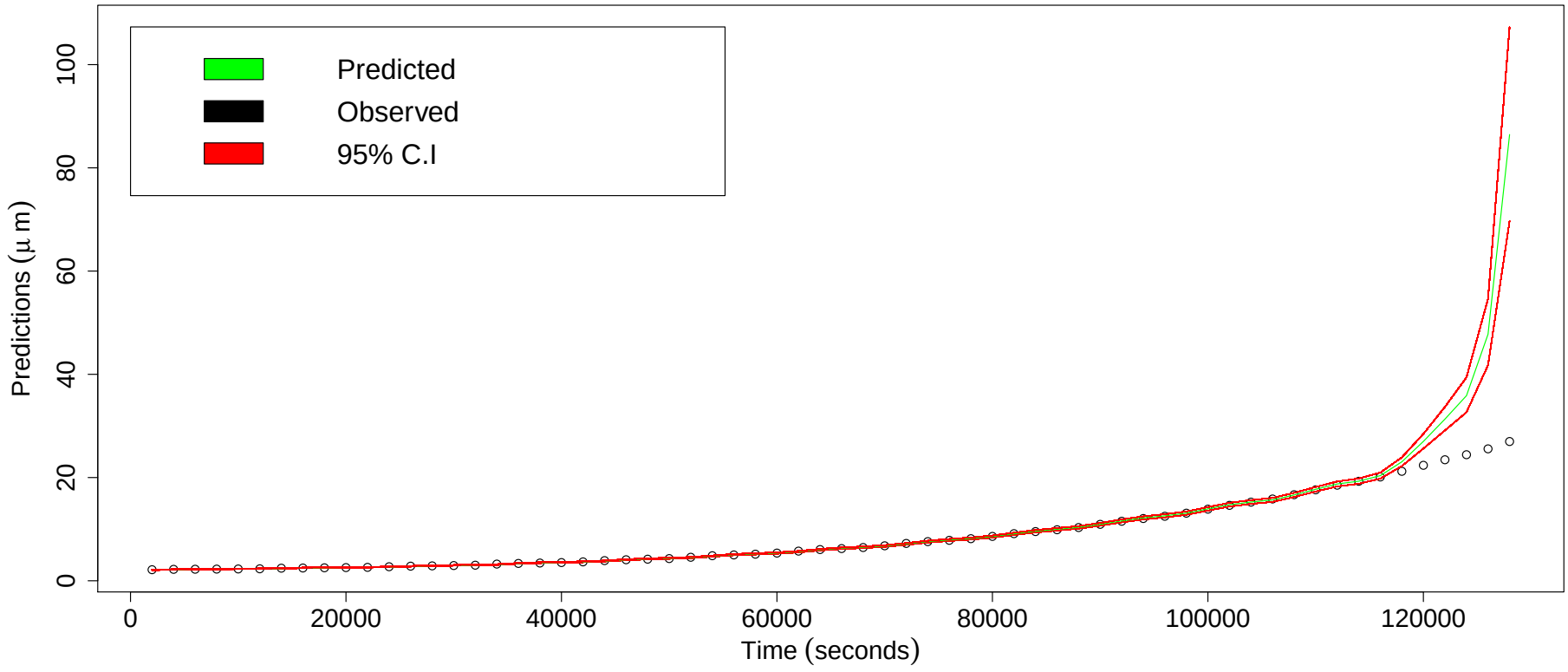


Figure: Comparison between floc diameter for LAMMPS model and emulator with 95% C.I over time for a selected point

Result 2

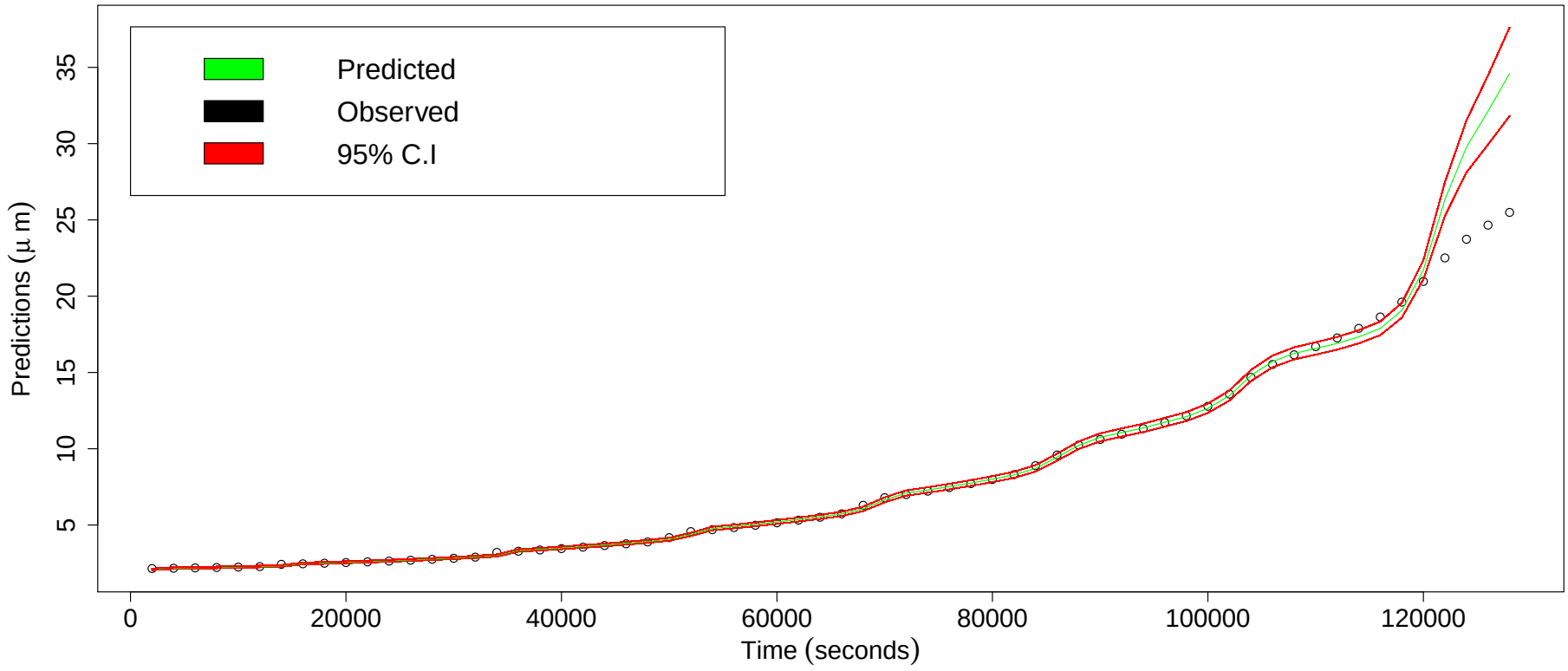


Figure: Comparison between floc diameter for LAMMPS model and emulator with 95% C.I

pdf plot

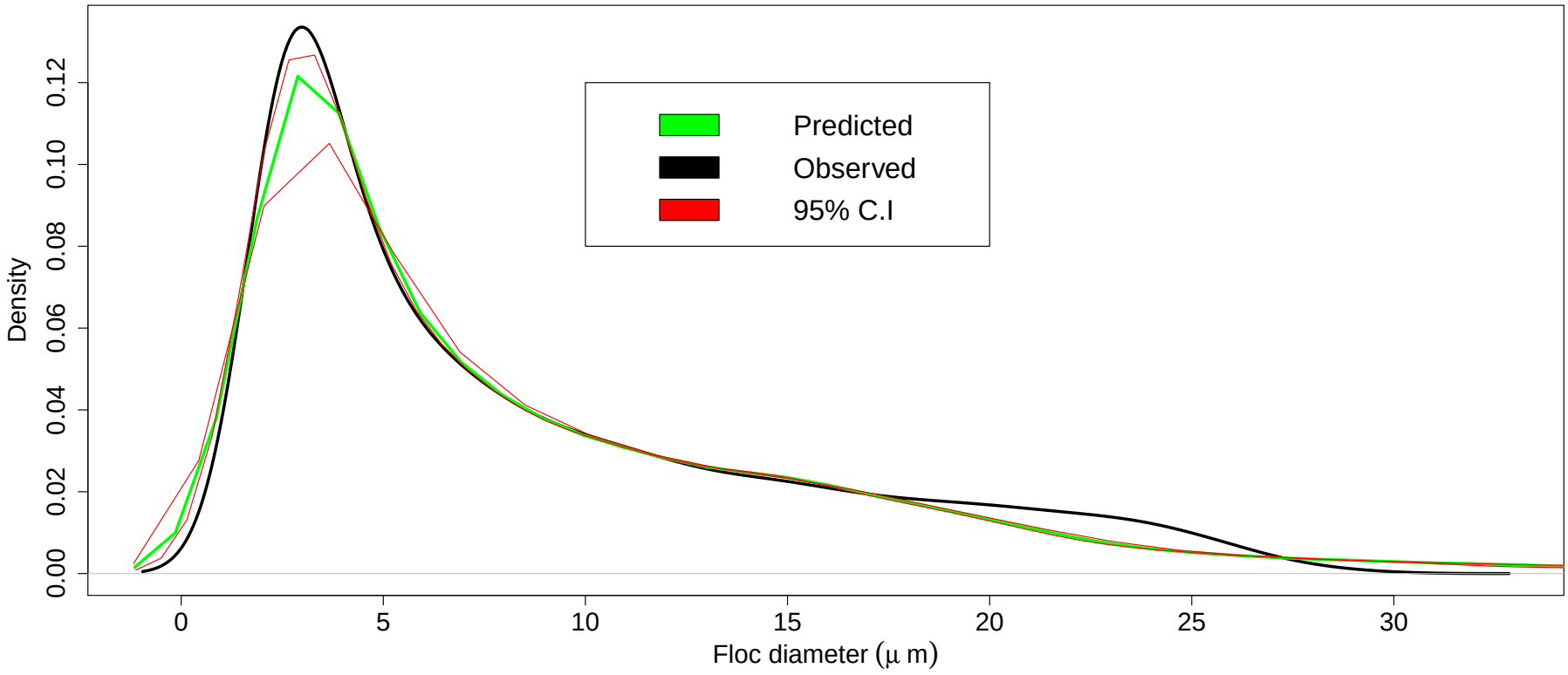


Figure: Probability density function of floc diameter for LAMMPS model and emulator with 95% C.I

Conclusion

- It takes LAMMPS model between (13-24 hours) to obtain 2 days simulation ensembles on 8G ram, 4-cores Linux machine
- Emulator gives results almost instantaneously (1 minute)
- This is 1100-fold increase in computational efficiency
- Explore other relevant statistical techniques that can handle big data emulation

References

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

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