Chapter 1

Emulation of LAMMPS outputs

We describe the procedure for building a simple emulator of LAMMPS output.

1.1 Experimental design

This section describes the procedure for generating the parameter combinations and variables at which the LAMMPS model is run. We run the LAMMPS code for a small sample of inputs using a Latin Hypercube Design (LHD). The design produces data for training the LAMMPS emulator to approximate the major LAMMPS outputs. LHD provides a good coverage of the input space with a relatively small number of design points. We use maximin LHS technique that optimises samples by maximizing the minimum distance between design points Santner et al. (2003). Suppose we want to sample a function of p variables, the range of each variable is divided into n reasonable intervals, n sample points are then drawn such that a Latin Hypercube is created.

We generate an $n \times p$ variables Latin Hypercube sample matrix with values uniformly distributed on interval [0,1]. We then transformed the generated sample to the quantile of a uniform distribution using the range of the parameters given in Table 3.1. LAMMPS model is computationally demanding, we limit our initial analysis to just $n = 100$ training points for each dimension of the $p = 22$ input spaces that are to be varied.

1.1.1 Simulation data

We describe two different simulation procedures. Firstly, let the design matrix which contain the input to the LAMMPS model be denoted by $D = (\theta_p^i, t_p, p = 1, \ldots, 22; i = 1, \ldots, 100)$, where the subscript p represents the 22 input parameters and superscript i denote 100 different realisations (design points), t_p is the time in seconds at which the output data is recorded. The design matrix $D_{100 \times 23}$ denotes the input values at which the LAMMPS model will be run for every combinations of x_p where x_p represents

(a) time t=0)

(b) time t=100000

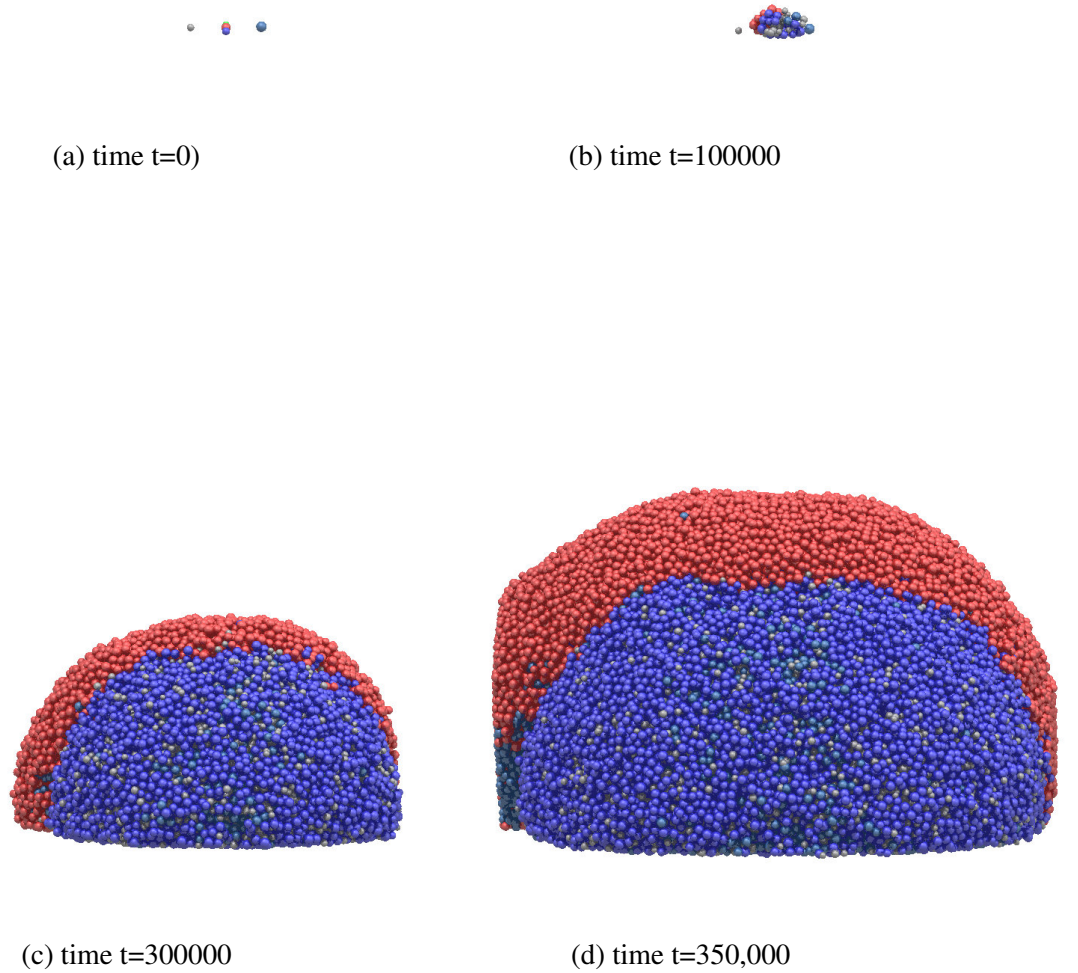(c) time t=300000

(d) time t=350,000

Figure 1.1: Sample of LAMMPS simulation data for particle growth for 4 different time points (seconds).

p^{th} row of D. The current LAMMPS code is set up to produce the following outputs namely particle diameter, position (3-dimensional), velocity (3-dimensional) and force

(3-dimensional). The LAMMPS code could be run for as long as there are sufficient computing resources to store the outputs. For the present analysis, the code is run for 352800 seconds to generate sufficient data for the emulation that is equivalent to ≈ 4 days of real time and generated output results are saved at a time-step of 2000 seconds which gives about 176 different time slices.

Therefore, a single run of the code for each of design point x_p will produce ten different outputs and 176-time steps. There are 5 different types (or species) of the particle in the simulation namely AOB, NOB, HET, EPS and inert. We note that the shape and number of particles as well as the composition of the floc at each time step vary in the simulation.

We perform the second simulation using the same input configuration as described above but repeated the runs for five times to incorporate stochastic variations in our outputs. The large design, of course, increases the amount of CPU time for the entire simulations even with running the bash script in parallel on a Linux machine.

1.2 Method 1: Combination of linear models and Gaussian process regression

1.2.1 Procedure

We describe the analysis of the first dataset here. The work focuses on predictions of mean floc diameter from LAMMPS simulation outputs together with their associated uncertainty levels. We propose a two-stage approach similar to O'Hagan (2006) where we shall combine a linear model in the first stage and a Gaussian process regression for residual interpolation in the second stage.

The step by step procedure for emulating one of the outputs of LAMMPS model, "particle floc diameter" is given here. We start by computing the average diameter of all the particles at each time step to obtain a vector $\mathbf{y}(x) = \bar{z}_1, \ldots, \bar{z}_{176}$ of the mean floc diameter to be emulated such that

$$\bar{z}_t = \frac{\sum_{k=1}^{N} z_{kt}}{N} \tag{1.1}$$

where $k = 1, \ldots, N$, N represents the total number of all 5 particles at each time slice. We also note that N varies across time point and in particular increases with time as seen in Figure 1.1 and z_{kt} is a simulation floc diameter for particle k at time t.

1.2.2 Stage 1: Linear models

For the training data, we sub-sample 75 out of 176-time slices for each design point x_p which gives a total of 7500 observations. The training data are (7500×1) vector of mean floc diameter $\mathbf{y}$ of LAMMPS output and (7500×23) matrix $\mathbf{X}$, which holds the values taken by the explanatory variables. (NOTE: sequence of time t at which output is sought is used as an additional input to the emulator). We fit a linear model to data $[\mathbf{y}, \mathbf{X}]$ and then obtain the predictions $\hat{\mathbf{y}}$. We also use the fitted model to predict the left out observations $\tilde{\mathbf{y}}^{new}$ for cross-validation purpose and to obtain standard error of predictions $\mathbf{s}^{new}$, which is a measure of prediction uncertainty in the linear model.

In order to construct emulator for the prediction of mean particle diameter, we apply a standard regression to the data in the first stage using equation (1.2) and use a Gaussian process regression for interpolating unexplained residual from linear model

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon = \beta_0' + \beta_1' x_1 + \ldots + \beta_p' x_p + \varepsilon \tag{1.2}$$

where $\mathbf{y}$ is the LAMMPS simulated mean floc diameter, $p = 23$ is the number of parameters for estimation and $\mathbf{x}_1, \ldots, \mathbf{x}_p$ are independent variables and $\beta_1', \ldots, \beta_{23}'$ are regression coefficients. We assume $\varepsilon \sim N(0, \sigma^2)$.

1.2.3 Summary of steps taken for stage 1 modelling

 (i) Design of experiment to identify important outputs (which outputs are to be emulated).

 (ii) Determine the relevant input factors to be varied (use all the 22 factors for now).

 (iii) Assign uniform probability density to each input variable (we have little knowledge about these parameters).

 (iv) Generate LHS of 100 design points on each of the 22 input parameters and transform into the quantile of uniform distribution using parameter ranges in Table 3.1.

 (v) Evaluate LAMMPS model for various input combinations to obtain the training data.

 (vi) Compute the LAMMPS mean floc diameter for each time-step from output data

 (vii) Randomly sample 75 time points out of 176 for each design point

 (viii) Fit a linear model with equation (1.2) using all 23 parameters (time is included as an additional variable to the linear model)

(ix) Obtain both predictions and standard error of predictions for all data points

1.2.4 Stage 2: GP modelling of residual data

The (7500×1) vector of residual $\varepsilon = \mathbf{y} - \hat{\mathbf{y}}$ obtained from stage 1 formed our new training data in this section. This residual can be modelled as $\varepsilon = \eta(X')$. We do not apply GP to all the 23 input variables in stage 1. Rather, we use standardized regression coefficients of equation (1.2) as a measure of sensitivity to select important explanatory variables for the GP regression such that $X' << X$ (ie p is now < 23). Since Gaussian process is described completely by its mean and covariance functions, the mean function is given as

$$E[\eta(X')|\beta] = h^{\mathrm{T}}(\mathbf{X}')\beta, \tag{1.3}$$

where $h(\mathbf{X}')$ is a vector of regression functions. For our analysis, we use a simple linear function $h(\mathbf{X}') = (1, \mathbf{x}^T)$ and β is an unknown hyperparameter to be estimated and covariance function

$$K = Cov[\eta(\mathbf{x}), \eta(\mathbf{x}')|\sigma^2, \alpha] = \sigma^2 \mathbf{C}(\mathbf{x}, \mathbf{x}^{\mathrm{T}}),$$

where σ^2 is a noise variance and $\mathbf{C}(\mathbf{x}, \mathbf{x}^{\mathrm{T}})$ is a correlation function with an hyperparameter α. We use a Gaussian correlation function of the form

$$\mathbf{C} = \left\{ \exp(-(x - x')^T \alpha(x - x')) \right\}, \tag{1.4}$$

where α is the correlation hyperparameters to be estimated from the data. It is difficult to apply a GP directly to the entire residual data because of the inversion of covariance matrix which scales cubically with the number of observations $O(N^3)$, we proceed by sub-sampling just 200 random observations from the data and denote the training data as $[\mathbf{y}, \mathbf{X}']$, where $\mathbf{y} = [\varepsilon_1 = \eta(x_1), \ldots, \varepsilon_{200} = \eta(x_{200})]$. We select just 4 important explanatory variables for the GP modelling and our correlation parameter is denoted as $\alpha = (\alpha_1, \ldots, \alpha_4)$.

For computing the posterior distribution, we assign a non-informative prior to the β, σ^2 and α parameters such that $p(\beta, \sigma^2) \sim \sigma^{-2}$ and $p(\alpha) \sim 1$. To obtain the posterior estimates using Bayes theorem, we update these prior distributions on some data $\mathbf{y}$. We estimate their values by conditioning on each parameter and maximize the resulting marginal likelihood function (Andrianakis & Challenor, 2009; Hankin, 2005). Therefore, the resulting posterior distribution is denoted as $\eta(.)|\mathbf{y} \sim N\left\{\mu^\star, K^\star\right\}$, where

$$\mu^\star(\mathbf{x}) = h(\mathbf{x})^T \hat{\beta} + t(\mathbf{x})^T \mathbf{C}^{-1}[\mathbf{y} - \mathbf{H}\hat{\beta}] \tag{1.5}$$

$$K^\star = \hat{\sigma}^2 \Big\{ c(\mathbf{x}, \mathbf{x}') - t(\mathbf{x})^T \mathbf{C}^{-1} t(\mathbf{x}) +$$
$$\Big(h(\mathbf{x})^T - t(\mathbf{x})^T \mathbf{C}^{-1} t(\mathbf{x}) \Big) (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \Big(h(\mathbf{x}')^T - t(\mathbf{x}')^T \mathbf{C}^{-1} t(\mathbf{x}') \Big)^T \Big\} \quad (1.6)$$

where $t(\mathbf{x})^T = [c(\mathbf{x}, \mathbf{x}_1), \ldots, c(\mathbf{x}, \mathbf{x}_n)]$ and $\mathbf{H}^T = [h(\mathbf{x}_1), \ldots, h(\mathbf{x}_n)]$, see more details in the appendix. We analyse our results using *emulator* package in *R*.

Finally, let the GP posterior mean and variance estimates of the residual at a new input point x^{new} (cross-validation) be denoted as $\mu^{\star\star}$ and $K^{\star\star}$ respectively, and predictions and standard error of predictions from linear model as $\tilde{\mathbf{y}}^{new}$ and $\mathbf{s}^{new}$ respectively. The final predictions of mean particle diameter $\hat{\mathbf{y}}^{new}$ is given as

$$\hat{\mathbf{y}}^{new} = \tilde{\mathbf{y}}^{new} + \mu^{\star\star}(\mathbf{x}) \tag{1.7}$$

and total uncertainty is

$$\hat{\mathbf{s}}^{new} = \mathbf{s} + \sqrt{|K^{\star\star}|} \tag{1.8}$$

1.2.5 Summary of stage 2

(i) Perform the GP emulation of the unexplained residual data

(ii) Given the posterior density of α as defined in equation (A.10c) of Appendix 1, compute the MLE of α, such that equation (A.10c) is maximized.

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

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

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

(vi) Compute posterior mean estimate $\mu^\star(\mathbf{x})$ given in equation (1.5)

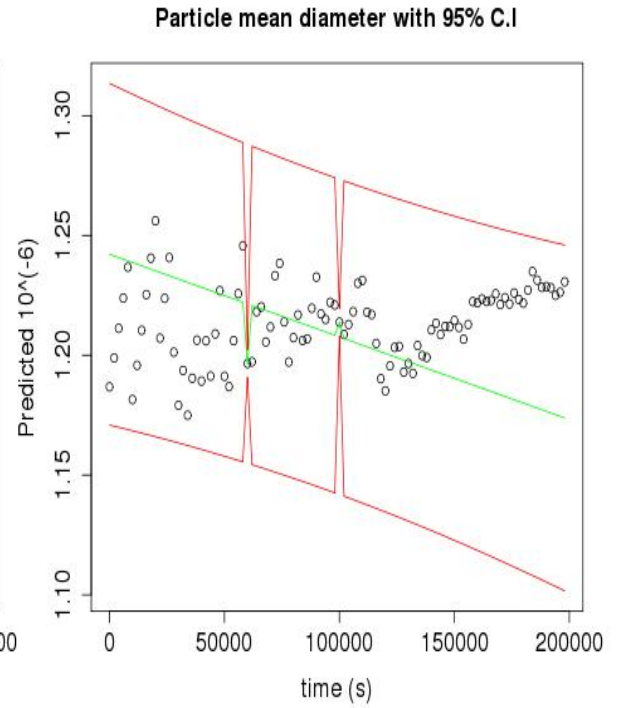(vii) Compute posterior variance estimate $K^\star$ given in equation (1.6) above
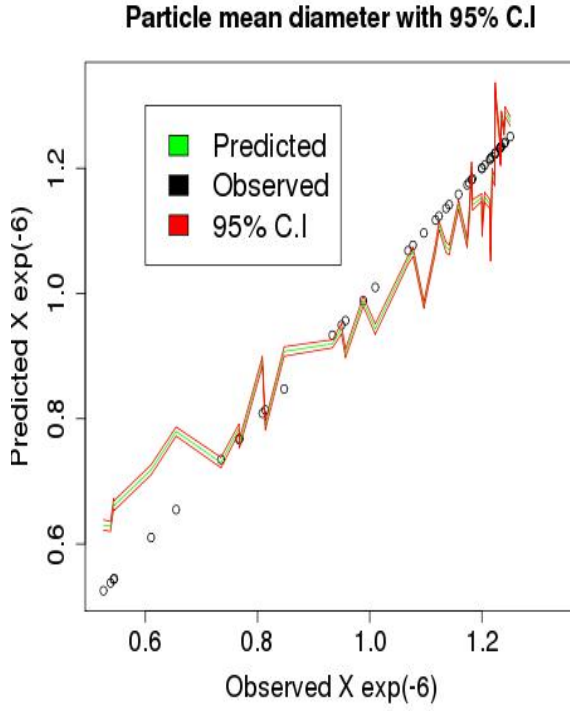
(a) LAMMPS and model predictions (emulator) for some randomly selected points
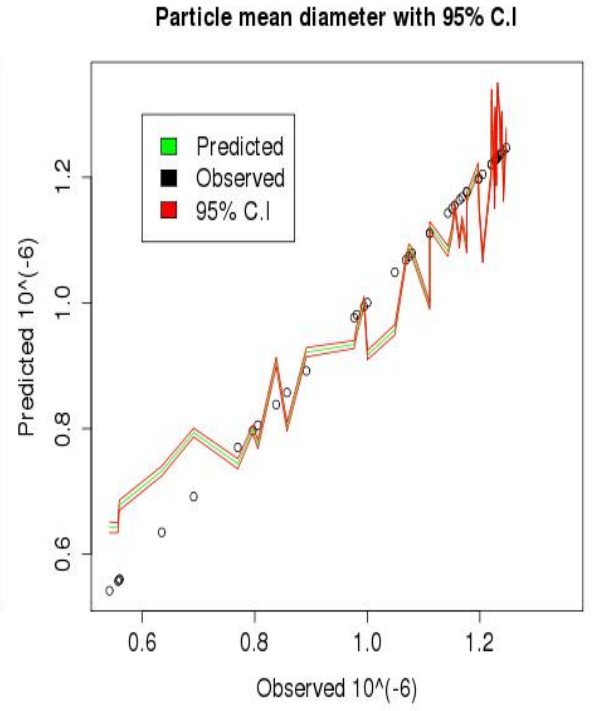
(b) LAMMPS and model predictions (emulator) for some randomly selected points
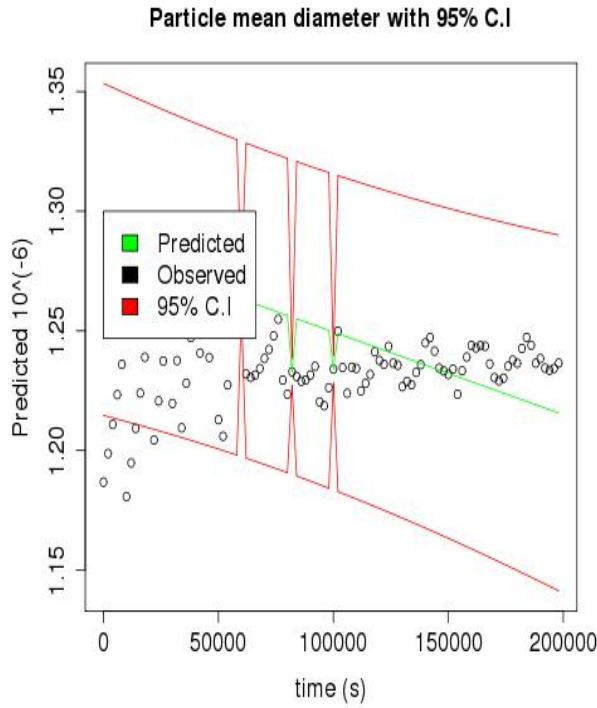
(c) LAMMPS and model predictions at 79th design point for all time steps

(d) LAMMPS and model predictions at 28th design point for all time steps

Figure 1.2: Pair plot of model predictions (emulator) vs. LAMMPS values with their 95% C.I for mean floc diameter (all species) from a two-stage technique

Chapter 2

Method 2: Kriging

Having obtained the simulation data by running the LAMMPS model as described in the previous section, we describe an alternative emulation approach for fitting LAMMPS output data. This approach involves using a kriging model, where the two stage techniques describe in previous section are combined as a single step. Kriging is a geostatistical technique for interpolating the value of an unknown random observation from data $\mathbf{y}(\mathbf{x})$ observed at known locations. They are also commonly used for building cheaper surrogate model of expensive computer codes Currin et al. (1991); Martin & Simpson (2004); Osio et al. (1996); Li & Sudjianto (2005). Here, $\mathbf{y}(\mathbf{x})$ can be decomposed into a mixture of deterministic (non-random trend) and a residual random variation. The trend could be modelled as a constant in ordinary/simple kriging or as an nth order polynomial as in universal kriging. In this section, we shall discuss the universal kriging technique. The model formulation is given as

$$\mathbf{y}(\mathbf{x}) = f(\mathbf{x}) + \varepsilon(\mathbf{x}) \tag{2.1}$$

where $\mathbf{y}(\mathbf{x})$ is as defined in equation (1.2) above, $\varepsilon(x)$ is a stationary Gaussian process with mean zero and known covariance function K. The deterministic function $f(\mathbf{x})$ is the mean approximation of the expensive computer simulator (eg LAMMPS) and f is a polynomial function. Under this assumption, $f(x)$ can be modelled as

$$f(\mathbf{x}) = \sum_{j=1}^{p} \beta_j h_j(x) = \mathbf{H}(x)\beta \tag{2.2}$$

$\beta = [\beta_1, \ldots, \beta_p]$ is a $(p \times 1)$ vector of unknown regression coefficients and $\mathbf{H}(x) = [h_1(x), \ldots, h_p(x)]^T$ is a $(n \times p)$ matrix of regression functions, $\varepsilon(\mathbf{x})$ is a stochastic Gaussian process characterized by its covariance function $K = Cov(\varepsilon(\mathbf{x}), \varepsilon(\mathbf{x})) = \sigma^2 \mathbf{C}(\mathbf{x}, \mathbf{x}')$, where σ^2 denotes the variance of $\varepsilon(\mathbf{x})$ also called process variance and $\mathbf{C}$ is a $(n \times n)$ positive definite matrix of correlation between $\varepsilon(\mathbf{x})$'s at the experimental design points. Similarly, $\mathbf{t}(x^{new}) = [Cor(x_1, x^{new}), \ldots, Cor(x_n, x^{new})]^T$ for the $(n \times 1)$ vector of corre-

lations between the $\varepsilon(\mathbf{x})$'s at the design points and new input points x^{new}. We use both Gaussian equation (1.4) and Matern correlation functions. The universal kriging predictor $\hat{\mathbf{y}}^{(x^{new})}$ of the value of $\mathbf{y}(x)$ at the new target point x^{new} is the linear predictor

$$\hat{\mathbf{y}}(x^{new}) = \sum_{j=1}^{n} \lambda_j \mathbf{y}(x_j) = \lambda(x)^T \mathbf{y}, \tag{2.3}$$

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

$$MSE[\hat{\mathbf{y}}(x^{new})] = E[\lambda^T \mathbf{y} - \mathbf{y}(x^{new})]^2, \tag{2.4}$$

subject to the unbiasedness constraint $E[\lambda^T(x)\mathbf{y}] = E[\mathbf{y}(x^{new})]$. The mean squared error in equation (2.4) can be rewritten by substituting the value of $\mathbf{y}$ in equation (2.1), we thus have

$$\sigma^2[1 + c\lambda^T(x)\mathbf{C}\lambda(x) - 2\lambda^T(x)\mathbf{t}(x^{new})], \tag{2.5}$$

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

$$\hat{\mathbf{y}}_{uk}(x) = h^T(x)\hat{\beta} + \mathbf{t}^T(x)\mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\hat{\beta}) \tag{2.6}$$

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

$$\begin{aligned} S_{uk}^2 = \hat{\sigma}^2 \Big\{ &C(\mathbf{x},\mathbf{x}') - \mathbf{t}(\mathbf{x})^T\mathbf{C}^{-1}\mathbf{t}(\mathbf{x}) + \\ \Big(h(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T\mathbf{C}^{-1}\mathbf{t}(\mathbf{x})\Big)&(\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\Big(h(\mathbf{x}')^T - \mathbf{t}(\mathbf{x}')^T\mathbf{C}^{-1}\mathbf{t}(\mathbf{x}')\Big)^T \Big\}. \end{aligned} \tag{2.7}$$

See more details in (Sacks et al., 1989; Santner et al., 2003)

NOTE: Kriging technique is equivalent to the Bayesian method describes in section (1.2.4) if we assign improper uniform priors on the β. In other words, non-informative Bayesian analysis often leads to kriging predictor and variance and these estimators appear respectively as conditional mean and variance in equations (1.5 and 1.6).

We use a Maximum Likelihood Estimation (MLE) technique as an objective estimator of the kriging model parameters, $\theta = (\beta, \sigma^2, \alpha)$. MLE assumes a Gaussian probability distribution. The likelihood of the model parameters is defined as the prob-

ability of the n observations $\mathbf{y} = y_1, , y_n$, given the model parameters such that

$$L(\theta|\mathbf{y}) = \Pi_{j=1}^n p(\mathbf{y}_j|\theta). \tag{2.8}$$

The expression for $L(\theta|\mathbf{y})$ is given by equation (A.10) below as

$$L(\beta, \sigma^2, \alpha; \mathbf{y}) \propto \frac{det(\mathbf{C})^{-\frac{1}{2}}}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{\frac{(\mathbf{y} - H\beta)^T \mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\beta)}{2\sigma^2}\right\} \tag{2.9}$$

where $\alpha = [\alpha_1, \ldots, \alpha_n]$ is a vector of correlation lengths and $det(\mathbf{C})$ is the determinant of correlation matrix $\mathbf{C}$. By taking the derivative of the log-likelihood of equation (A.10) with respect to β and σ^2 and solving for zero, the estimates $\hat{\beta}$ and $\hat{\sigma}^2$ are given respectively as $\hat{\beta} = (\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{C}^{-1}\mathbf{y}$ and $\hat{\sigma^2} = \frac{1}{n}\left[(\mathbf{y} - \mathbf{H}\hat{\beta})^T\mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\hat{\beta})\right]$.

In order to estimate α, we maximize over α the concentrated likelihood given below after plugging the values of $\hat{\beta}$ and $\hat{\sigma}^2$

$$-2logL(\hat{\beta}, \hat{\sigma}^2, \alpha; \mathbf{y}) = nlog(2\pi) + nlog\hat{\sigma}^2 + log(det(\mathbf{C})) + n \tag{2.10}$$

The trend and covariance parameters θ are computed quickly and very efficiently by using a global optimiser which is based on the extension of efficient algorithm proposed in Park & Baek (2001) for likelihood maximization. See further details in (Roustant et al., 2012).

2.1 Kriging results

Some results from kriging technique are given here. Recall from above that our training data are (7500×1) vector of mean floc diameter $\mathbf{y}$ of LAMMPS output and (7500×23) matrix $\mathbf{X}$ of input variables. Here, we use the same 75 out of 176-time slices but we sub-sample just 50 design points to reduce further the dimension of the data which gives a total of 3750 observations. The new training data for the kriging model is $D = [\mathbf{y}_{3750 \times 1}, X_{3750 \times 23}]$. We fit the data using the *DiceKriging* package in *R*. We consider only the linear terms for the trend model (no quadratic terms). We compare both Matern and Gaussian covariance functions with little difference in their predictions. The performance of the kriging emulator is tested by using the fitted model to predict the left out data.

We also perform the sensitivity analysis of the mean particle floc diameter to identify relevant variables. We use a Sobol technique that involves decomposing total output variance as a summand of increasing dimensionality. We sampled 1000 observations from a uniform distribution for each of the 23 input variables in Table 1. We compute the sensitivity indices for three different methods. Bootstrapping was used

to compute 95% confidence intervals on the estimated indices. The *sensitivity* package was used for this analysis. The indices are shown in Figure 2.2 and we can see that only 4 parameters are relevant. Finally, we refitted the kriging model using just these four important parameters, some visual plots of results comparing the LAMMPS simulations with the emulator are showing in Figure 2.1. We also compute the total percentage of variance explained by the final model that is $\sim 99\%$.
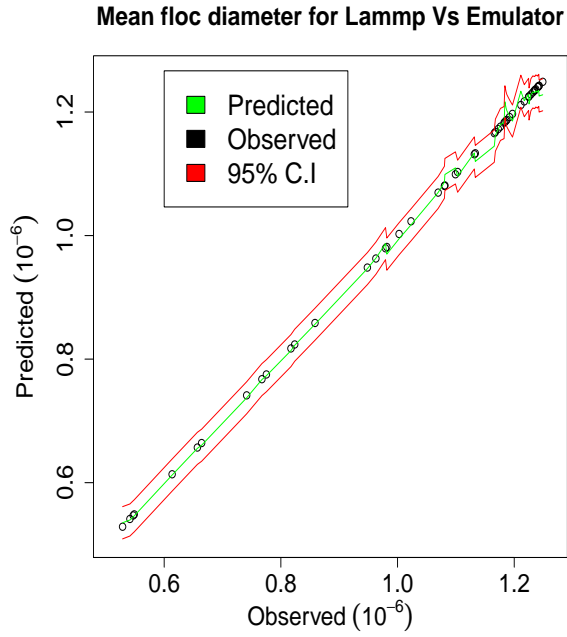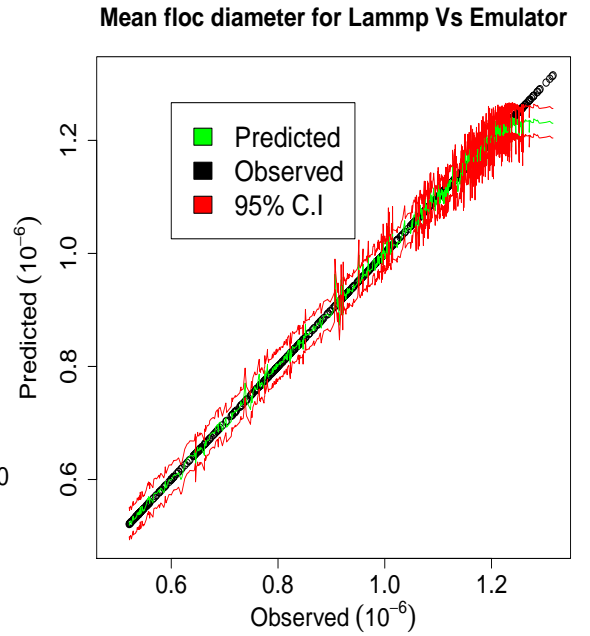
(a) LAMMPS and kriging predictions (emulator) for some randomly selected points
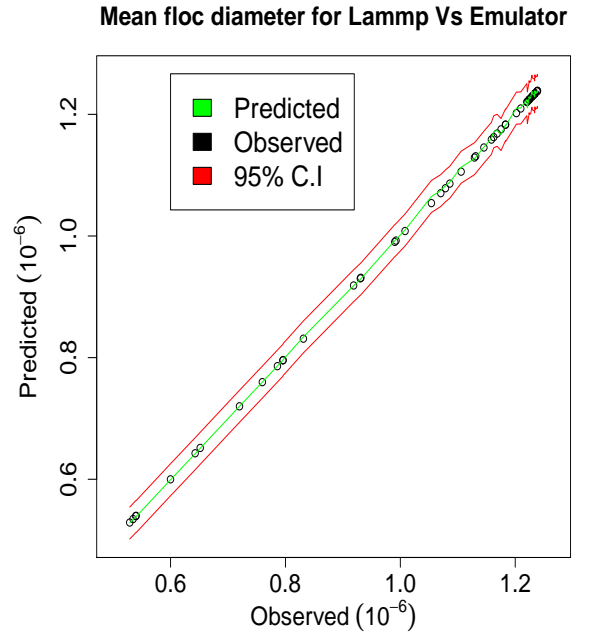
(b) LAMMPS and kriging predictions (emulator) for some randomly selected points
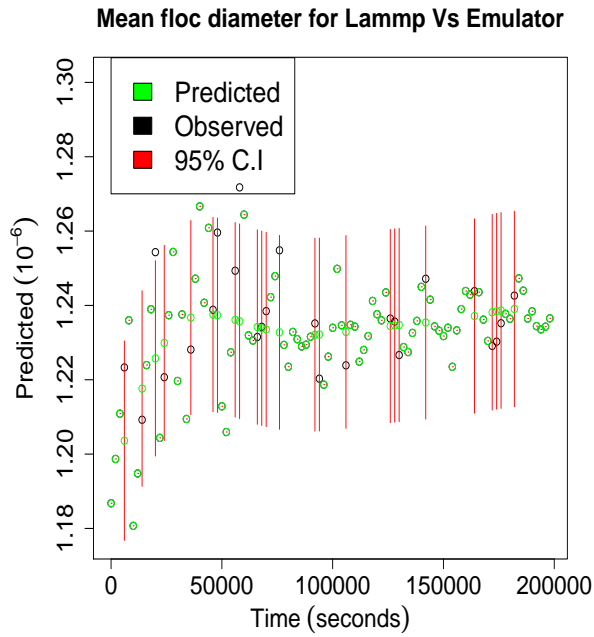
(c) LAMMPS and kriging predictions at 20th design point for all time steps. Note: The points with no C.I bands are the design points where MSE=0

(d) Pair plot for complete data set

Figure 2.1: Pair plot of kriging predictions (emulator) vs. LAMMPS values with their 95% C.I for mean floc diameter (all species) using a Gaussian covariance functions
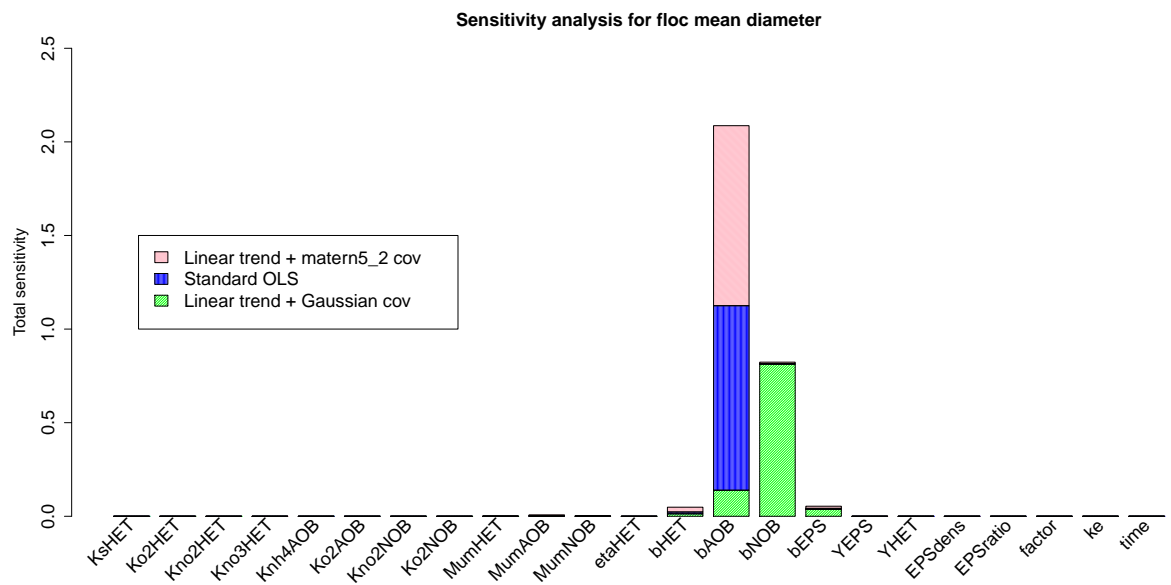
Figure 2.2: Barplots showing total sensitivity indices for mean floc diameter using Sobol based variance decomposition technique. Comparison for three different methods.

Chapter 3

Emulation of floc and biofilm

The detailed procedure of emulating the floc and biofilm will be described in this section.

References

Currin, C., Mitchell, T.J., Morris, M.D., and Ylvisaker, D. (1991). Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments. *Journal of the American Statistical Association*, $86(416), 953 - 963.$ 8

Martin, J. D., & Simpson, T. W. (2004). On the use of kriging models to approximate deterministic computer models. *In ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, $481 - 492.$ 8

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

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

Santner, T., Williams, B., Notz, W. (2003). The Design and Analysis of Computer Experiments. Springer. 1, 9, 16

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

Andrianakis, Y., & Challenor, P. G. (2009). Parameter estimation and prediction using Gaussian processes. *Technical report*, MUCM Technical Report 09/05, University of Southampton. 5, 16

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

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

Hankin, R. K. (2005). Introducing BACCO, an R package for Bayesian analysis of computer code output. *Journal of Statistical Software*, 14, 16. 5

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

Appendix 1: Estimation of prior hyperparameters

Noting that under GP regression, the prior distribution for the data is also a Gaussian distribution, the joint likelihood of the parameters is given as

$$p(\beta, \sigma^2, \alpha | \mathbf{y}) \propto \frac{det(\mathbf{C})^{-\frac{1}{2}}}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left\{\frac{(y - H\beta)^T \mathbf{C^{-1}}(\mathbf{y} - \mathbf{H}\beta)}{2\sigma^2}\right\} \tag{A.10}$$

where $\alpha = [\alpha_1, \ldots, \alpha_n]$ is a vector of correlation lengths and $det(\mathbf{C})$ is the determinant of correlation matrix $\mathbf{C}$, integrating out β using a non-informative (uniform) prior such that $p(\beta) \propto 1$. We have a marginal likelihood

$$p(\mathbf{y} | \sigma^2, \alpha) \propto \frac{\det(\mathbf{C})^{-\frac{1}{2}} \det(\mathbf{H^T C H})^{-\frac{1}{2}}}{(2\pi\sigma^2)^{\frac{n-p}{2}}} \exp\left\{\frac{(\mathbf{y} - \mathbf{H}\hat{\beta})^T \mathbf{C^{-1}}(\mathbf{y} - \mathbf{H}\hat{\beta})}{2\sigma^2}\right\}. \tag{A.10b}$$

Maximizing (A.10b) with respect to β and σ^2 will give

$$\hat{\beta} = (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C}^{-1} \mathbf{y} \tag{A.11}$$

$$\widehat{\sigma^2} = \frac{1}{n-p}\left[(\mathbf{y} - \mathbf{H}\hat{\beta})^T \mathbf{C^{-1}}(\mathbf{y} - \mathbf{H}\hat{\beta})\right] \tag{A.12}$$

Now, integrate out σ^2 such that $p(\sigma^2) \propto \frac{1}{\sigma^2}$, then we have

$$p(\mathbf{y} | \alpha) \propto \det(\mathbf{C})^{-\frac{1}{2}} \det(\mathbf{H}^T \mathbf{C} \mathbf{H})^{-\frac{1}{2}}\left\{(\mathbf{y} - \mathbf{H}\hat{\beta})^T \mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\hat{\beta})\right\}^{-(n-p)/2} \tag{A.10c}$$

$$= \det(\mathbf{C})^{-\frac{1}{2}} \det(\mathbf{H}^T \mathbf{C} \mathbf{H})^{-\frac{1}{2}} \widehat{\sigma^2}^{-(n-p)/2}.$$

The smoothing parameter α is estimated from the posterior distribution using the posterior mode by the value of α for which marginal likelihood (A.10c) is maximised. Here, we describe briefly the maximisation of the posterior distribution of α which is reparametrized as $\tau = 2log(\alpha)$ to make it unconstrained optimisation (Andrianakis & Challenor, 2009; Santner et al., 2003). Therefore, function $f(\tau) = log(p(\mathbf{y} | \exp(\frac{\tau}{2})))$ can be optimised using a derivative-free numerical optimisation of Nelder-Mead method which is the default in the "emulator" package that we use.

Table 3.1: List of all the parameters

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