Emulation of Lammp outputs

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

Experimental design

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

We generate an $n \times p$ variables Latin Hypercube sample matrix with values uniformly distributed on interval [0,1]. We then transformed the generated sample matrix to the quantile of a uniform distribution using the range of the parameters given in Table 1. 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 because Lammp model are computationally demanding.

Simulation data

We describe 2 different simulation procedures. Firstly, let the design matrix which contain the input to the Lammp 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 Lammp model will be run for every combinations of x_p where x_p represents p^{th} row of D. The current Lammp code is set up to produce the following outputs namely particle diameter, position (3-dimensional), velocity (3-dimensional) and force (3-dimensional). The Lammp

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 which 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 10 different outputs and 176 time steps. There are 5 different types of 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 5 times in order to incorporate stochastic variations in our outputs. This of course increases the amount of CPU time for the entire simulations even with running the bash script in parallel on a Linux machine.

Procedure

The work focuses on predictions of mean floc diameter from Lammp simulation outputs together with their associated uncertainty levels. We proposed a two-stage approach where we combined a linear model in the first stage and a Gaussian process regression for residual interpolation in the second stage.

Here, we describe the procedure for emulating the particle floc diameter. 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

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

where $k = 1, \ldots, N$, N represents the total number of all 5 particles at each time slice and varies across time point and z_{kt} is a simulation particle k at time t. 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 Lammp 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 standard error of predictions $\mathbf{s}^{new}$ of which is a measure of prediction uncertainty in the linear model. Finally, we apply a GP to model the residual field obtaining from the linear model. To construct our emulator for the prediction of

floc diameter, we fit a linear model to the data in the first stage using equation (2) and apply a Gaussian process regression for residual analysis

$$\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\varepsilon} = \beta_0' + \beta_1'\mathbf{x}_1 + \ldots + \beta_p'\mathbf{x}_p + \boldsymbol{\varepsilon} \tag{2}$$

where $\mathbf{y}$ is the Lammp 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 $\boldsymbol{\varepsilon} \sim N(0, \sigma^2)$.

Stage 2: GP modelling of residual data

The (7500×1) vector of training residual $\boldsymbol{\varepsilon} = \mathbf{y} - \hat{\mathbf{y}}$ can be modelled as $\boldsymbol{\varepsilon} = \eta(X')$. We use standardized regression coefficients of equation (2) as a measure of sensitivity to select important explanatory variables for the GP regression such that $X' << X$ (p is now < 23). Since Gaussian process is described completely by its mean and covariance functions, the mean function is given as

$$E[\eta(X')|\boldsymbol{\beta}] = h^{\mathrm{T}}(\mathbf{X}')\boldsymbol{\beta}, \tag{3}$$

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 $\boldsymbol{\beta}$ is an unknown hyperparameter to be estimated and covariance function

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

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

$$\mathbf{C} = \Big\{ \exp(-(x - x')^T\boldsymbol{\alpha}(x - x')) \Big\} \tag{4}$$

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 $\mathrm{O}(N^3)$, we proceed by randomly sub-sampling just 200 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 explanatory variables for the GP modelling and our correlation parameter is denoted as $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_4)$.

We assign a non-informative prior to the $\boldsymbol{\beta}$ and σ^2 parameters and update these prior distributions with respect to some data $\mathbf{y}$, in order to obtain the posterior estimates using Bayes theorem. We estimate their values by conditioning on each parameter and maximize the resulting marginal likelihood function. Therefore, the posterior distribution of $\eta(.)|\mathbf{y} \sim N\Big\{\mu^\star, K^\star\Big\}$, where

$$\mu^{\star}(\mathbf{x}) = h(\mathbf{x})^T\hat{\boldsymbol{\beta}} + t(\mathbf{x})^T\mathbf{C}^{-1}[\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}}] \tag{5}$$

$$\begin{aligned} K^{\star} = \hat{\sigma}^2\Big\{c(\mathbf{x},\mathbf{x}') - t(\mathbf{x})^T\mathbf{C}^{-1}t(\mathbf{x}) + \\ \Big(h(\mathbf{x})^T - t(\mathbf{x})^T\mathbf{C}^{-1}t(\mathbf{x})\Big)(\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\Big(h(\mathbf{x}')^T - t(\mathbf{x}')^T\mathbf{C}^{-1}t(\mathbf{x}')\Big)^T\Big\} \end{aligned} \tag{6}$$

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 (crossvalidation) 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

$$\hat{\mathbf{y}}^{new} = \tilde{\mathbf{y}}^{new} + \mu^{\star\star}(\mathbf{x}) \tag{7}$$

and total uncertainty is

$$\hat{\mathbf{s}}^{new} = \mathbf{s} + \sqrt{|K^{\star\star}|} \tag{8}$$

These are summary of steps taken for emulating Lammp output:

(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 1.

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

(vi) Compute the Lammp mean floc diameter for each time-step

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

(viii) Fit a linear model with equation (2) using all 23 parameters (time included as additional variable to the linear model)

Table 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 EPSdens	30
20	variable EPSratio	1.25
21	variable factor	1.5
22	variable ke	5e+10

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

(x) Perform the GP emulation on the unexplained residual data

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

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

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

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

Maximizing (A.10b) with respect to $\boldsymbol{\beta}$ and σ^2 will give

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

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

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

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

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

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