Gaussian process emulation of particle method for estimating free-surface heights

Yoshiki MIZUNO* and Seiichi KOSHIZUKA*

* School of Engineering, The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
E-mail: mizuno@mps.q.t.u-tokyo.ac.jp

Received: 17 February 2020; Revised: 11 August 2020; Accepted: 27 October 2020

Abstract
This paper presents the development of a statistical emulator to estimate free-surface heights with less computational time than a particle method. Particle methods can simulate free-surface flow problems by solving Navier-Stokes and continuity equations, but they require more computational time as the number of particles becomes greater in computational domains. Accordingly, it is not pragmatic to conduct statistical analysis of free-surface problems with respect to a variety of initial conditions by particle methods. In the place of the simulation methods, statistical emulators can estimate predictive values in these problems with less computational time. In this study, we apply a Gaussian process for designing a statistical emulator of the Explicit Moving Particle Simulation (EMPS) method and predict free-surface heights in dam break problems. Once it is developed based on a dataset made from only one simulation run of a dam break problem, the Gaussian process emulator is able to approximate these heights in other dam break problems. By measuring the coefficient of determination, root mean squared error, and mean absolute error, we evaluate the accuracy of emulated free-surface heights in dam break problems where the shapes of water columns are distinct from the original shape at the initial condition. We alter the initial lengths in the $x$-direction and the initial heights in the $z$-direction remaining the same initial width in the $y$-direction. Consequently, in terms of the computational speed and the accuracy, it is demonstrated that we can adopt the Gaussian process emulator as a replacement of the EMPS simulator especially when free-surface flow analysis is repeatedly conducted with different initial conditions.

Keywords: Collapse of water columns, Computational speed, Explicit Moving Particle Simulation (EMPS) method, Gaussian process regression, Initial conditions, Numerical analysis, Statistical emulator

1. Introduction

Particle methods, such as the Smoothed Particle Hydrodynamics (SPH) method (Lucy, 1977; Gingold and Monaghan, 1977; Monaghan, 1992), the Moving Particle Semi-Implicit (MPS) method (Koshizuka and Oka, 1996; Koshizuka et al., 1998; 2018), and the Explicit Moving Particle Simulation (EMPS) method (Shakibaeinia and Jin, 2010; Oochi et al., 2010, 2011; Yamada et al., 2011) have advantages with simulating free-surface flow problems described by the Navier-Stokes and continuity equations. Compared to mesh-based methods, they can discretize these equations by particles without node connectivity information and solve them without a convective term by the Lagrangian description. Specifically, EMPS is one of the most appropriate candidates for simulating free-surface flow problems with less computational time. It can calculate all the terms of the Navier-Stokes equations explicitly and conduct parallel computing more easily (Oochi et al., 2011). Based on a distributed parallel SPH method (Ferrari et al., 2009), Murotani et al. (2012; 2014) developed a distributed memory parallel algorithm for the EMPS method and reported that it required approximately a week to simulate a tsunami problem on a supercomputer. Based on this study, Mizuno et al. (2019) designed a new algorithm for more time-efficient simulations on parallel environments and reduced calculation cost in hydrostatic and dam break problems. However, it is still computationally expensive to conduct statistical analysis of free-surface problems based on a variety of initial conditions by the EMPS method.
In the place of the EMPS simulator, a Gaussian process emulator is expected to play a significant role in approximating predictive values much faster. In this context, emulators indicate stochastic expression of deterministic simulators (Rougier, 2008; Rougier et al., 2009) and can be constructed based on a Gaussian process, a collection of random variables with a joint Gaussian distribution (Rasmussen and Williams, 2006). For instance, Igarashi et al. (2016) utilized an emulator adopting a Gaussian process to estimate maximum tsunami wave heights observed in the Kii Peninsula of Japan. Also, since emulators can output predictive values instantaneously, they are applied for statistical analysis such as uncertainty quantification and sensitivity analysis (Oakley and O’Hagan, 2002; 2004). In general, statistical analysis demands a considerable number of simulation runs, and thus it is not pragmatic to use simulators even if a single model run requires only a few minutes (O’Hagan, 2006). Accordingly, Sarri et al. (2012) applied a Gaussian process emulator for uncertainty quantification and sensitivity analysis of landslide-generated tsunami waves, and they illustrated that the computational time required by the emulator became much faster than the corresponding simulator.

In the present study, we design a Gaussian process emulator of EMPS to predict free-surface heights in dam break problems where the size of water columns are varied at the initial conditions. Once the Gaussian process emulator is developed based on a dataset made from only a single simulation run by the particle method, it is able to estimate free-surface heights in the dam break problems much faster than the simulator using the EMPS method. We conduct leave-one-out cross validation (Kohavi, 1995) on the emulator and measure the coefficient of determination ($R^2$), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) to confirm the accuracy of the emulated values.

Although the study focuses only on dam break problems, it aims to utilize the emulation technique for tsunami runup as future works. If the developed emulator can capture main features of free-surface heights in the problems, it is also applicable for the estimation of tsunami wave heights depending on initial wave forms that run up a slope in coastal regions. Moreover, if the emulated free-surface heights are substituted into a reference model suggested by Asakura et al. (2003), for example, we can approximate pressure loads generated by tsunamis as well. The emulator has the potential to emulate runup elevation and pressure values immediately after tsunamis happen due to earthquakes.

The remaining chapters are outlined as follows. Chapter 2 briefly explains the EMPS method that discretizes the Navier-Stokes and continuity equations. Chapter 3 illustrates a Gaussian process utilized for the development of an emulator. Chapter 4 clarifies the accuracy of the prediction by the statistical emulator and demonstrates the computational time necessary for the emulation relative to that of the simulation. Chapter 5 describes the concluding remarks about this study.

2. Explicit Moving Particle Simulation method

EMPS was developed by Shakibaeinia and Jin (2010) to discretize the following Navier-Stokes and continuity equations:

$$\frac{Dv}{Dt} = -\frac{1}{\rho} \nabla p + v \nabla^2 v + g,$$

$$\frac{Dp}{Dt} + \rho \nabla \cdot v = 0,$$

where $\frac{D}{Dt}$, $v$, $\rho$, $p$, $\nabla$, and $g$ denote the Lagrange derivative, the velocity vector, the fluid density, the pressure, the kinetic viscosity, and the vector of acceleration due to gravity, respectively.

EMPS differs from MPS in that EMPS assumes fluids are weakly compressible and computes all the terms of the Navier-Stokes equations explicitly. It represents the derivatives of particle $i$ as

$$\langle \nabla p \rangle_i = \frac{d}{n^0} \sum_{j=1}^{n^0} \left[ \frac{(p_j + p_i)(r_j - r_i)}{|r_j - r_i|^2} w(|r_j - r_i|) \right],$$

$$\langle \nabla^2 v \rangle_i = \frac{2d}{\lambda^0 n^0} \sum_{j=1}^{n^0} \left[ (v_j - v_i) w(|r_j - r_i|) \right],$$

$$\langle \nabla \cdot v \rangle_i = \frac{d}{n^0} \sum_{j=1}^{n^0} \left[ \frac{(v_j - v_i) \cdot (r_j - r_i)}{|r_j - r_i|^2} w(|r_j - r_i|) \right],$$

where $d$, $n^0$, $r$, $w$, and $\lambda^0$ number the dimensions, the initial value of a correction parameter, the position of particles, a weight function, and the particle number density in the initial setting, respectively. In the present study, $w$ is defined as

$$w(r) = \frac{r_i^2}{r} - 1,$$
where $r_e$ denotes the effective radius. Also, the particle number density $n_i$ for particle $i$ is determined as

$$n_i = \sum_{j \neq i} w(|r_j - r_i|). \tag{7}$$

The EMPS method explicitly calculates the pressure of particle $i$ as

$$p_i = c^2 \rho \left( \frac{n_i}{\rho^0} - 1 \right), \tag{8}$$

where $c$ signifies the sound speed adjusted for numerical stability. Finally, the correction parameter $\lambda$ is calculated as

$$\lambda = \frac{\sum_{i \neq j} |r_j - r_i|^2 w(|r_j - r_i|)}{\sum_{i \neq j} w(|r_j - r_i|)}. \tag{9}$$

3. Gaussian process for development of emulator

Compared to computer models or simulators, statistical emulators are effective techniques for approximating predictive values with less computational time (Rougier, 2008; Rougier et al., 2009). In the present study, we utilize a Gaussian process to construct an emulator in the place of an EMPS simulator for estimating free-surface heights in dam break problems. A Gaussian process is a collection of random variables, such that every finite number of which has a joint Gaussian distribution (Rasmussen and Williams, 2006). Here, the random variables indicate the value of $f(x)$ at location $x$.

A Gaussian process is identified by its mean function $\mu(x)$ and covariance function $k(x,x')$. If $\mu(x)$ and $k(x,x')$ of a real process $f(x)$ are defined as

$$\mu(x) = E[f(x)], \tag{10}$$

$$k(x,x') = E[(f(x) - \mu(x))(f(x') - \mu(x'))], \tag{11}$$

the Gaussian process (GP) is expressed as

$$f(x) \sim \mathcal{GP}(\mu(x), k(x,x')). \tag{12}$$

If $\mu(x)$ is set to zero for concise expression, a Gaussian process is only specified by its covariance function. We can express this function as the combination of kernels and characterize the tendency of free-surface heights in dam break problems. We adopt a squared exponential term for a falling trend, a rational quadratic term for irregularities, and a noise term.

First, the squared exponential term is represented as

$$k_1(x,x') = \theta_1^2 \exp \left( -\frac{(x-x')(x-x')^T}{2\theta_2^2} \right), \tag{13}$$

where $\theta_1$ and $\theta_2$ denote hyperparameters for the amplitude and the characteristic length-scale, respectively. Next, the rational quadratic term is expressed as

$$k_2(x,x') = \theta_3^2 \exp \left( 1 + \frac{(x-x')(x-x')^T}{2\theta_4^2\theta_5} \right)^{-\theta_5}, \tag{14}$$

where $\theta_3$, $\theta_4$, and $\theta_5$ signify hyperparameters for the magnitude, the length-scale, and the shape parameter characterizing the diffuseness of the length-scale, respectively. Based on these two kernels, the final covariance function becomes

$$k(x,x') = k_1(x,x') + k_2(x,x'). \tag{15}$$

If function values $y$ include some noise $\varepsilon$:

$$y = f(x) + \varepsilon, \tag{16}$$

the prior on the noisy observations is written as

$$\text{cov}(y_p, y_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq} \quad \text{or} \quad \text{cov}(y) = K(X, X) + \sigma_n^2 I \tag{17}$$
where $\sigma_n^2$ represents the variance of independently identically distributed Gaussian noise $e$. Also, $\delta_{pq}$ is a Kronecker delta which equals to 1 if and only if $p=q$ and 0 otherwise.

We can describe the joint Gaussian distribution of the observed target values $y$ and the function values $f_i$ at the test locations as

$$
\begin{pmatrix} y \\ f_i \end{pmatrix} \sim \mathcal{N} \left( \begin{bmatrix} 0 \\ K(X, X) \end{bmatrix} \begin{bmatrix} K(X, X) + \sigma_y^2 I \\ K(X, X) \end{bmatrix} \begin{bmatrix} K(X, X) \\ K(X, X) \end{bmatrix} \right),
$$

(18)

If there exist $n$ training points and $n_t$ test points, $K(X, X_i)$ signifies the $n \times n_t$ matrix of the covariances computed at all pairs of these points, and similarly for the other matrices $K(X, X)$, $K(X_i, X)$, and $K(X_i, X_i)$. The predictive equations for a Gaussian process are finally expressed as

$$
f_i | X, y, X_i \sim \mathcal{N}(\hat{f}_i, \text{cov}(\hat{f}_i)),
$$

(19)

$$
\hat{f}_i \doteq K(X_i, X)[K(X, X) + \sigma_y^2 I]^{-1} y,
$$

(20)

$$
\text{cov}(f_i) = K(X_i, X_i) - K(X_i, X)[K(X, X) + \sigma_y^2 I]^{-1} K(X, X_i).
$$

(21)

We can obtain the emulated values of free-surface heights in dam break problems from Eq.(20).

Note that the hyperparameters $\theta = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \sigma_y)$ are optimized by the log marginal likelihood (Rasmussen and Williams, 2006). Also, we select $\theta = (1.69 \times 10^{-1}, 2.33 \times 10^3, 2.80 \times 10^{-1}, 5.61 \times 10^{-1}, 3.74 \times 10^{-1}, 1.00 \times 10^{-5})$ conducting leave-one-out cross validation (Kohavi, 1995).

4. Emulation of free-surface heights

4.1. Dam break problem

We apply a Gaussian process emulator for predicting free-surface heights in dam break problems. Figures 1 and 2 display the initial configuration of an original dam break problem and its simulation result under the condition listed in Table 1, respectively. We construct an emulator based on a dataset made by this simulation. The simulation using EMPS outputs an instance of the dataset per 0.05 [s] and at the very beginning of the simulation. Since the simulation time is 2.00 [s], the total number of instances $k$ becomes 41 from 0.00 to 2.00 [s].

The instances include three-dimensional positions of each fluid particle and the corresponding simulation time. For example, as the number of the fluid particles is 3,381 in the original problem (refer to Table 1), the first instance has 3,381 three-dimensional fluid positions and the simulation time $t = 0.00$ [s]. However, it is not necessarily true that each instance holds 3,381 fluid particles because some of the particles bounce out of the container as illustrated in Fig.2.(d). Accordingly, we calculate the mean positions of the fluid particles and regard them as the position of the fluid particles at each instance instead. If $\bar{x}^{(k)}$, $\bar{y}^{(k)}$, and $\bar{z}^{(k)}$ denote the mean values of the $x$-position, $y$-position, and $z$-position in the $k$-th instance, respectively, we have the instance $X = [\bar{x}^{(k)}, \bar{y}^{(k)}, \bar{z}^{(k)}]$. For example, $\bar{z}^{(k)}$ is calculated as

$$
\bar{z}^{(k)} = \frac{1}{n^{(k)}} \sum_{i=1}^{n^{(k)}} z_i^{(k)},
$$

(22)

where $i$ signifies the index of the fluid particles, $z_i^{(k)}$ indicates the height of fluid particle $i$ in the $k$-th instance, and $n^{(k)}$ denotes the total number of the fluid particles in the $k$-th instance. Based on the input values of the instance $X$ and Eq.(20), the Gaussian process emulator outputs the emulated values of the mean free-surface heights $\bar{z}^{(k)}$ at test locations $\bar{z}^{(k)}$ corresponds to $\hat{f}_i$ in the equation). Unless otherwise specified, “the free-surface heights” indicates “the mean free-surface heights” in the rest of the paper. To verify the accuracy of the emulation consequences, the following section compares the emulated free-surface heights at the selected time steps with the simulated heights.

Table 1 Condition for original dam break problem

| Total number of particles | 19,136 |
| Number of wall particles | 15,755 |
| Number of fluid particles | 3,381 |
| Time range | 2.00 [s] |
| Time step | $1.00 \times 10^{-4}$ [s] |
| Particle distance | $2.00 \times 10^{-2}$ [m] |
| Kinematic viscosity | $1.00 \times 10^{-6}$ [m²/s] |
| Fluid density | $1.00 \times 10^3$ [kg/m³] |
| Effective radius | $4.20 \times 10^{-2}$ [m] |
| Gravity | $-9.81$ [m/s²] |
Fig. 1 Initial configuration of original dam break problem. The blue particles represent a water column, and the red particles are wall particles that express a cuboid container. The volume of the initial column is $0.24 \times 0.18 \times 0.50$ [m$^3$], and that of the container is $1.00 \times 0.20 \times 0.60$ [m$^3$]. (a) is observed from the $x$-$z$ direction, and (b) is viewed from the $y$-$z$ direction. The fluid particles collapse due to gravity from the initial position.

Fig. 2 Simulation result of original dam break problem. The water column collapses due to gravity in (a), move rightward in (b), and impact on the right wall in (c). Subsequently, the top of the fluid particles goes to the highest position in the $z$-axis in (d) and is bounced back in (e). The fluid particles return to the left direction in (f), and impact on the left wall in (g). They move rightward as depicted in (h) and (i) again.

4.2. Accuracy of emulation

The free-surface heights $\bar{z}(k)$ are influenced by the initial shapes of water columns in dam break problems. We vary these shapes and verify the accuracy of $\bar{z}(k)$ estimated by the constructed emulator. As shown in Fig.1, the length in the $x$-direction is 0.24 [m], the width in the $y$-direction is 0.18 [m], and the height in the $z$-direction is 0.50 [m] at the initial configuration of the original dam break problem. While remaining the same width in the $y$-direction, we alter the original length in the $x$-direction and/or the original height in the $z$-direction. As listed in Table 2, we use eight dam break problems (Cases A-H) for the verification. As well as the original instances, Cases A-H hold 41 instances: $X = \{\bar{x}(k), \bar{y}(k), \bar{z}(k)\}$ for $k = 1, 2, ..., 41$. In Case A and Case B, only the lengths in the $x$-direction are changed. In Case C and Case D, only the heights in the $z$-direction are varied. In Case E-H, both the lengths and heights are altered based on the ratio of the initial length (0.24 [m]) to the initial height (0.50 [m]). Since this ratio is about 1 to 2, the lengths and heights are also fixed at the ratio of 1 to 2 in Case E-G. Conversely, the length-to-height ratio is set at 1 to 3 in Case H so as to check how these
ratios influence the accuracy of the free-surface heights. The table also indicates the volumes of the water columns in the respective cases.

We measure the coefficient of determination ($R^2$), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) to evaluate the accuracy in Cases A-H. Table 3 describes the results. First, Case B records a greater value of $R^2$ and smaller values of RMSE and MAE compared to those of Case A. This comes from the fact that the $x$-length in the Case B (0.30 [m]) is closer to that of the original case (0.24 [m]) than that of Case A (0.16 [m]). Second, Case C has a slightly greater value of $R^2$ relative to that of Case D. It also results in smaller values of RMSE and MAE, which indicates that Case D has more errors in some emulated values. This is because the height in Case C (0.24 [m]) is more similar to that of the original case (0.50 [m]) than that of Case D (0.64 [m]). Finally, the results of Cases E-H clarify that the ratios of the lengths to the heights do not necessarily play an important role in determining the accuracy of the emulated free-surface heights. Despite the fact that the ratio in Case G is mostly identical with that of the original case and the same with that of Case E and Case F (1 to 2), the values of $R^2$, RMSE, and MAE in Case G become 0.292, 0.045, and 0.042, respectively. However, they are 0.682, 0.029, and 0.027 in Case H where the length-to-height ratio is 1 to 3. As a matter of fact, the volume of the water column in Case G (0.032 [m$^3$]) is the most different from that of the original case (0.022 [m$^3$]), and the result of Case G is worse than any other cases including that of Case H. Conversely, since the volume in Case F (0.021 [m$^3$]) is mostly the same as the original volume, the consequence of Case F is in the best agreement with that of the original case. If the difference is smaller between the original volume and those of Case E-H, the value of $R^2$ becomes greater, and the values of RMSE/MAE are smaller.

From the different points of view, Figs.3-5 compare the 41 emulation results of the free-surface heights at the time range with those of the simulation in Cases A-H. Figure 3 expresses the comparison of the emulated values and simulated values with respect to the time ranged from 0.00 to 2.00 [s]. It confirms a tendency that the emulated values coincide with the simulated values more preferably as the values of the $R^2$ become greater. All the emulated values are outputted from Eq.(20) with the identical hyperparameters listed in Chapter 3. Although we fix these hyperparameters in every case to pursue the computational speed, the accuracy becomes better if we alter them for the respective cases. On the other hand, Fig.4 focuses on visualizing the coincidence of the emulated values with the simulated values. It demonstrates that the greater values of the $R^2$ do not always result in smaller values of RMSE/MAE. For instance, the $R^2$ is greater in Case B (0.788) than that of Case C (0.722), yet RMSE/MAE do not become smaller in Case B (0.020/0.019) relative to those of Case C (0.019/0.017). This indicates that the emulated values in Case B have larger errors than those of Case C at some time steps. Figure 5 illuminates the emulation and simulation consequences of the free-surface heights based on the mean $x$ values and the time range. It displays more details of Fig.3 by adding the mean values of $x$, which are one of the input variables in $X = \{x^{(i)}, \bar{x}^{(b)}, \bar{y}^{(b)}, \bar{z}^{(b)}\}$.

| Cases | $R^2$ | RMSE | MAE |
|-------|-------|------|-----|
| A | 0.754 | 0.027 | 0.025 |
| B | 0.788 | 0.020 | 0.019 |
| C | 0.722 | 0.019 | 0.017 |
| D | 0.721 | 0.033 | 0.030 |
| E | 0.782 | 0.020 | 0.019 |
| F | 0.988 | 0.005 | 0.005 |
| G | 0.292 | 0.045 | 0.042 |
| H | 0.682 | 0.029 | 0.027 |

Table 3 Results of emulation

| Cases | Total number of particles | Simulation time [s] | Emulation time [s] |
|-------|---------------------------|---------------------|--------------------|
| A | 18,096 | 511,484 | 0.002 |
| B | 19,916 | 707,341 | 0.003 |
| C | 18,616 | 549,843 | 0.003 |
| D | 20,046 | 738,882 | 0.002 |
| E | 18,516 | 567,072 | 0.003 |
| F | 19,006 | 596,963 | 0.002 |
| G | 20,716 | 794,664 | 0.002 |
| H | 18,186 | 541,161 | 0.003 |

Table 4 Results of computational time
Fig. 3  Comparison of emulated values and simulated values with respect to the time from 0.00 to 2.00 [s]. In Cases A-H, the red lines indicate the time transition of the "Emulated values," and the blue lines illustrate that of the "Simulated values."
Fig. 4 Coincidence of emulated values with simulated values. In Cases A-H, the red crosses show the “Emulated values” of the free-surface heights [m] at the selected time versus the “Simulated values” of the free-surface heights. On the dotted lines colored with blue, the emulated values coincide with the simulated values.
Fig. 5  Comparison of emulated values of free-surface heights in terms of time and $x$-position with those of simulated values. In Cases A-H, the red crosses visualize the “Emulated values” of free-surface heights [m] at the $x$-positions [m] and the selected time [s]. Similarly, the blue points show those of the “Simulated values.”
4.3. Computational time required by emulation

We measure the execution time necessary for the simulation and the emulation in Cases A-H on a PC with 1.6GHz dual-core Intel Core i5. Both the simulation and emulation are conducted once in each case. The former needs to repeat the whole calculation of EMPS described in Chapter 2 for the respective cases. By contrast, the latter requires the computation of Eqs.(18)-(21) to obtain the free-surface heights after the hyperparameters are selected based on the original dataset. Table 4 compares the results of the computational time required by the simulation with that of the emulation.

First, as Oochi et al. pointed out (2010), the table indicates that the simulation time tends to increase as the number of particles becomes greater. The total number of particles in Case G is greater than any other cases, and thus it requires the maximum simulation time. Conversely, the minimum time is required in Case A which has the smallest number of particles. The time difference between them is approximately 283 [s], and the average time is 626 [s] in Cases A-H.

Next, Table 4 illustrates that the emulation time is less than or equal to 0.003 [s] in all the cases once the emulator is constructed based on the dataset in the original case. The dataset is made by one simulation run of the original dam break problem illustrated in Fig.1. After selecting the appropriate hyperparameters by leave-one-out-cross validation (refer to Chapter 3), the developed emulator is able to predict the free-surface heights from the inputs of the other dataset in each case. Generally, the number of data points increases execution time required by emulators, however, the emulation time keeps mostly constant due to the exact same number of the data points in all the cases. Therefore, it can be stated in the present study that the emulator can estimate the objective values much faster than the simulator.

5. Conclusions

We designed a Gaussian process emulator to estimate free-surface heights in dam break problems faster than the EMPS simulator. The emulator required less than or equal to 0.003 [s] for computing them in a dam break problem, while it took 626 [s] on average by the simulator. We compared the emulation results with the simulated consequences by measuring $R^2$, RMSE, and MAE. We changed the initial shapes of water columns in dam break problems. In an original dam problem, which was utilized for the development of the emulator, the initial length in the $x$-axis and the initial height in the $z$-axis of the water column were set to 0.24 [m] and 0.50 [m], respectively. Since the initial shapes of the columns influenced the free-surface heights, we altered these scales and verified the accuracy of the emulation. As a result, for instance, $R^2$ was greater than 0.7 if the initial length was from 0.16 to 0.30 [m] and the initial height was 0.50 [m]. On the other hand, $R^2$ was greater than 0.7 if the initial height was between 0.42 to 0.64 [m] and the initial length was 0.24 [m]. Besides, it was clarified that the ratios of the lengths to the heights did not necessarily play an important role in determining the accuracy of the emulation results. The value of $R^2$ became greater and the values of RMSE/MAE were smaller if the difference was smaller between the original volume of the water column and the volumes in test cases.

Hence, there is a possibility that the Gaussian process emulator can be a replacement of the EMPS simulator for approximating free-surface heights based on a variety of initial shapes of water columns. To extend this study, we would like to (1) improve the designed emulator for more accurate time-series analysis and (2) apply it for the analysis of tsunami wave heights depending on initial wave forms that run up a slope in coastal regions.

References

Asakura, R., Iwase, K., Ikeya, T., Takao, M., Kaneto, T., Fujii, N. and Ohmori, M., The tsunami wave force acting on land structures, Proceedings of the 28th International Conference on Coastal Engineering (2003), pp.1191-1202.

Ferrari, A., Dumbser, M. and Toro, E.F., A new 3D parallel SPH scheme for free surface flows, Computers and Fluids, Vol.38 (2009), pp.1203-1217.

Gingold, R.A. and Monaghan J.J, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Monthly Notices of the Royal Astronomical Society, Vol.181, No.3 (1977), pp.375-389.

Igarashi, Y., Hori, T., Murata, S., Sato, K., Baba, T. and Okada, M., Maximum tsunami height prediction using pressure gauge data by a Gaussian process at Owase in the Kii Peninsula, Japan, Marine Geophysical Research, Vol.37, No.4 (2016), pp.361-370.

Kohavi, R., A study of cross-validation and bootstrap for accuracy estimation and model selection, Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol.2 (1995), pp.1137-1143.

Koshizuka, S. and Oka, Y., Moving-Particle Semi-implicit method for fragmentation of incompressible fluid, Nuclear Science and Engineering, Vol.123 (1996), pp.421-434.
Koshizuka, S., Nobe, A. and Y. Oka, Y., Numerical analysis of breaking waves using the Moving Particle Semi-implicit method, International Journal for Numerical Method in Fluids, Vol.26 (1998), pp.751-769.

Koshizuka, S., Shibata, K., Kondo, M. and Matsunaga, T., Moving Particle Semi-implicit method A meshfree particle method for fluid dynamics (2018), Academic Press.

Lucy, L.B., A numerical approach to the testing of the fission hypothesis, Astronomical Journal, vol.82 (1977), pp.1013-1024.

Mizuno, Y., Mitsume, N., Yamada, T. and Yoshimura, S., Time-based dynamic load balancing algorithm for domain decomposition with particle method adopting three-dimensional polygon-wall boundary model, Journal of Advanced Simulation in Science and Engineering, Vol.6, No.2 (2019), pp.282-297.

Monaghan, J.J., Smoothed Particle Hydrodynamics, Annual Review of Astronomy and Astrophysics (1992), pp.543-574.

Murotani, K., Oochi, M., Fujisawa, T., Koshizuka, S. and Yoshimura, S., Distributed memory parallel algorithm for Explicit MPS using ParMETIS, Transactions of the Japan Society for Computational Engineering and Science, Vol.2012 No.20120012 (2012). (in Japanese)

Murotani, K., Koshizuka, S., Tamai, T., Shibata, K., Mitsume, N., Yoshimura, S., Tanaka, S., Hasegawa, K., Nagai, N. and Fujisawa, T., Development of hierarchical domain decomposition Explicit MPS method and application to large-scale tsunami analysis with floating objects, Journal of Advanced Simulation in Science and Engineering, Vol.1, No.1 (2014), pp.16-35.

Oakley, J. and O’Hagan, A., Bayesian inference for the uncertainty distribution of computer model outputs, Biometrika, Vol.89, No.4 (2002), pp.769-784.

Oakley, J. and O’Hagan, A., Probabilistic sensitivity analysis of complex models: a Bayesian approach, Journal of the Royal Statistical Society Statistical Methodology Series B, Vol.66, No.3 (2004), pp.751-769.

O’Hagan, A., Bayesian analysis of computer code outputs: a tutorial, Reliability Engineering and System Safety, Vol.91, No.10-11 (2006), pp.1290-1300.

Oochi, M., Koshizuka, S. and Sakai, M., Explicit MPS algorithm for free surface flow analysis, Transactions of the Japan Society for Computational Engineering and Science, Vol.2010, No.20100013 (2010). (in Japanese)

Oochi, M., Yamada, Y., Koshizuka, S. and Sakai, M., Validation of pressure calculation in Explicit MPS method, Transactions of the Japan Society for Computational Engineering and Science, Vol.2011, No.20110002 (2011). (in Japanese)

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

Rougier, J., Efficient emulators for multivariate deterministic functions, Journal of Computational and Graphical Statistics, Vol.17, No.4 (2008), pp.827-843.

Rougier, J., Maute, A., Guillas, S. and Richmond, A. D., Expert knowledge and multivariate emulation: the thermosphere-ionosphere electrodynamics general circulation model (TIE-GCM), Special Issue on Computer Modeling, Vol.51, No.4 (2009), pp.414-424.

Sarri, A., Guillas, S. and Dias, F., Statistical emulation of a tsunami model for sensitivity analysis and uncertainty quantification, Natural Hazards and Earth System Sciences, Vol.12, No.6 (2012).

Shakibaeinia, A. and Jin, Y., A weakly compressible MPS method for modeling of open-boundary free-surface flow, International Journal for Numerical Methods in Fluids (2010), Vol.63, No.10, pp.1208-1232.

Yamada, Y., Sakai, M., Mizutani, S., Koshizuka, S., Oochi, M. and Murozono, K., Numerical simulation of three-dimensional free-surface flows with Explicit Moving Particle Simulation method, Transactions of the Atomic Energy Society of Japan, Vol.10, No.3 (2011), pp.185-193. (in Japanese)