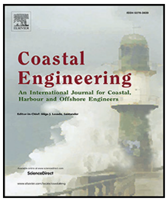
Coastal Engineering 174 (2022) 104118

Contents lists available at ScienceDirect

Coastal Engineering

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



Assessing erosion and flood risk in the coastal zone through the application of multilevel Monte Carlo methods

Mariana C.A. Clare [a,*], Matthew D. Piggott [a], Colin J. Cotter [b]

[a] Department of Earth Science and Engineering, Imperial College London, Exhibition Road, London SW7 2BU, UK
[b] Department of Mathematics, Imperial College London, Exhibition Road, London SW7 2BU, UK

ARTICLE INFO

Dataset link: https://github.com/mc4117/MLMC_XBeach

Keywords:
Multilevel Monte Carlo
Uncertainty analysis
Coastal flooding/erosion
Extreme events
Sediment transport

ABSTRACT

Coastal zones are vulnerable to both erosion and flood risk, which can be assessed using coupled hydro-morphodynamic models. However, the use of such models as decision support tools suffers from a high degree of uncertainty, due to both incomplete knowledge and natural variability in the system. In this work, we show for the first time how the multilevel Monte Carlo method (MLMC) can be applied in hydro-morphodynamic coastal ocean modelling, here using the popular model XBeach, to quantify uncertainty by computing statistics of key output variables given uncertain input parameters. MLMC accelerates the Monte Carlo approach through the use of a hierarchy of models with different levels of resolution. Several theoretical and real-world coastal zone case studies are considered here, for which output variables that are key to the assessment of flood and erosion risk, such as wave run-up height and total eroded volume, are estimated. We show that MLMC can significantly reduce computational cost, resulting in speed up factors of 40 or greater compared to a standard Monte Carlo approach, whilst keeping the same level of accuracy. Furthermore, a sophisticated ensemble generating technique is used to estimate the cumulative distribution of output variables from the MLMC output. This allows for the probability of a variable exceeding a certain value to be estimated, such as the probability of a wave run-up height exceeding the height of a seawall. This is a valuable capability that can be used to inform decision-making under uncertainty.

1. Introduction

The world's coastal zones have been at risk from erosion and flooding for millennia. There has been a growing awareness of these risks in recent decades with various international projects, such as FRMRC (Zurich Flood Resilience Alliance, 2019) and FLOODsite (FLOODsite, 2009), created in an attempt to investigate and mitigate the impacts. Nevertheless estimates suggest that the economic cost of damage due to coastal flooding in Europe will be two to three orders of magnitude greater than current levels by the end of the century (Vousdoukas et al., 2018). To assess the risk from these events, we must consider a combination of the probability of the events occurring and the consequences when they do (Garvey and Lansdowne, 1998). In this work, we focus on the former, but note that the modelling of coastal zones is a vital endeavour for the understanding of both of these factors.

For several decades, complex hydro-morphodynamic models have been used to simulate flooding and erosion in coastal zones. Whilst these models may produce accurate results for a given idealised scenario, there is a large degree of uncertainty associated with them when applied to the real world, originating from both the input data and the models themselves (see for example Unguendoli, 2018; Villaret et al., 2016). This uncertainty can be due to either incomplete knowledge or natural variability (Apel et al., 2004), and can be quantified by combining numerical hydro-morphodynamic models with statistical frameworks. A flexible but computationally intensive approach is to perform Monte Carlo simulations, translating the uncertainty in the inputs (say in the roughness parameterisations or random wave input) into uncertainty in the outputs. This can then be used to describe the uncertainty in the model output statistically. This method also has some limitations however, such as requiring a good estimate for the distribution of the uncertain input parameters (Reeve et al., 2014), but expert judgement can be used to place upper and lower bounds on this distribution. Moreover, this can be turned into an advantage, as the input distribution can be selected to focus on extreme values or another region of parameter space of particular interest. A greater limitation is that typically very large numbers of samples are needed for accurate Monte Carlo estimation, each sample requiring a separate model run. This is because of the $1/\sqrt{N}$ error estimate for Monte Carlo, where N is the number of samples. Although these samples can be performed in

[*] Corresponding author.
E-mail address: m.clare17@imperial.ac.uk (M.C.A. Clare).

https://doi.org/10.1016/j.coastaleng.2022.104118
Received 16 February 2021; Received in revised form 4 February 2022; Accepted 12 March 2022
Available online 22 March 2022

parallel, this can often represent a substantial computational resource commitment.

In this work, we explore uncertainty quantification techniques using the depth-averaged finite-volume based hydro-morphodynamic coastal ocean model XBeach (Roelvink et al., 2009) because it has been successfully applied and validated numerous times to simulate wave propagation, flow, sediment transport, and morphodynamic changes (see for example McCall et al., 2010; Riesenkamp, 2011; Callaghan et al., 2013; Roelvink et al., 2018; de Beer et al., 2020). XBeach has been used within a Monte Carlo framework before, for example in Riesenkamp (2011), Callaghan et al. (2013), Pender and Karunarathna (2013), Simmons et al. (2017) and Harris et al. (2018). However, like all complex hydro-morphodynamic models, it is relatively computationally expensive and, therefore, does not lend itself well to Monte Carlo simulations and hazard assessments, which necessitate a large number of individual model runs. For example, in Harris et al. (2018), a Monte Carlo simulation with 240,000 individual runs of the 1D version of XBeach was required to perform the desired study. This computational cost has limited the scenarios in which Monte Carlo based studies can be applied not just with XBeach but with other coastal models. Hence for more complex and long-term test cases, researchers have been forced to use simplified models, severely limiting the accuracy and scope of the results (Apel et al., 2004; Callaghan et al., 2013; Li et al., 2016; Toimil et al., 2017; Harris et al., 2018). A stark example of this is in Callaghan et al. (2013), where the much simpler semi-empirical model SBeach is used in the Monte Carlo simulation because even using the 1D version of XBeach would have taken an estimated four and a half millennia.

Rather than using simpler, less costly and less accurate models, we opt instead for using a more advanced statistical method with the original complex model. The multilevel Monte Carlo method (MLMC), first presented in Giles (2008), quantifies uncertainty by computing estimators for the expectation of discretised random variables for uncertain parameters. It seeks to reduce computational cost by accelerating the Monte Carlo approach through the use of a hierarchy of model configurations, each with a different level of resolution. It has been used in areas as diverse as mathematical finance (Giles, 2008), data assimilation (Gregory and Cotter, 2017a), tsunami generation (Sánchez-Linares et al., 2016) and seismic wave propagation (Ballesio et al., 2019). The aim of this work is to explore how MLMC can be applied to a complex hydro-morphodynamic coastal ocean model to investigate within a reasonable timeframe the impact of a variety of uncertain input parameters, such as wave height and bed slope angle, in both theoretical and real-world test cases. To the best of our knowledge this is the first application of MLMC with a hydro-morphodynamic model. A major advantage of Monte Carlo type methods over other more numerical methods such as adjoint methods or tangent linear models is that we can apply them as a wrapper around the model meaning that the underlying model does not have to be altered. This makes them easy to implement and means that the wrapper developed in our work could be applied to any other coastal ocean model. Whilst investigating the accuracy of the specific model we have used is beyond the scope of this work, this wrapper approach means that the numerous verification and validation studies conducted with XBeach still hold for our work (for example McCall et al., 2010; Riesenkamp, 2011; Roelvink et al., 2018).

A potential limitation when applying MLMC successfully to coastal problems is that decision makers are not only interested in the expected value of a variable, but also in the probability of a variable exceeding a certain value. This exceedance probability can then be used to not only determine which locations are at risk but also to determine the reliability of coastal structures such as sea walls, through structural reliability analysis (Melchers and Beck, 2018; Malliouri et al., 2021). Like any probability, the exceedance probability can be expressed as an expectation: here $\mathbb{E}[\mathbb{1}_{X\geq x}]$, where, for example, X is the maximum horizontal inundation distance, x the location of a physical structure, and $\mathbb{1}$ represents the indicator function. As discussed in Giles (2015), MLMC struggles with binary output variables, because a large number of samples are required to ensure accurate variance estimates. A number of different methods have been proposed to deal with this issue, including smoothing (Giles et al., 2015) and selective refinement (Elfverson et al., 2016). In this work, the ensemble generating method from Gregory and Cotter (2017b) is used (see Section 4), which is simple and computationally efficient. Unlike other approaches, this method generates the entire cumulative distribution function (CDF) for the output variable, which is of much greater value when assessing risk than a single statistic. Furthermore, generating the entire CDF using MLMC is a useful measure of the success of the method: if both the CDfs and expectations generated by the Monte Carlo method and MLMC are consistent, then this strongly suggests MLMC is able to recreate Monte Carlo results consistently over the whole parameter space of interest.

To summarise, the main novel contribution of this work is that it presents the first successful application of MLMC to a coupled hydro-morphodynamic model, demonstrating and quantifying the significant computational cost savings MLMC has over a standard Monte Carlo approach. This has been shown to be the case both for the calculation of the expected value of key variables of interest as well as their cumulative distribution function.

The remainder of this paper is structured as follows: in Section 2, we briefly outline the relevant MLMC theory; in Section 3, we integrate MLMC with XBeach and run a series of test cases; in Section 4, we estimate the cumulative distribution function of the output variable using MLMC and finally in Section 5, we present conclusions from this work.

2. Multilevel Monte Carlo method

The simplest method to estimate the expectation of a random variable $X(\omega)$ (where ω is a single sample from the sample space Ω) is to use a Monte Carlo method and take the average of $X(\omega)$ for N independent samples from Ω. However, the Monte Carlo method has an order of convergence of $N^{-1/2}$ (Caflisch, 1998) meaning to achieve an accuracy of ϵ requires $O(\epsilon^{-2})$ samples. This makes the simulation very computationally expensive. The multilevel Monte Carlo method (MLMC) was first introduced in Giles (2008). In this section, following Giles (2008, 2015), we discuss MLMC and its use in computing estimators for the expectation of discretised random variables given uncertain input parameters. We refer the reader to Giles (2008, 2015) for more details.

Whereas the Monte Carlo method considers the model at just one resolution, MLMC accelerates the Monte Carlo method by using a hierarchy of models at different levels of resolution. The fundamental idea underlying MLMC comes from the linearity of expectations: for two variables X_l and X_{l-1},

$$\mathbb{E}[X_l] = \mathbb{E}[X_{l-1}] + \mathbb{E}[X_l - X_{l-1}], \tag{1}$$

where $\mathbb{E}(\cdot)$ denotes the expectation. Extending this idea

$$\mathbb{E}[X_L] = \mathbb{E}[X_0] + \sum_{l=1}^{L} \mathbb{E}[X_l - X_{l-1}]. \tag{2}$$

In the MLMC framework, X_l denotes the numerical approximation to X on level l of the multilevel environment and thus X_0 and X_L denote the approximation on the coarsest and finest level respectively. Within the framework, each level l is defined by its numeric mesh element size

$$h_l = M^{-l}T, \tag{3}$$

where T is the total length of the domain and M the integer factor the mesh element size is refined by at each level (following standard practice, we use $M = 2$ throughout). Thus as l increases, the mesh becomes more refined. Note the domain can be multi-dimensional in which case there will be an h_l for every dimension of T.

In MLMC, the expectation estimator is defined as

$$\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l, \tag{4}$$

where $\hat{Y}_l$ is the difference estimator for $\mathbb{E}[X_l - X_{l-1}]$ defined as

$$\hat{Y}_l = \begin{cases} N_0^{-1} \sum_{i=1}^{N_0} X_0^{(i)} & l = 0, \\ N_l^{-1} \sum_{i=1}^{N_l} \left(X_l^{(i)} - X_{l-1}^{(i)} \right) & l > 0, \end{cases} \tag{5}$$

where N_l is the number of samples at each level (l, $l-1$) pair. Here the same random numbers are used to construct the variables X_l and X_{l-1} in order to minimise the variance of the difference estimator and hence the overall error. To ensure independence, different independent samples are used at each level, meaning $\mathrm{Cov}(\hat{Y}_i, \hat{Y}_j) = 0$ if $i \neq j$ and thus

$$\mathbb{V}(\hat{Y}) = \mathbb{V}\left(\sum_{l=0}^{L} \hat{Y}_l \right) = \sum_{l=0}^{L} N_l^{-1} V_l, \tag{6}$$

where $\mathbb{V}(\cdot)$ denotes the variance and V_l is the variance of the sample $X_l^{(i)} - X_{l-1}^{(i)}$. The overall cost of calculating the expectation estimator (4) is given by $\sum_{l=0}^{L} N_l C_l$ where C_l is the cost per level l.

To calculate the error of the estimator (4), we note that root mean square error (RMSE) is defined as

$$\mathrm{RMSE} = \sqrt{\mathbb{E}[(\hat{Y} - \mathbb{E}[X])^2]}. \tag{7}$$

This is simplified by noting $\hat{Y}$ is an unbiased estimator of $\mathbb{E}[X_L]$ and so

$$\mathrm{RMSE} = \sqrt{\mathbb{E}[(\hat{Y} - \mathbb{E}[X_L])^2] + (\mathbb{E}[X_L] - \mathbb{E}[X])^2}. \tag{8}$$

Here $\mathbb{E}[(\hat{Y} - \mathbb{E}[X_L])^2]$ is equivalent to (6) and is the Monte Carlo error; and $(\mathbb{E}[X_L] - \mathbb{E}[X])^2$ is the square of the bias and is the numerical discretisation error.

A bound on the RMSE (8) is provided by the key complexity theorem for MLMC. We outline it briefly here — the full details can be found in Giles (2015) with a proof given in detail in Cliffe et al. (2011). Briefly the theorem states that if the following conditions are met

$$\mathbb{E}[X_l - X] \leq c_1 2^{-\alpha l}, \tag{9a}$$

$$\mathbb{E}[\hat{Y}_l] = \begin{cases} \mathbb{E}[X_0], & l = 0, \\ \mathbb{E}[X_l - X_{l-1}], & l > 0, \end{cases} \tag{9b}$$

$$V_l \leq c_2 2^{-\beta l}, \tag{9c}$$

$$C_l \leq c_3 2^{\gamma l}, \tag{9d}$$

where $\alpha \geq 1/2 \min(\beta, \gamma)$, and α, β, γ, c_1, c_2, c_3 are positive constants, then there exists a finest level resolution L and a finite number of samples N_l such that for any $\epsilon < e^{-1}$, the estimator $\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l$ has

$$\mathrm{RMSE} \leq \epsilon. \tag{10}$$

Furthermore, the total computational cost of calculating this estimator with this RMSE is

$$C \leq \begin{cases} c_4 \epsilon^{-2}, & \beta > \gamma, \\ c_4 \epsilon^{-2} (\log \epsilon)^2, & \beta = \gamma, \\ c_4 \epsilon^{-2-(\gamma-\beta)/\alpha}, & 0 < \beta < \gamma. \end{cases} \tag{11}$$

where c_4 is a positive constant.

Note (9a), (9c) and (9d) represent a bound on the bias, variance and cost respectively at each level and (9b) follows from the definition in (5). As the mesh is refined, β controls the decay in the variance and γ the growth of the cost. Thus if $\beta > \gamma$, most of the computational cost is at the coarser levels and if $\beta < \gamma$ it is mostly at the finest levels. This means if $\beta < \gamma$, the order of convergence is better than the standard Monte Carlo method and if $\beta > \gamma$, it is equivalent to the Monte Carlo method. The values of α, β and γ are found using convergence tests: the gradient of the line of best fit of $\mathbb{E}(X_l - X_{l-1})$ against l is α, $\mathbb{V}(X_l - X_{l-1})$ against l is β and timed cost against l is γ.

A key part of MLMC is determining the optimum N_l to achieve the convergence stated in (10). Following standard practice (for example Giles, 2008), the optimum value of N_l is determined by using the Euler–Lagrange method to minimise the overall cost with respect to the fixed overall variance $\epsilon^2/2$. It can be shown that the optimum number of samples at each level is

$$N_l = \left\lceil 2\epsilon^{-2} \sqrt{\frac{V_l}{C_l}} \left(\sum_{l=0}^{L} \sqrt{V_l C_l} \right) \right\rceil. \tag{12}$$

This formula requires a knowledge of the variance and cost at level l. Following the proof in Cliffe et al. (2011), we instead use the conditions in (9) to create an upper bound on N_l (for example (9c) and (9d) mean that $V_l C_l < c_2 c_3 2^{(\gamma-\beta)l}$) and set this to be the optimum number of samples. Thus

- if $\beta > \gamma$:

$$N_l = \lceil 2\epsilon^{-2} c_2 \left(1 - 2^{-(\beta-\gamma)/2}\right)^{-1} 2^{-(\beta+\gamma)l/2} \rceil, \tag{13}$$

- if $\beta < \gamma$:

$$N_l = \lceil 2\epsilon^{-2} c_2 2^{(\gamma-\beta)L/2} \left(1 - 2^{-(\gamma-\beta)/2}\right)^{-1} 2^{-(\beta+\gamma)l/2} \rceil, \tag{14}$$

- if $\beta = \gamma$:

$$N_l = \lceil 2\epsilon^{-2} (L+1) c_2 2^{-\beta L} \rceil. \tag{15}$$

2.1. MLMC verification checks

Accompanying MLMC are the following checks, detailed in Giles (2015), to ensure the method implementation is both correct and appropriate.

- **Convergence check:** The convergence condition (9a) states that $\mathbb{E}[X_l - X_{l-1}] \propto 2^{-\alpha l}$ and thus it follows that the bias $\mathbb{E}[X - X_L] = \frac{\mathbb{E}[X_L - X_{L-1}]}{2^{\alpha}-1}$. Therefore we use the following as a convergence test

$$\frac{|\mathbb{E}[X_L - X_{L-1}]|}{2^{\alpha} - 1} < \frac{\epsilon}{\sqrt{2}}, \tag{16}$$

  where α is determined by calculating the gradient of the line of best fit of $\mathbb{E}(X_l - X_{l-1})$ against l. If this check fails, this means MLMC has not converged to within the specified tolerance — this may mean there is a mathematical or programming error or simply that the value of ϵ chosen is too small for the number of levels in the MLMC algorithm.
- **Consistency check:** Suppose a, b and c are estimates for $\mathbb{E}[X_{l-1}]$, $\mathbb{E}[X_l]$ and $\mathbb{E}[Y_l]$ respectively. Then $a - b + c \approx 0$ up to the expected error from Monte Carlo sampling. Furthermore, $\sqrt{\mathbb{V}[a - b + c]} \leq \sqrt{\mathbb{V}(a)} + \sqrt{\mathbb{V}(b)} + \sqrt{\mathbb{V}(c)}$. Therefore if calculated correctly the following ratio

$$\frac{|\mathbb{E}[X_l - X_{l-1}] - \mathbb{E}[X_l] + \mathbb{E}[X_{l-1}]|}{3\left(\sqrt{\mathbb{V}(X_l - X_{l-1})} + \sqrt{\mathbb{V}(X_l)} + \sqrt{\mathbb{V}(X_{l-1})}\right)}, \tag{17}$$

  has a probability of 0.3% of being greater than 1 (see Giles, 2015). If this ratio is greater than 1, this indicates that there is a difference between how X_l and X_{l-1} are computed and not just that they are on a different mesh.
- **Kurtosis check:** The kurtosis, κ, of $X_l - X_{l-1}$ determines the number of samples required for a good variance estimate. If κ is very large this indicates that the variance estimate is poor and that either the number of samples used is insufficient or the method chosen for sampling is incorrect. Thus following standard practice, if the kurtosis is greater than 100, an error is raised.

We use these checks to evaluate the coding and mathematical implementation of the test cases in this work, and to identify any inherent problems with applying MLMC to these test cases.

2.2. MLMC algorithm

We have now outlined the basic MLMC method and conclude this section with a statement of the MLMC algorithm used in this work, modified from Giles (2008):

Algorithm 1 Multilevel Monte Carlo Method

```
1: Start with $L = 0$
2: Estimate the variance $V_L$ using an initial estimate for the number
   of samples $N_L$
3: Define optimal $N_l$ for $l = 0, ..., L$ using (12)
4: If the optimal $N_l$ is greater than the number of samples you already
   have, evaluate the extra samples needed
5: If $L \geq 2$ test for convergence i.e. check if (16) is satisfied
6: If $L < 2$ or the algorithm has not converged set $L := L + 1$ and
   return to Step 2
```

3. Applying MLMC to XBeach

In this section, we apply MLMC to the XBeach model in order to estimate the expectation of random variables in test cases given uncertain input data. We apply MLMC through a Python wrapper around the XBeach model. We have also parallelised the wrapper meaning that, within the same level, multiple XBeach runs can be performed at the same time, thus increasing the efficiency of the application of the algorithm. Thus throughout, when time is referred to, this is the total time the simulation took to run multiplied by the number of cores used (40). Note our MLMC wrapper is written so that it can easily be applied to other models in further work. We begin this section by giving a brief overview of the XBeach model.

3.1. XBeach model

XBeach is a depth-averaged finite-volume coastal ocean model which simulates near-shore hydrodynamics and morphodynamics (Roelvink et al., 2015). It has been used successfully numerous times to simulate wave propagation, flow, sediment transport and morphodynamic changes in the coastal zone including for example in Roelvink et al. (2018) and de Beer et al. (2020). Within XBeach, surface elevation and flow are modelled using the Generalised Lagrangian Mean formulation of the depth-averaged shallow water equations (Andrews and McIntyre, 1978), enabling the modelling of short wave-induced mass fluxes and return flows. These equations are:

$$\frac{\partial \mathbf{u}^L}{\partial t} + \mathbf{u}^L \cdot \nabla \mathbf{u}^L + \mathbf{f} \wedge \mathbf{u}^L - \nu \nabla^2 \mathbf{u}^L = \frac{\boldsymbol{\tau}_s}{\rho h} - \frac{\boldsymbol{\tau}_b^E}{\rho h} - g\nabla\eta + \frac{\mathbf{F}}{\rho h}, \tag{18}$$

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (h\mathbf{u}^L) = 0, \tag{19}$$

where $\mathbf{u}^L$ is the Lagrangian velocity defined in Roelvink et al. (2015) as the distance travelled by an individual water particle during one wave period divided by the wave period, f the Coriolis vector, ν the viscosity, τ_s and τ_b^E the wind and bed shear stresses respectively where E denotes Eulerian velocity defined in Roelvink et al. (2015) as the fixed point short-wave-averaged velocity, η the elevation, $\mathbf{F}$ the wave-induced stresses, ρ the density and h the water depth.

Waves can be modelled using several different options in XBeach. In this work, the stationary wave model is used in Section 3.2 and the surfbeat model in Section 3.3. Both methods solve the following wave action equation for short wave energy

$$\frac{\partial A}{\partial t} + \frac{\partial c_x A}{\partial x} + \frac{\partial c_y A}{\partial y} + \frac{\partial c_\theta A}{\partial \theta} = -\frac{D_w + D_f + D_v}{\sigma}, \tag{20}$$

but in the stationary mode, $\frac{\partial A}{\partial t} = 0$. Here c_x is the group velocity in the x-direction, c_y the group velocity in the y-direction, c_θ the refraction speed, D_w the dissipation by wave breaking, D_f, the dissipation by bottom friction, D_v the dissipation by vegetation and A is the wave energy. Note the formulations of D_w and D_f are different in the stationary and surfbeat mode (see Bart, 2017 for further details).

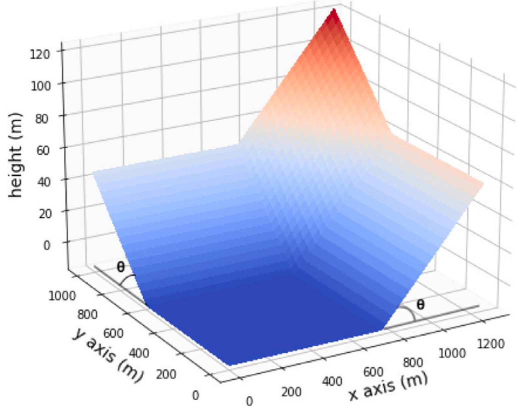


Fig. 1. Schematic of 2D bed with uncertain bed slope angles θ for 2D bed slope test case.

Finally, to model morphological changes in Section 3.3.1, XBeach uses the following standard depth-averaged advection–diffusion equation to model sediment transport

$$\frac{\partial hC}{\partial t} + \nabla \cdot (hC\mathbf{u}^{\mathbf{E}}) + \nabla \cdot (D_h h \nabla C) = \frac{hC_{\text{eq}} - hC}{T_s}, \tag{21}$$

where C is the depth-averaged sediment concentration, $\mathbf{u}^{\mathbf{E}}$ the Eulerian velocity, D_h the diffusivity coefficient, C_{eq} the equilibrium sediment concentration and T_s the adaptation time; and the Exner equation to model bed changes

$$(1-p)\frac{\partial z_b}{\partial t} + f_{\text{mor}} \nabla \cdot \mathbf{q} = 0, \tag{22}$$

where z_b is the bed, p the bed porosity, f_{mor} the morphological acceleration factor used to artificially increase the rate at which the bed changes compared with the underlying hydrodynamics and $\mathbf{q}$ the sediment transport rates.

In this work, we use version 1.23.5526 of the XBeachX release and unless explicitly stated, parameter values are left at the default for this version.

3.2. Uncertainty in bed slope angle

We now apply the MLMC framework from Section 2 to the XBeach model described above. In the following test case, we evaluate the uncertainty associated with the beach bed profile by considering the bed slope angle θ to be uncertain. This represents a significant source of uncertainty, as discussed in Unguendoli (2018).

3.2.1. Two-dimensional (2D) bed slope test case

For the first test case in this work, we consider a simple 2D hydrodynamics-only problem of a stationary wave approaching the beach with a wave period of 10 s and a maximum wave amplitude of 2.5 m and each simulation is run for 200 s. The beach slope is modified in both the x and y-directions by the same angle θ as depicted in Fig. 1.

The distribution of θ is set to be $\theta \sim \mathcal{U}(\arctan(0.035), \arctan(0.5))$. The lower bound is chosen to ensure that there is always an area of the bed which is above the initial water level. We seek the expected value of the maximum horizontal inundation distance during the simulation. In order to satisfy the constraints on ϵ required to achieve the convergence stated in (10), the inundation distance is normalised by dividing it by the total x-length of the domain (1250 m).

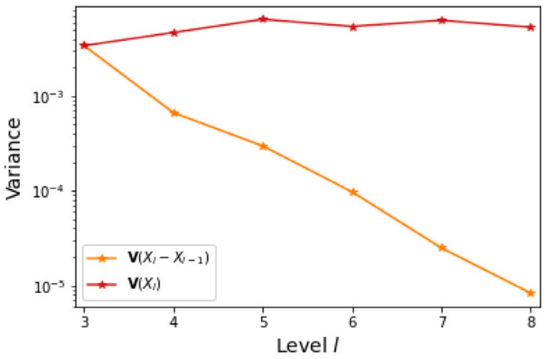


Fig. 2. Variance of difference, $X_l - X_{l-1}$, compared to variance of single variable, X_l, at each level for 2D bed slope test case showing the use of MLMC is justified.

Before running the full MLMC algorithm (Algorithm 1), we first run Steps 1 and 2, in order to check that the MLMC approach is appropriate for this test case. Recalling (3), the grid size is $\Delta x = 1250/2^l$ and $\Delta y = 1000/2^l$ and we use 7 levels ($l = 2$ to 8) and 500 samples at each level. Fig. 2, a log plot of $\mathbb{V}(X_l - X_{l-1})$ against level, shows that the variance decreases as the level number l increases, *i.e.* as the mesh gets finer. This observation is important because, recalling (6), this means fewer samples are needed on the finer meshes. The figure also shows that $\mathbb{V}(X_l - X_{l-1})$ is lower than $\mathbb{V}(X_l)$ for all l. This is also important because otherwise a lower variance could be achieved by using the same number of samples with a simple Monte Carlo simulation. Thus, we expect the MLMC simulation to be less computationally expensive than a Monte Carlo simulation meaning the choice of the MLMC approach for this test case is justified.

Using this figure, as well as the log plots of cost against level l and $\mathbb{E}(X_l - X_{l-1})$ against level l, convergence tests estimate the convergence values defined in (9) as $\alpha = 1.18; \beta = 1.69; \gamma = 2.93$ for this test case. Thus the numerical discretisation is approximately first order accurate (based upon the value of α). As $\gamma > \beta$ most of the computational cost in the MLMC algorithm will come from the finer levels and we expect the order of convergence to be better than the standard Monte Carlo method.

These observations from the preliminary study, as well as the fact that the consistency and kurtosis checks from Section 2.1 passed, means we can conclude that MLMC can be used successfully with XBeach and we can run the full algorithm. For this test case, the chosen range of tolerance values are $\epsilon = [0.001, 0.0017, 0.0025, 0.0035, 0.005, 0.006]$ recalling that ϵ controls the bound on the convergence test (16). As $\beta > \gamma$, we use these ϵ values in the formula (13) to derive the optimum number of samples required at each level, N_l. We also set a maximum limit on the finest level that l can reach of $L_{\max} = 8$ (see Step 6 of Algorithm 1). As $\gamma > \beta$, the optimum N_l are calculated using (14) and shown in Fig. 3. Most importantly as the level number increases (*i.e.* as the grid becomes finer), fewer samples are needed, which makes MLMC more efficient than a simple Monte Carlo method.

To test the accuracy of the MLMC estimator, we also estimate the real solution $\mathbb{E}[X]$ using a Monte Carlo simulation with $\Delta x = 1250/2^9 = 2.44\,\text{m}$ and $\Delta y = 1000/2^9 = 1.95\,\text{m}$, which is the same as the finest mesh in the MLMC simulation. According to Monte Carlo the maximum expected horizontal inundation distance along the x-axis over all time and all y divided by the total x-length of the domain is 0.6934 (to four significant figures). As the total domain length is 1250 m, this in fact means the water inundates a horizontal distance of 866.8 m. This agrees up to four significant figures with the value found using the MLMC simulation using $\epsilon = 0.001$. We can calculate the root mean square error of the MLMC simulation compared to the Monte Carlo simulation using

$$\text{RMSE of MLMC} = \sqrt{\left(\mathbb{E}[X_L] - \mathbb{E}[X_{MC}]\right)^2}. \tag{23}$$

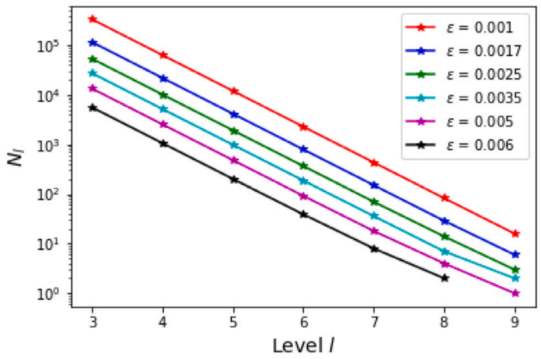


Fig. 3. Optimum number of samples required at each l for given tolerance value, ϵ for 2D bed slope test case.

Table 1
Computational cost improvement from using MLMC instead of Monte Carlo to achieve the same RMSE for the 2D slope test case.

	MLMC	MC
RMSE (3sf)	0.000175	0.000175
Time to achieve RMSE (h)	160	250,973
MLMC speed-up factor	1569	

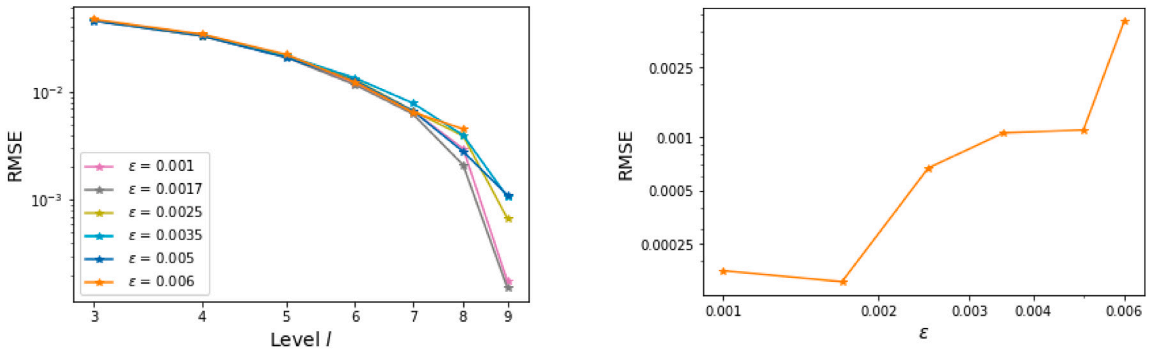
Fig. 4(a) shows that as the mesh becomes finer, the RMSE decreases for all ϵ values considered. In addition, Fig. 4(b) shows that as ϵ decreases, in general the final expected value becomes closer to the real solution showing that the convergence condition (16) has worked as expected.

Finally Fig. 5 compares the total cost and accuracy of the MLMC and Monte Carlo methods. Most significantly, the figure shows that MLMC achieves the same accuracy as Monte Carlo for 0.06% of the computational cost, as summarised in Table 1. We have thus significantly reduced the computational time by using MLMC rather than Monte Carlo.

Additionally, Fig. 5 shows that MLMC converges uniformly to the final value due to the use of the convergence test in Section 2.1 and optimum number of samples calculation. This is not true for the Monte Carlo simulation where further samples are required to check whether the algorithm has converged, thus increasing the overall computational cost of the Monte Carlo simulation.

As this is our first test case using MLMC with XBeach, we also conduct some further analysis to ensure that MLMC and the Monte Carlo method are consistent. So far in this section we have confirmed that the MLMC estimator for $\mathbb{E}[X]$ is consistent with that from the Monte Carlo method. We repeat the MLMC methodology in this section, this time estimating $\mathbb{E}[X^2]$, and only using the lowest tolerance considered $\epsilon = 0.001$. Note that in future MLMC can be used to calculate $\mathbb{E}[X]$ and $\mathbb{E}[X^2]$ as it can easily be extended to computing multiple outputs (Clare et al., 2022). Using $\mathbb{V}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$, we obtain an MLMC estimate for the variance of 0.00617 and a Monte Carlo estimate for the variance of 0.00618, where the latter is estimated using the same number of samples as for the Monte Carlo expectation. Therefore, we conclude that MLMC is able to accurately estimate both the expected value and the variance of the output variable in a computationally efficient manner.

Although this test case is somewhat abstract, the consistent estimates for both expectation and variance demonstrate that MLMC can be used with XBeach for a wave with known properties approaching a 2D beach with unknown slope. We have shown that the key advantage of MLMC is that it can achieve results considerably faster when compared to using a simple Monte Carlo approach. This means we can confidently apply our method to more realistic problems.



(a) RMSE of MLMC at each level l for given tolerance value ϵ.

(b) RMSE of MLMC once it has converged for each tolerance value ϵ.

Fig. 4. RMSE of MLMC (23) for 2D bed slope test case for varying tolerance value ϵ.

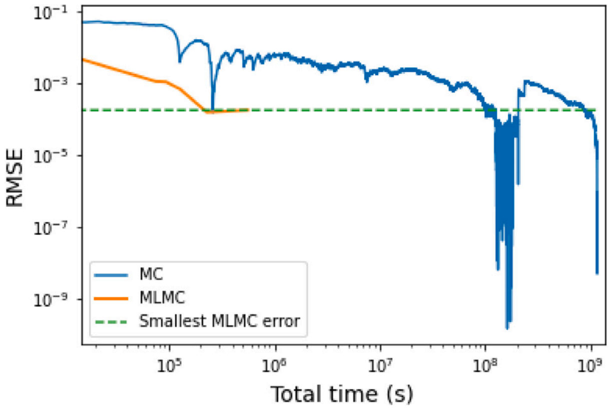


Fig. 5. Comparison of total cost and accuracy when using MLMC and Monte Carlo (MC) for 2D bed slope test case.

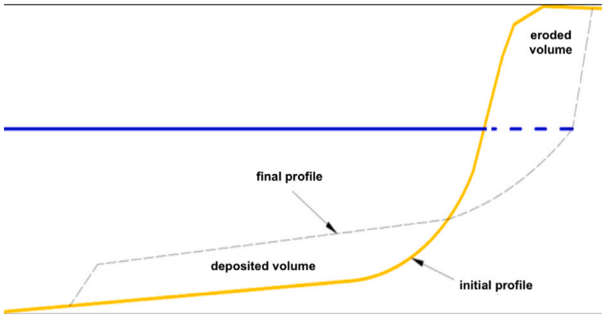


Fig. 6. Schematic showing the initial and final bed profiles after a simulation. If the volume change in the morphology test case is positive, this means that the erosion volume is greater than the accretion volume and vice versa.
Source: Adapted from Van Gent et al. (2007).

3.3. Uncertainty in wave height

In the previous section, we considered the uncertainty associated with a simple beach bed profile. Another significant source of uncertainty is the structure of the waves approaching the beach. Whilst the previous test case is purely theoretical, in this section we consider a lab test case and a test case using real world data. In real-world ocean environments, no two waves are alike, and they have varying heights, frequencies and directions. A common technique to deal with this is to use wave spectra (see Scheffner and Borgman, 1992; McCabe, 2011). These can be for specific time-periods and/or locations, such as in Cai et al. (2008), or generic. For the purposes of generality, here we use the generic JONSWAP wave spectrum, which has the advantage it is inbuilt in XBeach and suitable for oceans in which the unobstructed distance over which the wind blows is limited, *i.e.* fetch-limited oceans like the ones in our test cases (Hasselmann et al., 1973). The uncertain parameter is then the wave height in the JONSWAP wave spectrum, h (see Hasselmann et al., 1973 for more details on this spectrum).

A vital condition of MLMC theory is that for each X_l^i, X_{l-1}^i pair calculation, the only difference in the model must be the mesh element size. Thus, when calculating X_l^i, we use the XBeach option `random = 0` which means that the same random seed is used to initialise the wave boundary conditions for both X_{l-1}^i and X_l^i. Both test cases below pass the consistency check in Section 2.1 indicating that the formulation is both mathematically correct and implemented correctly.

3.3.1. Morphology test case

An important variable to consider when assessing the risk to coastal areas is the volume of material eroded. To do this we consider the standard DelflandStorm test case, available as part of the XBeach documentation (Roelvink et al., 2015). Note this is a 1D test case so we only alter the mesh element size in the x-direction.

In this test case, we seek the expected total volume change over the simulation, found by using the trapezium rule to approximate the integral of the difference between the initial bed profile and the final bed profile, as shown in Fig. 6. Thus if the number is positive, the erosion volume is greater than the accretion volume and the beach has lost material and been net eroded, and if the inverse is true the beach has accumulated material and grown. Note that as this is a test case in one horizontal dimension this volume is in fact an area but to avoid confusion with surface area we shall continue referring to it as a volume. We normalise the volume change using the total initial volume of the beach, *i.e.* the volume underneath the slope with the minimum value of z being the lower vertical limit. The distribution of the wave height is set to be $h \sim \mathcal{U}(0\,\text{m}, 5\,\text{m})$.

To check whether MLMC can be used for this test case, we run steps 1 and 2 of Algorithm 1. Fig. 7 shows that $\mathbb{V}\left(X_l - X_{l-1}\right)$ decreases as the mesh becomes finer and that for all l it is lower than $\mathbb{V}\left(X_l\right)$. Using convergence tests, the convergence values in (9) are $\alpha = 0.921, \beta = 1.77, \gamma = 2.40$. Thus the numerical discretisation is approximately first order accurate and as $\gamma > \beta$, we expect the order of convergence to be better than the standard Monte Carlo method. In addition, the test case also passes the standard checks in Section 2.1.

Given the success of the preliminary study, we run the full algorithm and consider tolerance values of $\epsilon = [1.5 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 10^{-4}]$ with a maximum limit for the finest level of $L_{\max} = 12$. As $\gamma < \beta$, the optimum N_l are calculated using (14) and are shown in Fig. 8. Importantly the number of samples required decreases as the mesh becomes finer.

In order to calculate the RMSE of MLMC for this test case, we estimate the real solution $\mathbb{E}[X]$ through a Monte Carlo simulation using

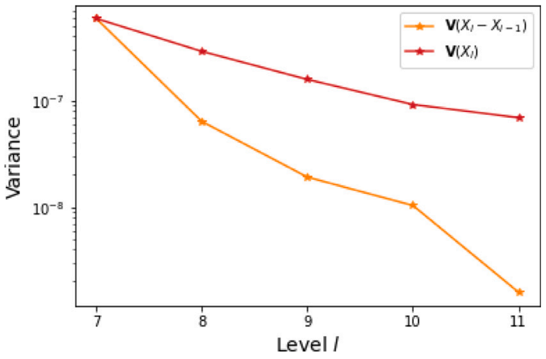


Fig. 7. Variance of difference, $X_l - X_{l-1}$, compared to variance of single variable, X_l, at each level for morphology test case showing the use of MLMC is justified.

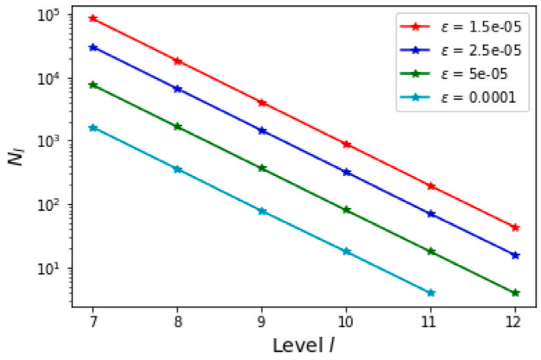


Fig. 8. Optimum number of samples required at each l for given tolerance value ϵ for the morphology test case.

$\Delta x = 1421.562/2^{12} = 0.347$ m, which is the same as the finest mesh utilised in the MLMC simulation. The expected volume change as a proportion of the total initial volume is 0.0006150 (to four significant figures). This is positive and thus means that the beach has been net eroded during the simulation. As the total volume is 16 333 m^2, this in fact means that 10 m^2 of beach material is eroded during the simulation (recall here that the test case is in one horizontal dimension so the volume is in fact an area). Figs. 9(a) and 9(b) confirm that the RMSE decreases as the level number increases and as ϵ decreases respectively, thus confirming MLMC is working as expected. Unlike with the other test cases considered, Fig. 9(b) shows that the relationship between the RMSE and ϵ is not uniform and there is a clear elbow in the plot at $\epsilon = 5\times10^{-5}$. This is a discrete effect caused by the fact that for $\epsilon = 10^{-4}$, the convergence test (16) passes without requiring any samples on the finest level ($l = 12$) (see Fig. 8).

Finally Fig. 10 shows that MLMC with $\epsilon = 1.5\times10^{-5}$ achieves the same accuracy as Monte Carlo for 2% of the computational cost, as summarised in Table 2. In addition, the figure shows that in this test case the convergence of the Monte Carlo simulation is notably less uniform than that from MLMC, meaning the Monte Carlo simulation must be run for a long time to ensure convergence. (Note that the elbow present in Fig. 9(b) is again present in the MLMC convergence for the same reason). Thus we have shown that MLMC can significantly decrease computational costs without compromising on accuracy and improve result stability for a full hydro-morphodynamic model with uncertainties in the wave height.

3.3.2. Boscombe beach test case

For the final test case, we consider the real world case of Boscombe Beach in Dorset, UK, which is a standard 2D test case provided in the XBeach documentation (Roelvink et al., 2015). The distribution of the wave height is set to be $h \sim \mathcal{U}(0\text{ m}, 3\text{ m})$. To make the problem simpler, the morphological and tidal component present in the standard test case are switched off, but all remaining values are kept the same as those in the standard test case.

Table 2
Computational cost improvement from using MLMC instead of Monte Carlo to achieve the same RMSE for the morphology test case.

	MLMC	MC
RMSE (3sf)	3.51×10^{-6}	3.51×10^{-6}
Time to achieve RMSE (h)	10,911	465,310
MLMC speed-up factor	43	

When assessing flood risk, an important quantity not yet considered in this work is the wave run-up height. Hunt (1959) conclude that the wave run-up height is dependent on the structure of the wave and the angle of the beach bedslope. Traditionally, the value calculated is the wave run-up height exceeded by 2% of incoming waves, $R_{u2\%}$ (shown in Fig. 11), also known as the extreme run-up. This value was first mentioned in Asbeck (1953), but has since been calculated in many works (for example Stockdon et al., 2006; Suanez et al., 2015; Cohn and Ruggiero, 2016). According to Van der Meer et al. (2018), it is not entirely clear why the choice of 2% was made, although one possible explanation may be that soft engineering measures such as clay and grass are considered sufficient to withstand the 2% of waves that exceed this height.

Although $R_{u2\%}$ is by definition a probabilistic measure, there exist a variety of standard empirical formulae that can be used to estimate this value deterministically for an individual simulation (see Melby et al., 2012). One such formula is derived in Holman (1986) using field data from natural beaches with slopes, $\tan\alpha$, between 0.07 and 0.2:

$$R_{u2\%} = 0.83\xi_0 H_{m0} + 0.2H_{m0}, \tag{24}$$

where H_{m0} is the significant wave height at the beach-toe (*i.e.* where the water level (z_s) and the bed (z_b) are equal to each other) and ξ_0 the Iribarren parameter defined by

$$\xi_0 = \frac{\tan\alpha}{\sqrt{H_{m0}/L_0}}, \tag{25}$$

where α is the beach slope angle at the beach-toe, L_0 the wavelength calculated by

$$L_0 = \frac{gT_p^2}{2\pi}, \tag{26}$$

and T_p the peak wave period.

Although relatively simple, this formula is a standard one for calculating extreme wave run-up on natural beaches (see for example Ruggiero et al., 2001; Stockdon et al., 2006; Melby, 2012; Díaz-Sánchez et al., 2014; Suanez et al., 2015; Park and Cox, 2016). Furthermore, the shallow slope of Boscombe Beach makes (24) an appropriate formula for our test case and thus we use it in our test case to find the expected value of the maximum run-up height $R_{u2\%}$ attained during the simulation over all values of y.

Convolution filter

To define the Boscombe Beach profile, we use the bed data provided in the XBeach documentation (Roelvink et al., 2015). Clearly in real world environments, the beach approach is not smooth as has been the case in the test cases considered so far in this work. Instead there may be sandbars or other similar physical features present, either natural or anthropogenic. This could potentially cause issues especially if a feature is not represented in coarser meshes and suddenly appears in finer meshes. However, with MLMC it is not necessary that the coarsest mesh is a good approximation to the finest mesh, merely that each mesh l is a good enough approximation to the next finest mesh $l + 1$.

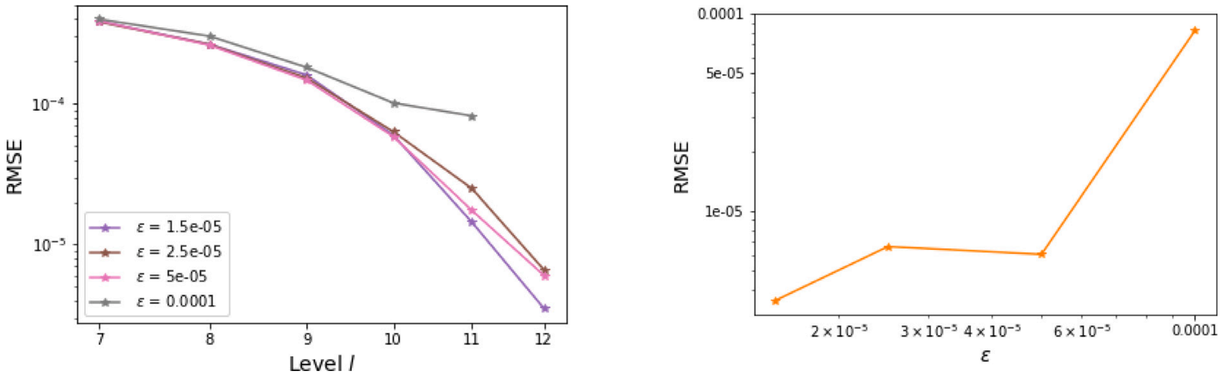


Fig. 9. RMSE of MLMC (23) for morphology test case for varying tolerance value ϵ.

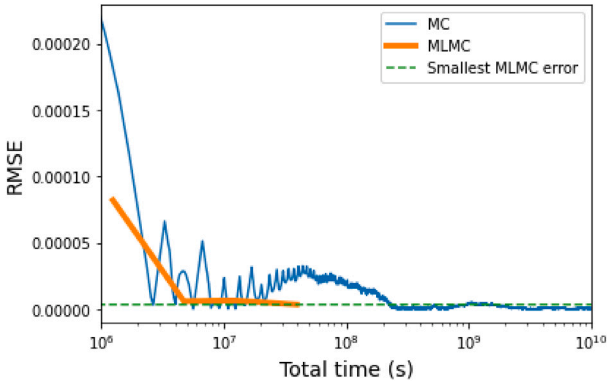


Fig. 10. Comparison of total cost and accuracy when using MLMC and Monte Carlo (MC) for morphology test case.

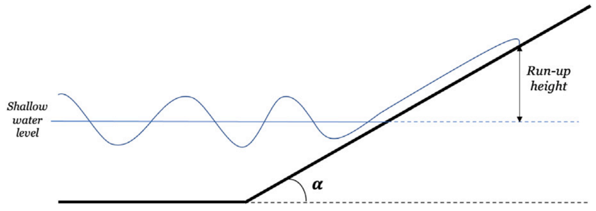


Fig. 11. Schematic showing wave run-up height at a beach (the quantity of interest in the Boscombe Beach test case) and key quantities needed to calculate it.

In order to mitigate the issue of features suddenly appearing, we use a convolution matrix filter which has the general expression

$$(f * g)(i) = \sum_{k=-\lambda}^{\lambda} g(k) f(i-k), \tag{27}$$

where $\lambda = \lfloor \frac{m}{2} \rfloor$ and m is the length of g. An appropriate filter kernel to use in this case is the normalised arithmetic mean

$$g(j) = \frac{1}{\sum_{i=1}^{m} g(i)}. \tag{28}$$

At the edges, we use the 'nearest' method meaning that the edge value is repeated outwards as many times as necessary for the multiplication of the weights.

For this test case, f is the underlying bed and the convolution filter is applied in the direction perpendicular to the coastline. The convolution filter length, m_l at each level l follows $m_l = \max((L_{\max} - l), 1)$ where $L_{\max}$ is the finest mesh level considered. At level $L_{\max}$, $m = 1$ which is the identity filter and thus the bed is unchanged. Hence, as the mesh becomes finer, the convolved bed slowly converges to the original bed and thus there are no extra errors associated with using a convolved bed.

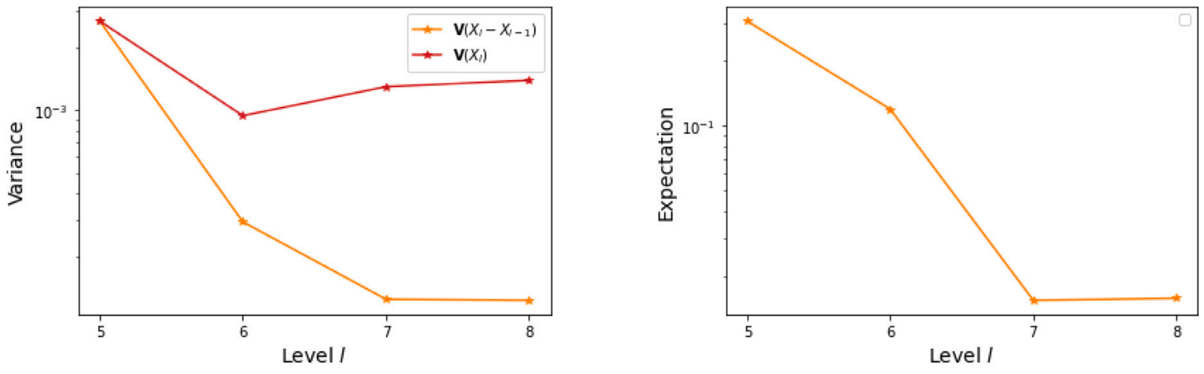
To check whether it is valid to use MLMC for this test case, we run steps 1 and 2 of Algorithm 1. Fig. 12(a) shows that $\mathbb{V}\left(X_l - X_{l-1}\right)$ decreases as the mesh gets finer and for all l is lower than $\mathbb{V}\left(X_l\right)$. Between the last two levels the decrease is only slight, probably due to the fact that, even with the convolution, there are still some features which increase in prominence at the finest mesh meaning $l = 7$ does not give the best approximation to $l = 8$, thus creating a relatively large variance. This effect is also seen in Fig. 12(b) which shows how $\mathbb{E}\left(X_l - X_{l-1}\right)$ varies as the mesh becomes finer. This plateau effect justifies the use of convolutions because without them, the approximation between $l = 7$ and $l = 8$ would be even worse.

Fig. 12(b) also shows that the numerical discretisation is approximately first order accurate, as with the test case in Section 3.3.1. In addition, the simulation passes the consistency and kurtosis checks from Section 2.1 and thus we can be confident that the MLMC algorithm is appropriate. Using convergence tests, the convergence values in (9) are $\alpha = 1.58$, $\beta = 1.46$ and $\gamma = 2.97$. As $\gamma > \beta$, we expect the order of convergence to be better than the standard Monte Carlo method.

Proceeding, we consider tolerance values of $\epsilon = [0.001, 0.0025, 0.005, 0.0075]$ and set a maximum limit of the finest mesh of $L_{\max} = 8$ because this corresponds to the mesh that the Boscombe Beach bed data is defined on. As $\gamma > \beta$, the optimum N_l are calculated using (14) and are shown in Fig. 13. Importantly the number of samples decreases as the mesh becomes finer.

As before we calculate the RMSE of the MLMC simulation by estimating the real solution $\mathbb{E}[X]$ through an Monte Carlo simulation with $\Delta x = 1123.8461/2^8 = 4.39\,\mathrm{m}$ and $\Delta y = 1604/2^8 = 6.27\,\mathrm{m}$ which is the same as the finest mesh in the MLMC simulation. The expected maximum value of $R_{u2\%}$ is found to be $0.1568\,\mathrm{m}$ (to four significant figures). Using this value, Figs. 14(a) and 14(b) confirm that the RMSE decreases as the level number increases and as ϵ decreases respectively, meaning the MLMC algorithm is working as expected. Interestingly, Fig. 14(a) shows that ϵ only has a notable effect on the error when the finest level is included. This makes sense because at previous levels, the convolved version of the bed is being used rather than the real data.

Finally Fig. 15 compares the total cost and accuracy of the MLMC versus Monte Carlo for this test case. It shows that MLMC with $\epsilon = 0.001$ achieves the same accuracy as the Monte Carlo simulation for 1.6% of the computational cost (as summarised in Table 3). We also note that, as has been the case for all the test cases considered in this work, the convergence with Monte Carlo is notably less uniform than that with MLMC and that with Monte Carlo the simulation must be run for longer than strictly necessary to ensure convergence, which is not the case for MLMC. Thus we have shown that for a real world test case with uncertain waveheight, MLMC can significantly decrease computational cost and increase result stability without compromising on accuracy.



(a) Variance of difference, $X_l - X_{l-1}$, compared to variance of single variable, X_l, at each level, l.

(b) Expectation of difference, $X_l - X_{l-1}$, at each level, l.

Fig. 12. Preliminary MLMC results for Boscombe Beach test case showing the use of MLMC is justified.

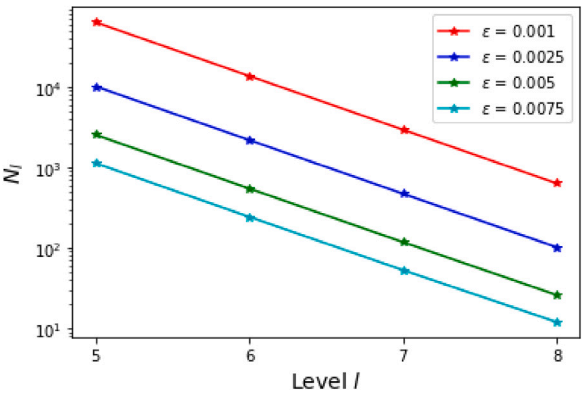


Fig. 13. Optimum number of samples required at each l for given tolerance value ϵ for Boscombe Beach test case.

Table 3
Computational cost improvement from using MLMC instead of Monte Carlo to achieve the same RMSE for the Boscombe Beach test case.

	MLMC	MC
RMSE (3sf)	1.38×10^{-4}	1.38×10^{-4}
Time to achieve RMSE (h)	5662	347294
MLMC speed-up factor	61.3	

4. Cumulative distribution functions

In an MLMC framework, the objective is normally to find the expectation of an output variable. However, to analyse risk in coastal problems, a key quantity of interest is the probability of an output variable exceeding a certain value, for example, a wave height exceeding a sea wall. As discussed in Section 1, this is a complicated output to compute because MLMC provides very few values on the same level from which to build that distribution.

To resolve this issue, we follow Gregory and Cotter (2017b) and use the inverse transform sampling method to evaluate the inverse cumulative distribution function (CDF), $F^{-1}(u)$, where $u \sim \mathcal{U}[0,1]$. If F is strictly increasing and absolutely continuous then $x \equiv F^{-1}(u)$ is unique. A simple consistent estimate for x is found by sorting the samples $X^i{}_{i=1,\ldots,N} \sim F$ such that $X^1 < X^2 < \cdots < X^N$ and then

$$\hat{F}^{-1}(u) = X^{\lceil N \times u \rceil}, \tag{29}$$

which, as argued in Gregory and Cotter (2017b), is consistent because it converges in probability to x as $N \to \infty$. For an MLMC approximation, the inverse CDF is then given by

$$F_L^{-1}(u) = R(X)_0^{\lceil N_0 \times u \rceil} + \sum_{l=1}^{L} \left(R(X)_l^{\lceil N_l \times u \rceil} - R(X)_{l-1}^{\lceil N_{l-1} \times u \rceil} \right), \tag{30}$$

where $R(X)_l^i$ represents the ith ordered statistic of X_l on each level l. Unlike with (2) there is no exact cancellation here as the approximations on each level are not unbiased.

Thus an ensemble of X_F^i can be generated, which are not samples from X_L, but consistent approximations to $F_L^{-1}(u)$. Thus, as $N_l \to \infty$, then $x \xrightarrow{p} F_L^{-1}(u)$ and the CDF of the MLMC approximation is given by

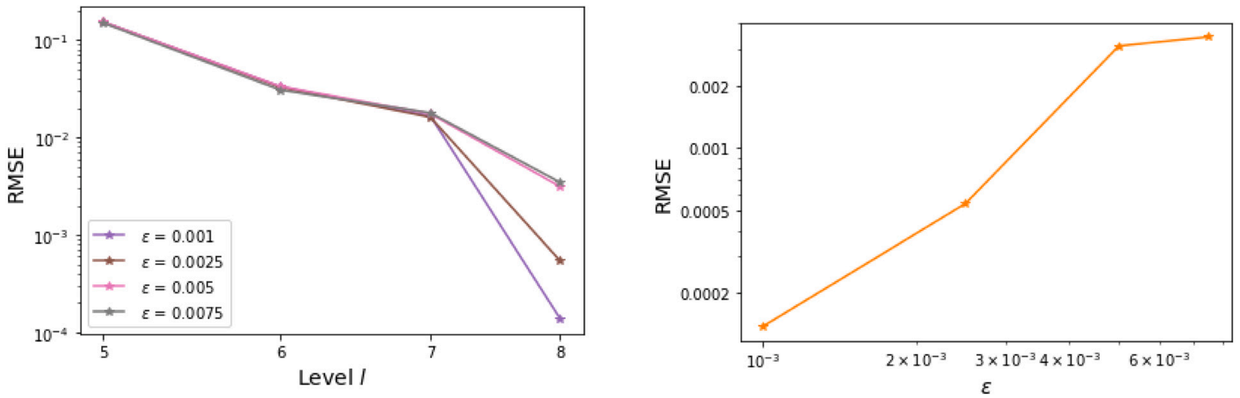
$$F(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{X_F^i \leq x}. \tag{31}$$

For more details on this method we refer the reader to Gregory and Cotter (2017b).

4.1. Applying inverse transform sampling to test cases

In this section, we apply the inverse transform sampling method to each of the four test cases discussed in Section 3. This method provides a CDF from which the probability that the output variable exceeds a certain value can be predicted. This is particularly useful for predicting the probability of extreme flooding/erosion events occurring, a useful tool in coastal risk assessment. To apply the method, we select the lowest ϵ value considered in each test case. The output is recorded at each level separately, ordered and then (30) is used to generate pseudo-samples on the finest level.

The results of applying this method to all four test cases are shown in Fig. 16, which also compares the MLMC results with those obtained by the Monte Carlo method. From these cumulative distribution functions, we can easily assess exceedance probability, for example, Fig. 16(a) shows that the probability that the maximum inundation distance will exceed 1090 m (recalling that the output is normalised and thus this is 0.87×1250 m) is 5% and thus, from a town planner's perspective allowing a permanent structure to be built in this area would be inadvisable. In all cases the MLMC results predict a similar CDF to the Monte Carlo method. The good approximation between MLMC and Monte Carlo is further shown by the small error norms between the two methods, shown in Table 4. Note each error norms has



(a) RMSE of MLMC at each level l for given tolerance value ϵ.

(b) RMSE of MLMC once it has converged for each tolerance value ϵ.

Fig. 14. RMSE of MLMC (23) for Boscombe Beach test case for varying tolerance value ϵ.

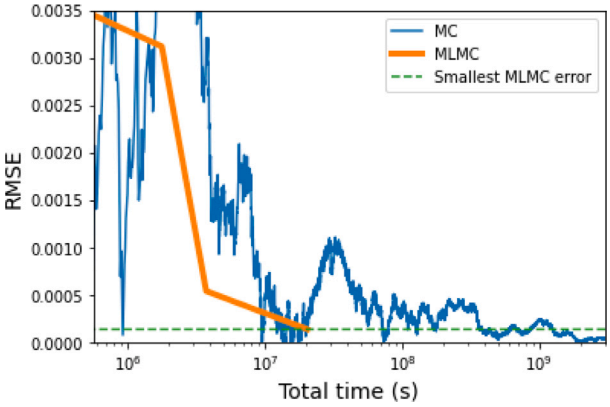


Fig. 15. Comparison of total cost and accuracy when using MLMC and Monte Carlo (MC) for Boscombe Beach test case.

Table 4
Error norms between MLMC and Monte Carlo cumulative distribution functions for each test case.

Test case	Normalised L2 error norm (3sf)
2D bed slope	0.00127
Morphology	0.00519
Boscombe Beach	0.00862

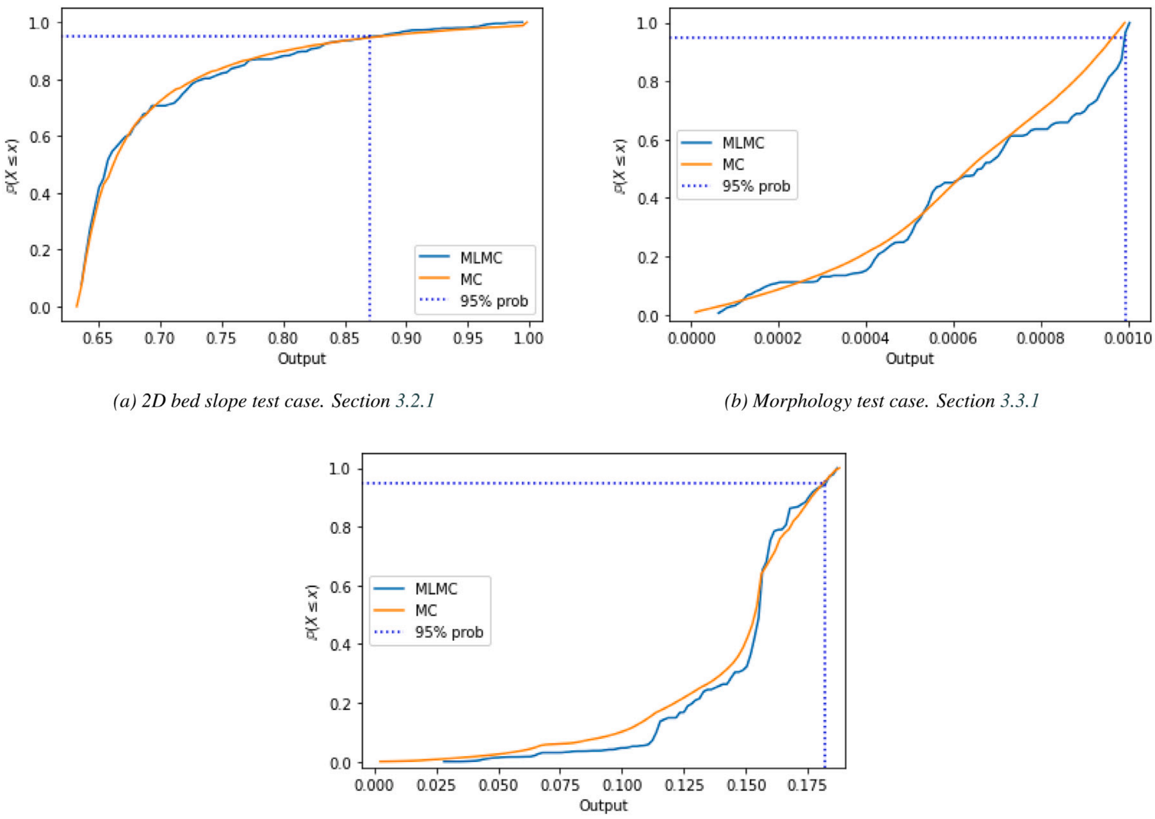
been normalised by dividing by the L2 norm of the respective MLMC CDF. Thus we conclude that we can use MLMC not only to find the expected value of output variables but also to determine the probability of extreme flooding and erosion events occurring and assess whether certain areas are at risk. Furthermore, we have shown that MLMC is able to generate accurate estimates of both the CDF and expected values for all test cases considered, in a much more computationally efficient manner. This strongly indicates that MLMC is able to consistently recreate Monte Carlo results over the whole parameter space and shows the quality of this method.

5. Conclusion

In this work, MLMC is applied for the first time to a complex hydro-morphodynamic model to evaluate risk in the coastal zone. We show that MLMC can be used successfully to estimate both the expected value of a hydro-morphodynamic output variable and its distribution for both 1D and 2D test cases and for both experimental and real-world test cases. We also show that the key advantage of MLMC over Monte Carlo is that it is a considerably faster approach whilst maintaining the same level of accuracy in the results. Another significant advantage is that the MLMC solution is stable, unlike the Monte Carlo one — in all test cases considered, decreasing ϵ (*i.e.*, the accuracy tolerance) in MLMC results in a more accurate solution. Therefore, there is no need to run the simulation for longer to ensure the solution has converged, again providing a computational cost advantage. These advantages mean that MLMC enables uncertainty analysis to be performed on previously unfeasible cases. For example, in Section 1 we discuss the test case in Callaghan et al. (2013). This test case is similar to both the morphology and Boscombe Beach test cases considered in our work, for which using MLMC resulted in a speed up of 43 and 61 times respectively. Given that the authors of Callaghan et al. (2013) estimate that it would take 4.5 millennia to run their test case using XBeach with Monte Carlo on a single processor, this suggests that using MLMC would 'only' take 90,000 years on the same processor. Whilst this is still a large cost, this test case would now not be outside the realm of certain supercomputers given our MLMC approach and the fact that our algorithm is parallelised.

For all test cases considered in this work, MLMC has no difficulties in dealing with the complex nature of the XBeach model. This is mainly due to the fact that MLMC only requires that the result from one mesh is a good approximation to the result from the next finest mesh, which makes it very flexible. This is an encouraging sign for more complex test cases, but even this loose restriction could become a limitation to the types of test cases where MLMC can be applied. However, a further advantage of Monte Carlo type methods is that they are very flexible and can be easily implemented using a wrapper approach without altering the underlying model. Therefore if this limitation becomes an issue with XBeach, it would be very simple to switch to a different coastal ocean model where this may not be an issue. In future work, we will consider more complex examples and take advantage of the flexibility of Monte Carlo type methods to consider multiple uncertain outputs at the same time.

This work has thus shown the first successful application of MLMC in the coastal engineering field and shown that it has the advantages of the Monte Carlo method when calculating both the expected value and the cumulative distribution function of key variables of interest, without the prohibitive computational cost.



(a) 2D bed slope test case. Section 3.2.1

(b) Morphology test case. Section 3.3.1

(c) Boscombe Beach test case. Section 3.3.2

Fig. 16. Cumulative distribution functions generated using inverse transform sampling method.

CRediT authorship contribution statement

Mariana C.A. Clare: Conceptualisation, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualisation. **Matthew D. Piggott:** Supervision, Writing – review & editing. **Colin J. Cotter:** Conceptualisation, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Computer code availability

The relevant code for the MLMC framework presented in this work can be found at https://github.com/mc4117/MLMC_XBeach.

Acknowledgements

We would like to thank Dr Alastair Gregory for his help in understanding how to calculate Cumulative Distribution Functions within an MLMC framework. The first author's work was funded through the EPSRC CDT in Mathematics for Planet Earth. All authors acknowledge funding from the Engineering and Physical Sciences Research Council (EPSRC) under grants EP/R512540/1, EP/L016613/1, EP/R007470/1 and EP/R029423/1.

References

Andrews, D.G., McIntyre, M.E., 1978. An exact theory of nonlinear waves on a Lagrangian-mean flow. J. Fluid Mech. 89 (4), 609–646.

Apel, H., Thieken, A.H., Merz, B., Blöschl, G., 2004. Flood risk assessment and associated uncertainty. Nat. Hazards Earth Syst. Sci. 4, 295–308.

Asbeck, W., 1953. Baron van, ferguson, ha, & schoemaker, hj 1953 new designs of breakwaters and seawalls with special reference to slope protection. Rome, Sect, In: Proc. 18th Int. Nav. Congress, vol. 2, p. 174.

Ballesio, M., Beck, J., Pandey, A., Parisi, L., von Schwerin, E., Tempone, R., 2019. Multilevel monte carlo acceleration of seismic wave propagation under uncertainty. GEM Int. J. Geomath. 10 (1), 22.

Bart, L.J.C., 2017. Long-Term Modelling with XBeach: Combining Stationary and Surfbeat Mode in an Integrated Approach (Masters thesis). TU Delft, Delft, The Netherlands.

de Beer, A., McCall, R., Long, J., Tissier, M., Reniers, A., 2020. Simulating wave runup on an intermediate–reflective beach using a wave-resolving and a wave-averaged version of xbeach. Coast. Eng. 163, 103788.

Caflisch, R.E., 1998. Monte Carlo and quasi-Monte Carlo methods. Acta Numer. 1998, 1–49.

Cai, Y., Gouldby, B., Hawkes, P., Dunning, P., 2008. Statistical simulation of flood variables: Incorporating short-term sequencing. J. Flood Risk Manag. 1, 3–12.

Callaghan, D.P., Ranasinghe, R., Roelvink, D., 2013. Probabilistic estimation of storm erosion using analytical, semi-empirical, and process based storm erosion models. Coast. Eng. 82, 64–75.

Clare, M.C.A., Leijnse, T.W.B., McCall, R.T., Diermanse, F.L.M., Cotter, C.J., Piggott, M.D., 2022. Multilevel multifidelity Monte Carlo methods for assessing coastal flood risk. Natural Hazards and Earth System Sciences Discussions 2022, 1–36. http://dx.doi.org/10.5194/nhess-2022-74.

Cliffe, K.A., Giles, M.B., Scheichl, R., Teckentrup, A.L., 2011. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. Comput. Vis. Sci. 14 (1), 3–15.

Cohn, N., Ruggiero, P., 2016. The influence of seasonal to interannual nearshore profile variability on extreme water levels: Modeling wave runup on dissipative beaches. Coast. Eng. 115, 79–92.

Díaz-Sánchez, R., López-Gutiérrez, J., Lechuga, A., Negro, V., 2014. Runup variability due to time dependence and stochasticity in the beach profiles: Two extreme cases of the spanish coast. J. Coast. Res. (70), 1–6.

Elfverson, D., Hellman, F., Målqvist, A., 2016. A multilevel Monte Carlo method for computing failure probabilities. SIAM/ASA J. Uncertain. Quantif. 4 (1), 312–330.

FLOODsite, 2009. Flood Risk Assessment and Flood Risk Management; An Introduction and Guidance Based on Experiences and Findings of FLOODsite (an EU-Funded Integrated Project). Deltares, Delft, The Netherlands.

Garvey, P., Lansdowne, Z., 1998. Risk matrix: An approach for identifying, assessing, and ranking program risks. Air Force J. Logist. 22, 18–21.

Giles, M.B., 2008. Multilevel Monte Carlo path simulation. Oper. Res. 56 (3), 607–617.

Giles, M.B., 2015. Multilevel Monte Carlo methods. Acta Numer. 24, 259–328.

Giles, M.B., Nagapetyan, T., Ritter, K., 2015. Multilevel Monte Carlo approximation of distribution functions and densities. SIAM/ASA J. Uncertain. Quantif. 3 (1), 267–295.

Gregory, A., Cotter, C.J., 2017a. A seamless multilevel ensemble transform particle filter. SIAM J. Sci. Comput. 39 (6), A2684–A2701.

Gregory, A., Cotter, C.J., 2017b. On the calibration of multilevel Monte Carlo ensemble forecasts. Q. J. R. Meteorol. Soc. 143 (705), 1929–1935.

Harris, D.L., Rovere, A., Casella, E., Power, H., Canavesio, R., Collin, A., Pomeroy, A., Webster, J.M., Parravicini, V., 2018. Coral reef structural complexity provides important coastal protection from waves under rising sea levels. Sci. Adv. 4 (2).

Hasselmann, K., Barnett, T., Bouws, E., Carlson, H., Cartwright, D., Enke, K., Ewing, J., Gienapp, H., Hasselmann, D., Kruseman, P., et al., 1973. Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP). Ergänzungsheft, pp. 8–12.

Holman, R., 1986. Extreme value statistics for wave run-up on a natural beach. Coast. Eng. 9 (6), 527–544.

Hunt, I.A., 1959. Design of seawalls and breakwaters. J. Waterw. Harb. Div. 85, 123–152.

Li, F., Gelder, P., Callaghan, D., Jongejan, R., Heijer, C., Ranasinghe, R., 2016. Probabilistic modeling of wave climate and predicting dune erosion. J. Coast. Res. 65, 760–765.

Malliouri, D., Memos, C., Soukissian, T., Tsoukala, V., 2021. Assessing failure probability of coastal structures based on probabilistic representation of sea conditions at the structures' location. Appl. Math. Model. 89, 710–730.

McCabe, M., 2011. Modelling Nearshore Waves, Runup and Overtopping. University of Manchester.

McCall, R.T., De Vries, J.V.T., Plant, N., Van Dongeren, A., Roelvink, J., Thompson, D., Reniers, A., 2010. Two-dimensional time dependent hurricane overwash and erosion modeling at Santa Rosa island. Coast. Eng. 57 (7), 668–683.

Van der Meer, J.W., Allsop, W.N., Bruce, T., De Rouck, J., Kortenhaus, A., Pullen, T., Schüttrumpf, H., Troch, P., Zanuttigh, B., 2018. EurOtop manual on wave overtopping of sea defences and related structures.

Melby, J., 2012. Wave Runup Prediction for Flood Hazard Assessment. US Army Corps of Engineers, Engineer Research and Development Center.

Melby, J., Caraballo-Nadal, N., Kobayashi, N., 2012. Wave runup prediction for flood mapping. In: Coastal Engineering Proceedings, vol. 1, management.79.

Melchers, R., Beck, A., 2018. Structural Reliability Analysis and Prediction. John Wiley & sons.

Park, H., Cox, D.T., 2016. Empirical wave run-up formula for wave, storm surge and berm width. Coast. Eng. 115, 67–78.

Pender, D., Karunarathna, H., 2013. A statistical-process based approach for modelling beach profile variability. Coast. Eng. 81, 19–29.

Reeve, D., Pedrozo-Acuña, A., Spivack, M., 2014. Beach memory and ensemble prediction of shoreline evolution near a groyne. Coast. Eng. 86, 77–87.

Riesenkamp, M., 2011. Probabilistic Modelling of Extreme Beach Erosion using XBeach (Masters thesis). TU Delft/ University of Queensland.

Roelvink, D., McCall, R., Mehvar, S., Nederhoff, K., Dastgheib, A., 2018. Improving predictions of swash dynamics in XBeach: The role of groupiness and incident-band runup. Coast. Eng. 134, 103–123.

Roelvink, D., Reniers, A., Van Dongeren, A., De Vries, J.V.T., McCall, R., Lescinski, J., 2009. Modelling storm impacts on beaches, dunes and barrier islands. Coast. Eng. 56 (11–12), 1133–1152.

Roelvink, D., Van Dongeren, A., McCall, R., Hoonhout, B., Van Rooijen, A., Van Geer, P., De Vet, L., Nederhoff, K., Quataert, E., 2015. XBeach Technical Reference: Kingsday Release. Technical report, Deltares, Delft, The Netherlands.

Ruggiero, P., Komar, P.D., McDougal, W.G., Marra, J.J., Beach, R.A., 2001. Wave runup, extreme water levels and the erosion of properties backing beaches. J. Coast. Res. 407–419.

Sánchez-Linares, C., de la Asunción, M., Castro, M.J., González-Vida, J.M., Macías, J., Mishra, S., 2016. Uncertainty quantification in tsunami modeling using multi-level Monte Carlo finite volume method. J. Math. Ind. 6 (1), 1–26.

Scheffner, N., Borgman, L., 1992. Stochastic time-series representation of wave data. J. Waterw. Port Coast. Ocean Eng. 118, 337–351.

Simmons, J.A., Harley, M.D., Marshall, L.A., Turner, I.L., Splinter, K.D., Cox, R.J., 2017. Calibrating and assessing uncertainty in coastal numerical models. Coast. Eng. 125, 28–41.

Stockdon, H.F., Holman, R.A., Howd, P.A., Sallenger, Jr., A.H., 2006. Empirical parameterization of setup, swash, and runup. Coast. Eng. 53 (7), 573–588.

Suanez, S., Cancouët, R., Floc'h, F., Blaise, E., Ardhuin, F., Filipot, J.-F., Cariolet, J.-M., Delacourt, C., 2015. Observations and predictions of wave runup, extreme water levels, and medium-term dune erosion during storm conditions. J. Mar. Sci. Eng. 3 (3), 674–698.

Toimil, A., Losada, I., Camus, P., Diaz-Simal, P., 2017. Managing coastal erosion under climate change at the regional scale. Coast. Eng. 128, 106–122.

Unguendoli, S., 2018. Propagation of Uncertainty Across Modeling Chains to Evaluate Hydraulic Vulnerability in Coastal Areas (Ph.D. thesis). Università Di Bologna, Bologna, Italy.

Van Gent, M.R., Coeveld, E.M., De Vroeg, H., Van De Graaff, J., 2007. Dune erosion prediction methods incorporating effects of wave periods. In: Coastal Sediments '07 - Proceedings of 6th International Symposium on Coastal Engineering and Science of Coastal Sediment Processes. pp. 612–625.

Villaret, C., Kopmann, R., Wyncoll, D., Riehme, J., Merkel, U., Naumann, U., 2016. First-order uncertainty analysis using algorithmic differentiation of morphodynamic models. Comput. Geosci. 90, 144–151.

Vousdoukas, M.I., Mentaschi, L., Voukouvalas, E., Bianchi, A., Dottori, F., Feyen, L., 2018. Climatic and socioeconomic controls of future coastal flood risk in europe. Nature Clim. Change 8 (9), 776–780.

Zurich Flood Resilience Alliance, 2019. The Flood Resilience Measurement for Communities (FRMC).