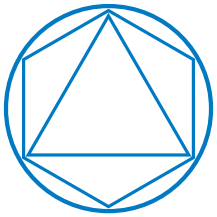


Technical University of Munich

Department of Mathematics

Sampling methods with application in tsunami simulation

Bachelorthesis

by

Erik Steiger

Supervisor:	Prof. Dr. Hans-Joachim Bungartz
Advisor:	Dr. Anne Reinarz
Submission Date:	30.09.2019

I assure the single handed composition of this bachelor's thesis is only supported by declared resources.

Moscow, 30.09.2019

Erik Steiger

Abstract

Zusammenfassung auf Deutsch

Diese Arbeit stellt Sampling-Methoden zur Generierung von Stichproben für die Bathymetrie in Anwendung mit Tsunami-Simulationen vor. Das Verfahren nimmt hierbei ein gröberes Raster der Bathymetrie über das die statistische Abweichung angewandt wird. Dieses Raster wird anschließend interpoliert und auf die gemessenen Bathymetriedaten addiert. Zur Bewertung der Ergebnisse nehmen wir vorhandene Daten des Tohoku Tsunamis aus dem Jahr 2011 und vergleichen die mit verschiedenen Parametern simulierten Monte Carlo Ergebnisse mit dem Resultat, welches ohne Unsicherheit gerechnet wurde.

Summary in english

This thesis presents sampling methods for the sample generation of bathymatry in application with tsunami simulation. The method samples a deviation with dependent variance over a coarser grid, i.e every nth data point, of the measured bathymatry and adds the deviation, where the missing points are interpolated, on the original bathymatry. We use available data of the Tohoku tsunami in 2011 to compare the Monte Carlo estimate of this method with different parameters to the estimate computed without a uncertainty.

Keywords — *sampling methods, bathymetry, tsunami, simulation, Monte Carlo, uncertainty quantification, PDE, Shallow water equations*

Contents

Notation

The following list covers a part of the notation used within this thesis.

General

$\chi_A(x)$	$:= \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$.	The indicator function.

Partial differential equations

∇	$:= \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_1} \end{pmatrix}$.	The gradient operator.
$D^k u(x)$	$:= \{\, D^\alpha u(x) \mid \lvert\alpha\rvert = k \}$.	The set of all partial derivatives of order k.
u_{x_i}	$:= \frac{\partial u}{\partial x_i}$.	

Probability theory and statistics

$\mathcal{N}(\mu, \sigma^2)$	Prob. density: $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	Normal dist. with mean μ and variance σ^2.

Chapter 1

Introduction

People have always attempted to predict the outcome of events and processes in nature, particularly those who have an impact on the daily life of millions of people. The best known example is the weather forecast. Although affecting nearly everyone's life the impact is considerably small nowadays. Other more impactful events like disasters and catastrophes, in our example an earth quake and the resulting tsunami, bear much greater consequences. The accurate estimation and prediction of such events is crucial for millions of lives. We came up with very accurate models for these events, but many of them are so complex that it would be unfeasible to use them on a larger scale, for instance larger coastal regions. In this case we use simplified methods. The simulation of tsunamis could be modelled with the three-dimensional Navier-Stokes equations but regarding the size of the computational domain, we will use a simplified derivation - the shallow water equations (SWE). Of course even the Navier-Stokes equations are only strong simplifications of the real process but for the right use case even the SWE computes surprisingly good results, given the right input parameters. For the SWE we have the initial water height, averaged water velocities and the 'topology of the sea ground', bathymetry, as input parameters. The data of the bathymetry consists hereby of measurements of the sea ground, using for instance the single beam method or the advanced multi beam method. Naturally, like all measurements, they are not perfect and we would expect a statistical deviation from the real value. Running the simulation with a measurement, slightly different from the real value, could give us a result which differs greatly from the real event. To avoid this, one would run the simulation on many measurement samples and compute with the results a confidence interval where we expect to find the real value with a given probability. But this would be obviously practically unfeasible. We cannot measure large areas like the east coast of Japan fifty times. Instead we take the one measurement we have and afflict it with an uncertainty. The method of generating the sample of the bathymatry plays an important role. The naive method, where we deviate every data point of our bathymetry, gives a poor result because the number of parameters which are introduced to the uncertainty is equal of the 7 million data points of the bathymatry. The proposed sampling method in this thesis uses in contrast a coarser discretisation of the domain and so reduces the number of uncertainty parameters to a moderate amount while keeping the finer structures, like steep hills and valleys. Compared to the naive approach this method leads to a better estimate and is considerably better than the result of the simulation without uncertainty.

Chapter 2

Preliminaries

This chapter gives a short overview of concepts used in the theory of partial differential equations, measure and probability theory as well as statistics.

2.1 Partial differential equations

This section follows the introductory part of [15].

Definition 2.1.1 (Partial differential equation, PDE)**.** *Let U be an open subset of $\mathbb{R}$ and fix an integer $k \geq 1$. An expression of the form*

$$F(D^k u(x), D^{k-1}u(x), \dots, Du(x), u(x), x) = 0 \quad (x \in U) \tag{2.1}$$

*is called a k^{th}-*order partial differential equation (PDE)*, where*

$$F : \mathbb{R}^{n^k} \times \mathbb{R}^{n^{k-1}} \times \cdots \times \mathbb{R}^n \times \mathbb{R} \times U \to \mathbb{R}$$

is given and $u : U \to \mathbb{R}$ is the unknown.

Definition 2.1.2. *An expression of the form*

$$\mathbf{F}(D^k \mathbf{u}(x), D^{k-1}\mathbf{u}(x), \dots, D\mathbf{u}(x), \mathbf{u}(x), x) = \mathbf{0} \quad (x \in U) \tag{2.2}$$

*is called a k^{th}-*order system of partial differential equations*, where*

$$\mathbf{F} : \mathbb{R}^{mn^k} \times \mathbb{R}^{mn^{k-1}} \times \cdots \times \mathbb{R}^{mn} \times \mathbb{R}^m \times U \to \mathbb{R}$$

is given and $\mathbf{u} : U \to \mathbb{R}^m$, $\mathbf{u} = (u^1, \dots, u^m)$ is the unknown.

Remark 2.1.1. *The term* PDE *is also used for a system of partial differential equations.*

Definition 2.1.3. *A PDE* (2.1) *is called linear if it has the following form*

$$\sum_{|\alpha| \leq k} a_\alpha(x) D^\alpha u = f(x) \tag{2.3}$$

for given functions a_α and f. The linear PDE is homogeneous *if $f \equiv 0$. See the examples below and note that the Maxwell's equations form a system of partial differential equations.*

Linear PDEs

$$\Delta u = \sum_{i=1}^{n} u_{x_i x_i} = 0. \qquad u_t - \Delta u = 0. \qquad \begin{cases} \mathbf{E}_t = \boldsymbol{\nabla} \times \mathbf{B} \\ \mathbf{B}_t = -\boldsymbol{\nabla} \times \mathbf{E} \\ \boldsymbol{\nabla} \cdot \mathbf{B} = \boldsymbol{\nabla} \cdot \mathbf{E} = 0. \end{cases}$$

Laplace's equation — Heat equation — Maxwell's equations

Non-linear PDEs

$$\mathbf{u}_t - \nabla^2 \mathbf{u} = \mathbf{f}(\mathbf{u}) \qquad \begin{cases} \mathbf{u}_t + \mathbf{u} \cdot D\mathbf{u} - \nabla^2 \mathbf{u} = -Dp \\ \boldsymbol{\nabla} \cdot \mathbf{u} = 0. \end{cases}$$

Reaction-diffusion system — Navier-Stokes equations for incompressible, viscous flow

2.1.1 Hyperbolic systems of first-order equations

Given a system of linear first-order partial differential equations having the form

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{i=1}^{d} \mathbf{B}_i \frac{\partial \mathbf{u}}{\partial x_i} = \mathbf{f} \tag{2.4}$$

with $\mathbf{u} : \mathbb{R}^d \times [0, \infty) \to \mathbb{R}^m$, $\mathbf{u} = (u^1, \ldots, u^m)$ as the unknown,
and given functions $\mathbf{B}_i : \mathbb{R} \times [0, \infty) \to \mathbb{M}^{m \times m}$, $\quad \mathbf{f} : \mathbb{R} \times [0, \infty) \to \mathbb{R}^m$.

Definition 2.1.4. *The system* (2.4) *is called* hyperbolic *if the matrix* $\mathbf{B}(x, t; y) := \sum_{i=1}^{d} y_i \mathbf{B}_i(x, t)$ *is diagonalizable for each* $x, y \in \mathbb{R}^d$, $t \geq 0$.

2.2 Measure and probability theory

For a complete overview regarding measure and probability theory please see [14].

2.2.1 Probability space

Definition 2.2.1. *A probability space is a triple* $(\Omega, \mathcal{F}, \mathbb{P})$*, where*

- Ω*: the* sample space*; set of all possible outcomes*
- $\mathcal{F}$*:* σ-algebra*; set of events*
- $\mathbb{P} : \mathcal{F} \to [0, 1]$ probability measure *that satisfies:*
  1. $\mathbb{P}(\varnothing) = 0$
  2. $\mathbb{P}(\Omega) = 1$
  3. *if* $\mathcal{A}_1, \ldots \in \mathcal{F}$ *with* $\mathcal{A}_i \cap \mathcal{A}_j = \varnothing$*, then*

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} \mathcal{A}_i \right) = \sum_{i=1}^{\infty} \mathbb{P}(\mathcal{A}_i)$$

Definition 2.2.2. *A random variable is a real-valued function* $X : \Omega \to \mathbb{R}$ *defined on* Ω *if it holds that for every Borel set* $\mathcal{B} \subset \mathbb{R}$ *that* $X^{-1}(\mathcal{B}) = \{ \omega : X(\omega) \in \mathcal{B} \} \in \mathcal{F}$.

Definition 2.2.3. *A random field is a collection of random variables* $\{X_t : t \in \mathcal{T}\}$ *indexed by* $t \in \mathcal{T}$*,* $\mathcal{T}$ *being a topological space.*

2.2.2 Expectation

Let $(\Omega, \mathcal{F}, \mathbb{F})$ be a probability space.

Definition 2.2.4. *Let $X : \Omega \to \mathbb{R}$ be a random variable. The* expected value *or* expectation *of X is defined as*

$$\mathbb{E}[X] := \int_\Omega X(\omega)\mathbb{P}(\omega)\,\mathrm{d}\omega, \tag{2.5}$$

provided the integral exists. Further if X has a density function $f(x)$ then the expectation of X is constituted by

$$\mathbb{E}[X] = \int_{\mathbb{R}} x f(x)\,\mathrm{d}x \tag{2.6}$$

Remark 2.2.1. *Often X just has a finite set of outcomes $x_1, x_2, \ldots, x_n$ and so (2.5) becomes the more commonly seen*

$$\mathbb{E}[X] = \sum_{i=1}^{n} x_i \mathbb{P}(X = x_i) \tag{2.7}$$

Definition 2.2.5. *Let X be a random variable with density f and $\mathbb{E}[X]$ exists. The* variance *of X is defined as*

$$\mathrm{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] \tag{2.8}$$

2.2.3 Strong Law of Large Numbers

Theorem 2.2.1 (Strong Law of Large Numbers)**.** *Let $X_1, X_2, \ldots$ be independent, identically distributed and $\mathbb{E}[|X_i|] < \infty$ with $\overline{X_n} := \frac{1}{n}\sum_{i=1}^{n} X_i$. Then one has*

$$\mathbb{P}\left(\overline{X_n} \xrightarrow[n\to\infty]{} \mathbb{E}[X_1]\right) = 1, \tag{2.9}$$

with $\mathbb{P}(\overline{X_n} \to \mathbb{E}[X_1]) = \mathbb{P}(\{\,\omega \mid \lim_{n\to\infty} \overline{X_n}(\omega) = \mathbb{E}[X_1]\})$.

Proof. [14, Chapter 2.4] □

2.3 Monte Carlo integration

Let $g : U \subset \mathbb{R} \to \mathbb{R}$ be a function and suppose we want to compute the integral $\int_a^b g(x)\,\mathrm{d}x$ (provided the integral exists). Trying to solve it analytically is often very difficult, if not impossible, so other options are the numerical integration [7] and the *Monte Carlo integration.*

Lemma 2.3.1. *Let X be a random variable with density $f(x)$ and let $g : \mathbb{R} \to \mathbb{R}$ be a measurable function. Then the expectation of $Y = g(X)$ is*

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x) f(x)\,\mathrm{d}x. \tag{2.10}$$

Given a random sample $X_1, X_2, \ldots, X_n$ from X, an unbiased estimator of $\mathbb{E}[g(X)]$ is the sample mean $\frac{1}{n}\sum_{i=1}^{n} x_i$.

Proof. [14, Chapter 1.6.3] □

Example 2.3.1. *Consider the initial problem of estimating* $\theta = \int_a^b g(x)\,\mathrm{d}x$. *Let* $X_1, X_2, \ldots, X_n$ *be a sample with* $X_i \sim \mathcal{U}(a,b)$ *uniformly distributed between* a *and* b *with density* $f(x) = \frac{1}{b-a}\chi_{[a,b]}(x)$ *for* $1 \leq i \leq n$. *It follows*

$$\theta = \int_a^b g(x)\,\mathrm{d}x = \frac{b-a}{1}\int_{-\infty}^{\infty} g(x)f(x)\,\mathrm{d}x = (b-a)\mathbb{E}\left[g(X)\right] \tag{2.11}$$

So a unbiased estimator $\hat{\theta}$ *for* $\theta = \mathbb{E}\left[g(X)\right]$ *is*

$$\hat{\theta} = \frac{b-a}{n}\sum_{i=1}^{n} g(X_i). \tag{2.12}$$

We know by the Strong Law of Large Numbers 2.2.1 that (2.12) *converges to* $\mathbb{E}\left[g(X)\right] = \theta$ *with probability 1 as* $n \to \infty$.

The variance of the estimator $\hat{\theta}$ is given by

$$\mathrm{Var}(\hat{\theta}) = \frac{(b-a)^2}{n^2}\,\mathrm{Var}\left(\sum_{i=1}^{n} g(X_i)\right) = \frac{(b-a)^2}{n}\,\mathrm{Var}\left(g(X)\right). \tag{2.13}$$

Chapter 3

The physical model

3.1 Fluid dynamics

A common way of describing a fluid flow is by expressing its velocity $\mathbf{u}$ at any point $\mathbf{x} \in D \subset \mathbb{R}^d$ and at any time $t \in I \subset \mathbb{R}$.

$$\mathbf{u} = \mathbf{u}(\mathbf{x}, t) \tag{3.1}$$

If we know the starting position $x_0 \in D$ of an element within the fluid at time $t_0 \in I$, we can compute its position at any other time $\hat{t} \in I$. If we take the Cartesian coordinates $\mathbf{x} = (x, y, z)$ then $\mathbf{u}$ denotes the three directional velocities u, v and w and (3.1) is the shorthand notation for

$$u = u(x, y, z, t), \quad v = v(x, y, z, t), \quad w = w(x, y, z, t).$$

It's important to note that $\mathbf{u}$ changes in both $\mathbf{x}$ and t, i.e if we fix a point in space its velocity will change over time. A special case is the *steady* flow which has the property

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{0}, \tag{3.2}$$

so $\mathbf{u}$ is independent of t and the velocity and direction of the flow is constant in every point in space over time.

Let $f : D \times I \subset \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}$ be a scalar function which describes some quantity of the fluid, for example the temperature T, the density ρ or the velocity in one direction. Whereas $\partial f / \partial t$ is the derivative of f at a fixed point $\mathbf{x}$, $\mathrm{d}f/\mathrm{d}t$ stands for the total derivative of f:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial}{\partial t} f\left[x(t), y(t), z(t), t\right] \tag{3.3}$$

where the change in time of x, y and z is given through $\mathbf{u}$:

$$\mathrm{d}x/\mathrm{d}t = u, \quad \mathrm{d}y/\mathrm{d}t = v, \quad \mathrm{d}z/\mathrm{d}t = w.$$

Using the chain rule we get

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial f}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}t} + \frac{\partial f}{\partial t}$$

and with (3.1) we can simplify it to

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} + v\frac{\partial f}{\partial y} + w\frac{\partial f}{\partial z}. \tag{3.4}$$

It is common to shorten (3.4) by using the *gradient operator* ∇:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial t} + (\mathbf{u} \cdot \nabla)f. \tag{3.5}$$

If we insert u, v and w in (3.5) for f we get

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}. \tag{3.6}$$

3.2 Fluid models

If we want to model a fluid we need to understand its physical properties and laws. As an illustrative example we will derive the equations of motion for an ideal fluid.

3.2.1 Equations of motion for an ideal fluid

We define an *ideal fluid* as one having the following properties:

1. It is incompressible, no part of the fluid can change its volume.
2. The density $\rho = \frac{\text{mass}}{\text{volume}}$ is constant over all fluid elements and for all time t.
3. The force exerted across a surface element $\nu\,\delta S$ within the fluid is $p\nu\,\delta S$, where p is the scalar function for pressure and ν the normal vector to the surface element δS.

For a volume V with surface S point (1.) implies that the net volume rate

$$\int_S \langle \mathbf{u}(\mathbf{x}), \nu(x) \rangle \,\mathrm{d}x = \int_V \boldsymbol{\nabla} \cdot \mathbf{u}(\mathbf{x}) \,\mathrm{d}x \tag{3.7}$$

at which the fluid is leaving V must be zero. The equality ensues from the divergence theorem 3.2.1.

Theorem 3.2.1 (The divergence theorem)**.** *Let $\Omega \subset \mathbb{R}^3$ open and bounded by a piecewise smooth closed surface S, and let n denote the outward unit normal vector on S. If f is continuously differentiable in a domain containing Ω, then*

$$\int_\Omega \boldsymbol{\nabla} \cdot f \,\mathrm{d}\Omega = \int_S f \cdot n \,\mathrm{d}S \tag{3.8}$$

Proof. [5, p.283] □

Since (3.7) must hold for every volume V we can conclude that

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \tag{3.9}$$

must hold everywhere in the fluid. Further if we integrate over the force exerted by the surrounding fluids on the volume V we get

$$-\int_S p\nu(x) \,\mathrm{d}x = -\int_V \nabla p \,\mathrm{d}x, \tag{3.10}$$

by again using the divergence theorem (3.8). Given ∇p is continuous we will assume that ∇p is constant over a volume δV. The net force on the volume due to the pressure of the surrounding

fluid results then in $-\nabla p\,\delta V$. Taking the gravitational force into account the resulting total force on a volume of the fluid is

$$(\rho\mathbf{g} - \nabla p)\delta V,$$

where $\mathbf{g}$ is the acceleration due to gravity. With Newton's second law of motion $F = ma$ this must be equal to

$$\rho\delta V\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}.$$

We end up with these two partial differential equations

$$\begin{aligned}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} &= -\frac{1}{\rho}\nabla p + \mathbf{g}\\ \boldsymbol{\nabla}\cdot\mathbf{u} &= 0,\end{aligned} \tag{3.11}$$

as the equations of motion for an ideal fluid. They are known as *Euler's equations* [4].

3.3 Navier-Stokes equations

The PDE (3.11) models the flow of a fluid in its most basic form and neglects physical properties like the viscosity of the fluid. We know from experience that water behaves different than honey. Introducing a stress tensor due to the viscosity we get the *Navier-Stokes equations for incompressible fluids*:

$$\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} = -\frac{1}{\rho}\nabla p + \mathbf{g}, \tag{3.12}$$

$\mathbf{u}$: flow velocities — ν : kinematic viscosity
$\mathbf{g}$: gravitational force — ρ : density

The effort for numerical solutions of (3.12) in two dimensions is manageable but for larger simulations especially in three dimensions the computational effort quickly exceeds the reasonable amount of time per simulation - even when using super computers. Flow simulations of lakes, the atmosphere or oceans near the shore are good examples. Fortunately the three of them share the property that the vertical scale *(depth)* of the fluid is small compared to the horizontal component. We can use this common feature to our advantage and average over the vertical scale and so derive the two-dimensional *shallow water equations*.

3.4 Shallow water equations

$$\begin{aligned}\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} &= 0\\ \frac{\partial hu}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial huv}{\partial y} &= -gh\frac{\partial b}{\partial x}\\ \frac{\partial hv}{\partial t} + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) + \frac{\partial huv}{\partial x} &= -gh\frac{\partial b}{\partial y}\end{aligned} \tag{3.13}$$

h : height of the water column — g : gravity
(u, v) : horizontal flow velocities — b : bathymetry

The Shallow water equations, also called Saint-Venant equations (after Adhémar Jean Claude Barré de Saint-Venant [1]), are a system of PDEs describing the flow in shallow regions like oceans, coastal regions, rivers and channels. A derivation can be found in [11].

Chapter 4

PDE solving

Due to the complexity of non-linear PDEs like (3.12) and (3.13) it is in general practically impossible to obtain analytical solutions. A numerical solving schema is used to get an approximation to the solution.

4.1 Discretization

Unlike analytic solutions, which are defined for every point in our domain, numerical solutions are only defined on a finite set of points often called grid nodes. The domain is subdivided into a set of non-overlapping cells (or elements) called mesh where the nodes are in general positioned at the centre of each cell or at the vertices depending on the discretisation. [20]
Taking the two-dimensional domain $D = I_1 \times I_2 \subset \mathbb{R}^2$ we will define the mesh as the pairwise disjoint partition into the cells

$$C_{i,j} := C_i^{(1)} \times C_j^{(2)} \subset I_1 \times I_2 \subset \mathbb{R}^2 \quad i \in X,\, j \in Y, \tag{4.1}$$

where X and Y are index sets.

4.2 Numerical schemes

For demonstrative purposes we will consider the incompressible Navier-Stokes equations (3.12):

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \nabla^2 \mathbf{u} = -\frac{1}{\rho}\nabla p + \mathbf{g}. \tag{4.2}$$

We would like to bring the equation into the following linearised matrix form:

$$\mathcal{M}\mathbf{u} = \mathbf{B} \tag{4.3}$$

where $\mathcal{M}$ is matrix of coefficients, $\mathbf{u}$ the unknown velocity vector for every cell in our mesh and $\mathbf{B}$ a right-hand side term. Given this we can use a variety of different linear equation solvers. There are many methods to bring equation (4.2) into the form (4.3). A very popular and often implemented method is the *finite volume method*.

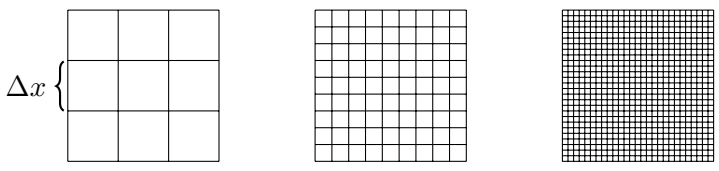


Figure 4.1: Uniform axiparallel quadrilateral meshes with different mesh sizes

4.2.1 Finite volume method

We will use the discretisation mentioned just above in chapter 4.1. In the second order finite volume method, the flow variables vary linearly across the cell. If we look at the owner cell with cell centroid P we call the adjacent cells neighbours with cell centroid N. The values of all cells are computed at the centre and they are connected over the so called cell faces.

We integrate the equation across the cell P

$$\int_V \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p - \mathbf{g}\right] \mathrm{d}V = 0. \tag{4.4}$$

Integration and summation are commutative thus we split up the integral into separate terms

$$\int_V \left[\frac{\partial \mathbf{u}}{\partial t}\right] \mathrm{d}V + \underbrace{\int_V \left[(\mathbf{u}\cdot\nabla)\mathbf{u}\right] \mathrm{d}V}_{\text{Convection}} = \underbrace{\int_V \left[\nu\nabla^2\mathbf{u}\right] \mathrm{d}V}_{\text{Diffusion}} + \int_V \left[-\frac{1}{\rho}\nabla p\right] \mathrm{d}V + \int_V \mathbf{g}\,\mathrm{d}V \tag{4.5}$$

where each term will be treated individually. Starting with the constant source term (like gravity):

$$\int_V \mathbf{g}\,\mathrm{d}V = \mathbf{g}V_p \tag{4.6}$$

where V_p is the volume of the owner cell P. We see that the constant source term doesn't depend on the velocity at all and will only contribute to the right-hand side of the matrix form equation (4.3). For source terms $\mathbf{g}^* = s\mathbf{u}$ which depend linearly on the velocity $\mathbf{u}$ we would get

$$\begin{aligned}\int_V [s\mathbf{u}] \,\mathrm{d}V &= s_p \int_V \mathbf{u}\,\mathrm{d}V = s_p \int_V [\mathbf{u}_p + (x - x_p)\nabla\mathbf{u}_p] \,\mathrm{d}V \\ &= s_p\mathbf{u}_p \int_V \mathrm{d}V + \nabla\mathbf{u}_p \underbrace{\int_V (x - x_p)\,\mathrm{d}V}_{=0} = s_p\mathbf{u}_p V_p\end{aligned} \tag{4.7}$$

with s_p being a scalar of the source term and $\mathbf{u}_p$ the velocities of cell P. The second equality arises from the linearity of the velocity and the fifth integral is zero because of the definition of P being the centre. This yields either to a contribution of s_pV_p to the matrix $\mathcal{M}$ of $\mathcal{M}\mathbf{u} = \mathbf{B}$ *(implicit treatment)* or to a contribution of $s_p\mathbf{u}_pV_p$ to $\mathbf{B}$ *(explicit treatment)*.

$$\mathcal{M} = \begin{pmatrix} -s_1V_1 & 0 & \cdots & 0 \\ 0 & -s_2V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -s_mV_m \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} s_1\mathbf{u}_1V_1 \\ s_2\mathbf{u}_2V_2 \\ \vdots \\ s_m\mathbf{u}_mV_m \end{pmatrix}$$

Now we will look at the more sophisticated Convection and Diffusion terms. Using the divergence theorem 3.2.1 we get for the Convection term

$$\int_V [\boldsymbol{\nabla}\cdot(\mathbf{uu})] \,\mathrm{d}V = \int_S [\mathbf{u}(\mathbf{u}\cdot\nu)] \,\mathrm{d}S \tag{4.8}$$

where S is the surface of the cell and ν the unit normal vector. We can further split up the integral over the four faces of the cell

$$\int_S [\mathbf{u}(\mathbf{u}\cdot\nu)]\,\mathrm{d}S = \sum_{i=1}^{4}\int_{S_i}\mathbf{u}_i(\mathbf{u}_i\cdot\nu_i)\,\mathrm{d}S_i \tag{4.9}$$

and approximate the integral over each face by the value at the face centre f

$$\sum_{i=1}^{4}\int_{S_i}\mathbf{u}_i(\mathbf{u}_i\cdot\nu_i)\,\mathrm{d}S_i \approx \sum_{i=1}^{4}\mathbf{u}_{\mathrm{f},i}(\mathbf{u}_{\mathrm{f},i}\cdot\nu_i)S_i. \tag{4.10}$$

We have to interpolate the velocity at the face centre $\mathbf{u}_\mathrm{f}$. For this we can choose out of a variety of face interpolation schemes *(Upwind, Central Differencing,...)*. The resulting sum over the four faces of every cell will contribute to the coefficient matrix $\mathcal{M}$ in a way that represents the connectivity of the cells. The diffusion term is treated in a similar manner and will not be discussed here. Please see [6] for a more detailed and complete insight as well as a proof of equation (4.10).

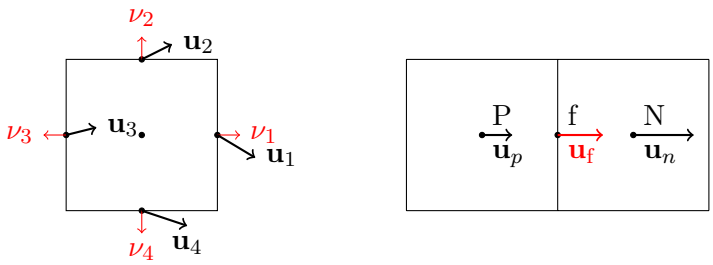


Figure 4.2: Cell fluxes of the finite volume scheme.

4.3 ExaHyPE - A hyperbolic PDE solver

The ExaHype engine offers ingredients to solve first-order hyperbolic PDEs in the following form [26]:

$$\mathbf{P}\frac{\partial\mathbf{Q}}{\partial t} + \nabla\cdot\mathbf{F}(\mathbf{Q}) + \sum_{i=1}^{d}\mathbf{B_i}(\mathbf{Q})\frac{\partial\mathbf{Q}}{\partial x_i} = \mathbf{S}(\mathbf{Q}) + \sum\delta_i \tag{4.11}$$

$\mathbf{P}$: material matrix	$\mathbf{S}$: algebraic source terms
$\mathbf{F}$: conserved flux vector	$\sum\mathbf{B_i}(\mathbf{Q})\frac{\partial\mathbf{Q}}{\partial x_i}$: non-conservative flux
$\mathbf{Q}$: state vector	$\sum\delta_i$: source terms

We can bring the shallow-water equations (SWE) into the respective form

$$\frac{\partial}{\partial t}\begin{pmatrix}h\\hu\\hv\\b\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}hu\\hu^2+\frac{1}{2}gh^2\\huv\\0\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}hv\\huv\\hv^2+\frac{1}{2}gh^2\\0\end{pmatrix} + \begin{pmatrix}0\\hgb_x\\hgb_y\\0\end{pmatrix} = 0. \tag{4.12}$$

The Exahype engine uses either the Finite Volume scheme or a Arbitrary High-Order Discontinuous Galerkin (ADER-DG) method proposed in [12]. ADER-DG is a more sophisticated method where the flow inside a cell is interpolated by a polynomial of arbitrary order. So even on a coarser mesh it can consider finer characteristics of the flow - given the flow is continuous. Unfortunately the SWE develops discontinuities in the coastal flood plains, called wetting and drying zones. In these zones water inundates and recedes as it is driven by tides and waves which results in moving boundary conditions for the SWE because they are only defined for wet areas, i.e water column height greater than zero. Problems of moving boundary conditions for the shallow water equations are often referred as wetting and drying problems. To cope with this a *limiting* method is proposed. The method first produces a candidate solution with the ADER-DG method. Then a set of detection criteria is applied to every cell of the solution. If a cell violates a condition then the last time step is recomputed using a more robust finite volume scheme on a smaller grid to ensure accuracy. This method combines the high-order accuracy of the ADER-DG scheme with the shock-capturing capabilities of the Finite Volume schemes. [21, 13]

Chapter 5

Uncertainty Quantification

> *"UQ studies all sources of error and uncertainty, including the following: systematic and stochastic measurement error; ignorance; limitations of theoretical models; limitations of numerical representations of those models; limitations of the accuracy and reliability of computations, approximations, and algorithms; and human error. A more precise definition is UQ is the end-to-end study of the reliability of scientific inferences."*
>
> - U.S. Department of Energy, *2009, p. 135*

In most modelling situations we want to simulate and predict a processes in nature, whether it is the weather, an earth quake or a tsunami. We have already seen in 3 that the models take often many simplification into account and thus can only give approximations of the real event. In addition to it these modelling errors they are also depended on the initial state - the input parameters. In the case of the shallow-water equations (3.13) these are the water height h and averaged velocities (u, v) and the bathymetry b, see 5.1.2. Some measurement methods return more precise results but we can never expect that the measurements will be exact. Every measurement is afflicted by a uncertainty. These uncertain parameters are than used to run the model which propagates the uncertainty to solution or the desired quantity of the event. If we can make statements about the uncertainties of the input parameter, for instance if it follows a statistical distribution, we would like to know the distribution of the solution and expected value. Based on this we can compute a confidence interval that covers the real event or quantity with a certain probability. This would enable us to include the variability of our input into our results.

5.1 Uncertainties and errors

We will talk about the different sources of uncertainties which arise in the modelling and simulation of processes in nature. [18]

5.1.1 Model errors

> *"Essentially, all models are wrong, but some are useful."* - *George E. P. Box,* [3]

One must understand that every model is just an approximation or imprecise representation of the underlying process. The number of possible dependencies with other objects and input parameters is too high to keep track of or even to incorporate into a model.

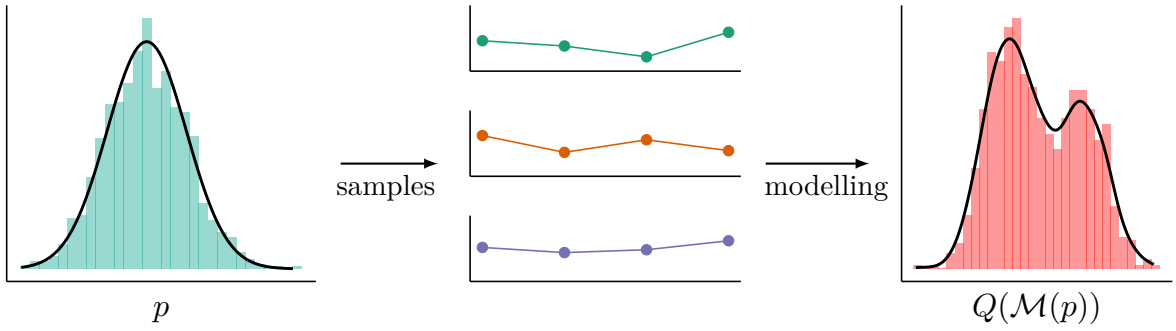


Figure 5.1: Parameter uncertainty propagation

We have seen that the Euler's equations (3.11) in 3.2 can be made more precisely by taking the viscosity of the fluid into account. One might have already noted the fact that the *kinetic viscosity* (3.12) although dependent on temperature was introduced as a constant over the whole fluid. This is again a simplification because the temperature can vary strongly in larger models, like simulations of oceans, lakes or the atmosphere. The choice of the right model plays a crucial role in the simulation or prediction of natural processes. Simpler models are often chosen ahead of sophisticated and more accurate models due to limited computational power, like in our case where we chose the Shallow water equations and not the Navier-Stokes Equations to get rid of one dimension and decrease the computational effort by a significant amount. Although very important we will not further discuss the uncertainty arising from modelling errors but rather concentrate on the uncertainty from input parameters.

5.1.2 Input uncertainties

Keeping in mind the approximative nature of models we still find many of them useful and their results are often close enough to real observations. But even if we would have a perfect model for a process we would have uncertain input parameters. These parameters - which determine the initial condition of our model - are often measured from nature and inherit a certain uncertainty. Given we have an input parameter p, which we know follows a certain distribution $\mathcal{P}$. We would draw samples $p_1, p_2, \ldots, p_n \sim \mathcal{P}$ from that distribution and run the simulation on each of them, getting the results $\mathcal{M}(p_1), \mathcal{M}(p_2), \ldots, \mathcal{M}(p_n)$. In most cases we are only interested in one quantity of the result, denoted by the output of the function Q. If we examine the quantities $Q(\mathcal{M}(p_1)), Q(\mathcal{M}(p_2)), \ldots, Q(\mathcal{M}(p_n))$ we see that they follow another distribution. The uncertainty of our initial parameter got propagated through the model. This approach, where we allow our parameters to follow a probability distribution, gives us the ability to predict the intervals where we expect our quantity to be found in the real world with a certain probability.
The following section will give a short introduction of one of the input parameters of the Shallow water equations - the bathymetry.

Bathymetry measurements

Bathymetry describes the measurement of water depth in oceans, seas and lakes analogous to topography for land surfaces. A commonly known method is the single beam method using a echo sounder. An echo sounder measures the time it takes for a pulse of sound to travel from

the source at the measuring platform to the sea bottom and return. The single beam method is prone to disturbances and errors through marine life forms, noise interference, absorption and unwanted reflections [2]. More advanced techniques like the multi-beam echo sounder achieve depth accuracies of less than 1% of the water depth, i.e the measurements are normally distributed around the estimated mean with standard deviation of 1% of the water depth. [9, 8]

Chapter 6

Monte Carlo methods

6.1 Monte Carlo methods for the Shallow water equations

Given the model $\mathcal{M}$ of the shallow-water equations (3.13) we need two input parameters:

- Q_0: the starting water height and averaged velocities. We will not threat them as uncertain but rather fix them for all samples.
- b: the bathymetry. This is our uncertainty parameter.

With these two parameters Q_0, b we can compute the respective solution $\mathcal{M}_\tau(Q_0, b) = Q_\tau(b)$, whereas τ stands for the specification and discretisation of the used schema, i.e used mesh-size and order. Let $\pi(Q_\tau(b)) = \pi$ denote a specific quantity of interest of our solution, for instance, the water height at a fixed position $\mathbf{x}$. With (2.10) it follows that the expectation of π is defined as

$$\mathbb{E}[\pi] = \int_\Omega \pi(Q_\tau(\omega))\mathbb{P}(\omega)\,\mathrm{d}\omega, \tag{6.1}$$

where Ω is the set containing all possible realisations of the bathymatry. The problem is that in practice we cannot compute the general solution Q_τ but rather solve it numerically for specific input parameters. To get an estimate of the expectation of π we use the Monte Carlo methods.

6.1.1 Simple Monte Carlo estimate

The main idea of the Monte Carlo method is to sample initial conditions and compute the corresponding solutions for each of them. The desired estimate is then computed by the average over the solutions.

Algorithm 1: Monte Carlo estimate (MC)

1 Generate M independent, identically distributed (iid.) input parameters $\beta_1, \beta_2, \dots \beta_M \sim \mathcal{B}$;

2 Compute the corresponding outputs $\pi(Q_h(\beta_i)) = \pi(Q_h^{(i)}), \quad i = 1, 2, \dots, M$;

3 Approximate the expectation by the sample mean

$$\hat{\pi}_{\mathrm{MC}} = \frac{1}{M}\sum_{i=1}^{M} \pi(Q_h^{(i)})$$

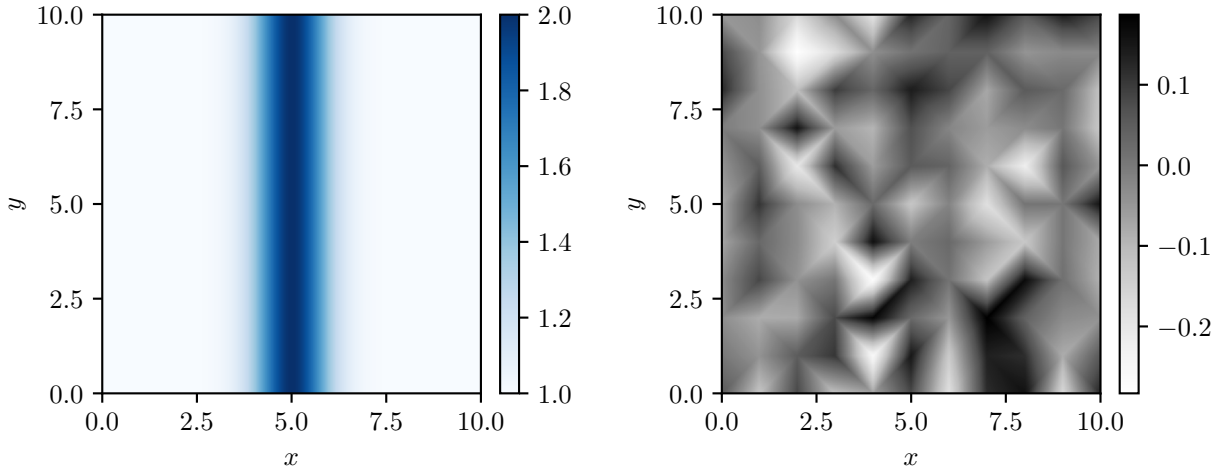


Figure 6.1: Initial water height $h(x, y)$ and bathymetry $b(x, y)$ for a sample of example 6.1.1

The variance of the MC estimate is

$$\mathrm{Var}(\hat{\pi}_{\mathrm{MC}}) = \frac{\mathrm{Var}(\pi(Q_h))}{M} \tag{6.2}$$

Example 6.1.1. *In this example we want to test the simple Monte Carlo method for the Shallow water equations and look at its convergence rate. Let* $D = [0, 10] \times [0, 10]$ *be the domain and* $T = [0, 1]$ *the computational time interval. The initial water height is a centred bell-shaped curve in the x dimension, i.e* $h(x, y) = 1 + \exp\{-(x - 5)^2\}$ *(Figure 6.1). The averaged velocities hu and hv are set to zero. We set the "real" bathymetry* $b_0(x, y) = 0$ *for all points in D and discretize it for the uncertainty quantification on a grid with* $11 \times 11 = 121$ *uniformly spread nodes. Following 5.1.2 we will afflict every node with a normal distributed uncertainty with mean* $\mu = 0$ *and standard deviation* $\sigma = 0.1$*. All points between the nodes are then interpolated linearly. In this first run we use the* ADERDG*-schema of the* Exahype *solver with order 1 computed on a mesh with* 75×75 *cells and corresponding mesh-size* $\Delta x = \frac{10}{75} \approx 0.134$ *at the time steps* $\{\, t_i = 0.01 \cdot i \mid t_i \in T\}$*. The quantity of interest will be the water height* $h(x, y, t)$ *at the point* $(x, y) = (7.5, 5)$*.*

The confidence interval (CI) to the confidence level $1 - \alpha$ and sample size n is computed with the following formula:

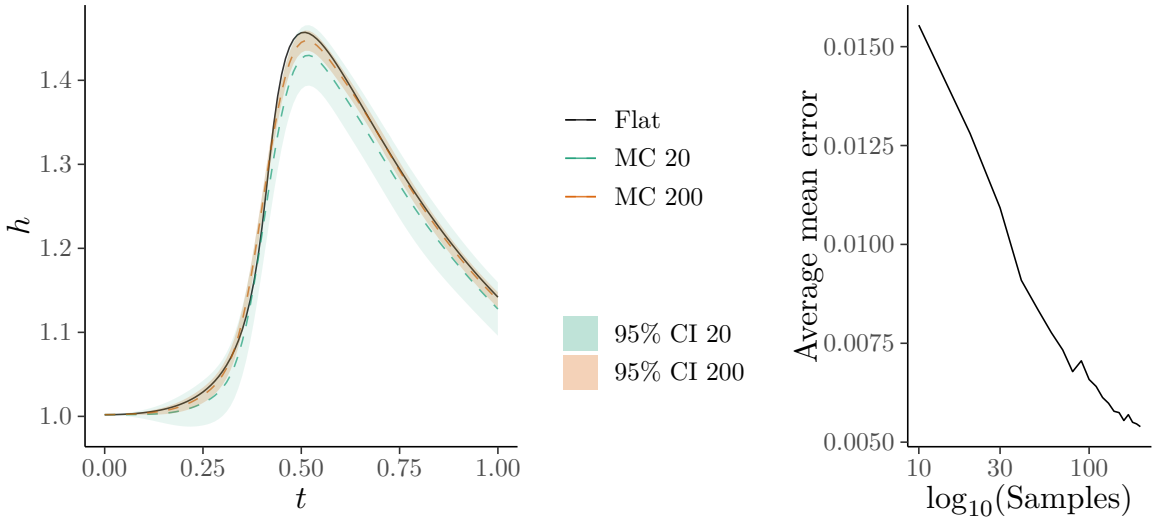
$$\mathrm{CI}_n = \left[\hat{\mu} - z\frac{\hat{\sigma}}{\sqrt{n}}, \hat{\mu} + z\frac{\hat{\sigma}}{\sqrt{n}}\right] \tag{6.3}$$

where $\hat{\mu}$ is the Monte Carlo estimate, z the $1 - \frac{\alpha}{2}$ quantile of the normal distribution, e.g. $z = \Phi^{-1}(1 - \frac{\alpha}{2})$ where Φ is the cumulative distribution function of the standard normal distribution, and $\hat{\sigma}$ is the unbiased estimator for the standard deviation of the Monte Carlo sample.

Mean squared error

$$\mathrm{MSE}(\hat{\pi}_{\mathrm{MC}}) = \mathbb{E}\left[(\pi(Q) - \hat{\pi}_{\mathrm{MC}})^2\right] = \underbrace{\mathbb{E}\left[\pi(Q) - \pi(Q_h)\right]}_{\text{discret. error}} + \underbrace{\frac{\mathrm{Var}(\pi(Q_h))}{M}}_{\text{MC error}} \tag{6.4}$$

The Mean squared error of the simple MC estimate consists of the discretisation error which arises from the inaccuracy of the used numerical scheme with the given discretisation (mesh size, order) and the MC error which is the variance of the method (cf. (6.2)). If we choose a more



		error		standard deviation	
method	M	average	maximum	average	maximum
MC 20	20	0.016	0.048	0.015	0.028
MC 200	200	0.005	0.028	0.005	0.007

Figure 6.2: Monte Carlo estimate $\hat{h}_{\mathrm{MC}}(7.5, 5)$ for example 6.1.1

accurate discretisation we can reduce the discretisation error but the "cost" to compute each $Q_h^{(i)}$ will rise. This will result in fewer computations (smaller M) and consequently to a higher MC error.

Complexity

Assume that the expected discretisation error $|\mathbb{E}[\pi(Q) - \hat{\pi}_{\mathrm{MC}}]|$ is of order $\mathcal{O}(h^\alpha)$ with $\alpha \in \mathbb{R}$ and the variance of the outputs $\mathrm{Var}(\pi(Q_h))$ of constant order $\mathcal{O}(1)$. The cost for each $Q_h^{(i)}$ will be assumed as $C_h = \mathcal{O}(h^{-\gamma})$ with $\gamma \in \mathbb{R}$. For a given tolerance tol of the error $(\pi(Q) - \hat{\pi}_{\mathrm{MC}})$ it follows from $\mathrm{MSE} = \mathcal{O}(tol^2)$ that $h = \mathcal{O}(tol^{\frac{1}{\alpha}})$ and $M = \mathcal{O}(tol^{-2})$. It follows for computational effort of the MC estimate:

$$\mathrm{Effort}(\hat{\pi}_{\mathrm{MC}}) = C_h M \lesssim tol^{-\frac{\gamma}{\alpha}} tol^{-2}. \tag{6.5}$$

To improve the Monte Carlo estimate without using a different discretisation we have to compute more samples. As long as the computation of each sample is rather cheap this is no problem but in many applications one simulation can get very expensive . In this case one can use the concept of control variates to reduce the variance and thus improve the estimate without increasing the number of samples.

6.1.2 Control variates

Let π be a function where we want to estimate $\theta = \mathbb{E}[\pi(X)]$. Given a function f where $\phi = \mathbb{E}[f(x)]$ is known and $\pi(X)$ and $f(X)$ are correlated one can check that for any constant c

$$\hat{\theta}_c = \pi(X) + c(f(X) - \phi) \tag{6.6}$$

is an unbiased estimator of θ. The variance

$$\mathrm{Var}(\hat{\theta}_c) = \mathrm{Var}(\pi(X)) + c^2 \,\mathrm{Var}(f(X)) + 2c \,\mathrm{Cov}(\pi(X), f(X)) \tag{6.7}$$

is a quadratic function of c and thus has a minimum. Let the minimum be at c^*. We get for the minimum variance

$$\mathrm{Var}(\hat{\theta}_{c^*}) = \mathrm{Var}(\pi(X)) - \frac{\mathrm{Cov}(\pi(X), f(X))^2}{\mathrm{Var}(\pi(X))}. \tag{6.8}$$

The reduction of variance is greater if $\pi(X)$ and $f(X)$ are strongly correlated. $f(X)$ is called a *control variate* for the estimator of $\pi(X)$ [10].
From this basic idea we can derive the multi-level Monte Carlo method where we use finer discretisations of our computational domain.

6.1.3 Multi-level Monte Carlo

In contrast to the (simple) Monte Carlo method we will now use solutions on different discretisation. Let $\tau_1, \tau_2, \ldots, \tau_L$ with $\tau_i = (h_i, o_i)$ denote a sequence of specifications for our schema, where h_i is the discretisation and o_i the used order. [17]

Algorithm 2: Multi-level Monte Carlo algorithm (MLMC)

Data: Sequence with specifications $\tau_1, \tau_2, \ldots, \tau_L$ with $h_1 > h_2 > \ldots > h_L$ and fixed order $o_i = o$. Sequence with sample sizes $\{M_i\}$ with $M_1 > M_2 > \ldots > M_L$.

1 **for** $l \in \{0, 1, \ldots, L\}$ **do**
2 Generate M_l independent, identically distributed (iid.) input parameters $\beta_{l,1}, \beta_{l,2}, \ldots, \beta_{l,M_l} \sim \mathcal{B}$
3 Compute the corresponding outputs $\pi(Q_{\tau_l}(\beta_i)) = \pi(Q^{(i)}_{\tau_l})$ and $\pi(Q_{\tau_{l-1}}(\beta_i)) = \pi(Q^{(i)}_{\tau_{l-1}})$, where $\pi(Q^{(i)}_{\tau_{-1}}) = 0 \quad i = 1, 2, \ldots, M_l$
4 Approximate the expectation with the sum

$$\hat{\pi}_{\mathrm{MLMC}} = \sum_{l=0}^{L} \frac{1}{M_l} \sum_{i=1}^{M_l} \left(\pi(Q^{(i)}_{\tau_l}) - \pi(Q^{(i)}_{\tau_{l-1}}) \right)$$

Note that the two outputs $\pi(Q^{(i)}_{\tau_l})$ and $\pi(Q^{(i)}_{\tau_{l-1}})$ are computed with the same input parameters (bathymetry) and thus are strongly correlated.

Example 6.1.2. *We will again consider the setting of example 6.1.1. The specification of the samples for the corresponding levels:*

l	cells	Δx	Cost
0	$21 \times 21 = 441$	$\frac{1}{21} \approx 0.0476$	0.08
1	$75 \times 75 = 5625$	$\frac{1}{75} \approx 0.0133$	1.0
2	$237 \times 237 = 56169$	$\frac{1}{237} \approx 0.0042$	10.0

The *average mean error* plot in figure 6.3 was computed by increasing the number of samples on each level level while keeping the ratio $9 : 3 : 1$ of samples per level.

Mean squared error

$$\mathrm{MSE}(\hat{\pi}_{\mathrm{MLMC}}) = \underbrace{\mathbb{E}\left[\pi(Q) - \pi(Q_{\tau_L})\right]}_{\text{discret. error level L}} + \underbrace{\sum_{l=0}^{L} \frac{\mathrm{Var}(\pi(Q_{\tau_l}) - \pi(Q_{\tau_{l-1}}))}{M_l}}_{\text{statistical error}} \tag{6.9}$$

The discretisation error has been decreased by the usage of a finer discretisations of the level L samples.

Complexity

With $V_l = \mathrm{Var}(\pi(Q_{\tau_l}) - \pi(Q_{\tau_{l-1}}))$ being the variance of differences and C_l the cost of the difference $(\pi(Q_{\tau_l}) - \pi(Q_{\tau_{l-1}}))$ we would like to know the optimal sample sizes M_l. Minimising the total work $\sum_{l=0}^{L} C_l M_l$ s.t. MSE $\simeq tol^2$ [24]:

$$M_l = \left\lceil tol^{-2} \sqrt{\frac{V_l}{C_l}} \left(\sum_{l=0}^{L} \sqrt{C_l V_l} \right) \right\rceil \tag{6.10}$$

For $h_l = h_0 s^{-l}$ assume $|\mathbb{E}\left[\pi(Q) - \pi(Q_{\tau_l})\right]| = \mathcal{O}(h_l^{\alpha})$, $V_l = \mathcal{O}(h_l^{\beta})$ and $C_l = \mathcal{O}(h_l^{-\gamma})$ with $2\alpha \geq \min\{\beta, \gamma\}$. Choosing $L = \mathcal{O}(tol^{\frac{1}{\alpha}})$ and M_l like in (6.10) we get

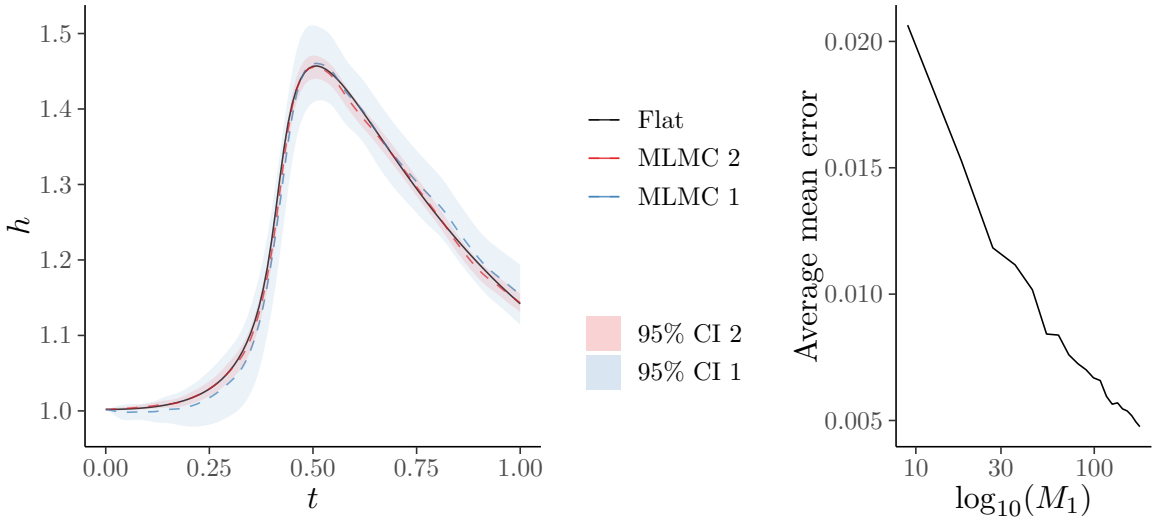
$$\mathrm{Effort}(\hat{\pi}_{\mathrm{MLMC}}) = \sum_{l=0}^{L} C_l M_l \lesssim \begin{cases} tol^{-2}, & \beta > \gamma \\ tol^{-2}(\log tol)^2, & \beta = \gamma \\ tol^{-2-\frac{\gamma-\beta}{2}}, & \beta < \gamma \end{cases}. \tag{6.11}$$

So to improve the complexity of the MLMC estimate we need to determine the the optimal maximum level L by estimating the error decay $|\mathbb{E}\left[\pi(Q) - \pi(Q_{\tau_l})\right]|$ and determine the optimal sample sizes $\{M_l\}_{l=0}^{L}$ by estimating the variance decay V_l. These can be computed by a MC estimate based on a pilot run. To avoid the high costs of the simulation on the finest level L one could use a extrapolation based on the results of previous levels.

Note that these results were worked out for finite volume schemes and cannot be directly applied to our used ADER-DG scheme. Nevertheless we expect the complexity analysis using the ADER-DG scheme to be analogical to that and just incorporate the order of the cell polynomials. [19, Theorem. 2.5]

6.1.4 Conclusion

Samples which are computed on a finer discretisation require more computational effort and are thus more costly. This also applies to higher order solutions. Considering the cost specification from example 6.1.2 which reflects the number of cells of the computational grid we can roughly compare the result of the MC estimate (MC 200) and the MLMC estimate (MLMC 2). We see no noteworthy improvement of the MLMC method against the (simple) MC method. The reason for this can be found firstly in a small improvement of the discretisation error. All of the simulations are computed using the ADER-DG scheme with order 3, which is in our usecase already very accurate. This means the discretisation error of the MC estimate (6.4) is small and the improvement to the finer level L negligible. Secondly the ExaHyPE solver uses a framework for the discretisation where the refinement is done by splitting up every two dimensional cell into 9 smaller cells. With such an aggressive refinement of order 3 in every dimension we cannot compute a MLMC method with more than two or three levels, because higher levels would



	M_l			error		standard deviation	
method	$l=1$	$l=2$	$l=3$	average	maximum	average	maximum
MLMC 1	18	6	2	0.01	0.034	0.02	0.034
MLMC 2	180	60	20	0.004	0.018	0.006	0.009

Figure 6.3: Multi-level Monte Carlo estimate $\hat{h}_{\text{MLMC}}(7.5, 5)$ for example 6.1.2.

rapidly blow up in size. Unfortunately the MLMC estimate is most effective when using many levels. We conclude that the efficiency improvement of the MLMC method is not decisive in our setup and we will not use the MLMC estimate for further simulations.

We will now concentrate on the sampling of our initial uncertainty parameter in the context of tsunami simulations.

Chapter 7

Tohoku 2011

The measurement of the water depth on our earth is in contrast to height measurements ashore not nearly as complete and accurate. The measurement data points in bathymetry are much more susceptible to uncertainties and many regions are only interpolated. For this reason one must be careful when working with bathymetry data and keep in mind the uncertainty of the data. One application were we need the underlying bathymetry is the simulation of tsunamis. In this example we will simulate the tsunami resulting from the 2011 Tohoku earthquake in the northeast region of Japan.

7.1 The Tohoku 2011 earthquake and tsunami

On the 11th March of 2011 at 2:46pm Japan standard time (JST) an earthquake of magnitude 9 erupted near the east coast of Tohoku, a region on the island Honshu. It was the strongest earthquake in Japan recorded till now. The earthquake and the resulting tsunami cost many people their life and led to serious damages at the Fukushima Nuclear Power Plant. [16]

Bathymetry

The used bathymetry data for the simulations of this region was provided by GEBCO [22]. The dataset consists of 7 million data points which cover an area of 28 million km^2 with a resolution of 2000m (distance between data points). Figure 7.1 shows the bathymetry of the underlying region. We will later on introduce a statistical deviation to the bathymetric data to perform an uncertainty quantification.

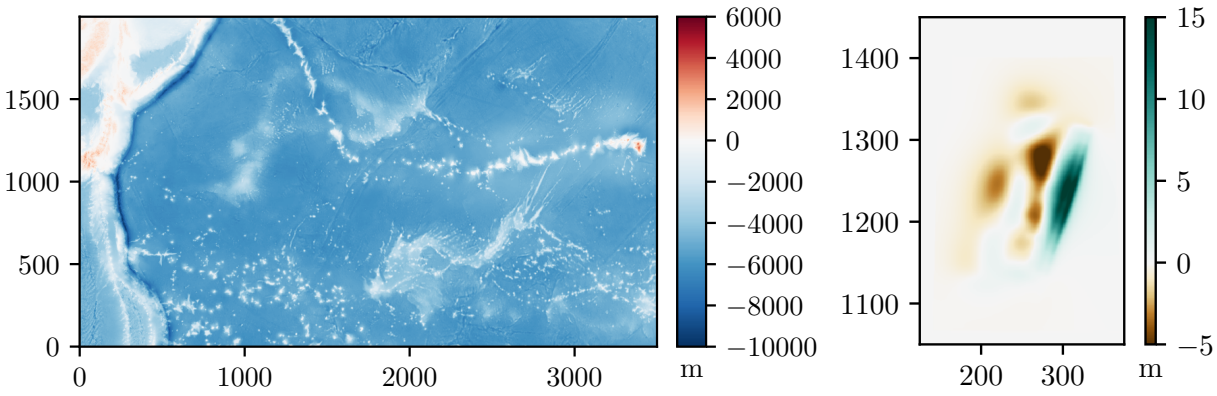


Figure 7.1: Tohoku bathymetry and displacement

Displacement

The displacement of the land masses caused by the earthquake will be set as initial water height, see Figure 7.1. As stated above the displacement will not play a role in the uncertainty quantification and will be fix for all simulations.

Buoy data

Will will compare our results to the historical data of the buoy station 21418 located at N38°44′42″ E148°39′54″ [25].

7.2 Simulation

7.2.1 Bathymetry sampling

The choice of the sampling of our input parameters - in our case the bathymetry - plays a decisive role of the results of the uncertainty quantification. The goal is to reduce the sources of uncertainties, namely the number of probability variables of one sample, to achieve convergence with a feasible number of samples while also giving the samples enough impact on the result to have a higher chance of covering the actual event. Our bathymetry data set is a two dimensional grid $\mathcal{B}$ of size 3500×2000 where each grid point has a value corresponding to his height above sea level. We set $B = (b_{ij}) \in \mathbb{R}^{(3500,2000)}$ as the matrix which holds the values. One can see in Figure 7.1 that parts of $\mathcal{B}$ match land. In the following we will present methods to generate samples from $\mathcal{B}$ with different uncertainties and discuss their usefulness.

Sampling method 1. *With this first very naive method we generate the samples by independently afflicting every grid point of $\mathcal{B}$ with a normal deviation. We have seen in 5.1.2 that the accuracy of measurements is depending on the depth. In conclusion our samples are from the random field 2.2.3*

$$\mathcal{B}^1 \sim \mathcal{N}\left(B, (pB)^2\right) \tag{7.1}$$

where p is a scalar with whom we can adjust the variance of the normal deviation. The normal deviation of a matrix will be defined as

$$\mathcal{N}\left(B, (pB)^2\right) := \left\{ \mathcal{N}\left(b_{ij}, (pb_{ij})^2\right) \mid (i,j) \in \{1,\dots,3500\} \times \{1,\dots,2000\} \right\}.$$

We can see in Figure 7.2 a 1-dimensional extract of samples with two different parameters p of this method compared to the actual bathymetry. Both are not very satisfactory. The samples are either to close to the reference data (for small variance, $p \ll 1$) or degenerate to a zigzag line with large jumps between adjacent points (for high variance, $(p \approx 1)$). For $p \ll 1$ this would introduce nearly no uncertainty into our model and for $(p \approx 1)$ it would result into a high variance of our samples. With every data point introducing a uncertainty parameter which leads to approximately 6.8 million sources of uncertainty we would need a impossible large number of samples. Beside the computational unfeasibility we would like to have samples which mimic possible real sea grounds. This is in Figure 7.2 obviously not the case.
In the next method we try to reduce the number of uncertainty parameters to a moderate amount by sampling over a coarser realisation of $\mathcal{B}$.

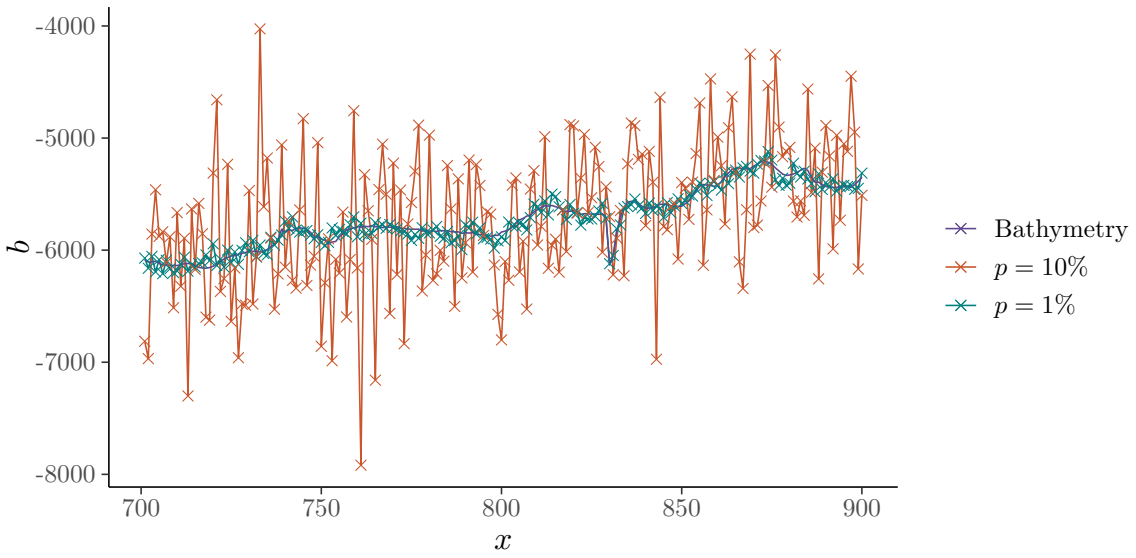


Figure 7.2: Naive sampling by afflicting every grid point independently, sampling method 1.

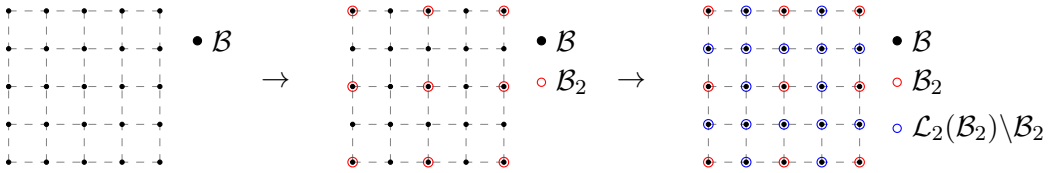
Sampling method 2. *In the first step we choose a coarser grid $\mathcal{B}_n$ of $\mathcal{B}$ with matrix B_n where n is the cell size of the new grid. Now we afflict every point of $\mathcal{B}_n$ with a uncertainty similar to 1. The missing points are then interpolated. We sample from*

$$\mathcal{B}^2 \sim \mathcal{L}_n \left(\mathcal{N} \left(B_n, (pB_n)^2 \right) \right). \tag{7.2}$$

where

$$\mathcal{L}_n : \mathbb{R}^{(k,l)} \to \mathbb{R}^{(3500,2000)}, \qquad k = \left\lfloor \frac{3500-1}{n} \right\rfloor + 1, \quad l = \left\lfloor \frac{2000-1}{n} \right\rfloor + 1 \tag{7.3}$$

denotes the interpolating operator. For the interpolation it is essential that the boundary points of $\mathcal{B}$ are included in $\mathcal{B}_n$ because we can only interpolate points inside the convex hull of $\mathcal{B}_n$.



The idea here is to get rid of the direct independence of adjacent points and the resulting fluctuations in the sampling method 1. Through the coarser grid and the linear interpolation one parameter (the height of one grid point) is affecting the points in his neighbourhood. Thru that we greatly decreased the number of parameters of the uncertainty and therefor need less samples for the estimate. It is noticeable, though, that the samples are highly influenced by the choice of the cell size n. Depending where the grid points of $\mathcal{B}_n$ are positioned some characteristics of the ground, like hills and indentations, are completely lost (cf. Figure 7.3).

Sampling method 3. *We take the basic idea of sampling method 2 and add the finer structures to the interpolation. We choose a coarser grid $\mathcal{B}_n$ and sample on that a normal deviation with dependent variance but mean $\mu = 0$ for all points. We interpolate the missing points over these uncertainties and add them to the original bathymetry matrix B. Concluding we get for the random field*

$$\mathcal{B}^3 \sim B + \mathcal{L}_n \left(\mathcal{N} \left(0, (\alpha + pB_n)^2 \right) \right) \tag{7.4}$$

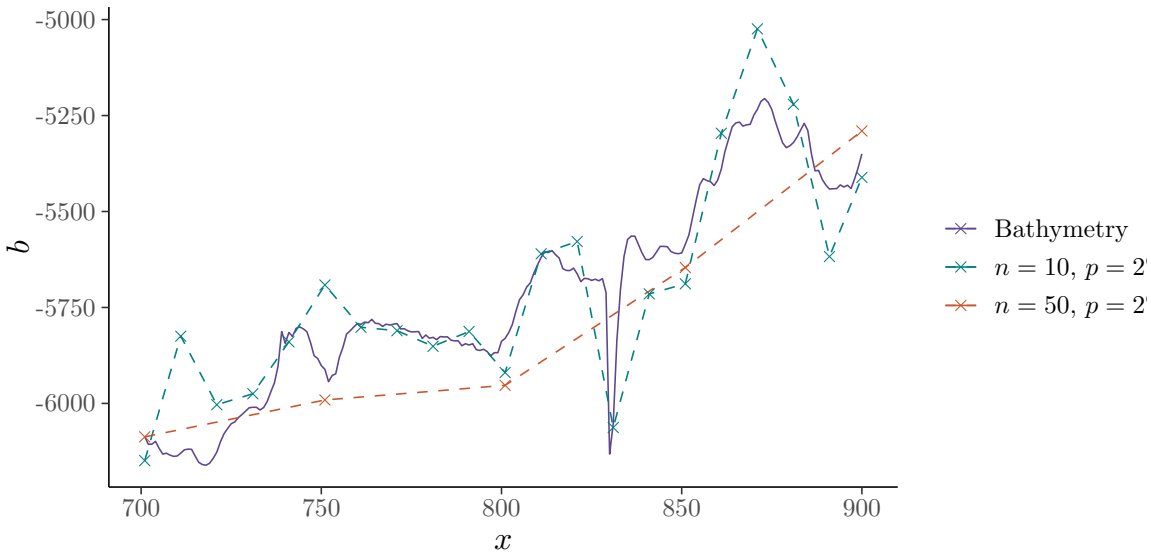


Figure 7.3: Coarser sampling with linear interpolation, sampling method 2.

We have added the parameter $\alpha \in \mathbb{R}$ to ensure if necessary a sufficient variability even in very shallow regions.

The results of this third sampling method are now satisfactory. While keeping the advantages of the second sampling method with linear interpolation we were able to add the finer structures of the ground. We have reduced the sampling space to samples with an arbitrary number of parameters and variance - by choosing n, α and p - while still keeping the features and characteristics of a real sea ground. We will use this last method for the following simulations.

Further modifications

- Often the bathymetric data set contains a additional matrix which holds the uncertainties of the measurements. If available one would use these values to determine the variance of the sampling.
- Many regions of the bathymatry are not measured directly but are just interpolated from the surrounding measurements. We would like to distinguish these regions and apply a higher variance there, depending on the distance to the nearest measured point. Given we have no further data to distinguish the interpolated areas we could use a machine learning ansatz in a preprocessing step to label the areas accordingly. A good indicator for interpolated areas is the "smoothness" of the data. The official GEBCO website states: "[...] there will be some areas that seem to show a much higher detail, and other areas that are relatively 'smooth'. Generally speaking, it is the smoother areas that are predicted whereas the areas of greatest detail are usually well-constrained by data." (cf. Figure 7.5) [23].
- The third sampling method could be modified to a multi-level method, where low levels sample on a coarser grid and influence the samples on a large scale while high levels with finer grids determine the finer structures of the samples. This method would have a greater control over the sample characteristics.

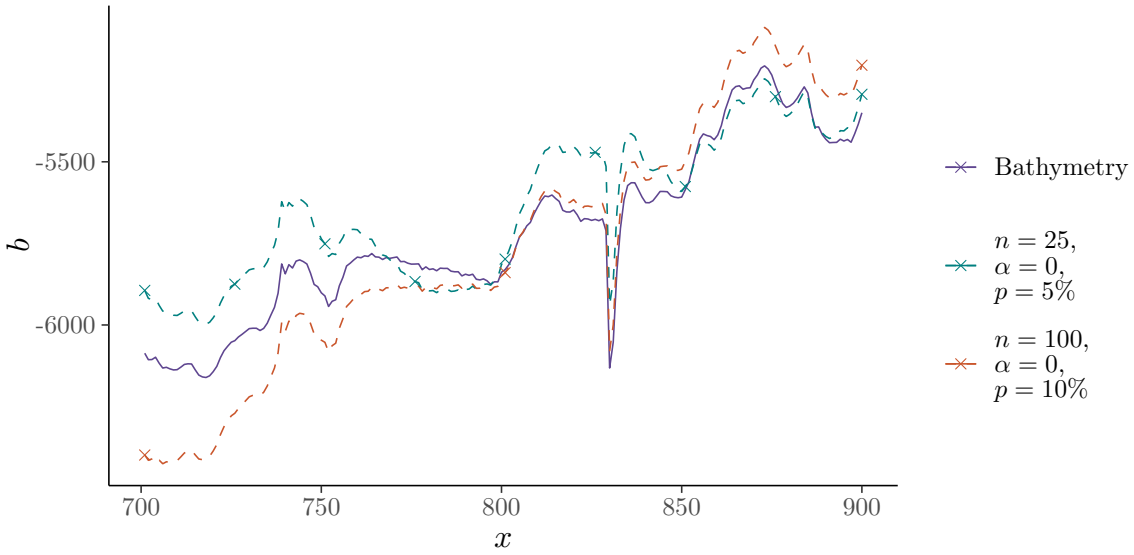


Figure 7.4: Coarser sampling while keeping the finer structures, sampling method 3.

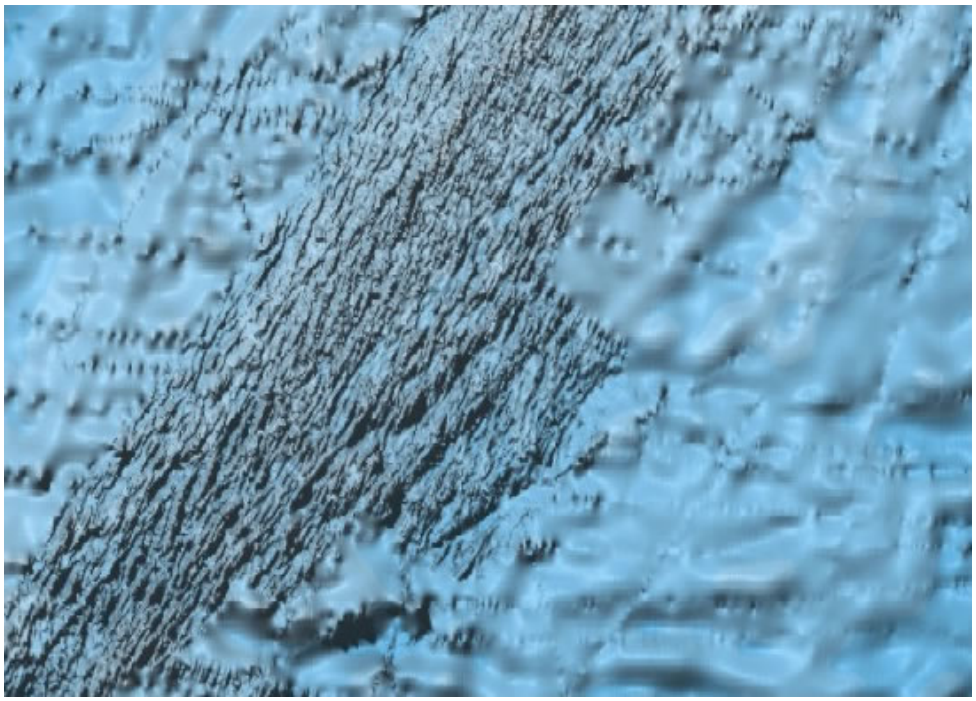
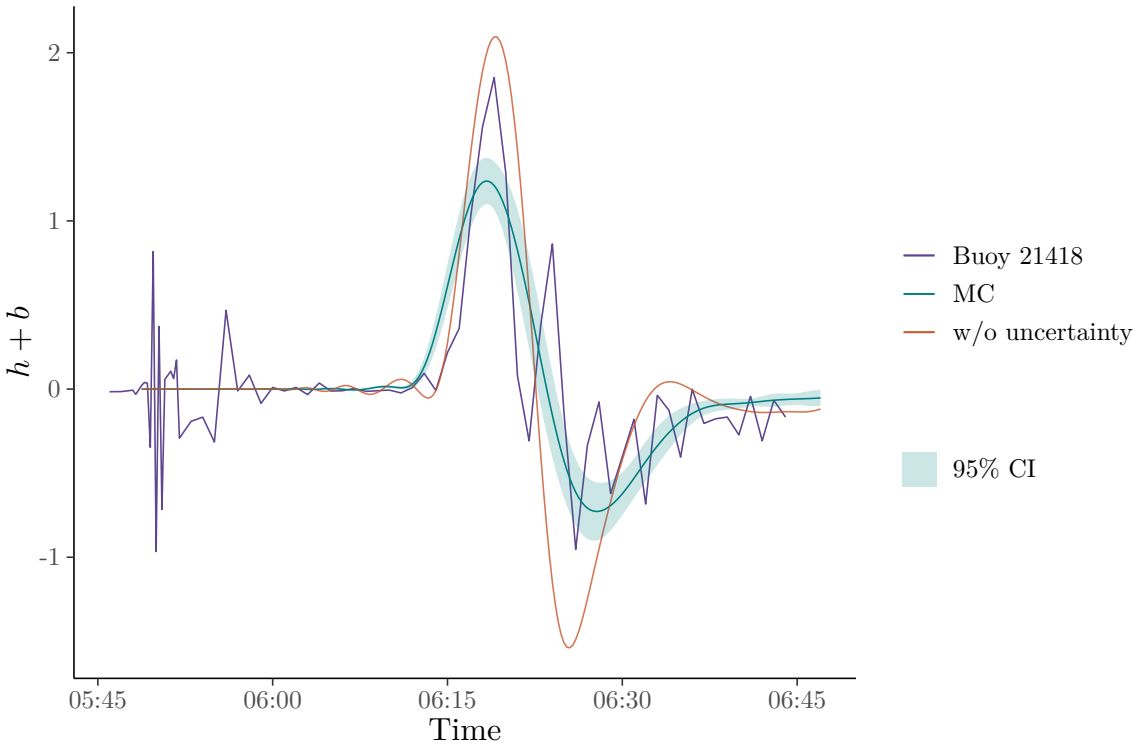


Figure 7.5: "[...] a section of the Reykjanes Ridge to the southeast of Iceland, note the apparent 'sharpness' of detail in the region running NE-SW, (which shows sub-sampled multibeam bathymetry data) compared to the much smoother areas to the southeast. There are a few thin 'stripes' or lines over this southeastern area and they represent single-beam echo-sounding lines that were used to constrain the satellite-predicted bathymetry." [23]

7.2.2 Estimate of the Buoy data

The following simulations are computed using the ADER-DG scheme with order 3. Using the third proposed sampling method we get the following results:



Method	# samples	n	α	p
MC	100	125	400	10%

Figure 7.6: Comparison of the MC estimate using sampling method 3 to the result without (w/o) uncertainty.

With $n = 125$ we have 479 sources of uncertainty. We see that our MC result is estimating the measured data of the buoy especially at 6:30 more precisely than the result with the measured bathymetry. It is surprising how big the difference is. One explanation could be that the measured data may be in fact very imprecise, containing bigger measurement errors or larger interpolated areas that are very incorrect.

7.2.3 Conclusion

The big difference of the two estimates in Figure 7.6 suggests that the bathymatry data may contain many data points who are imprecise. It is advisable to detect these in a preprocessing step - already mentioned in *"Further modifications"* - and treat them individually. It became obvious that the right choice of the sampling method can both affect the necessary sample size and such the computational effort. It use further proposed to use in future work a wavelet compression of ground to further reduce the parameters of the uncertainty sampling.

Acknowledgements

The author gratefully acknowledges the support by the Leibniz Rechenzentrum (LRZ) in Munich, Germany, for awarding access to the Linux Cluster for the extensive simulations.

Bibliography

[1] A.J.C. Barré de Saint-Venant. "Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et a l'introduction de marées dans leurs lits". In: *Comptes Rendus de l'Académie des Sciences* (1871).

[2] Honeywell ELAC. *An Introduction to Echo Sounding*. Kiel: Honeywell ELAC Nautik GmbH, 1982.

[3] G.E.P. Box and N.R. Draper. *Empirical model-building and response surfaces*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1987. ISBN: 9780471810339.

[4] D.J. Acheson and F.D.J. Acheson. *Elementary Fluid Dynamics*. Comparative Pathobiology - Studies in the Postmodern Theory of Education. Clarendon Press, 1990. ISBN: 9780198596790.

[5] Euiquio C. Young. *Vector and Tensor Analysis*. Marcel Dekker, 1992. ISBN: 0824787897.

[6] Hrvoje Jasak. "Error Analysis and Estimation for the Finite Volume Method With Applications to Fluid Flows". In: *Direct* (1996).

[7] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in applied mathematics. Springer, 2000. ISBN: 9780387989594.

[8] Olaf Boebel et al. "Risk assessment of ATLAS HYDROSWEEP DS-2 hydrographic deep sea multi-beam sweeping survey echo sounder". In: (Oct. 2004).

[9] A. Beyer. "Seafloor analysis based on multibeam bathymetry and backscatter data = Meeresbodenanalyse auf der Basis von Bathymetrie und akustischer Rückstreuung". In: *Berichte zur Polar- und Meeresforschung (Reports on Polar and Marine Research)* 540 (2006).

[10] M.L. Rizzo. *Statistical Computing with R*. Chapman & Hall/CRC The R Series. Taylor & Francis, 2007. ISBN: 9781584885450.

[11] Clint Dawson and Christopher M. Mirabito. "The Shallow Water Equations". In: (2008).

[12] Michael Dumbser et al. "A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes". In: *Journal of Computational Physics* 227.18 (Sept. 2008), pp. 8209–8253. DOI: 10.1016/j.jcp.2008.05.025. URL: https://doi.org/10.1016/j.jcp.2008.05.025.

[13] Shintaro Bunya et al. "A wetting and drying treatment for the Runge–Kutta discontinuous Galerkin solution to the shallow water equations". In: *Computer Methods in Applied Mechanics and Engineering* 198.17-20 (Apr. 2009), pp. 1548–1562. DOI: 10.1016/j.cma.2009.01.008. URL: https://doi.org/10.1016/j.cma.2009.01.008.

[14] R. Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010. ISBN: 9781139491136.

[15] Lawrence C. Evans. *Partial differential equations*. Providence, R.I.: American Mathematical Society, 2010. ISBN: 9780821849743.

[16] Nobuo Mimura et al. "Damage from the Great East Japan Earthquake and Tsunami - A quick report". In: *Mitigation and Adaptation Strategies for Global Change* (Oct. 2011). ISSN: 1573-1596. DOI: 10.1007/s11027-011-9297-7. URL: https://doi.org/10.1007/s11027-011-9297-7.

[17] S. Mishra, C. Schwab, and J. Šukys. "Multilevel Monte Carlo Finite Volume Methods for Shallow Water Equations with Uncertain Topography in Multi-dimensions". In: *SIAM Journal on Scientific Computing* 34.6 (2012), B761–B784. DOI: 10.1137/110857295.

[18] Ralph C. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2013. ISBN: 9781611973211.

[19] A. L. Teckentrup et al. "Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients". In: *Numerische Mathematik* 125.3 (Mar. 2013), pp. 569–600. DOI: 10.1007/s00211-013-0546-4. URL: https://doi.org/10.1007/s00211-013-0546-4.

[20] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*. 1st. Springer Publishing Company, Incorporated, 2015. ISBN: 9783319168739.

[21] Francesco Fambri, Michael Dumbser, and Olindo Zanotti. "Space–time adaptive ADER-DG schemes for dissipative flows: Compressible Navier–Stokes and resistive MHD equations". In: *Computer Physics Communications* 220 (Nov. 2017), pp. 297–318. DOI: 10.1016/j.cpc.2017.08.001. URL: https://doi.org/10.1016/j.cpc.2017.08.001.

[22] British Oceanographic Data Centre (BODC). *GEBCO - The General Bathymetric Chart of Oceans*. 2018. URL: https://www.gebco.net/ (visited on 09/07/2019).

[23] British Oceanographic Data Centre (BODC). *How do we know what areas show measured depths of the seabed and what areas are 'predicted'?* 2018. URL: https://www.gebco.net/about_us/faq/#about_ocean_depths (visited on 09/25/2019).

[24] Fabio Nobile. "Multi-level Monte Carlo methods in Uncertainty Quantification". In: *GAMM AGUQ workshop* (2018).

[25] National Oceanic and Atmospheric Administration. *NDBC - National Data Buoy Center*. 2018. URL: https://www.ndbc.noaa.gov/ (visited on 09/09/2019).

[26] Anne Reinarz et al. "ExaHyPE: An Engine for Parallel Dynamically Adaptive Simulations of Wave Problems". In: *CoRR* abs/1905.07987 (2019). arXiv: 1905.07987. URL: http://arxiv.org/abs/1905.07987.