Chapter 3
An application: Optimising the layout of tidal turbine arrays

3.1 Introduction

The motivation for this application comes from the desire to generate renewable energy from the tides in the Earth's oceans. Tides are a consequence of the gravitational attractions experienced within the Earth-Moon-Sun system. As this system evolves in time the cumulative gravitational forces vary and this has the effect of driving ocean (or tidal) currents which periodically and locally increase and decrease the depth of the oceans at global scales. As these tidal motions move from the deep ocean into the shallower continental shelf regions, conservation of volume accelerates the currents. They may then be even further accelerated as they interact with coastal features such as headlands and as they are constrained through narrow straits between islands. We thus have a situation where significant amounts of kinetic energy is contained within the coastal ocean relatively close to locations on shore where electricity is required. This has naturally led to significant interest in exploiting this resource through the renewable generation of electricity from tidal currents. Tides have the significant advantage over other renewable energy sources, such as solar, wind or wave, in that they are periodic and predictable, and hence do not suffer from supply reliability drawbacks. This tidal energy can be captured through tidal turbines in the case of the kinetic energy of currents discussed above, as well as barrages or lagoons in the case of the potential energy contained within tidal highs and lows. Here we consider the former case where potentially hundreds of tidal turbines may be deployed in arrays (Fig. 3.1), in much the same way that arrays of wind turbines operate.

T. Schwedes et al., *Mesh Dependence in PDE-Constrained Optimisation*, Mathematics of Planet Earth,
DOI 10.1007/978-3-319-59483-5_3

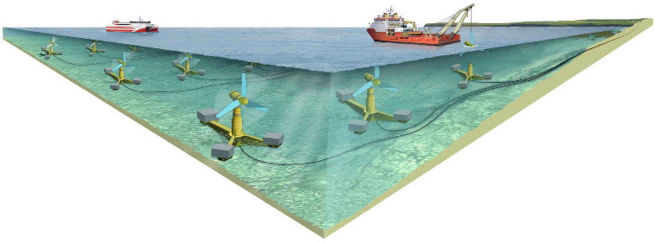


Fig. 3.1: Illustration of a tidal stream generator array. Image credit: MeyGen Limited.

Once tidal turbines are installed within a tidal current they will alter the flow across a range of scales. For example, a turbulent wake will be formed in the immediate lee of the turbine where the current speed is reduced, with locally accelerated flow above, below and to either side of the turbine and its wake (the bypass flow region). The power generated by a turbine is approximately dependent on the cube of the current speed it experiences. Hence, whether a down-stream turbine is installed within a wake zone or within the locally accelerated bypass flow will have a significant effect on the power yield of that turbine. Furthermore, large arrays can not only affect the local currents, but also the tidal resource at a regional scale. This regional effect limits the overall energy that can be extracted from a tidal array.

The optimal design of a tidal turbine array, in terms of the array location, the total number of turbines making up the array, and their individual locations (or micro-siting), is thus important in terms of array power yield and hence economic viability. The turbine layout optimisation problem is however very challenging: it has a large number of design parameters and, as discussed above, is coupled to the nonlinear hydrodynamics across a range of scales. Different approaches have been proposed to solve this problem, see for example [6, 9, 15, 17, 18], and for an overview see [20].

The goal of this chapter is to formulate the array layout problem as a PDE-constrained optimisation problem and solve it with the techniques presented earlier in this book. The hydrodynamic model will be based on the nonlinear shallow water equations, discretised with the finite element method on an unstructured mesh. One option to incorporate tidal turbines into the model is through locally enhanced seabed friction values at each turbine location. However, this approach requires relatively high mesh resolution and hence computational cost, because the size of a turbine is typically small with respect to the region considered for the (tidal/resource-scale) simulation. To mitigate this cost, here we represent the entire tidal array configuration via a turbine density function [5, 11, 19]. This density function is then used to locally increase the seabed friction coefficient to account for the impact of the local density of turbines on the flow. This “continuous turbine approach” allows for the optimisation of large arrays, or even multiple arrays close enough to interact with one another, within a regional context with relatively coarse meshes. An optimised turbine density function can then be used to infer the individual locations of turbines, and/or used as an initial guess at the array layout for a more costly discrete micro-siting optimisation [11].

3.2 Problem formulation

We treat the problem of finding the optimal configuration of a continuous turbine density function (or equivalently an enhanced seabed friction field) as a PDE-constraint optimisation problem:

$$\begin{aligned} &\max_{u,\eta,d} J(u,\eta,d), \\ \text{subject to } &F(u,\eta,d)=0, \\ &0 \leq d \leq d_u, \end{aligned} \tag{3.1}$$

where u is the horizontal (depth-averaged) velocity vector, η is the free surface displacement from a state of rest, d is the turbine density function, $J(u,\eta,d) \in \mathbb{R}$ is the functional of interest, $F(u,\eta,d)=0$ represents the nonlinear shallow water equations, and d_u is a upper bound for the turbine density function. The choice of the function space D containing d is essential for this work and is discussed in Sect. 3.2.3.

Assuming that through the PDE constraint equation $F(u,\eta,d)=0$, any given $d \in D$ is uniquely mapped to a solution u,η through the shallow water equations, we may regard u,η as implicit functions of the design parameters d, that is $u \equiv u(d)$ and $\eta \equiv \eta(d)$. By substituting this into the functional of interest J, one obtains the reduced optimisation problem

$$\begin{aligned} &\max_{d \in D} J(u(d),\eta(d),d), \\ &\text{subject to } 0 \leq d \leq d_u. \end{aligned} \tag{3.2}$$

3.2.1 Design parameters and box constraints

The design parameter d is a continuous turbine density function on the domain of interest Ω. More specifically, $d|_A \equiv 0$ for some area $A \subset \Omega$ implies that no turbines are deployed inside A, whereas $d|_A \gg 0$ implies that turbines are densely deployed within A. The turbine density field can be regarded as an approximation for the actual placement configuration of turbines within the array. The integrated turbine density function additionally provides an approximation for the number of turbines N of the configuration:

$$N = \int_\Omega d(x)\,\mathrm{d}x. \tag{3.3}$$

The point-wise bound constraint in (3.2) restricts d to feasible values. Firstly, the turbine density may only take non-negative values. Secondly, non-zero values are only expected in areas that are suitable for turbine placement. For instance, if an area A is too steep, too shallow, or environmentally protected, no turbines should be placed there, i.e. $d_u|_A \equiv 0$. All other locations combine to form valid array area(s), denoted Ω_{array}. In addition, the turbine density is assumed to be bounded above by a constant $\bar{d} > 0$. This bound takes into account that the number of turbines per unit area is limited, for example by a minimum feasible distance between any two turbines, or by hydrodynamic, installation and maintenance constraints. We conclude that

$$d_u(x) = \begin{cases} \bar{d} & \text{if } x \in \Omega_{\text{array}} \\ 0 & \text{if } x \in \Omega \setminus \Omega_{\text{array}}. \end{cases} \tag{3.4}$$

3.2.2 The PDE constraint

The optimisation problem (3.1) is subject to physical laws which are incorporated via the PDE $F(u,\eta,d) = 0$. Here, the PDE represents the nonlinear shallow water equations in steady state, that is

$$\begin{aligned} u \cdot \nabla u - \nu \nabla^2 u + g \nabla \eta + \frac{c_b + c_t(d)}{H} \|u\| u &= 0, \\ \nabla \cdot (Hu) &= 0, \end{aligned} \tag{3.5}$$

where $u : \Omega \to \mathbb{R}^2$ and $\eta : \Omega \to \mathbb{R}$ denote the depth-averaged horizontal velocity vector and free-surface displacement respectively, $H : \Omega \to \mathbb{R}$ is the total water depth (that is, $H = \eta + h$, where h is the water depth at rest, see Fig. 3.2), $g \approx 9.81\,\mathrm{m\,s^{-2}}$ is the gravitational acceleration, $\nu > 0$ is the kinematic viscosity coefficient, $c_b : \Omega \to [0,\infty)$ denotes the natural or background seabed friction value which may be spatially varying if this knowledge/data is available, and $c_t(d) : \Omega \to [0,\infty)$ is the additional 'friction' exerted on the flow by the presence of the turbines as encoded by the turbine density field d. The shallow water equations will be derived in detail in Sect. 3.3.

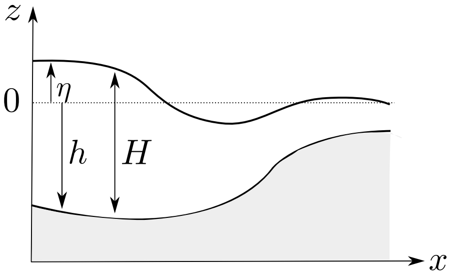


Fig. 3.2: Variables for the shallow water equations

3.2.3 The class of turbine density functions

When considering the optimisation problem (3.1), the question arises as to what class of functions are reasonable to consider for the turbine density d. At first glance, we expect the total number of turbines given by (3.3) to be finite, that is $d \in L^1(\Omega)$. If we were to assume that $d \in L^2(\Omega)$, a simple calculation using Cauchy-Schwarz and the finiteness of Ω convinces us that this implies that $d \in L^1(\Omega)$:

$$\|d\|_{L^1(\Omega)} = \int_\Omega d(x)\,\mathrm{d}x = \langle d, 1\rangle_{L^2(\Omega)} \leq \|d\|_{L^2(\Omega)} \cdot |\Omega|^{1/2} < \infty,$$

where $(\cdot,\cdot)_{L^2(\Omega)}$ denotes the inner product in $L^2(\Omega)$, and $\|\cdot\|_{L^2(\Omega)}$ and $\|\cdot\|_{L^1(\Omega)}$ denote the norms in $L^2(\Omega)$ and $L^1(\Omega)$, respectively. $L^2(\Omega)$ has the advantage over $L^1(\Omega)$ of being a Hilbert space, that is, it possesses an inner product and strong functional analytical results are valid such as the Riesz representation theorem. Therefore, functionals on $L^2(\Omega)$ have a unique representative in $L^2(\Omega)$ for their Fréchet derivative. This becomes particularly relevant when solving (3.1) using gradient-based methods which naturally require that representative. The question arises why not considering $d \in W^{k,2}(\Omega)$ for $k > 1$, such as $H^1(\Omega)$ for example. Such functions have weak derivatives that are $L^2(\Omega)$-integrable. That means they satisfy smoothness assumptions that we do not necessarily expect from, nor wish to impose on, the turbine density function. In fact, the continuous turbine approach is a practical means to approximate the optimal distribution of real, i.e. discrete individual, turbines. In reality, we naturally expect to observe discontinuities in the turbine distribution as at every point of the domain there is either a turbine in place or not, as well as the fact that the domain is made up of discrete regions where turbines are permitted. Therefore, we shall use a space that reflects this non-smoothness. In this context and for the reasons mentioned above, we believe that the choice of $L^2(\Omega)$ as the class of turbine density functions is sensible.

3.2.4 Relationship between turbine density and seabed friction

Referring to [11], the force exerted on the flow due to the enhanced friction term $c_t(d)$ produced by the turbine field in the shallow water equations (3.5) is given by

$$F_{\text{array}} = \int_{\Omega_{\text{array}}} \rho c_t(d(x)) \|u(x)\| u(x) \, \mathrm{d}x. \tag{3.6}$$

Using a common parameterisation for the drag force exerted by an individual turbine in 3D, the drag force exerted by an array of N turbines can be approximated as

$$F_{\text{array}} = \sum_{i=1}^{N} \frac{1}{2} \rho C_t A_t \|u_{\infty,i}\| u_{\infty,i}, \tag{3.7}$$

where C_t denotes the thrust coefficient corresponding to an individual turbine and A_t its cross-sectional area. Here we assume that all turbines are identical and are always aligned with the flow and so have common values for C_t and A_t. Further, $u_{\infty,i}$ denotes the free-stream or undisturbed velocity, i.e. the velocity that would be encountered at the ith turbine's location without the presence of the turbine. Here we simply take this to be the velocity at the turbine location from the model and note that this is a reasonable approximation as the local velocity of the model gets closer to the free-stream velocity the more the turbine density field is 'spread' through use of a relatively coarse computational mesh. See [1, 11, 12] for further discussions on this point, including corrections that are possible as mesh resolution is increased.

Re-formulating the drag force from its discrete version in (3.7) into a continuous one by replacing the sum by an integral containing the turbine density field yields

$$F_{\text{array}} = \int_{\Omega_{\text{array}}} \frac{1}{2} \rho C_t A_t d(x) \|u(x)\| u(x) \, \mathrm{d}x. \tag{3.8}$$

Equating with (3.6) leads to an expression for the friction function c_t in terms of the turbine density function d:

$$c_t(d) = \frac{1}{2} C_t A_t d. \tag{3.9}$$

3.2.5 The functional of interest

The functional of interest J in the optimisation problem (3.1) refers to the quantity that is to be maximised. In what follows we define two reasonable expressions for the functional of interest.

3.2.5.1 Power functional

One natural choice for the functional of interest is the total power which the array (or here the turbine density field) extracts from the flow. From (3.6) and the formula for mechanical power, $P = F_{\text{array}} \cdot u$, this functional can be expressed as:

$$J_{\text{power}}(u,d) = \int_{\Omega_{\text{farm}}} \rho c_t(d(x)) \|u(x)\|^3 \,\mathrm{d}x. \tag{3.10}$$

Note that the actual electricity produced by the turbine array is only a fraction of 3.10, as the functional as written also includes energy losses due to wake mixing effects behind turbines. Furthermore, one needs to account for the power losses due to power transformation to electrical energy, losses due to transporting the energy from turbines to the grid, and by the presence of turbine support structures which contribute part of the turbine induced friction.

3.2.5.2 Profit functional

In the context of evaluating possible turbine arrays from an economic perspective, a reasonable choice for the functional of interest is the profit of an array represented by its density function, generated over its entire lifetime. A simple choice for a profit functional is

$$\begin{aligned} J_{\text{profit}}(u,d) &= \text{Revenue}(u,d) - \text{Cost}(d) \\ &= IkTJ_{\text{power}}(u,d) - C\int_{\Omega} d(x)\,\mathrm{d}x, \end{aligned} \tag{3.11}$$

where $J_{\text{power}}(u,d)$ as given in (3.10) is the extracted power, T a turbine's average lifetime, $k \in (0,1)$ is a turbine efficiency coefficient, I is an income factor stating the financial value of the power generated, and C is the cost of installing and maintaining one turbine. Recall from (3.3) that the integral of d gives the total number of installed turbines. See [11] for further details.

3.3 Shallow water equations

In this section, for completeness, we review the derivation of the shallow water equations which are the partial differential equations that describe the flow of the system we consider. The shallow water equations are a two-dimensional (depth-averaged) approximation to the three-dimensional Navier-Stokes equations and are valid for scenarios in which the horizontal length scale is much greater than the vertical length scale.

3.3.1 Physical principles

The shallow water equations are based on depth-integrating or averaging the Navier-Stokes equations and therefore are a consequence of the principles of conservation of mass and momentum. Thus, we begin by briefly deriving these principles.

3.3.1.1 Conservation of mass

One essential law of physics is the conservation of mass. It states that the rate of change of the total mass within an arbitrary control volume V equals the net mass flux across its boundary ∂V. Mathematically, this can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho \,\mathrm{d}x = -\int_{\partial V} \rho u \cdot n \,\mathrm{d}A, \tag{3.12}$$

where ρ is the density of the fluid, u is the fluid velocity and n is the outward pointing unit normal vector to ∂V. Let us assume that V does not vary in time and that ρ is smooth. Applying Gauss's theorem to the right hand side allows us to write

$$\int_V \left(\frac{\partial \rho}{\partial t} - \nabla \cdot (\rho u)\right) \mathrm{d}x = 0. \tag{3.13}$$

As V was an arbitrary control volume, we can conclude the pointwise relationship

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0. \tag{3.14}$$

Note on the other hand that Eq. (3.14) trivially implies (3.12), and hence is equivalent to mass conservation. The PDE (3.14) is referred to as the continuity equation.

3.3.1.2 Conservation of momentum

Following Newton's second law, the rate of change of the total momentum in a control volume V equals the net flux of momentum across ∂V plus any internal forces exerted within V and any external forces exerted on ∂V. Mathematically this can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho u \,\mathrm{d}x = -\int_{\partial V} (\rho u) u \cdot n \,\mathrm{d}A + \int_V \rho f_{\text{int}} \,\mathrm{d}x + \int_{\partial V} S n \,\mathrm{d}A,$$

where f_{int} denotes the internal force density and S denotes the Cauchy stress tensor. Examples of internal forces are those due to gravity, Coriolis (which is actually a fictitious force that is required when we make the usual ocean modelling assumption of considering our domain in a non-inertial, i.e. here a rotating, reference frame), electric or magnetic forces. Assuming that the momentum ρu is smooth and applying

Gauss's theorem results in

$$\int_V \left(\frac{\partial}{\partial t}(\rho u) + \nabla \cdot (\rho u u^T) - \rho f_{\text{int}} - \nabla \cdot S\right) \mathrm{d}x = 0. \tag{3.15}$$

Since again V was chosen arbitrarily, we deduce that

$$\frac{\partial}{\partial t}(\rho u) + \nabla \cdot (\rho u u^T) - \rho f_{\text{int}} - \nabla \cdot S = 0. \tag{3.16}$$

The PDE (3.16) encodes the concept of conservation of momentum and is referred to as the momentum equation.

3.3.2 Navier-Stokes equations

In the following, we derive the Navier-Stokes equations for an incompressible Newtonian fluid using the principles of conservation of mass and momentum. For a Newtonian fluid, we have the stress tensor

$$S = -pI_3 + \tau, \tag{3.17}$$

where p denotes the pressure, I_3 the identity matrix in $\mathbb{R}^3$ and τ is the deviatoric stress tensor. Stokes' stress constitutive equation used for incompressible fluid states that

$$\tau = \mu\left(\nabla u + (\nabla u)^T\right), \tag{3.18}$$

where μ is the, assumed constant, dynamic viscosity coefficient of the fluid ($\mu = \rho\nu$ where ν is termed the kinematic viscosity). Thus, we find for the divergence of the Cauchy stress tensor:

$$\nabla \cdot S = -\nabla p + \mu \nabla^2 u. \tag{3.19}$$

Neglecting all internal forces besides gravity, we have $f_{\text{int}} = g$, where g is the gravitational acceleration vector. In addition to incompressibility, which states that ρ is independent of p, we assume that ρ is a constant in time and throughout the fluid, e.g. ignoring thermal effects. Applying (3.19), we therefore may infer from the mass continuity equation (3.14) and the momentum equation (3.16) the following system of PDEs:

$$\begin{aligned} \nabla \cdot u &= 0, \\ \frac{\partial u}{\partial t} + \nabla \cdot (u u^T) - \nu \nabla^2 u + \frac{1}{\rho}\nabla p - g &= 0. \end{aligned} \tag{3.20}$$

The equation pair (3.20) are known as the isothermal Navier-Stokes equations for an incompressible Newtonian fluid. If f contains another internal forcing vector, i.e. $f_{\text{int}} = g + f$, we get an additional f term on the right hand side of (3.20).

3.3.3 Shallow water equations

In what follows we sketch the derivation of the shallow water equations via the depth-integration, or averaging, of the Navier-Stokes equations (3.20). For a more complete derivation and mathematical discussions see [21], [3]. With $u = (u_1, u_2, u_3)^T$, the momentum equation in (3.20) may be written in non-vector form as

$$\begin{aligned} \frac{\partial u_1}{\partial t} + \sum_{i=1}^{3} u_i \frac{\partial u_1}{\partial x_i} - \nu \sum_{j=1}^{3} \frac{\partial^2 u_1}{\partial x_j^2} + \frac{1}{\rho} \frac{\partial p}{\partial x_1} &= 0, \\ \frac{\partial u_2}{\partial t} + \sum_{i=1}^{3} u_i \frac{\partial u_2}{\partial x_i} - \nu \sum_{j=1}^{3} \frac{\partial^2 u_2}{\partial x_j^2} + \frac{1}{\rho} \frac{\partial p}{\partial x_2} &= 0, \\ \frac{\partial u_3}{\partial t} + \sum_{i=1}^{3} u_i \frac{\partial u_3}{\partial x_i} - \nu \sum_{j=1}^{3} \frac{\partial^2 u_3}{\partial x_j^2} + \frac{1}{\rho} \frac{\partial p}{\partial x_3} + g &= 0, \end{aligned} \tag{3.21}$$

where g now represents the magnitude of the gravitational acceleration vector. Further, the continuity equation in (3.20) may be written as

$$\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3} = 0. \tag{3.22}$$

The shallow water equations are based on the essential assumption that the horizontal length scale of the dynamics, e.g. its wavelength, is much larger than the vertical length scale, e.g. the water depth at rest. Based upon this and via a scaling analysis of the terms in the vertical momentum equation we may conclude that vertical acceleration as well as viscous effects are small, and that this equation reduces to the so-called hydrostatic balance relation

$$\frac{1}{\rho} \frac{\partial p}{\partial x_3} = -g, \tag{3.23}$$

This states that pressure in our system is well approximated by its hydrostatic value, i.e. the weight of water above the location in question. Integrating (3.23) in the vertical, i.e. with respect to x_3, yields

$$p = \rho g(\eta - x_3) + p_a, \tag{3.24}$$

where η is the free-surface displacement from a state of rest (Fig. 3.2), and $p_a = p(x_1, x_2, \eta)$ denotes the atmospheric pressure. For the sake of simplicity we further assume that $\nabla p_a = 0$. For applications where this term becomes relevant, e.g. storm surge modelling, it can easily be incorporated as an additional source term in the final equations. Differentiating (3.24) in the horizontal yields

$$\frac{\partial p}{\partial x_1} = \rho g \frac{\partial \eta}{\partial x_1}, \qquad \frac{\partial p}{\partial x_2} = \rho g \frac{\partial \eta}{\partial x_2}. \tag{3.25}$$

Noting that this pressure gradient forcing is independent of depth, x_3, allows us to conclude from the horizontal momentum equations that if the horizontal velocities are initially independent of depth then they will remain so. While this is the case we assume here, and is entirely consistent with the simulation of tidal dynamics at large scales, it should be noted that this assumption would break down at small scales including when the flow interacts with individual turbines. The use of fully-3D equation sets and models in the design of turbine arrays is the subject of ongoing and future work, e.g. see [1].

We can thus interpret the horizontal velocities as depth-averaged values, or depth-integrals following multiplication by the layer depth, and within the horizontal momentum equations ignore all derivatives with respect to x_3. More rigorous derivations, including arriving at a similar result from the vertical integration of the momentum equations, may be found in [21], [3]. We are thus left with new shallow water momentum equations

$$\begin{aligned} \frac{\partial u_1}{\partial t} + \sum_{i=1}^{2} u_i \frac{\partial u_1}{\partial x_i} - \nu \sum_{j=1}^{2} \frac{\partial^2 u_1}{\partial x_j^2} + g \frac{\partial \eta}{\partial x_1} &= 0, \\ \frac{\partial u_2}{\partial t} + \sum_{i=1}^{2} u_i \frac{\partial u_2}{\partial x_i} - \nu \sum_{j=1}^{2} \frac{\partial^2 u_2}{\partial x_j^2} + g \frac{\partial \eta}{\partial x_2} &= 0. \end{aligned} \tag{3.26}$$

We can now also remove the presence of the vertical derivative of u_3 in the continuity equation (3.22) by integrating over the total water depth:

$$\begin{aligned} \int_{-h}^{\eta} \nabla \cdot u \, \mathrm{d}x_3 &= \int_{-h}^{\eta} \left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} \right) \mathrm{d}x_3 + u_3|_{x_3=\eta} - u_3|_{x_3=-h} \\ &= \frac{\partial}{\partial x_1} \int_{-h}^{\eta} u_1 \, \mathrm{d}x_3 + \frac{\partial}{\partial x_2} \int_{-h}^{\eta} u_2 \, \mathrm{d}x_3 - \left(u_1|_{x_3=\eta} \frac{\partial \eta}{\partial x_1} + u_1|_{x_3=-h} \frac{\partial h}{\partial x_1} \right) \\ &\quad - \left(u_2|_{x_3=\eta} \frac{\partial \eta}{\partial x_2} + u_2|_{x_3=-h} \frac{\partial h}{\partial x_2} \right) + u_3|_{x_3=\eta} - u_3|_{x_3=-h}, \end{aligned} \tag{3.27}$$

where h denotes the bathymetry, i.e. the water depth at rest. The following so-called kinematic boundary conditions state that the normal components of the flow at the bottom of the domain $(x_3 = -h)$ and at the free surface $(x_3 = \eta)$ are equal to zero:

$$\begin{aligned} \left(u_1\frac{\partial h}{\partial x_1}+u_2\frac{\partial h}{\partial x_2}+u_3\right)\bigg|_{x_3=-h} &= 0, \\ \left(\frac{\partial \eta}{\partial t}+u_1\frac{\partial \eta}{\partial x_1}+u_2\frac{\partial \eta}{\partial x_2}-u_3\right)\bigg|_{x_3=\eta} &= 0. \end{aligned} \tag{3.28}$$

This states that particles cannot traverse these surfaces. Since the horizontal velocities are assumed to be independent of x_3 we have

$$\frac{\partial}{\partial x_i}\int_{-h}^{\eta} u_i \,\mathrm{d}x_3 = \frac{\partial}{\partial x_i}(Hu_i) \quad \forall\, i=1,2, \tag{3.29}$$

where $H=h+\eta$ denotes the total water depth. Applying the boundary conditions (3.28) to (3.27) then results in

$$\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x_1}(Hu_1)+\frac{\partial}{\partial x_2}(Hu_2)=0. \tag{3.30}$$

Equation (3.30) is termed the depth-averaged continuity equation. In 2D vector notation (3.30) and (3.26) may be written as

$$\frac{\partial \eta}{\partial t}+\nabla\cdot(Hu)=0, \tag{3.31}$$

$$\frac{\partial u}{\partial t}+u\cdot\nabla u-\nu\nabla^2 u+g\nabla\eta=0, \tag{3.32}$$

which we term the nonlinear shallow water equations. It is straightforward to see that in the case where we have an additional term F in the momentum equation (3.16) representing a force or source vector, it has to be added in the momentum equation (3.32). In this work, we have an additional bottom friction term given by the quadratic drag

$$F=-\frac{c_b+c_t(d)}{H}\|u\|u, \tag{3.33}$$

where c_b is the natural bottom friction coefficient and $c_t(d)$ the coefficient corresponding to the friction field induced by the turbine density d. Thus, the shallow water momentum equation we consider in this work takes the form

$$\frac{\partial u}{\partial t}+u\cdot\nabla u-\nu\nabla^2 u+g\nabla\eta+\frac{c_b+c_t(d)}{H}\|u\|u=0. \tag{3.34}$$

Finally, the steady form of the system (3.31) and (3.34) follows from neglecting the time derivative terms and is often considered in situations where the velocity can be assumed to be time independent, or the fully time-dependent simulation is numerically too costly. With this simplification, we arrive at the PDE system stated in (3.5).

3.4 Aspects of numerical solution

This section discusses details of the numerical solution of problem (3.2). First, the shallow water equations are stated in their weak form and solved by applying a finite element discretisation (Sect. 3.4.1). From the resulting discrete solutions, the discrete adjoint equations are derived (Sect. 3.4.2). Their solution provides the discretised derivative of the functional of interest with respect to the turbine density, which is used in several gradient-based optimisation methods. Section 3.4.3 provides an illustration of what happens during one iteration of the optimisation loop performed in order to solve (3.2), describes the Python libraries used and performs an implementation verification via a Taylor remainder test.

3.4.1 Solving the shallow water equations

In what follows, we briefly explain the derivation of the weak formulation of the shallow water equations (3.5) in steady state and discuss how their weak form is solved using the finite element method. One derives the weak formulation of the shallow water equations by multiplying them by test functions $\Psi \in V$ and $\Phi \in Q$, respectively, integrating over the domain Ω and applying integration by parts to the viscosity and divergence terms. Using the notation $\langle\cdot,\cdot\rangle_A := \langle\cdot,\cdot\rangle_{L^2(A)}$ for $A \subset \overline{\Omega}$, this yields

$$\begin{aligned} \langle u\cdot\nabla u, \Psi\rangle_\Omega + \nu\,\langle \nabla u, \nabla\Psi\rangle_\Omega + g\,\langle\nabla\eta,\Psi\rangle_\Omega + \left\langle \frac{c_b + c_t(d)}{H}\|u\|u, \Psi\right\rangle_\Omega &= 0, \\ -\langle Hu, \nabla\Phi\rangle_\Omega + \langle Hu\cdot n, \Phi\rangle_{\partial\Omega_{\mathrm{In}}} &= 0, \end{aligned} \tag{3.35}$$

where $\partial\Omega_{\mathrm{In}}$ denotes the inflow boundary, on which strong Dirichlet conditions are imposed for the normal component of the velocity vector u. To ensure that all terms in (3.35) are well-defined, a natural choice for the test function spaces is $V = [H^1(\Omega)]^2$ and $Q = H^1(\Omega)$. The domain boundary can be decomposed into $\partial\Omega = \partial\Omega_{\mathrm{In}} \cup \partial\Omega_{\mathrm{Out}} \cup \partial\Omega_{\mathrm{Coast}}$, where $\partial\Omega_{\mathrm{In}}$ denotes the inflow boundary, $\partial\Omega_{\mathrm{Out}}$ the outflow boundary and $\partial\Omega_{\mathrm{Coast}}$ the coastal boundaries. For the weak formulation (3.35), we have used the following integration by parts relation:

$$\langle\nabla\cdot(Hu), \Phi\rangle_\Omega = -\langle Hu, \nabla\Phi\rangle_\Omega + \langle Hu\cdot n, \Phi\rangle_{\partial\Omega} \tag{3.36}$$

$$= -\langle Hu, \nabla\Phi\rangle_\Omega + \langle Hu\cdot n, \Phi\rangle_{\partial\Omega_{\mathrm{In}}}, \tag{3.37}$$

where we applied an either weakly or strongly imposed homogeneous Dirichlet boundary conditions on $\partial\Omega_{\mathrm{Coast}}$. Further, a strong Dirichlet boundary condition on η was imposed on the outflow boundary $\partial\Omega_{\mathrm{Out}}$, hence the test function Φ vanishes on the boundary and thus the integral over $\partial\Omega_{\mathrm{Out}}$ vanishes as well. Integration by parts has also been applied to the viscous term:

$$-\langle \Delta u, \Psi \rangle_{\Omega} = \langle \nabla u, \nabla \Psi \rangle_{\Omega} - \langle \nabla u \cdot n, \Psi \rangle_{\partial\Omega} \tag{3.38}$$

$$= \langle \nabla u, \nabla \Psi \rangle_{\Omega} . \tag{3.39}$$

The inflow boundary term disappears since strong Dirichlet boundary conditions are imposed on $\partial\Omega_{\text{In}}$, and therefore the test function Ψ vanishes. The outflow boundary term equals zero due to the velocity-pressure duality which, because of the strong Dirichlet boundary condition on η, imposes a zero Neumann boundary condition on u.

Problem (3.35) is solved using a finite element approach. The function spaces V and Q are replaced by finite-dimensional subspaces $V_h \subset V$ and $Q_h \subset Q$ using a triangulation of the domain. V_h and Q_h are chosen as the Taylor-Hood P_2-P_1 mixed finite element pair which satisfies the desirable inf-sup or Ladyzhenskaya-Babuka-Breezi (LBB) stability condition; see [2, Sect. 12.5] for details.

3.4.2 Adjoint equations

The adjoint equations can be derived with the approach presented in Sect. 1.4. The adjoint shallow water equations are

$$\begin{aligned}(\nabla \cdot u)\lambda_u - u \cdot \nabla \lambda_u - \nu \nabla^2 \lambda_u - H \nabla \lambda_\eta + \frac{c_b + c_t(d)}{H}\left(\|u\|\lambda_u + \frac{u \cdot \lambda_u}{\|u\|}u\right) &= \frac{\partial J^*}{\partial u}, \\ -g\nabla \cdot \lambda_u - \nabla \lambda_\eta \cdot u - \frac{c_b + c_t(d)}{H^2}\|u\|u &= 0,\end{aligned} \tag{3.40}$$

where λ_u is the adjoint velocity and λ_η the adjoint free-surface displacement. For a detailed derivation of (3.40), we refer to [10, Appendix C]. The functional gradient for a given turbine density field d is computed in three steps:

1. Solve the weak formulation of the shallow water equations given by (3.35) for u and η.
2. Solve (3.40) for λ_u and λ_η. Using the same finite element discretisation as in step 1 ensures that the gradient is consistent with the discrete shallow water model.
3. Evaluate the functional gradient using (1.149). The only term in the shallow water equations which explicitly depends on d is the one containing $c_t(d)$with $c_t = \frac{1}{2}C_t A_t d$. Therefore,

$$\frac{\partial F}{\partial d}(u, \eta, d; \lambda_u, \lambda_\eta) = \frac{1}{2}\frac{C_t A_t}{H}\|u\|u \cdot \lambda_u,$$

and the gradient is evaluated as

$$\frac{\mathrm{d}J}{\mathrm{d}d}(u, \eta, d) = -\frac{1}{2}\frac{C_t A_t}{H}\|u\|u \cdot \lambda_u + \frac{\partial J}{\partial d}(u, \eta, d).$$

3.4.3 Implementation

The implementation of the turbine layout optimiser is available in the open-source package OpenTidalFarm. Figure 3.3 visualises the optimisation loop and the software components that are used internally.

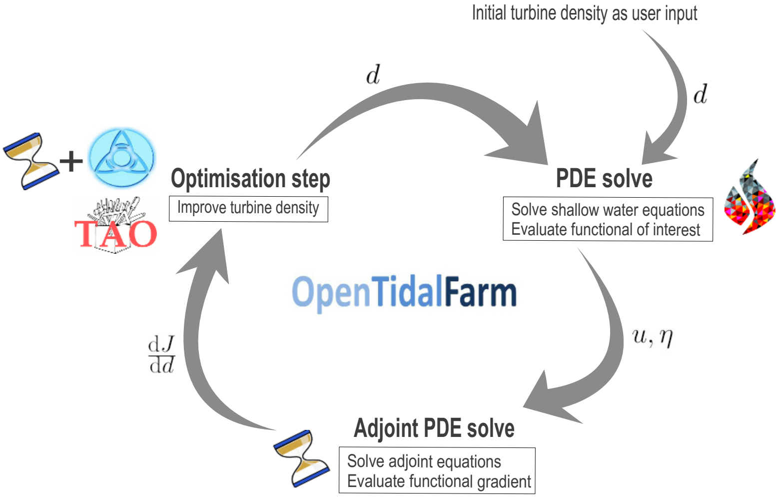


Fig. 3.3: The optimisation loop of the turbine layout designer: Initialised by a user-defined turbine density d, the FEniCS framework [13] is employed to solve the shallow water equations. In the next step, the adjoint shallow water equations and the functional gradient are derived and calculated using dolfin-adjoint [8]. In the optimisation step, we use either the interior point method from Optizelle [22] or the LMVM method provided by TAO [14]. This yields an improved turbine density, which is given to the PDE step and the procedure is iterated until a convergence criteria is met

To test the correctness of the adjoint equation and the functional gradient, a Taylor convergence test can be performed as described in Sect. 1.4.6. The Taylor convergence tests are performed here for three different choices of turbine friction functions, defined by

$$c_{t,1}(x,y) = 0.03,$$
$$c_{t,2}(x,y) = 3 \cdot 10^{-9} \cdot \left(1 + x^2 + 2y^2\right),$$
$$c_{t,3}(x,y) = 0.01 \cdot \left(2 + \sin(2\pi x) - \cos(2\pi y)\right),$$

for $(x,y) \in \Omega_{\text{array}} = [1500, 2500]^2$, and equal to zero on $\Omega \setminus \Omega_{\text{array}}$ with $\Omega = [0,4000]^2$. By applying the relation $d_i = 2c_{t,i}/(C_t A_t)$ with $C_t = 0.6$ and $A_t = 314.15m^2$, this corresponds to the turbine density functions given by

$$d_1(x,y) = 3.183 \cdot 10^{-4},$$
$$d_2(x,y) = 3.183 \cdot 10^{-11} \cdot \left(1 + x^2 + 2y^2\right),$$
$$d_3(x,y) = 1.061 \cdot 10^{-4} \cdot \left(2 + \sin(2\pi x) - \cos(2\pi y)\right),$$

for $(x,y) \in \Omega_{\text{array}}$ and equal to zero on $\Omega \setminus \Omega_{\text{array}}$, where the physical unit m^{-2} is omitted.

The example turbine densities d_1, d_2 and d_3 all satisfy the inequality condition in (3.1) with d_u given in (3.4) with $\bar{d} \approx 6.25 \cdot 10^{-4}\,\text{m}^{-2}$. The $\bar{d}$ value is derived from $\bar{d} = 1/D_{\text{min}}^2$ with a realistic minimal distance $D_{\text{min}} \approx 40\,\text{m}$ between individual turbines. Moreover, d_1, d_2 and d_3 produce outcomes for the power functional in a range that we also encounter in our more realistic simulations.

All other parameters relevant for the turbine array optimisation problem, particularly the model parameters used in the shallow water equations, can be found in table 3.4, except the viscosity which is chosen here as $\nu = 50\,\text{m}^2/\text{s}$. In the model, homogeneous Dirichlet boundary conditions for u are applied at the Northern and Southern boundaries to the domain. On the western boundary, an inflow is generated by imposing $\eta = 0.1$ while at the eastern outflow boundary $\eta = 0$ is imposed. The turbine density is represented by cubic Lagrange finite elements P_3 and the underlying mesh contains 10,202 elements.

The test results for the first and second-order Taylor remainders for d_1, d_2 and d_3 are given in tables 3.1, 3.2 and 3.3, respectively. The first-order Taylor remainders decrease roughly by half between subsequent evaluations, and the second-order Taylor remainders decrease roughly by four. The convergence order should be equal to one for first-order Taylor remainders and two for second-order Taylor remainders. The results of the Taylor remainder demonstrate the expected orders of convergence, giving confidence that the adjoint and gradient computations are correct.

3.5 Mesh dependence in tidal turbine array layout optimisation

In this section, we analyse mesh-dependent convergence in the PDE-constraint optimality problem presented in Sect. 3.2 with respect to different Riesz gradient representations for the functional of interest. Here, the functional of interest

h	$R_0(h\delta d)$	order	$R_1(h\delta d)$	order
0.01	$2.412 \cdot 10^6$		$4.020 \cdot 10^5$	
0.005	$1.302 \cdot 10^6$	**0.89**	$1.055 \cdot 10^5$	**1.93**
0.0025	$0.677 \cdot 10^6$	**0.94**	$0.270 \cdot 10^5$	**1.96**
0.00125	$0.345 \cdot 10^6$	**0.97**	$0.068 \cdot 10^5$	**1.98**
0.000625	$0.174 \cdot 10^6$	**0.99**	$0.017 \cdot 10^5$	**1.99**

Table 3.1: Taylor reminders $R_0 = |J(d_1 + h\delta d) - J(d_1)|$ and $R_1 = |J(d_1 + h\delta d) - J(d_1) - h\nabla J(d_1)\delta d|$ for the turbine layout example with functional given by (3.10). The perturbation direction δd is a P_3 function nodal values given by random samples from $[0,1]$

h	$R_0(\delta d)$	order	$R_1(\delta d)$	order
0.01	$1.620 \cdot 10^6$		$2.752 \cdot 10^5$	
0.005	$0.876 \cdot 10^6$	**0.89**	$0.720 \cdot 10^5$	**1.93**
0.0025	$0.455 \cdot 10^6$	**0.94**	$0.184 \cdot 10^5$	**1.97**
0.00125	$0.232 \cdot 10^6$	**0.97**	$0.046 \cdot 10^5$	**1.98**
0.000625	$0.117 \cdot 10^6$	**0.99**	$0.012 \cdot 10^5$	**1.99**

Table 3.2: Taylor reminders $R_0 = |J(d_2 + \delta d) - J(d_2)|$ and $R_1 = |J(d_2 + \delta d) - J(d_2) - \nabla J(d_2)\delta d|$ for the turbine layout example with functional given by (3.10). The perturbation direction δd is a P_3 function nodal values given by random samples from $[0,1]$

h	$R_0(\delta d)$	order	$R_1(\delta d)$	order
0.01	$4.192 \cdot 10^6$		$6.965 \cdot 10^5$	
0.005	$2.261 \cdot 10^6$	**0.89**	$1.834 \cdot 10^5$	**1.93**
0.0025	$1.175 \cdot 10^6$	**0.94**	$0.471 \cdot 10^5$	**1.96**
0.00125	$0.599 \cdot 10^6$	**0.97**	$0.19 \cdot 10^5$	**1.98**
0.000625	$0.302 \cdot 10^6$	**0.99**	$0.030 \cdot 10^5$	**1.99**

Table 3.3: Taylor reminders $R_0 = |J(d_3 + \delta d) - J(d_3)|$ and $R_1 = |J(d_3 + \delta d) - J(d_3) - \nabla J(d_3)\delta d|$ for the turbine layout example with functional given by (3.10). The perturbation direction δd is a P_3 function nodal values given by random samples from $[0,1]$

is chosen as the profit functional from (3.11), using a profit margin $m = 0.4$, i.e. $J_{\text{profit}} = 0.4 \cdot \text{Revenue}$, and a cost coefficient $C/(IkT) = 452.39\ kW$. The other relevant parameters for our simulations are stated in table 3.4. Further, we define the domain $\Omega = [0,2000] \times [0,1000]$ and the farm area $\Omega_{\text{array}} = [750,1250] \times [350,650]$ with an initial turbine friction equal to zero (no turbines placed within the array). In the model, weak homogeneous Dirichlet boundary conditions for u are applied on the Northern and Southern boundaries of the domain. On the western side boundary, an inflow is generated by imposing $u = (2,0)^T$ strongly, with the outflow boundary condition on the eastern boundary strongly imposed by $\eta = 0$. The turbine density is represented by discontinuous Galerkin finite elements. The underlying meshes used contain between 1258 and 8604 elements, depending on the mesh refinement (see Sect. 3.5.1). The optimisation problem is solved using a primal log-barrier interior point method [22] that allows for user-defined inner products. The optimisation is terminated when the $L^2(\Omega)$ norm of the functional gradient, computed using the $L^2(\Omega)$ inner product, is smaller than 10^{-7}.

The reader is reminded that this is a continuous optimisation problem, that is the turbine density with respect to which the optimisation is performed is a member of an infinite-dimensional function space, namely given by $L^2(\Omega)$. A discussion of this choice of function class was given in Sect. 3.2.3. Besides the PDE-constraint, which here is given by the nonlinear shallow water equations in steady-state form, the optimisation problem is subject to inequality constraints, that is a lower and upper bound for the turbine density function. For the optimisation step, the interior point method presented in Sect. 1.5.5 is applied, where its implementation allows the choice of the underlying inner product. Throughout this section, linear Lagrange finite elements were used for the spatial discretisation of the turbine density function.

Table 3.4: Parameters for the shallow water equations and the tidal turbine farm used for the continuous turbine density optimisation

Parameter	Symbol	Value	Units
Water density	ρ	1000	kg/m^3
Viscosity	ν	5	m^2/s
Water depth	h	50	m
Gravity	g	9.81	m/s^2
Natural bottom friction	c_b	0.0025	–
Turbine radius		10	m
Minimum distance between turbines		40	m
Maximum turbine density	$\bar{d}$	$6.25 \cdot 10^{-4}$	m^{-2}
Maximum turbine friction		0.059	–
Thrust coefficient	C_t	0.6	–
Turbine cross section	A_t	314.15	m^2

3.5.1 Mesh refinement

For the numerical analysis of the mesh-dependent behaviour when solving this optimisation problem, two approaches for a non-uniform refinement of the underlying meshes are considered.

In the first approach, meshes are refined according to a random refinement rule: starting with a mesh that is uniform in the turbine farm area in the middle of the rectangular mesh, the $(i+1)^{th}$ mesh is given by randomly refining the farm region in the i^{th} mesh, i.e. the refinement schema introduced in 2.3.2.1 is applied. The probability with which an element is labelled for refinement equals 0.15. The resulting randomly refined meshes used in the experiments are presented in Fig. 3.4.

In the second refinement approach, only a region around a single location within the mesh is refined. Starting with a mesh that is uniform in the turbine area, the resolution in element size around this point increases by a factor of five for every refinement. For the sake of saving space and due to the smaller complexity of this refinement schema, only one of these so-called point refined meshes is displayed, see Fig. 3.5.

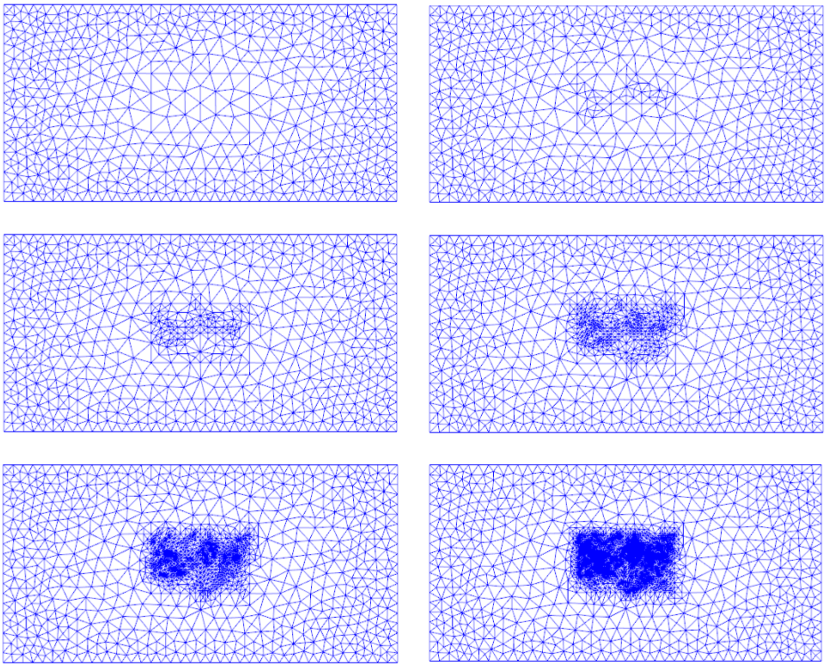


Fig. 3.4: Successively, randomly refined meshes in the turbine array area used for the simulations with results given in Fig. 3.8 and 3.9

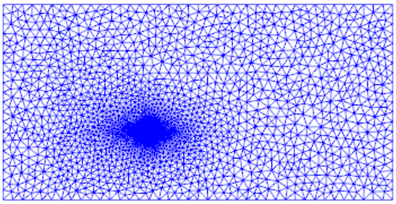


Fig. 3.5: Mesh with increased resolution around one point located on the boundary of the turbine region

3.5.2 Numerical experiments

In this section, the impact of using the ℓ^2 and L^2 Riesz representations of the functional derivative on the numerical solutions is analysed. Figure 3.6 displays the evolution of the profit functional with respect to iteration numbers for successively refined meshes.

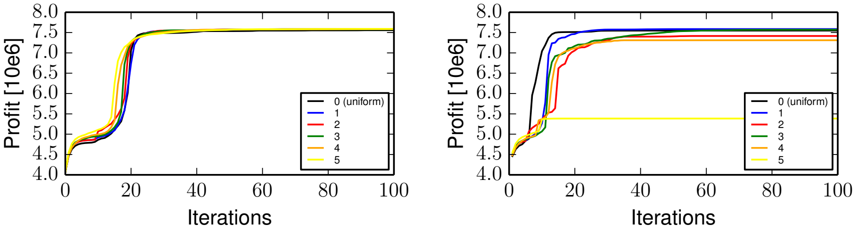
The results for the randomly refined meshes are shown in Figs. 3.7a and 3.7b. We make the following observations: for the uniform mesh (refinement level 0), the algorithm using the ℓ^2 representation of the functional derivative converges to the optimal value in fewer iterations than with the L^2 representation. However, successively refining the mesh decreases the rate of convergence behaviour for the ℓ^2 based algorithm. Furthermore, the optimal profit functional value achieved decreases with increasing mesh refinements.

Other than for the case with the uniform mesh, the speed of the algorithm with respect to both the iteration number as well as the convergence limit for the reduced functional is larger if the functional derivative representation in L^2 is used in the optimisation. The convergence limit and speed is stable for the optimisation method using L^2 Riesz represented gradients over the range of the underlying meshes considered.

Overall, similar behaviour in the numerical results between randomly refined and point refined meshes from Figs. 3.7c and 3.7d is observed. However, the point refinement rule does not lead to convergence limits of the profit functional in the ℓ^2 case that lie significantly below the optimum achieved using L^2 Riesz represented gradients. The convergence limit of the profit functional is reached after around 30 iterations using the L^2 inner product, independent of the refinement level. Using the ℓ^2 inner product, the iteration number increases with refinement. On the finest mesh, around 80 iterations are necessary.

The gradients with respect to both inner product representations have a different geometrical structure, see Fig. 3.7. In the first iteration ($d(x) = 0$), the gradients represented with respect to L^2 demonstrate a mostly flat increase of the friction field over the entire turbine area. It is further interesting to observe the Galerkin mass matrix scaling in the point refinement case for the ℓ^2 gradient representation

(a) L^2 representation & random refinement (b) ℓ^2 representation & random refinement



(c) L^2 representation & point refinement (d) ℓ^2 representation & point refinement

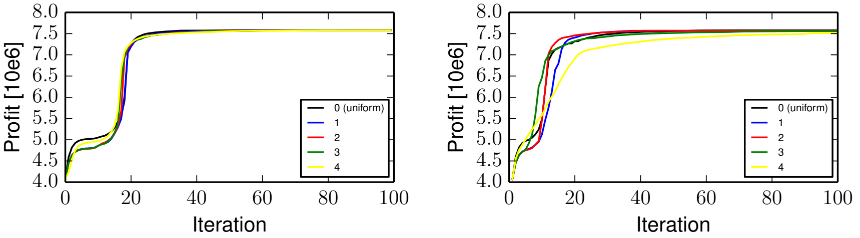


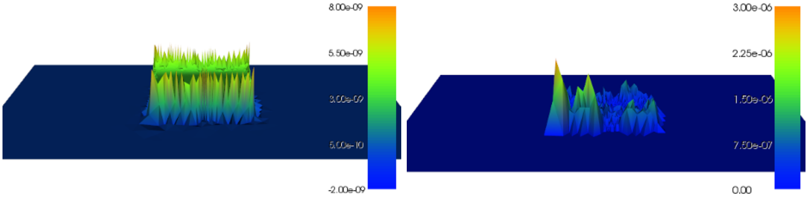
Fig. 3.6: Functional of interest given by the profit functional plotted as a function of optimisation iterations using L^2 and ℓ^2 gradient representations for successively refined meshes. For the upper figures, the random refinement rule is applied. For the lower figures, the point refinement rule is applied. The numbers in the legend state the degree of refinement of the underlying mesh

in such a clear manner: The smaller the size of the underlying element, the smaller the gradient values on this element.

Figures 3.8 and 3.9 present plots of the optimal friction field associated with the optimal turbine density function, obtained using the L^2 and ℓ^2 inner product representation for the optimisation method, respectively. The underlying meshes correspond to those shown in Fig. 3.4 (randomly refined meshes).

The reader is reminded that the boundary conditions for the shallow water equations, that are discussed in Sect. 3.4.1, describe a constant inflow from the western side of the domain towards the eastern side, and in this simple flat-bottomed channel geometry the flow remains close to west to east throughout the domain. Based on this information, the following consideration for an optimal turbine density distribution can be made: To maximise the power output (as part of the profit functional), a friction field associated with the turbine density field is formed that extracts the most possible power from the flow. Starting from $d = 0$, it makes sense if a considerable part of the increased friction field is orthogonal to the velocity. Due to the presence of increased friction in the farm area, and a consequent 'blockage' effect, the flow is diverted towards the Northern and Southern boundaries of the domain. To

(a) L^2 representation & random refinement (b) ℓ^2 representation & random refinement



(c) L^2 representation & point refinement (d) ℓ^2 representation & point refinement

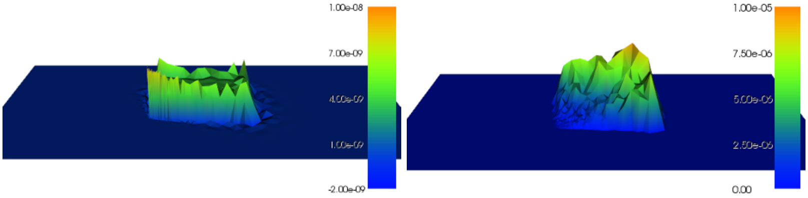


Fig. 3.7: Plots of the L^2 and ℓ^2 representations of the functional derivative on the finest mesh for both refinement schemas (top: random refinement; bottom: point refinement) for $d = 0$

'capture' most of the flow, and hence maximise energy extraction, the friction along the Northern and Southern boundaries of the turbine region might be expected to increase. Summarizing, one might expect the optimal friction distribution to have a "U" shape with the open side to the western side of the domain. This physical intuition is confirmed by the results given in Fig. 3.8, where the L^2 inner product was used. In this case, the overall friction field distribution does not change significantly when refining the underlying mesh, which confirms the observation of mesh independence in the optimisation for the reduced functional above.

Considering the results for the optimal friction distribution displayed in Fig. 3.9 (using ℓ^2) we observe that the friction field in this case does change with mesh refinement. Distributions that contradict the physical consideration from above yield slightly (refinement levels 2 and 4) to significantly (refinement level 5) lower profits.

To confirm the dependence of the ℓ^2 based algorithm on the non-uniformity in the mesh, a series of experiments using a series of uniformly refined meshes were performed as a reference (not displayed). For these the iteration number to reach convergence is observed to not increase using the ℓ^2 (as well as L^2) based algorithm with refinement. Analogously, the final optimal friction field (and hence profit functional value) does not change significantly.

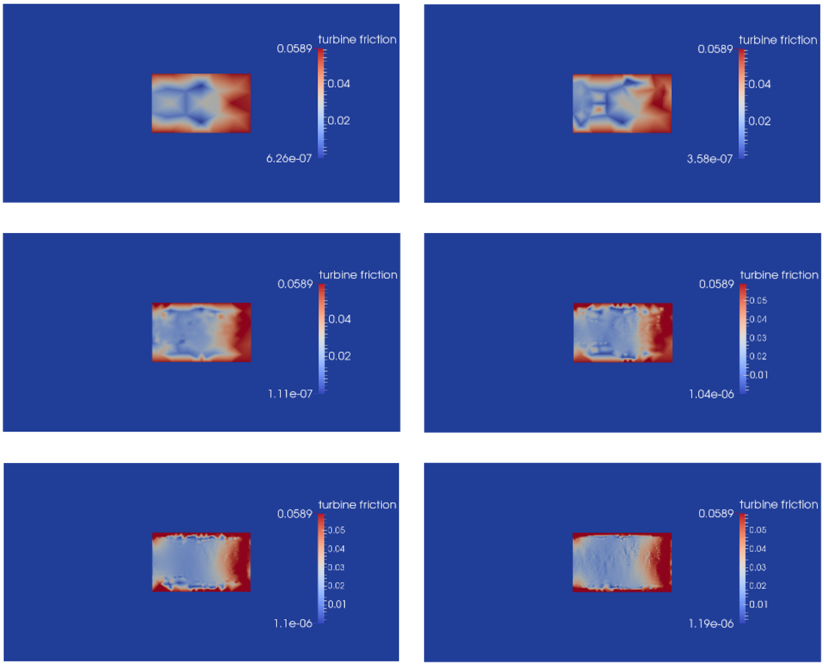


Fig. 3.8: Results of turbine density optimisation using the L^2 representation of the functional derivative for successively finer meshes corresponding to Fig. 3.4 (random refinement)

A Python script that performs the tidal turbine array optimisation with the configuration from this chapter can be found in [16]. In the optimisation step, the bounded LMVM method in the TAO library [14] is used.

3.6 Tidal resources assessment in the Pentland Firth, Scotland

In this example we apply the turbine layout optimisation to a realistic domain: the Pentland Firth between mainland Scotland and the Orkney Islands. The Pentland Firth is considered highly suitable for tidal turbine deployments [7], due to its fast tidal streams of up to 5 m/s speed in certain locations.

The aim of our experiment is to identify suitable deployment areas for tidal turbine farms. More specifically, the idea is to use the layout optimisation method as described in the previous sections, but apply it to all suitable areas across the Pentland Firth where turbines could be deployed. Since current tidal turbine technology can typically only be installed in waters with a certain depth range, we define this permissible area here as:

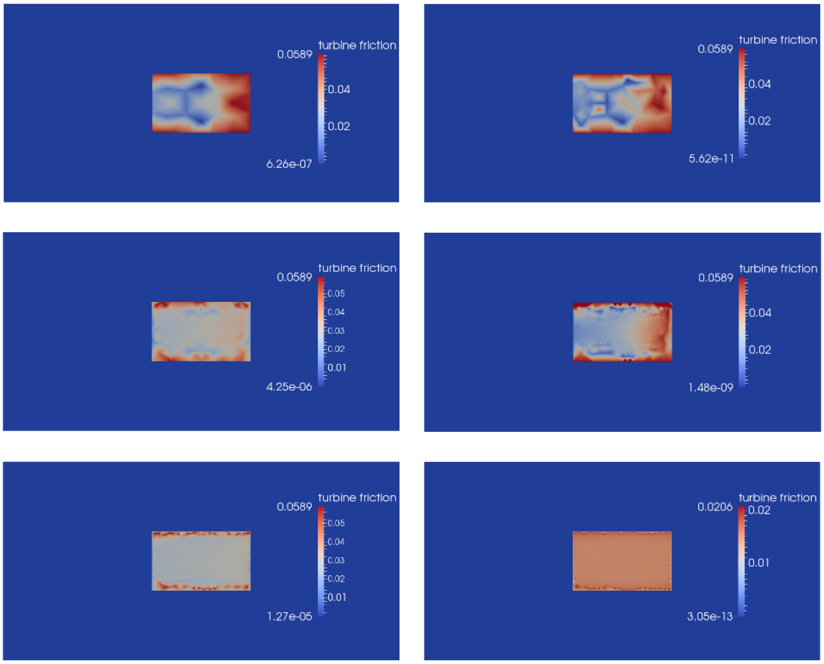
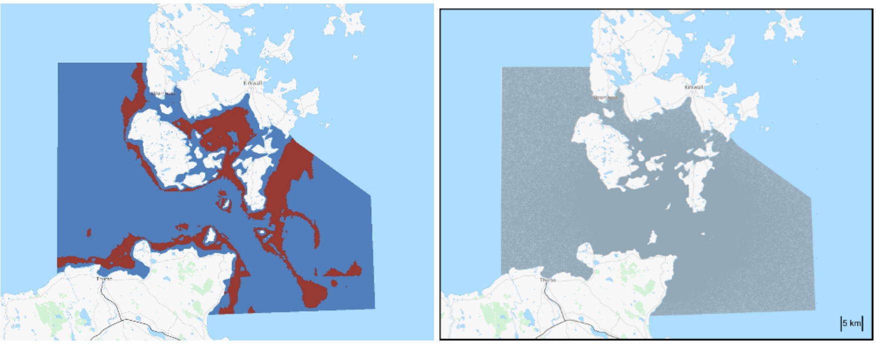


Fig. 3.9: Results of turbine density optimisation using the ℓ^2 representation of the functional derivative for successively finer meshes corresponding to Fig. 3.4 (random refinement)

$$\Omega_{\text{farm}} = \{x \in \Omega \text{ with } 25 \text{ m} \leq h(x) \leq 60 \text{ m}\}. \tag{3.41}$$

Figure 3.10a shows the domain of interest, and the sub-domain of potential turbine installations. The discretised domain is shown in Fig. 3.10b. The mesh element sizes are approximately 15 m in the proximity of the coast and vary up to 500 m in the deeper coastal ocean.



(a) Potential turbine area in red (b) Domain and mesh

Fig. 3.10: The domain, mesh and potential turbine areas (Ω_{farm}) of the Pentland Firth example. The Pentland Firth is located in Scotland, UK and is of high interest for tidal turbine array development

The formulation of the farm optimisation problem (3.1) was extended to capture the reversing flood-ebb flow of a tidal cycle by replacing the single shallow water constraint by two subsequent, steady-state shallow water solves. Each solve corresponds to one peak (flood and ebb) flow within one tidal cycle. For our experiment, the specific date stamps were 13:55 on the 18 September 2001, yielding a east-west flow, and 20:10 on the 18th September 2001, yielding a west-east flow in the Pentland Firth.

The objective functional is defined as the sum of the power productions (using (3.10)) at the two peak flows and a regularisation term:

$$J_P(u_1,d) + J_P(u_2,d) + C \int_{\Omega_{\text{farm}}} \|\nabla d(x)\|^2 \mathrm{d}x,$$

where $u_i, i = 1,2$ are the horizontal velocities from the two shallow water solves, d is the farm turbine density, and $C > 0$ is the regularisation coefficient. The H_0^1 regularisation term helps to enforce smoothness in the turbine density function, and is used to avoid infeasible "checkerboard" configurations. A list of the model parameters is shown in table 3.5. Overall, the parameters were chosen more as a demonstration setup for the numerical technology rather than to provide physically realistic predictions. In particular, the viscosity needed to be chosen artificially high in order for the steady-state solvers to converge. A more accurate model would therefore use a time-dependent shallow water solver.

The problem was solved with the bounded LMVM method in the TAO library [14]. To study the mesh dependency of this problem, the solver was executed both with the ℓ^2 and the L^2 inner products for the control variable. The convergence plots for both optimisations are plotted in Fig. 3.11. In both cases, the optimisation

Table 3.5: Parameters for the shallow water equations and the tidal turbine farm used for the continuous turbine density optimisation

Parameter	Symbol	Value	Units
Water density	ρ	1000	kg/m^3
Viscosity	ν	$10,000$	m^2/s
Water depth	h	[4]	m
Gravity	g	9.81	m/s^2
Natural bottom friction	c_b	0.0025	
Maximum turbine friction	$c_t(1)$	10.0	
Regularisation coefficient	C	10^4	

algorithm terminates in less than 25 iterations. The optimisation with the $L^2(\Omega)$ inner product can be seen to reach an optimum with a higher functional value.

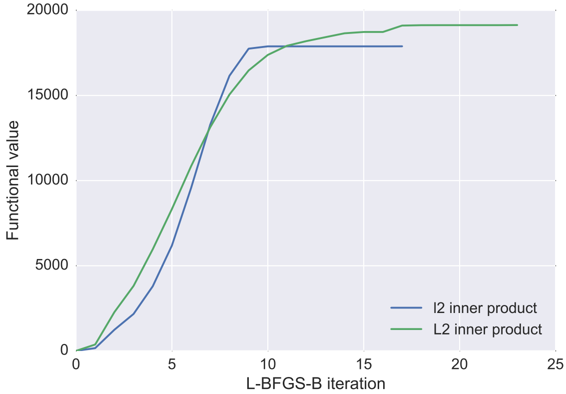
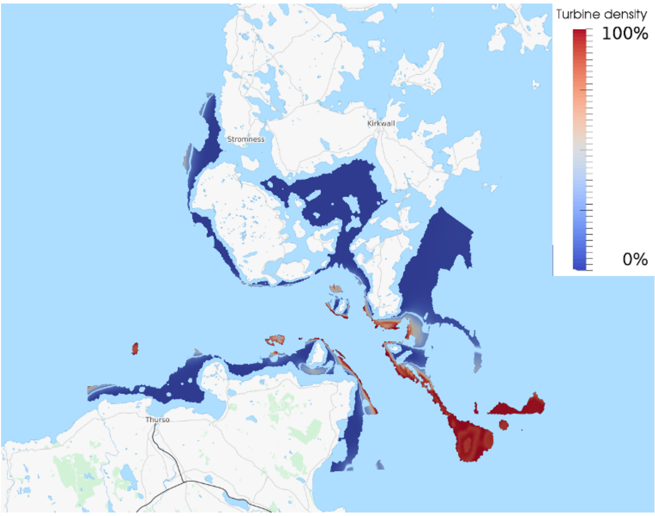
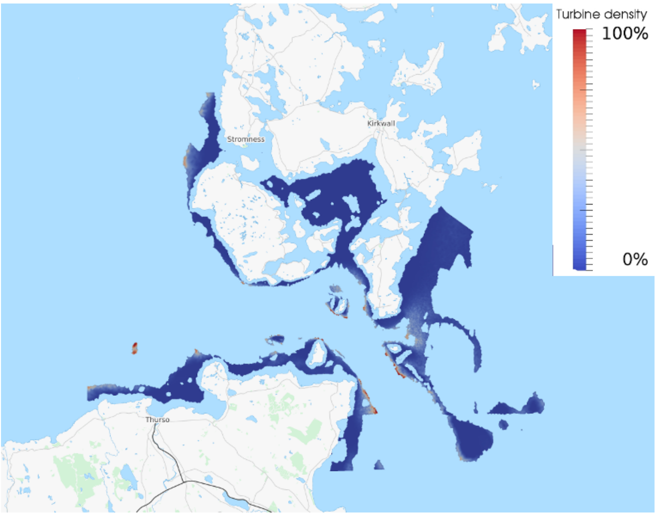


Fig. 3.11: The optimisation with the ℓ^2 inner product terminates after 17 iterations with a functional value of 17,889. The optimisation with the L^2 inner product terminates after 23 iterations with a functional value of 19,136.

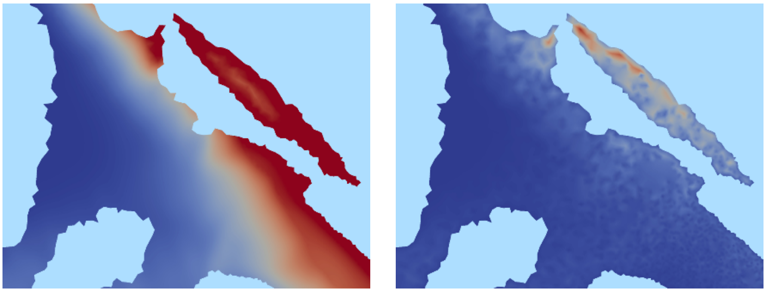
The optimal turbine configuration obtained for both the ℓ^2 and the L^2 based algorithms is shown in Fig. 3.12. The difference in the two solutions is clearly visible. In particular, in the south-east region, the configuration with the L^2 inner product (3.12a) deploys significantly more turbines compared to the solution with the ℓ^2 inner product (3.12b). Figures 3.12c and 3.12d provide a close-up view to the area east of Stroma island. In these figures, the ℓ^2-based solution shows the mesh-dependent structure, while the L^2-based solution is smoother and shows no mesh dependency.



(a) $L^2(\Omega)$



(b) ℓ^2



(c) $L^2(\Omega)$ (zoom) (d) ℓ^2 (zoom)

Fig. 3.12: The optimised turbine densities for the Pentland Firth example, computed with the L^2 inner product (figure a) and the ℓ^2 inner product (figure b). Figures c and d show a zoom of the region to the east of Stroma Island.

3.7 References

[1] Mohammad Amin Abolghasemi et al. “Simulating tidal turbines with multi-scale mesh optimisation techniques”. In: *Journal of Fluids and Structures* 66 (2016), pp. 69–90. ISSN: 0889-9746. DOI: `http://dx.doi.org/10.1016/j.jfluidstructs.2016.07.007`. URL: `http://www.sciencedirect.com/science/article/pii/S0889974615301729`.

[2] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. 3rd Edition, Springer Science+Business Media, LLC, 2008.

[3] Didier Bresch. “Shallow-Water Equations and Related Topics”. In: *Handbook of Differential Equations: Evolutionary Equations*. Ed. by C.M. Dafermos and Milan Pokorný. Vol. 5. Handbook of Differential Equations: Evolutionary Equations. North-Holland, 2009, pp. 1–104. DOI: `http://dx.doi.org/10.1016/S1874-5717(08)00208-9`. URL: `http://www.sciencedirect.com/science/article/pii/S1874571708002089`.

[4] British Oceanographic Data Centre (BODC). *GEBCO_08 bathymetry grid, version 20100927*. `http://www.gebco.net`. 2010.

[5] D Culley. *A hierarchy of approaches for the optimal design of tidal turbine farms*. `http://www.icoe2014canada.org/wp-content/uploads/2014/11/CulleyDavid_5-4.pdf`. 5th International Conference on Ocean Energy. 2014.

[6] Scott Draper et al. “An electrical analogy for the Pentland Firth tidal stream power resource”. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 470.2161 (2013). ISSN: 1364-5021. DOI: `10.1098/rspa.2013.0207`.

[7] Crown Estate. “Wave and tidal energy in the Pentland Firth and Orkney waters: How the projects could be built”. In: *The Crown Estate, London* (2011).

[8] P. E. Farrell et al. “Automated derivation of the adjoint of high-level transient finite element programs”. In: *SIAM Journal on Scientific Computing* (2012). accepted.

[9] P. E. Farell, S. W. Funke and M. D. Piggott. “Tidal turbine array optimisation using the adjoint approach”. In: *Renewable Energy* 63 (2014), pp. 658–673. DOI: `10.1016/j.renene.2013.09.031`.

[10] S.W. Funke. “The Automation of PDE-constrained Optimisation and its Applications”. PhD thesis. Imperial College London, UK, 2012.

[11] S.W. Funke, S.C. Kramer, and M.D. Piggott. “Design optimisation and resource assessment for tidal-stream renewable energy farms using a new continuous turbine approach”. In: *Renewable Energy* 99 (2016), pp. 1046–1061. ISSN: 0960-1481. DOI: `http://dx.doi.org/10.1016/j.renene.2016.07.039`. URL: `http://www.sciencedirect.com/science/article/pii/S0960148116306358`.

[12] Stephan C. Kramer and Matthew D. Piggott. “A correction to the enhanced bottom drag parameterisation of tidal turbines”. In: *Renewable Energy* 92

(2016), pp. 385–396. ISSN: 0960-1481. DOI: `10.1016/j.renene.2016.02.022`.

[13] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*. Vol. 84. Springer Science & Business Media, 2012.

[14] Todd Munson et al. *TAO 2.1 Users Manual*. Tech. rep. ANL/MCS-TM-322. USA: Argonne National Laboratory, 2012.

[15] Brian L. Polagye and Philip C. Malte. "Far-field dynamics of tidal energy extraction in channel networks". In: *Renewable Energy* 36.1 (2011), pp. 222–234. DOI: `http://dx.doi.org/10.1016/j.renene.2010.06.025`.

[16] Tobias Schwedes, David Ham, and Simon W Funke. *Code for simulations in Chapter 3 of ResearchBrief Code for simulations in Chapter 3 of ResearchBrief*. Dec. 2016. DOI: `10.5281/zenodo.224251`. URL: `https://doi.org/10.5281/zenodo.224251`.

[17] G. Sutherland, M. Foreman, and C. Garrett. "Tidal current energy assessment for Johnstone Strait, Vancouver Island". In: *Proceedings of the Institution of Mechanical Engineers A 221*. 2007, pp. 147–157.

[18] R. Vennell. "Tuning turbines in a tidal channel". In: *Journal of Fluid Mechanics* 663 (2010), pp. 253–267. DOI: `10.1017/S0022112010003502`.

[19] Ross Vennell. "Estimating the power potential of tidal currents and the impact of power extraction on flow speeds". In: *Renewable Energy* 36.12 (2011), pp. 3558–3565. ISSN: 0960-1481. DOI: `10.1016/j.renene.2011.05.011`.

[20] Ross Vennell et al. "Designing large arrays of tidal turbines: A synthesis and review". In: *Renewable and Sustainable Energy Reviews* 41 (2015), pp. 454–472. DOI: `10.1016/j.rser.2014.08.022`.

[21] C. B Vreugdenhil. *Numerical Methods for Shallow-Water Flow*. Springer, 1994.

[22] J. Young. *Optizelle: An open source software library designed to solve general purpose nonlinear optimization problems*. `www.optimojoe.com`, Open source software. 2016.