# UC San Diego

## UC San Diego Electronic Theses and Dissertations

### Title

Scalable Enforcement of Geometric Non-interference Constraints and Application to Gradient-based Wind Farm Layout Optimization

### Permalink

https://escholarship.org/uc/item/5w97h53z

### Author

Dunn, Ryan

### Publication Date

2023

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA SAN DIEGO

# Scalable Enforcement of Geometric Non-interference Constraints and Application to Gradient-based Wind Farm Layout Optimization

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Engineering Sciences (Aerospace Engineering)

by

Ryan C. Dunn

Committee in charge:

Professor John T. Hwang, Chair
Professor Thomas R. Bewley
Professor Tania K. Morimoto

2023

Copyright

Ryan C. Dunn, 2023

All rights reserved.

The Thesis of Ryan C. Dunn is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGEMENTS

I would like to acknowledge Professor John T. Hwang for his support as the chair of my committee. Through the multiple years of research, meetings, and ideas, his guidance has been invaluable to the development of this Thesis.

I would also like to acknowledge all co-authors to this material, as their insights and contributions were critical to the development of this Thesis. In particular, I want to acknowledge Anugrah Joshy for all of his mentorship and help throughout my degree.

Additionally, I acknowledge that none of this would have been possible without the support from my life-long friends and family. Especially my brother, my mother, and my father for supporting me my whole life and developing me into who I am today and who I will be tomorrow.

Sections 1.1 & 2.1 and Chapters 3 & 4, in full, are currently being prepared for submission for publication of the material. Anugrah J. Joshy, Jui-Te Lin, Cédric Girerd, Tania K. Morimoto, and John T. Hwang. The thesis author was the primary investigator and author of this material.

Sections 1.2 & 2.2 and Chapters 5 & 6, in part, are currently being prepared for submission for publication of the material. Anugrah Jo Joshy and John T. Hwang. The thesis author was a contributor to this material.

ABSTRACT OF THE THESIS

Scalable Enforcement of Geometric Non-interference Constraints and Application to Gradient-based Wind Farm Layout Optimization

by

Ryan C. Dunn

Master of Science in Engineering Sciences (Aerospace Engineering)

University of California San Diego, 2023

Professor John T. Hwang, Chair

Many design optimization problems, including the optimal layout of wind turbines within a wind farm, enforce constraints to prevent the design from crossing the boundary of the feasible space. When applying gradient-based optimization to such problems, the models must provide an accurate representation of the feasible space and be continuous, smooth, amenable to differentiation, and fast-to-evaluate. This thesis presents the work from two intertwined topics involved in modeling geometric non-interference constraints and wind farm optimization. First, we formulate a new method that uses B-splines to generate an efficient-to-evaluate level set function that approximates the signed distance

function for boundary constraint enforcement. Adapting ideas from the field of surface reconstruction, we formulate an energy minimization problem to calculate the B-spline control point values. Unlike previous constraint formulations, the new method requires an initial setup, but results in a more efficient and scalable representation of the geometric non-interference constraint. We present the results of accuracy and scaling studies performed on our formulation. Second, we perform optimization studies to the design of a wind farm. Utilizing the aforementioned constraint method, we perform a layout, yaw misalignment, and hub height optimizations to improve the annual energy production of a wind farm. These models used are verified and compared to industry-leading frameworks that conduct similar optimization studies. Overall, the new constraint method provides an effective way to conduct optimization studies, such as the one conducted in this thesis, with high computational efficiency, and shows promise as a tool for many more design optimization problems.

# Chapter 1

# Introduction

## 1.1 Geometric non-interference constraints

Accurate detection of physical interference between two or more bodies is crucial in the design of many engineering systems. Modeling interference between physical bodies is, therefore, an important problem in computational design.

Non-interference constraints appear in numerical optimization problems that manipulate an object within an environment containing other objects such that there is no collision. Efficient and accurate modeling of the non-interference constraints is critical for fast and reliable solutions in the overarching optimization problem. Prior literature on these problems describe these constraints using inconsistent terminology, e.g., anatomical constraints [1, 2], spatial integration constraints [3], boundary constraints [4, 5], and interference checks [6]. We observe that these terms represent the same underlying constraint. We propose to call these constraints *geometric non-interference constraints* since they are employed in design optimization to ensure a design where there exists no interference between two or more geometric shapes or paths of motion.

In our study, a *geometric shape* is associated with the design configuration of an engineering system at a particular instance of time. The geometric shapes of interest in this paper are curves in two dimensions, or orientable surfaces in three dimensions. We assume that the geometric shapes are non-self-intersecting but make no assumptions on

whether they are open or closed. A *path of motion* or *trajectory* is the set of points that traces the motion of a point on the engineering system as the system changes configuration over time. The paths considered in this paper are simply curves in two or three dimensions. We use the term *layout* to refer to a set of geometric shapes.

Based on the definitions above, we identify three major classes of optimization problems with geometric non-interference constraints: *layout optimization*, *shape optimization*, and *optimal path planning*. All three classes are parts of the scope of problems we address in this paper. Layout optimization optimizes the positions of geometric shapes via translation subject to geometric non-interference, with or without additional boundary constraints. For example, the wind farm layout optimization problem (WFLOP) consists in positioning the wind turbines within a wind farm in an optimal way such that interference between turbines and the boundary of the farm is avoided [4, 5, 7–9]. Another example of a layout optimization problem is the packing problem. Packing problems consist of positioning objects within a space to ensure the minimum space is occupied or the maximum number of objects are placed without geometric interference [3, 6, 10].

Shape optimization seeks to optimize geometric shapes subject to geometric non-interference, with or without additional boundary constraints. For example, shape optimization of an aircraft fuselage optimizes the shape of a fuselage with constraints ensuring that the passengers, crew, payload, and all the subsystems fit inside the fuselage [3].

Optimal path planning optimizes the trajectory of a point or a set of points subject to geometric non-interference, with or without additional boundary constraints. Robot motion planning is a class of problems that falls under optimal path planning and is widely researched [11]. The design optimization of surgical robots is an example of a problem involving robot motion planning that has had recent attention [1, 2]. In the design optimization of surgical robots, non-interference constraints are imposed such that the robot does not collide with the anatomy of a patient during operation. Additionally, it is desirable for the robot to maintain a safe distance from the anatomy, motivating the

use of a distance-based non-interference constraint formulation in such problems.

The problems just mentioned are solved using numerical optimization algorithms. Historically, gradient-free algorithms have been more commonly used to solve such problems, e.g., in layout optimization [12–14] and in robot motion planning [1]. A major reason behind this was the difficulty in efficiently computing the derivatives for a complex model. As models become more complex, that is, with more disciplines and design variables, solutions become impracticable with gradient-free algorithms since these algorithms scale poorly with the number of design variables. However, the recent emergence of modeling frameworks such as OpenMDAO [15] has enabled efficient design of large-scale and multidisciplinary systems using gradient-based optimization, including some of the aforementioned problems with geometric non-interference constraints [2–4, 8, 9].

Geometric non-interference constraint functions for gradient-based optimization require special consideration. These functions must be continuously differentiable or smooth in order to be used with a gradient-based optimization algorithm. They should also be efficient to compute because optimization algorithms evaluate constraint functions and their derivatives repeatedly over many optimization iterations. During some iterations, the optimizer may violate an interference constraint, and useful gradient information on such iterations is still required despite it being infeasible. Consequently, any non-interference constraint function must be defined in the event of an overlap between objects and provide necessary gradient information.

Figure 1.1 shows a diagram with two iterations of a design body in an optimization problem. One of the designs shown is feasible while the other is not. The feasible design is the one where the design body is completely inside the feasible space whereas the infeasible design has at least one point on the design body lying outside the feasible space. For the $\phi$ defined in Fig. 1.1, enforcing the optimization constraint $\phi(\mathbf{x}^{(i)}) \geq \epsilon$ for certain representative points $\mathbf{x}^{(i)}$ chosen on the surface of the design body guarantees non-interference by ensuring that all $\mathbf{x}^{(i)}$ stay within the feasible region for the final

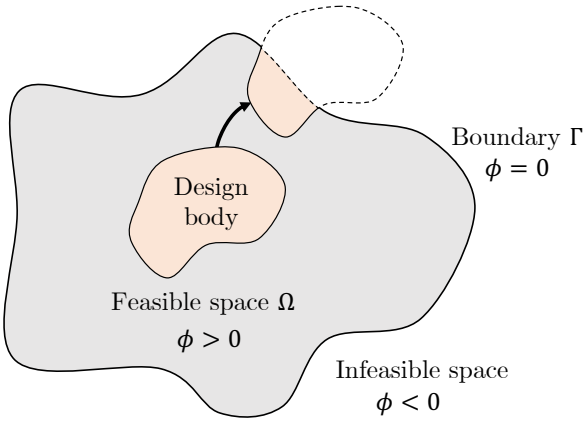


**Figure 1.1.** An ideal constraint function $\phi$ indicates geometric interferences using signed distances of representative points $\mathbf{x}^{(i)}$ on a body from the boundary $\Gamma$ defining the feasible space $\Omega$.

optimized design. The constant $\epsilon$ can be any small positive value appropriate for a given problem.

Existing non-interference constraint formulations suffer from various limitations. The formulation of quasi-phi-functions by Stoyan et al. [16] provides an analytical form to represent an interference for simple geometric shapes. Quasi-phi-functions are continuous but only piecewise continuously differentiable. These functions are also not generalized to represent any arbitrary shape. The formulation by Brelje et al. [3] is generalized to any triangulated 3D geometric shape, but has computational limitations. The computational complexity of their method is $\mathcal{O}(N_\Gamma)$, where $N_\Gamma$ is the number of elements in the triangulation. They are able to overcome this scaling issue by making use of graphics processing units (GPUs) but demonstrate their formulation on a geometric shape with only 626 elements in the triangulation. In their recent work on the WFLOP, Risco et al. [4] formulate a generic explicit method for geometric shapes in 2D, but the method suffers from the same scaling issues as in [3] and contains discontinuous derivatives. The formulation by Bergeles et al. [1] employs a distance potential function that is calculated with the $k$-nearest neighbors. With the use of a k-d tree structure, the computational complexity of the $k$-nearest neighbor search scales better than linearly, $\mathcal{O}(k \log(N_\Gamma))$ on

average, but the structure is not suitable for gradient-based optimization because the derivatives are discontinuous when the set of $k$-nearest neighbors switches.

Outside the domain of non-interference constraint formulations currently employed in optimization, we discovered a significant body of research conducted on a remarkably similar problem by the computer graphics community. Surface reconstruction in the field of computer graphics is the process of converting a set of points into a surface for graphical representation. A popular approach for surface reconstruction is the representation of surfaces by an implicit function. Implicit surface reconstruction methods such as Poisson [17], Multi-level Partition of Unity (MPU) [18], and Smooth Signed Distance (SSD) [19], to name a few, construct an implicit function from a point cloud to represent a surface. We observed that some of these distance-based formulations can be applied to overcome prior limitations in enforcing geometric non-interference constraints in gradient-based optimization.

The first objective of this thesis is to devise a general methodology based on an appropriate surface reconstruction method to generate a smooth and fast-to-evaluate geometric non-interference constraint function from an oriented point cloud. It is desired that the function locally approximates the signed distance to a geometric shape and that its evaluation time scales independently of the number of points sampled over the shape $N_\Gamma$. The function must also be an accurate implicit representation of the surface implied by the given point cloud. The contribution of this paper is a new formulation for representing geometric non-interference constraints in gradient-based optimization. We investigate various properties of the proposed formulation, its efficiency compared to existing non-interference constraint formulations, and its accuracy compared to state-of-the-art surface reconstruction methods. Additionally, we demonstrate the computational speedup of our formulation in an experiment with a path planning optimization and shape optimization problem.

This section, in full, is currently being prepared for submission for publication of

the material. Anugrah J. Joshy, Jui-Te Lin, Cédric Girerd, Tania K. Morimoto, and John T. Hwang. The thesis author was the primary investigator and author of this material.

## 1.2 Wind farm optimization

Wind energy is a sustainable method for electric power generation that mitigates greenhouse gas emissions from other power generation resources, such as with fossil fuels. Predictions show that the climate change mitigation from wind energy development ranges from 0.3°C to 0.8°C by 2100 [20]. Off-shore wind farms can also mitigate the impacts of hurricanes for coastal communities [21]. As such an impactful energy resource, the field of wind farm optimization has gained recent attention to maximize the energy production and economic feasibility of developing wind farms.

The increased adoption of multidisciplinary design optimization (MDO) techniques by the wind energy community has produced many recent works including the optimization of wind turbine designs [22], wind farm layouts [23], and active wind farm control [24]. In general, turbine design, wind turbine layout, and active turbine control strategies are the three main methods to increase wind farm efficiency by reducing the wake interaction between turbines [5]. Although these methods individually may increase the net efficiency, it has been shown that considering multiple or all three methods can further the a more optimal model. Recent simultaneous optimization studies include control and layout optimization [25, 26] and turbine design and layout optimization [5].

Numerical optimization, as an important design tool to solving these problems, has been widely used for wind farm optimization. Gradient-based and gradient-free algorithms are the two main algorithms to perform optimization. Historically, gradient-free algorithms have been used for wind farm optimization problems due to the high multi-modality in the design space of these problems [7, 27, 28]. Gradient-free optimizers are robust to local minima, while gradient-based optimizers often converge to a local optima. However,

as these problems increase in scale and the number of disciplines, the dimensionality of the design space may become impractical for gradient-free optimization. Gradient-free optimizers scale poorly in the number of function evaluations as the number of design variables increase in these complex wind farm problems. Gradient-based optimization, especially with analytic gradients, scales better in the number of function evaluations over gradient-free optimizers in these cases. In addition, recent developments have added methods for gradient-based optimizers to navigate the multi-modal design space of these problems [4, 29]. As a result, gradient-based optimization continues to play a key role in optimizing wind farms.

When modeling wind farms for gradient-based optimization, it is important to consider the computational speed and differentiability of the models. High fidelity models are often very computational expensive to evaluate, and these models must be evaluated up to hundreds of times during optimization. Therefore, lower fidelity models that are less computationally expensive are often considered for use in gradient-based optimization. Additionally, the differentiability of the models is a requirement in order to perform gradient-based optimization. The ability to calculate derivatives within the model has not always been readily available. Oftentimes, significant effort must be made to hand-derive the derivatives [22, 30], or in the worst case, using the finite difference method for derivatives [31], which is on the same order of function evaluations as gradient-free optimization. Current state-of-the-art gradient-based optimizations are performed using automatic differentiation [4, 5, 23], however it still requires a level of effort to implement into new models, especially when local smoothing techniques are required [32].

A notable research problem in wind farm layout optimization is the representation of wind farm boundary constraints. Boundary constraints in wind farm layout optimization prevent the placement of a wind turbine on regions outside of the permitted zone. Examples of exclusion zones for off-shore wind farms include unsuitable seabed gradients, shipwrecks, and shipping lanes. These zones are often disjoint, non-convex, and highly irregular

shapes represented in 2D. There exists a lack of a generic method to represent these boundaries in the wind farm optimization community [33]. Additionally, the state-of-the-art methods suffer from the same problems noted in Section 1.1, where the computational complexity scales with the number of points representing the polygonal wind farm boundary [4]. Conveniently, the first contribution of this thesis addresses this issue. The new geometric non-interference constraint formulation provides a smooth, differentiable, and fast-to-evaluate constraint function that represents the wind farm boundary suitable for gradient-based optimization.

Another tool that may show to benefit gradient-based wind farm optimization is a new modeling code language called the computational system design language (CSDL) [34]. CSDL is an algebraic modeling language for defining numerical models that fully automates adjoint-based sensitivity analysis. Additionally, CSDL contains a three-stage compiler system that constructs an optimized computational graph representation of the models. As a new design language, it shows potential to improving the convenience and speed of developing the models to perform gradient-based wind farm optimization.

The second objective of this thesis is to implement the two aforementioned tools–the geometric non-interference constraint formulation and the computational system design language (CSDL)–and perform optimization studies on multiple wind farm optimization problems. We conduct optimization studies on turbine hub heights, turbine yaw misalignment, and wind farm layout, and investigate their properties as it pertains to gradient-based optimization. These three problems demonstrate the potential of gradient-based optimization in turbine design, wind farm control, and wind farm layout optimization problems. Using well know analytical models, we conduct multiple optimization studies using CSDL as a modeling paradigm and verify its accuracy with other industry-leading optimization frameworks. Additionally, we perform a wind farm layout optimization with a real-world wind farm, highlighting the accuracy and efficiency of the geometric non-interference constraint formulation.

This section, in full, is currently being prepared for submission for publication of the material. Anugrah J. Joshy and John T. Hwang. The thesis author was a contributor to this material.

# Chapter 2

# Literature Review

## 2.1 Related work for geometric non-interference constraints

This section presents an overview of related work to the defined research problem. We begin by reviewing prior methods for enforcing non-interference constraints in gradient-based optimization in subsection 2.1.1. We then review the problem of surface reconstruction and its complexities in subsection 2.1.2. In subsection 2.1.3, we present a literature review for methods that approximate the signed distance function to contextualize our formulation within the field of surface reconstruction.

### 2.1.1 Prior methods for enforcing non-interference constraints in gradient-based optimization

We identify two preexisting methods for enforcing geometric non-interference constraints in gradient-based optimization that are both continuous and differentiable. Previous constraint formulations that utilize the nearest neighbor distance, e.g., Risco et al. [4] and Bergeles et al. [1], have been used in optimization, but we note the that they are non-differentiable and may incur numerical difficulties in gradient-based optimization.

Brelje et al. [3] implement a general mesh-based constraint formulation for non-interference constraints between two triangulations of objects. Two nonlinear constraints

define their formulation. The first constraint is that the minimum distance of the design shape to the geometric shape is greater than zero, and the second constraint is that the intersection length between the two bodies is zero, i.e., there is no intersection. A binary check, e.g., ray tracing, must be used to reject optimization iterations where the design shape is entirely in the infeasible region, where the previous two constraints are satisfied. As noted by Brelje et al., this formulation may be susceptible to representing very thin objects, where the intersection length is very sensitive to the step size of the optimizer. Additionally, the constraint function has a computational complexity of $\mathcal{O}(N_\Gamma)$, which may be addressed by the use of graphics processing units (GPUs).

Lin et al. [2] implement a modified signed distance function, making it differentiable throughout. Using an oriented set of points to represent the bounds of the feasible region, the constraint function is a distance-based weighted sum of signed distances between the points and a set of points on the design shape. This representation is inexact and is found to compromise accuracy for a smoothness in the constraint representation in practice. Additionally, their formulation has a computational complexity of $\mathcal{O}(N_\Gamma)$.

### 2.1.2 Surface reconstruction

Our first objective—to derive a smooth level set function from a set of oriented points—closely aligns with the problem of surface reconstruction in computer graphics. Surface reconstruction is done in many ways, and we refer the reader to [35, 36] for a full survey on surface reconstruction methods from point clouds. We, in particular, focus on surface reconstruction with implicit function representations from point clouds. Implicit surface reconstruction is done by constructing an indicator function between the interior and exterior of a surface, whose isocontour represents a smooth surface implied by the point cloud. The methodologies for surface reconstruction use implicit functions as a means to an end; however, the focus of our investigation is on the implicit function itself for enforcing non-interference constraints. We identify that the direct connection between

non-interference constraints and implicit functions in surface reconstruction is that the reconstructed surface represents the boundary between the feasible and infeasible region in a continuous and differentiable way.

The surface reconstruction problem begins with a representation of a geometric shape. Geometric non-interference constraints may be represented by geometric shapes using scanned samples of the surface of an anatomy [1, 2], outer mold line (OML) meshes [3], user defined polygons [4], and a sampled set of points of seabed depths [7]. Many geometric shape representations, including those mentioned, can be sampled and readily converted into an oriented point cloud and posed as a surface reconstruction problem.

The construction of any point cloud comes with additional complexities. For example, machine tolerance of scanners introduce error into scans, and meshing algorithms produce different point cloud representations for the same geometric shape. As a result, implicit surface reconstruction methods often take into consideration nonuniform sampling, noise, outliers, misalignment between scans, and missing data in point clouds. Implicit surface reconstruction methods have been shown to address these issues well, including hole-filling [37–39], reconstructing surfaces from noisy samples [17, 40, 41], reconstructing sharp corners and edges [40], and reconstructing surfaces without normal vectors in the point cloud [41, 42].

Basis functions are commonly used to define the space of implicit functions for implicit surface reconstruction. Basis functions are constructed from a discrete set of points scattered throughout the domain, whose distribution and locations play an important role to defining the implicit function. Examples of these points include control points for B-splines, centers for radial basis functions, and shifts for wavelets. Implicit surface reconstruction methods distribute these points in various ways.

One approach is to adaptively subdivide the implicit function's domain using an octree structure. Octrees, as used by [17–19, 41, 43, 44], recursively subdivide the domain into octants using various heuristics in order to form neighborhoods of control points near

the surface. Heuristics include point density [19], error-controlled [18], and curvature-based [44] subdivisions. Octrees are notable because the error of the surface reconstruction decays with the sampling width between control points, which decreases exponentially with respect to the octree depth [17]. Additionally, the neighborhoods of control points from octrees can be solved for and evaluated in parallel using graphics processing units (GPUs), which allows for on-demand surface reconstruction as demonstrated in [43].

Another approach for distributing the points that control the implicit function is to locate them directly on the points in the point cloud. In the formulation by Carr et al. [45], a chosen subset of points in the point cloud and points projected in the direction of the normal vectors are used to place the radial basis function (RBF) centers, resulting in fewer centers than octrees that are still distributed near the surface. The explicit formulation by Hicken and Kaur [46] uses all points in the point cloud to define the implicit function and shows favorable decay in surface reconstruction error as the number of points in the point cloud $N_\Gamma$ increases. This structure has been used in combination with RBFs for hole-filling in [37] and anisotropic basis functions for representing sharp corners in [40].

Another approach is to construct a uniform grid of points to control the implicit function. Unlike the aforementioned approaches, the distribution of points is decoupled from the resolution of the point cloud. As a result, deformations to the geometric shape can be represented without loss in accuracy near the surface as shown by Zhao et al. [38]. This makes it a popular structure in partial differential equation (PDE) based reconstruction methods that evolve the surface during reconstruction, such as in [47, 48]. In general, more points representing the implicit function are required to achieve the same level of accuracy to other approaches. As a result, implicit functions defined by a uniform grid are more computationally expensive to solve for in both time and memory usage than the aforementioned approaches, as experienced by Sibley and Taubin [49], but can be reduced by a GPU-based multigrid approach as implemented by Jakobsen et al. [48].

### 2.1.3 Formulations for signed distance function approximation

The signed distance function (SDF) presents an ideal candidate for implicit surface reconstruction and geometric non-interference constraints. It is known that the zero level set of the SDF is a smooth representation of the points in a point cloud, and its gradient field is a smooth representation of the normal vector field from the normal vectors in a point cloud. As a result, many formulations to approximate the SDF have been researched for implicit surface reconstruction. We note that there exists other methodologies, such as wavelets [50] and a Fast Fourier Transform (FFT) based method [51], that fit a smooth indicator function instead, but are less applicable for non-interfernce constraints where a measurement of distance is desired. We identify four categories that approximate the SDF in some way: explicit formulations, interpolation formulations with RBFs, PDE-based formulations, and energy minimization formulations.

**Explicit formulations**

Explicit formulations use the data defined in the point cloud to define linear approximations to the SDF. These formulations then apply smoothing to these linear approximations in order to define the level set function. Risco et al. [4] present the simplest approach which uses the nearest edge and normal vector to define the function explicitly. The resultant constraint function is piecewise continuous but non-differentiable at points where the nearest edge switches. Belyaev et al. [52] derive a special smoothing method for defining signed $L_p$-distance functions, which is a continuous and smooth transition between piecewise functions. Hicken and Kaur [46] use modified constraint aggregation methods to define the function in a smooth and differentiable way. Upon the investigation of Hicken and Kaur, the signed $L_p$-distance functions give poor approximations of the surface. Additionally, Hicken and Kaur's formulation is shown to increase in accuracy as the data in the point cloud, number of points $N_\Gamma$, increases. We identify Hicken and Kaur's explicit formulation as a good candidate for enforcing non-interference constraints,

as it is continuous and differentiable with good accuracy.

Given an oriented point cloud is a set of ordered pairs $\mathcal{P} = \{(\mathbf{p}_i, \vec{\mathbf{n}}_i) : i = 1, \ldots, N_\Gamma\}$, where $\mathbf{p}_i$ is the location of the points sampled over the geometric shape, and $\vec{\mathbf{n}}_i$ are the unit normal vectors at $\mathbf{p}_i$, the level set function by Hicken and Kaur [46] is given as

$$\phi_H(\mathbf{x}) = \frac{\sum_{i=1}^{N_\Gamma} d_i(\mathbf{x}) e^{-\rho(\Delta_i(\mathbf{x}) - \Delta_{min})}}{\sum_{j=1}^{N_\Gamma} e^{-\rho(\Delta_j(\mathbf{x}) - \Delta_{min})}}, \tag{2.1}$$

where $d_i(\mathbf{x})$ is the signed distance to the hyperplane defined by the point and normal vector pair in the point cloud $(\mathbf{p}_i, \vec{\mathbf{n}}_i)$, $\Delta_i(x)$ is the Euclidean distance to the point $\mathbf{p}_i$, $\Delta_{min}$ is the Euclidean distance to the nearest neighbor, and $\rho$ is a smoothing parameter. To improve accuracy, Hicken and Kaur suggest modifications to make the linear approximation in to a quadratic approximation by using the principal curvatures of the surface. Unless readily provided by a smooth geometric representation, the principal curvatures must be approximated from the point cloud, similar to the approximation method by Tang and Feng [44]. To reduce the computational complexity, Hicken and Kaur suggest only evaluating the $k$-nearest neighbors, since the basis weights exponentially decay with distance. However, using the $k$-nearest neighbors will remove the function's differentiability as the set of $k$-nearest neighbors changes. As a result, their formulation must scale in computational complexity by $\mathcal{O}(N_\Gamma)$ to be differentiable. While not originally purposed for geometric non-interference constraints, the formulation by Hicken and Kaur is on par in computational complexity with current non-interference constraint formulations [2–4]. No calculations are required to set up the function because the equation is explicitly defined by the data in the point cloud, but it is susceptible to the various complexities of point clouds, such as noise and outliers.

**Interpolation formulations with radial basis functions**

Another method to construct the level set function is to solve an interpolation problem given an oriented point cloud $\mathcal{P}$. Because the data points of $\mathcal{P}$ always lie on the zero contour, nonzero interpolation points for the implicit function can be defined on the interior and exterior, as originally done by Turk and O'Brien [53]. Radial basis functions (RBFs) are then formulated to interpolate the data. To avoid overfitting, thin-plate splines can be used to formulate the smoothest interpolator for the data, as noted in [37, 45]. Solving for the weights of a RBF involves solving a linear system, which is often dense and very computationally expensive due to their global support. Turk and O'Brien [53] solve up to 3,000 RBF centers, and improvements by Carr et al. [45] allow up to 594,000 RBF centers to be constructed in reasonable time (hours). On top of the significant computational expense, interpolating RBFs have been criticized for having *blobby* reconstructions [40, 53] which poorly represent sharp features in the geometric shapes.

**PDE-based formulations**

Another approach is to construct the level set function as a smooth vector field that smoothly approximates the normal vectors $\vec{\mathbf{n}}_i$ given by the point cloud $\mathcal{P}$. The vector field is then integrated and fit, usually by a least squares fitting, to make the zero level set fit the point cloud. We classify the methods that solve for the vector field as a solution to a partial differential equations (PDEs) as PDE-based methods. Poisson's method [17] uses variational techniques to Poisson's equation to construct a vector field. Improvements to this method add penalization weights to better fit the zero contour to the point cloud in [54]. Tasdizen et al. [47] prioritize minimal curvature and minimal error in the vector field by solving a set of coupled second order PDEs to derive their level set function. Zhao et al. [38] use the level set method, originally introduced by Osher and Sethian [55], for surface reconstruction, with the advantage of modeling deformable shapes. In the aformentioned

PDE-based methods, the setup for the implicit function reduces to solving a PDE by time-stepping [38, 47] or a sparse linear system [17, 51] in the case of Poisson's equation. Kazhdan et al. [17] note that care should be taken when choosing a smoothing filter for the normal field defined by $\vec{\mathbf{n}}_i$, especially for nonuniformly sampled points. In the analysis done by Calakli and Taubin [19], they found that Poisson's method often over-smooths some surfaces. We also note that solutions to PDEs are more difficult to implement than other methods in practice.

**Energy minimization formulations**

Another methodology is to solve an optimization problem that minimizes some energy function with respect to the values of the basis function directly. The smooth signed distance (SSD) surface reconstruction method [19] minimizes an energy function with three terms. Minimizing these three terms maximizes smoothness and minimizes the approximation error of the zero level set and the gradient field to the data in $\mathcal{P}$, all in a least squares sense. Alternative forms, such as in [18, 44], propose a different energy term to this formulation, which does a direct least squares fit to the approximate signed distance function. We perform a more thorough discussion of the four energy terms in Chapter 3, as our method also poses an energy minimization problem.

The energy minimization problem posed by these papers is a well-posed unconstrained quadratic programming (QP) problem. The solution to these unconstrained QP problems reduces to the solution of a linear system. Making use of hierarchical structures, such as octrees, and compactly supported basis functions, the linear system is sparse and recursively solved at increasing depths of the structure. These advantages allow for fast solutions on the order of minutes as reported by [19, 44]. It should be noted that the time and space (memory) consumed by hierarchical approaches grows exponentially with the depth of the octree, so many implementations limit the depth up to 11. The resultant number of control points in Tang and Feng [44] are on the order of $10^6$.

**Summary**

We note that interpolation formulations with RBFs, PDE-based formulations, and energy minimization formulations are different approaches to the same problem of approximating the SDF. The primary differences lie within the derivation and implementation of such methods. The energy minimization formulation by Calakli and Taubin [19] performs a least squares fit to the data in the point cloud. Thin-plate spline RBFs are an exact solution to the same energy minimization formulation to interpolate the data and maximize smoothness, as derived by [56]. The two-step energy minimization formulation by Sibley and Taubin [49] follows the same approach as PDE-based methods, where a vector field is solved for and then a least squares fit is done to fit the surface. We recommend the interested reader to Calakli and Taubin [19] who discuss the similarities and differences between SSD and Poisson surface reconstruction methods.

**Table 2.1.** Signed distance approximations and their basis function representations.

| | Formulation | Basis Function | Point Distribution |
|---|---|---|---|
| [46] | Explicit | Exponential | On-surface |
| [53] | Interpolating | Thin-plate RBF | On- & off-surface |
| [45] | Interpolating | Polyharmonic RBF | On- & off-surface |
| [40] | Interpolating | Anisotropic RBF | On- & off-surface |
| [47] | PDE-based | Polynomial | Uniform |
| [38, 48] | PDE-based | Linear | Uniform |
| [17] | PDE-based | Quadratic | Octree |
| [54] | PDE-based | Quadratic B-splines | Octree |
| [49] | Energy minimization | Linear | Uniform |
| [19] | Energy minimization | Linear | Octree |
| [18, 41, 44] | Energy minimization | Quadratic B-splines | Octree |
| Our method | Energy minimization | Cubic B-splines | Uniform |

We summarize the context for all the methods in Table 2.1, highlighting the main differences in their formulation, basis function representation, and distribution of points controlling the function. We note our method is an energy minimization formulation, which uses the same energy terms as Calakli and Taubin [19], but with a different basis

function and different distribution of control points.

This section, in part, is currently being prepared for submission for publication of the material. The authors of this work are Ryan C. Dunn, Anugrah Jo Joshy, Jui-Te Lin, Cédric Girerd, Tania K. Morimoto, and John T. Hwang. The thesis author was the primary investigator and author of this material.

## 2.2 Related work for wind farm optimization

**Objective functions**

Wind farm optimization problems contain an objective to be minimized (or maximized) with respect to design variables. The objective function is important to the optimization problem because its minimum defines the optimal design of the wind farm. When an objective function is narrow in scope, the optimal result often compromises other aspects of the design. We tabulate the different objective functions in Table 2.2. Historically, many wind farm optimizations have focused on the annual energy production (AEP) of a wind farm, which is considered to be too narrow of a scope. When optimizing AEP, solutions may be difficult to manufacture, economically impractical, and over-reliant on the model's assumptions making it impractical in the real-world. The levelized cost of energy (LCoE) objective function is larger in scope and emphasizes efficient and economically feasible solutions, and is often considered a better objective than AEP [57]. Recent work has converted an existing hub height optimization using AEP [58] to improve the economic viability and competitiveness using LCoE as the objective function [59]. Other optimizations such as cable length, noise, and mass are considered more narrow in scope. In some cases, multiple objective functions are considered in an optimization [28, 60]. For further conversation on the selection of an appropriate objective function, we direct the reader to [57].

**Table 2.2.** Objective functions used in wind farm design optimization.

| Objective Functions | |
|---|---|
| Annual Energy Production (AEP)* | [23–25, 32, 57, 58, 61–64] |
| Levelized Cost of Energy (LCoE) | [5, 22, 31, 57, 59] |
| Noise level | [60, 65–67] |
| Cable length | [31] |
| Power density | [31] |
| Turbine mass | [57, 68] |

Note: (*) indicates what is investigated in this thesis.

### Design variables

The design variables of the wind farm optimization are also important to defining the problem. In terms of decreasing the wake interactions of wind turbines, three main methods emerge that define the design variables of each problem. The first method uses the turbine positions as design variables and is called the *wind farm layout* optimization problem (WFLOP). The WFLOP has a highly multi-modal design space that is especially challenging for gradient-based optimization. The second method uses the turbine's tower, blade, and rotor-nacelle assembly as design variables and is the *turbine design* optimization problem. The third method uses the turbine's control to affect its wakes and is the *turbine control* optimization problem. We refer the reader to two reviews on wind turbine controls strategies [69, 70]. A summary of the design variables considered in previous gradient-based optimization studies is in Table 2.3. Note that some recent studies consider two optimization problems at once: simultaneous layout and control optimization [25] and simultaneous layout and turbine design optimization [5]. In these studies, the optimal results were improved compare to solving the optimization problems individually.

### Boundary constraints

An important problem in the wind farm layout optimization problem (WFLOP) is the representation of the wind farm boundary. The wind farm boundary defines the feasible regions of a wind farm, and can be limited due to the geographic features of

**Table 2.3.** Design variables used in the three types of wind farm design optimization problems.

| Optimization Problem | Design Variables | |
|---|---|---|
| Wind farm layout | Positions $(x, y)^*$ | [23, 25, 26, 31, 60, 63–65, 67] |
| Turbine design | Hub heights $(z_h)^*$ | [5, 22, 58, 59] |
| | Rotor diameters $(D)$ | [5, 22, 57] |
| | Blade geometry | [22, 57, 66] |
| | Tower geometry | [5, 22, 68] |
| Turbine control | Yaw misalignment $(\gamma)^*$ | [23, 24, 26, 31, 32, 62] |
| | Turbine derating | [25, 57, 61] |

Note: $(^*)$ Indicates what is investigated in this thesis.

the land or sea. In optimization, these zones must be enforced as constraints to ensure turbines are placed in a feasible zone. For gradient-free optimization, the representation of these boundaries are simple. Wind farm boundaries may be enforced by a binary function [28] or by discretizing the domain and excluding the points outside of the feasible space [7]. However, a gradient-based boundary constraint function must be continuous and differentiable. Many papers have tried to address this issue in their own unique ways. A common method seen is a polygonal representation of the boundary. With these polygons, Guirguis et al. uses the nearest edge to define an equivalent circular constraint [30]. Risco et al. uses the nearest edge of the polygon to define the signed distance function (SDF) [4]. Note that both of these method's use of the nearest edge in their formulation, which means that the constraint function is non-differentiable when there are two edges of equal distance away from the wind turbine. Additionally, there is a trade-off between computational cost and accuracy in the constraint function with respect to the number of edges in the polygon [4]. The unique approach by Reddy poses an interpolation problem using a support vector domain description (SVDD) to describe the boundaries in an optimal way [33]. This approach is continuous and differentiable, but the report does not provide any accuracy and scaling studies, so it is unknown how it may perform with increasingly complex boundaries.

To the author's knowledge, no other wind farm boundary constraint has taken the unique implicit surface reconstruction perspective as was described in Section 2.1. Implicit surface reconstruction constructs a level set function that represents the wind farm boundary by its zero level set. The inputs to most surface reconstruction methods is a point cloud, which is easily obtainable from the polygonal representations of previous studies. Note that Reddy's SVDD method is very similar to the interpolation formulations in surface reconstruction. In this way, SVDD may suffer from the aforementioned limitations, including *blobby* reconstructions [40, 53] and significant computational expense for large scale SVDDs [45].

The SDF is useful when the feasible zones are disconnected. In gradient-based optimization, it is not possible for wind turbines to move across disconnected domains in a continuous way, however, a relaxation approach to the boundary constraints can allow for wind turbines to move freely across regions until the true constraint is represented. The relaxation approach is shown to work well in avoiding turbines getting stuck within disconnected domains [4]. With the SDF approximation described in Chapter 3, the relaxation approach is easily repeatable using the surface reconstruction method.

**Wind farm optimization frameworks**

Many frameworks to model and support the optimization of wind farms have been developed. We identify three of these frameworks relevant to the work within this thesis: PyWake, TOPFARM, and FLORIS. PyWake is an open-source Python library that contains many engineering wake models to calculate the annual energy production of wind farms. TOPFARM is an OpenMDAO-based Python library that performs gradient-based optimization on wind farms using PyWake as its backend for modeling turbine wakes and calculating annual energy production. FLORIS is an open-source Python library that is controls-focused framework that models the wakes of yawed turbines and supports gradient-free and gradient-based optimization. Gradient-based optimization within FLORIS is at a

limited capacity, as it only supports automatic differentiation on some models. Otherwise, the finite difference method is used for derivative calculation. The finite difference method is less accurate and requires many more model evaluations.

This section, in part, is currently being prepared for submission for publication of the material. The authors of this work are Anugrah Jo Joshy, Ryan C. Dunn, and John T. Hwang. The thesis author was a contributor to this material.

# Chapter 3

# Novel methodology for Enforcing Geometric Non-interference Constraints

## 3.1 Signed distance function

We now present our methodology to compute an implicit level set function $\phi$ by approximating the signed distance function (SDF) of a geometric shape. We assume that the geometric shape partitions its neighboring space into a feasible region and an infeasible region as shown in Fig. 3.1. Our goal is to generate a level set function such that the zero contour of the function approximates the boundary between the feasible and infeasible regions, i.e., the geometric shape. We also require that evaluating the implicit function at any point in the domain of interest will determine if the point is located on the boundary or within one of the two regions, as indicated by the signed distance of the point from the boundary. We follow the convention of denoting distances in the feasible region as positive and those in the infeasible region as negative. The signed distance function in the neighborhood $\mathcal{N} \subset \mathbb{R}^n$ of a point on the geometric shape can then be defined as

$$d_\Gamma(\mathbf{x}) = \begin{cases} +D(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega \\ -D(\mathbf{x}) & \text{if } \mathbf{x} \in \mathcal{N} \setminus \Omega \end{cases} \tag{3.1}$$

where $D : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ measures the shortest distance of a point $\mathbf{x}$ to the boundary $\Gamma$. The local feasible region $\Omega \subset \mathcal{N}$ and the local infeasible region $\mathcal{N} \setminus \Omega$ are separated by the local boundary $\Gamma_l = \Gamma \cap \mathcal{N}$ within the neighborhood, and $\Gamma_l \subset \Omega$. Note that $n = 2$ or $n = 3$ for geometric shapes: $n = 2$ implies $\Gamma$ is a curve in two dimensions while $n = 3$ implies $\Gamma$ is an orientable surface in three dimensions. We assume that $\Gamma$ is always a connected set. We make no assumptions on the surface or curve being open or closed. However, we assume that $\Gamma$ does not contain the boundary points or curves if the curve or surface is open to ensure the existence of a neighborhood where the definition of $d_\Gamma$ is valid. We also note that for closed geometric shapes, the definition of a local neighborhood is not necessary, as the feasible and infeasible regions can be simply defined as the inside and outside of the closed boundary $\Gamma$ (see Fig. 1.1), or vice versa. This is identical to the standard definition of the signed distance function.

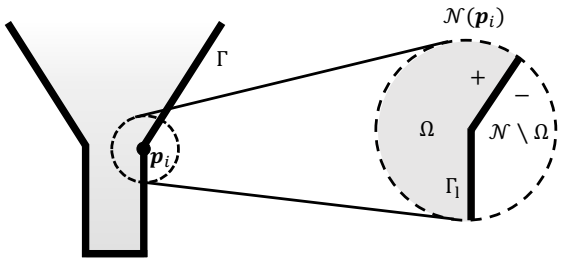


**Figure 3.1.** A 2D open funnel partitions the neighborhood of a point $\mathbf{p}_i$ into a feasible region $\Omega$ and an infeasible region $\mathcal{N} \setminus \Omega$.

## 3.2 B-spline functions

Our desired level set function is a smooth approximation to the signed distance function of a geometric shape and is defined as a mapping $\phi : \mathbb{V} \subset \mathbb{R}^n \to \mathbb{R}$, where $\mathbb{V}$ is the space where we wish to evaluate the level set function as a non-interference constraint function during optimization. This means that the zero level set $S = \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$ implicitly approximates the given geometric shape. To achieve such an approximation, we

utilize tensor product B-splines. A B-spline volume $\mathbf{P} : [0,1]^3 \to \mathbb{R}^3$ is defined as

$$\mathbf{P}(u,v,w) = \sum_{i,j,k=0}^{n_i,n_j,n_k} \mathbf{C}_{i,j,k}\mathbf{B}_{i,d_1}(u)\mathbf{B}_{j,d_2}(v)\mathbf{B}_{k,d_3}(w), \tag{3.2}$$

where $(u,v,w) \in [0,1]^3$ are the normalized parametric coordinates, $\mathbf{B}_{i,d_1}(u)$, $\mathbf{B}_{j,d_2}(v)$, $\mathbf{B}_{k,d_3}(w)$ are the B-spline basis functions of degrees $d_1, d_2, d_3$ in the $i,j,k$ directions respectively, and $\mathbf{C}_{i,j,k}$ are the control points that form the $(n_i+1)\times(n_j+1)\times(n_k+1)$ control net in the physical coordinate system. Equation (3.2) is essentially a tensor product, and hence $\mathbf{P}$ is also called a tensor product B-spline.

A B-spline volume maps a volumetric space in the parametric coordinate system $(u,v,w)$ to the physical coordinate system $(x,y,z)$ by translating and deforming the volumetric space according to the control net $\mathbf{C}_{i,j,k}$. The volumetric space we wish to represent in the physical coordinate system is a rectangular prism $\mathbb{V}$. This is the physical space where geometric non-interference constraints need to be evaluated; therefore, we refer to $\mathbb{V}$ as the domain of interest. We require that $\mathbb{V}$ encompasses the minimum bounding box for a given point cloud. The minimum bounding box is the smallest closed box in $\mathbb{R}^n$ that contains the input point cloud representing a geometric shape. We uniformly space the control points $\mathbf{C}_{i,j,k}$ across $\mathbb{V}$ and consider them to be constant. The domain of interest is not always a cube; therefore, parametric coordinates may be scaled differently along different directions. To compensate for this, some directions may contain more control points than others depending on the dimensions of $\mathbb{V}$.

The B-spline basis functions $\mathbf{B}_{i,d_1}(u)$, $\mathbf{B}_{j,d_2}(v)$, and $\mathbf{B}_{k,d_3}(w)$ are generated by the de Boor's recursion formula [71]. The formulas for all three directions are identical, and

$\mathbf{B}_{i,d_1}(u)$ along the $i$ direction is computed by recursion

$$\mathbf{B}_{i,0}(u) = \begin{cases} 1 & \text{if } t_i \leq u < t_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{3.3}$$
$$\mathbf{B}_{i,k}(u) = \frac{u - t_i}{t_{i+k} - t_i}\mathbf{B}_{i,k-1}(u) + \frac{t_{i+k+1} - u}{t_{i+k+1} - t_{i+1}}\mathbf{B}_{i+1,k-1}(u),$$

where $t_i$ denotes the knots in the $i$ direction. The basis functions corresponding to $\mathbf{C}_{i,j,k}$ provides support only for $(u,v,w) \in [u_i, u_{i+d_1+1}] \times [v_j, v_{j+d_2+1}] \times [w_k, w_{k+d_3+1}]$; thus the basis functions are sparse. This also means that the number of nonzero terms in the summation of Eq. (3.2) is proportional to the degrees $d_1$, $d_2$, and $d_3$.

We define our B-splines with standard uniform knot vectors and uniformly spaced control points $\mathbf{C}_{i,j,k}$ across $\mathbb{V}$. This makes the mapping $(u,v,w) \to (x,y,z)$ a linear, one-to-one relationship with $(u,v,w) \in [0,1]^3$ spanning the entire volumetric space of $\mathbb{V}$. Thus, $\frac{\partial u}{\partial x}$, $\frac{\partial v}{\partial y}$, and $\frac{\partial w}{\partial z}$ are constants that depend only on the dimensions of the rectangular prism $\mathbb{V}$. Since we are using a standard uniform knot vector, it should be noted that the control points must lie beyond $\mathbb{V}$ in order for the mapping to hold for all points in $\mathbb{V}$. With these assumptions, we apply our target function $\phi$ to both sides of Eq. (3.2) to obtain

$$\phi(x,y,z) = \sum_{i,j,k=0}^{n_i,n_j,n_k} \mathbf{C}^{(\phi)}_{i,j,k}\mathbf{B}_{i,d_1}(u(x))\mathbf{B}_{j,d_2}(v(y))\mathbf{B}_{k,d_3}(w(z)), \tag{3.4}$$

where $u(x), v(y)$, and $w(z)$ map the physical coordinates back to normalized parametric coordinates, and $\mathbf{C}^{(\phi)}_{i,j,k}$ are the values of the function $\phi$ at the control points. The derivatives with respect to the spatial coordinates and derivatives with respect to the control points are easily derived from this form. The sparsity of the basis functions over the entire domain and the use of standard uniform knot vectors make the computation of $\phi$ at any given point $(x,y,z)$ in the domain highly efficient. Moreover, $\phi$ becomes continuously differentiable with easily computed higher order derivatives for appropriately defined B-spline degrees.

Note that while computing the level set function $\phi$ for a curve in two dimensions, $\mathbb{V}$ is a rectangle, $\mathbf{P}$ reduces to a B-spline surface, and we omit terms along the $k$ direction in Eq. (3.4). In all of the remaining discussion, we assume $n = 3$ with the geometric shape being a surface in three dimensions.

## 3.3 Energies for B-spline fitting

The core of our methodology lies in computing appropriate $\mathbf{C}^{(\phi)}_{i,j,k}$ values on the control net so that the level set function $\phi$ approximates the signed distance function with favorable properties for gradient-based optimization. Note that since our approximation uses B-splines, the resulting function will already be smooth and fast-to-evaluate. Therefore, the remaining task is to formulate an approach for reliably estimating $\mathbf{C}^{(\phi)}_{i,j,k}$ using the information available from an oriented point cloud.

An oriented point cloud is a set of ordered pairs $\mathcal{P} = \{(\mathbf{p}_i, \vec{\mathbf{n}}_i) : i = 1, \ldots, N_\Gamma\}$, where $\mathbf{p}_i$ are the physical coordinates of the points sampled over the geometric shape, and $\vec{\mathbf{n}}_i$ are the unit surface (or curve) normal vectors at $\mathbf{p}_i$ oriented towards the infeasible region. Our method always requires an oriented point cloud as its input. However, we note that in cases where only a point cloud without normal information is available, Principal Component Analysis (PCA) along with a Minimum Spanning Tree (MST) algorithm can be used for estimating normals and their orientation [42]. Edge-Aware Resampling (EAR) [72] is another method that can be used for generating noise-free normals that also preserves sharp features.

We calculate $\mathbf{C}^{(\phi)}_{i,j,k}$ values by minimizing an energy function consisting of multiple energies. The terms in the energy function are adopted from existing surface reconstruction methods [18, 19, 41, 44]. Since the zero contour of our desired level set function $\phi$ should approximate the geometric shape represented by the point cloud, it is straightforward to see that we should minimize energies to approximately satisfy $\phi(\mathbf{p}_i) = 0$ and $\nabla\phi(\mathbf{p}_i) = -\vec{\mathbf{n}}_i$.

Hence we first define energies

$$\mathcal{E}_p = \frac{1}{N_\Gamma} \sum_{i=1}^{N_\Gamma} \phi(\mathbf{p}_i)^2 \quad \text{and} \tag{3.5}$$

$$\mathcal{E}_n = \frac{1}{N_\Gamma} \sum_{i=1}^{N_\Gamma} \left\| \nabla\phi(\mathbf{p}_i) + \vec{\mathbf{n}}_i \right\|^2, \tag{3.6}$$

where $\mathcal{E}_p$ estimates the approximation error as the average of squared distances of the point cloud from the zero contour of $\phi$, and $\mathcal{E}_n$ measures the average of squared alignment errors of the level set function's gradient when compared to the negative of the unit normal vectors in the point cloud. Note that we take the negative of the normals (oriented toward the infeasible region) from the point cloud since we want distances given by $\phi$ to be positive inside the feasible region. Minimizing $\mathcal{E}_p$ forces the zero contour of $\phi$ to pass through all the points in the point cloud, and minimizing $\mathcal{E}_n$ tries to orient the function's direction of steepest increase $\nabla\phi$ along the normal to the geometric shape while pointing toward the feasible direction, both in the least squares sense. Minimizing $\mathcal{E}_n$ is important since the derivatives of the exact signed distance function $d_\Gamma$ on the boundary of a geometric shape is along the normal to the boundary, and $d_\Gamma$ always satisfies the eikonal equation, i.e., $\|\nabla d_\Gamma\| = 1$.

If we perform a direct minimization of energies $\mathcal{E}_p$ and $\mathcal{E}_n$, the resulting function attempts to accurately fit the point data on the geometric shape, and since these energies do not control the behavior of $\phi$ away from the geometric shape, it could create superfluous zero contours away from the point cloud as reported in previous studies [19]. To overcome this issue, we define the regularization energy

$$\mathcal{E}_r = \frac{1}{|V|} \int_{\mathbb{V}} \left\| \nabla^2\phi(\mathbf{x}) \right\|_F^2 d\mathbb{V}, \tag{3.7}$$

where $\nabla^2\phi(\mathbf{x})$ is the Hessian matrix of $\phi$ evaluated at $\mathbf{x}$, $\|\cdot\|_F$ represents the Frobenius norm, and $|V| = \int_{\mathbb{V}} d\mathbb{V}$ is the total volume of $\mathbb{V}$. The regularization energy $\mathcal{E}_r$ is interpreted

as the aggregate curvature of $\phi$ over the entire volumetric space of $\mathbb{V}$. The minimization of $\mathcal{E}_r$ smooths the function $\phi$ since forcing the Hessian to be zero forces the variations in the gradient field $\nabla\phi$ to a minimum. Since the gradient of $\phi$ is approximately aligned with the unit normals on the point cloud when minimizing $\mathcal{E}_n$, trying to maintain a constant $\nabla\phi$ by minimizing $\mathcal{E}_r$ also helps satisfy the eikonal equation $\|\nabla\phi\| = 1$ for points further away from the point cloud. We evaluate the integral in $\mathcal{E}_r$ using the B-spline control points $\mathbf{C}_{i,j,k}$ lying inside $\mathbb{V}$ as quadrature points with unit quadrature weights. We define the set of quadrature points as $\{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_N\}$. Therefore, the regularization energy is approximated as a discrete sum is given by

$$\mathcal{E}_r = \frac{1}{|V|}\int_{\mathbb{V}} \left\|\nabla^2\phi(\mathbf{x})\right\|_F^2 d\mathbb{V} \approx \frac{1}{N}\sum_{i=1}^{N} \left\|\nabla^2\phi(\mathbf{q}_i)\right\|_F^2, \tag{3.8}$$

where $N$ is total number of quadrature points, typically about the same as the number of control points $N_{cp} = (n_i + 1) \times (n_j + 1) \times (n_k + 1)$.

Some surface reconstruction techniques employ another energy term $\mathcal{E}_d$, which attempts to fit the signed distance function over the entire domain $\mathbb{V}$. However, minimizing this energy was found to create overfitting issues and produce high frequency oscillations in the level set function $\phi$ in our investigation and previous studies [44]. As a result, we neglect this energy in our formulation. Nevertheless, we present it here for the sake of completeness. The signed distance energy is given by

$$\mathcal{E}_d = \frac{1}{N}\sum_{i=1}^{N} \left(\phi(\mathbf{x}_i) - d_\Gamma(\mathbf{x}_i)\right)^2, \tag{3.9}$$

where signed distances $d_\Gamma(\mathbf{x}_i)$ are evaluated at the control points within $\mathbb{V}$ (same as quadrature points in $\mathcal{E}_r$). The signed distances $d_\Gamma(\mathbf{x}_i)$ can be approximated using distances to the nearest neighbor in the point cloud and its normal [44], or by evaluating the explicit equation (2.1). Note that minimizing $\mathcal{E}_r$ can act as a regularization to avoid overfitting

caused by $\mathcal{E}_d$ but careful weighting of the four energies according to the geometric shape is necessary.

## 3.4 Final energy minimization problem

Finally, we define the total energy function $f$ as

$$f = \underbrace{\frac{1}{N_\Gamma}\sum_{i=1}^{N_\Gamma}\phi(\mathbf{p}_i)^2}_{\mathcal{E}_p} + \underbrace{\frac{\lambda_n}{N_\Gamma}\sum_{i=1}^{N_\Gamma}\|\nabla\phi(\mathbf{p}_i)+\vec{\mathbf{n}}_i\|^2}_{\lambda_n\mathcal{E}_n} + \underbrace{\frac{\lambda_r}{N}\sum_{i=1}^{N}\|\nabla^2\phi(\mathbf{q}_i)\|_F^2}_{\lambda_r\mathcal{E}_r}, \tag{3.10}$$

where $\lambda_n$ and $\lambda_r$ are the relative penalization weights for $\mathcal{E}_n$ and $\mathcal{E}_r$ with respect to $\mathcal{E}_p$. The energy minimization problem that yields the desired level set function $\phi$ is then given by

$$\begin{aligned} &\text{minimize} && f = \mathcal{E}_p + \lambda_n\mathcal{E}_n + \lambda_r\mathcal{E}_r \\ &\text{with respect to} && \mathbf{c}_\phi \end{aligned} \tag{3.11}$$

Note that the function values at the control points $\mathbf{C}^{(\phi)}_{i,j,k}$ directly affect $\mathcal{E}_p, \mathcal{E}_n$, and $\mathcal{E}_r$ through the definition of $\phi$ using B-splines (see Eq. (3.4)). If the geometric shape is a curve in two dimensions, then the optimization variables are $\mathbf{C}^{(\phi)}_{i,j}$. The choice of penalization weights is not obvious. Penalization weights may require tuning on a case-by-case basis depending on the geometric shape. In general, we recommend $\lambda_n \sim 10^{-2}$ and $\lambda_r \sim 10^{-4}$ based on the parameter study presented in Sec. 4.

We provide a summary of our methodology in Algorithm 1 for geometric shapes that are surfaces in three dimensions. The algorithm is easily adapted for curves in two dimensions by simply omitting terms along the $k$ direction.

We also provide a visualization of the various grids represented by the non-interference constraint. Figure A.1 visualizes the surface points, control points, quadrature points, minimum bounding box, and domain of interest for a non-interference constraint

**Algorithm 1.** A scalable and differentiable geometric non-interference constraint formulation

1: Discretize the given geometric shape into an oriented point cloud $\mathcal{P}$.
2: Define the domain $\mathbb{V}$ for non-interference constraint evaluation, where $\mathbb{V}$ is a closed box in $\mathbb{R}^3$ and contains the minimum bounding box of $\mathcal{P}$.
3: Select the appropriate numbers of control points $n_i$, $n_j$, and $n_k$ along each direction, and define the corresponding standard uniform knot vectors and spatially uniform control points $\mathbf{C}_{i,j,k}$. Note that some of the control points may lie outside $\mathbb{V}$.
4: Select appropriate weights $\lambda_n$ and $\lambda_r$, and solve the energy minimization problem (3.11) to obtain $\mathbf{C}^{(\phi)}_{i,j,k}$.
5: Evaluate the geometric non-interference constraint(s) during each optimization iteration using the $\phi$ given by Eq. (3.4) and optimized $\mathbf{C}^{(\phi)}_{i,j,k}$ from Step 4.

representing an ellipse.

## 3.5 Implementation details

This section derives the solution to the energy minimization problem (3.11). The energy minimization problem is a well-posed unconstrained quadratic programming (QP) problem. Thus, it may be reduced to the solution of a linear system.

### 3.5.1 Unconstrained quadratic programming problem

The general form of an unconstrained quadratic programming problem with $n$ design variables is

$$\begin{aligned} \text{minimize} \quad & f = \tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} \\ \text{with respect to} \quad & \mathbf{x}, \end{aligned} \tag{3.12}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a the vector of design variables, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the Hessian matrix, and $\mathbf{b} \in \mathbb{R}^n$ is a vector. The two sufficient conditions for a strong local minima for an unconstrained

optimization problem are

$$(1) \quad \nabla f(\mathbf{x}) = 0 \quad \text{and} \tag{3.13}$$

$$(2) \quad \nabla^2 f \succ 0. \tag{3.14}$$

To satisfy condition (1), we take the gradient of the objective function given in problem (3.12) and set it to zero to get

$$\nabla f = \mathbf{A}\mathbf{x} + \mathbf{b} = 0 \tag{3.15}$$

$$\mathbf{A}\mathbf{x} = -\mathbf{b}. \tag{3.16}$$

A unique solution to Eq. (3.16) exists if and only if $\mathbf{A}$ is invertible. If we assume that sufficient condition (2) is satisfied by $\mathbf{A} \succ 0$, then $\mathbf{A}$ is invertible. The global minimum to problem (3.12) is given by the solution $\mathbf{x}^* = -\mathbf{A}^{-1}\mathbf{b}$ that is a unique solution to the linear system of equations defined in Eq. (3.16).

### 3.5.2 Derivation to the linear system for energy minimization

The following derivation will be for a B-spline volume representing a 3D geometric shape in problem (3.11). The derivation is generic, and one may ignore the $k$-direction to acquire a solution for a B-spline surface that represents a 2D geometric shape.

Given an oriented point cloud containing a set of ordered pairs of points and normal vectors $\mathcal{P} = \{(\mathbf{p}_i, \vec{\mathbf{n}}_i) : i = 1, 2, \ldots, N_\Gamma\}$, we define a B-spline volume $\phi$ with a domain no smaller than the minimum bounding box of the point cloud, uniformly spaced control points, and standard uniform knot vectors. We define a vector $\mathbf{c}_\phi \in \mathbb{R}^{N_{cp}}$ that contains the full set of B-spline control point values $\mathbf{C}^{(\phi)}_{i,j,k}$ in the B-spline $\phi$. In order to solve problem (3.11), a vectorized approach is taken to evaluate the B-spline and calculate the objective function.

Given the set of points in the point cloud $\mathcal{P}$, the vectorized evaluation of $\phi(\mathbf{p})$ and its gradients $\nabla\phi(\mathbf{p})$ are matrix-vector multiplications, where the basis matrix is precomputed as a function of the point cloud $\mathcal{P}$ and the vector is the control point values $\mathbf{c}_\phi$. The basis functions are given by Eq. (3.3), which may be evaluated after setting up the B-spline knot vectors and parametric coordinate transformation for the points in the point cloud. The vectorized equations for $\phi(\mathbf{p})$ and $\nabla\phi(\mathbf{p})$ in three dimensions are given by

$$\phi(\mathbf{p}) = \mathbf{A}_0\mathbf{c}_\phi \tag{3.17}$$

$$\frac{d\phi}{dx}(\mathbf{p}) = \mathbf{A}_x\mathbf{c}_\phi \quad \frac{d\phi}{dy}(\mathbf{p}) = \mathbf{A}_y\mathbf{c}_\phi \quad \frac{d\phi}{dz}(\mathbf{p}) = \mathbf{A}_z\mathbf{c}_\phi, \tag{3.18}$$

where $\mathbf{A}_0 \in \mathbb{R}^{N_\Gamma \times N_{cp}}$, $\mathbf{A}_x \in \mathbb{R}^{N_\Gamma \times N_{cp}}$, $\mathbf{A}_y \in \mathbb{R}^{N_\Gamma \times N_{cp}}$, and $\mathbf{A}_z \in \mathbb{R}^{N_\Gamma \times N_{cp}}$ are the precomputed bases with respect to the points in the point cloud $\mathbf{p}$. Given the set of quadrature points $\mathcal{Q} = \{\mathbf{q}_i : i = 1, 2, \ldots, N\}$, which lie on the control points within the domain of interest, the evaluation of $\nabla^2\phi(\mathbf{q})$ are matrix-vector multiplications, where the basis matrices are precomputed as a function of the quadrature points $\mathbf{q}$, and the vector is the control point values $\mathbf{c}_\phi$. The vectorized equations for $\nabla^2\phi(\mathbf{q})$ in three dimensions are given by

$$\begin{aligned}
\frac{d^2\phi}{dx^2}(\mathbf{q}) &= \mathbf{A}_{xx}\mathbf{c}_\phi & \frac{d^2\phi}{dxdy}(\mathbf{q}) &= \mathbf{A}_{xy}\mathbf{c}_\phi \\
\frac{d^2\phi}{dy^2}(\mathbf{q}) &= \mathbf{A}_{yy}\mathbf{c}_\phi & \frac{d^2\phi}{dxdz}(\mathbf{q}) &= \mathbf{A}_{xz}\mathbf{c}_\phi \\
\frac{d^2\phi}{dz^2}(\mathbf{q}) &= \mathbf{A}_{zz}\mathbf{c}_\phi & \frac{d^2\phi}{dydz}(\mathbf{q}) &= \mathbf{A}_{yz}\mathbf{c}_\phi,
\end{aligned} \tag{3.19}$$

where $\mathbf{A}_{xx} \in \mathbb{R}^{N \times N_{cp}}$, $\mathbf{A}_{xy} \in \mathbb{R}^{N \times N_{cp}}$, $\mathbf{A}_{yy} \in \mathbb{R}^{N \times N_{cp}}$, $\mathbf{A}_{yz} \in \mathbb{R}^{N \times N_{cp}}$, and $\mathbf{A}_{zz} \in \mathbb{R}^{N \times N_{cp}}$ are the precomputed bases with respect to the quadrature points $\mathbf{q}$.

The energy terms are now computed by substituting Eqs. (3.17, 3.18, 3.19) into the corresponding energy terms defined in Eqs. (3.5, 3.6, 3.7). The vectorized representation

of $\mathcal{E}_p$ is given by

$$\mathcal{E}_p = \frac{1}{N_\Gamma}\mathbf{c}_\phi^T \mathbf{A}_p \mathbf{c}_\phi, \tag{3.20}$$

where $\mathbf{A}_p = \mathbf{A}_0^T \mathbf{A}_0 \in \mathbb{R}^{N_{cp} \times N_{cp}}$ is the quadratic contribution of $\mathcal{E}_p$. The representation of $\mathcal{E}_n$ uses the following equality that

$$\|\nabla\phi(\mathbf{p}_i) + \vec{\mathbf{n}}_i\|^2 = \|\nabla\phi(\mathbf{p}_i)\|^2 + 2\vec{\mathbf{n}}_i^T \nabla\phi(\mathbf{p}_i) + \|\vec{\mathbf{n}}_i\|^2. \tag{3.21}$$

Note that $\|\vec{\mathbf{n}}_i\|^2$ is a constant and it may be ignored when formulating it into the energy term because it does not change with respect to the B-spline control point values $\mathbf{c}_\phi$. Using the equality in Eq. (3.21), the vectorized representation of $\mathcal{E}_n$ is given by

$$\mathcal{E}_n = \frac{1}{N_\Gamma}\mathbf{c}_\phi^T \mathbf{A}_n \mathbf{c}_\phi + \frac{1}{N_\Gamma}\left(2\mathbf{n}_x^T \mathbf{A}_x + 2\mathbf{n}_y^T \mathbf{A}_y + 2\mathbf{n}_z^T \mathbf{A}_z\right)\mathbf{c}_\phi, \tag{3.22}$$

where $\mathbf{A}_n = \mathbf{A}_x^T \mathbf{A}_x + \mathbf{A}_y^T \mathbf{A}_y + \mathbf{A}_z^T \mathbf{A}_z \in \mathbb{R}^{N_{cp} \times N_{cp}}$ is the quadratic contribution of $\mathcal{E}_n$, $\mathbf{n}_x \in \mathbb{R}^{N_\Gamma}$, $\mathbf{n}_y \in \mathbb{R}^{N_\Gamma}$, and $\mathbf{n}_z \in \mathbb{R}^{N_\Gamma}$ are the vectorized components of the normal vectors pointing in the $x$, $y$, and $z$ directions of the coordinate frame, respectively. The vectorized representation of $\mathcal{E}_r$ is given by

$$\mathcal{E}_r = \frac{1}{N}\mathbf{c}_\phi^T \mathbf{A}_r \mathbf{c}_\phi, \tag{3.23}$$

where $\mathbf{A}_r = \mathbf{A}_{xx}^T \mathbf{A}_{xx} + 2\mathbf{A}_{xy}^T \mathbf{A}_{xy} + \mathbf{A}_{yy}^T \mathbf{A}_{yy} + 2\mathbf{A}_{xz}^T \mathbf{A}_{xz} + 2\mathbf{A}_{yz}^T \mathbf{A}_{yz} + \mathbf{A}_{zz}^T \mathbf{A}_{zz} \in \mathbb{R}^{N_{cp} \times N_{cp}}$ is the quadratic contribution of $\mathcal{E}_r$.

Assembling all energy terms by substituting Eqs. (3.20, 3.22, 3.23) into Eq. (3.10) and dividing by 2, we have

$$f = \frac{1}{2}\mathbf{c}_\phi^T \underbrace{\left(\frac{1}{N_\Gamma}\mathbf{A}_p + \frac{\lambda_n}{N_\Gamma}\mathbf{A}_n + \frac{\lambda_r}{N}\mathbf{A}_r\right)}_{\tilde{\mathbf{A}}} \mathbf{c}_\phi + \underbrace{\frac{\lambda_n}{N_\Gamma}\left(\mathbf{n}_x^T \mathbf{A}_x + \mathbf{n}_y^T \mathbf{A}_y + \mathbf{n}_z^T \mathbf{A}_z\right)}_{\tilde{\mathbf{b}}^T} \mathbf{c}_\phi, \tag{3.24}$$

where $\lambda_n > 0$ and $\lambda_r > 0$ are the penalization weights. The form of Eq. (3.24) fits the generic form of Eq. (3.12) as

$$f = \frac{1}{2}\mathbf{c}_\phi^T \tilde{\mathbf{A}} \mathbf{c}_\phi + \tilde{\mathbf{b}}^T \mathbf{c}_\phi. \tag{3.25}$$

Therefore, the energy minimization problem reduces to an unconstrained quadratic programming problem. In a similar way as Eq. (3.16), satisfying sufficient condition (1) given the objective function in Eq. (3.25) results in solving the linear system

$$\tilde{\mathbf{A}} \mathbf{c}_\phi = -\tilde{\mathbf{b}}, \tag{3.26}$$

where $\tilde{\mathbf{A}} \in \mathbb{R}^{N_{cp} \times N_{cp}}$ is the collective Hessian matrix and $\tilde{\mathbf{b}}$ is the right hand side vector. Note that $\tilde{\mathbf{b}}$ only depends on $\mathcal{E}_n$ and that $\tilde{\mathbf{A}}$ will always be a symmetric positive definite square matrix, satisfying sufficient condition (2). The informal proof that $\tilde{\mathbf{A}} \succ 0$ is described here. The matrix $\tilde{\mathbf{A}}$ is guaranteed to be symmetric positive semi-definite because all precomputed basis matrices are left multiplied by their transposes and comprise the terms of the weighted sum to $\tilde{\mathbf{A}}$. Additionally, the $\mathbf{A}_r$ matrix will always have positive entries along its diagonal because every control point in the domain of interest has a corresponding quadrature point in $\mathcal{Q}$ for which it has a nonzero positive weight. The diagonals along $\mathbf{A}_p$ and $\mathbf{A}_n$ are greater than or equal to zero. Therefore, $\tilde{\mathbf{A}}$ will have positive values along its diagonal. As shown, $\tilde{\mathbf{A}}$ is positive semi-definite and it has nonzero positive entries along the diagonal. Because all diagonal entries are positive and nonzero, the determinant of $\tilde{\mathbf{A}}$ is positive, and therefore, $\tilde{\mathbf{A}}$ is positive definite, invertible, and satisfies sufficient condition (2). The sparsity of $\tilde{\mathbf{A}}$ is visualized in Fig. 3.2.

In order to solve Eq. (3.26), an iterative solver is recommended. For most applications of the geometric non-interference constraint, $N_{cp} \sim 10^5$ or greater is required to accurately represent most basic objects. Because the size of $\tilde{\mathbf{A}}$ scales with $\mathcal{O}(N_{cp}^2)$, it is not practical to compute its inverse directly. Instead, we elect to use a conjugate gradient (CG)

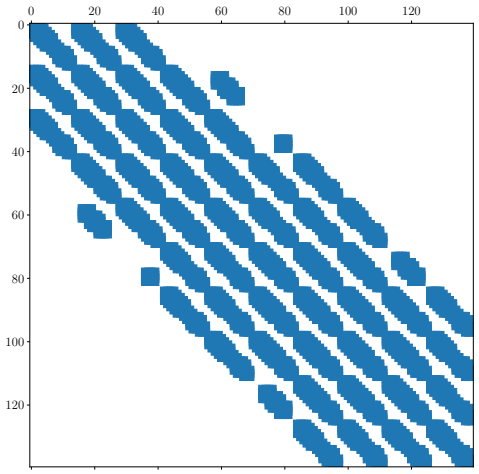


**Figure 3.2.** The sparsity of $\tilde{\mathbf{A}}$ for a simple problem.

solver, which takes advantage of the symmetry of $\tilde{\mathbf{A}}$. The computational speed of the CG solver is typically on the order of seconds, as demonstrated in the results in Chapter 4.

This chapter, in part, is currently being prepared for submission for publication of the material. The authors of this work are Ryan C. Dunn, Anugrah Jo Joshy, Jui-Te Lin, Cédric Girerd, Tania K. Morimoto, and John T. Hwang. The thesis author was the primary investigator and author of this material.

# Chapter 4

# Results of the Novel Geometric Non-interference Formulation

This chapter presents the results of various numerical studies using our formulation. We begin by studying our method using simple two-dimensional geometric shapes in Section 4.1. In Section 4.2, we investigate the dependence of our method on various weights using the Stanford Bunny dataset. Section 4.3 then compares our method to previous non-interference constraint formulations using the Stanford Bunny, and other surface reconstruction methods using three datasets from the Stanford 3D Scanning Repository. We demonstrate the accuracy of our method in representing geometric shapes for aircraft design optimization in Section 4.4. We conclude the section by demonstrating the application of our method by solving a medical robot design optimization problem with non-interference constraints in Section 4.5.

We implement the proposed method in a Python environment, and run all experiments on a desktop with an 8-core Ryzen 7 @ 3.6 GHz processor and 32 GB of RAM. We do not implement multi-threading or parallelization with GPUs in any of our numerical experiments.

## 4.1 Investigations using simple geometric shapes in two dimensions

We begin by applying our formulation to curves in two dimensions. Because our formulation is generic, no modifications are required to Eq. (3.10), and the terms in the $k$ direction are simply ignored for 2D geometric shapes.

For 2D curves, the isocontours from the level set function (LSF) $\phi$ can be readily visualized to facilitate a better understanding of the function both near and far from the curve. Figure 4.1 visualizes the isocontours of the initialized and energy minimized LSF for a rectangle using our formulation. We initialize $\mathbf{C}_{i,j}^{(\phi)}$ using the explicit equation (2.1). Neglecting the sharp corners in this example, the contours of the initialized function closely match the exact signed distance function (SDF). Thus, the explicit method provides an excellent approximation of the SDF. We observe that the contours of the LSF are more rounded near the corners after energy minimization. Minimizing $\mathcal{E}_r$ smooths sharp corners on all isocontours, however, not to a degree that compromises $\mathcal{E}_n$ and $\mathcal{E}_p$ near the zero contour. We note that $\mathcal{E}_n$ and $\mathcal{E}_p$ have less influence compared to $\mathcal{E}_r$ on the isocontours corresponding to 1 and 2, hence these contours are even more rounded.

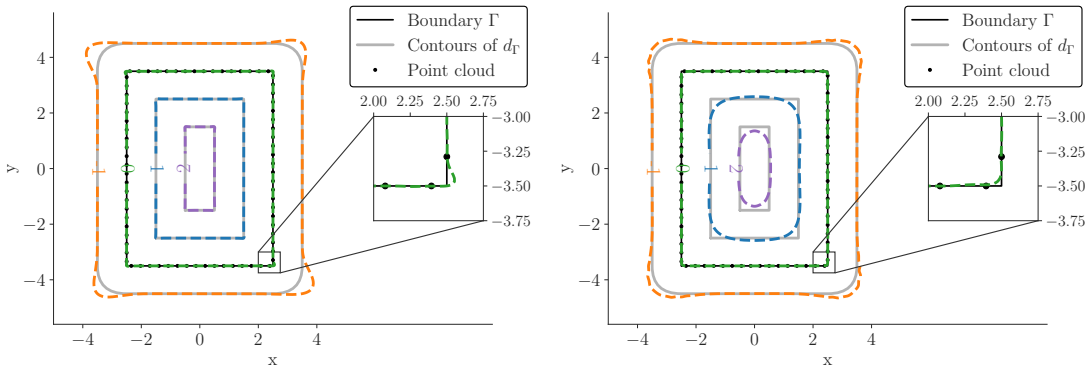


**Figure 4.1.** The −1, 0, 1, and 2 contours of the initialized (left) and the energy minimized (right) level set function $\phi$.

A LSF representing multiple geometric shapes may also be obtained using a single B-spline. Figure 4.2 shows the exact SDF and a one-dimensional slice of our energy

minimized LSF $\phi$ along the $x$ axis for a domain containing multiple circles. Notice that the non-differentiable points in the exact SDF can lie inside or outside of a geometric shape. This example illustrates how our energy minimization formulation balances the tradeoff between minimizing the curvature of $\phi$ and maximizing the accuracy at representing the SDF near points of non-differentiability and high curvature. We see that our LSF $\phi$ poorly approximates the SDF near points of non-differentiability and high curvature. However, in regions without any non-differentiabilities or high curvatures, the zero level set preserves a good approximation to the exact SDF. For the remainder of the numerical results section, we only consider a single geometric shape within the domain of interest $\mathbb{V}$, because the error of our formulation increases with the minimum bounding box diagonal.

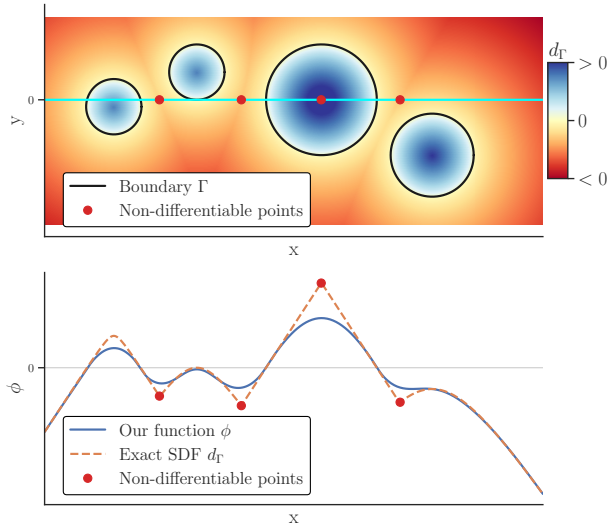


**Figure 4.2.** The exact SDF (top) and the energy minimized LSF $\phi$ along a 1D slice (bottom) for multiple geometric shapes.

## 4.2 Investigations using a complex geometric shape in three dimensions

We use the well known Stanford Bunny scanned dataset (shown in Fig. 4.3) to analyze our formulation's performance on three-dimensional geometric shapes. This dataset contains a large oriented point set which we consider as an exact surface. We coarsely

sample this point set and apply our method, measuring the accuracy of the resultant energy minimized LSF. The Stanford Bunny contains small scale features, sharp corners, flat surfaces, and smooth surfaces which will test the accuracy of our formulation in representing different geometric features. The sampled point set is free of noise, missing data, and nonuniformity, which are challenges not investigated in this paper.

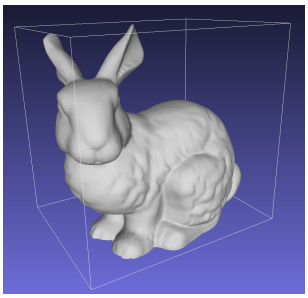

**Figure 4.3.** The Stanford Bunny model.

We use the root-mean-squared (RMS) error and max error to evaluate the accuracy of our energy minimized LSF in approximating the signed distance function. The errors are normalized by the minimum bounding box diagonal $L$ to ensure that they are independent of the size of a geometric shape. This allows for a common metric for comparing accuracies across different geometric shapes. The errors are defined as

$$\text{RMS error} = \frac{1}{L}\sqrt{\frac{\sum_{i=1}^{N_e}(\phi(\mathbf{x}_i) - d_\Gamma(\mathbf{x}_i))^2}{N_e}}, \text{ and} \tag{4.1}$$

$$\text{max error} = \max_{i=1,2,\ldots,N_e} \frac{1}{L}|\phi(\mathbf{x}_i) - \mathrm{d}_\Gamma(\mathbf{x}_i)|, \tag{4.2}$$

where $N_e$ is the number of points $\mathbf{x}_i$ used to calculate the error, and the signed distances are approximated using the explicit equation (2.1). The on-surface error is evaluated on the geometric shape $\Gamma$ where the true value is zero. The off-surface error is evaluated on points that are near but do not lie on $\Gamma$. To acquire these sample points, we take the original sample points and move them in the direction of the normal vectors by specified distances.

The energy minimization problem has two different resolution scales: (1) resolution of the input dataset, and (2) resolution of the B-spline control grid. The energy minimized LSF's ability to approximate the signed distance is best when both resolutions are very fine. Unlike hierarchical structures or explicit methods, the control grid resolution for our method is independent of the input dataset resolution. As a result, an important consideration in our method is the decision of how many control points with which to represent the implicit function. Table 4.1 tabulates the on-surface errors for our energy minimized function, varying the two resolution scales. A total of nine experiments were ran, using three different resolutions for each resolution scale. It was found that only increasing $N_\Gamma$ was not correlated to any increase in the computation time. Thus, we only consider the average time to solve across the three experiments with the same $N_{cp}$. In terms of the on-surface error, increasing both resolutions correlates to a decrease in both RMS and maximum error. In terms of the time to solve the energy minimization problem (setup time), increasing the number of control points $N_{cp}$ increases the number of design variables in the minimization problem (3.11), consequently increasing the time to solve for $\mathbf{C}^{(\phi)}_{i,j,k}$.

**Table 4.1.** The relative on-surface error for our method applied to the Stanford Bunny model. Penalization weights used were $\lambda_n = 10^{-2}$, and $\lambda_r = 5 \times 10^{-4}$.

| | | $32 \times 31 \times 26$ $N_{cp} = 25,792$ | $40 \times 39 \times 32$ $N_{cp} = 49,920$ | $46 \times 45 \times 37$ $N_{cp} = 76,590$ |
|---|---|---|---|---|
| $N_\Gamma = 5$k | RMS | $1.0 \times 10^{-3}$ | $7.5 \times 10^{-4}$ | $6.7 \times 10^{-4}$ |
| | Max | $4.7 \times 10^{-3}$ | $3.7 \times 10^{-3}$ | $3.3 \times 10^{-3}$ |
| $N_\Gamma = 25$k | RMS | $8.2 \times 10^{-4}$ | $5.4 \times 10^{-4}$ | $4.6 \times 10^{-4}$ |
| | Max | $3.7 \times 10^{-3}$ | $3.3 \times 10^{-3}$ | $2.7 \times 10^{-3}$ |
| $N_\Gamma = 64$k | RMS | $7.8 \times 10^{-4}$ | $5.1 \times 10^{-4}$ | $4.2 \times 10^{-4}$ |
| | Max | $6.1 \times 10^{-3}$ | $5.1 \times 10^{-3}$ | $2.9 \times 10^{-3}$ |
| Average setup time (seconds) | | 4.79 | 8.90 | 15.87 |

While we do not propose an exact method for selecting the penalization weights $\lambda_n$

and $\lambda_r$, we provide a fixed point parameter study on each weight. Figure 4.4 shows the resulting errors from varying each penalization weight about the fixed point $\lambda_n = \lambda_r = 1$ using the Stanford Bunny dataset. The study on $\lambda_r$ shows that small values ($\lambda_r < 1$) have very little effect on the RMS error, and large values ($\lambda_r > 1$) significantly reduce the energy minimized function's accuracy. The study on $\lambda_n$ suggests that for a given geometric shape, there exists an optimum value of $\lambda_n$ that minimizes the energy minimized function's error. In all studies, we observed an increase in the minimization time as the corresponding weight increased (not visualized in the figure). These observations lead us to recommend the use of $\lambda_n \sim 10^{-2}$ and $\lambda_r \sim 10^{-4}$ for reasonable accuracy and fast setup time for the energy minimized function.

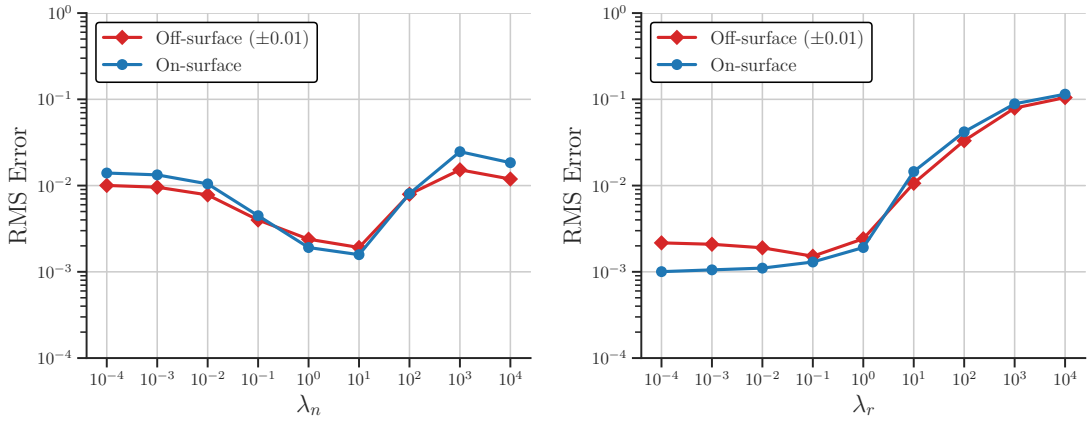


**Figure 4.4.** The RMS errors for on- and off-surface points while varying $\lambda_n$ and $\lambda_r$ about 1. This result uses the Stanford Bunny sampled at $N_\Gamma = 25,000$, and a control point grid of $31 \times 31 \times 26$.

The ability of our function to represent nonzero level sets of the Stanford Bunny is visualized in Fig. 4.5. The level sets form good approximations of the offset surfaces, with a maximum relative distance error of $9.8 \times 10^{-3}$. In these visualizations, we observe the region of highest error to be near the neck and feet of the model, where edges and corners exist. Most notably, the 0.005 and 0.01 level sets remove the ears of the model, despite them being in the exact SDF representation. As a thin feature, the removal of the ears in the 0.005 and 0.01 level sets is consistent with similar observations by Tang and Feng [44].

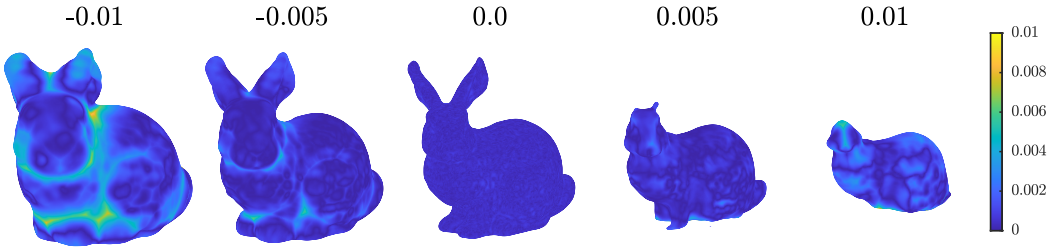


**Figure 4.5.** Isocontours of the energy minimized LSF for the Stanford Bunny and color representation of the error with the true signed distance value normalized by the minimum bounding box diagonal.

## 4.3 Comparison to other methods using complex geometric shapes in three dimensions

We show the computation time and accuracy of our method compared to alternative non-interference constraint formulations for gradient-based methods in Fig. 4.6, varying the sample size $N_\Gamma$ of the Stanford Bunny. We observe that the method presented by Lin et al. [2] and the explicit method presented by Hicken and Kaur [46] scale in computational complexity with $\mathcal{O}(N_\Gamma)$, while our method scales independently of $N_\Gamma$. We note that formulation from Lin et al. is not an attempt at approximating the signed distance function, thus is neglected from the RMS error comparisons. In terms of on-surface error, the explicit method has a steady decay in RMS error with respect to increasing $N_\Gamma$, suggesting a power law relationship. Our method has a similar decay up to $N_\Gamma = 10^4$, where the RMS error decays more slowly for larger $N_\Gamma > 10^4$. Similarly, the off-surface RMS error of the explicit method steadily decays for both the $\pm 0.005$ and $\pm 0.01$ contours, and our method decays until $N_\Gamma = 10^4$. For $N_\Gamma > 10^4$, the off-surface error of our method decays slowly. Our method's $\pm 0.01$ contours have significantly more error than the $\pm 0.005$ contours, while the explicit method has similar error for both sets of isocontours. For both on-surface and off-surface error, our method performs better in terms of accuracy up until the $\pm 0.005$ contours and $N_\Gamma < 2 \times 10^4$. From this information, we conclude that the explicit method will outperform in terms of accuracy and underperform in terms evaluation time compared

to our method for most very finely sampled geometries. We note that our method can achieve better accuracy than shown in Fig. 4.6 by a tradeoff in setup time as shown in Table 4.1, and the explicit method requires a noise-free, uniform sampling to achieve the presented results, which is not always feasible.

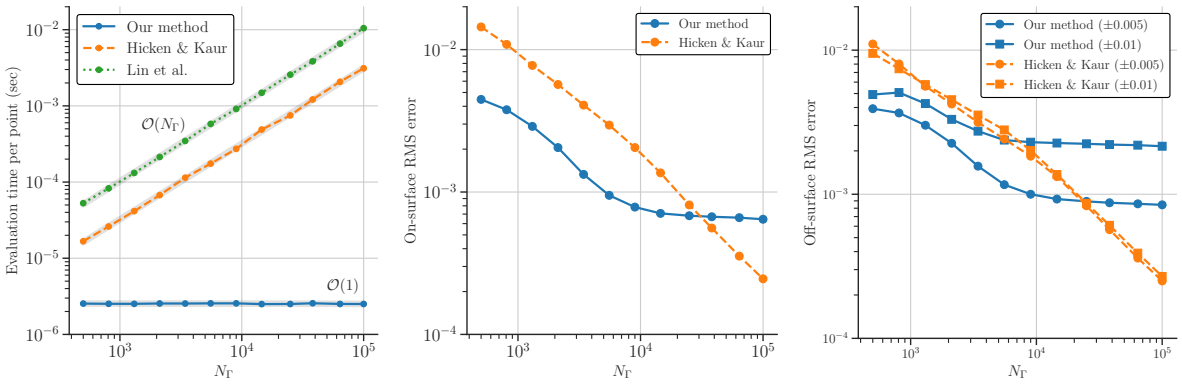


**Figure 4.6.** Scaling of computation time (left) and on-surface (center) and off-surface root-mean-square error (right) with respect to $N_\Gamma$ for the Bunny model with a grid of $(31 \times 31 \times 26)$ and $\lambda_n = 10^{-2}, \lambda_r = 5 \times 10^{-4}$.

We apply our method using two additional scanned datasets from the Stanford 3D Scanning Repository. Table 4.2 records the results of the on-surface error of our method, as well as the reported on-surface error from four notable surface reconstruction methods for necessary context. The methods are smooth signed distance (SSD) reconstruction [19], Multi-Level Partition of Unity (MPU) [18], wavelets [50], and screened Poisson (SP) [54]. The results for the surface reconstruction methods were obtained from [18, 41, 44, 54] and were not reproduced in our investigation, resulting in missing data in the table. Of the three scanned datasets, our energy minimized LSF maintains on-surface RMS error and max error on the same order of magnitude compared to the four other methods.

**Table 4.2.** Reported on-surface error of surface reconstruction methods and our method for three benchmarking datasets. We reconstruct using $N_{cp} = 25{,}000$, $\lambda_n = 10^{-2}$, and $\lambda_r = 5 \times 10^{-4}$.

| Model | | SSD [19] | MPU [18] | Wavelets [50] | SP [54] | Our method |
|---|---|---|---|---|---|---|
| Bunny | RMS | $8.0 \times 10^{-4}$ | $1.0 \times 10^{-3}$ | $1.1 \times 10^{-3}$ | $8.0 \times 10^{-4}$ | $6.3 \times 10^{-4}$ |
| | Max | ... | ... | ... | ... | $4.5 \times 10^{-3}$ |
| Armadillo | RMS | $3.0 \times 10^{-4}$ | ... | $1.2 \times 10^{-3}$ | $4.0 \times 10^{-4}$ | $1.3 \times 10^{-3}$ |
| | Max | $9.0 \times 10^{-4}$ | $1.9 \times 10^{-3}$ | $2.0 \times 10^{-3}$ | $8.0 \times 10^{-4}$ | $7.6 \times 10^{-3}$ |
| Dragon | RMS | $3.5 \times 10^{-4}$ | $8.0 \times 10^{-4}$ | $1.4 \times 10^{-3}$ | $5.1 \times 10^{-4}$ | $1.4 \times 10^{-3}$ |
| | Max | ... | $4.8 \times 10^{-3}$ | ... | $5.1 \times 10^{-3}$ | $1.0 \times 10^{-2}$ |

## 4.4 Accuracy of our method for aircraft design optimization

We now apply our formulation to a number of geometric shapes involved in novel aircraft design. Aircraft design optimization is a long standing problem and has been the subject of recent interest in problems involving geometric non-interference constraints, e.g., the layout optimization of air cargo [10] and aerodynamic shape optimization [3]. To enable gradient-based design optimization involving these constraints, a new generic method is required to represent numerous components within an aircraft's design. We recognize the potential for our formulation and demonstrate its capabilities by conducting an experiment.

In this experiment, we apply our formulation and quantify the resultant errors of five geometric shapes commonly associated with aircraft design. The geometries we model include a fuselage and a wing from a novel electric vertical take-off and landing (eVTOL) concept vehicle [73], a human avatar [74], a luggage case, and a rectangular prism representing a battery pack within the wing. A visualization of these components in a feasible design configuration is illustrated in Fig. 4.7.

Table 4.3 tabulates the on-surface error of the energy minimized LSF for each geometry. We observe that the smallest relative on-surface error is of the smooth fuselage

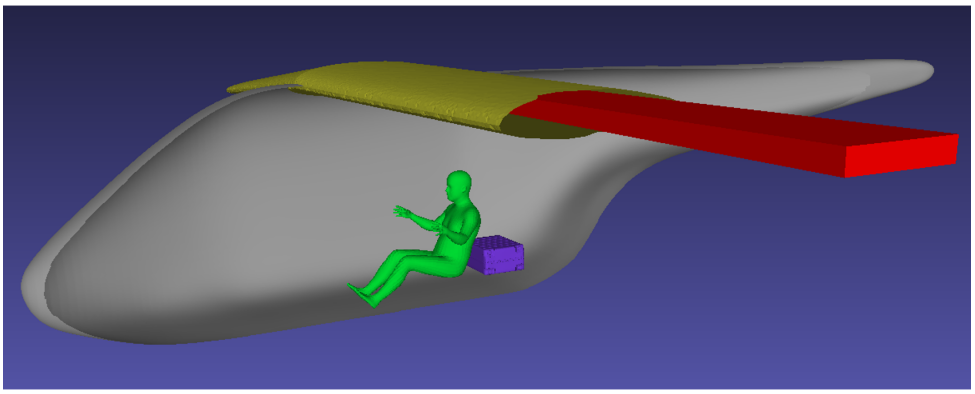

**Figure 4.7.** Components necessary for spatial integration in aircraft design optimization. A cross section view is shown for an aircraft fuselage, wing, battery pack, human avatar, and luggage.

shape, while the largest relative error is of the human avatar. We note from this example that geometries with features reasonably proportioned to their minimum bounding box diagonal are easier to represent using our method, hence our formulation poorly represents small scale features (e.g. hands and feet) of the human avatar while it can represent the smooth fuselage very well. We observe that the bounding boxes of the fuselage, wing, and battery pack are poorly proportioned, yet they do not result in an increase of relative error compared to other geometries. However, their longer minimum bounding box diagonals will result in larger absolute errors.

**Table 4.3.** Relative on-surface error of our method on various components of engineering systems. All geometries were sampled at $N_\Gamma = 25,000$. The optimization weights $\lambda_n = 10^{-2}$ and $\lambda_r = 5 \times 10^{-4}$ were used.

| | | Fuselage | Luggage | Wing | Battery pack | Human avatar |
|---|---|---|---|---|---|---|
| Relative error | RMS | $7.6 \times 10^{-5}$ | $1.8 \times 10^{-4}$ | $2.5 \times 10^{-4}$ | $4.4 \times 10^{-4}$ | $7.8 \times 10^{-4}$ |
| | Max | $5.7 \times 10^{-4}$ | $1.8 \times 10^{-3}$ | $1.0 \times 10^{-3}$ | $1.8 \times 10^{-3}$ | $3.8 \times 10^{-3}$ |
| Absolute error | RMS | 0.24 cm | 0.01 cm | 1.28 cm | 0.66 cm | 0.14 cm |
| | Max | 1.78 cm | 0.15 cm | 5.29 cm | 2.61 cm | 0.67 cm |
| Discretization | | $47 \times 29 \times 29$ | $32 \times 44 \times 26$ | $29 \times 47 \times 29$ | $28 \times 47 \times 28$ | $37 \times 32 \times 34$ |
| Setup time (seconds) | | 6.7 | 13.5 | 39.4 | 15.6 | 21.3 |

## 4.5 Application to a medical robot design problem

We now apply our method for enforcing geometric non-interference constraints to a medical robot design problem involving concentric tube robots (CTRs). CTRs are composed of two or more long and slender pre-curved tubes made of superelastic materials. They can be designed to reach points in a large region of interest by rotating and translating the tubes relative to each other at their bases. These characteristics make them ideal for minimally invasive surgeries where a surgeon can operate on a small region of interest with high dexterity through actuation at the base.

In the foundational works of Sears and Dupont [75] and Webster et al. [76], expressions for the shape and tip position of the CTR are derived with respect to the robot's geometric and control variables. Bergeles et al. [1] use these expressions to perform gradient-free optimization of the CTR's geometric and control variables with anatomical constraints. These anatomical constraints, i.e., geometric non-interference constraints, enforce that the CTR does not interfere with the anatomy (e.g., the right ventricle of the heart shown in Fig. 4.8) during operation. Recent work by Lin et al. [2] shows that gradient-based optimization enables an efficient and scalable solution to simultaneously optimize the large set of the tube's geometric and control variables while enforcing anatomical constraints. The experiment we now present follows the workflow of Lin et al. [2], however, using our new formulation for representing the anatomical constraint function.

The presented workflow involves the solution of multiple optimization problems, including an initial path planning problem, and the geometric design and control of the CTR (the 'simultaneous optimization problem' described by Lin et al. [2]). The path planning problem solves for a parametric 3D curve that represents an optimal collision-free path to the surgical site within the anatomy. Then, points along this path serve as inputs to the geometric design and control optimization of the CTR, which involve a kinematic model of the robot. In both subproblems, the non-interference constraints are enforced

by evaluating a discrete set of points along the path or physical CTR to ensure that no points lie outside of the anatomy.

We begin our experiment with an investigation in the heart anatomy which represents the non-interference constraint of the problem. The initial oriented point cloud of the heart is obtained from segmentation and 3D reconstruction by magnetic resonance imaging (MRI) scans. Due to the limited machine accuracy, error introduced by aligning multiple scans, and normal approximation, the oriented point cloud is noisy, nonuniform, and contains poorly oriented normals. We perform a simple and necessary smoothing step on this point cloud as illustrated in Fig. 4.8. Although less precise at capturing small scale features, the smoothing step assists our method in reconstructing a smooth zero contour for constraint representation.

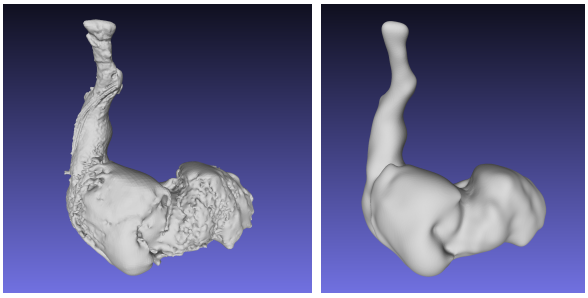

**Figure 4.8.** Preprocessing the raw scanned data (left) to a smooth approximate model (right).

The smooth representation has relative errors $3.1 \times 10^{-3}$ (RMS) and $1.9 \times 10^{-2}$ (max) compared to the original noisy representation. The error in our energy minimized function obtained from the smoothed heart model is tabulated in Table 4.4. We observe that the on-surface RMS and max error of our representation is an order of magnitude less than the error introduced by the smoothing step. This implies that our representation of the smooth model is no worse than the smoothing step itself. We see that our method generates a function with a reliable zero level set of the smooth heart geometry, with an on-surface RMS error of $2.1 \times 10^{-4}$. This error is lower compared to all the other examples in Table 4.2, and we attribute this to the smoothness of the heart geometry. We also note

that the max on- and off-surface absolute errors of our representation are of the same order as the diameter of the CTR itself, typically 0.5-2.0 mm.

**Table 4.4.** Error for the non-interference constraint with a minimum bounding box diagonal of 244.75 mm, sampled at $N_\Gamma = 100,000$, and a control grid of $(28 \times 23 \times 37)$, using $\lambda_n = 10^{-2}$, and $\lambda_r = 10^{-4}$.

| | | Relative error | Absolute error |
|---|---|---|---|
| On-surface | RMS | $2.1 \times 10^{-4}$ | 0.053 mm |
| | Max | $1.8 \times 10^{-3}$ | 0.432 mm |
| 5mm Off-surface | RMS | $3.1 \times 10^{-3}$ | 0.758 mm |
| | Max | $4.3 \times 10^{-3}$ | 1.058 mm |

We now solve the two optimization subproblems using our energy minimized LSF of the smoothed heart model to enforce the geometric non-interference constraint. In the model from Lin et al. [2], the non-interference constraint was imposed using a penalization function $g(\mathbf{x})$, where it was defined as negative for the feasible region, and positive for the infeasible region. In our implementation, we represent this function with our energy minimized LSF in the form $g(\mathbf{x}) = -\phi(\mathbf{x})$. The results from this experiment are shown in Table 4.5, where the number of function evaluations and optimization time are tabulated for each subproblem and non-interference constraint method. The time to solve the energy minimization problem for our method is denoted as the setup time. Between the two subproblems, the number of function evaluations and optimization time is significantly more for the design subproblem due to the inclusion of the kinematics models. Between the two non-interference constraint methods, we observe a significant decrease in optimization time by using our new method for both subproblems. Even when accounting for the setup time, our method provides a significant speedup for the design subproblem. However, we note that the speedup provided by our method for computationally inexpensive optimization problems, such as the path planning subproblem, may be negated by the setup time to solve for the energy minimized LSF. For geometries with larger $N_\Gamma$ and more complex optimization problems requiring more function evaluations, we expect the speedup in

optimization time to be more pronounced.

**Table 4.5.** Constraint function evaluations and optimization time for the concentric tube robot's path planning and design optimization. The anatomy is represented using $N_\Gamma = 1,842$.

| | Lin et al. [2] | | Our method | |
|---|---|---|---|---|
| Subproblem | Path planning | Design | Path planning | Design |
| Function evaluations | 37 | 35,142 | 62 | 20,793 |
| Optimization time | 4.2 sec | 3 hr 11 min | 0.9 sec | 1 hr 24 min |
| Setup time | N/A | | 8.7 sec | |

This chapter, in part, is currently being prepared for submission for publication of the material. The authors of this work are Ryan C. Dunn, Anugrah Jo Joshy, Jui-Te Lin, Cédric Girerd, Tania K. Morimoto, and John T. Hwang. The thesis author was the primary investigator and author of this material.

# Chapter 5

# Methodology of Wind Farm Layout Optimization

## 5.1 Turbine model

Models of the wind turbine are necessary for characterizing the flow field in its wakes and for the power production of an individual turbine. For the analytical wake models and the calculation of annual energy production, we solely need a wind turbine to provide coefficient of thrust ($C_T$) and power production ($P$). For simplicity and computational efficiency, the wind turbine model is simplified to a surrogate model where $C_T$ and $P$ are simply functions of the wind speed experienced at the rotor. We note that any increase to the fidelity of the model will improve the power and $C_T$ accuracy, however the analytical wake models used in this study do not consider varying thrust/power at different radial sections of the blades. We do recommend a more accurate turbine model in the future work that will account for structural loading and fatigue on the blade structures.

The two wind turbines considered in this study are the National Renewable Energy Laboratory (NREL) 5MW open-source turbine model [77], and the International Energy Agency Wind Task 37 (IEA37) 15MW turbine model [78]. These wind turbines were selected for their size and viability in off-shore wind farms on small and large scale. The power and coefficient of thrust curves for the NREL 5MW and IEA37 15MW reference wind turbines are shown in Fig. 5.1 and Fig. B.1, respectively. The parameters of each

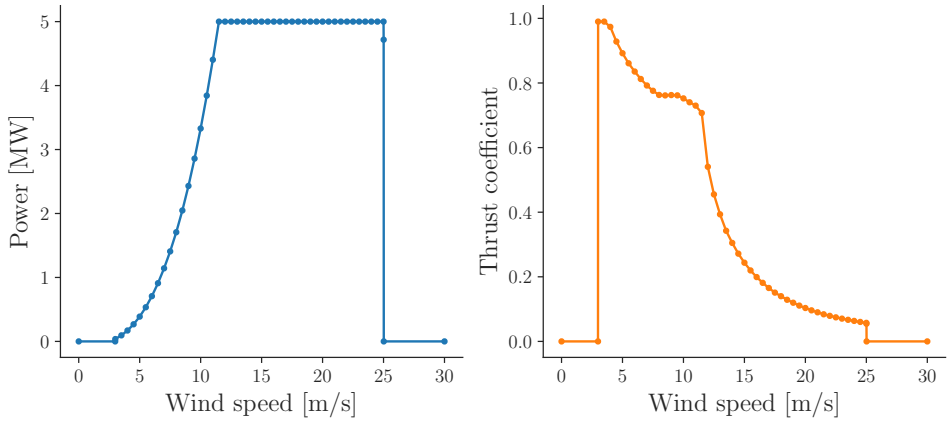


**Figure 5.1.** Power and $C_T$ curves of the NREL 5MW reference turbine [77].

wind turbine are also tabulated in Table 5.1. Note the significant diameter size.

**Table 5.1.** Parameters of the NREL 5MW and IEA37 15MW reference turbines.

| Parameter | NREL 5-MW | IEA37 15-MW |
|---|---|---|
| Rotor diameter | 126 m | 240 m |
| Hub height | 90 m | 150 m |
| Rated Power | 5 MW | 15 MW |
| Cut-in speed | 4 m/s | 3 m/s |
| Rated speed | 9.8 m/s | 10.9 m/s |
| Cut-out speed | 25 m/s | 25 m/s |

Wind speed experienced by the rotor is not uniform across the entire rotor's swept area. As a result, it is common to see rotor sampling methods to approximate the average wind speed experienced by the wind turbine. In our model, we consider the velocity at the hub (center of the rotor) for the average velocity of the rotor. We justify this decision by prioritizing the computational efficiency of the model over the accuracy, as sampling more points would increase the computational complexity of the model.

## 5.2 Wind speed and direction

Wind direction and wind speed and the two main factors that characterize the ambient wind in a wind farm. We represent the wind speed by a weibull disctribution.

The weibull probability distribution as a function of wind speed $U$ is given by

$$pdf(U) = \frac{k}{A}\left(\frac{U}{A}\right)^{k-1} \exp\left(-\left(\frac{U}{A}\right)^{k}\right), \tag{5.1}$$

where $A$ is a scale factor given in units of [m/s] and $k$ is a shape factor. A visualization of a weibull distribution is shown in Fig. 5.2, where the shape paramter $k$ is varied. It can be seen that a larger shape factor results in a distribution more centered around the scale factor $A$. It is common practice for weibull parameters $A$ and $k$ to be given for multiple wind directions from real-world data studies, as the weibull distribution varies from direction to direction. Note that we do not consider a atmospheric boundary layer with a power law profile. This is a significant limitation to the model and is of high priority for future work.

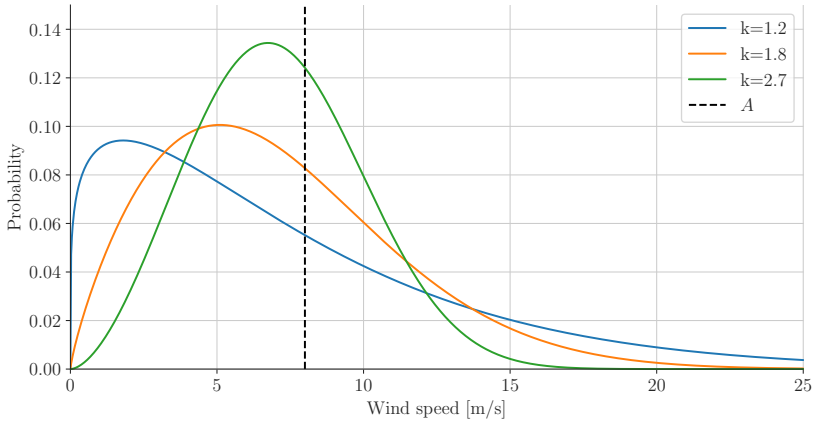


**Figure 5.2.** Weibull distribution varying the shape parameter $k$ with Weibull scale $A = 8$.

The characterization of wind direction is emperical. The visualization of wind direction distributions is given by a polar plot called a wind rose. The wind rose describes the probability distribution of the radial direction of wind at a particular wind farm site. Examples of wind roses are shown in Figures B.5, B.12, and B.14. The data in Fig. B.5 is based on the International Energy Agency (IEA) Wind Task 37 (IEA37) [79]. The data in Fig. B.12 is based on the Lillgrund off-shore wind farm with data presented by [25] and

the weibull parameters are detailed in Table B.1. The data in Fig. B.14 is based on the Hollandse Kust (West) (HKW) off-shore wind farm with data presented by [80] and the weibull parameters are detailed in Table B.2.

## 5.3 Flow field calculation

### 5.3.1 Wake deficit models

#### Jenson top-hat

For sake of completeness, we present the historically significant wake model developed by Jensen [81]. The wake deficit is given by

$$\frac{\Delta U}{U_{ref}} = \frac{1-\sqrt{1-C_T}}{\left(1+\frac{2k_{wake}x}{D}\right)^2}, \tag{5.2}$$

where $k_{wake}$ is the wake growth rate, $C_T$ is the coefficient of thrust of the wind turbine, $D$ is the diameter of the wind turbine, and $x$ is the downwind distance in reference to the upwind wind turbine. It is common practice to see the velocity deficit $\Delta U$ be a normalized value with respect to some reference wind speed $U_{ref}$. More context to $U_{ref}$ is given in subection 5.3.2. This model, commonly referred to as the top-hat model, has a point of non-differentiability where the unaffected flow collides with the wake deficit field. As a result, we do not implement this model.

#### Bastankhah

Another very popular wake model is the Bastankhah wake model [82]. Bastankhah, unlike the Jensen model, assumes a gaussian distribution in the wake deficit, making is smooth and favorable for gradient-based optimization. The Bastankhah wake deficit is

given by

$$\epsilon = 0.2\sqrt{\frac{1+\sqrt{1-C_T}}{2\sqrt{1-C_T}}} \tag{5.3}$$

$$\frac{\Delta U}{U_{ref}} = \left(1-\sqrt{1-\frac{C_T}{8\left(k^*x/D+\epsilon\right)^2}}\right) \exp\left(\frac{-1}{2\left(k^*x/D+\epsilon\right)^2}\left(\left(\frac{y}{D}\right)^2+\left(\frac{z-z_h}{D}\right)^2\right)\right), \tag{5.4}$$

where $k^*$ is the wake growth rate, $\epsilon$ is the wake diamater at the rotor, $C_T$ is the coefficient of thrust of the wind turbine, $D$ is the wind turbine diameter, $z_h$ is the hub height of the wind turbine, and $x, y$ are the downwind and crosswind distances in reference to the upwind wind turbine.

**Gaussian**

In another one of Bastankhah's works, a different model was derived to characterize the deflection of the wake induced by turbine yaw [83]. This model is called the Gaussian model. The wake deficit of the Gaussian model is given by

$$x_0 = \frac{D\cos(\gamma)(1+\sqrt{1-C_T}}{\sqrt{2}(2.32I+0.154(1-\sqrt{1-C_T}))} \tag{5.5}$$

$$\frac{\Delta U}{U_{ref}} = \left(1-\sqrt{1-\frac{C_T\cos(\gamma)D^2}{8\sigma_y\sigma_z}}\right)\exp\left(-\frac{1}{2}\left(\frac{y-\delta}{\sigma_y}\right)^2-\frac{1}{2}\left(\frac{z-z_h}{\sigma_z}\right)^2\right), \tag{5.6}$$

where $x_0$ is the downwind location where the velocity distribution achieves self-similarity, $D$ is the turbine diameter, $C_T$ is the thrust coefficient of the turbine, $\gamma$ is the turbine's yaw misalignment angle in radians, $I$ is the turbulence intensity, $\delta$ is the wake deflection distance, $\sigma_y$ and $\sigma_z$ are the widths of the wake in th $y$ and $z$-directions, respectively, $y$ is the horizontal distance in the crosswind direction, $z$ is the vertical height, and $z_h$ is the vertical hub height of the wind turbine. For our model, we use a modified equation from the Crespo and Hernández turbulence intensity model [84]. The added turbulence

intensity from a wind turbine is given by

$$I = 0.5a^{0.8} I_a^{0.5} \left(\frac{x}{D}\right)^{-0.32}, \tag{5.7}$$

where $a$ is the axial induction, $I_a$ is the ambient turbulence intensity, $x$ is the downwind distance, and $D$ is the wind turbine diameter. Axial induction is computed by the simple relation $a = \frac{1}{2\cos(\gamma)}\left(1 - \sqrt{1 - C_T \cos(\gamma)}\right)$ [83].

The near wake deflection $\delta_{near}$ for $x < x_0$ is given by

$$\theta_{c0} = \frac{0.3\gamma}{\cos(\gamma)} \sqrt{1 - \sqrt{1 - C_T \cos(\gamma)}} \tag{5.8}$$

$$\delta_{near} = x \tan(\theta), \tag{5.9}$$

where $\theta_{c0}$ is the added flow skew angle. The far wake deflection distance $\delta_{far}$ for $x > x_0$ is given by

$$\sigma_{z0} = D \frac{\sqrt{1 + \sqrt{1 - C_T \cos(\gamma)}}}{8(1 + \sqrt{1 - C_T})} \tag{5.10}$$

$$\sigma_{y0} = \sigma_{z0} \cos(\gamma) \tag{5.11}$$

$$\sigma_y = k_y(x - x_0) + \sigma_{y0} \tag{5.12}$$

$$\sigma_z = k_z(x - x_0) + \sigma_{z0} \tag{5.13}$$

$$C_0 = 1 - \sqrt{1 - C_T} \tag{5.14}$$

$$E_0 = (C_0)^2 - 3(C_0)e^{1/12} + 3e^{1/3} \tag{5.15}$$

$$\delta_{far} = x_0 \tan(\theta_{c0}) + \frac{\theta_{c0} E_0}{5.2} \sqrt{\frac{\sigma_{y0}\sigma_{z0}}{k_y k_z C_T}} \exp\left(\frac{(1.6 + \sqrt{C_T})\sqrt{\frac{1.6\sigma_y\sigma_z}{\sigma_{y0}\sigma_{z0}}} - \sqrt{C_T}}{(1.6 - \sqrt{C_T})\sqrt{\frac{1.6\sigma_y\sigma_z}{\sigma_{y0}\sigma_{z0}}} + \sqrt{C_T}}\right), \tag{5.16}$$

where $\sigma_{y0}$ and $\sigma_{z0}$ are the wake widths $\sigma_y$ and $\sigma_z$ at $x_0$, $k_y$ and $k_z$ are the wake growth rates in the $y$ and $z$-directions, respectively, $C_0$ is the normalized velocity deficit in the

center of the deficit core, and $E_0$ comes from the derivation.

**Wake expansion continuation**

Wake Expansion Continuation (WEC) is a model developed to decrease the local minima within layout optimization problems [29]. At its core, WEC performs a simple manipulation to the wake models in order to increase the width of the turbine wakes, producing the intended effect. Adding the WEC factor $\xi$ to the Bastankhah wake deficit model from Eq. (5.4) we obtain

$$\epsilon = 0.2\sqrt{\frac{1+\sqrt{1-C_T}}{2\sqrt{1-C_T}}} \tag{5.17}$$

$$\begin{aligned}\frac{\Delta U}{U_{ref}} = &\left(1-\sqrt{1-\frac{C_T}{8\left(k^*x/D+\epsilon\right)^2}}\right) \\ &\exp\left(\frac{-1}{2\left(k^*x/D+\epsilon\right)^2}\left(\left(\frac{y}{\xi D}\right)^2+\left(\frac{z-z_h}{\xi D}\right)^2\right)\right),\end{aligned} \tag{5.18}$$

where a WEC factor of $\xi = 1$ is identical to the original model. The result of adding this term is a new wake model that is more favorable to gradient-based optimization. The wake has the same centerline velocity deficit as the original model, but a wider area of effect. The resultant affect on the annual energy production (AEP) of a 3 wind turbine row in a simple example is shown in Fig. 5.3. It is clear that the model reduces the local minima and creates a smooth function easier to navigate with a gradient-based optimizer.

### 5.3.2 Wake superposition

Wake superposition is an important method to combine the wakes from multiple wind turbines into the resultant velocity deficit in the downwind flow field. Four methods are presented for completeness. Two of these models are based on the superposition of energy deficit ($\Delta u^2$) [85, 86] and the other two are based on the superposition of the velocity deficit ($\Delta u$) [87, 88]. The second unique feature of these models is the underlying

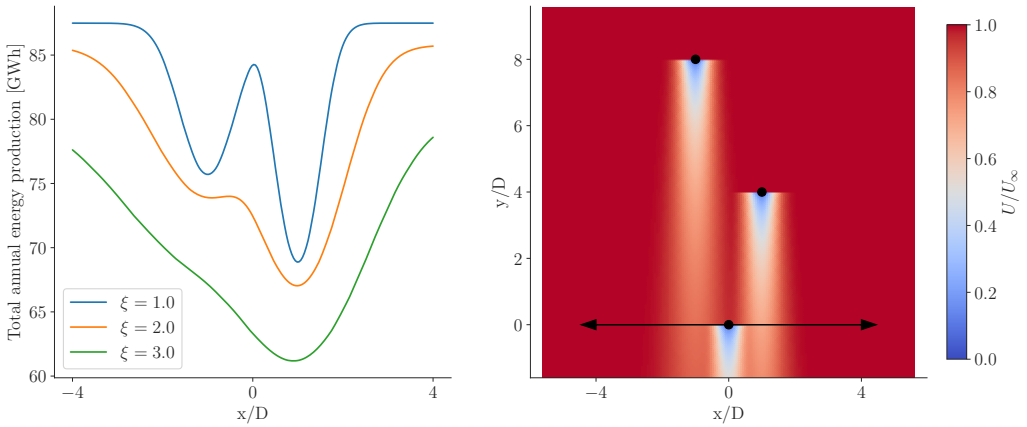


**Figure 5.3.** Annual energy production with Wake Expansion Continuation on a 3 wind turbine row varying position of the last turbine.

assumption of the value of $U_{ref}$. Rotor-based models assume that $U_{ref}$ is the velocity at the rotor of the wind turbine. Ambient-based models assume that $U_{ref}$ is approximately $U_\infty$, or the ambient wind speed. It should be noted that the rotor-based methods are not grounded in theoretical justification, but do show favorable accuracy in cases with small wind turbine spacing and large velcity deficits $\Delta U$ such that the prior assumption $U_{rotor} \approx U_\infty$ no longer holds [89].

The superposition models are given by

$$U_w = U_\infty - \sqrt{\sum_{i=1}^{N_t} \left[ U_i \left( \frac{\Delta U}{U_{ref}} \right)_i \right]^2} \tag{5.19}$$

$$U_w = U_\infty - \sqrt{\sum_{i=1}^{N_t} \left[ U_\infty \left( \frac{\Delta U}{U_\infty} \right)_i \right]^2} \tag{5.20}$$

$$U_w = U_\infty - \sum_{i=1}^{N_t} U_i \left( \frac{\Delta U}{U_{ref}} \right)_i \tag{5.21}$$

$$U_w = U_\infty - \sum_{i=1}^{N_t} U_\infty \left( \frac{\Delta U}{U_{ref}} \right)_i D, \tag{5.22}$$

where $U_w$ is the velocity deficit in the wake, $U_i$ is the wind speed at the rotor of wind turbine $i$, Eq. (5.19) is the rotor-based superposition based on a root sum-of-squares [88],

Eq. (5.21) is the rotor-based superposition based on a linear sum [86], Eq. (5.20) is the ambient-based superposition based on a root sum-of-squares [87], and Eq. (5.22) is the ambient-based superposition based on a linear sum [85]. A visualization of the different superpositions on a three wind turbine row is shown in Fig. B.2. We note that the default superposition options within FLORIS and TOPFARM is the ambient-based superposition using a root sum-of-squares method (Eq. (5.20)). We choose to follow the popular choice and apply Eq. (5.20) in all optimization studies.

### 5.3.3 Iterative solver

In order to solve for the wind turbine's power production, a solution must be calculated to the flow field in order to calculate the wind speed at the rotor. For wind turbines not in the wake of any other turbine, their wind speed is trivially equal to $U_\infty$. However, for all wind turbines within the wakes of other turbines, the corresponding velocity deficit must be computed. When multiples turbines are within multiple other wind turbine wakes, the problem becomes quickly unmanageable to solve directly. Instead, a common approach is to formulate the velocity of each wind turbine into an iterative solver that converges to a flow field solution for the entire wind farm.

We define the residual function $\mathcal{R}$ to represent the implicit function solved by the iterative solver. The residual function is given by

$$\mathcal{R}(U_{eff}) = U_{eff} - (U_\infty - U_w(U_{eff})), \tag{5.23}$$

where $U_{eff}$ is the state variable representing the effective wind speed at the rotor hubs, $U_w$ is the velocity deficit in the wake calculated as a function of the aforementioned models which rely on $U_{eff}$ primarily for the calculation of each wind turbine's thrust coefficient $C_T$. In practice, an initial guess of $U_{eff} = U_\infty$ is used, and at each iteration the wakes of the upwind-most turbines are induced on downwind turbines.

In PyWake, this solver is called *All2AllIterative*, and in FLORIS it is called *sequential_solver*. At their core, these models iterate through all wind turbines or until convergence is met (the latter more often). In our formulation, we choose to use the built-in Nonlinear Block Gauss-Seidel (NLBGS) solver in CSDL. NLBGS is a generic fixed-point iterative solver and is suitable to the problem.

## 5.4 Annual energy production

Annual energy production (AEP) is defined as the amount of energy produced in a year by a particular wind energy system. AEP behaves as a metric to measure the effectiveness of a wind energy system in all conditions that may arise throughout the year. The equation for AEP of a wind farm is given by the expected value function

$$\text{AEP} = 8760 \int_0^{360} \int_0^{U_{max}} \rho(\theta, U) \left( \sum_{k=1}^{N_t} P_k(\theta, U) \right) \, d\theta dU, \tag{5.24}$$

where 8760 comes from the number of hours in a year, $\rho$ is the joint distribution function for the wind speed $U$ and wind direction $\theta$, $N_t$ is the number of wind turbines, and $P_k$ is the power production of an individual turbine at a given state of wind direction and speed. In practice, AEP is calculated by discretizing the random variables $\theta$ and $U$. The approximated value of AEP is then given by

$$\text{AEP} \approx 8760 \sum_{i=1}^{N_{wd}} \sum_{j=1}^{N_{ws}} \rho(\theta_i, U_j) \left( \sum_{k=1}^{N_t} P_k(\theta_i, U_j) \right), \tag{5.25}$$

where $N_{wd}$ is the number of wind direction bins, and $N_{ws}$ is the number of wind speed bins. The most accurate results are produced when $N_{wd}$ and $N_{ws}$ are large, but increases the complexity of the function.

## 5.5 Optimization problems

The layout optimization problem is defined as

$$\begin{aligned} \text{maximize} \quad & f = \text{AEP} \\ \text{with respect to} \quad & \mathbf{x}, \mathbf{y} \in \mathbb{R}^{N_t}, \mathbb{R}^{N_t} \\ \text{subject to} \quad & \textit{spacing constraints} \\ & \textit{boundary constraints}, \end{aligned} \tag{5.26}$$

where the spacing constraint enforces the distance between each wind turbine be greater than or equal to some minimum and the boundary constraint enforces that each wind turbine be placed within a user defined function. In all optimizations presented within this thesis, the spacing constraint was that the turbines had at least 1.8 turbine diameters between one another. Note that the number of spacing constraints is $N_t(N_t - 1)/2$ and the number of boundary constraints are $N_t$. In practice, these constraints are represented using KS functions in order to reduce the number of constraints enforced by the optimizer to improve performance. The yaw control optimization problem is defined as

$$\begin{aligned} \text{maximize} \quad & f = \text{AEP} \\ \text{with respect to} \quad & \gamma \in \mathbb{R}^{N_{wd} \times N_{ws} \times N_t} \\ \text{subject to} \quad & \gamma_{lower} \leq \gamma \leq \gamma_{upper}, \end{aligned} \tag{5.27}$$

where $\gamma_{lower}$ and $\gamma_{upper}$ are the lower and upper bounds for the design variable $\gamma$, respectively. These design variable bounds are not considered constraints in the optimization problem, but nonetheless presented for completeness. The hub height optimization problem

is defined as

$$\begin{aligned} \text{maximize} \quad & f = \text{AEP} \\ \text{with respect to} \quad & \mathbf{z_h} \in \mathbb{R}^{N_t} \\ \text{subject to} \quad & z_{lower} \leq z_h \leq z_{upper}, \end{aligned} \tag{5.28}$$

where $z_{lower}$ and $z_{upper}$ are the lower and upper bounds for the design variable $z$, respectively. These design variable bounds are not considered constraints in the optimization problem, but nonetheless presented for completeness. A summary of all optimization problems and their properties are tabulated in Table 5.2.

**Table 5.2.** Table of properties for the three investigated wind farm optimization problems.

| | Layout | Yaw | Hub Heights |
|---|---|---|---|
| Contains local minima | ✓ | ✓ | ✓ |
| # of Design Variables | $2N_t$ | $N_{wd}N_{ws}N_t$ | $N_t$ |
| # of Constraints | 2 | 0 | 0 |

Note that the complexity of all models will scale with $\mathcal{O}(N_{wd}N_{ws}N_t^2)$. This scaling comes from the wake models which must calculate the wake deficit from each turbine, to each other turbine, for each wind speed, and for each wind direction. In terms of the iterative solver, $N_t$ iterations is a worst-case number of iterations required. Some frameworks address this scaling. In PyWake and FLORIS, the iterative solvers only perform calculations on downwind turbines whose rotor wind speed is affected by another wake. In a fully continuous and differentiable framework like CSDL, these operations are not continuous and are instead considered in full by using smooth masking matrices.

Using the aforementioned models and problem setups, we modeled a wind farm within CSDL. CSDL provided convenience with automatic sensitivity analysis, fewer lines of code, and familiarity with the authors. The model's structure for the defined problems is the same, apart from the inclusion or exclusion of some values as design variables. The design structure matrix is shown in Fig. 5.4

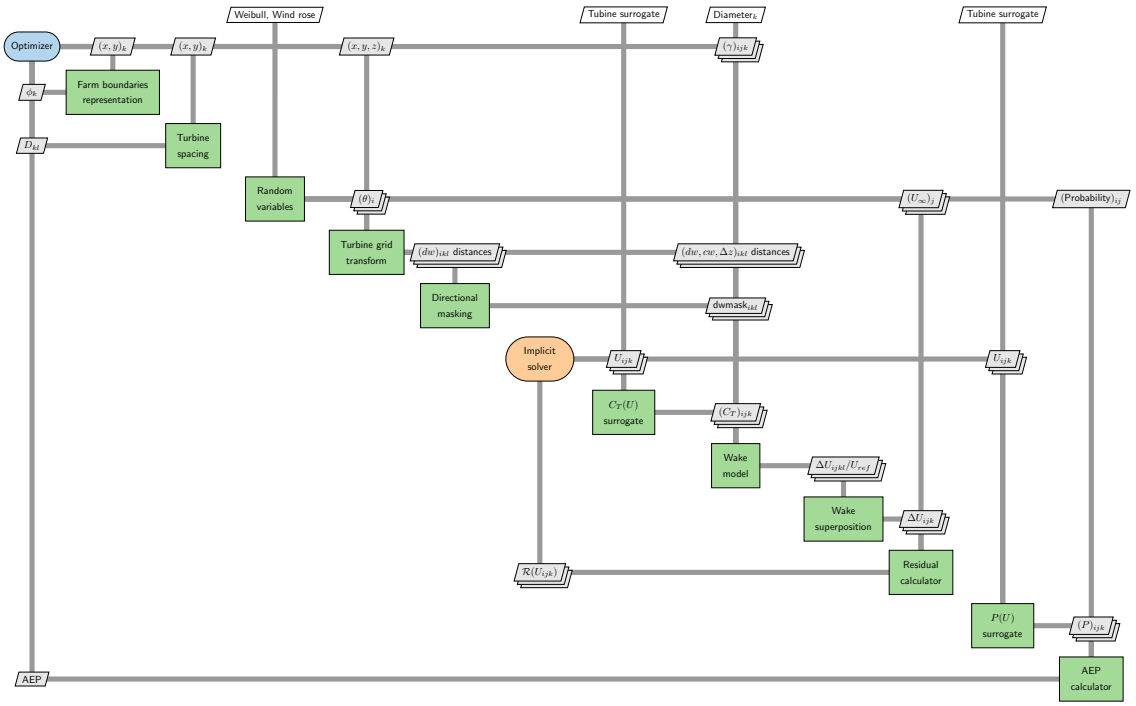


**Figure 5.4.** Design structure matrix for wind farm optimization.

## 5.6 Model verification

While this thesis does not provide validation with ground-truth, verification is provided to the existing models within PyWake and FLORIS. The major assumptions with this model depends on the accuracy of models individually. Validation studies are important and would be a valuable future work to this thesis. A study on flexible and stiff blades [90] found that low fidelity models have significant discrepancies at low wind speeds and small inter-turbine spacing. Additionally, we rely heavily on the assumption an accurate thrust coefficient curve for the given wind turbine.

An example to verify the Bastankhah wake model (Eq. (5.4)) in a simple 3 turbine row is done in Fig. B.3. The error in the AEP calculated from our model on the order of machine precision, confirming the model's accuracy. Derivative calculation is automatically done in CSDL, but for additional validation it is verified with finite differencing during optimization, and is confirmed to be on the order of the step length of the finite difference

$\mathcal{O}(10^{-8})$.

A verification of the Gaussian wake model (Eq. (5.6)) in a simple 3 turbine row is done in Fig. B.4. The net AEP calculated from our model has a normalized error on the order of machine precision, confirming the model's accuracy. Derivative calculation is automatically done in CSDL, but for additional validation it is confirmed to be on the order of the step length of the finite difference $\mathcal{O}(10^{-8})$.

This chapter, in part, is currently being prepared for submission for publication of the material. The authors of this work are Anugrah Jo Joshy, Ryan C. Dunn, and John T. Hwang. The thesis author was a contributor to this material.

# Chapter 6

# Results of Wind Farm Layout Optimization

All studies within this chapter were executed on a desktop with an 8-core Ryzen 7 @ 3.6 GHz processor and 32 GB of RAM. We do not implement multi-threading or parallelization with GPUs in any of our numerical experiments.

## 6.1 Computational model

We conduct a scaling study on our model and the TOPFARM model to compute the annual energy production (AEP) and spacing constraints of a wind farm with varying numbers of wind turbines. The expectation of the models is to scale in computational complexity with $\mathcal{O}(N_{wd}N_{ws}N_t^2)$ in both model evaluation and sensitivity analysis. We consider a range of the number of turbines between 10 for a small wind farm and 1000 for a very large study with multiple wind farms. For each of the data points, the models were evaluated 10 times and an average was taken in their computation time to minimize the variance from the computer used. The results of this scaling study are shown in Fig. 6.2, with logarithmic scaling to highlight the quadratic dependance on $N_t$. As expected, the asymptotic scaling of both models depends on $\mathcal{O}(N_{wd}N_{ws}N_t^2)$, or a slope of 2 in the figure. In terms of model evaluation, our new model is faster to evaluate for small wind farms ($N_t < 60$), and slower to evaluate for larger wind farms ($N_t > 60$) compared to TOPFARM.

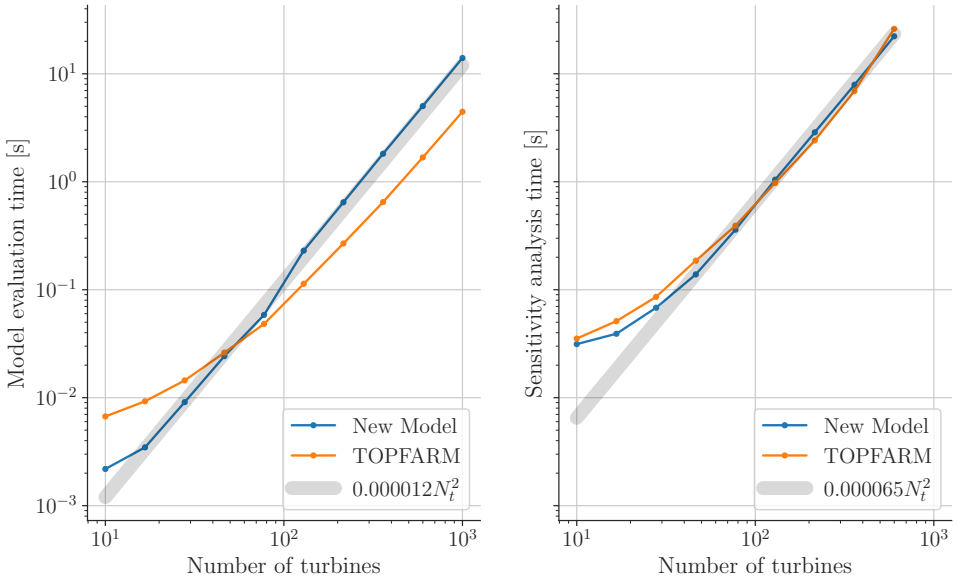


**Figure 6.1.** Model evaluation and sensitivity analysis time scaling with respect to the number of wind turbines of our new model compared to TOPFARM.

Although the differences appear negligible, the gradient-based optimization of a wind farm will evaluate the model hundreds of times and further exaggerate the difference in model efficiency. We predict these minor differences in model evaluation time source from the difference in the frameworks used, PyWake and CSDL. In terms of sensitivity analysis, the two models have nearly identical computation times. The difference between the two models is on the order of the machine variance between the multiple evaluations done, so no meaningful differences between the models are identified for sensitivity analysis.

We conduct another scaling study between our model and the TOPFARM model in model evaluation and sensitivity analysis computation time with respect to the number of points representing the wind farm boundary. In this study, a model evaluation and sensitivity analysis includes the annual energy production, spacing constraints, and the boundary constraints of the model. We consider a range of the number of points representing the boundary $N_\Gamma$ from 10 for a simple boundary, $10^3$ for a simple wind farm boundary, and up to $10^5$ for a complex wind farm boundary, such as in Fig. B.15. We extend this study to $N_\Gamma = 10^6$ in order to confirm the computational scaling for very large values. On

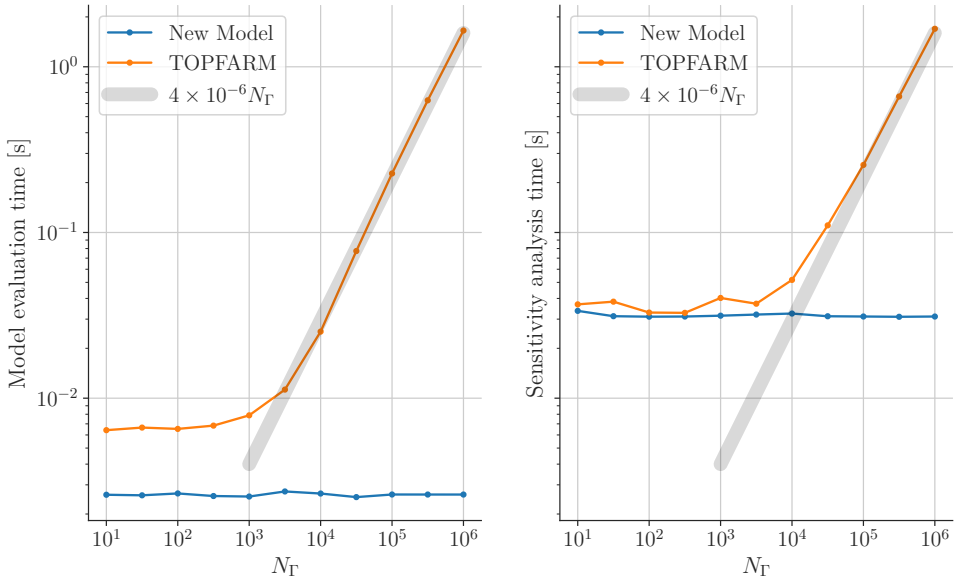


**Figure 6.2.** Model evaluation and sensitivity analysis computation time increasing the number of points discretizing the boundary for our new model compared to TOPFARM.

the backend, TOPFARM uses the explicit method of Risco et al. [4], which is expected to scale in computational complexity with $\mathcal{O}(N_\Gamma)$. Our model uses the new method presented in Chapter 3, which is expected to scale in computational with $\mathcal{O}(1)$. Similarly, each data point was evaluated 10 times and an average was taken across the computation time to mitigate the variance from the computer. The results of this study is shown in Fig. 6.2. The asymptotic scaling for the model evaluation and sensitivity analysis of Risco et al. is $\mathcal{O}(N_\Gamma)$, while ours remains $\mathcal{O}(1)$. For models with simple boundaries ($N_\Gamma < 10^3$), the boundary constraint's time-to-evaluate and perform sensitivity analysis is negligible compared to the objective calculation. As a result, the results in Fig. 6.1 show a flat region with no scaling with respect to $N_\Gamma$. The expectation for larger models with greater $N_t$ is a vertical shift in the graphs. This result confirms and illustrates the significance of the contribution to the new scalable constraint formulation from Chapter 3.

## 6.2 Simple layout optimizations

As part of the International Energy Agency Wind Task 37 (IEA37) effort, a set of standardized wind farm layout optimization problems were created to encourage collaboration among researchers to find the best algorithms to locate the most optimal solution. A report on the IEA37 cases with multiple optimization algorithms is shown in [9], where the gradient-based optimization with SNOPT and Wake Expansion Continuation (SNOPT+WEC) consistently exceeded other approaches.

The IEA37 initial layouts for 16, 36, and 64 wind turbines are shown in Figures B.6,B.8, and B.10, respectively. The wind farm boundary for these cases are circular and simple to represent. The windrose for these problems is shown in Fig. B.5. Note that $N_{wd} = 16$ and $N_{ws} = 1$. The layout optimization problem is given by problem (5.26), and the Bastankhah wake model was used (Eq. (5.4)).

Our optimization results using SNOPT and our model with NREL's 5MW reference turbine for the 16, 36, and 64 wind turbine cases are shown in Figures B.7, B.9, and B.11, respectively. In the optimal layout of 16 wind turbines, we note a distinct row of wind turbines along the 122.8° wind direction. This alignment is shown in Fig. 6.3 and is a biproduct of the optimizer exploiting the wind direction discretization, as this wind direction lies directly between the 112.5° and 135.0° directions evaluated in the model. To address this explotation issue, we recalculate the windrose with linear interpolation into more wind direction bins. At its highest resolution, we perform a full calculation of the AEP in 1° increments. The original 16 bin model has a 6.71% error in AEP calculation to the 360 bin model in the optimal position. While more accurate, the 360 bins is very computationally expensive and therefore impractical to use in optimization. Instead, we find that by using 36 wind direction bins we reduce the error in AEP calculation of the optimal result to 1.40% compared to the 360 wind direction discretization. We find that with any wind direction discretization, the optimizer will exploit the gaps between wind

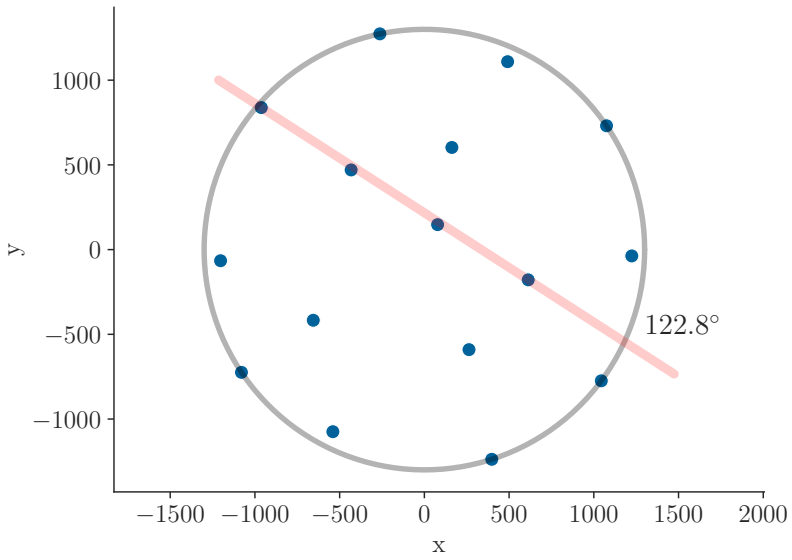


**Figure 6.3.** Optimized layout of the IEA37 test case with 16 wind turbines, discretized with 16 wind directions, with a row of aligned turbines.

direction bins.

We recognize that this problem is highly multi-modal, and the application of SNOPT alone is prone to fall into local minima. As a result, we optimize the problems using the Wake Expansion Continuation (WEC) model [29] (Eq. (5.18)). With WEC, two optimizations are performed sequentially. First, an optimization with $\zeta = 4$, then with $\zeta = 1$, where $\zeta = 1$ is equivalent to the original Bastankhah model of Eq. (5.4). The selection of $\zeta = 4$ is by trial and error for this particular problem, and may vary in performance depending on the layout. Tables 6.1, 6.2, and 6.3 tabulate the results of the optimization studies conducted with and without WEC for the 16, 36, and 64 wind turbine cases, respectively. As expected, SNOPT+WEC improves the optimizer's ability to navigate through local minima to find a more optimal result. However, the use of WEC resulted in an increase in the number of model evaluations. This result confirms the findings in [9] and highlights the impact that WEC has on gradient-based layout optimization of wind turbines.

**Table 6.1.** Optimization results on the IEA37 16 wind turbine case with SNOPT and SNOPT+WEC.

| Method | # of Model Evaluations | Optimized AEP | AEP Increase |
|---|---|---|---|
| SNOPT | 87 | 392.634 GWh | 1.52% |
| SNOPT+WEC | 140 | 393.590 GWh | 1.76% |

**Table 6.2.** Optimization results on the IEA37 36 wind turbine case with SNOPT and SNOPT+WEC.

| Method | # of Model Evaluations | Optimized AEP | AEP Increase |
|---|---|---|---|
| SNOPT | 167 | 846.056 GWh | 3.95% |
| SNOPT+WEC | 158 | 848.214 GWh | 4.22% |

## 6.3 Hub height optimization

In the following two studies (Sec. 6.3 and Sec. 6.4), the Lillgrund wind farm site is used. Lillgrund is an off-shore wind farm located off of the coast of Sweden and has 48 Siemens SWT2.3-93 wind turbines installed. To the author's knowledge, there is no open-source data with these wind turbines, and therefore we continue to model using the NREL 5MW reference wind turbine. While this removes capabilities to compare across studies, the use of the Lillgrund site provides a realistic layout, wind speed, and wind direction distributions that are otherwise theoretical without a reference site. The windrose of the Lillgrund site is shown in Fig. B.12, with corresponding weibull data in Table B.1, and the layout is shown in Fig. B.13. Note that $N_{wd} = 12$, $N_{ws} = 6$, and $N_t = 48$. This site is characterized by small inter-turbine spacing and is the subject of many previous optimization studies [25, 58, 59].

In the following study, we consider optimization of the hub heights of the wind turbines in the Lillgrund site layout. The optimization problem is stated in problem (5.28). Using the Lillgrund site, the optimization problem has a total of 48 design variables. The initialization is given by the standard hub height of 90 meters for each turbine and has a minimum hub height such that the blade tips remain 20 meters above the ground and

**Table 6.3.** Optimization results on the IEA37 64 wind turbine case with SNOPT and SNOPT+WEC.

| Method | # of Model Evaluations | Optimized AEP | AEP Increase |
|---|---|---|---|
| SNOPT | 105 | 1496.165 GWh | 2.60% |
| SNOPT+WEC | 392 | 1503.011 GWh | 3.07% |

a maximum hub height of 150 meters. The optimization results are shown in Fig. 6.4. The results produce an expected result that the hub heights are at the upper and lower bounds to minimize the overlap between wakes of the downwind turbines. Additionally, the hub heights of neighboring wind turbines have a somewhat alternating pattern, similar to the results of [58]. The results from this study have significant limitations As previously mentioned, the number of wind direction bins (12 in this optimization) are not enough to accurately calculate the AEP. Additionally, the using a gradient-based optimizer for this problem is prone to falling into a local minima. For example, changing the 20th turbine indicated on the figure from the lower bound to the upper bound hub height would increase the AEP to 1058.45 GWh. Furthermore, the accuracy of the low-fidelity models is not accurate at capturing the significant compromises needed to be made with changing hub heights. Increased reinforcements to the towers, the increased cost of the towers, atmospheric boundary layer, and larger rotor sampling for calculating the wind speed at the rotor are examples of models needed to increase the realism of our results. For the interested reader, a study conducted a similar optimization using the internal rate of return as the objective function which is expected to be much more realistic in [59].

## 6.4 Yaw optimization

In this optimization study, we perform a yaw misalignment optimization with the Lillgrund site. We utilize FLORIS's built in boolean optimization tool called Serial Refine [62] in order to initialize our optimization. Serial Refine reduces the design space by considering wind turbines in order of upwind to downwind, and recursively discretizes

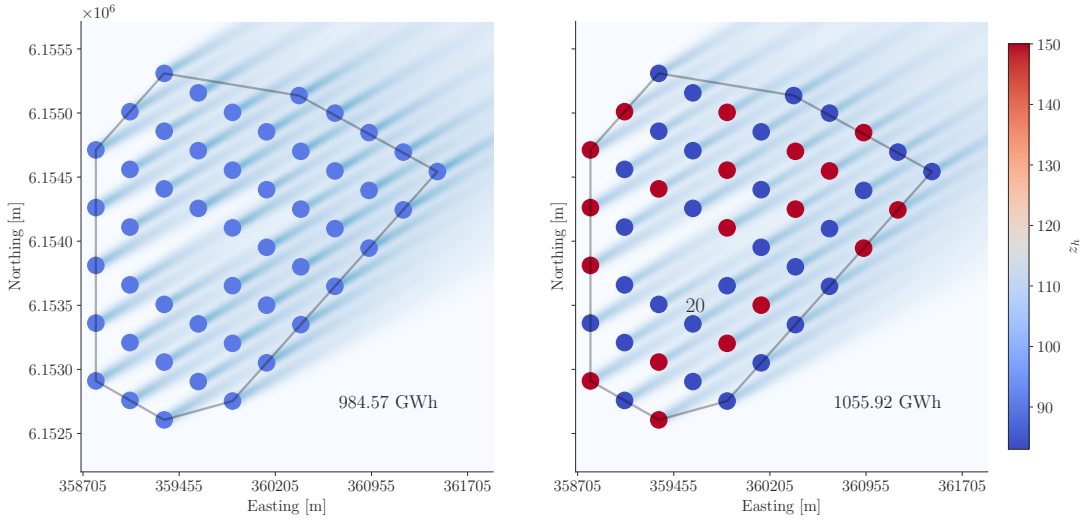


**Figure 6.4.** Initial hub height and optimized hub heights for the Lillgrund site.

the design space (yaw misalignment angles) and checks the model evaluations for the maximum value. In this way, it is a gradient-free optimizer, and it has shown to be very fast at finding a near-optimal solution and not susceptible to local minima. Because SR is fast and robust to local minima, we may apply it to get a good initial guess for our optimization. Our new model can take this suboptimal initial guess and improve it further using gradient-based optimization by considering yaw as a fully continuous variable.

The yaw optimization problem is shown in problem 5.27, and the Gaussian wake model was used (Eq. (5.6)) with the Crespo and Hernández turbulence intensity model (Eq. (5.7)). Using the Lillgrund site, the optimization problem has a total of $3,456$ design variables. The results of multiple optimizations are shown in Table 6.4, where the design space is gradually reduced. According to prior studies, positive yaw misalignment angles reduce the fatigue loads and increase the overall lifespan of the turbine blades opposed to negative yaw misalignment angles [91]. As a result, the optimization study favors design spaces with more positive yaw misalignment angles. It is shown that the SR optimization is much faster than our gradient-based optimizer, despite having a good initialization. However, by using the gradient-based optimizer, a higher AEP is able to be achieved for

approximately an order of magnitude increase in optimization time. In a previous study, it was shown that the optimization time of a gradient-based optimizer to same problem as problem 5.27 was approximately 100x slower than SR [62]. However, the gradient-based optimizer used was SLSQP using finite difference to calculate derivatives, which is much less efficient than our model. The increase in AEP by using our gradient-based model is less significant for smaller design spaces, as the $(-10°, 10°)$ yaw bounds only achieves an increase of 0.145 GWh opposed to the $(-30°, 30°)$ yaw bounds increase of 1.360 GWh in the optimal result. We attribute this to the fact that the SR discretization of the design space is held constant, so as the design space reduces in size so too does the step size between yaw angle samples. With a smaller step size, SR is expected to find a result closer to the optimal result. The optimization result for the $(-30°, 30°)$ yaw bounds is shown in Fig. 6.5.

**Table 6.4.** Yaw optimization results for the Lillgrund site using Serial Refine (SR) and the new model with different design variable bounds. Time is in seconds and AEP is in GWh.

| Yaw Bounds (°) | Optimization Time | | Optimized AEP | | Increase |
|---|---|---|---|---|---|
| (lower, uppper) | SR | New Model | SR | New Model | in AEP |
| $(-30°, 30°)$ | 20.1 | 305.5 | 958.092 | 959.453 | 1.360 |
| $(-15°, 30°)$ | 21.6 | 333.1 | 948.962 | 949.879 | 0.916 |
| $(-10°, 25°)$ | 22.4 | 277.0 | 940.651 | 940.959 | 0.308 |
| $(0°, 25°)$ | 19.8 | 266.8 | 935.219 | 935.701 | 0.482 |
| $(-10°, 10°)$ | 22.1 | 317.6 | 923.384 | 923.529 | 0.145 |

## 6.5 Layout optimization with a complex boundary

The Hollandse Kust (West) site VI (HKW) is an off-shore wind farm zone in the Dutch North Sea. Using a publicly released metocean survey, the HKW site's windrose and weibull data are presented in Fig. B.14 and Table B.2, respectively [80]. Note that $N_{wd} = 12$ and $N_{ws} = 6$ are used in this optimization study. The HKW site is planned to be commissioned in 2026 with 54 wind turbines ($N_t$) and a total capacity of 756 MW

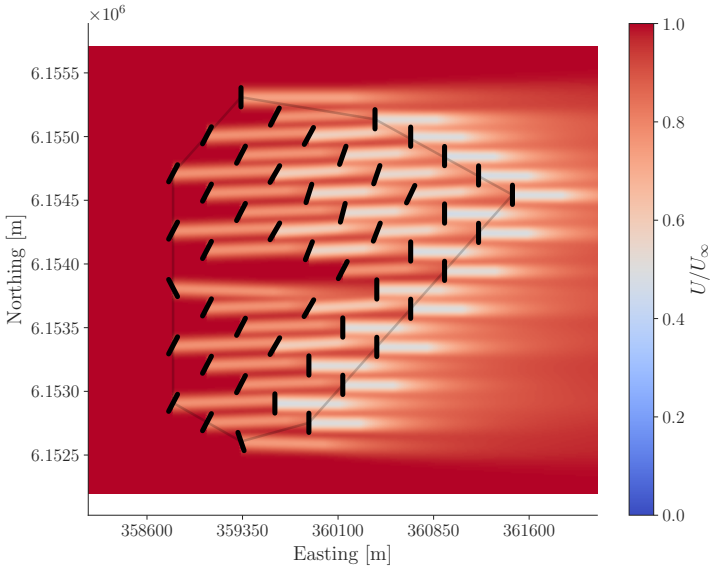


**Figure 6.5.** Visualization of the flow field for the optimal yaw misalignment of the wind turbines at the Lillgrund site at a 270° wind direction and 12.4 m/s wind speed.

[92]. As a result of the predicted size, we use the IEA37 15MW off-shore reference wind turbine to perform a layout optimization [78]. The parameters for this turbine are in Table 5.1 and $C_p$ and power curves in Fig. B.1. We note that in such a large optimization problem the optimization tolerance was almost never met. Therefore, the optimizations were terminated when the AEP did not increase by any significant margins $\mathcal{O}(10^{-3})$ and the constraints were satisfied.

The HKW wind farm boundary is defined by sets of disconnected and irregularly shaped polygons shown in Fig. B.15. The exclusion zones represent shipwrecks, unexploded ordinances, shipping lanes, steep seabed gradients, and other regions where the placement of a wind turbine is infeasible. To begin a layout optimization study on this site, we must represent these boundaries in a continuous and smooth way for gradient-based optimization. We apply the aforementioned method of geometric non-interference constraints given by Algorithm 1. The resultant values from solving the energy minimization problem (3.11) are in Table 6.5, and a visualization of the zero contour is given in Fig. B.16. Notably, the accuracy of the boundary constraint representation has an on-surface root-mean-square

(RMS) error of 6.29 meters, which is on the same order of magnitude of the tower's diameter.

**Table 6.5.** Energy minimization values for the boundary constraint function of the Hollandse Kust (West) site.

| | |
|---|---|
| Number of points ($N_\Gamma$) | 80,710 |
| Number of disconnected polygons | 5,180 |
| Setup time | 87.3 seconds |
| Control grid size | $440 \times 550$ |
| Bounding box diagonal ($L$) | $20,717$ meters |
| Relative surface RMS error | $2.250 \times 10^{-4}$ |
| Relative surface Max error | $2.882 \times 10^{-3}$ |
| Absolute surface RMS error | 6.29 meters |
| Absolute surface Max error | 59.7 meters |

In order to quantify the computational speedup of the new boundary constraint model, constraint evaluations were ran for the TOPFARM model and our new model. Running an optimization with the same initial starts would produce a different optimal layout because the constraint functions are inherently different, despite the wake models being identical. As a result, it is only appropriate to highlight the differences in a single model evaluation. With 54 wind turbines and $N_\Gamma = 80,710$, a single model evaluation in TOPFARM takes 1.060 seconds, and our new model takes 0.034 seconds. The timing results are the average timings of 10 model evaluations. Given this result, the significant increase in model evaluation time makes it significantly more computationally expensive to solve a layout optimization problem with this boundary in TOPFARM. As the number of model evaluations required for optimization, typically hundreds to thousands, the additional computational expense of using TOPFARM becomes impractical. As a result, we strongly recommend using the new boundary constraint formulation for wind farm layout optimization.

In order to optimize the wind farm, two measures were taken to avoid local minima. The first method is the implementation of the aforementioned WEC model to decrease the

local minima in the flow fields. As done in the optimization study presented in Sec. 6.2, the optimization is completed with $\zeta = 4$ first, then with $\zeta = 1$. The second method is the implementation of relaxation to the boundary constraints. The constraints are first enforced to be within 250 meters, then within 25 meters, then fully enforced to be within the wind farm boundaries. A visualization of the 250 meters boundary 25 meters boundary, and zero boundary are shown in Fig B.17. In total, the optimization problem (5.26) is completed by solving four sequential optimizations described in Table 6.6. The application of these two methods represent the state-of-the-art methodologies to avoid local minima within gradient-based wind farm layout optimization, and are still subject to local minima in practice.

**Table 6.6.** The Wake Expansion Continuation (WEC) and boundary constraint relaxation values for the four sequential optimization problems to solve the layout optimization problem of Hollandse Kust (West) site VI.

| Optimization | Initialization | WEC value ($\xi$) | Boundary Relaxation |
|---|---|---|---|
| 1 | Random | 4 | -250 m |
| 2 | From (1) | 4 | -25 m |
| 3 | From (2) | 4 | 0 m |
| 4 | From (3) | 1 | 0 m |

We conduct a wind farm layout optimization using $N_t = 54$ for the HKW site. The initialization is shown in Fig. B.18, the result of the first optimization in Fig. B.19, the result of the second optimization in Fig. B.20, the result of the third optimization in Fig. B.21, and the result of the last optimization in Fig. B.22, with annual energy production and the median distance to the boundary shown for all optimization iterations. The optimal layout found from the four sequential optimizations is shown in Fig. 6.6. The error of the wind direction discretization with respect to the $1^\circ$ discretization is 2.24%. The 360 wind direction bins were approximated using linear interpolation. Note that many wind turbines lie on the outer-most boundary, maximizing their distance from one another. Additionally, note that many wind turbines are on disconnected zones from one

another. This result is difficult to reproduce in gradient-based optimization without a boundary constraint relaxation approach. The AEP was increased by 6.82% from a random initialization that violated the boundary constraint within 250 meters. No meaningful conclusions can be made about improvement of AEP because it was subject to a random initialization. Using a heuristic-based or gradient-free approach to achieve a better initial guess is of high priority for future work. Nonetheless, the result represents the capability of using gradient-based optimization on a wind farm layout optimization problem.

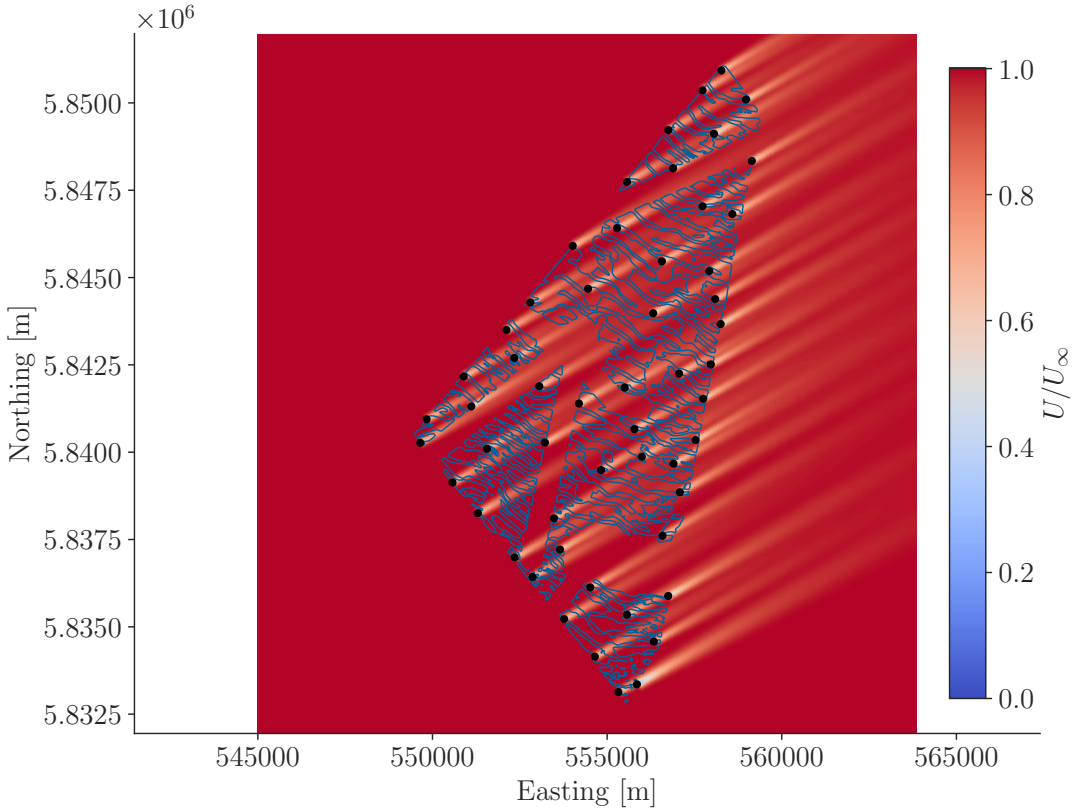


**Figure 6.6.** Optimal layout of 54 wind turbines for the Hollandse Kust (West) site VI. The wakes are shown for the 240° direction and 12.4 m/s speed.

In addition, we conduct an optimization study varying the number of wind turbines at the HKW site and tabulated the results in Table 6.7. For each data point, the sequential optimization was completed with their own random initializations. Due to the randomness of the initialization, no conclusive differences were drawn based on optimization time, as

the quality of the initial start plays an important role to the number of model evaluations required to optimize. For reference, the scaling of one model evaluation and sensitivity analysis times are shown in Fig. 6.1. Based on the results of this study, is unclear that the addition of more wind turbines than the baseline $N_t = 54$ will harm the efficiency of the overall wind farm. As shown in Table 6.7, the amount of AEP produced on a per turbine basis remains steady with the largest difference between 46 and 70 wind turbines. The incorrect conclusion drawn from this data is that increasing the number of wind turbines will increase the AEP of the wind farm without significant loss in efficiency due to wakes. In reality, this result represents the limitation of the models in our implementation in capturing diminishing returns. A better model would consider levelized cost of energy (LCoE), increased fatigue loading based on wake interactions on the turbine blades, and other models to capture to the lifecycle of the wind farm. This should be addressed in future work.

**Table 6.7.** Optimization results for the Hollandse Kust (West) site VI. Annual energy production (AEP) is in GWh.

| # of Turbines | Initial AEP | Optimized AEP | Improvement | AEP GWh/Turbine |
|---|---|---|---|---|
| 46 | 864.07 | 919.27 | 6.39% | 19.98 |
| 50 | 936.54 | 995.82 | 6.33% | 19.92 |
| 54* | 1002.94 | 1071.35 | 6.82% | 19.84 |
| 58 | 1068.91 | 1134.35 | 6.12% | 19.56 |
| 62 | 1135.91 | 1204.71 | 6.06% | 19.34 |
| 66 | 1202.89 | 1284.45 | 6.78% | 19.46 |
| 70 | 1269.09 | 1336.83 | 5.33% | 19.10 |
| 74 | 1329.21 | 1429.07 | 7.51% | 19.31 |

Note: (*) indicates the real-world number of turbines.

This chapter, in part, is currently being prepared for submission for publication of the material. The authors of this work are Anugrah Jo Joshy, Ryan C. Dunn, and John T. Hwang. The thesis author was a contributor to this material.

# Chapter 7

# Conclusion

**Summary**

Two topics were explored in this thesis. The first topic was concerned with developing a new scalable geometric non-interference constraint formulation. In Sec. 1.1, we consolidated the terminology used in prior literature and call this category of constraints 'geometric non-interference constraints'. Additionally, we framed the set of optimization problems with geometric non-interference constraints into three groups: layout optimization, shape optimization, and optimal path planning problems. Section 2.1 reviewed the existing geometric non-interference constraint formulations in gradient-based optimization and contextualized our formulation within the field of surface reconstruction. In Chapter 3, we drew upon ideas from surface reconstruction techniques to construct our constraint formulation. Our formulation is based on an approximation of the signed distance function generated by solving an energy minimization problem for the values of the B-spline control points. Chapter 4 presented accuracy and scaling studies with our formulation. We also solved a path planning and shape optimization problem using our new formulation.

The second topic was concerned with investigating gradient-based wind farm optimization problems. In Sec. 1.2, we introduced the scope of optimization and its importance to wind farm design in combating climate change. Section 2.2 reviewed the

literature that conducted studies on wind farm optimization. A clear gap in these studies was identified within gradient-based optimization, and particularly, the continuous and smooth representation of wind farm boundary constraints. In Chapter 5, we presented the models used to represent the turbines, wakes, and optimization model to conduct the optimization studies. Chapter 6 presented the results of scaling and optimization studies, including hub heights, yaw misalignment, and a layout optimization problem.

**Results**

The first topic of this thesis contributes a new formulation for representing geometric non-interference constraints in gradient-based optimization. This formulation involves a scalable, smooth, and fast-to-evaluate constraint function that approximates the local signed distance to a geometric shape. The use of B-spline functions is key to our formulation being scalable, smooth, and fast-to-evaluate. We showed that our formulation achieves a level of accuracy on the same order of magnitude as surface reconstruction methods used in computer graphics. Additionally, our formulation scales better in accuracy, up to a certain limit, and computational time with respect to the number of points sampled on the geometric shape $N_\Gamma$ compared to previous non-interference constraint formulations used by the optimization community. Our resultant computational speed is on the order of $10^{-6}$ seconds per point as measured on a modern desktop workstation, entirely independent of the number of sample points $N_\Gamma$. The method results in a 78% and 56% speedup in optimization time for a path planning and design subproblem, respectively, for an existing concentric tube robot (CTR) gradient-based design optimization problem.

The second topic of this thesis investigates the use of gradient-based optimization for wind farm layout design. We investigated three subproblems to wind farm optimization problems including turbine hub heights, yaw misalignment control, and wind farm layout optimization. In the optimization of hub heights, we identified consistent patterns as previous works, however were subject to local minima. In the optimization of yaw

misalignment angles, an increase of 1.360 GWh AEP was achieved for only an order a magnitude increase in optimization time, compared to a gradient-free optimizer. It was shown that the new geometric non-interference constraint formulation provides an efficient way to enforce complex wind farm boundary constraints. Model evaluations with the new constraint function were 30x faster than previously in an industry leading optimization framework. In addition, the constraint formulation is well-suited for relaxation approaches, as it provides an approximate distance to the boundary. We investigated the vulnerabilities of our model including the lack of diminishing returns when adding more wind turbines, local minima in the results, and inaccuracies in the annual energy production (AEP) calculation due to the wind direction discretization.

**Future work**

We identify multiple directions for future work for the first topic. Adaptive octrees with B-splines can represent small-scale features such as edges and sharp corners more accurately. Using octrees for discretization instead of using a uniform grid can clearly yield faster and more accurate solutions in problems where any of the modeled geometries remain constant during optimization iterations, e.g., the CTR or wind farm layout optimization problems. However, it is worth restating that when geometries evolve during optimization, rediscretizing surfaces using octrees in each optimization iteration becomes unreasonably expensive, and we only recommend a uniform discretization in such cases. Acceleration with multi-threading or graphics processing units (GPUs) is another possible direction for future research.

For the second topic, there are many directions to improve the models or expand the results. Uncertainty in wind farm optimization is a major issue to address. Using uncertainty quantification to address the variability in the wind speed, wind direction, and performances of individual turbines will increase the confidence in optimal solutions. In addition, the inclusion of more models to represent the distribution of wind speeds at

the rotor, tower and blade structures, rotor-nacelle assemblies, and cost models would greatly increase the fidelity of the models. More accurate wake models than the ones presented in this thesis are also under continued research, although may not be well suited for gradient-based optimization. To expand the results, more optimization studies where design variables are considered simultaneously should be considered. The computational cost of these models should also be addressed with parallelization, such as in the AEP calculation.

# Appendix A

# Geometric non-interference constraint

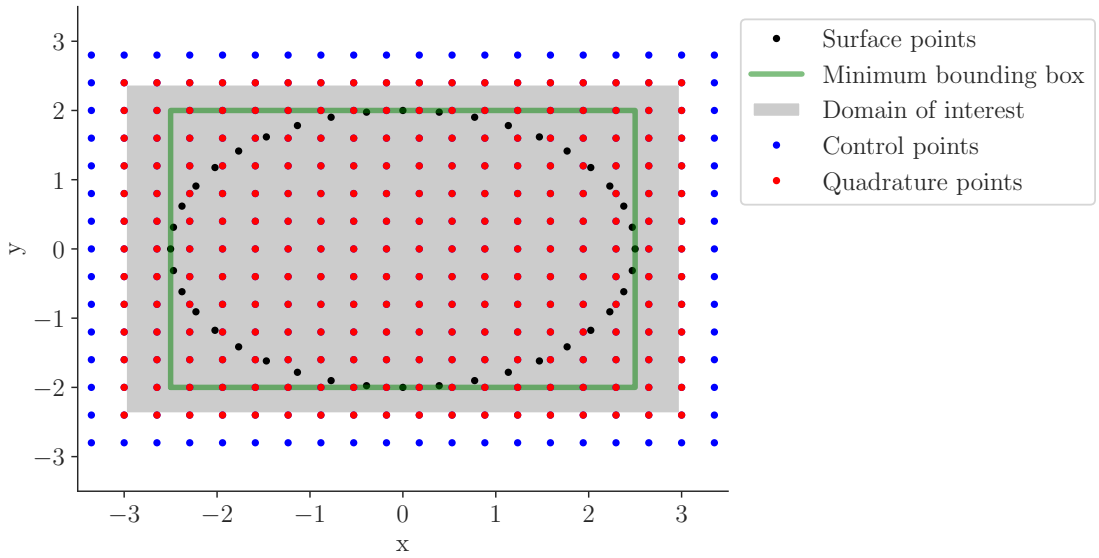


**Figure A.1.** The various grids in the geometric non-interference constraint representation for an ellipse in 2D.

# Appendix B

# Wind farm optimization

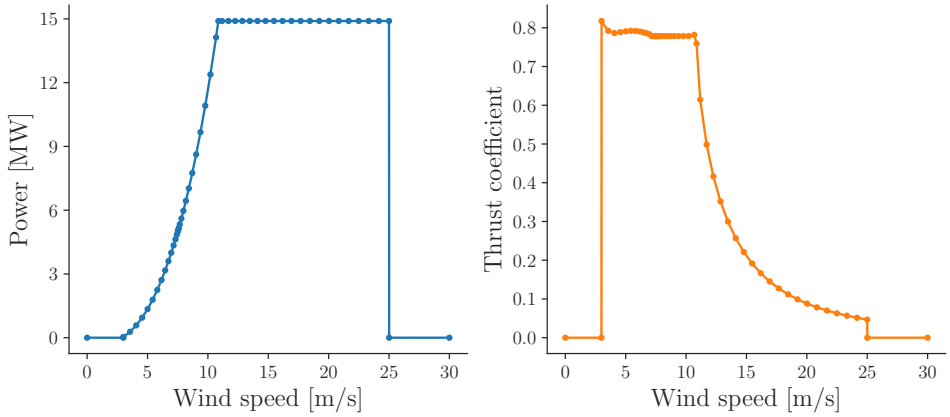


**Figure B.1.** Power and $C_T$ curves of the IEA37 15MW reference turbine [78].

**Table B.1.** Weibull shape and scale parameters for the Lillgrund site divided into 12 sectors. Data according to [25].

| Centered wind sector (°) | Frequency (%) | Weibull scale ($A$) | Weibull shape ($k$) |
|---|---|---|---|
| 0 | 3.8 | 4.5 | 1.69 |
| 30 | 4.5 | 4.7 | 1.78 |
| 60 | 0.4 | 3.0 | 1.82 |
| 90 | 2.8 | 7.2 | 1.70 |
| 120 | 8.3 | 8.8 | 1.97 |
| 150 | 7.5 | 8.2 | 2.49 |
| 180 | 9.9 | 8.4 | 2.72 |
| 210 | 14.8 | 9.5 | 2.70 |
| 240 | 14.3 | 9.2 | 2.88 |
| 270 | 17.0 | 9.9 | 3.34 |
| 300 | 12.6 | 10.3 | 2.84 |
| 330 | 4.1 | 6.7 | 2.23 |

**Table B.2.** Weibull shape and scale parameters for the Hollandse Kust (west) site VI divided into 12 sectors. Data according to [80].

| Centered wind sector (°) | Frequency (%) | Weibull scale ($A$) | Weibull shape ($k$) |
|---|---|---|---|
| 0 | 6.8 | 9.60 | 2.22 |
| 30 | 6.0 | 9.14 | 2.33 |
| 60 | 6.9 | 9.45 | 2.36 |
| 90 | 6.8 | 9.91 | 2.30 |
| 120 | 5.5 | 9.30 | 2.26 |
| 150 | 4.7 | 9.25 | 2.18 |
| 180 | 7.7 | 11.04 | 2.14 |
| 210 | 13.8 | 12.65 | 2.32 |
| 240 | 14.2 | 12.61 | 2.41 |
| 270 | 11.4 | 12.12 | 2.23 |
| 300 | 8.8 | 11.14 | 2.16 |
| 330 | 7.4 | 10.62 | 2.16 |

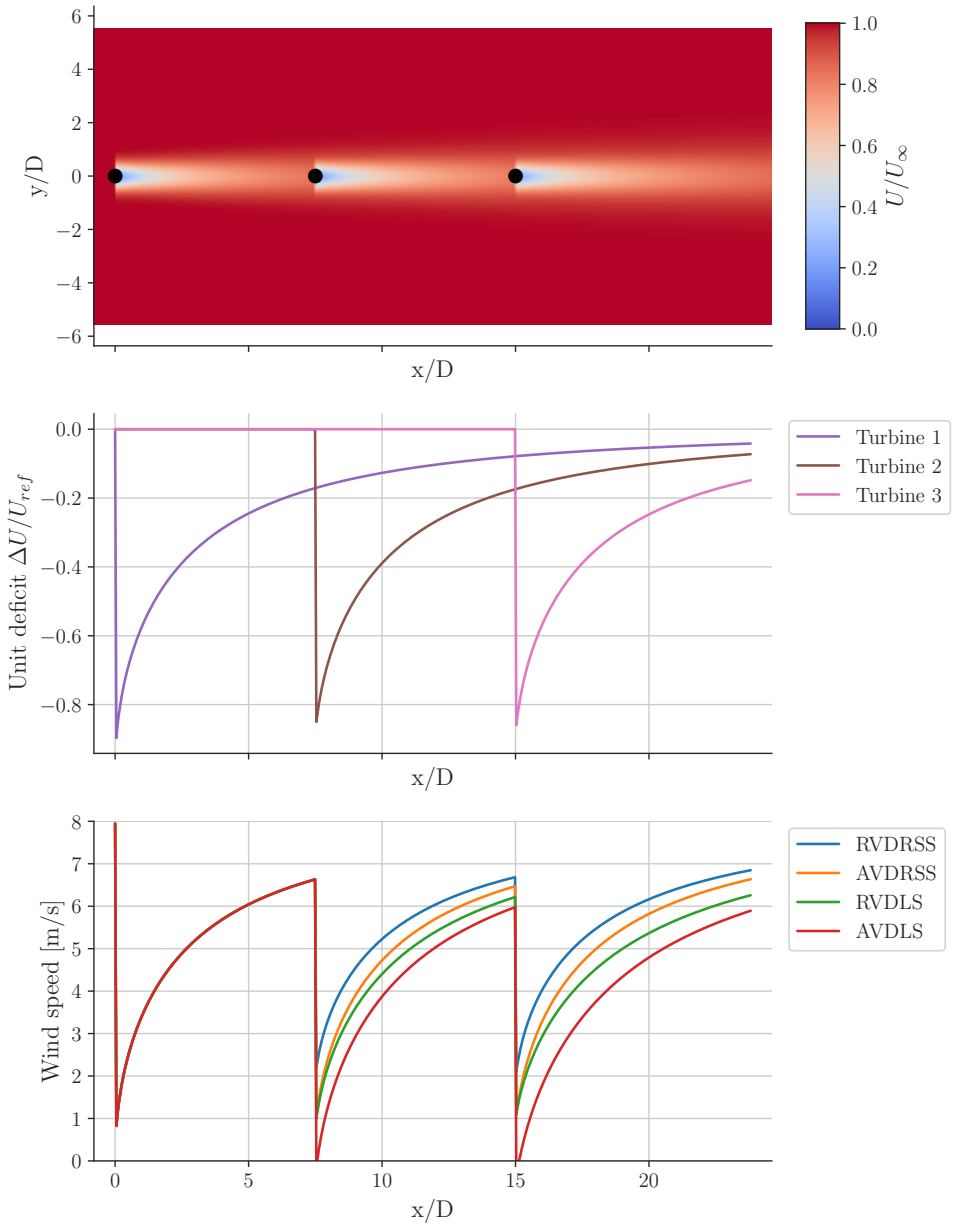


**Figure B.2.** Centerline velocity of a 3 wind turbine row varying superposition method.

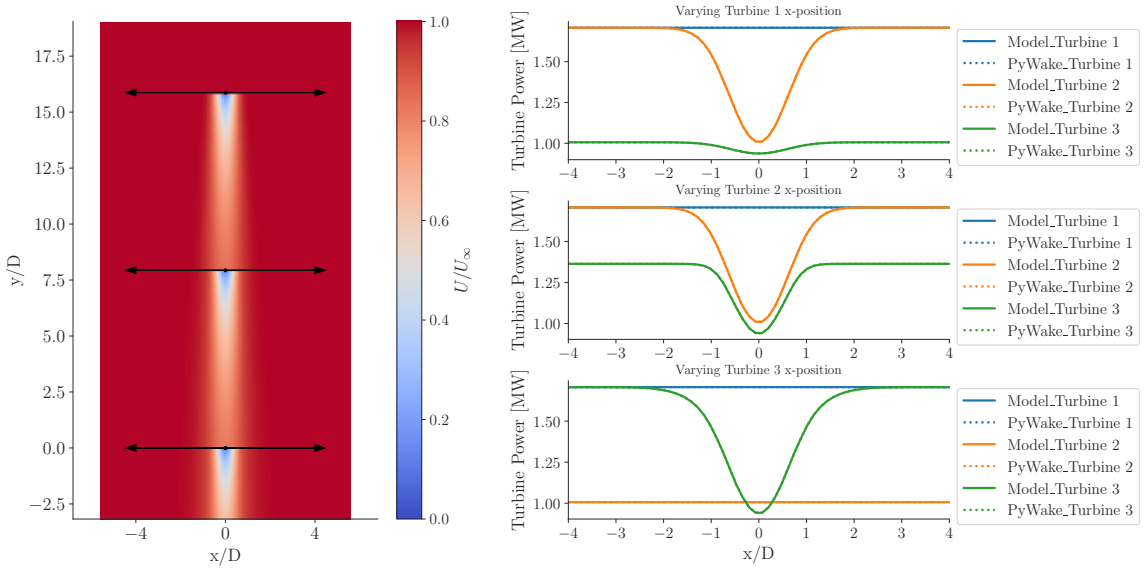


**Figure B.3.** Verification study of the new model with PyWake by repositioning 3 turbines in a row and comparing each turbine's power generation at 8 m/s.

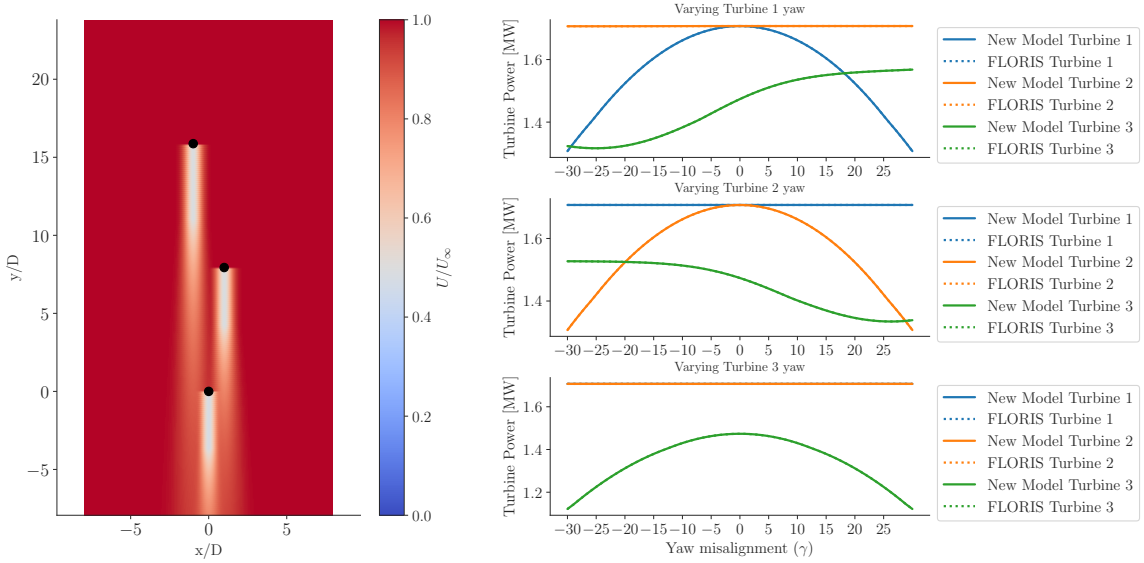


**Figure B.4.** Verification study of the new model with FLORIS by yawing 3 turbines in a slightly misaligned row and comparing each turbine's power generation at 8 m/s.

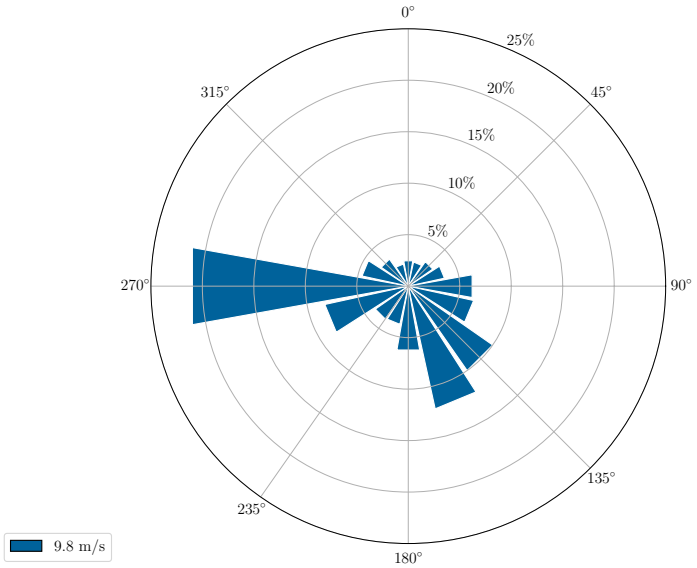


**Figure B.5.** Wind rose for the IEA37 test case sampled into 16 wind direction bins and 1 wind speed bin.

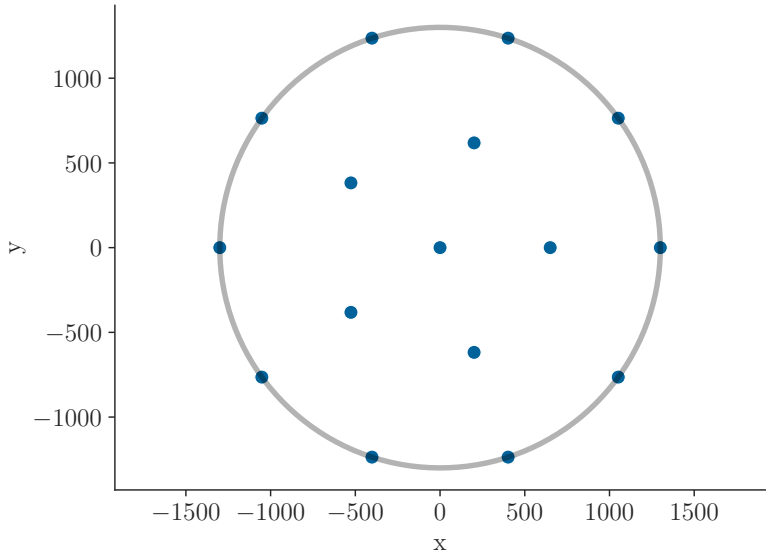


**Figure B.6.** Initial layout for the IEA37 test site with 16 wind turbines in a circular farm boundary.

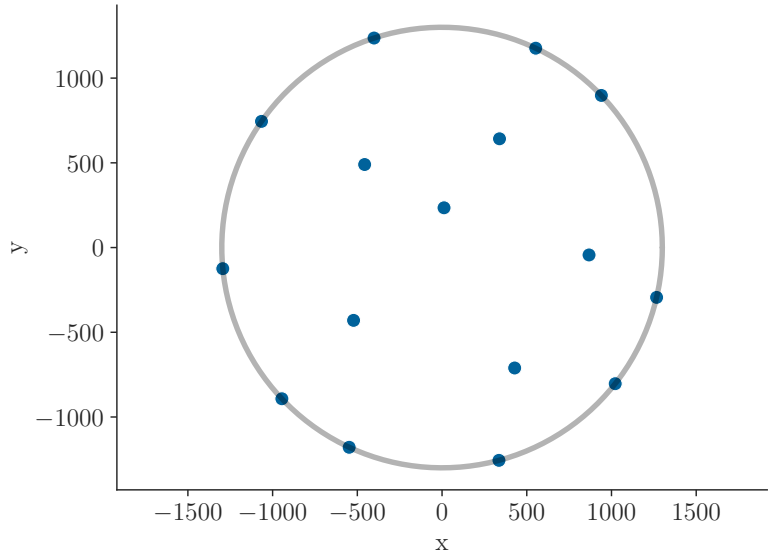


**Figure B.7.** Optimized layout for the IEA37 test site with 16 wind turbines in a circular farm boundary.

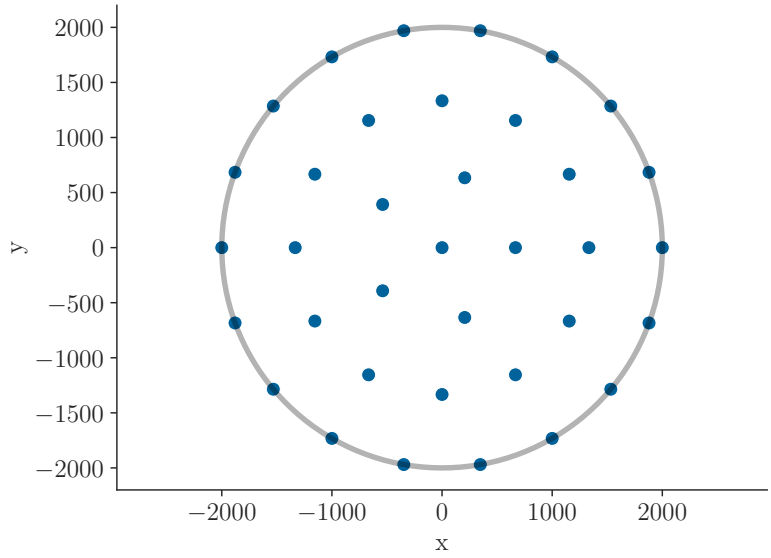


**Figure B.8.** Initial layout for the IEA37 test site with 36 wind turbines in a circular farm boundary.

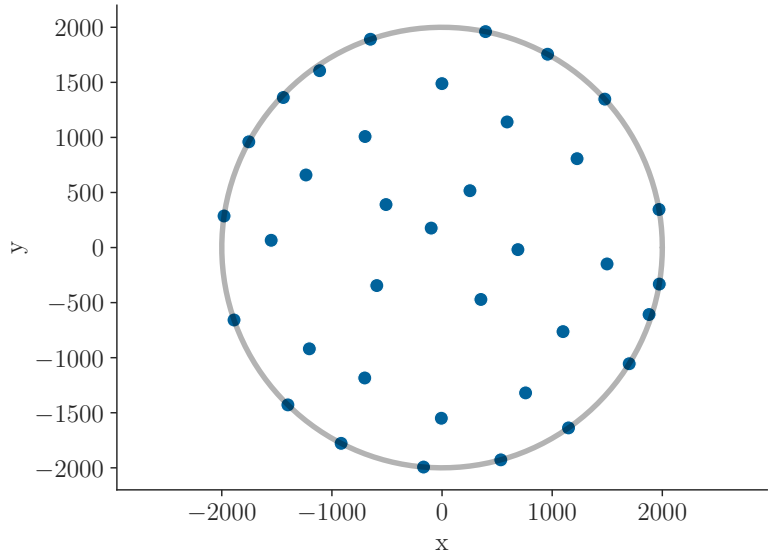


**Figure B.9.** Optimized layout for the IEA37 test site with 36 wind turbines in a circular farm boundary.

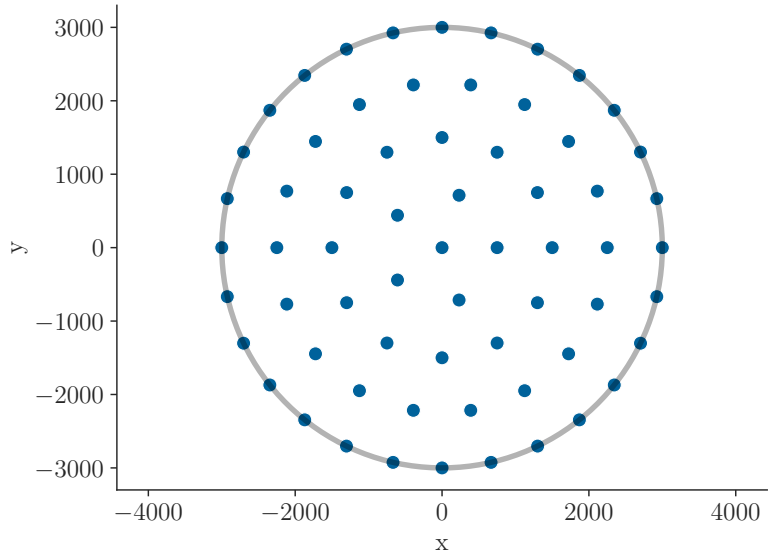


**Figure B.10.** Initial layout for the IEA37 test site with 64 wind turbines in a circular farm boundary.

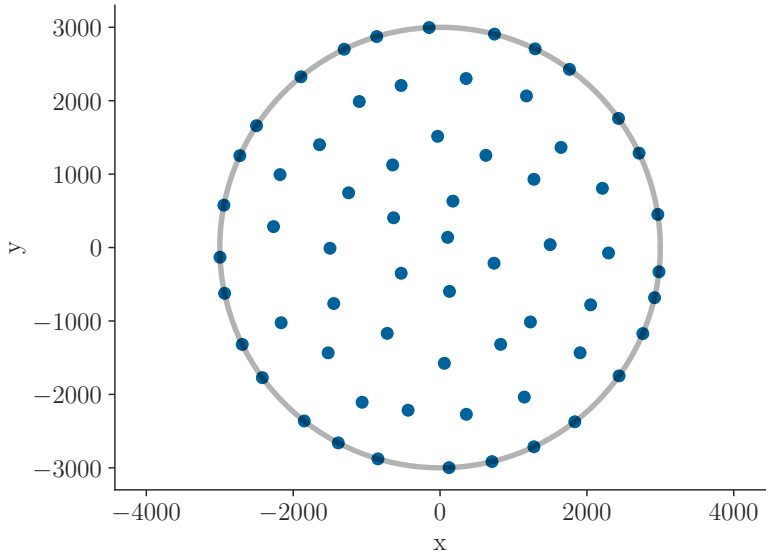


**Figure B.11.** Optimized layout for the IEA37 test site with 64 wind turbines in a circular farm boundary.

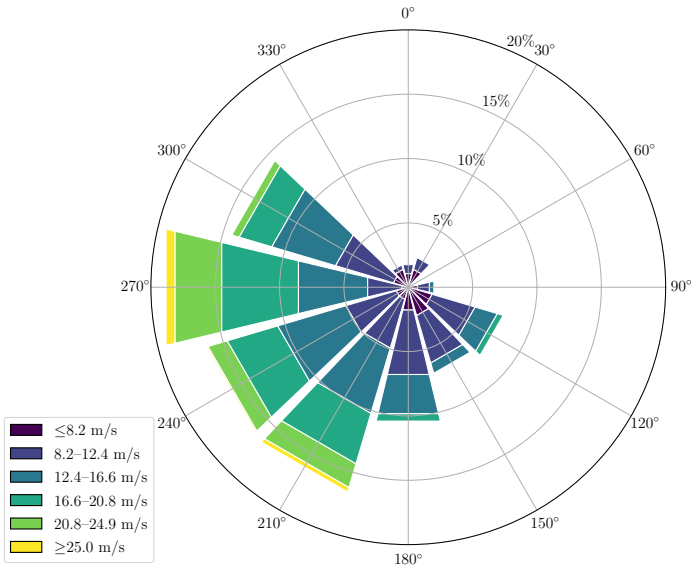


**Figure B.12.** Wind rose for the Lillgrund site sampled into 12 wind direction bins and 6 wind speed bins. Data according to [25].

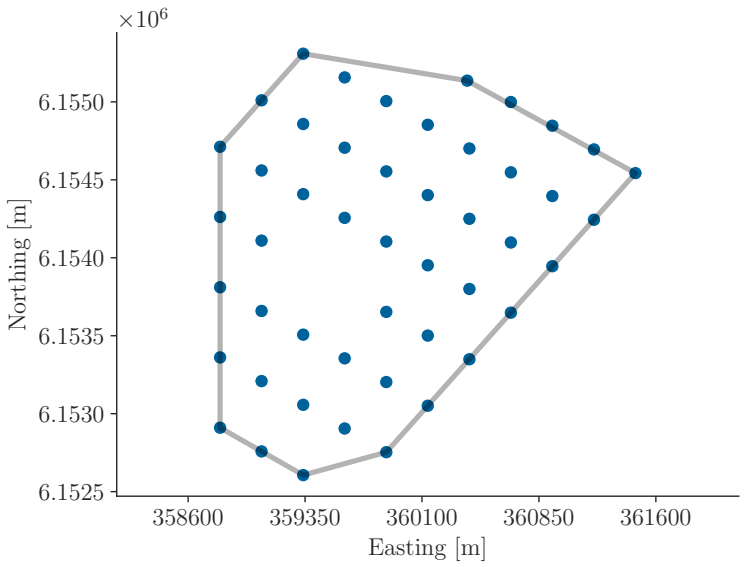


**Figure B.13.** The Lillgrund site with 48 wind turbines within the farm's boundaries.

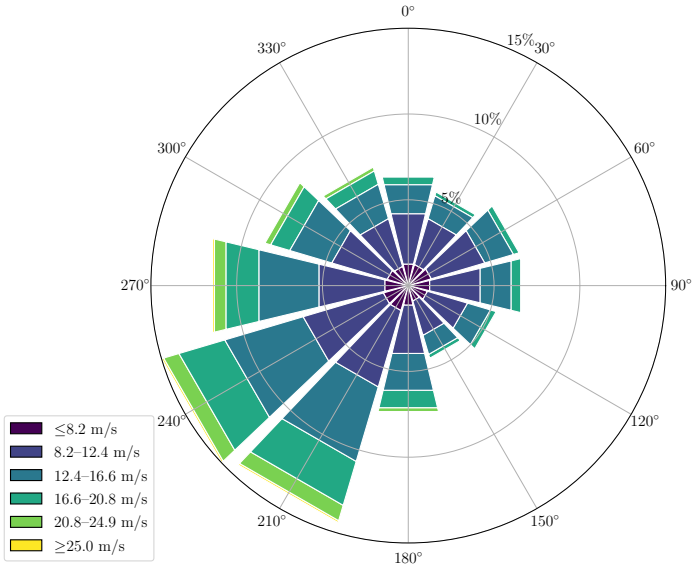


**Figure B.14.** Wind rose for the Hollandse Kust (West) site VI, sampled into 12 wind direction bins and 6 wind speed bins. Data according to [80].

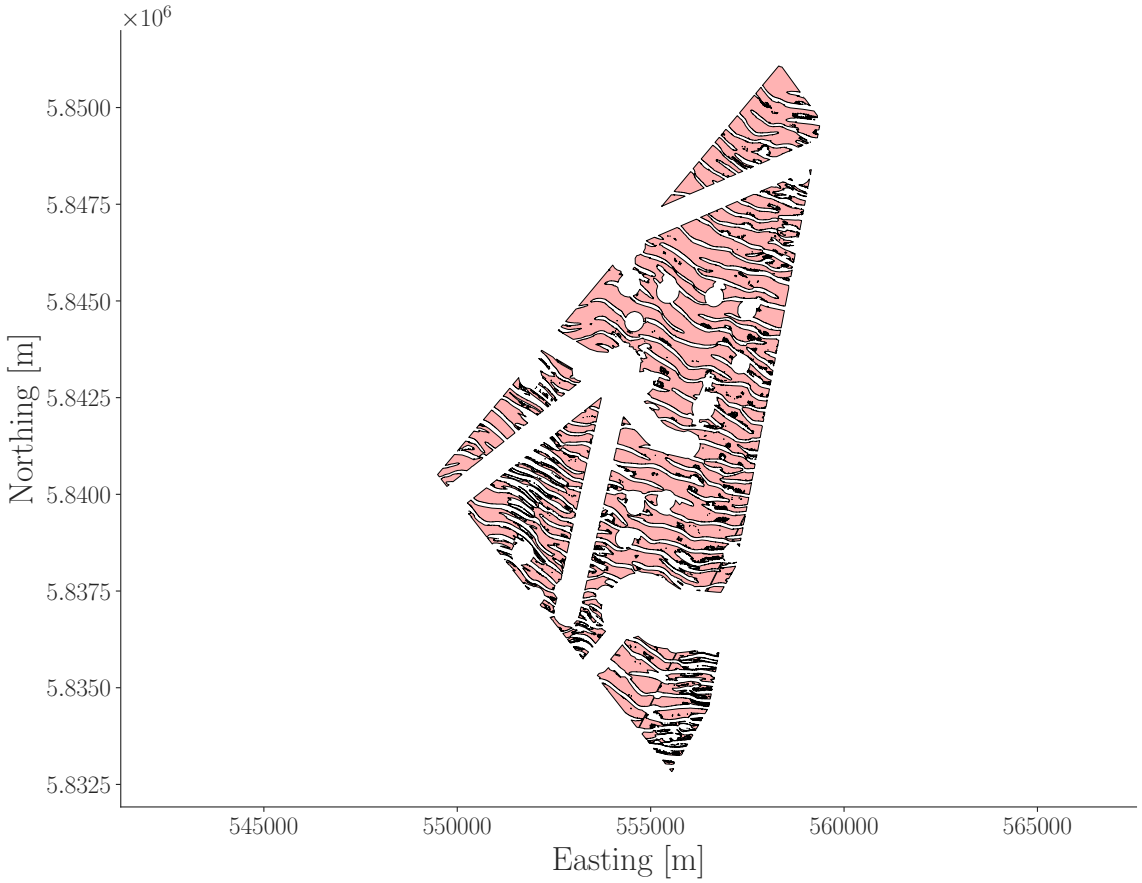


**Figure B.15.** The wind farm boundary of the Hollandse Kust (West) site VI.

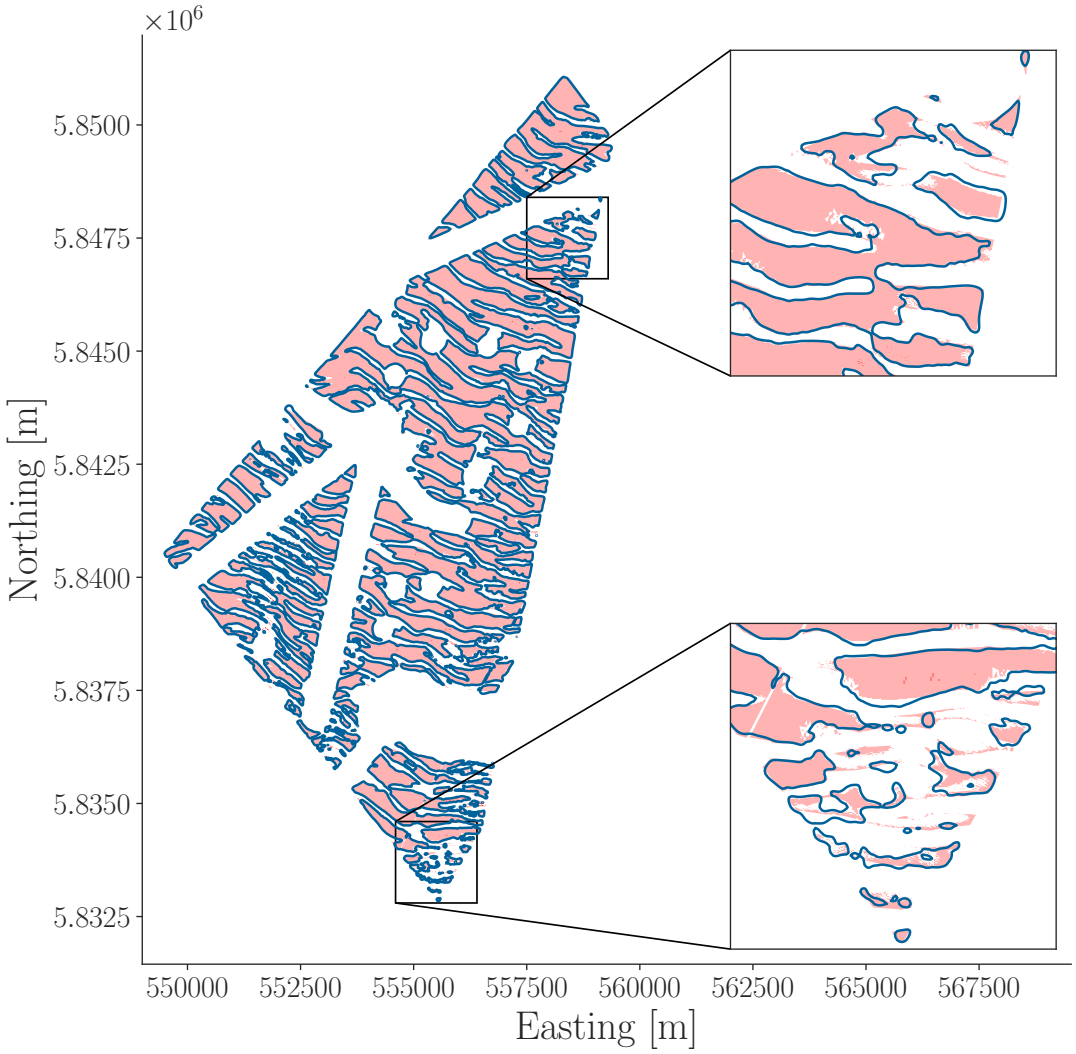


**Figure B.16.** Zero level set of the wind farm boundary constraint for the Hollandse Kust (West) site VI.

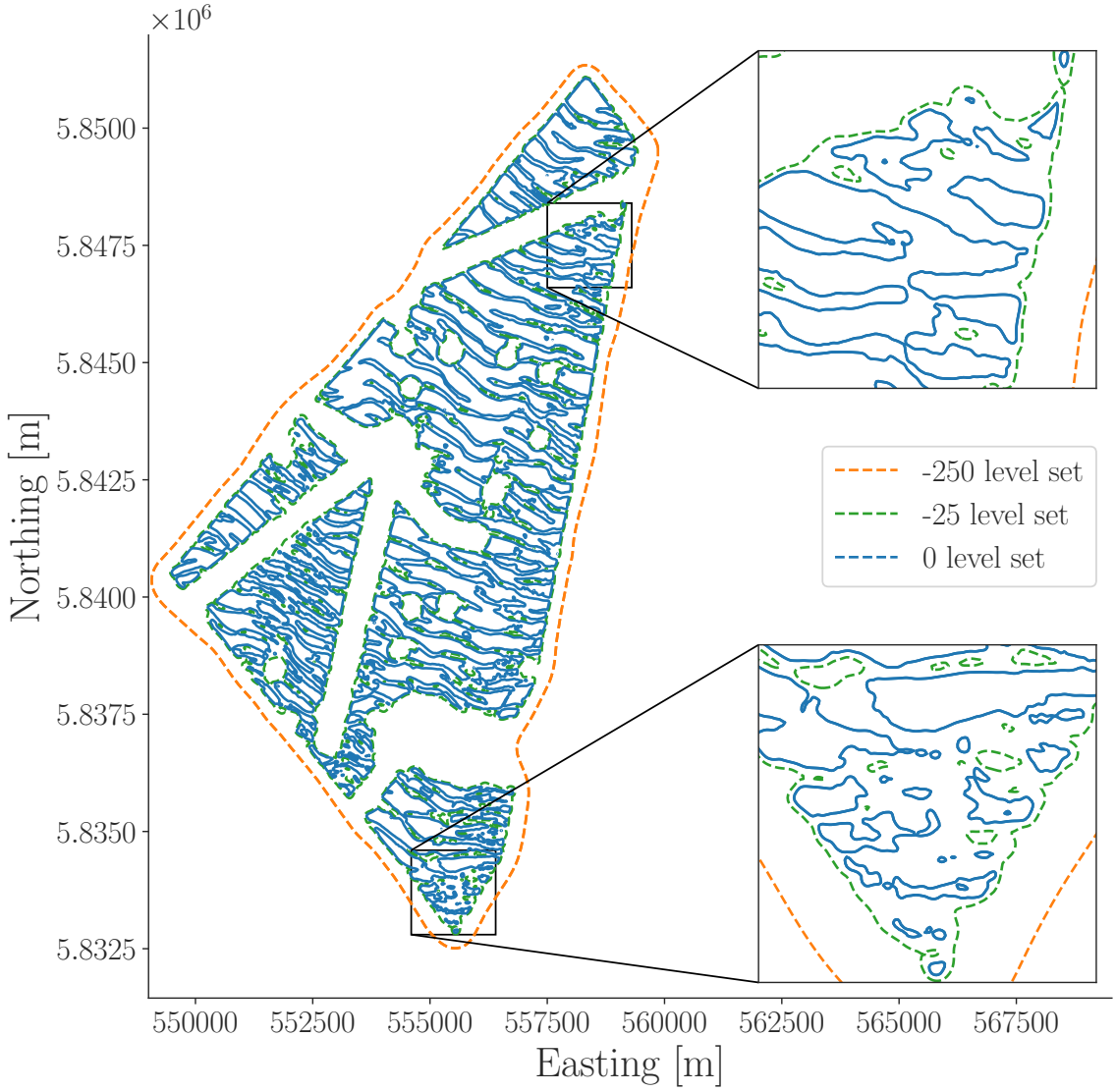


**Figure B.17.** The 250, 25, and 0 level sets of the non-interference constraint representing the boundary of the Hollandse Kust (West) site VI.

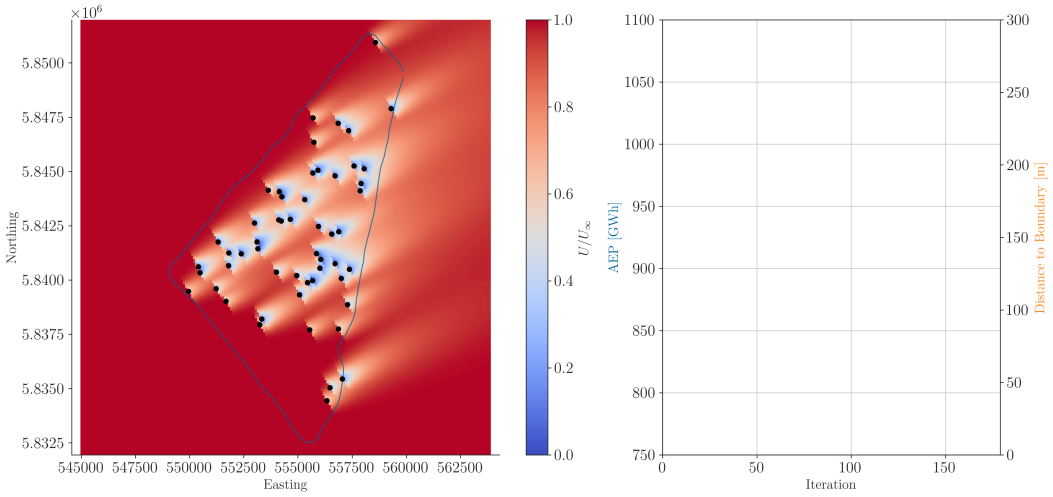


**Figure B.18.** The random initialization of 54 wind turbines in the HKW site with a relaxed boundary of 250 meters.

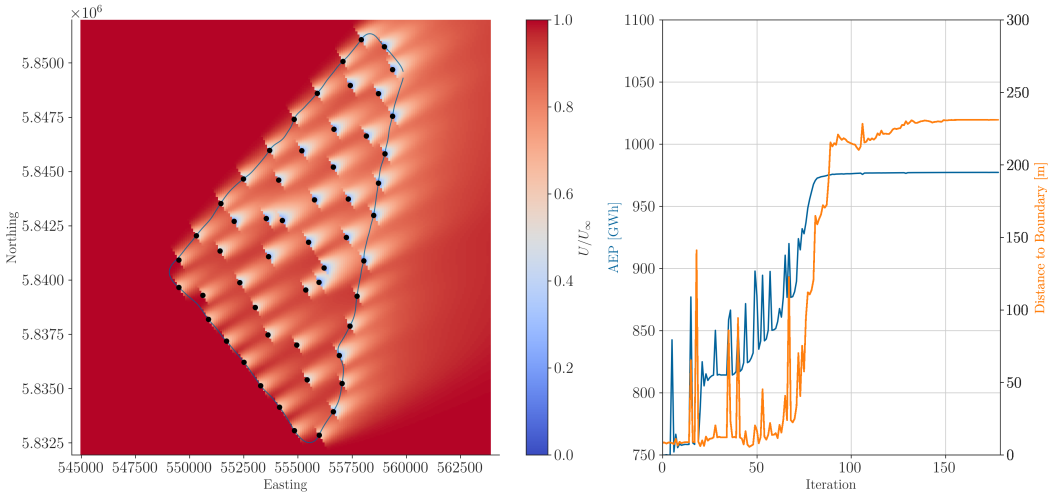


**Figure B.19.** Layout optimization 1 results of the HKW site with a relaxed boundary of 250 meters. AEP and median distance violation of the boundary constraint is plotted for each optimization iteration.

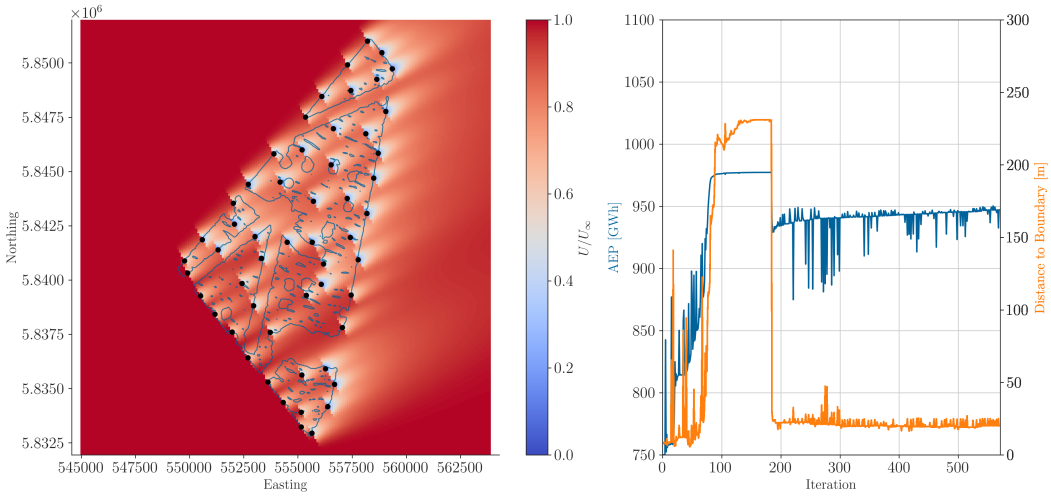


**Figure B.20.** Layout optimization 2 results of the HKW site with a relaxed boundary of 25 meters. AEP and median distance violation of the boundary constraint is plotted for each optimization iteration.

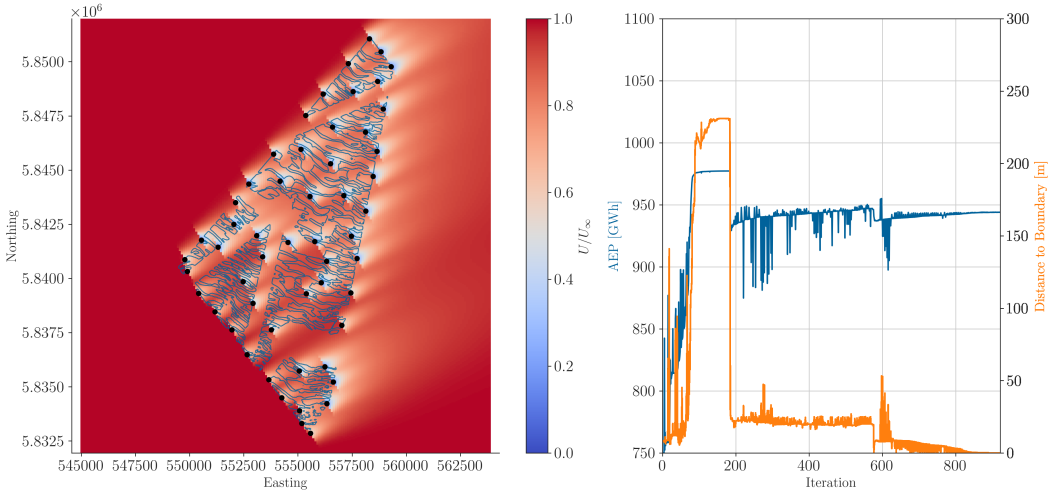


**Figure B.21.** Layout optimization 3 results of the HKW site. AEP and median distance violation of the boundary constraint is plotted for each optimization iteration.

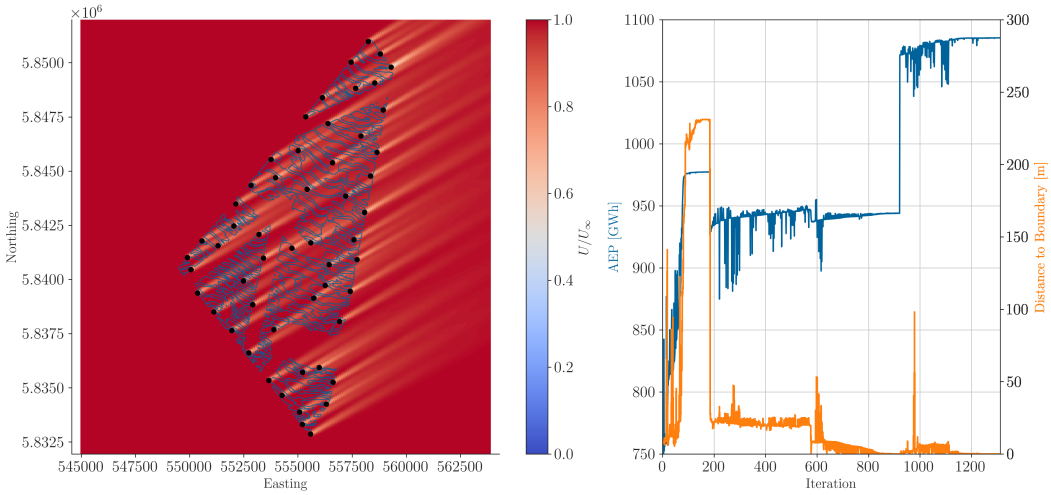


**Figure B.22.** Layout optimization 4 results of the HKW site. AEP and median distance violation of the boundary constraint is plotted for each optimization iteration.

# Bibliography


[1] Christos Bergeles, Andrew H. Gosline, Nikolay V. Vasilyev, Patrick J. Codd, Pedro J. del Nido, and Pierre E. Dupont. "Concentric Tube Robot Design and Optimization Based on Task and Anatomical Constraints". In: *IEEE Transactions on Robotics* 31.1 (2015), pp. 67–84. DOI: 10.1109/TRO.2014.2378431.

[2] Jui-Te Lin, Cédric Girerd, Jiayao Yan, John T. Hwang, and Tania K. Morimoto. "A Generalized Framework for Concentric Tube Robot Design Using Gradient-Based Optimization". In: *IEEE Transactions on Robotics* 38.6 (2022), pp. 3774–3791. DOI: 10.1109/TRO.2022.3180627.

[3] Benjamin J. Brelje, Josh L. Anibal, Anil Yildirim, Charles A. Mader, and Joaquim R. R. A. Martins. "Flexible Formulation of Spatial Integration Constraints in Aerodynamic Shape Optimization". In: *AIAA Scitech 2019 Forum*. 2019. DOI: 10.2514/6.2019-2355. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2019-2355. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2019-2355.

[4] J. Criado Risco, R. Valotta Rodrigues, M. Friis-Møller, J. Quick, M. Mølgaard Pedersen, and P.-E. Réthoré. "Gradient-based Wind Farm Layout Optimization With Inclusion And Exclusion Zones". In: *Wind Energy Science Discussions* 2023 (2023), pp. 1–24. DOI: 10.5194/wes-2023-5. URL: https://wes.copernicus.org/preprints/wes-2023-5/.

[5] Andrew PJ Stanley and Andrew Ning. "Coupled wind turbine design and layout optimization with nonhomogeneous wind turbines". In: *Wind Energy Science* 4.1 (2019), pp. 99–114.

[6] Georges M. Fadel and Margaret M. Wiecek. "Packing Optimization of Free-Form Objects in Engineering Design". In: *Optimized Packings with Applications*. Ed. by Giorgio Fasano and János D. Pintér. Cham: Springer International Publishing, 2015, pp. 37–66. ISBN: 978-3-319-18899-7. DOI: 10.1007/978-3-319-18899-7_3. URL: https://doi.org/10.1007/978-3-319-18899-7_3.

[7] Davide Cazzaro and David Pisinger. "Variable neighborhood search for large offshore wind farm layout optimization". In: *Computers and Operations Research* 138 (2022),

p. 105588. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2021.105588. URL: https://www.sciencedirect.com/science/article/pii/S0305054821003130.

[8] J. J. Thomas, N. F. Baker, P. Malisani, E. Quaeghebeur, S. S. Perez-Moreno, J. Jasa, C. Bay, F. Tilli, D. Bieniek, N. Robinson, A. P. J. Stanley, W. Holt, and A. Ning. "A Comparison of Eight Optimization Methods Applied to a Wind Farm Layout Optimization Problem". In: *Wind Energy Science Discussions* 2022 (2022), pp. 1–43. DOI: 10.5194/wes-2022-90. URL: https://wes.copernicus.org/preprints/wes-2022-90/.

[9] Nicholas F. Baker, Andrew P. Stanley, Jared J. Thomas, Andrew Ning, and Katherine Dykes. "Best Practices for Wake Model and Optimization Algorithm Selection in Wind Farm Layout Optimization". In: *AIAA Scitech 2019 Forum*. DOI: 10.2514/6.2019-0540. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2019-0540. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2019-0540.

[10] Felix Brandt. "The air cargo load planning problem". PhD thesis. Dissertation, Karlsruhe, Karlsruher Institut für Technologie (KIT), 2017, 2017.

[11] M.G. Mohanan and Ambuja Salgoankar. "A survey of robotic motion planning in dynamic environments". In: *Robotics and Autonomous Systems* 100 (2018), pp. 171–185. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2017.10.011. URL: https://www.sciencedirect.com/science/article/pii/S0921889017300313.

[12] Andrea Lodi, Silvano Martello, and Michele Monaci. "Two-dimensional packing problems: A survey". In: *European Journal of Operational Research* 141.2 (2002), pp. 241–252. ISSN: 0377-2217. DOI: https://doi.org/10.1016/S0377-2217(02)00123-6. URL: https://www.sciencedirect.com/science/article/pii/S0377221702001236.

[13] J. Cagan, K. Shimada, and S. Yin. "A survey of computational approaches to three-dimensional layout problems". In: *Computer-Aided Design* 34.8 (2002), pp. 597–611. ISSN: 0010-4485. DOI: https://doi.org/10.1016/S0010-4485(01)00109-9. URL: https://www.sciencedirect.com/science/article/pii/S0010448501001099.

[14] Giorgio Fasano. *Solving non-standard packing problems by global optimization and heuristics*. Springer, 2014.

[15] Justin S. Gray, John T. Hwang, Joaquim R. R. A. Martins, Kenneth T. Moore, and Bret A. Naylor. "OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization". In: *Structural and Multidisciplinary Optimization* 59 (4 2019), pp. 1075–1104. DOI: 10.1007/s00158-019-02211-z.

[16] Yuriy Stoyan, Tatiana Romanova, Alexander Pankratov, and Andrey Chugay. "Optimized Object Packings Using Quasi-Phi-Functions". In: *Optimized Packings with Applications*. Ed. by Giorgio Fasano and János D. Pintér. Cham: Springer Interna-

tional Publishing, 2015, pp. 265–293. ISBN: 978-3-319-18899-7. DOI: 10.1007/978-3-319-18899-7_13. URL: https://doi.org/10.1007/978-3-319-18899-7%5C_13.

[17] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. "Poisson Surface Reconstruction". In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP '06. Cagliari, Sardinia, Italy: Eurographics Association, 2006, pp. 61–70. ISBN: 3905673363.

[18] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. "Multi-Level Partition of Unity Implicits". In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 463–470. ISSN: 0730-0301. DOI: 10.1145/882262.882293. URL: https://doi.org/10.1145/882262.882293.

[19] F. Calakli and Gabriel Taubin. "SSD: Smooth Signed Distance Surface Reconstruction". In: *Computer Graphics Forum* 30 (Nov. 2011), pp. 1993–2002. DOI: 10.1111/j.1467-8659.2011.02058.x.

[20] Rebecca J Barthelmie and Sara C Pryor. "Climate change mitigation potential of wind energy". In: *Climate* 9.9 (2021), p. 136.

[21] Mark Z Jacobson, Cristina L Archer, and Willett Kempton. "Taming hurricanes with arrays of offshore wind turbines". In: *Nature climate change* 4.3 (2014), pp. 195–200.

[22] Andrew Ning and Derek Petch. "Integrated design of downwind land-based wind turbines using analytic gradients". In: *Wind Energy* 19.12 (2016), pp. 2137–2152. DOI: https://doi.org/10.1002/we.1972. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.1972. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/we.1972.

[23] Pieter Gebraad, Jared J. Thomas, Andrew Ning, Paul Fleming, and Katherine Dykes. "Maximization of the annual energy production of wind power plants by optimization of layout and yaw-based wake control". In: *Wind Energy* 20.1 (2017), pp. 97–107. DOI: https://doi.org/10.1002/we.1993. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.1993. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/we.1993.

[24] Wim Munters and Johan Meyers. "Dynamic Strategies for Yaw and Induction Control of Wind Farms Based on Large-Eddy Simulation and Optimization". In: *Energies* 11.1 (2018). ISSN: 1996-1073. DOI: 10.3390/en11010177. URL: https://www.mdpi.com/1996-1073/11/1/177.

[25] M. M. Pedersen and G. C. Larsen. "Integrated wind farm layout and control optimization". In: *Wind Energy Science* 5.4 (2020), pp. 1551–1566. DOI: 10.5194/wes-5-1551-2020. URL: https://wes.copernicus.org/articles/5/1551/2020/.

[26] Kaixuan Chen, Jin Lin, Yiwei Qiu, Feng Liu, and Yonghua Song. "Joint optimization of wind farm layout considering optimal control". In: *Renewable Energy* 182 (2022),

pp. 787–796. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2021.10.032. URL: https://www.sciencedirect.com/science/article/pii/S0960148121014890.

[27] G. Mosetti, C. Poloni, and B. Diviacco. "Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm". In: *Journal of Wind Engineering and Industrial Aerodynamics* 51.1 (1994), pp. 105–116. ISSN: 0167-6105. DOI: https://doi.org/10.1016/0167-6105(94)90080-9. URL: https://www.sciencedirect.com/science/article/pii/0167610594900809.

[28] Ju Feng, Wen Zhong Shen, and Chang Xu. "Multi-Objective Random Search Algorithm for Simultaneously Optimizing Wind Farm Layout and Number of Turbines". In: *Journal of Physics: Conference Series* 753.3 (Sept. 2016), p. 032011. DOI: 10.1088/1742-6596/753/3/032011. URL: https://dx.doi.org/10.1088/1742-6596/753/3/032011.

[29] Jared J. Thomas, Spencer McOmber, and Andrew Ning. "Wake expansion continuation: Multi-modality reduction in the wind farm layout optimization problem". In: *Wind Energy* 25.4 (2022), pp. 678–699. DOI: https://doi.org/10.1002/we.2692. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.2692. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/we.2692.

[30] David Guirguis, David A. Romero, and Cristina H. Amon. "Toward efficient optimization of wind farm layouts: Utilizing exact gradient information". In: *Applied Energy* 179 (2016), pp. 110–123. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2016.06.101. URL: https://www.sciencedirect.com/science/article/pii/S0306261916308765.

[31] Paul A. Fleming, Andrew Ning, Pieter M. O. Gebraad, and Katherine Dykes. "Wind plant system engineering through optimization of layout and yaw control". In: *Wind Energy* 19.2 (2016), pp. 329–344. DOI: https://doi.org/10.1002/we.1836. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.1836. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/we.1836.

[32] Jared J Thomas, Pieter MO Gebraad, and Andrew Ning. "Improving the FLORIS wind plant model for compatibility with gradient-based optimization". In: *Wind Engineering* 41.5 (2017), pp. 313–329.

[33] Sohail R. Reddy. "An efficient method for modeling terrain and complex terrain boundaries in constrained wind farm layout optimization". In: *Renewable Energy* 165 (2021), pp. 162–173. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2020.10.076. URL: https://www.sciencedirect.com/science/article/pii/S0960148120316426.

[34] Victor Gandarillas, Anugrah Jo Joshy, Mark Z Sperry, Alexander K Ivanov, and John T Hwang. "A graph-based methodology for constructing computational models that

automates adjoint-based sensitivity analysis". In: *Structural and Multidisciplinary Optimization* (2022).

[35] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gaël Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. "A Survey of Surface Reconstruction from Point Clouds". In: *Computer Graphics Forum* 36.1 (2017), pp. 301–329. DOI: https://doi.org/10.1111/cgf.12802. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12802. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12802.

[36] Zhangjin Huang, Yuxin Wen, Zihao Wang, Jinjuan Ren, and Kui Jia. *Surface Reconstruction from Point Clouds: A Survey and a Benchmark*. 2022. DOI: 10.48550/ARXIV.2205.02413. URL: https://arxiv.org/abs/2205.02413.

[37] J.C. Carr, W.R. Fright, and R.K. Beatson. "Surface interpolation with radial basis functions for medical imaging". In: *IEEE Transactions on Medical Imaging* 16.1 (1997), pp. 96–107. DOI: 10.1109/42.552059.

[38] Hong-Kai Zhao, S. Osher, and R. Fedkiw. "Fast surface reconstruction using the level set method". In: *Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision*. 2001, pp. 194–201. DOI: 10.1109/VLSM.2001.938900.

[39] J. Davis, S.R. Marschner, M. Garr, and M. Levoy. "Filling holes in complex surfaces using volumetric diffusion". In: *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*. 2002, pp. 428–441. DOI: 10.1109/TDPVT.2002.1024098.

[40] Huong Quynh Dinh, G. Turk, and G. Slabaugh. "Reconstructing surfaces using anisotropic basis functions". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.* Vol. 2. 2001, 606–613 vol.2. DOI: 10.1109/ICCV.2001.937682.

[41] Maodong Pan, Weihua Tong, and Falai Chen. "Phase-Field Guided Surface Reconstruction Based on Implicit Hierarchical B-Splines". In: *Comput. Aided Geom. Des.* 52.C (Mar. 2017), pp. 154–169. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2017.03.009. URL: https://doi.org/10.1016/j.cagd.2017.03.009.

[42] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. "Surface Reconstruction from Unorganized Points". In: *SIGGRAPH Comput. Graph.* 26.2 (July 1992), pp. 71–78. ISSN: 0097-8930. DOI: 10.1145/142920.134011. URL: https://doi.org/10.1145/142920.134011.

[43] Kun Zhou, Minmin Gong, Xin Huang, and Baining Guo. "Data-Parallel Octrees for Surface Reconstruction". In: *IEEE transactions on visualization and computer graphics* 17 (May 2010). DOI: 10.1109/TVCG.2010.75.

[44] Yizhi Tang and Jieqing Feng. "Multi-scale surface reconstruction based on a curvature-adaptive signed distance field". In: *Computers and Graphics* 70 (2018). CAD/Graphics 2017, pp. 28–38. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2017.07.015. URL: https://www.sciencedirect.com/science/article/pii/S0097849317301085.

[45] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. "Reconstruction and Representation of 3D Objects with Radial Basis Functions". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, pp. 67–76. ISBN: 158113374X. DOI: 10.1145/383259.383266. URL: https://doi.org/10.1145/383259.383266.

[46] Jason Hicken and Sharanjeet Kaur. "An Explicit Level-Set Formula to Approximate Geometries". In: Jan. 2022. DOI: 10.2514/6.2022-1862.

[47] Tolga Tasdizen, Ross Whitaker, Paul Burchard, and Stanley Osher. *Geometric Surface Processing via Normal Maps*. 2002.

[48] Bjarke Jakobsen, J Andreas Bærentzen, and Niels Jørgen Christensen. "Variational Volumetric Surface Reconstruction from Unorganized Points." In: *VG@ Eurographics*. 2007, pp. 65–72.

[49] Peter Sibley and Gabriel Taubin. "Vectorfield Isosurface-Based Reconstruction from Oriented Points". In: Jan. 2005, p. 29. DOI: 10.1145/1187112.1187146.

[50] J. Manson, G. Petrova, and S. Schaefer. "Streaming Surface Reconstruction Using Wavelets". In: *Computer Graphics Forum* 27.5 (2008), pp. 1411–1420. DOI: https://doi.org/10.1111/j.1467-8659.2008.01281.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2008.01281.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2008.01281.x.

[51] Michael Kazhdan. "Reconstruction of Solid Models from Oriented Point Sets". In: *Proceedings of the Third Eurographics Symposium on Geometry Processing*. SGP '05. Vienna, Austria: Eurographics Association, 2005, 73–es. ISBN: 390567324X.

[52] Alexander Belyaev, Pierre-Alain Fayolle, and Alexander Pasko. "Signed Lp-distance fields". In: *Computer-Aided Design* 45.2 (2013). Solid and Physical Modeling 2012, pp. 523–528. ISSN: 0010-4485. DOI: https://doi.org/10.1016/j.cad.2012.10.035. URL: https://www.sciencedirect.com/science/article/pii/S0010448512002400.

[53] Greg Turk and James F. O'Brien. "Modelling with Implicit Surfaces That Interpolate". In: *ACM Trans. Graph.* 21.4 (Oct. 2002), pp. 855–873. ISSN: 0730-0301. DOI: 10.1145/571647.571650. URL: https://doi.org/10.1145/571647.571650.

[54] Michael Kazhdan and Hugues Hoppe. “Screened Poisson Surface Reconstruction”. In: *ACM Trans. Graph.* 32.3 (July 2013). ISSN: 0730-0301. DOI: 10.1145/2487228.2487237. URL: https://doi.org/10.1145/2487228.2487237.

[55] Stanley Osher and James A Sethian. “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations”. In: *Journal of Computational Physics* 79.1 (1988), pp. 12–49. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(88)90002-2. URL: https://www.sciencedirect.com/science/article/pii/0021999188900022.

[56] Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003. DOI: 10.1017/CBO9780511543241.

[57] S. Andrew Ning, Rick Damiani, and Patrick J. Moriarty. “Objectives and Constraints for Wind Turbine Optimization”. In: *Journal of Solar Energy Engineering* 136.4 (June 2014). 041010. ISSN: 0199-6231. DOI: 10.1115/1.4027693. eprint: https://asmedigitalcollection.asme.org/solarenergyengineering/article-pdf/136/4/041010/6325637/sol_136_04_041010.pdf. URL: https://doi.org/10.1115/1.4027693.

[58] Ahmadreza Vasel-Be-Hagh and Cristina L. Archer. “Wind farm hub height optimization”. In: *Applied Energy* 195 (2017), pp. 905–921. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2017.03.089. URL: https://www.sciencedirect.com/science/article/pii/S0306261917303306.

[59] Andreas Wolf Ciavarra, Rafael Valotta Rodrigues, Katherine Dykes, and Pierre-Elouan Réthoré. “Wind farm optimization with multiple hub heights using gradient-based methods”. In: *Journal of Physics: Conference Series* 2265.2 (May 2022), p. 022012. DOI: 10.1088/1742-6596/2265/2/022012. URL: https://dx.doi.org/10.1088/1742-6596/2265/2/022012.

[60] Wing Yin Kwong, Peter Yun Zhang, David Romero, Joaquin Moran, Michael Morgenroth, and Cristina Amon. “Multi-Objective Wind Farm Layout Optimization Considering Energy Generation and Noise Propagation With NSGA-II”. In: *Journal of Mechanical Design* 136.9 (July 2014). 091010. ISSN: 1050-0472. DOI: 10.1115/1.4027847. eprint: https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/136/9/091010/6225598/md_136_09_091010.pdf. URL: https://doi.org/10.1115/1.4027847.

[61] J.A. Vitulli, G.C. Larsen, M.M. Pedersen, S. Ott, and M. Friis-Møller. “Optimal open loop wind farm control”. In: *Journal of Physics: Conference Series* 1256.1 (July 2019), p. 012027. DOI: 10.1088/1742-6596/1256/1/012027. URL: https://dx.doi.org/10.1088/1742-6596/1256/1/012027.

[62] A. P. J. Stanley, C. Bay, R. Mudafort, and P. Fleming. "Fast yaw optimization for wind plant wake steering using Boolean yaw angles". In: *Wind Energy Science* 7.2 (2022), pp. 741–757. DOI: 10.5194/wes-7-741-2022. URL: https://wes.copernicus.org/articles/7/741/2022/.

[63] A. P. J. Stanley and A. Ning. "Massive simplification of the wind farm layout optimization problem". In: *Wind Energy Science* 4.4 (2019), pp. 663–676. DOI: 10.5194/wes-4-663-2019. URL: https://wes.copernicus.org/articles/4/663/2019/.

[64] J J Thomas and A Ning. "A method for reducing multi-modality in the wind farm layout optimization problem". In: *Journal of Physics: Conference Series* 1037.4 (June 2018), p. 042012. DOI: 10.1088/1742-6596/1037/4/042012. URL: https://dx.doi.org/10.1088/1742-6596/1037/4/042012.

[65] Le Chen, Chris Harding, Anupam Sharma, and Erin MacDonald. "Modeling noise and lease soft costs improves wind farm design and cost-of-energy predictions". In: *Renewable Energy* 97 (2016), pp. 849–859. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2016.05.045. URL: https://www.sciencedirect.com/science/article/pii/S0960148116304566.

[66] Simão S Rodrigues and André C Marta. "On addressing noise constraints in the design of wind turbine blades". In: *Structural and Multidisciplinary Optimization* 50 (2014), pp. 489–503.

[67] Sami Yamani Douzi Sorkhabi, David A. Romero, Gary Kai Yan, Michelle Dao Gu, Joaquin Moran, Michael Morgenroth, and Cristina H. Amon. "The impact of land use constraints in multi-objective energy-noise wind farm layout optimization". In: *Renewable Energy* 85 (2016), pp. 359–370. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2015.06.026. URL: https://www.sciencedirect.com/science/article/pii/S0960148115300495.

[68] Katherine Dykes, Rick Damiani, Owen Roberts, and Eric Lantz. "Analysis of Ideal Towers for Tall Wind Applications". In: *2018 Wind Energy Symposium*. DOI: 10.2514/6.2018-0999. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-0999. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2018-0999.

[69] Ali C. Kheirabadi and Ryozo Nagamune. "A quantitative review of wind farm control with the objective of wind farm power maximization". In: *Journal of Wind Engineering and Industrial Aerodynamics* 192 (2019), pp. 45–73. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2019.06.015. URL: https://www.sciencedirect.com/science/article/pii/S0167610519305240.

[70] Ryan Nash, Reza Nouri, and Ahmad Vasel-Be-Hagh. "Wind turbine wake control strategies: A review and concept proposal". In: *Energy Conversion and Management* 245 (2021), p. 114581. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.

enconman.2021.114581. URL: https://www.sciencedirect.com/science/article/pii/S0196890421007573.

[71] Carl De Boor. "On calculating with B-splines". In: *Journal of Approximation theory* 6.1 (1972), pp. 50–62.

[72] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. "Edge-aware point set resampling". In: *ACM transactions on graphics (TOG)* 32.1 (2013), pp. 1–12.

[73] Christopher Silva, Wayne R. Johnson, Eduardo Solis, Michael D. Patterson, and Kevin R. Antcliff. "VTOL Urban Air Mobility Concept Vehicles for Technology Development". In: *2018 Aviation Technology, Integration, and Operations Conference*. DOI: 10.2514/6.2018-3847. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-3847. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2018-3847.

[74] Matthew P Reed, Ulrich Raschke, Rishi Tirumali, and Matthew B Parkinson. "Developing and implementing parametric human body shape models in ergonomics software". In: *Proceedings of the 3rd international digital human modeling conference, Tokyo*. 2014.

[75] Patrick Sears and Pierre Dupont. "A Steerable Needle Technology Using Curved Concentric Tubes". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006, pp. 2850–2856. DOI: 10.1109/IROS.2006.282072.

[76] Robert J. Webster, Allison M. Okamura, and Nah J. Cowan. "Toward Active Cannulas: Miniature Snake-Like Surgical Robots". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006, pp. 2857–2863. DOI: 10.1109/IROS.2006.282073.

[77] J Jonkman, S Butterfield, W Musial, and G Scott. "Definition of a 5-MW Reference Wind Turbine for Offshore System Development". In: (Feb. 2009). DOI: 10.2172/947422. URL: https://www.osti.gov/biblio/947422.

[78] Evan Gaertner, Jennifer Rinker, Latha Sethuraman, Frederik Zahle, Benjamin Anderson, Garrett Barter, Nikhar Abbas, Fanzhong Meng, Pietro Bortolotti, Witold Skrzypinski, George Scott, Roland Feil, Henrik Bredmose, Katherine Dykes, Matt Sheilds, Christopher Allen, and Anthony Viselli. *Definition of the IEA 15-Megawatt Offshore Reference Wind Turbine*. Tech. rep. International Energy Agency, 2020. URL: https://www.nrel.gov/docs/fy20osti/75698.pdf.

[79] Katherine L Dykes, Frederik Zahle, Karl Merz, Mike McWilliam, and Pietro Bortolotti. "IEA Wind Task 37: Systems Modeling Framework and Ontology for Wind Turbines and Plants". In: (Aug. 2017). URL: https://www.osti.gov/biblio/1375625.

[80] Helena Jane Hunt. *Hollandse Kust (West) Wind Farm Zone Certification Report Metocean Conditions*. DNV GL Energy. 2020, pp. 134, 411.

[81] Niels Otto Jensen. *A note on wind generator interaction*. Vol. 2411. Citeseer, 1983.

[82] Majid Bastankhah and Fernando Porté-Agel. "A new analytical model for wind-turbine wakes". In: *Renewable Energy* 70 (2014). Special issue on aerodynamics of offshore wind energy systems and wakes, pp. 116–123. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2014.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0960148114000317.

[83] Majid Bastankhah and Fernando Porté-Agel. "Experimental and theoretical study of wind turbine wakes in yawed conditions". In: *Journal of Fluid Mechanics* 806 (2016), pp. 506–541. DOI: 10.1017/jfm.2016.595.

[84] A. Crespo and J. Herna´ndez. "Turbulence characteristics in wind-turbine wakes". In: *Journal of Wind Engineering and Industrial Aerodynamics* 61.1 (1996), pp. 71–85. ISSN: 0167-6105. DOI: https://doi.org/10.1016/0167-6105(95)00033-X. URL: https://www.sciencedirect.com/science/article/pii/016761059500033X.

[85] P. B. S. Lissaman. "Energy Effectiveness of Arbitrary Arrays of Wind Turbines". In: *Journal of Energy* 3.6 (1979), pp. 323–328. DOI: 10.2514/3.62441. eprint: https://doi.org/10.2514/3.62441. URL: https://doi.org/10.2514/3.62441.

[86] Amin Niayifar and Fernando Porté-Agel. "Analytical Modeling of Wind Farms: A New Approach for Power Prediction". In: *Energies* 9.9 (2016). ISSN: 1996-1073. DOI: 10.3390/en9090741. URL: https://www.mdpi.com/1996-1073/9/9/741.

[87] I Katic, Jørgen Højstrup, and Niels Otto Jensen. "A simple model for cluster efficiency". In: *European wind energy association conference and exhibition*. Vol. 1. A. Raguzzi Rome, Italy. 1986, pp. 407–410.

[88] S Voutsinas, K Rados, and A Zervos. "On the analysis of wake effects in wind parks". In: *Wind Engineering* (1990), pp. 204–219.

[89] Fernando Porté-Agel, Majid Bastankhah, and Sina Shamsoddin. "Wind-turbine and wind-farm flows: A review". In: *Boundary-layer meteorology* 174 (2020), pp. 1–59.

[90] Jaime Liew, Georg Raimund Pirrung, and Albert Meseguer Urbán. "Effect of varying fidelity turbine models on wake loss prediction". In: *Journal of Physics: Conference Series* 1618.6 (Sept. 2020), p. 062002. DOI: 10.1088/1742-6596/1618/6/062002. URL: https://dx.doi.org/10.1088/1742-6596/1618/6/062002.

[91] Paul Fleming, Pieter M.O. Gebraad, Sang Lee, Jan-Willem van Wingerden, Kathryn Johnson, Matt Churchfield, John Michalakes, Philippe Spalart, and Patrick Moriarty.

"Simulation comparison of wake mitigation control strategies for a two-turbine case". In: *Wind Energy* 18.12 (2015), pp. 2135–2143. DOI: https://doi.org/10.1002/we.1810. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.1810. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/we.1810.

[92] Ministry of Economic Affairs and Climate Policy. "Shell and Eneco receive permit for Hollandse Kust (west) Site VI offshore wind farm". In: *Netherlands Enterprise Agency: Sustainability* (Dec. 2022). (Accessed: May 20th, 2023). URL: https://english.rvo.nl/news/shell-and-eneco-receive-permit-hollandse-kust-west-site-vi-offshore-wind-farm.