Polynomial chaos to efficiently compute the annual energy production in wind farm layout optimization

Andrés Santiago Padrón¹, Jared Thomas², Andrew P. J. Stanley², Juan J. Alonso¹, and Andrew Ning²

¹Department of Aeronautics & Astronautics, Stanford University, Stanford, CA, 94305, USA
²Department of Mechanical Engineering, Brigham Young University, Provo, UT, 84602, USA

Correspondence to: Andrew Ning (aning@byu.edu)

Abstract.

In this paper, we develop computationally-efficient techniques to calculate statistics used in wind farm optimization with the goal of enabling the use of higher-fidelity models and larger wind farm optimization problems. We apply these techniques to maximize the Annual Energy Production (AEP) of a wind farm by optimizing the position of the individual wind turbines. The AEP (a statistic itself) is the expected power produced by the wind farm over a period of one year subject to uncertainties in the wind conditions (wind direction and wind speed) that are described with empirically-determined probability distributions. To compute the AEP of the wind farm, we use a wake model to simulate the power at different input conditions composed of wind direction and wind speed pairs. We use polynomial chaos (PC), an uncertainty quantification method, to construct a polynomial approximation of the power over the entire stochastic space and to efficiently (using as few simulations as possible) compute the expected power (AEP). We explore both regression and quadrature approaches to compute the PC coefficients. PC based on regression is significantly more efficient than the rectangle rule (the method most commonly used to compute the expected power). With PC based on regression, we have reduced by as much as an order of magnitude the number of simulations required to accurately compute the AEP, thus enabling the use of more expensive, higher-fidelity models or larger wind farm optimizations. We show how the PC method can be used to efficiently compute the gradients of the AEP and we perform a large suite of gradient-based optimizations with different initial turbine locations and with different numbers of samples to compute the AEP. The optimizations with PC based on regression result in optimized layouts that produce the same AEP as the optimized layouts found with the rectangle rule but using only one-third of the samples. Furthermore, for the same number of samples, the AEP of the optimal layouts found with PC is 1% higher than the AEP of the layouts found with the rectangle rule.

1 Introduction

In 2015, wind energy growth accounted for almost half of the global electricity supply growth. In the United States, it accounted for 41% of new power capacity, raising the wind energy supply to 4.7% of the total electricity generated in 2015 and on target to reach 10% by 2020 (U.S. Department of Energy, 2015; AWEA, 2016; GWEC, 2016). Most of the current and upcoming wind energy comes from large turbines (greater than 1 MW) situated in clusters—wind farms. A problem with putting turbines
Figure 1. Examples of applications that require many model evaluations. In deterministic optimization (a), we evaluate the model at different values of the design variables while searching for an optimum response. In uncertainty quantification (b), we query the model multiple times at different instances of the uncertain variables to generate an ensemble of responses from which we can compute statistics and probabilities of the model response. In optimization under uncertainty (OUU), we optimize a statistic. OUU is computationally expensive as it requires many model evaluations because of the nested loop in the problem formulation (c).

Together in confined spaces is that they operate in the wakes of other turbines, i.e., in regions of reduced speed and increased turbulence. This leads to an underproduction of power and decreased (10–20%) energy output for the farm (Barthelmie et al., 2007, 2009; Briggs, 2013) when compared to ideal conditions. This loss in energy capture results in millions of dollars of loss for operators and investors and increased economic uncertainty for new installations. Many current wind farms have grid-like layouts, where wind turbines are aligned in rows, which further exacerbate the wake losses. By optimizing the layout of the wind farm, the wake losses can be minimized, with a corresponding increase in energy production and revenue.

Wind farm optimization is a complex, multi-disciplinary and high-dimensional problem. The wind farm may contain dozens or even hundreds of wind turbines, where each turbine may be parameterically described using several design variables. Furthermore, the wind conditions (wind direction, wind speed, wind turbulence, etc.) are stochastic (uncertain), and thus we need a statistic to evaluate the performance of the wind farm. A common statistic is the expected power or the Annual Energy Production (AEP). Many model simulations are needed to estimate the statistic (Padrón et al., 2016; Murcia et al., 2015). The statistic is usually the objective function of the optimization (Herbert-Acero et al., 2014); thus, the wind farm optimization is an optimization under uncertainty problem (Fig. 1). Optimization under uncertainty (OUU) differs from deterministic optimization in that it contains a nested uncertainty quantification loop to compute statistics. In the OUU problem, for every optimization step many model evaluations are needed to compute the relevant statistics. Thus, even with a very small number of design variables per turbine, the total number of variables and simulations required by the wind farm optimization can grow very rapidly (Gebraad et al., 2017) and quickly make the problem infeasible, especially when using a high-fidelity model for the wind farm simulation.
We see three approaches to improving wind farm optimization capabilities. Each approach focuses on the different blocks of the optimization under uncertainty problem (Fig. 1c). The first approach is to improve the modeling quality of entire wind farms, i.e., improve the models in all the disciplines (aerodynamics, structures, controls, electrical, acoustics, atmospheric physics, policy, economics, etc.) that are relevant to building and operating a wind farm, as well as the interaction between the different turbines. The second approach is to improve the optimization problem formulation and the algorithms to solve the optimization. And the third approach is to improve the treatment of the stochastic nature of the problem, i.e., develop better uncertainty quantification methods to efficiently compute the relevant wind farm statistics (and their gradients with respect to the design variables of the problem) used in the optimization under uncertainty problem.

The first approach increases the fidelity of the model whereas the second (optimization) and third (uncertainty quantification) approaches seek to reduce the number of model evaluations, as this enables the study of larger and more realistic problems.

Here, we focus on the uncertainty quantification approach (the third approach), as it has not been considered in detail before. The most recent and thorough review of the wind farm optimization literature (Herbert-Acero et al., 2014) does not mention it. It only mentions the first two approaches. In the existing work in the literature, the third approach typically focuses on simple integration methods to compute the statistics, which quantify the effect of the stochastic inputs (Kusiak and Song, 2010; Kwong et al., 2012; Chowdhury et al., 2013; Fleming et al., 2016; Gebraad et al., 2017). Simple integration methods, such as the rectangle rule, are inefficient in the number of samples (simulations of the model) needed to accurately estimate a statistic, such as the AEP. They are especially inefficient if multiple stochastic inputs are considered simultaneously. Normally only the wind direction and/or the wind speed are considered as stochastic input variables. Recent work (Padrón et al., 2016; Murcia et al., 2015) is starting to move beyond these simple integration techniques to compute the AEP and instead using the uncertainty quantification method of polynomial chaos to compute the AEP.

In this paper, which is meant as a comprehensive introduction to uncertainty quantification methods applied to wind-farm simulations, we describe in detail the polynomial chaos (PC) method and show that, for the efficient (small number of model simulations) computation of the AEP, the PC method based on regression should be used. An additional benefit of the PC method is that it makes it feasible to consider multiple uncertain variables (e.g., wind direction, wind speed, wind turbulence, wake model parameter) that impact the computation of the AEP (Padrón, 2017).

In addition, we show how to compute gradients of the PC method can be used to efficiently compute the gradient of statistics, such as the AEP, with a modified version of the PC method. Eldred (2011) describes how to compute the gradients for PC based on quadrature. Here we show how to compute the gradients for PC based on regression (Sect. 4.3). The use of gradients allows us to efficiently tackle much larger optimization problems (Gebraad et al., 2017). To compute the gradients of the wake model, we use the recently developed Floris wake model with the modifications by Thomas et al. (2017) that provide analytic and continuous gradients of the wake model.

We first discuss the details of computing the power and the AEP of a wind farm in Sect. 2. Then, we discuss uncertainty quantification in Sect. 3 and the polynomial chaos method in Sect. 4. Finally, we discuss the details of the problem formulation in Sect. 5 and the results in Sect. 6.
Figure 2. Schematic of the Floris wake model. The model has three zones with varying diameters, $D_{w,q}$, that depend on tuning parameters $k_e$ and $m_{e,q}$. The effective hub velocity is computed using the overlap ratio, $A_{OL}^q$, of the part of the rotor-swept area overlapping each wake zone respectively to the total rotor-swept area.

2 Computing the power and the annual energy production of a wind farm

We first describe the aerodynamic wake model we use (Sect. 2.1). The wake model gives an estimate of the hub-height velocity at each wind turbine, from which we can compute the power produced by the wind farm (Sect. 2.2). Then, to obtain the Annual Energy Production (AEP) we need to integrate the power over all wind conditions that occur in a year (Sect. 2.3) and weigh the results proportionally to the frequency with which such wind conditions manifest themselves.

2.1 Floris

The Floris (FLow Redirection and Induction in Steady-state)\(^1\) (Gebraad et al., 2016) wake model is an enhancement of the Jensen wake model (Jensen, 1983) and the wake deflection model presented in (Jiménez et al., 2009). The Floris model builds on the Jensen model by defining three separate wake zones with differing expansion and decay rates (controlled by tunable coefficients) to more accurately describe the velocity deficit across the wake region. A simple overlap ratio of the area of the rotor in each zone of each shadowing wake to the full rotor area is used to determine the effective hub velocity of a given turbine. A simple overview of the Floris model, showing the zones and overlap areas, is shown in Fig. 2. We use the Floris wake model with changes to remove discontinuities and add curvature to regions of non-physical zero gradient to make the model more suitable for gradient-based optimization (Thomas et al., 2017). In this work, we use the parameter values recommended in (Gebraad et al., 2016) and (Thomas et al., 2017) and set the yaw-offset angle of each turbine to zero.

\(^1\)We use the name Floris for the model, instead of FLORIS, the name used in (Gebraad et al., 2016).
Table 1. The variables used for calculating the power.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncertain</td>
<td>$\xi$ - wind direction.</td>
</tr>
<tr>
<td></td>
<td>freestream wind speed.</td>
</tr>
<tr>
<td>Design</td>
<td>$x$ - The $x$ location of each turbine.</td>
</tr>
<tr>
<td></td>
<td>$y$ - The $y$ location of each turbine.</td>
</tr>
<tr>
<td>Parameters</td>
<td>$\theta$ - yaw angles, turbine characteristics,</td>
</tr>
<tr>
<td></td>
<td>and wake model parameters.</td>
</tr>
</tbody>
</table>

2.2 Computing the power of a wind farm

We will consider the power of the wind farm to be a function of three classes of variables: uncertain variables $\xi$, design variables $x$, and parameters $\theta$,

$$P = P(\xi, x, \theta).$$

Uncertain variables are variables that follow a probability distribution, design variables are variables that an optimizer can vary, and parameters are important constants that govern the behavior of the system. The classification of the variables is problem dependent. For instance, the rotor yaw could be considered a design variable, a parameter, or an uncertain variable to account for yaw-offset measurement error. A tunable parameter of a wake model, such as the wake expansion coefficient, could be considered as a parameter or as an uncertain variable given by a particular distribution.

For the problems considered in this work, Table 1 lists in which category we place each variable that influences the power computation. The uncertain variables are the wind direction and wind speed with probability distributions described in Sect. 5.1.

The power of the wind farm for a given wind direction and wind speed is equal to the sum of the power produced by each turbine $P = \sum_{i=1}^{n_{turb}} P_i$. The power of each turbine is calculated from

$$P_i = \frac{1}{2} \rho C_P A U_i^3,$$

where $\rho$ is the air density, $A$ is the rotor swept area, $C_P$ is the power coefficient, and $U_i$ is the effective hub velocity for each turbine, which is calculated by the wake model and is a function of the three types of variables described above

$$U_i = f(\xi, x, \theta).$$

The power coefficient captures both the aerodynamic and electromechanical properties of the wind turbine. It is a complex function of many variables (Herbert-Acero et al., 2014) and is usually reported by wind turbine manufacturers as a function of the tip-speed ratio, which depends on wind speed at hub height $U_i$. A simple expression for $C_P$ can also be computed using the classical actuator disk theory (Sanderse, 2009).
2.3 Computing the Annual Energy Production (AEP) of a wind farm

The Annual Energy Production (AEP) is an important metric used to describe a wind farm. The AEP is a statistic. Specifically, it is a mean, as it is a function of the expected power multiplied by the number of hours in a year,

\[
AEP = 8760 \frac{hr}{yr} E[P(\xi)].
\]  

(4)

The expected power, \(E[P(\xi)]\), or the mean of the power, \(\mu_P\), is defined as

\[
\mu_P = E[P(\xi)] = \int_{\Omega} P(\xi) \rho(\xi) d\xi,
\]  

(5)

where \(\xi = (\xi_1, \xi_2, \ldots, \xi_n)\) is a vector of random variables, which we refer to as the uncertain variables, \(\rho(\xi)\) is the joint probability density function of the uncertain variables, \(\Omega\) is the domain of the uncertain variables, and \(P\) is the power produced by the wind farm (Eq. (1)). Common uncertain variables are the wind direction and the freestream wind speed.

The expected power, and hence the AEP, is normally computed as a weighted average, which amounts to the rectangle rule of integration (Sect. 3.1.1). Other uncertainty quantification methods (Sect. 3) can be used to compute the expected value (AEP). Specifically, we can compute the AEP efficiently by polynomial chaos (Sect. 4).

3 Uncertainty quantification

Uncertainty quantification (UQ) is the process of (1) characterizing input uncertainties, and then (2) propagating these input uncertainties through a computational model with the goal of quantifying their effect on the model’s output. There are many sources of uncertainty in the modeling of a problem, and different classifications of the uncertainties have been proposed (Kennedy and O’Hagan, 2001; Oberkampf et al., 2001; Beyer and Sendhoff, 2007). A common classification is to divide the uncertainty into aleatory and epistemic uncertainties (Oberkampf et al., 2001).

In this work, we consider aleatory uncertainties that arise from the variability in the inputs to our model caused by changing environmental conditions. We describe this input variability as random variables with associated probability distributions. Thus, the first step of characterizing the input uncertainties is concerned with finding the probability distributions that describe the model’s inputs. This process is known as statistical inference, model calibration and inverse uncertainty quantification (Smith, 2014). Here, we assume that this step has been completed, i.e., we have distributions that characterize the uncertain inputs (Sect. 5.1). We focus on the second step of propagating the input uncertainties to find the statistics that describe the output.

3.1 Uncertainty propagation methods

The goal of uncertainty propagation methods is to compute the statistics that describe the effect of the uncertain inputs on the model output. There are several methods to propagate the uncertainties and compute statistics (Le Maître and Knio, 2010; Smith, 2014), where each method has its advantages/disadvantages depending on the type and size of the problem. The most common methods are sampling or Monte Carlo methods (Caflisch, 1998). Other methods include direct integration methods
and stochastic expansion methods. Direct integration methods are the currently used method to compute the AEP of a wind farm; we briefly describe them below (Sect. 3.1.1). We describe the stochastic expansion method of polynomial chaos in detail in Sect. 4, and compare the different methods to propagate the uncertainties to compute the AEP in Sect. 6.2.

### 3.1.1 Direct numerical integration (Rectangle rule)

As the name implies, this method numerically evaluates the integrals in the definition of the statistics. The integrals to evaluate for the mean (or expected value) and the variance are

\[
\begin{align*}
\mu_R &= \mathbb{E}[R] = \int_\Omega R(\xi) \rho(\xi) \, d\xi, \\
\sigma_R^2 &= \text{Var}[R] = \mathbb{E}[(R(\xi) - \mathbb{E}[R(\xi)])^2] \\
&= \mathbb{E}[R(\xi)^2] - (\mathbb{E}[R(\xi)])^2 \\
&= \int_\Omega R(\xi)^2 \rho(\xi) \, d\xi - \mu_R^2,
\end{align*}
\]

where \(R(\xi)\) is the model output and \(\xi\) the uncertain input variables. The random vector \(\xi = (\xi_1, \xi_2, \ldots, \xi_n)\) with joint probability distribution \(\rho(\xi)\) describes the input variability over the domain \(\Omega\). Each random input can follow a particular distribution \(\rho_i(\xi_i)\). For the case of independent random variables, the joint distribution is the product of each univariate distribution \(\rho(\xi) = \prod_{i=1}^n \rho_i(\xi_i)\). In this work, we will assume that the random input variables are independent. For a description of methods for dealing with cases when the variables are dependent see Padrón (2017).

There are many quadrature methods to evaluate integrals (Ascher and Greif, 2011). We describe the rectangle rule, as this is what is currently used in the wind farm community to compute the AEP.

**Rectangle rule.** The rectangle rule, or mid-point rule, is the simplest and most straightforward quadrature method. To approximate the mean or expected value,

\[
\mu_R = \mathbb{E}[R] = \int_\Omega R(\xi) \rho(\xi) \, d\xi 
\]

with the rectangle rule, we divide the domain of the uncertain variable \(\Omega = [a, b]\) into \(m\) equal subintervals of length \(\Delta \xi = (b - a)/m\). Next, we construct rectangles with base \(B = \Delta \xi\) and height equal to the product of the response of the model and the density evaluated at the mid-point of the subinterval \(H = R(\xi_j) \rho(\xi_j)\). Then, the rectangle rule approximates the expected value by adding up the areas of the \(m\) rectangles

\[
E[R] = \int_a^b R(\xi) \rho(\xi) \, d\xi \approx \sum_{j=1}^m R(\xi_j) \rho(\xi_j) \Delta \xi. 
\]

---

\(^2\)For simplicity, we consider describe the 1-dimensional case, \(\xi = \xi\).
A simple improvement is to integrate the density exactly within each subinterval

\[
E[R] = \int_a^b R(\xi) \rho(\xi) \, d\xi \approx \sum_{j=1}^{m} R(\xi_j) \int_{\xi_{j-1/2}}^{\xi_{j+1/2}} \rho(\xi) \, d\xi.
\]  

This is easily done as the density is known. This modification is helpful for a small number of evaluations \(m\). We will use this modified rectangle rule and simply refer to it as the rectangle rule.

5 Polynomial chaos

Polynomial chaos (PC) is the name of an uncertainty quantification (UQ) method that approximates a function with a polynomial expansion made up of orthogonal polynomials. This function has \textit{random variables} as inputs, and we are interested in the effects of the random (uncertain) inputs on the output of this function. Statistics of the output can describe the effects of the inputs. We use the polynomial chaos method to efficiently compute these statistics and the gradients of these statistics.

We first describe the polynomial chaos method in Sect. 4.1. We then discuss two methods—quadrature and regression—to compute the coefficients used in the PC expansion (Sect. 4.2). We finish with \textit{an extension of polynomial chaos to compute gradients—a description of how to compute the gradients of statistics from the polynomial chaos expansion} (Sect. 4.3).

4.1 Polynomial chaos expansion

Let \(R(\xi)\) be a function of interest that depends on the uncertain variable \(\xi\). We can approximate the function by a polynomial expansion

\[
R(\xi) \approx \hat{R}(\xi) = \sum_{i=0}^{p} \alpha_i \phi_i(\xi).
\]  

The approximate response \(\hat{R}(\xi)\) is a polynomial of order \(p\). Usually, the larger the polynomial order the closer the approximation is to the true response \(R(\xi)\).

The polynomial basis \(\{\phi_i(\xi)\}_{i=0}^{p}\) is determined by the distribution of the uncertain variable—the polynomial basis is orthogonal with a weight function that corresponds up to a constant to the probability density function of the uncertain variable. Common random (uncertain) variables (Normal, Uniform, Exponential, Beta) have corresponding classical orthogonal polynomials (Hermite, Legendre, Laguerre, Jacobi) (Eldred et al., 2008). \textit{It is necessary to numerically generate orthogonal polynomials for uncertain variables with empirically-determined Empirically-determined distributions, such as those obtained from wind conditions, to do not have corresponding classical orthogonal polynomials. For the distributions obtained from the wind conditions, we need to numerically generate custom orthogonal polynomials in order to preserve the optimal convergence property of the polynomial chaos expansion (Oladyshkin and Nowak, 2012). Also, the use of orthogonal polynomials allows us to analytically compute statistics from the polynomial chaos expansion (Sect. 4.1.1).} Details about the numerical generation of orthogonal polynomials can be found in Gautschi (2004) and an example of the generation of orthogonal polynomials for wind
distributions in Padrón (2017). In addition to the **orthogonal polynomials**, optimal convergence properties, the use of orthogonal polynomials allows us to analytically compute statistics from the polynomial chaos expansion (Sect. 4.1.1).

In addition to the orthogonal polynomials, the other component of the expansion Eq. (12) are the coefficients \( \alpha_i \). The coefficients can be computed either by quadrature or regression as described in Sect. 4.2.

For the case of multiple uncertain variables \( \xi = (\xi_1, \xi_2, \ldots, \xi_n) \) and using a multi-index \( i = (i_1, i_2, \ldots, i_n) \), we write the multi-dimensional polynomial approximation as

\[
R(\xi) \approx \hat{R}(\xi) = \sum_{i \in I_p} \alpha_i \Phi_i(\xi) \tag{13}
\]

The multi-dimensional basis functions \( \Phi_i(\xi) \) are given by products of the 1-dimensional orthogonal polynomials

\[
\Phi_i(\xi) = \prod_{j=1}^{n} \phi_{i_j}(\xi_j) \tag{14}
\]

When the uncertain variables are independent, the multi-dimensional basis functions are also orthogonal. The values of the elements \( i_j \) of the multi-index depend on how the expansion is truncated, i.e., on how the index set \( I_p \) is defined. There are two common ways in which to define the index set: **total-order expansion** and **tensor-product expansion**.

In **total-order expansion** a total polynomial order bound \( p \) is enforced:

\[
I_p = \{ i : |i| \leq p \}, \quad |i| = i_1 + i_2 + \cdots + i_n. \tag{15}
\]

And in **tensor-product expansion** a per-dimension polynomial order bound \( p_j \) is enforced

\[
I_p = \{ i : i_j \leq p_j, \ j = 1, \ldots, n \}. \tag{16}
\]

An example showing the multi-dimensional basis polynomials Eq. (14) for both the **total-order expansion** and **tensor-product expansion** can be found in Padrón (2017). The **tensor-product expansion** is the preferred approach when the coefficients are computed with quadrature (Sect. 4.2.1) because of increased monomial coverage and accuracy (Eldred and Burkardt, 2009).

The **total-order expansion** is the preferred approach when the coefficients are computed with regression (Sect. (4.2.2)) because it keeps the sampling requirements lower (Eldred and Burkardt, 2009).

### 4.1.1 Mean and variance from the polynomial chaos expansion

The mean and variance of the function of interest \( R(\xi) \) are a function of the coefficients \( \alpha_i \) of the polynomial chaos expansion. The statistics are obtained by substituting the polynomial chaos expansion Eq. (13) into the definitions of the mean Eq. (6) and variance Eq. (9), and by integrating the expansion and simplifying using the orthogonality of the polynomials.

The mean is the zeroth coefficient

\[
\mu_R = \alpha_0 \tag{17}
\]
and the variance is the sum of the product of the square of the coefficients—excluding the zeroth coefficient—with the inner product $\langle \Phi_i^2(\xi) \rangle$,

$$\sigma_R^2 = \sum_{i \in \mathcal{I}_p \setminus \{0\}} \alpha_i^2 \langle \Phi_i^2(\xi) \rangle,$$

(18)

where the inner product is defined as $\langle \Phi_i^2(\xi) \rangle = \int_{\Omega} \Phi_i(\xi)\Phi_i(\xi)\rho(\xi)\,d\xi$ and $0$ is the first multi-index—the one with all zero elements.

4.2 Calculating polynomial chaos coefficients

The coefficients of the polynomial chaos expansion Eq. (13) can be calculated via quadrature or by linear regression.

4.2.1 Quadrature

To obtain the coefficients of the polynomial chaos expansion

$$R(\xi) = \sum_{i \in \mathcal{I}_p} \alpha_i \Phi_i(\xi),$$

(19)

via quadrature, we take the inner product of both sides of Eq. (19) with respect to $\Phi_j(\xi)$ to yield

$$\langle R, \Phi_j \rangle = \sum_{i \in \mathcal{I}_p} \alpha_i \langle \Phi_i, \Phi_j \rangle.$$  \hspace{1cm} (20)

Making use of the orthogonality of the polynomials and solving for the coefficients in Eq. (20), we obtain

$$\alpha_i = \frac{\langle R(\xi), \Phi_i(\xi) \rangle}{\langle \Phi_i^2(\xi) \rangle} = \frac{1}{\langle \Phi_i^2(\xi) \rangle} \int_{\Omega} R(\xi)\Phi_i(\xi)\rho(\xi)\,d\xi,$$

(21)

where the domain $\Omega$ is the Cartesian product of $1D$ domains $\Omega_j$ for each dimension, $\Omega = \Omega_1 \times \cdots \times \Omega_n$, and $\rho(\xi) = \prod_{j=1}^n \rho_j(\xi_j)$ is the joint probability density of the stochastic parameters. The inner product $\langle \Phi_i^2(\xi) \rangle$ is known analytically or inexpensively computed. Thus, most of the computational expense in solving for the coefficients resides in evaluating the model $R(\xi)$ in the multi-dimensional integral $\int_{\Omega} R(\xi)\Phi_i(\xi)\rho(\xi)\,d\xi$. This integral is solved with quadrature (numerical integration). Note that the zero coefficient in Eq. (21) reduces to the definition of the mean

$$\mu_R = \alpha_0 = \int_{\Omega} R(\xi)\rho(\xi)\,d\xi,$$

(22)

which the direct numerical integration methods attempt to compute directly (Sect. 3.1.1).

4.2.2 Regression

To obtain the coefficients of the polynomial chaos expansion Eq. (13) via regression, we construct a linear system

$$\Phi \alpha = R$$

(23)
and solve for the coefficients $\alpha$ that best represent a set of responses $R$. The set of responses is generated by evaluating the model at $m$ realizations of the uncertain vector $\xi$. The $m$ uncertain vectors are most commonly obtained by sampling the density of the uncertain variables (Hosder et al., 2007).

Each row of the matrix $\Phi$ contains the orthogonal polynomials $\Phi_j$ evaluated at a sample $\xi_i$

\[
\begin{bmatrix}
\Phi_0(\xi_1) & \cdots & \Phi_{n-1}(\xi_1) \\
\vdots & \ddots & \vdots \\
\Phi_0(\xi_m) & \cdots & \Phi_{n-1}(\xi_m)
\end{bmatrix}
\begin{bmatrix}
\alpha_0 \\
\vdots \\
\alpha_{n-1}
\end{bmatrix}
= 
\begin{bmatrix}
R_1 \\
\vdots \\
R_m
\end{bmatrix}
\] (24)

The size of the $m \times n$ matrix is determined by the number of samples $m$ and by how the polynomial chaos expansion is truncated (Sect. 4.1) which results in $n$ terms. It is common to specify a total order expansion along with a collocation ratio $cr = m/n$ to determine the number of samples $m$. The collocation ratio determines if the system is overdetermined $cr > 1$ or underdetermined $cr < 1$.

For overdetermined systems, the most popular method (and the one we use) to estimate the coefficients is least squares, in which we pick coefficients $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ that minimize the residual sum of squares

\[
\alpha = \arg \min ||\Phi \alpha - R||_2^2.
\] (25)

For underdetermined systems, solving a regularized least squares problem is preferred (Doostan and Owhadi, 2011).

For a given number of samples $m$ in the linear system Eq. (24) we can use cross-validation (Hastie et al., 2009) to pick the best polynomial order $n$ to approximate the response.

4.3 Gradients of statistics with polynomial chaos

Let $R(\xi, x)$ be a function of interest that depends on uncertain variables $\xi$ and also on design variables $x$. We assume independence between the design and uncertain variables\(^3\). Now the polynomial chaos expansion—over the uncertain variables—becomes

\[
R(\xi, x) \approx \hat{R}(\xi, x) = \sum_{i \in I_p} \alpha_i(x) \Phi_i(\xi).
\] (26)

This expansion is only valid for a particular design vector—the coefficients $\alpha_i(x)$ are a function of the design variables. Therefore, the statistics are also a function of the design variables. Specifically the mean and the variance are

\[
\mu_R(x) = \alpha_0(x),
\]

\[
\sigma_R^2(x) = \sum_{i \in I_p \setminus 0} \alpha_i^2(x) \langle \Phi_i^2(\xi) \rangle.
\] (28)

\(^3\)For most applications, the design and uncertain variables are independent. For instance, the design variables are the wind turbines location and the uncertain variables are the wind conditions.
4.3.1 Gradients of the statistics with polynomial chaos

We want to know the gradients of the statistics with respect to the design variables, and we proceed to derive them below. For simplicity, we drop the subscript from the statistics $\mu_R = \mu$, the explicit variable dependence $R(\xi, x) = R$, the bolded notation, and we use the following notation for the gradient $\frac{df}{dx} \equiv \nabla f$.

The gradient of the mean from Eq. (27) is

$$\frac{d\mu}{dx} = \frac{d\alpha_0}{dx},$$  \hspace{1cm} (29)

and the gradient of the variance from Eq. (28) is

$$\frac{d\sigma^2}{dx} = \sum_{i \in I_p \setminus \emptyset} \langle \Phi_i^2 \rangle \frac{d\alpha_i^2}{dx} = 2 \sum_{i \in I_p \setminus \emptyset} \langle \Phi_i^2 \rangle \alpha_i \frac{d\alpha_i}{dx}. \hspace{1cm} (30)$$

Both, the mean and the variance gradients, depend on the gradient of the coefficients $\frac{d\alpha_i}{dx}$.

4.3.2 Gradients of the coefficients

The gradient of the coefficients can be computed with quadrature or regression, similarly to how the coefficients can be calculated with quadrature (Sect. 4.2.1) or regression (Sect. 4.2.2).

**Quadrature.** We start from the equation for the coefficients Eq. (21) and take the gradient to obtain

$$\frac{d\alpha_i}{dx} = \frac{1}{\langle \Phi_i^2 \rangle} \int_\Omega \frac{dR}{dx} \Phi_i \rho d\xi = \frac{\langle \frac{dR}{dx}, \Phi_i \rangle}{\langle \Phi_i^2 \rangle}. \hspace{1cm} (31)$$

Replacing this equation into the gradient of the mean Eq. (29) we obtain

$$\frac{d\mu}{dx} = \langle \frac{dR}{dx} \rangle. \hspace{1cm} (32)$$

And replacing Eq. (31) into the gradient of the variance Eq. (30) we obtain

$$\frac{d\sigma^2}{dx} = 2 \sum_{i \in I_p \setminus \emptyset} \alpha_i \langle \frac{dR}{dx}, \Phi_i \rangle. \hspace{1cm} (33)$$

To obtain the gradients of the statistics with respect to each design variable we need to evaluate the multi-dimensional integral containing $\frac{dR}{dx}$. The integral is evaluated with quadrature (Sect. 4.2.1) and requires the computation of the gradient of the response at each of the quadrature points. Ideally, one would use adjoint methods Giles and Pierce (2000) or algorithmic differentiation Griewank and Walther (2008) to compute the gradients, $\frac{dR}{dx}$, efficiently.
Regression. We start from the linear system Eq. (23) and take the gradient to obtain
\[
\frac{d\Phi}{dx} = \frac{dR}{dx} \tag{34}
\]
\[
\Phi \frac{d\alpha}{dx} = \frac{dR}{dx} \tag{35}
\]
\[
\begin{bmatrix}
\Phi_0(\xi_1) & \cdots & \Phi_{n-1}(\xi_1) \\
\vdots & \ddots & \vdots \\
\Phi_0(\xi_m) & \cdots & \Phi_{n-1}(\xi_m)
\end{bmatrix}
\begin{bmatrix}
\frac{d\alpha_0}{dx_1} & \cdots & \frac{d\alpha_0}{dx_d} \\
\vdots & \ddots & \vdots \\
\frac{d\alpha_{n-1}}{dx_1} & \cdots & \frac{d\alpha_{n-1}}{dx_d}
\end{bmatrix}
= 
\begin{bmatrix}
\frac{dR_1}{dx_1} & \cdots & \frac{dR_1}{dx_d} \\
\vdots & \ddots & \vdots \\
\frac{dR_m}{dx_1} & \cdots & \frac{dR_m}{dx_d}
\end{bmatrix}
\tag{36}
\]

To solve for the gradient of the coefficients, we solve the linear system one column at a time of the \( \frac{d\alpha}{dx} \) matrix with the corresponding column of the matrix of the gradients \( \frac{dR}{dx} \). The linear system for the multiple right hand sides can be solved with the methods described in Sect. 4.2.2.

Again, the gradient of the mean Eq. (29) and the gradient of the variance Eq. (30) are a function of the gradient of the coefficients. Thus, to obtain the gradient of the mean take the first row of the \( \frac{d\alpha}{dx} \) matrix; and for the gradient of the variance use the gradient of the coefficients from all the other rows.

4.3.3 Gradients of the statistics by direct numerical integration

Similarly to computing the mean and variance with direct numerical integration (Sect. 3.1.1), we can also compute the gradients of the mean and variance directly with numerical integration by differentiating the definitions of the mean, Eq. (6), and the variance, Eq. (9).

5 Problem details

Here we describe the details of computing AEP in the wind farm. We first describe and discuss the inputs, which are the wind direction and wind speed (uncertain variables) (Sect. 5.1) and the wind turbine positions (design variables) (Sect. 5.2). Then, we describe details about the output, the average AEP error (Sect. 5.3), that we use to compare the different methods we consider to compute the AEP (Sect. 5.4).

5.1 Probability distributions of the uncertain wind conditions

We consider the wind direction and the wind speed as uncertain variables with probability distributions shown in Fig. 3. The distributions show the likelihood of a particular wind direction or wind speed occurring during a year. The wind direction distribution (Fig. 3a) is constructed by linearly interpolating For simplicity, we assume that the wind direction and wind speed distributions are independent (Sect. 4.1). We construct the independent distributions starting from the wind measurements taken near the Princess Amalia wind farm (Sect. 5.2) by the NoordZeeWind meteorological mast during a year (Brand et al., 2012). The mast is situated near the Princess Amalia wind farm (Sect. 5.2), and the measurements were taken from July. For a wind rose of the measurements, see Gebraad et al. (2017). The wind rose has a wind direction bin width of 5° and a wind speed bin
width of 1 , from June 30, 2005, to June 30, 2006, before the construction of the nearby Noordzee offshore wind farm m/s throughout the operational range of the turbine.

We construct the wind direction distribution (Fig. 3a) by linearly interpolating the wind measurements of the likelihood of the wind coming from a particular binned wind direction for all wind speeds. The zero degree direction is set at North and increases clockwise.

The wind speed is usually correlated with the wind direction, but for simplicity, we assume that they are independent (Sect. 3.1.1). With this assumption, we fit a single Weibull distribution to the meteorological mast data. For the wind speed, instead of linearly interpolating the data, we fit a Weibull distribution and then truncate it (Fig. 3b). The Weibull distribution is the preferred distribution to model the wind speed distribution (Belu and Koracin, 2013; Herbert-Acero et al., 2014). For the Princess Amalia data, the resulting wind speed distribution is the Weibull,

\[ \rho(\xi; \alpha, \beta) = \frac{\alpha}{\beta} \left( \frac{\xi}{\beta} \right)^{\alpha-1} e^{-\left(\frac{\xi}{\beta}\right)^{\alpha}}, \quad \xi \geq 0, \]

with shape parameter \( \alpha = 1.8 \) and scale parameter \( \beta = 12.552983 \). The operational range of the turbine is from the cut-in speed of 3 m/s to the cut-out speed of 25 m/s. We will consider truncate instead at the upper limit to be of 20 m/s as, above this speed, the power produced by the wind farms is almost always constant at its rated power and thus contributes only a constant term to AEP.

The probability distributions are the weight functions of the orthogonal polynomials used in the polynomial chaos method (Sect. 4.1). For both of the distributions (weight functions), we use a histogram of 50 equal width bins to describe each distribution. Using these weight functions, we then numerically generate their corresponding orthogonal polynomials.

5.2 The wind farm layouts

To showcase the results, we will focus on four representative layouts: Grid, Amalia, Optimized and Random (Fig. 4). The layouts have sixty turbines represented by individual dots in each of the figures we show in the results. Each dot is to scale, where the diameter represents the rotor swept area. The Grid layout fits in a box of equivalent area to that of the Amalia layout. The Amalia layout, which is grid-like, is that of the Princess Amalia wind farm located 23 km off the coast of the Netherlands. The Optimized layout is a representative optimal layout obtained by running the optimization problem (Sect. 6.3.1). When we refer to this particular optimized layout, we will capitalize the word optimized. The Random layout was generated by random sampling and keeping the turbines that are contained within the convex hull of the Amalia wind farm without enforcing any spacing constraints between the turbines. We will refer to the Grid and Amalia as grid-like layouts and the Optimized and Random as non-grid-like layouts.

\footnote{The fitted Weibull \( \rho(\xi; \alpha, \beta) = \frac{\alpha}{\beta} \left( \frac{\xi}{\beta} \right)^{\alpha-1} e^{-\left(\frac{\xi}{\beta}\right)^{\alpha}}, \xi \geq 0 \) has shape parameter \( \alpha = 1.8 \) and scale parameter \( \beta = 12.55 \). The Weibull distribution is the preferred distribution to model the wind speed distribution (Belu and Koracin, 2013; Herbert-Acero et al., 2014).}

\footnote{The truncated distribution is no longer a Weibull distribution and needs to be scaled to ensure it is a valid probability density function (it integrates to 1).}
Figure 3. The uncertain variables probability distributions. We assume the distributions are constructed from independent and construct them based on wind measurements taken by the NoordZeeWind meteorological mast during a year. The wind direction distribution is a linear interpolation of the data, and the wind speed distribution is a truncated Weibull fit to the meteorological wind speed data averaged over all wind directions. The vertical lines show the cut-in and cut-out speed of a single wind turbine.

Figure 4. Representative wind farm layouts used in the results. Each dot represents a wind turbine to scale—the diameter of the dot represents the swept area of the rotor.

In reality, the 60 turbines in the Princess Amalia wind farm are the Vestas V80 model. For each of the layouts in our study, we use the NREL 5-MW reference turbine (Jonkman et al., 2009), as this turbine is similar to the Vestas V80, and has an open source design.

5.3 Convergence metric — the average AEP error

We use an ensemble of 10 AEP results to compute the average AEP error. The average AEP error allows us to better illustrate the differences between the different methods used to compute the AEP and to avoid drawing conclusions from one-off solutions. We found that averaging over 10 AEP results is enough to illustrate the difference between methods and to smooth out the co-
vergence of the AEP error (Fig. 8). The ensemble of results account for the fact that the zero (starting) position for the wind direction is arbitrary (it could be North, South, East, West, etc.). Also, averaging the AEP errors helps smooth out the AEP convergence curves. We generate the ensemble of AEP results by selecting 10 different input sets. For example, for the rectangle rule, if we consider 36 wind directions the 10 sets are \{[0, 10, 20, \ldots, 340, 350 degrees]; [1, 11, 21, \ldots, 341, 351]; \ldots; [9, 19, 29, 349, 359]\}. For the polynomial chaos based on quadrature, the quadrature points are the numerically generated Gaussian quadrature points for the interval. Thus, to create 10 different sets, we pick 10 different intervals, i.e., we pick different starting positions. When considering 36 wind directions the chosen intervals are \{[0, 360]; [10, 370]; [20, 380]; \ldots; [340, 700]; [350, 710]\}. For both the rectangle and polynomial chaos based on quadrature, the wind speed points for each set are the same. For the polynomial chaos based on regression and for Monte Carlo, the wind directions and wind speed pairs are generated by sampling the distribution. Thus, to obtain 10 different sets, 10 different samplings are performed. We use the average AEP error as the convergence metric

\[
\text{average AEP error} = \frac{1}{10} \sum_{i=1}^{10} \left| \frac{AEP_i - \text{baseline AEP}}{\text{baseline AEP}} \right| \times 100\%.
\] (37)

5.3.1 Baseline AEP

We take as the baseline or true AEP the AEP computed with 200,000 Monte Carlo samples. We picked 200,000 MC samples to ensure the 99 % confidence interval for the true AEP was smaller than 1 % of the computed AEP value for all layouts. We consider an AEP within 1 % of the baseline AEP to be accurate, and we will use it as a reference for the results. AEP predictions of real wind farms usually have an error of 10–20 % (Barthelmie et al., 2007; Briggs, 2013) due to uncertainty in wind conditions and to the errors of wake models, thus resolving the AEP to less than 1 % is unnecessary.

5.4 Methods to compute the AEP

Here, we provide the details of the methods used to compute the AEP, as well as, the abbreviations for the methods:

- rect rectangle rule (Sect. 3.1.1)
- PC-Q polynomial chaos based on quadrature (Sect. 4.2.1)
- PC-R polynomial chaos based on regression (Sect. 4.2.2)
- MC Monte Carlo (Caflisch, 1998)

For the quadrature-based methods (rect, PC-Q), we use tensor product quadrature to compute multi-dimensional integrals. We use the same number of points for each dimension because we did not see any benefit in favoring a particular dimension. For PC-Q, we use Gaussian quadrature. For Monte Carlo, we use the traditional Monte Carlo method, i.e., random samples. For PC-R, we use Latin-Hypercube sampling (McKay et al., 1979) to generate the samples needed to construct the linear system. We solve the linear system with least-squares. For a given number of samples, given by the total polynomial order \( p \), we use 10-fold cross-validation to find the least-squares best fit from polynomials of total order 1 up to total order \( p \). The sampling methods and the polynomial chaos methods we use are implemented in the open-source DAKOTA toolkit (Adams et al., 2017).
Figure 5. Power contours as a function of wind direction and wind speed for the Grid, Amalia, Optimized and Random layouts. The grid-like layouts (top row) have larger peaks due to wind turbines being aligned along particular directions and thus experiencing full-wake conditions. For all layouts as the speed increases the power increases until the wind farm reaches its rated power 300 MW.

6 Results

The Annual Energy Production (AEP) of a wind farm is obtained by integrating the power output of the farm over the different wind conditions. Thus, we first characterize the power output of the wake model for different input conditions (Sect. 6.1). Next, we focus on the convergence of the AEP (Sect. 6.2), and then on the wind farm layout optimization problem to maximize the AEP (Sect. 6.3).

6.1 Power response as a function of the uncertain variables

The power production, computed with the Floris wake model, for the four wind farm layouts (Sect. 5.2) as a function of both the wind direction and the wind speed is shown in Fig. 5. The wind direction is measured from North and increases clockwise. The power response as a function of the wind direction is highly oscillatory, and for layouts with structure\(^6\), it is periodic. The

\(^6\)Wind turbines are aligned in particular directions. The layouts are grid-like and have symmetry planes.
peaks in the contour lines identify wind directions for which there is a poor performance (low power). The grid-like layouts (top row of Fig. 5) have larger peaks due to wind turbines being aligned along particular directions and thus experiencing full-wake conditions. The worst wind direction for the Grid layout is directly from North (0°) or South (180°) when rows of ten turbines are aligned. The power for the Optimized layout is worst at around 125°. This shows that the optimizer took into account that the likelihood of the wind coming from the 125° direction is minimal (Fig. 3a). For all layouts, as the speed increases

The power response as a function of the wind direction has similar shapes for different wind speeds, until the speed is high enough (larger than the rated speed of the layout) that the response gets clipped at 300MW (all sixty turbines are operating at their rated power 5MW). As a function of the wind speed, for each wind direction, the power response starts after the cut-in speed of 3 m/s and is cubic (concave up) until individual turbines start reaching rated power. After that, the power increases until the wind more slowly (concave down) until the farm reaches its rated power of 300 MW. The less grid-like layouts show a smoother transition to the maximum power because each turbine reaches its rated speed at different freestream speeds.

The power response as a function of the uncertain variables (wind direction and wind speed) is a complicated function, which needs to be integrated, weighted by the likelihood of the wind conditions, to obtain the AEP of the wind farm. Next, we show that using polynomial chaos, instead of the rectangle rule, results in better estimates of the AEP.

6.2 AEP convergence: polynomial chaos vs. rectangle rule

We consider the AEP as a function of two uncertain variables: the wind speed and the wind direction. The AEP is usually considered a function of these two variables. We compare the convergence of the AEP for the different methods to compute the AEP: polynomial chaos (based on quadrature and regression), the rectangle rule and Monte Carlo (Fig. 6). The polynomial chaos based on regression (PC-R) performs the best for all layouts. It is followed, by the polynomial chaos based on quadrature (PC-Q) and the rectangle rule, which perform similarly. The worst performer is Monte Carlo. The slow convergence of statistics with Monte Carlo is well known. Monte Carlo will start to outperform the other methods when the AEP is a function of a large number (5–10) of uncertain variables, as it does not suffer from the curse of dimensionality.

The superior performance of the polynomial chaos based on regression, especially for the grid like-layouts (Grid and Amalia), is due to the following: the polynomial chaos fit based on regression does not chase all the high-frequency oscillations in the power response (Fig. 5), i.e., it smooths out the response. The PC-R fit is usually not higher than an eight total-order polynomial (Sect. 4.1). Whereas, the PC-Q order fit is higher, as it is directly proportional to the number of samples per dimension. A downside of the PC-R being able to predict the mean (AEP) accurately, is that it can underpredict the true variance (standard deviation) of the response (Fig. 7). Usually, the standard deviation of the power response (energy) over a year is not considered as a function to optimize. A common objective in wind farm optimization is to maximize the total amount of energy produced over a year independent of the variability in the power production over the year. For a wind farm, the variability of the energy produced over a year is less important than the variability caused by the changing wind conditions during the day.

\[7\] For the two-dimensional problem at 625 samples (25 x 25 grid) the polynomial order in each dimension is 24.
Figure 6. The average AEP error as a function of the number of samples for polynomial chaos (based on regression and quadrature), the rectangle rule and Monte Carlo. The AEP is a function of two uncertain variables the wind direction and wind speed. The polynomial chaos based on regression performs best for all layouts.

In what follows we will compare the PC-R (the best performing method) with the rectangle rule (the method currently used in practice) to quantify the reduction of samples needed to compute the AEP accurately. Also, we will sometimes use polynomial chaos to refer to polynomial chaos based on regression. Figure 8 only keeps PC-R and the rectangle rule results from Fig. 6, and in addition, the figure shows the average AEP error computed with 10 and 100 sets of samples for each method. For the rectangle rule, there is hardly any difference between the average AEP error computed with 10 or 100 sets. For the PC-R method, the average error with 100 sets shows a smoother convergence. In general, averaging the AEP error over 10 sets of samples is enough to minimize the AEP’s sample location sensitivity and to clearly see the differences between the methods used for computing the AEP (Sect. 5.3).

In Fig. 8, we see that the PC-R convergence curve is consistently below the rectangle rule curve, i.e., PC-R has a smaller error for the same number of samples or the same error for a smaller number of samples. Using the 1% average AEP error
Figure 7. The average standard deviation of the energy (STD) error as a function of the number of samples for polynomial chaos (based on regression and quadrature), the rectangle rule and Monte Carlo. The STD is a function of two uncertain variables the wind direction and wind speed. Note that the PC-R response is biased for the Grid and Amalia layouts. The PC-R underpredicts the true STD at the expense of computing the mean (AEP) more accurately (see Fig. 6).

as a metric, we see that PC-R achieves this error with fewer samples than the rectangle rule. The reduction in the number of samples is on the order of six times for the Grid layout, eight times for the Amalia, ten times for the Optimized and no significant improvement for the Random. These reductions in the number of samples are considerable. In addition to providing faster convergence of the AEP, the polynomial chaos based on regression method converges the AEP more smoothly—less oscillatory, more monotone convergence. Smooth convergence is always a desired property, and it especially useful when performing an optimization. Both methods, PC-R and the rectangle rule, perform better for the less grid-like layouts because there is less variability in the power responses as a function of wind direction (see Fig. 5). This is beneficial as in the first few iterations of an optimization the starting grid-like layouts become non-grid-like and start to resemble the Optimized layout.
Figure 8. The average AEP error as a function of the number of samples for both the rectangle rule and polynomial chaos based on regression (this is the same as Fig. 6, with only keeping the PC-R and the rectangle rule results). In addition, we show the average AEP error computed with 10 (solid line) and 100 (dashed line) sets of samples (Sect. 5.3). The AEP is a function of two uncertain variables the wind direction and wind speed. The polynomial chaos method computes the AEP more accurately with fewer samples.

The polynomial chaos response-based on regression is not only better on average, as we have seen in Fig. 8, but it is also better in general, as shown in Fig. 9. The shaded area in Fig. 9 shows the spread between the maximum and minimum AEP for 10 realizations of each of the number of samples (Sect. 5.3). The solid line shows the average of those 10 realizations and the dashed lines the ±1 % of the baseline AEP. We see that the spread is significantly smaller for the polynomial chaos based on regression and that by around 300 samples the predictions are almost always within 1 % of the true AEP for the grid-like layouts (Grid and Amalia) and around 100 samples for the non-grid-like layouts (Optimized and Random). In contrast, for the rectangle rule, the error in the AEP is still larger than 1 % at 600 samples for the Amalia layout and 500 samples for the Optimized layout.
Figure 9. Variability in the convergence of the AEP. The AEP is a function of the wind direction and wind speed. The figures of the left column are for polynomial chaos based on regression and those of the right column for the rectangle rule. The shaded area shows the spread between the maximum and minimum AEP for 10 realizations of each of the number of samples. The range in the plots corresponds to ±5 % of the true AEP for each layout. The variability is significantly smaller for polynomial chaos, which shows that in general, it outperforms the rectangle rule. The PC-R method obtains the smaller variability by smoothing out the high-frequency oscillations present in the power response (Fig. 5).
6.3 Wind farm layout optimization

6.3.1 Optimization problem

The objective of the wind farm layout optimization is to maximize the AEP (Sect. 2.3) by changing the position of the wind turbines. We assume a fixed number of turbines, 60, of the same type (NREL 5-MW (Jonkman et al., 2009)) and constrain the turbines to stay within a given area and with a minimum separation between them. This objective and constraints result in a nonlinear optimization under uncertainty problem with deterministic constraints:

\[
\begin{align*}
\text{maximize} & \quad AEP(x, y, \xi) \\
\text{subject to} & \quad S_{i,j} \geq 2D \quad i, j = 1 \ldots 60 \quad i \neq j \\
& \quad N_{i,b} \geq 0 \quad i = 1 \ldots 60 \quad b = 1 \ldots 14,
\end{align*}
\]

where \( S_{i,j} \) is the distance between each pair of turbines \( i \) and \( j \), and \( D \) is the turbine diameter. The normal distance, \( N_{i,b} \), from each turbine \( i \) to each boundary \( b \) is defined as positive when a turbine is inside the boundary, and negative when it is outside of the boundary. The boundary is the convex hull of the Princess Amalia layout—a fourteen-sided convex polygon (dashed-line boundary in the upper-left of Fig. 10). The design variables are the \( x, y \) coordinates of the 60 wind turbines (same number as in the Amalia layout), resulting in 120 design variables. The uncertain variables \( \xi \) in the objective are the wind direction and wind speed.

We solve the optimization problem with the gradient-based sequential quadratic programming optimizer SNOPT (Gill et al., 2005). We use OpenMDAO (Gray et al., 2010) and its wrapper for pyOptSparse (Perez et al., 2012) to call SNOPT from Python. We scale the variables, constraints and the objective to make them of order one, and set the tolerances to \( 1 \times 10^{-4} \) per the function (AEP) precision.

6.3.2 Optimization results

The AEP in the optimization objective is a function of two uncertain variables the wind direction and wind speed, with probability distributions given in Sect. 5.1. We solve the optimization under uncertainty problem, Eq. (38), for two different starting layouts: Amalia and Random. We compute the AEP (objective) with different precision (number of samples) and with different methods: the rectangle rule, and polynomial chaos based on quadrature and regression. For each method, we run 10 optimizations, where each optimization uses different sample points to compute the AEP (see Sect. 5.3). The 10 optimizations enable us to get a better understanding of which method is better at finding layouts with high AEP and to avoid drawing conclusions from one-off local optima.

The results of the optimizations are reported in Table 2. The table contains the statistics of the AEP of the optimal layouts for the 10 optimizations of each method. The values of the AEP reported in Table 2 are computed with 200,000 Monte Carlo samples. The same 200,000 samples—wind direction, wind speed pairs—are used to test the optimal layouts obtained by each method.
Polynomial chaos based on regression on average produces the best optimums when the different methods to compute the AEP use roughly the same number of samples ($\sim 225$). Furthermore, PC-R optimums are comparable with the optimums obtained with the rectangle rule that used 625 samples to compute the AEP. Starting the optimizations from an already good layout (Amalia), will in general, find layouts that are better than starting from a bad layout (Random). The values of the AEP reported are computed with 200,000 Monte Carlo samples.

<table>
<thead>
<tr>
<th>Method</th>
<th># samples</th>
<th>Starting layout: Amalia AEP statistics (GWh)</th>
<th>Starting layout: Random AEP statistics (GWh)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mean</td>
<td>max</td>
</tr>
<tr>
<td>PC-Q</td>
<td>225</td>
<td>1174</td>
<td>1179</td>
</tr>
<tr>
<td>PC-R</td>
<td>231</td>
<td>1182</td>
<td>1184</td>
</tr>
<tr>
<td>rect</td>
<td>225</td>
<td>1179</td>
<td>1180</td>
</tr>
<tr>
<td>rect-fine</td>
<td>625</td>
<td>1183</td>
<td>1184</td>
</tr>
</tbody>
</table>

Polynomial chaos based on regression on average produces the best optimums when the different methods to compute the AEP use roughly the same number of samples ($\sim 225$). Furthermore, PC-R optimums are comparable with the optimums obtained with the rectangle rule that used 625 samples to compute the AEP. Starting the optimizations from an already good layout (Amalia) will, in general, find layouts that are better than starting from a bad layout (Random). When considering all the methods, on average, the optimal layouts starting from the Amalia have an AEP that is 0.62% higher than those starting from the Random layout. However, starting from random layouts gives the turbines more freedom to move around the design space with the potential of finding novel and better layouts.

The optimal layouts with the maximum AEP for each method and starting layout are shown in Fig. 10. For the optimums obtained starting from the Amalia layout, we see that the distribution of the turbines is similar to the one of the starting layout. And for the optimums starting from the Random layout, we see that the optimal layouts obtained are less grid-like and produce more novel layouts. For each method, as shown in Fig. 10, the optimal layouts starting from the Amalia have a higher AEP than those from the Random. But a random start has the potential to find better optimums as the turbines explore more of the design space (Fig. 11).

A typical optimization history as a number function calls is shown in Fig. 12. A function call is the AEP computation which requires hundreds (the number of samples specified for each method in Table 2) of calls to the wake model for the wind farm resulting in tens of thousands of calls to the wind farm wake model per optimization. If the gradients of the AEP are computed with a first-order finite difference, 120 (the number of design variables) times as many wind farm wake model calls would be needed per optimization. The optimization starting from the Amalia layout (Fig. 12a) converges faster than that starting from the Random layout (Fig. 12b). The optimization with the Amalia starting layout converges in 114 optimization iterations as opposed to 367 for the Random starting layout. The design variables (turbines’ positions) change an order of magnitude less for the Amalia starting layout (Fig. 12c) than for the Random starting layout (Fig. 12d). For the Amalia case, the cumulative
Figure 10. Optimal wind farm layouts achieved for each method to compute the AEP. The layouts in the first column show the turbines starting position for the optimization along with the boundary constraint (dashed-line). The optimal layouts correspond to those with the maximum AEP from Table 2. The first row shows the optimums obtained starting from the Amalia layout; we see that the distribution of the turbines is similar to the one of the starting layout. The second row shows the optimums starting from the Random layout; we see that the optimal layouts obtained are less grid-like and produce more novel layouts. For each method, the optimal layouts starting from the Amalia have a higher AEP than those from the Random.

Figure 11. The initial and optimized position of the turbines. The turbines starting from the Random layout move more and explore more of the design space. These are the optimum layouts obtained with the PC-R method from Fig. 10.
**Figure 12.** Convergence history of the optimization. The top row shows the objective convergence and bottom row shows the design variable convergence. The left column is for the optimization with the Amalia starting position, and the right column is for the optimization with the Random starting position (note the different scales). The optimization of the Amalia starting layout converges in fewer AEP computations. The final AEP for both starting layouts is similar. The turbines move more and thus explore more of the design space for the Random starting layout.

Displacement of all turbines measured in turbine diameters is 315 and for the Random case 2790. In general, most optimizations we performed behaved similarly to those shown in Fig. 12. The optimizations starting from the Amalia layout run for fewer iterations (usually on the order of 100) with less design variable change than those starting from the Random layout. Most of the optimizations we ran either successfully converged or were on their way to satisfying the convergence criteria before they reached the time limit we set to save computational cost.

To properly compare the results obtained by the different methods used to compute the AEP in the optimization, we should compute the AEP of the optimal layout with a method that was not used for the optimization. We use 200,000 Monte Carlo samples—wind direction, wind speed pairs—to test the AEP of the optimal layouts. For the optimal layouts in Fig. 10, with the Amalia starting layout, the average improvement in AEP over the starting layout for all the methods is 0.14 % when measured with the Monte Carlo samples and 1.74 % when measured by the method used in the optimization. For the optimizations starting from the Random layout, the average AEP improvements are 15.70 % for Monte Carlo and 18.37 % for the method.
Table 3. The improvement in AEP of the optimized layouts from Fig. 10 over their starting layouts. The improvements are smaller when measured by 200,000 Monte Carlo samples rather than directly by the method used to compute the AEP in the optimization.

<table>
<thead>
<tr>
<th>Method</th>
<th>Starting layout: Amalia</th>
<th>Starting layout: Random</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AEP % improvement as measured</td>
<td>AEP % improvement as measured</td>
</tr>
<tr>
<td></td>
<td>by Monte Carlo</td>
<td>by method</td>
</tr>
<tr>
<td>PC-Q</td>
<td>-0.09</td>
<td>1.23</td>
</tr>
<tr>
<td>PC-R</td>
<td>0.35</td>
<td>2.02</td>
</tr>
<tr>
<td>rect</td>
<td>0.02</td>
<td>1.35</td>
</tr>
<tr>
<td>rect-fine</td>
<td>0.30</td>
<td>2.36</td>
</tr>
<tr>
<td>average</td>
<td>0.14</td>
<td>1.74</td>
</tr>
</tbody>
</table>

used in the optimization (see Table 3). The reporting of the AEP computed by Monte Carlo explains why some of the results reported in Table 2 and Fig. 10 have a smaller AEP than its starting layout.

The improvements starting from the Amalia layout are similar to those found by Gebraad et al. (2017) for turbine position optimization. The large improvements in AEP for the layouts starting from the Random layout show that the optimizer can find good layouts from a bad starting position. Thus, to search for the best layout, we would perform many optimizations with random starts.

From all the optimizations runs, we conclude that computing the AEP with polynomial chaos based on regression produces the best layouts—around 1% higher AEP when starting from a random start—for the same number of samples. This is because PC-R computes the AEP more accurately than the other methods for the same number of samples (Sect. 6.2). Furthermore, the optimization with PC-R finds optimums comparable to those found with the rectangle rule that used roughly three times as many simulations. We found that starting an optimization from a good layout will, in general, find better optimums than starting from a random layout, but starting from the random layout can lead to novel layouts and possibly better layouts as the turbines explore more of the design space. Finally, to properly compare between methods an ensemble of optimizations should be used and evaluated in the same way.

The application we considered was the layout optimization of a wind farm. The goal of

7 Discussion and conclusions

A single wind farm layout optimization requires tens to hundreds of thousands of model evaluations. During the design phase of a new wind farm, designers would need to perform many optimizations. The designers may explore scenarios with different turbine types, different sites, larger farms with a different number of turbines and possibly even systems of wind farms. 

The rectangle rule used almost three times as many simulations to compute the AEP for each optimization iteration (see the number of samples in Table 2), which results in the optimization using roughly three times as many simulations.
Also, the presence of local optima would require many optimizations with different restarts to find the best layout. And, furthermore, there is a desire to increase the fidelity of the models used to simulate the wind farm, which will increase the time and computational cost of the optimizations\(^9\). Thus, to facilitate the design phase of new wind farms and to incorporate the use of higher fidelity models in the design phase requires significant improvements to reduce the simulation requirements (computational cost) of performing the optimization was to maximize the Annual Energy Production (AEP) of the wind farm.

The layout optimization is an optimization-optimization under uncertainty (OUU) problem (Fig. 1c) with the wind speed and wind direction as uncertain variables and the positions of the wind turbines as design variables. The AEP was computed with the uncertainty quantification method.

We proposed the use of polynomial chaos (PC). We tailored the use of PC and extended PC to reduce the number of high-fidelity simulations needed to accurately compute statistics, such as the AEP, and the gradient of those statistics.

We have shown to improve the uncertainty quantification step of the OUU problem. We showed that polynomial chaos based on regression (PC-R) could compute the AEP accurately using up to an order of magnitude fewer simulations than the rectangle rule, the method currently used in practice in the wind industry. For the case when the AEP is a function of two uncertain variables, wind direction and wind speed, polynomial chaos based on regression-PC-R computes the AEP accurately (error less than 1%) usually with only two hundred simulations or less depending on the wind farm layout—a significant improvement over the current industry practice of using more than a thousand model evaluations to compute the AEP.

The layout of the wind farm influences the convergence of the AEP because the layout has a significant effect on the power output of the farm as the wind conditions vary (Fig. 5). We considered four representative layouts: Grid, Amalia, Optimized and Random. The power response of the grid-like layouts (Grid and Amalia) has large oscillations caused by the large drops in power that occur when rows of wind turbines are aligned with particular wind directions. Because of the larger variability in the power response, the grid-like layouts require more simulations than the non-grid-like layouts (Optimized and Random) to converge the AEP, independent of the method used to compute the AEP.

We extended polynomial chaos to obtain gradients. An extension of this work that could potentially further improve the convergence of the AEP, especially for the grid-like layouts, would be to build an approximation to the power output that takes into consideration the oscillatory and periodic behavior of the wind farm response with respect to the wind direction (Fig. 5). For instance, by using Fourier series to approximate the wind direction response and polynomials for the wind speed response.

The benefit of using polynomial chaos is that in addition to computing the AEP efficiently it can also compute the gradient of the AEP efficiently. The use of gradients is essential to enable large-scale wind farm optimization. We described how to compute the gradients of the AEP in (Sect. 4.3). Another benefit of using PC is that it can easily incorporate multiple fidelity models to accelerate the convergence of statistics (Ng and Eldred, 2012; Palar et al., 2016), such as the AEP from the gradients of (Padrón, 2017). Furthermore, the power at the simulation samples. Making use of the gradients of the AEP, we

\(^9\)If instead of using the Floris wake model in the optimization, we had used a high-fidelity fully resolved Large Eddy Simulation to model the wind farm, a back of the envelope calculation shows that the optimization would require the total Annual Energy Production (AEP) of the wind farm to power the computers used in the simulation.
performed multiple-effect of other uncertain variables (such as wake model parameters (Padrón, 2017)) on the AEP, can easily be incorporated in the polynomial chaos framework.

Here, we assumed that the uncertain variables—wind direction and wind speed—are independent. In general, this is not the case as the wind direction and wind speed are usually correlated. The polynomial chaos method has been generalized to accommodate correlated variables. One approach is to perform a (linear or non-linear) variable transformation (Feinberg and Langtangen, 2011) to uncorrelate the variables before applying polynomial chaos. These transformations can degrade the convergence rate (Eldred et al., 2008; Oladyshkin and Nowak, 2012).

Another approach is to construct polynomials that are orthogonal to the multivariate distribution instead of orthogonal polynomials for each dimension (Navarro et al., 2014). A third approach is to find subsets of the wind direction in which the wind direction and wind speed can be considered independent, and then combine the polynomial chaos expansions of each of the subsets (Murcia et al., 2015). This is known as multi-element polynomial chaos (Wan and Karniadakis, 2006). An interesting extension of this work would be to study the effect of different wind distributions, including correlated distributions, on the convergence of the AEP.

We performed multiple gradient-based wind farm layout optimizations to compare the optimization results obtained with the different methods to compute the AEP: polynomial chaos based on regression, polynomial chaos based on quadrature, and the rectangle rule. We showed that to properly compare methods an approach is needed to determine the optimal layouts obtained using the different methods to compute the AEP. Also, for proper comparison, the AEP of the optimized layouts should be evaluated in the same way and with a method different from the one used in the optimization. To be confident about the differences between the optimal layouts, we evaluated the AEP of each optimized layout using the same 200,000 Monte Carlo samples. We found that the benefits of being able to efficiently compute the AEP with PC-R translate to being able to find better optima than those obtained when computing the AEP with the rectangle rule when using the same number of simulations. With PC-R we find optima that are 1% better than those found with the rectangle rule. This is a significant improvement as a 1% increase in the AEP for a modern large wind farm can increase its annual revenue by millions of dollars.

An interesting and relevant extension of the optimization work presented here would be to allow for yaw-based wake control. Including wake deflection in the wind farm layout optimization increases the complexity of the problem in terms of the number of design variables because every wind direction (and to a lesser extent, wind speed) calls for different yaw values for each turbine. The increased number of design variables makes the problem much more difficult for gradient-free optimization methods and can slow down gradient-based optimization methods as well (Thomas et al., 2017). As we have seen, polynomial chaos reduces the number of simulations (wind directions and wind speeds considered). Thus it could greatly reduce the number
of simulations in optimization with yaw control. The layouts obtained by including yaw in the layout optimization are similar to those obtained without yaw in the layout optimization (Gebraad et al., 2017). However, the expected gains from optimized yaw are non-trivial. Gebraad et al. (2017) found that including yaw for each wind direction in a secondary layout optimization, after layout optimization without yaw, resulted in an additional 3.7% increase in AEP.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. Funding for this research was provided by the National Science Foundation under a collaborative research grant No. 1539384 and 1539388.
References


Interactive comment on “Polynomial chaos to efficiently compute the annual energy production in wind farm layout optimization” by Andrés Santiago Padrón et al.

Andrés Santiago Padrón et al.

padronas@gmail.com

Received and published: 15 April 2018

Thank you for taking the time to thoroughly review the manuscript and for your insightful comments. We have updated the manuscript to reflect your comments as detailed below under each specific comment. In general, we included many of the comments in a new Discussions and conclusions (DC) section that greatly expanded our previous Conclusions section.

RC2 - Comment 1: "PC is interpolating/regressing figure 5, the power as a function of wind-speed (y) and direction (x), which shows a highly irregular pattern in the x. I suggest polynomials may be quite a poor choice for approximating this function. This
could be verified by the authors if they plotted the implied response surfaces of PC-R and PC-Q and compared with this reference - oscillations may be present, as well as high sensitivity to the sample locations (hence perhaps their 10 runs with varying samples). In contrast the rectangle rule is just "pixellating" Fig 5. Given the periodicity of and shape of x, I would use combination of a Fourier series in x, and a polynomial in y. An equivalent integral approximation can be built, and since the underlying representation better matches the response, the AEP should be better."

We agree that one could potentially build a better approximation in the x-direction, especially using more samples, but we are not sure that for the same number of samples it could provide a better estimate of the AEP. We've included a couple of sentences with your suggestions and to encourage future research to look into other approximations at the end of the third paragraph of the new Discussions and conclusions (DC) section. One of our goals of this work is to encourage people to look into better ways to approximate the AEP than the rectangle rule, and we will be happy to see others try to improve the results shown here.

Also, we have included a paragraph (paragraph 4 of DC) discussing why polynomial chaos is a good choice for this problem. We have also emphasized the fact that we show how to compute the gradients of the statistics with polynomial chaos. The gradients are novel for the case of the polynomial chaos based on regression. These changes have been made in a couple of places including modifying a sentence in the abstract and adding a sentence in the introduction, among other places (see the pdf marked up with the differences).

During our work, we have visualized the polynomial approximations and the rectangle rule approximations to the power response (figure 5), they indeed do not exactly match the power responses shown in figure 5. However, the approximations do get better as the number of samples used to construct the approximation increases, as can be seen in figure 8. The power response is a complicated function, and the approximations we build with only a couple hundred samples are not going to be able to capture all the
high-frequency oscillations, but they can give good estimates of the AEP as we have shown. For reference, Figure 5 was constructed using 32,400 samples (a grid made by 360 wind directions (every degree) and 90 wind speeds). Also, as mentioned in the paper, even using 200,000 MC samples to estimate the AEP only ensures with 99 % confidence that the true AEP is within ± 1 % of estimated AEP.

RC2 - Comment 2: "The fact that PC is perhaps not a good choice here, is additionally suggested by that fact that it performs only very slightly better than the rectangle rule (I don’t agree with the authors interpretation of significant improvements in Figures 6-8). Rectangle should be 2nd-order while PC should be spectral. I suggest the lower variance of PC-R in Fig 9 is most likely the effect of PC-R filtering noise with regression. PC is likely not significantly improving the representation of Fig 5, compared to the under-sampling of the rectangle rule."

We believe that reductions of up to an order of magnitude in the number of simulations are significant. As mentioned above, we have included a paragraph (paragraph 4 of DC) discussing why polynomial chaos is a good choice for this problem, in addition to that of computing the AEP more efficiently than the rectangle rule. You are correct in your observation of Fig 9. We have added this to the caption of Fig 9.

RC2 - Comment 3: "The authors should not underestimate the effect non-independence of wind-speed and direction may have on the AEP. In my experience (in unrelated problems) dependence relationships in inputs are significantly more important than non-Gaussianity (skewness, kurtosis, etc.) of 1d-marginals. This makes the careful choice of Weibull potentially irrelevant for the purposes of comparing layouts. Please plot the 2d distribution in Fig 3, so we can see how strong the dependence is. Mention how PC could be generalized to allow for this (there is some literature on the subject). Computing the effect of this on the AEP would also be a very nice addition."

We have made it clearer that we constructed the wind direction and wind speed to be independent. We use the NoordZeeWind meteorological mast data, which is depen-
dent, as inspiration to create the probability distributions (see the modifications in 5.1 Probability distributions of the uncertain wind conditions). In the paper, we provide a reference that shows a plot of the experimental correlated data we used as a base to construct the independent distributions.

We have added a paragraph (paragraph 5 of DC) discussing different methods to generalize PC for dependent variables, and we have stated that an interesting extension to the paper would be to study the effects of different wind distributions, including correlated distributions, on the convergence of the AEP.

RC2 - Comment 4: "Justify why wind-speed is fit with a distribution, but direction not."

Both, the wind speed and wind direction distributions are specified by histograms containing 50 bins of equal width. We have clarified this in the updated section 5.1 Probability distributions of the uncertain wind conditions. We created the histogram data for the wind direction by linearly interpolating the experimental data. For the wind speed, we did a two-step approach. First, we fitted a Weibull to the experimental data, then we truncated the Weibull and used its likelihood values to create the histogram. We fitted the Weibull to smooth the data and because this is commonly done for the wind speed (we have included a couple of citations referring to this in section 5.1 as well). Granted, we could have constructed the speed histogram without fitting the Weibull, but we followed the wind industry standard of fitting the wind speed with a Weibull.

RC2 - Comment 5: "Justify why computing time of this problem is relevant. This is a one-off optimization for a farm that might last 20 years."

The final optimization would be a one-off optimization. However, to get to the final optimization, many optimizations are required during the design phase. For example, designers may need to explore scenarios with different turbine types, different sites, larger farms with a different number of turbines and possibly even systems of wind farms. Also, the presence of local optima would require many optimizations with different restarts to find the best layout. And, furthermore, there is a desire to increase the
fidelity of the models used to simulate the wind farm, which will increase the time and computational cost of the optimizations. Thus reducing the computational time is relevant as it will enable the use of higher fidelity models and also facilitate the exploration of many different and larger wind farm designs. In the paper, we have included this justification of why the computing time is relevant in the first paragraph of the Discussions and conclusions section.

RC2 - Comment 6: "Given your results it seems that the layout problem could have a very large subspace of close-to-optimal designs - all essentially equivalent. Do you agree? Please comment."

We agree. We have commented on this in the new Discussions and conclusions (DC) section. Specifically, at the end of paragraph 6 of DC section.

RC2 - Comment 7: "I have a personal interest in wake-deflection, which is mentioned in connection with FLORIS. Could the authors comment on how the layout problem would change if optimal wake deflection were allowed?"

We have commented on this in the last paragraph of the Discussions and conclusions section.

Interactive comment on “Polynomial chaos to efficiently compute the annual energy production in wind farm layout optimization” by Andrés Santiago Padrón et al.

Andrés Santiago Padrón et al.

padronas@gmail.com

Received and published: 15 April 2018

Thank you again for your comments and questions. We really appreciate them. We just wanted to follow up on our previous response to inform you that we have now addressed your comments in the paper. Specifically, see the fifth paragraph of the Discussions and conclusions section.