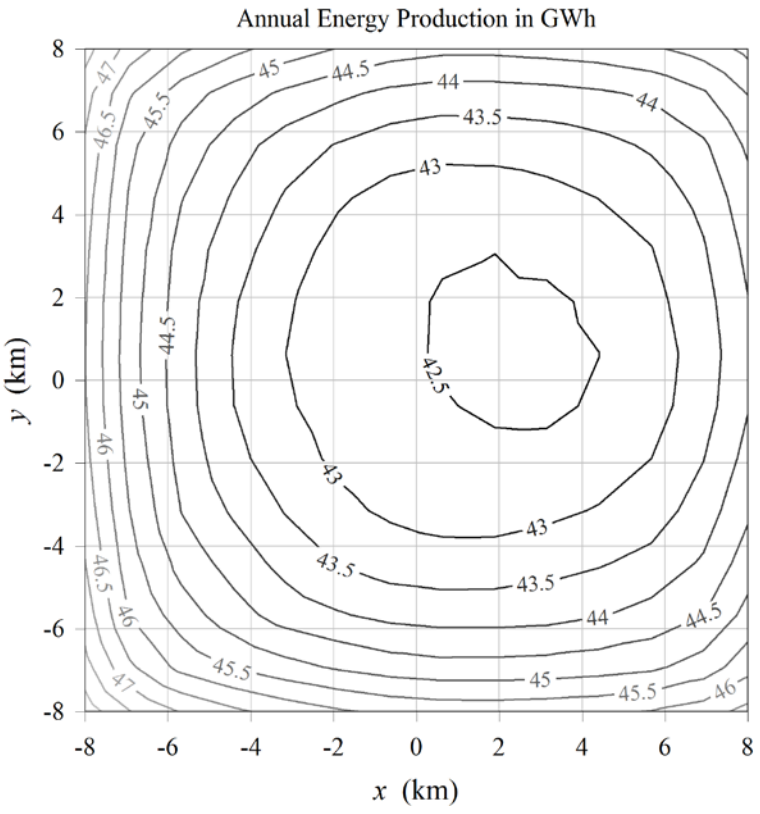
SINTEF

TR A7382- Unrestricted

Report

Viper: A Tool for Computing Energy Production of Large Offshore Wind Farms

Author(s)
Karl O. Merz



SINTEF Energy Research
Electric Power Systems
2014-02-03

SINTEF Energi AS
SINTEF Energy Research
Address:

NO-
NORWAY
Telephone:+47 73597200
Telefax:+47

energy.research@sintef.no
www.sintef.no/energi
Enterprise /VAT No:
NO 939 350 675 MVA

KEYWORDS:
wind farm
boundary layer
energy production

Report

Viper: A Tool for Computing Energy Production of Large Offshore Wind Farms

VERSION
1.0

DATE
2014-02-03

AUTHOR(S)
Karl O. Merz

CLIENT(S)
NOWITECH

CLIENT'S REF.

PROJECT NO.
502000059-4

NUMBER OF PAGES/APPENDICES:
24

ABSTRACT

Abstract heading

Viper (Vindpark Energiproduksjon) is a collection of Matlab scripts for computing the annual energy production of large offshore wind farms. The analysis method is based on the numerical finite-difference solution of the two-dimensional boundary-layer equations. The method has been validated against empirical formulas for flat plates, and full-scale measurements at the Horns Rev and Nysted wind farms.

PREPARED BY
Karl O. Merz

SIGNATURE
Karl O. Merz 03. Feb 2014

CHECKED BY
Harald G. Svendsen

SIGNATURE
Harald G Svendsen 4 Feb 2014

APPROVED BY
Knut Samdal

SIGNATURE
Knut Samdal

REPORT NO.	ISBN	CLASSIFICATION	CLASSIFICATION THIS PAGE
TR A7382	978-82-594-3581-1	Unrestricted	Unrestricted

Document history

VERSION	DATE	VERSION DESCRIPTION
1.0	2014-02-03	Original document

SINTEF

Table of contents

1 Background

Viper (Vindpark energiproduksjon) is a set of Matlab[1] scripts that can be used to calculate the annual energy production from a large offshore wind farm. The results can be used in wind farm layout assessment, or to evaluate the influence on energy production of different wind farm control strategies.

The method is based on a numerical solution of the two-dimensional boundary-layer equations, using an eddy viscosity/mixing length approach to account for the effects of turbulence. This provides an estimate of hub-height velocity at each wind turbine, from which the power output is computed. While two-dimensional boundary layer theory is simplified with respect to the real flow conditions, Viper provides predictions which match data collected at existing offshore wind farms. On the same theoretical basis, the method is applicable to non-standard cases, such as wind farms with variable turbine diameters, tower heights, rated powers, and spacings; or wind farms consisting of several blocks of turbines, with separation between the blocks.

Local wake models, like that developed by Jensen [10], are commonly employed to compute the power output of turbines within a wind farm.[2] Jensen's model assumes that the wake behind each turbine is axisymmetric, and that the diameter of the wake grows linearly with the downstream distance. The wake velocity then follows from the conservation of momentum. Individual turbine wakes are superposed. The assumption of axisymmetry and superposition is reasonable for the wakes behind a couple rows of wind turbines, but it is not valid deep within a large wind farm, where the wind farm influences flow patterns in the atmospheric boundary layer well above the height of the turbines. The Jensen model is also not valid for predicting the recovery of windspeed as a function of distance downstream of a wind farm.

It has been recognized since the early days of commercial wind energy that a wind farm influences the atmospheric boundary layer, and that this in turn dictates the recovery of windspeed – the turbulent mixing of higher-velocity air from above – both within and downstream of the wind farm.[3] The simplest way to account for the development of the boundary layer is by representing the wind farm as an equivalent surface roughness parameter. An increased surface roughness exerts drag on the flow, progressively slowing and thickening the boundary layer. Frandsen [7], Calaf *et al*. [2], and Meneveau [15] describe models of this type. The accuracy is dependent on the empirical calibration of the surface roughness parameter.

A more robust class of numerical methods has been developed by atmospheric scientists seeking to determine the influence on the atmospheric boundary layer of forests, buildings, and other obstructions. These are known as "canopy" methods; Viper can be placed in this category. The body of literature on canopy methods is vast, and the scope is here limited to wind energy applications. In particular, motivated by the development of large offshore wind farms, Frandsen *et al*. [8],[9] have developed a practical engineering method that considers simultaneously local and global wake effects. The global model employs the canopy method described by Belcher *et al*. [1]. This differs from Viper in that the former linearizes the boundary layer equations and solves them in Fourier space, whereas Viper directly solves the nonlinear boundary layer equations.

1.1 Coordinate Systems

Three coordinate systems are relevant to the Viper analysis: farm, wind, and analysis coordinates. These are sketched in Figure 1. When it is necessary to refer a variable to a particular coordinate system, this is done with a superscript "f", "w", or "a".

[1] ... or its free equivalent Octave, which was used to develop the code ...

[2] Variations on Jensen's method are referred to in the literature as the Katic model or the Park wake model. Lackner and Elkinton [14], Kusiak and Song [13], Chowdhury *et al*. [3], and Turner *et al*. [21] are among recent studies that have employed this type of wake model.

[3] Templin [20] was the first to introduce the idea. Milborrow [16] provides a survey of such studies conducted during the 1970's.

The farm coordinate system remains fixed throughout the analysis. The locations of the turbines are defined relative to the farm coordinate system. In the present code it is assumed that the y^f axis points North, however this is not a requirement of the analysis method. The z^f axis points vertically upwards.

The analysis is performed considering various directions of the remote wind V_∞. The wind coordinate system is defined such that the x^w axis is aligned with the direction of the remote wind. The associated compass direction is measured in the opposite sense, and points into the wind.

The x^a axis of the analysis coordinate system is also aligned with the wind direction, but the origin may be offset relative to the wind coordinate system. The y^a axis points vertically upwards, such that the $x^a y^a$ plane forms a 2D section along which the boundary layer flow is calculated. For a given wind direction, there are in general several analysis lines, and there is an analysis coordinate system associated with each.

Note that the analysis coordinate system is defined with the y^a axis vertically upwards, which differs from the convention for the farm and wind coordinate systems, whose z axes point vertically upwards. The analysis coordinate system was defined in this manner in order to make the derivation of the 2D boundary layer equations consistent with that of Schlichting [19]. To put it another way, the analysis coordinate system is most naturally viewed "from the side", while the farm and wind coordinate systems are most naturally viewed "from above".

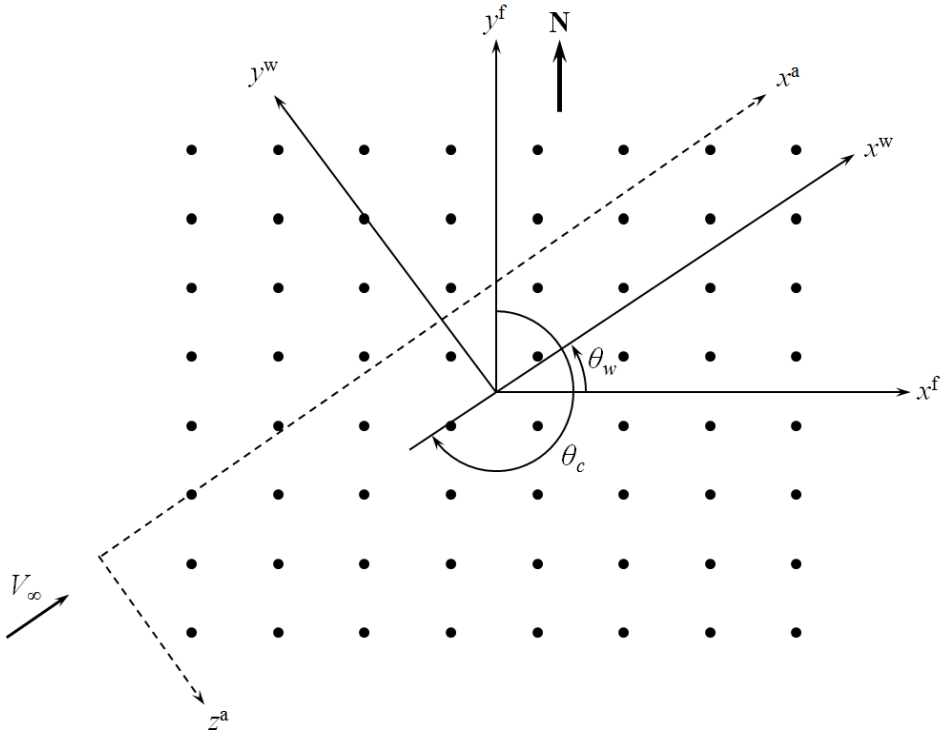


Figure 1: Coordinate systems, illustrated for a wind farm with a regular layout pattern, and for a given wind direction

2 Boundary Layer Analysis

Viper, as well as other engineering methods such as Frandsen *et al.*, solve the boundary-layer equations in two dimensions. The x^a axis points downwind, and the y^a axis is vertical. (The derivation is performed in analysis coordinates, therefore the "a" superscript is henceforth omitted.) The equations are written and solved in the xy plane. There is theoretical and experimental justification for limiting the analysis to two dimensions. A representative height of the atmospheric boundary layer is 1 km [2], while the lateral dimension of a large wind farm is at least several kilometers, and velocity recovery in the wake of a large wind farm is measured in tens of kilometres. The lateral length scale is thus an order of magnitude larger than the vertical length scale. This implies that the turbulent mixing of high-speed wind into the farm

and its wake occurs predominantly from above, and not from the sides. Sattelite measurements by Christiansen and Hasager [4] on the Horns Rev wind farm, as well as 3D CFD analyses by Réthoré [18], support this view. The wake of the wind farm does not change in width as a function of downstream distance, indicating little mixing from the sides.

2.1 2D Boundary Layer Equations

The two-dimensional, steady, turbulent boundary-layer equations are:[4]

$$\text{momentum:}\quad \overline{u}\frac{d\overline{u}}{dx}+\overline{v}\frac{d\overline{u}}{dy}=-\frac{1}{\rho}\frac{d\overline{p}}{dx}+\frac{\mu}{\rho}\frac{d^2\overline{u}}{dy^2}-\frac{d\overline{(u'v')}}{dy} \tag{2.1}$$

$$\text{continuity:}\quad \frac{d\overline{u}}{dx}+\frac{d\overline{v}}{dy}=0 \tag{2.2}$$

subject to the boundary conditions:

$$\overline{u}=\overline{v}=0 \text{ at } y=0;\quad \overline{u}=u_e \text{ at } y=h_e. \tag{2.3}$$

Here the overbar denotes a time-average, and, as all quantities will be time-averaged through the remainder of the derivation, the overbars are henceforth omitted. The variables u and v are the x and y components of the mean velocity, while u' and v' are turbulent fluctuations about the mean. The fluid viscosity is μ, and the density is ρ. The pressure gradient dp/dx is negligible in the present application, where the remote wind is assumed to be steady and the surface topography is flat.

2.2 Eddy Viscosity and Mixing Length

Equation 2.1 includes the product $u'v'$, which is unknown. Prandtl argued that:

$$u'v'=-\kappa^2 y^2\left(\frac{du}{dy}\right)^2, \tag{2.4}$$

with $\kappa\approx 0.4$. Equation 2.4 can be regarded as an empirical fact.[5] The term $d(u'v')/dy$ acts as a positive quantity on the right-hand side of Equation 2.1, and its effects can be considered as increasing the effective viscosity of the fluid – the "eddy viscosity" – in a nonlinear manner. The terminology "mixing length" comes from Prandtl's reasoning as to why $u'v'$ is proportional to y^2; the details are not relevant here.

With Equation 2.4, the momentum equation becomes:

$$u\frac{du}{dx}+v\frac{du}{dy}=\frac{\mu}{\rho}\frac{d^2u}{dy^2}+2\kappa^2 y\left(\frac{du}{dy}\right)^2+2\kappa^2 y^2\frac{du}{dy}\frac{d^2u}{dy^2}. \tag{2.5}$$

2.3 Thrust Forces

The thrust force of a given wind turbine on the air F_T can be written in terms of a nondimensionalized thrust coefficient C_T:

$$F_T=\frac{1}{2}\rho\frac{\pi D^2}{4}u_h^2\,C_T. \tag{2.6}$$

[4] Unless stated otherwise, Schlichting [19] is the reference for all equations developed in this section.

[5] There is also theoretical justification in the case of simple wall-bounded shear flows, as a relationship such as Equation 2.4 is necessary in order for the velocity profiles of the inner viscous region and outer turbulent region to overlap. [5]

Here D is the rotor diameter and u_h is the hub-height windspeed, taken to be representative of the rotor-average windspeed.[6] For a given wind turbine design, the steady-state thrust coefficient can be tabulated as a function of mean windspeed; thus C_T is a function of u_h.

The terms in Equation 2.5 each represent a force per unit volume, divided by the fluid density ρ. The thrust force must be expressed in the same manner, in order that it may be incorporated into the momentum equation.

In a given wind farm layout, let there be a surface area associated with each turbine. For example, in a regular rectangular pattern with x^a spacing s_x and z^a spacing s_z, (or equivalently x^w and y^w spacing, respectively), the surface area would be $A_s = s_x s_z$.

Now a crucial assumption is made: the thrust force from each wind turbine is distributed over a volume of air spanning the diameter of the turbine and the surface area A_s. This assumption is consistent with canopy methods as employed by, for instance, Belcher *et al*. [1], or the equivalent roughness method of Meneveau [15]. If the volume of air associated with a given turbine is $A_s D$, then a "turbine density", the number of turbines per unit volume, is:

$$\rho_T(y) = \begin{cases} 1/(A_s D) & \text{for } H - D/2 < y < H + D/2 \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

The thrust force per unit volume is:

$$f = \rho_T F_T = \frac{1}{2}\rho_T \rho \frac{\pi D^2}{4} u_h^2 C_T, \tag{2.8}$$

so that the term in the momentum equation is:

$$\frac{f}{\rho} = \frac{\pi \rho_T D^2 u_h^2 C_T}{8}. \tag{2.9}$$

The thrust force opposes the motion of the fluid, thus the above expression enters the momentum equation with a negative sign:

$$u\frac{du}{dx} + v\frac{du}{dy} = \frac{\mu}{\rho}\frac{d^2u}{dy^2} + 2\kappa^2 y\left(\frac{du}{dy}\right)^2 + 2\kappa^2 y^2 \frac{du}{dy}\frac{d^2u}{dy^2} - \frac{\pi \rho_T D^2 u_h^2 C_T}{8}. \tag{2.10}$$

The value of ρ_T is set to zero for $y < H - D/2$ and $y > H + D/2$, where H is the hub height.

For a given turbine layout, which in a large wind farm may consist of hundreds of turbines, it is not desirable to have to specify explicitly the surface area associated with each turbine. Even in a regular rectangular pattern, when the wind is not exactly aligned with the rows or columns, it will seldom intersect precisely the location of a turbine rotor (Figure 2). Therefore some approximation or interpolation is required in order to represent the effective thrust force along the line of analysis.[7]

[6] Approximating the rotor-average windspeed with the single-point windspeed at hub height is reasonable for the on-average flow quantities of relevance here. This is not a valid approximation when explicitly modelling a turbulent windfield.

[7] The "line of analysis" or "analysis line" refers to the orientation, relative to the pattern of turbines, of the xy plane in which a given two-dimensional boundary-layer analysis is performed. This line is by definition parallel with the wind direction. The x axis of the boundary-layer analysis corresponds to the analysis line.

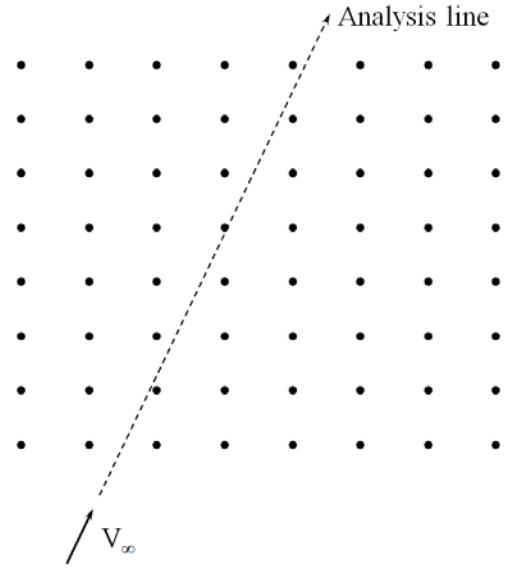


Figure 2: A regular turbine layout pattern with off-axis wind direction

Viper takes as input the effective turbine density ρ_T as a function of the x coordinate along the analysis line. For a regular case like Figure 2 the effective turbine density is constant and straightforward to compute. For irregular cases some assumptions must be made. It is outside the present scope to provide methods or guidance for the general case. It can be said, though, that for configurations where determining an effective turbine density is challenging, the applicability of a two-dimensional boundary-layer analysis should be questioned.

In the example of Figure 2 it is seen that the analysis line does not directly intersect many turbine locations. If adjacent turbines do not produce exactly the same thrust, then it is not immediately evident what should be the thrust associated with the analysis line. Explicit interpolation between turbines is problematic, since variations in the wind speed along the z direction (perpendicular to the plane of analysis) are not known upfront. And as stated previously, it is not desired to burden the user with explicitly specifying surface areas – or area "elements" – associated with each turbine.

The concept of turbine density is extended in order to provide a practical method for calculating the effective thrust force along the analysis line. Rather than assuming that the thrust associated with a given turbine is applied over a discrete region of space, the relationship between C_T and u_h is applied as a continuous function of x, $C_T(u_h(x))$. Then, at each coordinate x along the analysis line, the force term is evaluated with the value of C_T corresponding to the local value of $u_h(x)$. In Section 3, it is shown that this approach works well for regularly-spaced wind farms. (There is no validation data for large, irregularly-spaced wind farms.)

At present, to keep the input simple, only $C_T(u_h(x))$ is implemented in Viper; that is, all the turbines are assumed to be operating with the same thrust coefficient curve, a function of the local hub-height windspeed. It is also assumed that the turbine diameter is constant. In principle, though, the thrust force term could be:

$$\frac{\pi\ \rho_T(x)\ D(x)^2\ u_h(x)^2\ C_T\left(x,u_h(x)\right)}{8}$$

where $\rho_T(x)$, $D(x)$, and $C_T(x,u_h(x))$ are provided as input. Here $\rho_T(x)$ represents the variation in turbine spacing, and $D(x)$ and $C_T(x,u_h(x))$ could represent different turbine types or control strategies.

2.4 Numerical Solution

Equation 2.10 is solved by discretizing the domain (the xy plane) into rectangular elements, as shown in Figure 3. The corners of each element are nodes at which the state variables are defined.

2.4.1 Keller Box Method

The Keller Box method [12] is used to discretize the governing equations.[8] First the higher-order momentum equation is transformed into a first-order differential equation by introducing an intermediate state variable:

$$q \equiv \frac{du}{dy}, \tag{2.11}$$

such that:

$$u\frac{du}{dx} + vq = \frac{\mu}{\rho}\frac{dq}{dy} + 2\kappa^2 yq^2 + 2\kappa^2 y^2 q\frac{dq}{dy} - \frac{\pi\rho_T D^2 u_h^2 C_T}{8}. \tag{2.12}$$

Then Equations 2.2 and 2.12 are written at the center of each element (such as point A in Figure 3), while Equation 2.11 is written at the center of the right-hand (positive x) edge of each element (point B).

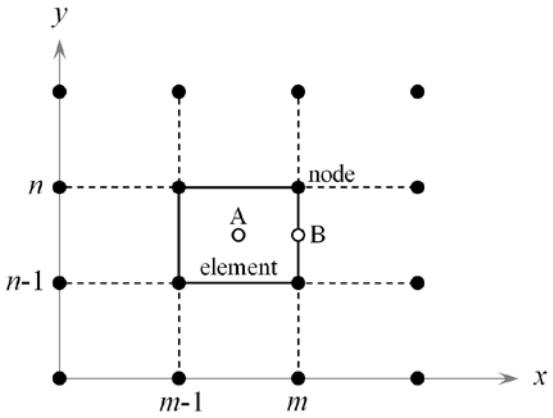


Figure 3: Discretization of the analysis domain into nodes and elements

In writing the differential equations at A and B, the best estimates of the quantities at these locations are used. This gives at point A:

$$\phi = \frac{1}{4}\left(\phi_{m-1,n-1} + \phi_{m-1,n} + \phi_{m,n-1} + \phi_{m,n}\right) \tag{2.13}$$

$$\frac{d\phi}{dx} = \frac{1}{2\Delta x}\left(-\phi_{m-1,n-1} - \phi_{m-1,n} + \phi_{m,n-1} + \phi_{m,n}\right) \tag{2.14}$$

$$\frac{d\phi}{dy} = \frac{1}{2\Delta y}\left(-\phi_{m-1,n-1} + \phi_{m-1,n} - \phi_{m,n-1} + \phi_{m,n}\right) \tag{2.15}$$

and:

$$y = \frac{1}{2}\left(y_{n-1} + y_n\right). \tag{2.16}$$

At point B:

$$\phi = \frac{1}{2}\left(\phi_{m,n-1} + \phi_{m,n}\right) \tag{2.17}$$

and:

[8] The equations are discretized in their original form, with velocities as the state variables. We do not introduce the stream function and transform the equations. Although employing the stream function is common historically, as Keller [11] notes, "there is no great advantage in introducing a stream function when employing the Box scheme." (p 422)

$$\frac{d\phi}{dy} = \frac{1}{\Delta y}\left(-\phi_{m,n-1} + \phi_{m,n}\right). \tag{2.18}$$

Here ϕ can represent *u*, *q*, or *v*.

2.4.2 Newton's Method

Beginning with initial conditions for *u*, *q*, and *v* on the upwind boundary, together with the boundary conditions (Equation 2.3), the solution advances forward step-by-step. Thus, during a given step, values in column $m-1$ are known, while those in column *m* are unknown. The discretized problem can be written in the form:

$$\vec{g}(\vec{u},\vec{q},\vec{v}) = 0\,, \tag{2.19}$$

or, combining $\vec{u}$, $\vec{q}$, and $\vec{v}$ into a single state vector $\vec{z}$, simply:

$$\vec{g}(\vec{z}) = 0\,. \tag{2.20}$$

If there are *N* nodes in a column, the vector $\vec{g}$ contains $3(N-1)$ equations: Equations 2.2, 2.11, and 2.12 written for each of the $N-1$ elements. There are also $3(N-1)$ unknowns: *N* values of each of *u*, *q*, and *v* at the nodes, minus the 3 boundary conditions.

Equation 2.20 is a nonlinear system of equations; and a good initial guess is available by taking the values from the $(m-1)^{\text{th}}$ column. Newton's method is well-suited to such a case. The vector of functions is expanded in a Taylor series about some "current" state $\vec{z}^{\,j}$:

$$\vec{g}(\vec{z}^{\,j} + \delta\vec{z}) = \vec{g}(\vec{z}^{\,j}) + \left[J\right]\delta\vec{z} + O(\delta\vec{z}^{\,2})\,, \tag{2.21}$$

with [*J*] the Jacobian matrix. Only first-order terms are retained. It is desired to find $\delta\vec{z}$ such that $\vec{g}(\vec{z}^{\,j} + \delta\vec{z}) = 0$. This can be found, using the first-order approximation of Equation 2.21, by solving the system of linear equations:

$$-\left[J\right]\delta\vec{z} = \vec{g}(\vec{z}^{\,j})\,. \tag{2.22}$$

In the basic implementation of Newton's method, after obtaining $\delta\vec{z}$, the estimate for $\vec{z}$ is updated as:

$$\vec{z}^{\,j+1} = \vec{z}^{\,j} + \delta\vec{z}\,, \tag{2.23}$$

where Equations 2.22 and 2.23 are iterated to convergence, and *j* is the iteration number. However, this is not always stable. Press *et al.* [17] suggest a means to stabilize Newton's method. Instead of Equation 2.23, the update formula is:

$$\vec{z}^{\,j+1} = \vec{z}^{\,j} + \lambda\,\delta\vec{z}\,, \tag{2.24}$$

with λ a scalar parameter, $0 < \lambda \le 1$. As the method converges, $\vec{g}^{\,j}$ goes to zero. (Here $\vec{g}^{\,j}$ is shorthand for $\vec{g}(\vec{z}^{\,j})$.) Thus $(\vec{g}^{\,j+1})^T\,\vec{g}^{\,j+1}$ represents a positive scalar that decreases to zero upon convergence. The vector $\delta\vec{z}$ is by definition a descent direction for $(\vec{g}^{\,j+1})^T\,\vec{g}^{\,j+1}$, for at least some infinitesimal step length. The convergence strategy is therefore to try Equation 2.23 (equivalent to $\lambda = 1$); check whether $(\vec{g}^{\,j+1})^T\,\vec{g}^{\,j+1} < (\vec{g}^{\,j})^T\,\vec{g}^{\,j}$; and, if not, employ Equation 2.24, progressively halving λ until $(\vec{g}^{\,j+1})^T\,\vec{g}^{\,j+1} < (\vec{g}^{\,j})^T\,\vec{g}^{\,j}$. This does not guarantee convergence, however it greatly increases the robustness of Newton's method.

2.4.3 Stabilizing Hourglass Modes

It was observed that for initial conditions that did not perfectly satisfy the boundary layer equations, the Keller Box method had a tendency to "hourglass": successive values of nodal variables oscillated along the *x* and *y* directions, such that the averaged quantities at the calculation points (A and B in Figure 3)

nonetheless satisfied the governing equations. To suppress hourglassing, the state variables $\vec{z}_m$ at column m are computed as:

$$\vec{z}_m = \vec{z}_{m-1} + \frac{1}{2}\left(\frac{\vec{z}\,'_m - \vec{z}_{m-1}}{\Delta x} + \frac{\vec{z}_{m-1} - \vec{z}_{m-2}}{\Delta x} \right) \Delta x = \vec{z}_{m-1} + \frac{1}{2}\left(\vec{z}\,'_m - \vec{z}_{m-2} \right), \tag{2.25}$$

where $\vec{z}\,'_m$ denotes the converged output from Newton's method, which could potentially include hourglassing. Equation 2.25 updates $\vec{z}_m$ by projecting from $\vec{z}_{m-1}$, using the average of the "reliable" slope between $m-2$ and $m-1$, and the "unreliable" (possibly hourglassing) slope between $m-1$ and m. As hourglassing is characterized by adjacent slopes alternating between large, nearly identical positive and negative values, taking the average of adjacent slopes when projecting the updated values of $\vec{z}_m$ effectively cancels any tendency to hourglass.

2.4.4 Discretized Governing Equations

In order to arrive at Equation 2.22, all variables $\phi_{m,n}$ and $\phi_{m,n-1}$ in Equations 2.13 through 2.18 are written as $\phi^j_{m,n} + \delta\phi_{m,n}$ and $\phi^j_{m,n-1} + \delta\phi_{m,n-1}$. Then terms of orders $\delta\phi^2$ and higher are dropped. The $\delta\phi$ terms are collected on the left-hand side, while the remaining terms are collected on the right-hand side. After much algebra, the momentum equation for each element is:

$$k_1\,\delta u_{m,n-1} + k_2\,\delta q_{m,n-1} + k_3\,\delta v_{m,n-1} + k_4\,\delta u_{m,n} + k_5\,\delta q_{m,n} + k_6\,\delta v_{m,n} + k_h\,\delta u_{m,h} = g^j \tag{2.26}$$

where :

$$k_1 = k_4 = \frac{1}{4\Delta x}\left(u^j_{m,n-1} + u^j_{m,n} \right) \tag{2.27}$$

$$k_2 = \frac{1}{16} v_{\text{sum}} + \frac{\mu}{2\rho\,\Delta y} - \frac{\kappa^2}{8} y_{\text{sum}} q_{\text{sum}} + \frac{\kappa^2}{8\Delta y} y^2_{\text{sum}} \left(q_{m-1,n-1} + q^j_{m,n-1} \right) \tag{2.28}$$

$$k_3 = k_6 = \frac{1}{16} q_{\text{sum}} \tag{2.29}$$

$$k_5 = \frac{1}{16} v_{\text{sum}} - \frac{\mu}{2\rho\,\Delta y} - \frac{\kappa^2}{8} y_{\text{sum}} q_{\text{sum}} - \frac{\kappa^2}{8\Delta y} y^2_{\text{sum}} \left(q_{m-1,n-1} + q^j_{m,n-1} \right) \tag{2.30}$$

and:

$$k_h = \frac{\pi \rho_T D^2 C_T}{16}\left(u_{m-1,h} + u^j_{m,h} \right), \tag{2.31}$$

with:

$$q_{\text{sum}} = q_{m-1,n-1} + q_{m-1,n} + q^j_{m,n-1} + q^j_{m,n} \tag{2.32}$$

$$v_{\text{sum}} = v_{m-1,n-1} + v_{m-1,n} + v^j_{m,n-1} + v^j_{m,n} \tag{2.33}$$

$$y_{\text{sum}} = y_{n-1} + y_n\,. \tag{2.34}$$

In Equation 2.31, the "h" subscript refers to the node nearest the hub-height of the turbine. The right-hand side of Equation 2.26 can be written:

$$g^j = -\frac{1}{8\Delta x} u_{\text{sum}} u_{\text{diff}} + \frac{\mu}{2\rho\,\Delta y} q_{\text{diff}} + \frac{\kappa^2}{16} y_{\text{sum}} q^2_{\text{sum}} - \frac{1}{16} v_{\text{sum}} q_{\text{sum}} \tag{2.35}$$

$$+\frac{\kappa^2}{16\Delta y}y_{\text{sum}}^2 q_{\text{sum}} q_{\text{diff}} + \frac{\pi\rho_T D^2 C_T}{32}\left(u_{m-1,h}+u_{m,h}^j\right)^2,$$

where:

$$u_{\text{sum}} = u_{m-1,n-1}+u_{m-1,n}+u_{m,n-1}^j+u_{m,n}^j \tag{2.36}$$

$$u_{\text{diff}} = -u_{m-1,n-1}-u_{m-1,n}+u_{m,n-1}^j+u_{m,n}^j \tag{2.37}$$

$$q_{\text{diff}} = -q_{m-1,n-1}+q_{m-1,n}-q_{m,n-1}^j+q_{m,n}^j \tag{2.38}$$

For simplicity, the thrust coefficient C_T is evaluated at the known value $u_{m-1,h}$, rather than the unknown value $u_{m,h}^j$.

The continuity equation for each element is:

$$\frac{1}{2\Delta x}\delta u_{m,n-1}-\frac{1}{2\Delta y}\delta v_{m,n-1}+\frac{1}{2\Delta x}\delta u_{m,n}+\frac{1}{2\Delta y}\delta v_{m,n}=-\frac{1}{2\Delta x}u_{\text{diff}}-\frac{1}{2\Delta y}v_{\text{diff}}, \tag{2.39}$$

where:

$$v_{\text{diff}} = -v_{m-1,n-1}+v_{m-1,n}-v_{m,n-1}^j+v_{m,n}^j. \tag{2.40}$$

The equation for q is:

$$\frac{1}{\Delta y}\delta u_{m,n-1}+\frac{1}{2}\delta q_{m,n-1}-\frac{1}{\Delta y}\delta u_{m,n}+\frac{1}{2}\delta q_{m,n}=\frac{1}{\Delta y}(u_{m,n}^j-u_{m,n-1}^j)-\frac{1}{2}(q_{m,n-1}^j+q_{m,n}^j). \tag{2.41}$$

Equations 2.26, 2.39, and 2.41, when written for each element, form the vector Equation 2.22.

2.5 Normalization

Given the very different x and y scales of boundary layer flow, the governing equations are normalized to improve the numerical conditioning of the solution. Lengths in x are normalized by L, the length of the computational domain; lengths in y are normalized by h_e, the height of the atmospheric boundary layer; and time is normalized by u_e/L. This results in the following normalized variables:

$$\hat{y}=\frac{y}{h},\quad \hat{x}=\frac{x}{L},\quad \hat{u}=\frac{u}{u_e},\quad \hat{v}=\frac{L}{hu_e}v. \tag{2.42}$$

Substituting into Equation 2.10 gives:

$$\hat{u}\frac{d\hat{u}}{d\hat{x}}+\hat{v}\frac{d\hat{u}}{d\hat{y}}=\frac{\mu L}{\rho u_e h^2}\frac{d^2\hat{u}}{d\hat{y}^2}+\frac{2\kappa^2\hat{y}L}{h}\left(\frac{d\hat{u}}{d\hat{y}}\right)^2+\frac{2\kappa^2\hat{y}^2L}{h}\frac{d\hat{u}}{d\hat{y}}\frac{d^2\hat{u}}{d\hat{y}^2}-\frac{\pi\rho_T D^2 L C_T}{8}\hat{u}_h^2. \tag{2.43}$$

Defining modified constants:

$$\hat{\kappa}=\kappa\sqrt{\frac{L}{h}},\quad \hat{\mu}=\frac{\mu L}{u_e h^2},\quad \hat{D}=D\sqrt{L}, \tag{2.44}$$

Equation 2.43 can be written:

$$\hat{u}\frac{d\hat{u}}{d\hat{x}}+\hat{v}\frac{d\hat{u}}{d\hat{y}}=\frac{\hat{\mu}}{\rho}\frac{d^2\hat{u}}{d\hat{y}^2}+2\hat{\kappa}^2\hat{y}\left(\frac{d\hat{u}}{d\hat{y}}\right)^2+2\hat{\kappa}^2\hat{y}^2\frac{d\hat{u}}{d\hat{y}}\frac{d^2\hat{u}}{d\hat{y}^2}-\frac{\pi\rho_T\hat{D}^2C_T}{8}\hat{u}_h^2 \tag{2.45}$$

which is the same form as Equation 2.10, to be solved by the procedures described in Section 2.4. The continuity equation and the definition of q also apply, without modification, for the nondimensional variables in Equation 2.42.

3 Validation

When the thrust coefficient C_T is set to zero, the methods of Section 2 are applicable to the case of a simple boundary layer over a flat plate. For laminar flow, this has an analytical solution, while for turbulent flow, there exists the log law, calibrated to experimental data. For full-scale validation including turbine thrust forces, windspeed data has been published for the Horns Rev and Nysted offshore wind farms.

3.1 Laminar Flow over a Flat Plate

Blausius obtained an analytical solution of Equations 2.1-2.3, for the case of laminar flow. (Laminar flow is obtained by setting κ to zero.) Schlichting [19] tabulates the solution in a form suitable for a spreadsheet calculation.

Figure 4 compares numerical (Viper) and analytical results for the development of a laminar boundary layer over a smooth, flat plate, beginning at 0.05 m from the leading edge. The remote flow velocity u_e is 1 m/s, and the fluid is air, with typical properties $\rho = 1.225$ kg/m^3 and $\mu = 1.789 \times 10^{-5}$ kg/ms. 100 steps were taken between x locations of 0.05 m and 1.05 m. 50 elements spanned the y direction of the computational domain, from the wall to a height h_e of 0.05 m. The elements were more densely packed near the wall, and more sparse towards the outer edge of the domain. Locations of the nodes are seen as black symbols in Figure 4.

The results demonstrate that the basic method of calculation is implemented correctly.

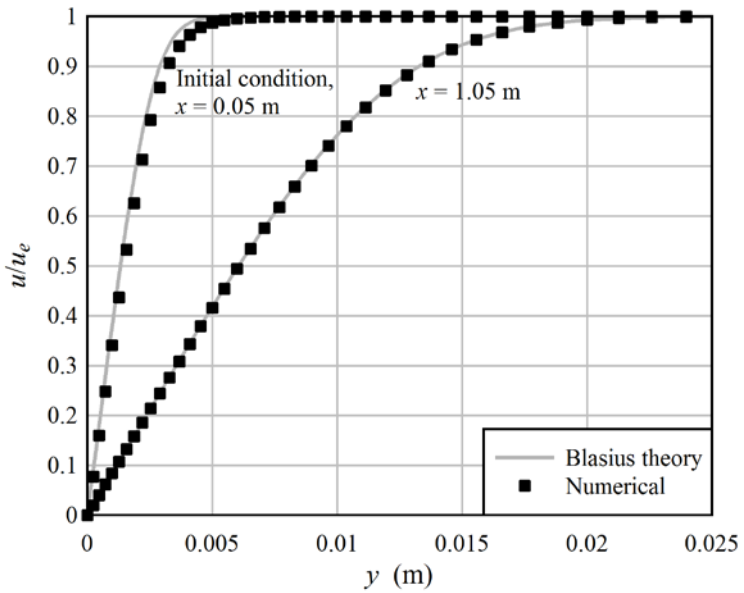


Figure 4: Development of a laminar boundary-layer over a flat plate

3.2 Turbulent Flow over a Flat Plate

The experimental measurements of a turbulent boundary layer over a flat plate are less precise than the theoretical solution for a laminar boundary layer. Theory indicates that a turbulent boundary layer should follow a logarithmic velocity profile for at least some fraction of the boundary layer height, deviating from this profile close to the wall, where the fluid viscosity becomes important. The shape and evolution of the logarithmic profile is calibrated to measurements. Schlichting [19] gives a summary of the data and recommends a set of parameters. Specifically, one computes the Reynolds number based upon the position x along the plate:

$$\mathrm{Re}_x = \frac{\rho u_e x}{\mu}, \tag{3.1}$$

the skin friction coefficient, here Schlichting's curve-fit to analytical results obtained by Prandtl:

$$c_f' = \left(2\log \mathrm{Re}_x - 0.65\right)^{-2.3} \tag{3.2}$$

the shear stress at the wall:

$$\tau_w = \frac{1}{2}\rho u_e^2 c_f' \tag{3.3}$$

the friction velocity:

$$v^* = \sqrt{\frac{\tau_w}{\rho}} \tag{3.4}$$

a nondimensional height:

$$\eta = \frac{v^* \rho}{\mu} y \tag{3.5}$$

and the fluid velocity u according to:

$$\frac{u}{v^*} = A_1 \log \eta + D_1 \tag{3.6}$$

where $A_1 = 5.85$ and $D_1 = 5.56$ are empirical constants.

As an alternative to a logarithmic profile, it is reasonable to assume a power law. The velocity profile is then computed as:

$$u = u_e \left(\frac{y}{\delta}\right)^{1/7}, \tag{3.7}$$

where the exponent 1/7 is empirical, as is an expression for the boundary-layer thickness δ:

$$\delta = 0.37 x \,\mathrm{Re}_x^{-1/5}. \tag{3.8}$$

Figure 5 shows results for the development of a turbulent boundary layer along a smooth, flat plate. The profiles were calibrated at a distance from the leading-edge x = 1 m. The profile was computed through x = 10 m, using a step size Δx of 0.1 m. The remote fluid velocity was 10 m/s, and the physical properties were those given in Section 3.1, typical of air. 100 elements spanned the height of the computational domain, which was set to 0.3 m.

The agreement between analytical and empirical results is not perfect, especially very near the wall, where the empirical formulas are not valid. The numerical and empirical velocity profiles agree closely over most of the height of the boundary layer (say, $0.02 \le y \le 0.2$ m).

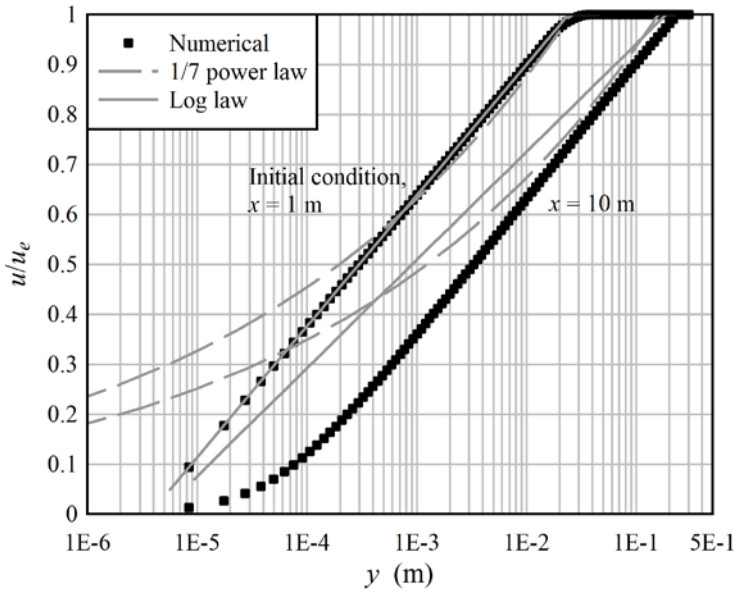


Figure 5: Development of a turbulent boundary layer over a flat plate

3.3 Flow at Atmospheric Scale

Figure 6 shows a fully-developed boundary layer at an atmospheric scale. The boundary layer is 1 km thick. It was initialized to a logarithmic velocity profile such that u_e = 10 m/s. The height of the first node, which to some extent represents the surface roughness length, was 1 mm. 101 nodes were evenly spaced on a logarithmic scale, between the wall and y = 1 km. The analysis was run from x = 0 to x = 10 km, with increments Δx of 100 m. As expected, the fully-developed profile did not change, except immediately adjacent to the wall, where the given initial conditions were not precise. The results indicate that the calculation is stable, with a negligible accumulation of numerical error over the computational domain.

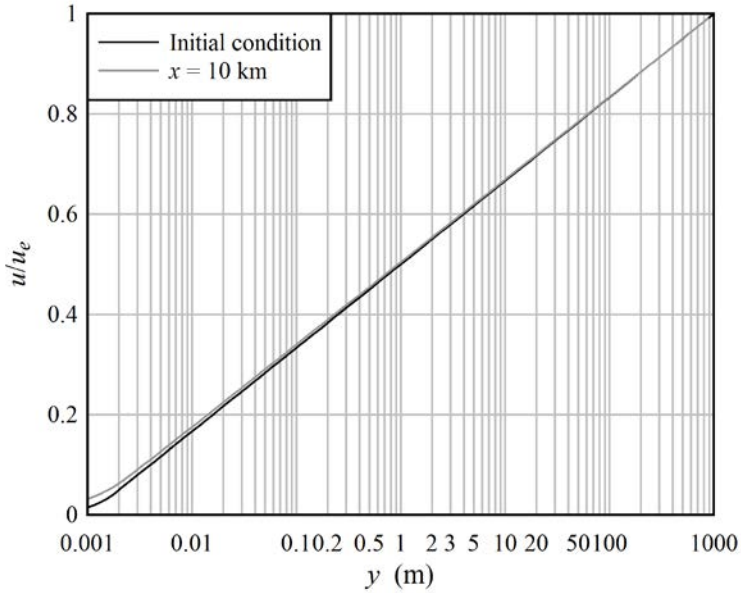


Figure 6: A fully-developed atmospheric boundary layer initialized to a logarithmic profile

3.4 Flow Through a Wind Farm

Frandsen *et al*. [8] summarize velocity deficit data collected in the wakes of the Horns Rev and Nysted wind farms. Hub-height windspeed data was collected at each turbine, as well as two meteorological masts placed several kilometers behind each of the wind farms. In both cases the mean remote windspeed was 8.5 m/s. The mean wind direction was aligned with the turbine rows, within a tolerance of $\pm 10^{o}$. It is appropriate to allow such a tolerance, as the present boundary-layer methods are not applicable for the case where the wind is aligned exactly along a row of turbines, and local wake effects become dominant.[9]

For the numerical calculations, the turbines were assigned a uniform C_T of 0.80, which is a representative value for operation below the rated windspeed. 150 elements spanned the height of the computational domain, which was set to 1 km; the nodes were spaced uniformly on a logarithmic scale, with the first node at a height of 1 mm. The step size Δx was 100 m (though the analysis was also run with Δx of 200 m, with identical results).

The initial conditions were specified as a logarithmic windspeed profile, like Figure 6. The remote windspeed u_e above the boundary layer was obtained by calibrating the slope of the logarithmic profile such that the hub-height windspeed, upstream of the wind farm, was equal to 8.5 m/s.

The upstream boundary of the wind farm was located a distance of 10 Δx from the upstream boundary of the computational domain, in order to eliminate any inaccuracy associated with imprecise initial conditions.

Figure 7 shows the hub-height windspeed as a function of position within and behind the Nysted wind farm. Figure 8 shows the same for the Horns Rev wind farm. The coordinate $x = 0$ has been defined as the position of the last turbine along the row. Note that the thrust force associated with the last turbine has been applied over a downstream distance equal to the separation between turbines. This is consistent with the assumption that the thrust force is distributed over a volume of air associated with each turbine.

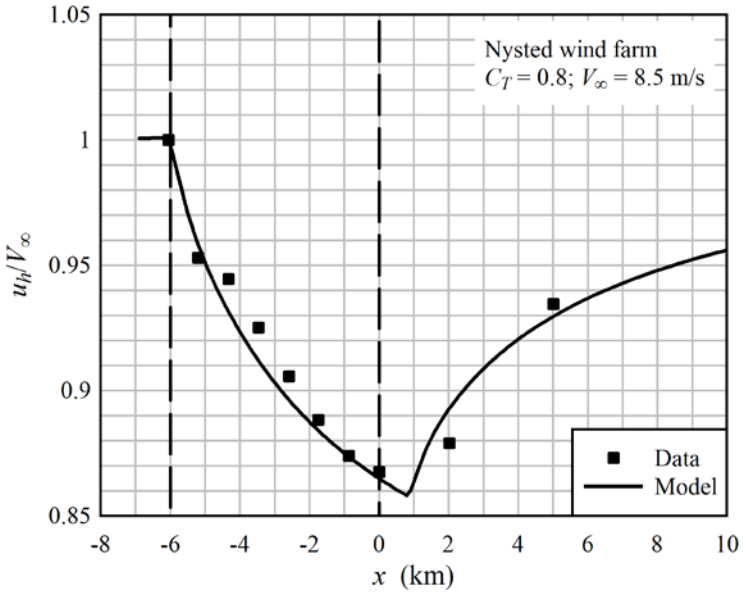


Figure 7: Hub-height windspeed through and behind the Nysted wind farm; wind direction aligned with turbine rows $\pm 10^{o}$; dashed lines indicate the coordinates of the first and last turbines in the farm

[9] The implication is that the bin width for the statistical distribution of wind direction should not be less than about 20^{o}, otherwise the power will be overpredicted for those bins where the wind is aligned directly along a row or column of turbines.

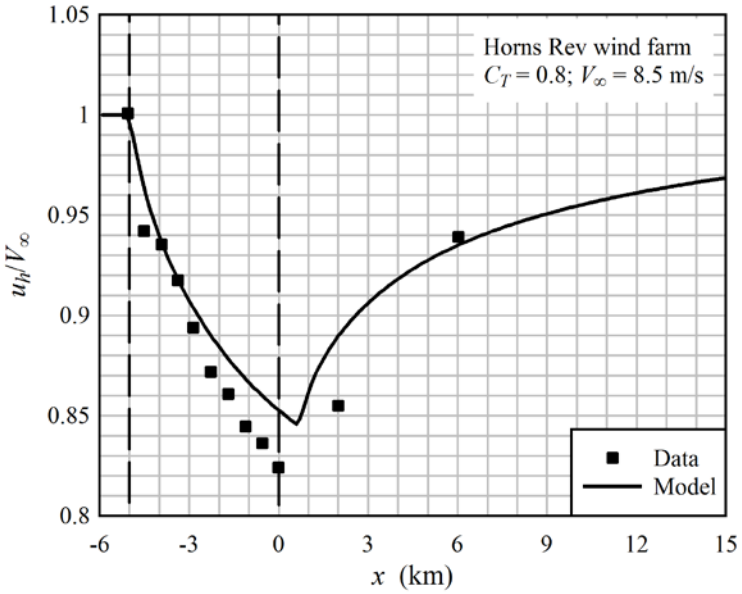


Figure 8: Hub-height windspeed through and behind the Horns Rev wind farm; wind direction aligned with turbine rows $\pm 10^o$

Distributing the rotor thrust force over a volume of air appears to result in underprediction of the velocity deficit for the second column of turbines; this is not surprising, since the local wake is not accounted for in the present model. Beyond the second column, deeper in the wind farm, the boundary-layer method provides a good estimate of the hub-height windspeed, although the windspeed deep in the Horns Rev park was overpredicted by 3%. There is not enough data to conclude whether the long, asymptotic return to the undisturbed windspeed, downstream of the wind farm, is realistic. Both Figure 7 and Figure 8 hint that the recovery may be somewhat faster than predicted; but in any case, the wake is seen to extend for many kilometers downstream.

The physics behind the hub-height velocity curves is elaborated in Figure 9, which shows the calculated velocity profiles, as a function of height *y*, at various locations through the Horns Rev wind farm. The coordinate x = -5 km is the undisturbed logarithmic boundary layer profile, while the coordinate x = 0 km is the downstream edge of the wind farm. Hub height is 70 m, and the rotor diameter is 80 m. The wake of the first turbine (second line from the right) extends only marginally beyond the diameter of the rotor. Deeper in the wind farm, the perturbation to the boundary-layer grows to a height much greater than that of the wind turbines; and it also begins to asymptotically approach an equilibrium condition. Downstream of the wind farm (not shown in Figure 9), where there are no turbine thrust forces, the process of boundary-layer development must reverse; but due to the large perturbation, a distance of many kilometers is required before the boundary layer asymptotically approaches its original state. Implicit in the velocity profiles is the turbulent mixing into the boundary-layer of high-velocity air from above.

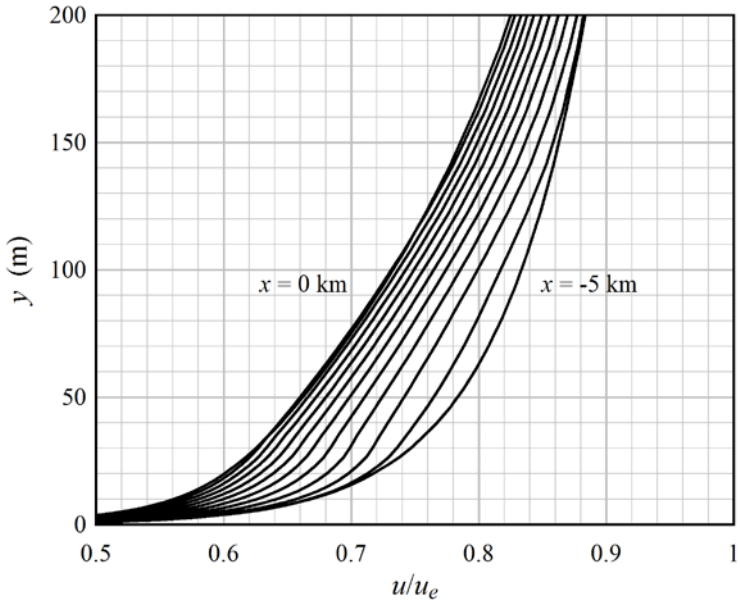


Figure 9: Velocity profiles through the Horns Rev wind farm, illustrating the development of the atmospheric boundary layer; u_e is the windspeed at a height of 1 km; hub height is 70 m, and the rotor diameter is 80 m

4 Annual Energy Production (AEP)

Hub-height velocity profiles such as those shown in Figure 7 and Figure 8 can be used to compute the power output of a given turbine along the line of analysis:

$$P = \frac{1}{2}\rho\frac{\pi D^2}{4}u_h^3 C_P \tag{4.1}$$

where $C_P(u_h)$ is specified as input.[10] For turbines not placed exactly along an analysis line, the hub-height windspeed is estimated by interpolating linearly from adjacent analysis lines. Note that the distribution of rotor thrust over a representative volume of air prevents an analysis line from "missing" the influence of a nearby turbine.

4.1 Wind Directions and Analysis Lines

Wind statistics at a given site are usually provided as a wind rose: essentially a table of the probability of a given combination of wind compass direction and speed. It is customary that the wind rose direction indicates where the wind is coming *from*, so 180° must be added to obtain the downwind direction. The compass is divided into a number of discrete wind directions. A Viper analysis along several lines is performed for each wind direction, and the hub-height windspeed at each turbine is obtained by interpolating between adjacent analysis lines.

Viper computes the evolution of the boundary layer along a two-dimensional *xy* plane. The *x* axis is aligned with the downwind direction, and the *y* axis points vertically upwards. Seen from above, the plane of

[10] As noted in Section 2.3 in the context of thrust force, it is feasible to specify $D(x_j)$ and $C_P(x_j,u_h)$, for the j^{th} turbine, in order to model different turbine types or wind farm control strategies. However, $C_T(x,u_h)$ is applied continuously over the domain whereas $C_P(x_j,u_h)$ is applied at the location of each turbine. The distribution of $C_T(x,u_h)$ in the farm should, of course, be consistent with the distribution of turbines at which $C_P(x_j,u_h)$ is evaluated.

a single boundary-layer analysis is a line through the pattern of turbines in a wind farm; Figure 2 is an example. The analysis must be performed along a sufficient number of such lines so that the hub-height windspeed at a given turbine location can be reliably interpolated. A requirement is that a line should be located at or just within each of the outer boundaries of the wind farm. An example of a particular farm layout and wind direction is shown in Figure 10. In the case labelled "not acceptable", the outermost lines are in the undisturbed flow, and thus will have a constant windspeed as a function of x^a. Interpolating to these outer lines will tend to overpredict the hub-height windspeed at the outer rows of turbines.

Figure 11 shows some suggested analysis line placements for a pair of representative cases. Note in particular how one line is located immediately at or inside each of the outer borders of the wind farm.

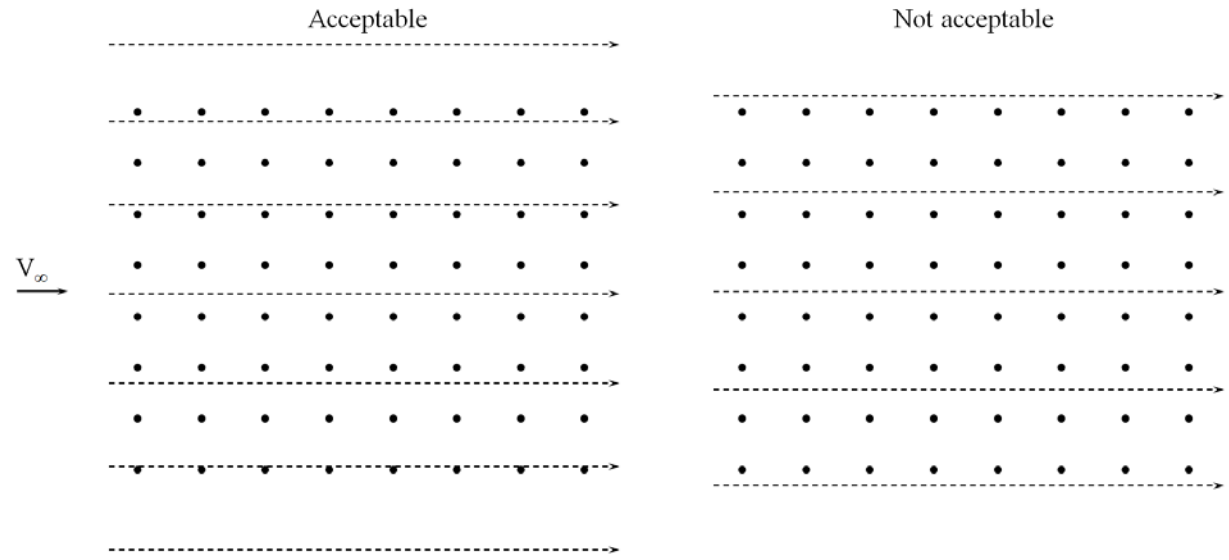


Figure 10: Examples of placement of analysis lines (dashed) in a wind farm, emphasizing that a line should be located immediately within each of the outer boundaries to avoid overpredicting the windspeed

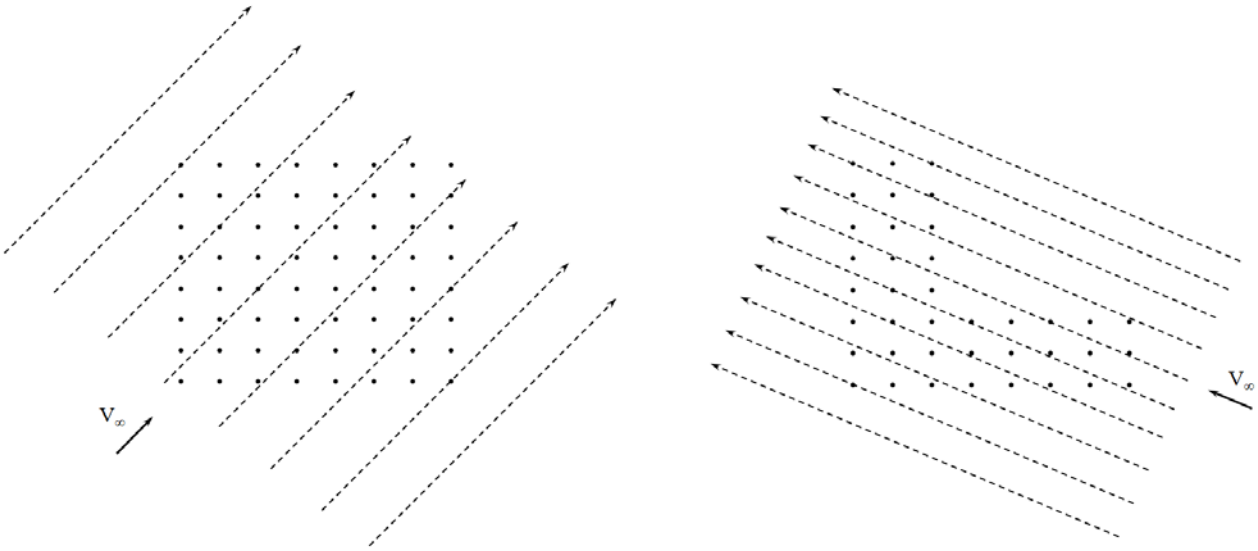


Figure 11: Some suggested analysis line placements to ensure reliable interpolation

4.2 Example of Dogger Bank

The Dogger Bank offshore wind zone will be used as an example for an AEP calculation. A wind rose for Dogger Bank is provided in a report by the Forewind Consortium [6]. The reference height for the wind measurements is not given; it is assumed that these are representative of the hub-height windspeed. Table I shows the probability data obtained from the wind rose. A good fit to this data is obtained with a probability density function described by a Weibull distribution:

$$p(V_\infty, \theta_c) = p_\theta \left(\frac{\beta}{\alpha(\theta_c)^\beta} \right) V_\infty^{\beta-1} \exp\left[-\left(\frac{V_\infty}{\alpha(\theta_c)} \right)^\beta \right] \tag{4.2}$$

where β is a constant 2.2, and p_θ and α are functions of the compass direction θ_c; each can be represented by a Fourier series:

$$A_0 + \sum_{k=1}^{N} A_k \sin(k\theta_c + B_k)\,. \tag{4.3}$$

The coefficients are listed in Table II, with all angles in radians.

Table I: Probability data (scatter diagram) obtained from a wind rose for the Dogger Bank Zone, assumed to represent the hub-height windspeed; the sum of all the entries in the table is 1

		Compass direction											
		0	30	60	90	120	150	180	210	240	270	300	330
V_∞	0-5	0.0131	0.0118	0.0111	0.0107	0.0107	0.0118	0.0123	0.0142	0.0155	0.0155	0.0146	0.0140
	5-10	0.0314	0.0236	0.0191	0.0203	0.0221	0.0263	0.0362	0.0414	0.0423	0.0389	0.0364	0.0359
	10-15	0.0175	0.0102	0.0124	0.0155	0.0170	0.0218	0.0314	0.0445	0.0457	0.0362	0.0259	0.0298
	15-20	0.0064	0.0022	0.0030	0.0048	0.0064	0.0062	0.0127	0.0245	0.0258	0.0187	0.0108	0.0109
	20-25	0.0004	0.0000	0.0000	0.0004	0.0004	0.0007	0.0032	0.0065	0.0074	0.0064	0.0025	0.0025

Table II: Coefficients on the Fourier series approximating the Dogger Bank wind probability density function

	p_θ	α
A_0	0.15915	11.0580
A_1	0.07945	-1.6950
A_2	-0.01503	0.2963
A_3	-0.01744	0.3500
A_4	0.00000	-0.1856
A_5	0.00000	-0.2093
B_1	-2.7086	0.5157
B_2	-3.0958	-0.8202
B_3	-0.5858	3.1416
B_4	0.0000	-0.3674
B_5	0.0000	0.5767

The compass is divided into a number of discrete wind directions; 12 is recommended, giving the directions listed in Table I. Likewise, the feasible range of windspeeds is divided into discrete bins; it is recommended to use 11 bins, centered at V_∞ = 5, 7, ..., 23, and 25 m/s. These resolutions on direction and windspeed are recommended based upon experience, representing a balance between accuracy and

computation time. For each wind direction and windspeed, a Viper analysis is run to obtain the hub-height windspeed at each turbine. Denoting the power output from the j^{th} of N_T turbines as P_j, the AEP associated with the m^{th} wind direction and n^{th} windspeed is:

$$\mathrm{AEP}_{m,n} = \sum_{j=1}^{N_T} p(V_{\infty,n}, \theta_m)\, \Delta V_\infty\, \Delta\theta\, P_{j,m,n}\, Y\,, \tag{4.4}$$

where ΔV_∞ and $\Delta\theta$ are the bin widths of the windspeed and compass directions, respectively, and Y is the number of seconds in a year (31,557,600 s on average, including leap years).

It is possible to shorten the required computation time by omitting repetetive analyses. In particular, it has been observed that above some windspeed like 15 m/s or 17 m/s, all the turbines will produce their rated power; in this case, a boundary-layer analysis is not necessary to estimate the energy production. Also, it is recommended to take advantage of symmetry in the wind farm layout, where possible.

Figure 12 shows example results for a uniformly-spaced 14-by-14 array of wind turbines. The rotor diameter is 180 m, and the spacing is 7D. The contours indicate annual energy production per turbine, in GWh. The ratio of production from the lowest- and highest-producing turbines is about 0.87. These values can also be compared against a production of 51.4 GWh from an isolated turbine, with no wind farm effects.

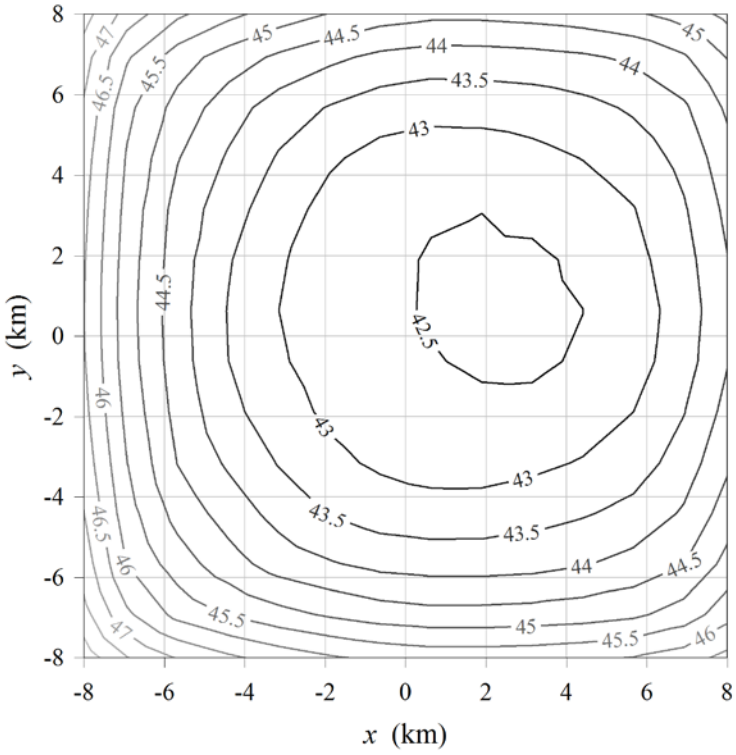


Figure 12: AEP per turbine for a square wind farm; 14-by-14 array of 180 m diameter turbines, uniform 7D spacing, Dogger Bank wind climate

4.3 Accelerated Tabular Analysis for Simple Homogeneous Layouts

The calculation of AEP can be accelerated greatly for simple, homogenous turbine layouts. Let the turbines in a wind farm be placed in a regular pattern, with a uniform, isotropic turbine spacing. Let all the turbines have the same diameter and $C_T(u_h)$ and $C_P(u_h)$ relationships. Let the layout pattern form a wind farm boundary that is a rectangle or other simple convex polygonal shape.[11] Then, as stated in Section 2.3, it

[11] Convexity ensures that no part of the farm lies in the shadow of a remote part of the farm, regardless of the wind direction; for instance, the shape of the farm cannot be an "L" if the present short-cut methods are to be applied.

is assumed that the thrust force from each turbine is spread throughout the characteristic volume of air associated with the turbine spacing. It follows that, for a given remote windspeed V_∞, the hub-height windspeed u_h at a given location is only a function of the depth in the farm (that is, the distance from the upwind boundary of the wind farm to the turbine, along the wind direction) and the turbine spacing.

A table can be generated by running a Viper analysis for each of several turbine spacings, and recording the hub-height windspeed as a function of position. Here the turbine density is constant over the computational domain, and the length of the computational domain is longer than the maximum breadth of the actual wind farm.

Once the table is generated, the hub-height windspeed for any turbine in the farm can be interpolated as a function of spacing and depth in the farm.

5 Description of Script Files

Here a brief overview of the script files is given, including the primary inputs and outputs. The main script is viper.m. Inputs and outputs are as follows, with all variables in SI units:

Input	Type	Description
fout	int	The file ID to which to write output. The file must first be opened as, for instance, `fout = fopen('out.txt','w')`
rho	int	Air density
mu0	int	Air viscosity
Hhub	real	Hub height
D	real	Rotor diameter
Nt	int	Number of turbines in the wind farm
xt	real: Nt × 2	Position, in farm coordinates, of each wind turbine
h	real	Height of the computational domain: 1500 m is recommended
y0	real	Surface roughness length: 0.001 m is recommended
Ny	int	Number of nodes spanning the y^a direction of the computational domain: 151 is recommended
ymin	real	Minimum element size near the wall (sea level): it is recommended to set this equal to y0
Nx	int	Number of steps to take in the x^a direction, such that Nx*dx is the length of the computational domain
dx	real	Step length in the x^a direction used in the boundary layer flow computation: 200 m is recommended
Nl	int: Nw	Number of analysis lines along which to perform a boundary layer flow computation
xl0_w	real: Nw × 2	For each wind direction, xl0_w specifies the origin (x^w,y^w), in wind coordinates, of the (x^a,y^a) coordinate system for the first analysis line
dyl	real: Nw	Spacing between analysis lines: this is specified separately for each wind direction, as the breadth of the wind farm may vary depending on the direction
Nw	int	Number of wind directions
thc	real: Nw	Wind directions for analysis, radians
dthw	real	Bin width of wind direction discretization, radians
Nv	int	Number of windspeeds
V0	real	Initial windspeed
dv	real	Increment in windspeed
vcpct	real: 25 × 2	A table listing the C_P and C_T coefficients of the turbine at windspeeds of 2, 3, ..., 25, and 26 m/s.

Output	Dimensions	Description
AEPt	real: Nt	Annual energy production of each turbine

The script viper.m calls several other functions, as follows:

Function	Description
ymesh.m	Defines the y^a coordinates of the nodes spanning the height of the computational domain; these are evenly spaced on a logarithmic scale
windPDF.m	This implements Equations 4.2 and 4.3
turbulentBL.m	Specifies an initial logarithmic velocity profile to initialize the boundary layer analysis
turbineDensity.m	A user-defined function returning the value of the turbine areal density ($\rho_T D = 1/A_s$) as a function of position in farm coordinates
runBL.m	Executes the boundary layer analysis along a single analysis line
interpVCpCt.m	Interpolates the table vcpct
NSNewton.m	Applies the Newton-Raphson method to converge on the velocity field such that the governing equations are satisfied
NSfunction.m	Formulates the linearized boundary-layer equations, returning $-[J]$ and $\vec{g}$ from Equation 2.22

6 Conclusions

A set of Matlab scripts, named Viper, has been developed for computing the annual energy output of a wind farm. The analysis method is based on a numerical finite-difference solution of the two-dimensional boundary-layer equations. The results have been validated based on empirical formulas for turbulent flow over a flat plate, and full-scale measurements at the Horns Rev and Nysted wind farms. Although validation data is lacking, on the same theoretical basis Viper can be used to predict the performance of irregular wind farms, for instance with variable turbine diameters, tower heights, rated powers, and spacings; or wind farms consisting of several blocks of turbines, with separation between the blocks.

The method employed by Viper has limitations, most importantly that it is based upon two-dimensional boundary-layer analysis. A three-dimensional analysis of the flow through a wind farm would be more accurate, although more computationally intensive.

It is possible that the Viper analysis could be accelerated significantly by linearizing the boundary-layer equations, in the manner of Belcher *et al.* [1]. Linearization implies some loss of accuracy, and if implemented should be validated against nonlinear calculations.

A Viper analysis can be greatly accelerated for regular layouts, either by taking advantage of symmetry or by generating tables of hub-height windspeed as a function of depth in the farm.

References

[1] Belcher S.E., *et al.*; Adjustment of a turbulent boundary layer to a canopy of roughness elements; Journal of Fluid Mechanics 488 (2003) 369-398

[2] Calaf M., *et al.*; Large eddy simulation study of fully developed wind-turbine array boundary layers; Physics of Fluids 22 (2010) 015110 1-16

[3] Chowdhury S., *et al.*; Optimizing the arrangement and the selection of turbines for wind farms subject to varying wind conditions; Renewable Energy 52 (2013) 273-282

[4] Christiansen M.B., Hasager C.B.; Wake effects of large offshore wind farms identified from satellite SAR; Remote Sensing of Environment 98 (2005) 251-268

[5] Davidson P.A.; *Turbulence – An Introduction for Scientists and Engineers*; Oxford University Press, 2004

[6] Forewind Consortium; *Environmental Statement Chapter 5: Project Description*; Report F-OFC-CH-005, Forewind Consortium, 2013
[7] Frandsen S., *et al*.; Analytical modelling of wind speed deficit in large offshore wind farms; Wind Energy 9 (2006) 39-53
[8] Frandsen S., *et al*.; Summary report: The shadow effect of large wind farms: measurements, data analysis, and modelling; Report Risø-R-1615(EN), Risø National Laboratory (Denmark), 2007
[9] Frandsen S., *et al*.; The making of a second-generation wind farm efficiency model complex; Wind Energy 12 (2009) 445-458
[10] Jensen N.O.; A Note on Wind Generator Interaction; Report Risø-M-2411, Risø National Laboratory (Denmark), 1983
[11] Keller H.B.; Numerical methods in boundary-layer theory; Annual Review of Fluid Mechanics 10 (1978) 417-433
[12] Keller H.B., Cebeci T.; Accurate numerical methods for boundary-layer flows. II: Two-dimensional turbulent flows; AIAA Journal 10 (1972) 1193-1199
[13] Kusiak A., Song Z.; Design of wind farm layout for maximum wind energy capture; Renewable Energy 35 (2010) 685-694
[14] Lackner M.A., Elkinton C.N.; An analytical framework for offshore wind farm layout optimization; Wind Engineering 31 (2007) 17-31
[15] Meneveau C.; The top-down model of wind farm boundary layers and its applications; Journal of Turbulence 13 (2012) 1-12
[16] Milborrow D.J.; The performance of arrays of wind turbines; Journal of Industrial Aerodynamics 5 (1980) 403-430
[17] Press W.H., *et al*.; *Numerical Recipes – The Art of Scientific Computing*; Third Edition, Cambridge University Press, 2007
[18] Réthoré P.-E., *et al*.; A CFD model of the wake of an offshore wind farm: using a prescribed wake inflow; Journal of Physics: Conference Series 75 (2007) 012047 1-7
[19] Schlichting H.; *Boundary-Layer Theory*; Sixth Edition, McGraw-Hill, 1968
[20] Templin R.J.; An estimate of the interaction of windmills in widespread arrays; National Aeronautical Establishment Laboratory Report LTR-LA-171, National Research Council of Canada, 1974
[21] Turner S.D.O., *et al*.; A new mathematical programming approach to optimize wind farm layouts; Renewable Energy 63 (2014) 674-680

Technology for a better society

www.sintef.no