

# Fast Multilevel Panel Method for Wind Turbine Rotor Flow Simulations

A. van Garrel<sup>\*</sup>, C.H. Venner<sup>†</sup>, H.W.M. Hoeijmakers<sup>‡</sup>

*University of Twente, Enschede, 7500AE, The Netherlands*

A fast multilevel integral transform method has been developed that enables the rapid analysis of unsteady inviscid flows around wind turbine rotors. A low order panel method is used and the new multi-level multi-integration cluster (MLMIC) method reduces the computational complexity for calculating the wake deformation downstream of the wind turbine rotor from  $\mathcal{O}(N^2)$  for a conventional approach to  $\mathcal{O}(N)$ . The method discretizes the volume surrounding the configuration with cubes. Each cube contains a grid of nodes that are used in the interpolation of the Green's functions underlying the panel method.

The formulation of the panel method is described concisely and verified using exact solutions for a tri-axial ellipsoid in uniform flow and for a rotating ellipsoid in air at rest. For these tests the panel method exhibits an error varying quadratically with panel size.

The MLMIC fast multilevel method is described and its accuracy and  $\mathcal{O}(N)$  computational speed are verified for some model problems. Surface pressure distributions obtained with the fast panel method are compared with results from the MEXICO wind tunnel experiment and with results from a state-of-the-art numerical simulation method based on the Reynolds-averaged Navier-Stokes (RaNS) equations.

This work repositions panel methods in the computational landscape as valuable intermediate fidelity computational design method for wind turbine engineering.

## Nomenclature

$L$	Length	$\vec{F}, \vec{M}$	Force, moment vectors
$h$	Box size	$\vec{x}, \vec{y}$	Point coordinates
$p$	Pressure, polynomial order	$\vec{u}$	Velocity vector
$\vec{x}, \vec{y}$	Point coordinates	$\vec{n}$	Normal vector of unit length
$\vec{r}$	Distance vector ( $= \vec{x} - \vec{y}$ )	$\Phi$	Velocity potential
$r$	Distance ( $=  \vec{r} $ )	$\varphi$	Velocity perturbation potential
$\rho$	Mass density	$a, b, c$	Ellipsoid semi-axis dimensions
$C_p$	Pressure coefficient	$\sigma, \mu$	Source, dipole strengths
$\vec{g}$	Gravitational acceleration vector		
<i>Subscripts</i>			
$i, j$	Index numbers	ref	Reference value
$k, m$	Fictitious and real flow domains	rel	In moving frame of reference
$\infty$	Onset flow	b, w	Body, wake
s	Surface		

---

<sup>\*</sup>PhD. Student, Faculty of Engineering Technology, Engineering Fluid Dynamics, P.O. Box 217, a.vangarrel@utwente.nl.

<sup>†</sup>Professor, Faculty of Engineering Technology, Engineering Fluid Dynamics, P.O. Box 217, c.h.venner@utwente.nl.

<sup>‡</sup>Professor, Faculty of Engineering Technology, Engineering Fluid Dynamics, P.O. Box 217, h.w.m.hoeijmakers@utwente.nl, AIAA Senior Member.

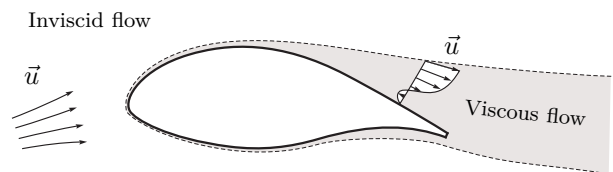
# I. Introduction

## A. Background

The ongoing trend in wind turbine development is towards larger rotors due to the resulting lower cost of energy. These large rotors lead to relatively flexible structures that are more susceptible to the unsteady aerodynamic loading occurring in normal operating conditions. An accurate prediction of these loadings is important for the design of an economically viable and technically reliable wind turbine.

The challenge for wind turbine applications is to ultimately integrate a mix of simulation methods for unsteady aerodynamics, structural dynamics, and control strategies for use in a wind turbine engineering environment with balanced problem turnaround times and simulation accuracy. The success of flow solvers based on the Navier-Stokes equations, often using large computer systems (many cores), obscures that for day-to-day practical use in a wind turbine engineering environment there is still a need for a method with higher fidelity than the blade-element-momentum methods currently in use in such a mix of simulation methods while maintaining reasonable computational demands and short problem turnaround times on relatively small scale computer systems.

A possible answer to this challenge is the Viscous-Inviscid Interaction (VII) technology that was started in the 1970's (see Lock and Williams<sup>21</sup>) when inviscid flow methods were coupled in strong interaction with a method solving the boundary layer equations. The latter to account for the effects of viscosity, specifically in the boundary layers (see Figure 1). The numerical simulation methods for the two regions are coupled strongly so that the simulation of mildly separated flows is feasible. This



**Figure 1. Domain decomposition into an inviscid flow region and a viscous flow region.**

research was almost completely abandoned once practically applicable RaNS based computational methods started to appear. Drela<sup>10</sup> was one of the first to implement successfully a computational method that coupled a panel method for the inviscid external flow with an Integral Boundary Layer (IBL) method for two-dimensional airfoils. His XFOIL<sup>9</sup> computer program is nowadays open source, freely available, and the de-facto industry standard.

A fully three dimensional and unsteady flow approach was initiated at ECN in 2004 by Van Garrel<sup>13</sup> targeting wind turbine rotor aerodynamics. For the inviscid external flow a low order panel method was selected<sup>14</sup>. The development of an unsteady 3D IBL method was started by Özdemir<sup>25</sup> and the strong interaction between the two computational methods was investigated by Bijleveld<sup>3</sup>. At several other places similar research is taking place, the most notable is the work of Drela<sup>11</sup>.

## B. Present Work

For wind turbine applications the dynamic character of the deformation and roll-up of the wake has to be taken into account in all unsteady flow situations. A substantial drawback, however, is the computational effort that grows quadratically with problem size  $N$  when a conventional panel method implementation for the rolling-up of the wake is used. The number of panels  $N$  in the wake easily reaches hundreds of thousands. To render such numerical simulations practical the current research focuses therefore on reducing the  $\mathcal{O}(N^2)$  computer run times to  $\mathcal{O}(N)$  through the development of a fast multilevel integral evaluation scheme capable of handling highly distorted wake geometries.

The current work builds upon the multilevel multi-integration concept of Brandt and Lubrecht<sup>7</sup> and by Venner and Lubrecht<sup>30</sup>. We adapt it so that it is also practically applicable to irregular wind turbine wake surfaces. The new approach is referred to as the MLMIC (Multi-Level Multi-Integration Cluster) method. The MLMIC method reduces the work for the simulation of wind turbine rotor wakes to  $\mathcal{O}(N)$ . For details of this approach the reader is referred to the thesis work<sup>15</sup> of the first author.

## II. Low Order Panel Method

### A. Background

#### 1. Mathematical Formulation

We consider the fluid dynamics equations for wind turbine applications and assume the flow to be unsteady, inviscid, and incompressible. Fluid mass density is considered constant. With these assumptions the equations for conservation of mass and momentum reduce to

$$\nabla \cdot \vec{u} = 0. \quad (1)$$

and

$$\rho \frac{\partial \vec{u}}{\partial t} + \rho(\vec{u} \cdot \nabla) \vec{u} + \nabla(p - \rho \vec{g} \cdot \vec{x}) = \vec{0}. \quad (2)$$

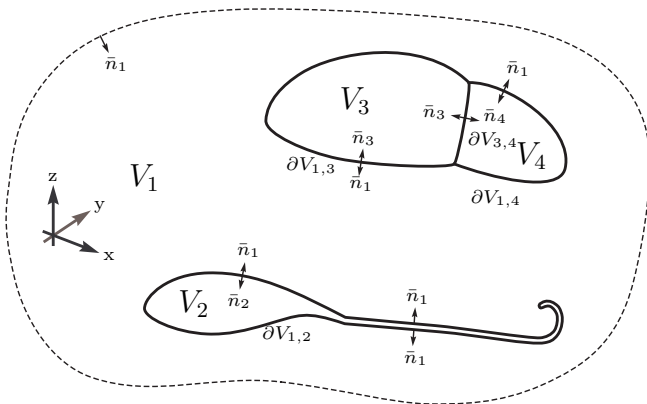
A significant reduction in complexity can be achieved when it is assumed that rotational flow is confined to infinitesimal thin boundary layers and wakes, and is irrotational everywhere else, that is  $\nabla \times \vec{u} = \vec{0}$ . This allows us to write the velocity vector field  $\vec{u}(\vec{x}, t)$  as the gradient of a scalar velocity potential function  $\Phi(\vec{x}, t)$ , that is  $\vec{u} = \nabla \Phi$ . Substitution of this expression for the velocity in the continuity equation (1) gives the Laplace equation for the velocity potential in domain  $V$ :

$$\nabla \cdot \nabla \Phi = 0. \quad (3)$$

Substituting this velocity potential gradient in the momentum conservation equations (2) results in the Bernoulli equation for unsteady potential flow

$$\frac{\partial \Phi}{\partial t} + \frac{1}{2} \nabla \Phi \cdot \nabla \Phi + \frac{p}{\rho} - \vec{g} \cdot \vec{x} = C(t), \quad (4)$$

in the whole domain. We will solve the Laplace equation with a low order panel method (see Hoeijmakers<sup>16</sup>, Katz and Plotkin<sup>17</sup>) in which boundary conditions are enforced at a finite set of collocation points. We decompose domain  $V$  into a set of non-overlapping volumes  $V_m$  with boundaries  $\partial V_m$  as depicted in Figure 2 and introduce the notation  $S_{m,k}$  for the part of the boundary that the two volumes  $V_m$  and  $V_k$  have in common. When approaching  $S_{m,k}$  from volume  $V_m$ , that side of the surface is indicated by  $S_{m\cdot}$  and likewise  $S_{\cdot k}$  for the other side.



**Figure 2.** The flow domain  $V \in \mathbb{R}^3$  composed of non-overlapping volumes  $V$  and inner boundaries  $\partial V_{m,k}$  that separate volume  $V_m$  from volume  $V_k$ . Unit normal vector  $\vec{n}_m$  is defined to point into volume  $V_m$ .

ity potential contributions  $\varphi_\mu(\vec{x}, t)$  and  $\varphi_\sigma(\vec{x}, t)$  by integrals involving dipole singularity distributions  $\mu(\vec{y}, t)$  and source singularity distributions  $\sigma(\vec{y}, t)$ , respectively, on the boundaries  $S_{m,k}$ , that is

$$\Phi = \Phi_\infty + \varphi_\mu + \varphi_\sigma. \quad (5)$$

We will use the formulation utilizing the internal Dirichlet boundary condition for the solution of the Laplace equation (3) as was introduced by Morino and Kuo<sup>24</sup> and later popularized by Maskew<sup>22</sup>. In this formulation it is assumed that only the solution in the volume on one side of the surface is of interest. Let  $S_{m\cdot}$  denote the side of surface  $S_{m,k}$  where we want to obtain a solution of the flow problem, and let  $S_{\cdot k}$  denote the side that is considered non-physical domain and contains a fictitious flow. Boundary conditions will be enforced at collocation points at the fictitious flow side of the surface.

The velocity potential function in the Laplace equation (3) can be expressed for a point  $\vec{x}$  in volume  $V$  in terms of a reference onset velocity potential  $\Phi_\infty(\vec{x}, t)$  and perturbation velocity potential  $\Phi_\mu(\vec{x}, t)$  and perturbation velocity potential  $\Phi_\sigma(\vec{x}, t)$ .

The perturbation velocity potentials induced at point  $\vec{x}$  by the dipole and source distributions on all surfaces  $S_{m,k}$  are defined by

$$\varphi_\mu(\vec{x}, t) = \frac{-1}{4\pi} \iint_S \mu \frac{\bar{n}_m \cdot \vec{r}}{r^3} dS, \quad (6)$$

$$\varphi_\sigma(\vec{x}, t) = \frac{-1}{4\pi} \iint_S \sigma \frac{1}{r} dS, \quad (7)$$

where  $\bar{n}_m(\vec{y}, t)$  is the unit normal vector in  $\vec{y} \in S_{m,k}$  that is pointing into volume  $V_m$ . The distance vector  $\vec{r}$  is defined as the vector from a point  $\vec{y}$  on the surface to evaluation point  $\vec{x}$ , whereas the distance between the two points is denoted by  $r$ , that is,

$$\vec{r} = \vec{x} - \vec{y}, \quad r = |\vec{r}|, \quad \text{for } \vec{y} \in S_{m,k}. \quad (8)$$

The dipole strength  $\mu(\vec{y}, t)$  and the source strength  $\sigma(\vec{y}, t)$  at point  $\vec{y}$  at the surface, are related to the velocity potential values  $\Phi_m(\vec{y}, t)$  and  $\Phi_k(\vec{y}, t)$  on both sides of the surface by

$$\mu(\vec{y}, t) = -(\Phi_m - \Phi_k), \quad (9)$$

$$\sigma(\vec{y}, t) = \nabla(\Phi_m - \Phi_k) \cdot \bar{n}_m. \quad (10)$$

The boundary condition at surface side  $S_{m\cdot}$  is such that at the surface the flow velocity in normal direction is equal to normal component of the surface velocity  $\vec{u}_s(\vec{x}, t)$ , plus a specified outflow velocity  $v_n(\vec{x}, t)$  relative to the moving boundary in normal direction:

$$\nabla\Phi_m \cdot \bar{n}_m = \vec{u}_s \cdot \bar{n}_m + v_n, \quad \vec{x} \rightarrow S_{m\cdot}. \quad (11)$$

The distribution of the outflow velocity  $v_n$  can be used for example to simulate the effects of the boundary layer displacement thickness or for the simulation of inflow through a suction slot. The local surface velocity  $\vec{u}_s$  may be composed of solid body rotation, surface translation, surface rate of deformation and so on.

Suppose the velocity potential  $\Phi_k$  and its gradient  $\vec{u}_k(\vec{x}, t) = \nabla\Phi_k$  in the fictitious flow domains  $V_k$  are known in advance. For point  $\vec{x} \in V_k$  the boundary integral equation (5) reads

$$\varphi_\mu(\vec{x}, t) + \varphi_\sigma(\vec{x}, t) = \Phi_k(\vec{x}, t) - \Phi_\infty(\vec{x}, t), \quad \vec{x} \in V_k. \quad (12)$$

In the limit of point  $\vec{x}$  approaching the surface  $S_{\cdot k}$  at point  $\vec{y}$ , equation (5) is expressed in terms of Principal Value and Finite Part integrals as

$$\frac{1}{2}\mu + \varphi_\mu^p(\vec{x}, t) + \varphi_\sigma^p(\vec{x}, t) = \Phi_k(\vec{x}, t) - \Phi_\infty(\vec{x}, t), \quad \vec{x} \rightarrow \vec{y} \in S_{\cdot k}. \quad (13)$$

Substitution of the boundary condition (11) and the known velocity potential  $\Phi_k$  in the definition of the source strength (10), gives an expression for the source strength in terms of known quantities:

$$\sigma(\vec{y}, t) = (\vec{u}_s - \vec{u}_k) \cdot \bar{n}_m + v_n, \quad \vec{y} \in S_{m,k}. \quad (14)$$

The boundary integral equation (12) at  $\vec{x} \rightarrow S_{\cdot k}$  now gives an expression involving the unknown dipole strength  $\mu(\vec{y}, t)$  as a function of known quantities. At the surface side  $S_{m\cdot}$  of interest the surface gradient of the dipole strength (9) gives us the tangential component of the velocity as

$$\nabla_S\Phi_m(\vec{x}, t) = \nabla_S\Phi_k - \nabla_S\mu, \quad \vec{x} \rightarrow S_{m\cdot}. \quad (15)$$

The surface gradient of the velocity potential  $\Phi_k$  is the velocity vector component of  $\vec{u}_k$  tangential to the surface and can be written as

$$\nabla_S\Phi_k = \bar{n}_m \times (\vec{u}_k \times \bar{n}_m). \quad (16)$$

Combining the normal velocity from boundary condition (11), the expression for the source strength (14), and the tangential velocity (15) gives an expression for the velocity at the surface, in the inertial coordinate system, as

$$\nabla\Phi_m(\vec{x}, t) = (\vec{u}_s \cdot \bar{n}_m + v_n)\bar{n}_m + \nabla_S\Phi_k - \nabla_S\mu, \quad \vec{x} \rightarrow S_{m\cdot}, \quad (17)$$

which can also be written as

$$\vec{u}_m(\vec{x}, t) = \vec{u}_k + \sigma \vec{n}_m - \nabla_s \mu, \quad \vec{x} \rightarrow S_m. . \quad (18)$$

Equation (18) states that the velocity at the side of the surface of interest is composed of a known base flow field  $\vec{u}_k$  and a perturbation flow field due to the source and dipole singularity distributions. For an observer at point  $\vec{x} \rightarrow S_m. .$  moving with the local surface velocity  $\vec{u}_s$ , the relative velocity that is experienced will be

$$\vec{u}_{\text{rel}} = \vec{u}_m - \vec{u}_s. \quad (19)$$

Still, the velocity potential field  $\Phi_k$ , and consequently velocity field  $\vec{u}_k$ , has to be defined in the fictitious domains. We follow Morino and Kuo<sup>24</sup> and Maskew<sup>22</sup> and set the fictitious flow field equal to the onset flow field, that is  $\Phi_k = \Phi_\infty$ , and  $\vec{u}_k = \vec{u}_\infty$ . The definition of the dipole strength (9) gives  $\varphi_m = -\mu$  for the perturbation potential at the exterior side of the surface.

At the trailing edge of a lifting body, the point where the flow leaves the surface, wake surfaces are explicitly added in the potential flow model. A Kutta condition will be imposed at the trailing edge so that a smooth flow with finite velocity at that point results. Here, the Kutta condition that was introduced by Morino and Kuo<sup>24</sup> is used. The condition equates the dipole strength at the first point of the wake to the jump in dipole strengths across the trailing edge:

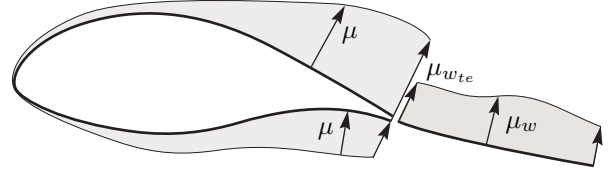


Figure 3. Trailing edge Kutta condition.

$$\mu_{w_{te}} = [[\varphi]]_{te} = [[\mu]]_{te}. \quad (20)$$

For the evolution of the geometry of the wake we will make use of the theorems of Helmholtz and Kelvin for vorticity dynamics (see Batchelor<sup>1</sup>, Cottet and Koumoutsakos<sup>8</sup>, Saffman<sup>27</sup>) that state that in inviscid incompressible flow the vorticity field moves with the fluid and its strength remains constant. In terms of the evolution of wake element position  $\vec{x}_w$  and wake element dipole strength  $\mu_w$  the corresponding equations are

$$\frac{d\vec{x}_w}{dt} = \vec{u}, \quad \vec{x}_w(t_0) = \vec{x}_{te}(t_0), \quad (21)$$

and the material derivative of the wake dipole strength

$$\frac{D\mu_w}{Dt} = 0, \quad \mu_w(t_0) = \mu_{w_{te}}(t_0), \quad (22)$$

where  $t_0$  is the time the creation of the of wake element.

## 2. Pressure, Forces, and Moments

An expression for the pressure  $p$  can be obtained from the unsteady Bernoulli equation (4) by relating upstream flow quantities with perturbed local quantities. Let  $p_\infty$  be the undisturbed upstream pressure, and let  $\Phi_\infty(\vec{x}, t)$  be the upstream total velocity potential, so that the unperturbed onset velocity is  $\vec{u}_\infty = \nabla\Phi_\infty$ . Likewise the perturbed local quantities are pressure  $p$ , total velocity potential  $\Phi = \Phi_\infty + \varphi_m$ , and local flow velocity  $\vec{u}_m = \nabla\Phi_\infty + \nabla\varphi_m$ . Substituted in the Bernoulli equation (4) this results in

$$\frac{p - p_\infty}{\frac{1}{2}\rho_\infty} = \vec{u}_\infty \cdot \vec{u}_\infty - \vec{u}_m \cdot \vec{u}_m - 2\frac{\partial\varphi_m}{\partial t} + 2\vec{g} \cdot (\vec{x} - \vec{x}_{\text{ref}}), \quad (23)$$

where  $\vec{x}_{\text{ref}}$  is defined as the point where the pressure  $p$  is equal to the undisturbed upstream pressure  $p_\infty$  when the fluid is at rest.

For the definition of a pressure coefficient we first introduce the material derivative of the perturbation potential  $\varphi_m$  at a point on the surface moving with velocity  $\vec{u}_s$ :

$$\begin{aligned} \frac{D_s\varphi_m}{Dt} &= \frac{\partial\varphi_m}{\partial t} + \vec{u}_s \cdot \nabla\varphi_m \\ &= \frac{\partial\varphi_m}{\partial t} + \vec{u}_s \cdot (\vec{u}_m - \vec{u}_\infty). \end{aligned} \quad (24)$$

We define the local reference velocity  $\vec{v}_{\text{ref}}$  to be

$$\vec{v}_{\text{ref}} = \vec{u}_{\infty} - \vec{u}_s. \quad (25)$$

Substituting (19), (24), and (25) in equation (23) we find after some manipulations an expression for the pressure coefficient:

$$C_p \stackrel{\text{def}}{=} \frac{p - p_{\infty}}{\frac{1}{2}\rho_{\infty}v_{\text{ref}}^2} = 1 - \frac{u_{\text{rel}}^2}{v_{\text{ref}}^2} - \frac{2}{v_{\text{ref}}^2} \frac{D_s \varphi_m}{Dt} + \frac{2}{v_{\text{ref}}^2} \vec{g} \cdot (\vec{x} - \vec{x}_{\text{ref}}), \quad (26)$$

where  $u_{\text{rel}}^2 = \vec{u}_{\text{rel}} \cdot \vec{u}_{\text{rel}}$  and  $v_{\text{ref}}^2 = \vec{v}_{\text{ref}} \cdot \vec{v}_{\text{ref}}$ .

The total force vector and moment vector are determined by integrating the pressure distribution over the surface of the configuration (see Figure 4):

$$\vec{F}(t) = - \iint_S p \vec{n} dS, \quad (27)$$

$$\vec{M}(t) = \iint_S p \vec{n} \times \vec{r} dS, \quad (28)$$

where  $\vec{r}$  is the position vector from a moment reference point to the surface, and  $p$  the local pressure acting on the surface of the body in the direction towards the surface.

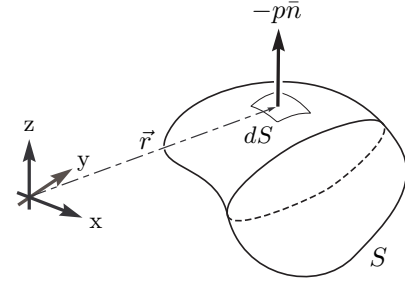


Figure 4. Surface pressure integration.

### 3. Numerical Details

Boundary integral equation (12) is discretized as a low-order panel method (see Hoeijmakers<sup>16</sup>, Katz and Plotkin<sup>17</sup>) using the definitions for the perturbation potentials for dipole distributions (6) and source distributions (7). In the low order approach each body panel  $j$  carries a constant source distribution  $\sigma_j$  of known strength and a constant dipole distribution  $\mu_j$  of unknown strength. The  $N_w$  downstream wake panels  $j_w$  have a known dipole strength  $\mu_{j_w}$  that follows from Equations 20 and 22. Boundary condition (11) is enforced at each body panel collocation point  $\vec{x}_i$ , located just below the panel's surface midpoint, and determines through (14) the panel source strength  $\sigma_j$ . The discrete set of equations for  $N_b$  collocation points to solve for the  $N_b$  unknown body dipole strengths  $\mu_j$  that results is

$$\sum_j^{N_b} A_{ij} \mu_j = - \sum_j^{N_b} B_{ij} \sigma_j - \sum_{j_w}^{N_w} A_{ij_w} \mu_{j_w}, \quad \text{for } i = 1..N_b, \quad (29)$$

where  $A_{ij}$  and  $B_{ij}$  are so-called aerodynamic influence coefficients defined by

$$A_{ij} = \frac{-1}{4\pi} \iint_{S_j} \frac{\vec{n}_j \cdot \vec{r}}{r^3} dS, \quad B_{ij} = \frac{-1}{4\pi} \iint_{S_j} \frac{1}{r} dS, \quad (30)$$

in which  $\vec{r} = \vec{x}_i - \vec{y}$ ,  $r = |\vec{r}|$ , and  $\vec{y} \in S_j$ . The gradient with respect to  $\vec{x}$  of above expressions for the velocity potentials gives the associated velocity vector fields:

$$\vec{C}_{ij} = \frac{-1}{4\pi} \nabla_x \iint_{S_j} \frac{\vec{n}_j \cdot \vec{r}}{r^3} dS, \quad \vec{D}_{ij} = \frac{-1}{4\pi} \nabla_x \iint_{S_j} \frac{1}{r} dS. \quad (31)$$

In Katz and Plotkin<sup>17</sup> analytical expressions for (30) and (31) are given for flat panels. In case the panel and collocation point are far apart the integrals are approximated by a simple single-point quadrature rule. For the velocity due to a constant strength dipole distribution on a bilinear panel geometry we make use of the equivalence with a ring vortex consisting of line elements along the edge of the panel. The velocity contribution of a vortex line element with vortex strength  $\Gamma = \mu$  is also known as the Biot-Savart law and reads

$$\vec{u}_{\Gamma}(\vec{x}_p) = \frac{-1}{4\pi} \int \Gamma \frac{\vec{r} \times d\vec{l}}{r^3}. \quad (32)$$

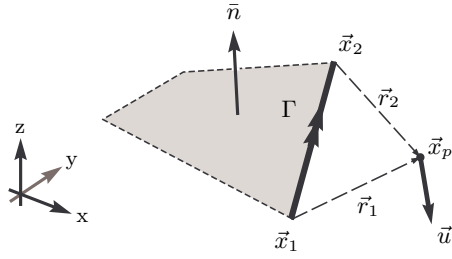


Figure 5. Ring vortex line element.

For a straight line element with constant vortex strength  $\Gamma$  its contribution to the integral for the ring vortex velocity field can be expressed analytically as

$$\vec{u}_\Gamma(\vec{x}_p) = \frac{\Gamma}{4\pi} \frac{(r_1 + r_2)(\vec{r}_1 \times \vec{r}_2)}{r_1 r_2 (r_1 r_2 + \vec{r}_1 \cdot \vec{r}_2)}. \quad (33)$$

The terms  $\vec{r}_1$  and  $\vec{r}_2$  are the position vectors of the first and last point of the line  $\vec{x}_1$  and  $\vec{x}_2$  respectively, to the evaluation point  $\vec{x}_p$  (see Figure 5). The  $r_1$  and  $r_2$  terms are corresponding distances and the orientation of the vortex line element is determined by the panel normal vector in a right-hand sense.

Summation of the contributions of the panel line elements for  $\Gamma = 1$  gives the dipole velocity vector coefficient  $\vec{C}_{ij}$ .

Given source  $\sigma$  and dipole  $\mu$  strengths, the velocity vector at a point  $\vec{x}_p$  is evaluated by

$$\vec{u}(\vec{x}_p) = \sum_j^{N_b} \vec{D}_{pj} \sigma_j + \sum_j^{N_b} \vec{C}_{pj} \mu_j + \sum_{j_w}^{N_w} \vec{C}_{pj_w} \mu_{j_w}. \quad (34)$$

For the deformation of the wake in time we determine the velocity vector at all downstream wake panel corner points and integrate equation (21) over a time interval. Each wake panel preserves its dipole strength throughout the simulation, as stated by equation (22). For wind turbine rotors the number of wake panels easily exceeds the number of panels on the rotor blades by a factor 10 and thus most of the computational cost for the wake deformation comes from the wake panel dipole distributions  $\mu_{j_w}$  themselves. The work involved in determining the velocities at  $N_w$  wake panel corner points by evaluation of the contributions of the  $N_w$  wake panels is  $\mathcal{O}(N_w^2)$ .

For the velocity vector  $\vec{u}_m(\vec{x}, t)$  on the surface of the configuration the relationship in equation (18) is used. The surface gradient of the dipole strength  $\nabla_s \mu$  is determined with the help of the gradient theorem:

$$\iint_S \nabla_s \mu dS = \int_{\partial S} \mu \vec{\nu} dl, \quad (35)$$

where  $\vec{\nu}$  is the unit outward vector normal to the integration contour  $\partial S$  and tangential to the surface (see Figure 6):

$$\vec{\nu} = \frac{\vec{\tau} \times \vec{n}}{|\vec{\tau} \times \vec{n}|}, \quad (36)$$

in which  $\vec{\tau}$  is the vector tangential to the contour and  $\vec{n}$  the local unit normal vector. This gives an expression for the average dipole surface gradient inside contour  $\partial S$

$$\overline{\nabla_s \mu} = \frac{1}{S} \int_{\partial S} \mu \vec{\nu} dl. \quad (37)$$

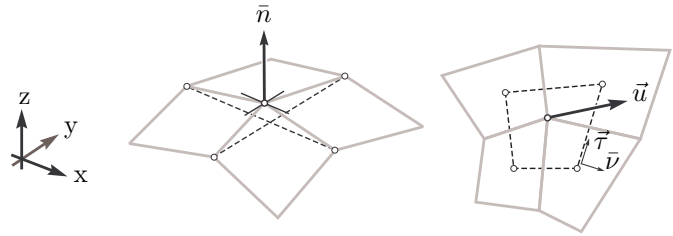


Figure 6. Definitions used in the contour integral to determine the surface tangential perturbation velocity in a grid node due to a dipole distribution.

In the current implementation the contour  $\partial S$  is defined by the straight lines  $\vec{\tau}$  through the collocation points of the four neighboring panels and the unit normal vector  $\vec{n}$  in a panel node is constructed from a vector cross product using the four nodes that connect to it, see Figure 6. The perturbation velocity at the grid node due to the dipole distribution is assigned this average surface gradient.

Equations (18) and (19) give the total and the relative velocity vector fields in the grid nodes of the configuration.

## B. Verification of the Panel Method

In this section we compare panel method results with the exact solution for a tri-axial ellipsoid in uniform onset flow and for a rotating ellipsoid in air at rest. For these test cases the low order panel method exhibits an accuracy of  $\mathcal{O}(h^2)$ , with  $h$  a characteristic panel length. In the tests we use an ellipsoid with semi-axes  $(a, b, c) = (4, 2, 1)$ .

The surface of an ellipsoid with semi-axes  $(a, b, c)$  is described by

$$\left(\frac{x_e}{a}\right)^2 + \left(\frac{y_e}{b}\right)^2 + \left(\frac{z_e}{c}\right)^2 = 1, \quad (38)$$

where  $\vec{x}_e = (x_e, y_e, z_e)^T$  is a point on the surface of the ellipsoid.

The discrete version of the  $L_2$ -norm used in this paper as a measure for the error is

$$L_2(x) = \left(\frac{1}{n} \sum_{i=1}^n |x_i|^2\right)^{1/2}, \quad (39)$$

with  $x_i$  the error at point  $i$ .

### 1. Tri-axial Ellipsoid in Uniform Onset Flow

For ellipsoidal bodies in a uniform onset flow analytical potential flow solutions exist (see Durand<sup>12</sup>, Lamb<sup>20</sup>, Milne-Thomson<sup>23</sup>). The perturbation velocity potential on the surface of this ellipsoid for a general uniform onset flow  $\vec{u}_\infty = (u_\infty, v_\infty, w_\infty)^T$ , is given by

$$\varphi(\vec{x}_e) = u_\infty x_e \frac{\alpha_0}{2 - \alpha_0} + v_\infty y_e \frac{\beta_0}{2 - \beta_0} + w_\infty z_e \frac{\gamma_0}{2 - \gamma_0}, \quad (40)$$

where the coefficients are related to the elliptic integrals

$$\alpha_0 = abc \int_0^\infty \frac{1}{(a^2 + \lambda) \Delta_\lambda} d\lambda, \quad \text{with} \quad \Delta_\lambda = \sqrt{(a^2 + \lambda)(b^2 + \lambda)(c^2 + \lambda)}. \quad (41)$$

and similar expressions for  $\beta_0$  and  $\gamma_0$ .

For the total potential on the surface of the ellipsoid we have  $\Phi(\vec{x}_e) = \vec{u}_\infty \cdot \vec{x}_e + \varphi(\vec{x}_e)$ . The surface velocity is obtained by taking the surface gradient of  $\Phi$ .

Figure 7 shows the errors in the velocity potential and in the pressure coefficient as function of characteristic panel length  $h$  for onset flows along each of the three axes. We define the dimensionless  $L_2$ -norm of the error in the perturbation potential as

$$\epsilon_\varphi = \frac{L_2(\varphi_{\text{exact}} + \mu)}{U_{\text{ref}} L_{\text{ref}}}, \quad (42)$$

and set the reference length and velocity equal to  $L_{\text{ref}} = 8$  and  $U_{\text{ref}} = 1$ , respectively. For this test case the error is  $\mathcal{O}(h^2)$  in the  $L_2$  norm for both the velocity potential and the pressure coefficient.

### 2. Rotating Tri-axial Ellipsoid in Flow at Rest

Following the treatises by Lamb<sup>20</sup>, Milne-Thomson<sup>23</sup>, and Munk in Durand<sup>12</sup>, the perturbation velocity potential of a tri-axial ellipsoid in a fluid at rest, rotating around an arbitrarily oriented axis  $\vec{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^T$  through its center, for a point  $\vec{x}_e$  on the surface of the ellipsoid can be written as

$$\varphi(\vec{x}_e) = \Omega_x y_e z_e \xi_0 + \Omega_y z_e x_e \eta_0 + \Omega_z x_e y_e \zeta_0, \quad (43)$$

where the coefficient  $\xi_0$  is defined by

$$\xi_0 = \frac{-abc(b^2 - c^2)^2}{2(b^2 - c^2) + (b^2 + c^2)(\beta_0 - \gamma_0)} \int_0^\infty \frac{1}{(b^2 + \lambda)(c^2 + \lambda) \Delta_\lambda} d\lambda, \quad (44)$$



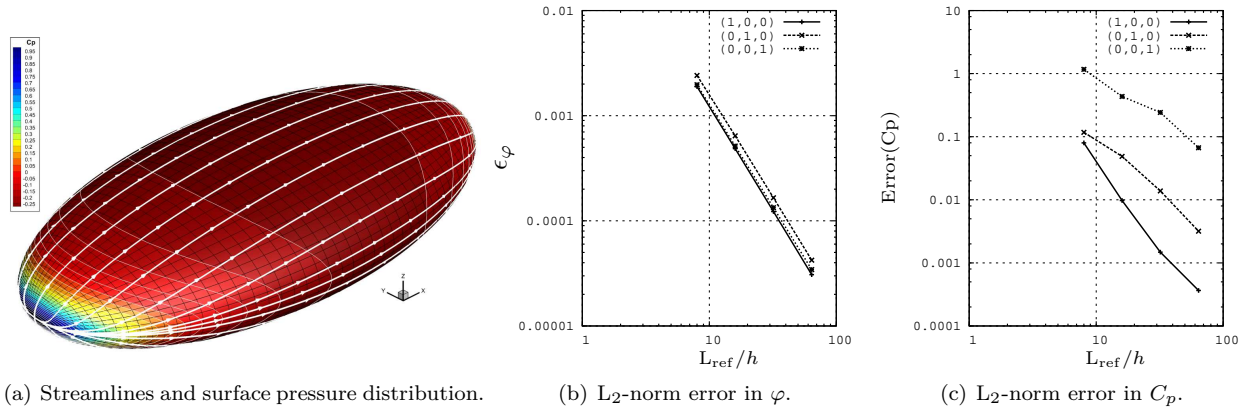


Figure 7. Error norms as a function of characteristic panel length  $h$ , for onset flows along the three coordinate axes.

with similar relationships for  $\eta_0$  and  $\zeta_0$ . After some manipulation the values for  $\xi_0$ ,  $\eta_0$ , and  $\zeta_0$  can be expressed (see Van Garrel<sup>15</sup>) in terms of the values  $\alpha_0$ ,  $\beta_0$ , and  $\gamma_0$  for the ellipsoid in uniform onset flow discussed in the preceding section.

The perturbation velocity vector tangential to the surface can now be determined from the surface gradient of the perturbation potential. The normal velocity component follows from the normal component of the surface velocity:

$$\vec{u}_n(\vec{x}_e) = \left( (\vec{\Omega} \times \vec{x}_e) \cdot \vec{n}_e \right) \vec{n}_e. \quad (45)$$

The surface pressure coefficient is obtained via the Bernoulli equation (26) from the velocity at the surface and the material derivative of the perturbation velocity potential. However, in this case a reference velocity constructed from the (zero) onset velocity and the local surface velocity would give a zero reference velocity at points on the rotation axis and lead to an amplification of the error in the pressure coefficient near these points. Therefore, a constant reference velocity of  $U_{\text{ref}} = |\vec{\Omega}| L_{\text{ref}}/2 = 4$  is used to obtain non-dimensional pressure coefficients and determine the surface distribution of the error.

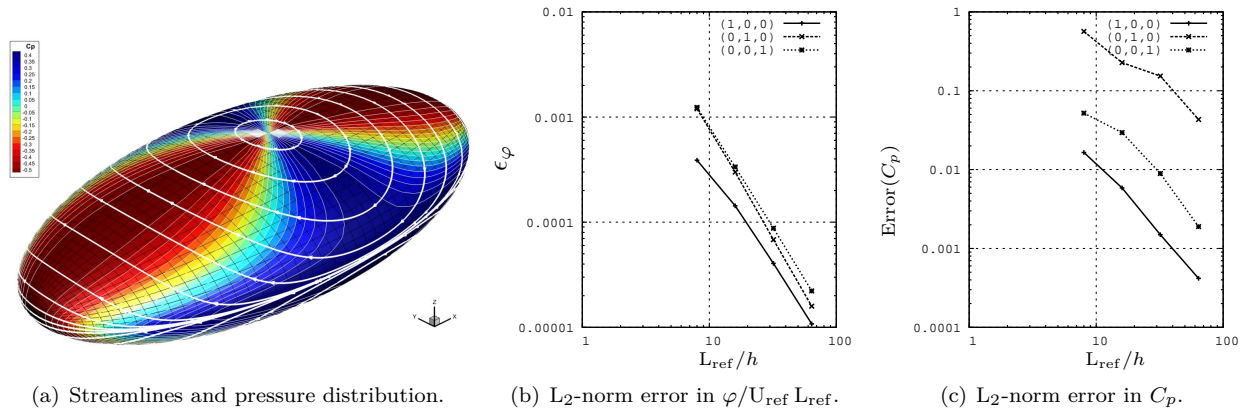


Figure 8. Error norms as a function of characteristic panel length  $h$ , for rotation along the three coordinate axes in air at rest.

Figure 8 shows the  $L_2$  norm of the error in the perturbation potential and in the pressure coefficient for rotation along each of the three coordinate axes. The error is  $\mathcal{O}(h^2)$  in the  $L_2$  norm for the velocity potential. The behavior of the error in the pressure coefficient is less regular than the error in the velocity potential.

### III. Fast Multilevel Algorithm

#### A. Background

We apply the panel method to the simulation of unsteady three-dimensional incompressible inviscid flow around wind turbine rotor blades. The wakes that emanate from the trailing edge of the rotor blades increase the problem size every time step and the discretization easily reaches hundreds of thousands of panels for a single rotor. Moreover, each new time step the geometry is altered and a new evaluation of the aerodynamic influence coefficients has to be performed.

In a wind turbine engineering environment short problem turnaround times are a prerequisite in the iterative computational design process of the rotor blade, as well as the phase of the wind turbine structural dynamics analysis. Problem turnaround times consist of the preparation time needed to setup a simulation plus the actual simulation run time. Therefore, a conventional  $\mathcal{O}(N^2)$  approach quickly becomes infeasible as Figure 9 illustrates for a single time step in a relevant 3D wake roll-up simulation. The development of advanced multilevel algorithms provides a very significant alleviation of this computational burden. In our work<sup>15</sup> it has been proven that a practical solution can be developed that reduces this burden to  $\mathcal{O}(N)$  work. The significance of the multilevel method developed in this work is already illustrated in the same figure and its advantage for large problem sizes is obvious.

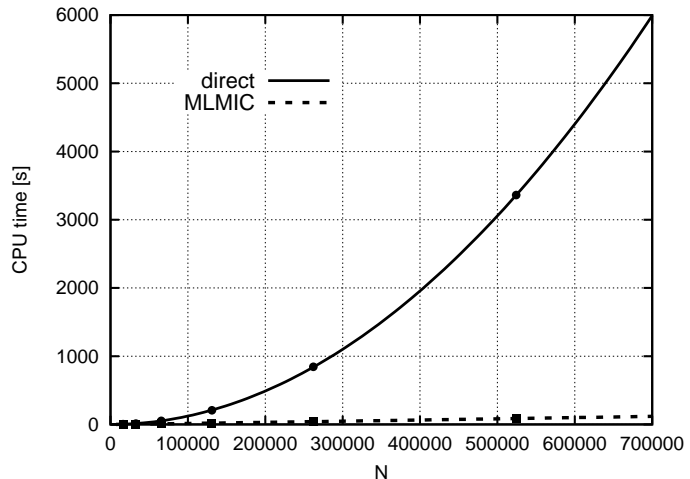


Figure 9. Work growing quadratically with problem size  $N$  for a conventional panel method implementation. For a relevant 3D test case the new MLMIC scheme shows a substantial reduction of work.

The algorithm builds upon the multilevel multi-integration concept of Brandt and Lubrecht,<sup>7</sup> Brandt,<sup>5</sup> Venner and Lubrecht.<sup>30</sup> New concepts are introduced so that the method is also practically applicable to highly irregular surfaces as appearing in the (panel method) wakes of wind turbine rotor blades. We will refer to the new approach as the MLMIC (Multi-Level Multi-Integration Cluster) scheme.

The MLMIC algorithm is described first for the 1D case. The extension to 3D space is discussed later. Before we start describing the basic concepts of the MLMIC scheme there are first some remarks to be made on the wording in the rest of this section.

The term *source* is used in its general meaning and refers to any kernel function. *Receivers* denote the locations where the evaluation of an integral transform is needed. The term *node* is used exclusively for vertices in the (background) grid boxes and for all other situations the word *point* is used. We use the term *box* for a rectangular bounded domain, and a set of points in such a bounded domain is called a *cluster*. A box containing a cluster is said to be *active*. Furthermore, we will use lower case and upper case letters for fine and coarse grid variables, respectively. Identifier  $x$  and subscript  $i$  are used for receiver quantities, and identifier  $y$  and subscript  $j$  for source quantities. Discrete values are identified with a superscript  $h$ . For the  $d$ -dimensional formulation subscripts  $1, \dots, d$  will be used.

The canonical form of the integral transform in  $d$ -dimensional space  $\Omega \in \mathbb{R}^d$  is

$$\phi(x) = \int K(x, y) \sigma(y) dy, \quad (46)$$

which expresses the value of  $\phi(x)$  at receiver point  $x$  as the integral of the product of a source function  $\sigma(y)$  in  $y$  and a kernel function  $K$  that depends both on  $x$  and  $y$ , with possibly the domains of  $x$  and  $y$  overlapping. Often both receiver and source domains  $x$  and  $y$  are lower dimensional manifolds  $\partial\Omega \in \mathbb{R}^{d-1}$  but this is not a requirement. In linear potential theory the kernel function  $K(x, y)$  is a Green's function that only depends on the relative position of source and receiver points  $K(x, y) = K(|x - y|)$ . However, this property is not

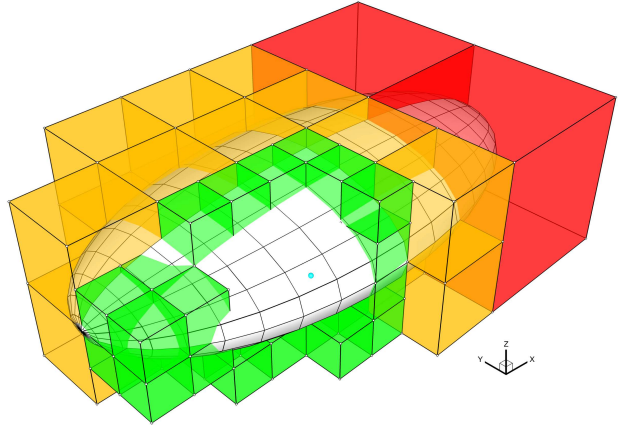
required for the multilevel algorithm to be applicable.

The continuous integral transform (46) is discretized (using the panel method) and evaluated at receiver points  $x_i^h$  as

$$\phi^h(x_i^h) = \phi_i^h = \int K(x_i^h, y) \tilde{\sigma}^h(y) dy = \sum_j K_{i,j}^{hh} \sigma_j^h, \quad (47)$$

where  $\tilde{\sigma}^h(y)$  is a piecewise polynomial representation of the source distribution and the degrees of freedom  $\sigma_j^h$  and  $K_{i,j}^{hh}$  defined such that equation (47) holds. In the panel method in  $\mathbb{R}^d$  the domain of integration is most often a surface in  $\mathbb{R}^{d-1}$  discretized using a distribution of non-overlapping panels. In case of a piecewise constant source distribution, each degree of freedom  $\sigma_j^h$  used to describe the source distribution is associated with exactly one panel. For clarity it is first assumed that the source distribution is concentrated in points  $y_j^h$  with strength  $\sigma_j^h$  in one-dimensional space, that is,  $\tilde{\sigma}^h(y) = \sigma_j^h \delta(y - y_j^h)$ . In this case  $K_{i,j}^{hh} = K(x_i^h, y_j^h)$ . We also assume that we have a scalar kernel function.

In the MLMI algorithm by Brandt and Lubrecht,<sup>7</sup> Brandt,<sup>5</sup> and Venner and Lubrecht<sup>30</sup> a fast evaluation is obtained by the combination of geometric coarsening and interpolation. The boundary  $\partial\Omega$  is represented by a hierarchy of coarsened grids (levels) that are constructed from the definition of the boundary geometry. For singular kernel functions the algorithm exploits the smoothness of the kernel function for distant receiver and source points on the boundary by replacing the exact values of the kernel by polynomial interpolations from coarse grid values. The influence of sources on receivers at the same grid level are corrected when the receiver and source points are too close and the interpolation of the kernel function becomes inaccurate. A prerequisite for such a scheme is that the underlying geometry and parametrization is sufficiently smooth to warrant the kernel interpolation. Wind turbine rotor wakes, however, deform strongly under self-induction and the resulting manifolds are highly irregular even for normal flow conditions.

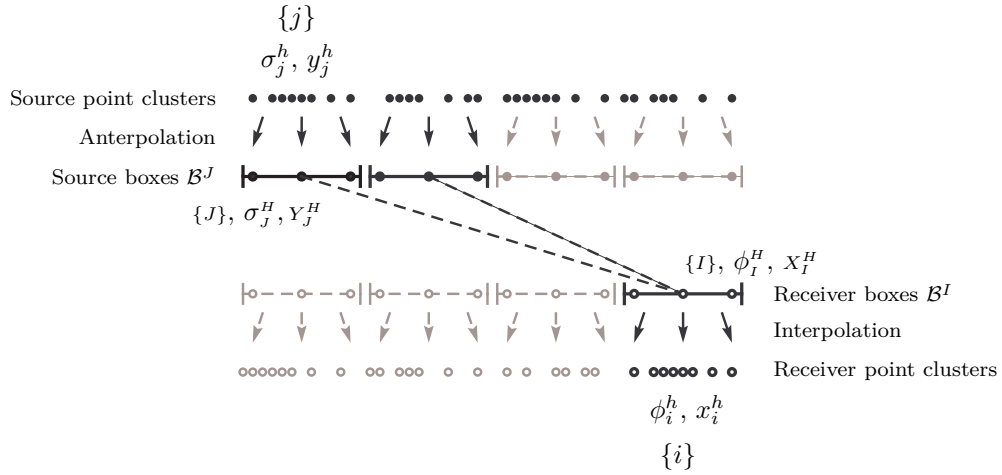


**Figure 10. Hierarchy of boxes around an evaluation point on a surface.**

The key idea here is that the kernel  $K(x, y)$  is generally a well defined function not only on boundary  $\partial\Omega \in \mathbb{R}^{d-1}$  but throughout the domain  $\Omega \in \mathbb{R}^d$ . This observation is used to remove the smoothness constraint on boundary  $\partial\Omega$  when  $y \in \partial\Omega$  through the introduction of (background) grids in  $\Omega$  for which grid smoothness can be enforced and  $y \in \Omega$ . This will make it possible to apply the scheme even in cases with highly distorted surfaces such as deformed wind turbine wake vortex sheets. Obviously this new flexibility comes at the cost of the introduction of an extra space dimension. Volume grids were also discussed by Brandt,<sup>56</sup> in the context of many-body interactions.

For asymptotically smooth, singular kernels like  $K(x, y) = |x - y|^{-1}$  the function and its higher-order derivatives become unbounded when receiver points  $x$  and source points  $y$  approach each other. As a result, a naive replacement of the kernel function by an interpolating polynomial through  $p$  nodes in a box of size  $h$  will give an unacceptable high error of  $\mathcal{O}(K^{(p)} h^p)$  which translates for the above given kernel function to an error of  $\mathcal{O}(|x - y|^{-(p+1)} h^p)$ . This suggests two measures that can be taken to avoid these problems: decrease the box size  $h$  or use the interpolating polynomial only when the kernel function is behaving smoothly and its derivatives are bounded, that is when receiver points  $x$  and source points  $y$  are distant and not in each others near field (according to some criterion yet to be defined). Alternatively one can replace the original kernel with a function with bounded higher order derivatives (kernel softening) and correct for the error a posteriori as for example done by Sandak.<sup>28</sup>

To describe the MLMIC method concisely let us introduce the notation  $\mathcal{B}^J$  for a source box,  $\mathcal{B}^I$  for a receiver box, and  $\{\cdot\}$  for a set. We introduce the shorthand notation  $\{i\}$  for the set of receiver points  $x_i^h$  contained in box  $\mathcal{B}^I$ , and  $\{I\}$  for the set of grid nodes  $x_I^h$  in receiver box  $\mathcal{B}^I$ . Similarly for the sources:  $\{j\}$  denotes the set of source points  $y_j^h$  in box  $\mathcal{B}^J$ , and  $\{J\}$  is the set of grid nodes  $y_J^h$  in source box  $\mathcal{B}^J$ .



**Figure 11.** Concepts in the MLMIC algorithm for asymptotically smooth, singular kernels using uniform boxes. Source boxes in the near field of a receiver box do not contribute to the result at that level.

The notion of separating space around a receiver box  $\mathcal{B}^I$  into a near field and a far field is key in the MLMIC method for asymptotically smooth kernels. The MLMIC method is centered around the concept of point clusters contained in boxes, so the natural choice is to distinguish near field and far field based upon the distance between receiver and source box centers:

$$d = \text{dist}(\mathcal{B}^I, \mathcal{B}^J)/h, \quad (48)$$

where  $h$  is the box size and  $\text{dist}()$  is defined as the maximum of the distances in each separate coordinate direction between the centers. We introduce a near field distance parameter  $d_{\text{nf}}$  that defines whether a source box is in the near field of a receiver box or in its far field:

$$\begin{aligned} d \leq d_{\text{nf}} & : \text{ near field,} \\ d > d_{\text{nf}} & : \text{ far field.} \end{aligned} \quad (49)$$

Thus, a “ring” of  $d_{\text{nf}}$  source boxes defines the perimeter of a receiver box near field. For example, in Figure 11 where  $d_{\text{nf}} = 1$  only the immediate neighbors of a receiver box are in its near field.

For convenience let us write the discretized canonical integral equation here again

$$\phi_i^h = \sum_j K_{i,j}^{hh} \sigma_j^h, \quad (50)$$

and introduce the approximation for kernel values  $K_{i,j}^{hh}$  by interpolating the kernel function through receiver and source grid nodes  $x_I$  and  $y_J$ , respectively. For the interpolating polynomials on the receiver and source sides we use the shorthand notations  $L^I(x)$  and  $L^J(y)$ , respectively. We can now write

$$K_{i,j}^{hh} = \tilde{K}_{i,j}^{hh} + \mathcal{O}(\epsilon), \quad (51)$$

where

$$\tilde{K}_{i,j}^{hh} = \sum_I \sum_J K_{I,J}^{HH} L^I(x_i^h) L^J(y_j^h), \quad (52)$$

and

$$K_{I,J}^{HH} = K(x_I^H, y_J^H). \quad (53)$$

We will subdivide the domain into multiple boxes of the same size and use Lagrange interpolation through  $p$  background grid nodes in each of the active boxes that encompass receiver and source point clusters. The error we make by this approximation is of  $\mathcal{O}(K^{(p)}h^p)$ , with  $h$  the size of the box. The only requirement

for the receiver and source background grids is that the kernel function can be represented accurately by polynomial interpolation on these grids. In our implementation we will use a Chebyshev node distribution of the first kind in which the nodes in each grid box are distributed more densely towards the boundaries of the box. This results in a more evenly distributed interpolation error. An accessible description of the barycentric form of Lagrange interpolation and the advantages of using Chebyshev nodes is discussed by Berrut and Trefethen.<sup>2</sup>

Substitution of (52) into (50) gives

$$\phi_i^h \approx \sum_j \sum_I \sum_J K_{I,J}^{HH} L^I(x_i^h) L^J(y_j^h) \sigma_j^h, \quad (54)$$

which can be reordered without any assumptions into

$$\phi_i^h \approx \sum_I L^I(x_i^h) \left( \sum_J K_{I,J}^{HH} \left( \sum_j L^J(y_j^h) \sigma_j^h \right) \right). \quad (55)$$

Let us introduce the notation  $\{J\}_{\text{ff}}$  for the source nodes in boxes  $\mathcal{B}^J$  in the far field of  $\mathcal{B}^I$ , that is when  $\text{dist}(\mathcal{B}^I, \mathcal{B}^J)/h > d_{\text{nf}}$ . In the same spirit we introduce  $\{j\}_{\text{nf}}$  for the clusters of source points in boxes  $\mathcal{B}^J$  in the near field of  $\mathcal{B}^I$  for which  $\text{dist}(\mathcal{B}^I, \mathcal{B}^J)/h \leq d_{\text{nf}}$ .

In equation (55) the first three components in the MLMIC scheme for asymptotically smooth kernels can be distinguished going inward-out of the parentheses. With the help of Figure 11, and introducing some new nomenclature, the resulting MLMIC scheme for asymptotically smooth kernels is now as follows:

### 1. Anterpolation

For each block  $\mathcal{B}^J$  compose clusters of source points in  $y_j^h \in \mathcal{B}^J$  with strength  $\sigma_j^h$  and transfer these strengths to “pseudo-sources”  $\sigma_j^H$  located in the source background grid nodes  $y_j^H \in \mathcal{B}^J$  by

$$\sigma_j^H = \sum_{\{j\}} L^J(y_j^h) \sigma_j^h. \quad (56)$$

### 2. Coarse grid summation

For each receiver box  $\mathcal{B}^I$  the kernel values  $K_{I,J}^{HH}$  representing the influence of source box  $\mathcal{B}^J$  on receiver block  $\mathcal{B}^I$  are used to account for the effects of the “pseudo-sources”  $\sigma_j^H$  in the far field of box  $\mathcal{B}^I$ . The resulting receiver values  $\phi_I^H$  in the receiver box grid nodes  $x_I^H \in \mathcal{B}^I$  are obtained through

$$\phi_I^H = \sum_{\{J\}_{\text{ff}}} K_{I,J}^{HH} \sigma_j^H. \quad (57)$$

By excluding the contributions from the source boxes in the near field the error that would result from interpolation of the kernel function is avoided.

### 3. Interpolation

For each receiver box  $\mathcal{B}^I$  the receiver values  $\phi_i^h$  at the grid nodes  $x_i^H \in \mathcal{B}^I$  of the receiver grid through

$$\phi_i^h = \sum_{\{I\}} L^I(x_i^h) \phi_I^H. \quad (58)$$

### 4. Cluster near field

For each receiver box  $\mathcal{B}^I$  the receiver values  $\phi_i^h$  in points  $x_i^h \in \mathcal{B}^I$  are complemented with the contributions from the near field source points with strength  $\sigma_j^h$  by

$$\phi_i^h := \phi_i^h + \sum_{\{j\}_{\text{nf}}} K_{i,j}^{hh} \sigma_j^h. \quad (59)$$

One of the crucial differences between the current scheme and the MLMI scheme by Brandt and Lubrecht<sup>7</sup>, Brandt<sup>5</sup>, and Venner and Lubrecht<sup>30</sup> is that it avoids the introduction of corrections to the intermediate results on the coarser grids. This is a consequence of the containment of information in the boxes. An additional advantage of the separate final cluster near field contribution in equation (59) is that it avoids expensive removal of self-inductions as were encountered by Sandak.<sup>28</sup>

The coarse grid summation in equation (57) can be carried out recursively using a hierarchy of increasingly larger boxes. The nodes in smaller boxes at the finer grid level take the role of the points described in the algorithm above.

For the extension to higher dimensions the one-dimensional Lagrange basis polynomials  $L^I$  and  $L^J$  are replaced by their tensor product equivalents. Anterpolation and interpolation between boxes are performed one dimension at a time.

In the current 3D MLMIC implementation we use cubes to fill the domain and employ the same polynomial order  $p$  for the interpolation in receiver and source boxes in each direction and on all grid levels. Each box is given a Chebyshev node distribution of the first kind. The near field criterion (see equation (48)) is set to  $d_{\text{nf}} = 1$ , and the coarsest grid level consists of at most 3 boxes in each direction giving a total of 27 boxes. Each box at a coarser grid level consists of 8 ‘child’ boxes at the finer grid level. The same box decomposition of space is employed for the receiver and for the source side.

Figure 12 shows a 2D representation of the box setup and the source boxes  $\mathcal{B}^J$  contributing to the result in a single receiver block  $\mathcal{B}^I$  at the same grid level. The final cluster near field contribution (equation (59)) comes from the source points or from the panels from which they originate. In this setup the free parameters are the polynomial interpolation order  $p$  and the size  $h$  of the boxes at the finest grid level.

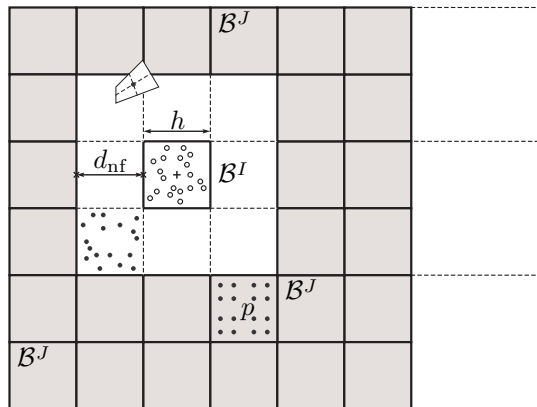


Figure 12. Active source boxes  $\mathcal{B}^J$  interacting with receiver cube  $\mathcal{B}^I$  on the same grid level. Here a ring of  $d_{\text{nf}} = 1$  source boxes outside the receiver box defines the perimeter of its near field.

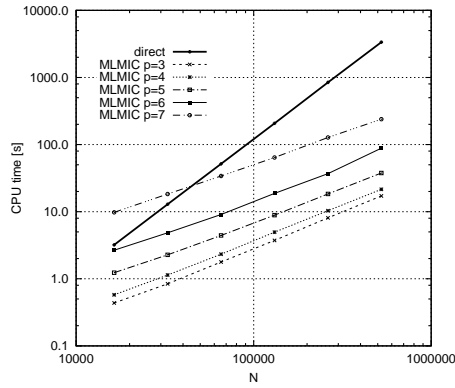
## B. Verification of the MLMIC Method

On the basis of model problems we demonstrate in this section that the error in integral values  $\phi_i^h$  and the computational time in the MLMIC scheme is controlled through the choice of polynomial order and the finest grid box sizes.

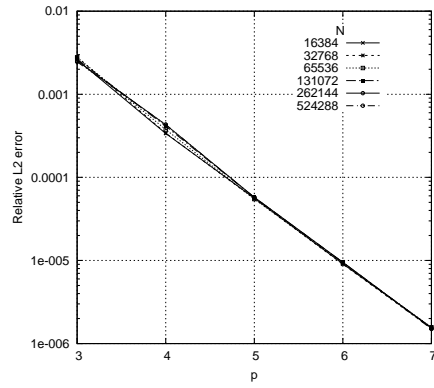
In this test we start with a tri-axial ellipsoid with semi-axes (4, 2, 1) and a total of 16,384 panels (128 along the x-axis, and 128 along the circumference) and double the size and number of panels in x-direction until we have an ellipsoid with semi-axes (128, 2, 1) and a total of 524,288 panels, now with 4096 panels along the x-axis. This situation, for which the addition of extra panels results in an increase in domain size in one direction, is similar to the case of a growing wake downstream of a wind turbine.

We use the panel areas as source strength and panel midpoints as the cloud of point sources. The receiver points have a slight offset from point sources. The box size on the finest grid level was set to  $h = 0.25$  for the first set of simulations. For a second set of simulations we chose a  $h = 0.50$  at the finest grid level. The polynomial order was varied from 3 to 7 and the results are compared with the results from the direct evaluation. Figure 13 shows the results for the 3D dipole potential kernel, that is  $K(\vec{x}, \vec{y}) = (\vec{n} \cdot \vec{r})/r^3$  with  $r = |\vec{r}|$ , and  $\vec{r} = \vec{x} - \vec{y}$ .

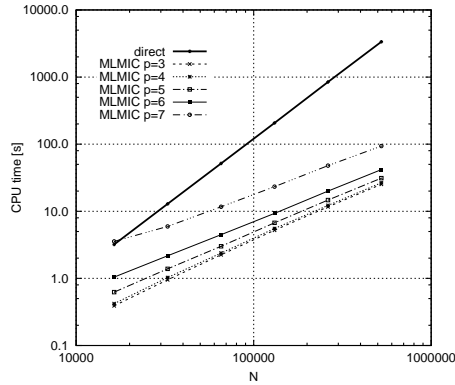
The computational work as function of problem size  $N$  is shown in the left side of Figure 13 and we can deduce that the cost of the direct evaluation is  $\mathcal{O}(N^2)$  and that of the MLMIC scheme in this setup is  $\mathcal{O}(N)$ . Notice that especially for the lower polynomial orders large reductions in computational time result from the MLMIC scheme.



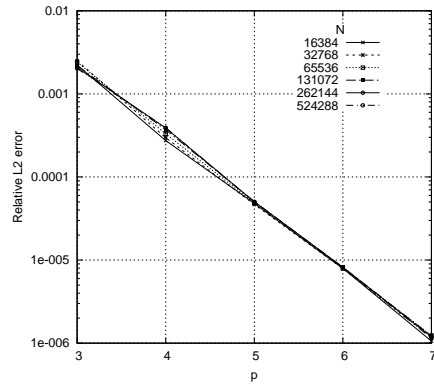
(a) CPU time as function of problem size  $N$ .  
Finest grid box size  $h = 0.25$



(b) Relative  $L_2$ -norm errors as function of polynomial order  $p$ .  
Finest grid box size  $h = 0.25$



(c) CPU time as function of problem size  $N$ .  
Finest grid box size  $h = 0.50$



(d) Relative  $L_2$ -norm errors as function of polynomial order  $p$ .  
Finest grid box size  $h = 0.50$

**Figure 13.** MLMIC scheme CPU time and  $L_2$ -errors for the 3D dipole potential kernel. Source points are distributed on a set of tri-axial ellipsoids of varying length and the finest grid box sizes  $h = 0.25$  and  $h = 0.50$ .

For box size  $h = 0.50$  at the finest grid level the lines of constant interpolation order  $p$  in the figures of CPU time as function of problem size  $N$  are closer to each other as here the final near field contribution dominates the computational time.

The right side of Figure 13 shows the accompanying relative  $L_2$ -norm of the errors in values  $\phi$  as incurred by the MLMIC method as function of polynomial order  $p$ . For interpolation of kernels with bounded higher order derivatives the maximum error is  $\epsilon \in \mathcal{O}(h^p)$ , so that  $\ln(\epsilon) \propto p$  for fixed box size  $h$ . This linear dependence of the logarithm of the  $L_2$ -norm of the error on interpolation order  $p$  is apparent in Figure 13. The same figure shows a weak dependence of the  $L_2$ -norm of the error on problem size  $N$ . Here we defined the relative  $L_2$ -norm error  $\epsilon_{\text{rel}}$  as

$$\epsilon_{\text{rel}} = \frac{L_2(\phi_{\text{MLMIC}} - \phi_{\text{direct}})}{L_2(\phi_{\text{direct}})}. \quad (60)$$

## IV. Application

We compare the results from our fast multilevel panel method with the results from the ENSOLV<sup>19</sup> flow solver of NLR and the wind tunnel test data from the MEXICO (Model EXperiments In COntrolled conditions) project. The numerical simulation method ENSOLV is a high-order finite-volume method that solves the Reynolds-averaged Navier-Stokes equation for which a  $k - \omega$  turbulence model<sup>18</sup> is selected for the present simulation runs. The MEXICO wind tunnel measurements were performed for a 4.5 [m] diameter, 3-bladed model wind turbine in two separate campaigns in the Large Scale Low Speed Facility (LLF) of the German Dutch Wind Tunnels (DNW). In 2006 the initial measurements<sup>29</sup> were executed and in 2014 a new set of measurements<sup>4</sup> was performed in the same wind tunnel for the same model wind turbine. Figure 14 gives an impression of the blade geometry and the three defining airfoil sections: NACA64-418, Risø A2-21, and DU91-W2-250.

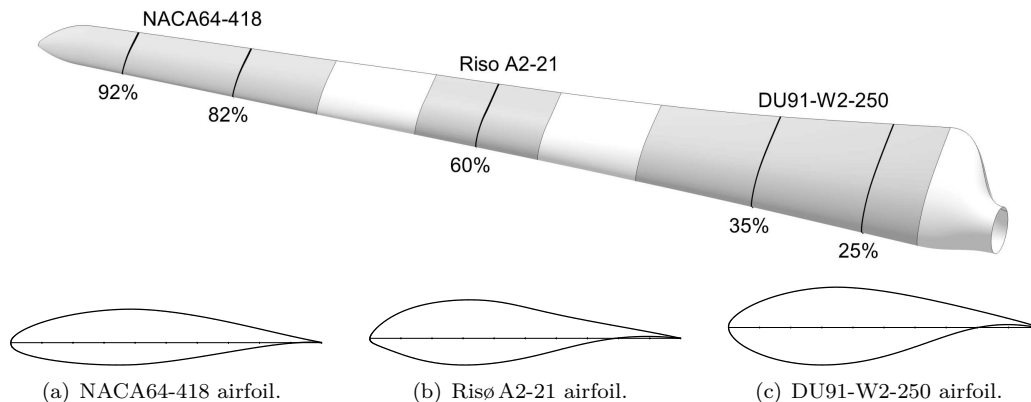


Figure 14. Airfoil sections used for the MEXICO rotor blade.

The geometry of the rotor blade has been discretized into a fine grid of 92 panels in spanwise direction and 120 panels in cross-section circumferential direction. For two of the three rotor blades a coarse grid of  $46 \times 30$  panels was selected. All three wake surfaces extend 2000 panels downstream of the trailing edge. In total this grid configuration consists of 13,800 panels for the three blade surfaces and 368,000 panels for the three wake surfaces. Figure 15 shows the three discretized blades and the discretized wake of the blade with the fine grid.

Figure 16 shows the pressure distributions obtained from the panel method, the ENSOLV method, and the MEXICO wind tunnel experiment for the rotor design condition: wind speed  $14.7 \text{ [m} \cdot \text{s}^{-1}]$  and tip speed ratio  $\lambda = U_{\text{tip}}/U_{\infty} = 6.8$ .

The surface pressure distributions for the NACA64-418 airfoil sections at 92% and 82% radius are shown in Figures 16(a) and 16(b). The panel method is based on an inviscid flow model and shows, as expected, a lower pressure along the suction side of the airfoil than measured. The effects of viscosity are represented in the ENSOLV method and as a result the computed surface pressure distributions along the suction side of the blade are closer to the experimental data. Along the pressure side of the blade sections the boundary



layer is much thinner and the result from the panel method is in very good agreement with the experimental data.

The Risø A2-21 airfoil section at 60% radius is an airfoil with a different character of the surface pressure distribution (see Figure 16(c)): a higher adverse pressure gradient between about 40% and 60% chord along the blade suction side. The results of the two numerical simulation methods show a substantial lower pressure than measured in the experiment. Note that the difference between the ENSOLV result and the experimental data is larger than the difference between the result of the panel method and that of ENSOLV. Along the pressure side of the blade section the results of both numerical simulation methods are in excellent agreement with the experimental data.

For the DU91-W2-250 airfoil section at 35% and that at 25% radius a consistent pattern is shown between the results of the panel method and that of ENSOLV, see Figures 16(d) and 16(e): the panel method produces lower pressures along the airfoil suction side. The pressure gradient in the ENSOLV results is reduced by the effects of viscosity. At the pressure side of the blade the two numerical simulation methods give similar results, except near the trailing edge where effects due to the boundary layer become more pronounced. The surface pressure distribution from the experiment shows erratic behavior for the section at 25%. This can be attributed to the sensors measuring absolute pressures and the low dynamic pressure at the innermost sections<sup>4,26</sup>, resulting in larger error bounds.

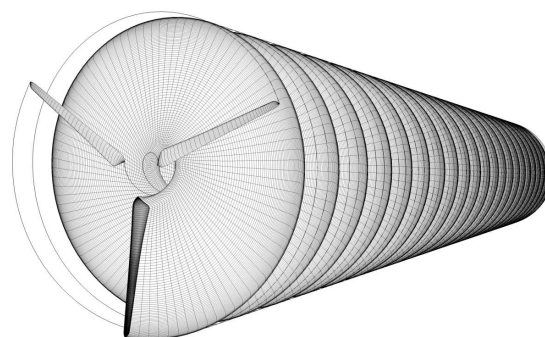


Figure 15. Discretized MEXICO rotor geometry and wake surfaces used in the panel method. For clarity only one wake surface is displayed.

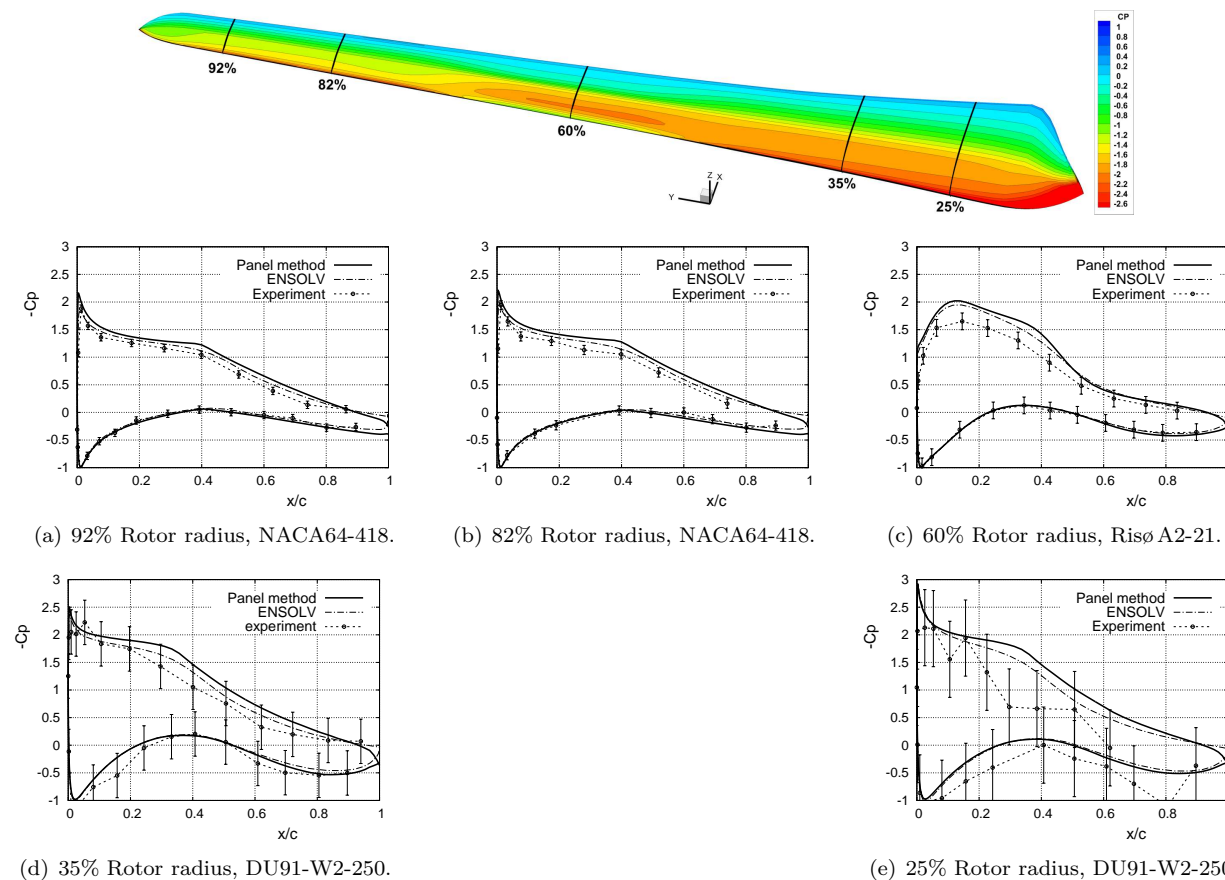


Figure 16. MEXICO rotor in uniform onset flow velocity of  $14.7 \text{ [m}\cdot\text{s}^{-1}]$  operating at tip speed ratio  $\lambda = 6.8$  (design conditions). Comparison of measured surface pressure distributions with numerical results from panel method and from ENSOLV, a state-of-the-art numerical simulation method that solves the Reynolds-averaged Navier-Stokes equations.

Similar comparisons (see Van Garrel<sup>15</sup>) have been performed for the MEXICO rotor in axial onset flow for a higher tip speed ratio ( $\lambda = 10.0$ ) as well as a lower tip speed ratio ( $\lambda = 4.2$ ) than the design condition ( $\lambda = 6.8$ ). In the case of a high tip speed ratio the local incidence of the blade sections is lower than in design condition and a similar difference between the results of the fast multilevel panel method, the ENSOLV solution method, and the experiment emerges. For the case of a low tip speed ratio the experimental data indicate that the rotor blade experiences boundary layer separation, for all sections considered, for the aft 50% to 70% of the blade suction side. The panel method does not represent the boundary layer nor flow separation. Therefore it continues to have a lower pressure along the suction side of the airfoil than obtained from the measurement. In this case the differences with the results from ENSOLV and from experimental data are larger than for the design condition and for the case of a higher tip speed ratio.

We compare the CPU times for a conventional panel method implementation with those for the present fast MLMIC panel method. The CPU time per time step is divided into three computational steps. The first step is the contribution of the wake to the system of equations, that is, the right-most term in equation (29). The second step is the solution of the system of equations by an iterative GMRES solution method. The third step is the determination of the velocities at all wake nodes required to compute the deformation and roll-up of the rotor wake, that is, the right-hand side in equation (34). For the test case at hand the dense matrices with aerodynamic influence coefficients involving the panels on the surface of the body, the two leftmost terms in equation (29), are computed once and are left out of the comparison. Table 1 shows the CPU times for the conventional panel method and those for the fast MLMIC panel method for the configuration in which the three wake surfaces extend 2000 panels downstream of the trailing edge. The CPU times have been scaled such that the conventional panel method has a total CPU time of 1.0 for the three computational steps considered.

**Table 1. Scaled CPU times per time step for the wake extending 2000 panels downstream of the trailing edge.**

Computational step	CPU Conventional	CPU MLMIC
1. Wake RHS	0.033	0.0005
2. GMRES	0.065	0.066
3. Wake roll-up	0.902	0.006
Total	1.0	0.073

For the conventional panel method the computation of the wake deformation and roll-up (step 3. in Table 1) contributes the most to the total CPU time. The fast MLMIC method reduces the CPU time for the roll-up of the wake by a factor of 150. For an off-the-shelf personal computer with a 3.4 GHz Intel Core i7-2700 CPU the computational time for this step was reduced from 6990 seconds ( $\approx 1.9$  hr) to less than 46 seconds per time step. Also the CPU time for the computation of the wake induction (step 1. in Table 1) is reduced significantly, but this has less impact as it did not contribute much to the total CPU time in the conventional panel method. The solution of the system of equations by an iterative GMRES method (step 2. in Table 1) is obtained by the same number of (dense) matrix-vector multiplications in both the conventional panel method and the fast MLMIC panel method and consequently these CPU times are almost the same.

In the present fast MLMIC panel method the GMRES step contributes the most, approximately 90%, to the total CPU time of the three steps. Replacing the (dense) matrix-vector multiplications in the GMRES method by fast MLMIC integral transform evaluations will bring down the total CPU time for the fast MLMIC panel method even further.

## V. Conclusion

This paper presents the theory and the practical implementation of a fast multilevel integral transform in a computational method. We utilize this multilevel scheme in a low-order panel method. It is demonstrated that for the simulation of the wake of wind turbine rotors the computational burden is reduced from  $\mathcal{O}(N^2)$  for a conventional panel method to  $\mathcal{O}(N)$  for the present method.

From the results achieved regarding computational times and simulation accuracy we conclude that the panel method is repositioned as an excellent numerical design tool for wind turbine rotor blades in an industrial

engineering environment.

## Acknowledgments

The authors would like to thank NLR, specifically Sebastiaan ten Pas for sharing the results of the ENSOLV simulations and Jaap van Muijden and Kees Wijnberg who made the cooperation possible.

## References

- <sup>1</sup>G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.
- <sup>2</sup>J-P. Berrut and L.N. Trefethen. *Barycentric Lagrange Interpolation*. SIAM Review, 46(3):501–517, 2004.
- <sup>3</sup>H.A. Bijleveld. *Application of Quasi-Simultaneous interaction method for the determination of the aerodynamic forces on wind turbine blades*. PhD thesis, Rijksuniversiteit Groningen, 2012.
- <sup>4</sup>K. Boorsma and J.G. Schepers. *New MEXICO experiment. Preliminary overview with initial validation*. ECN-E--14-048, Energy research Centre of the Netherlands, 2014.
- <sup>5</sup>A. Brandt. *Multilevel computations of integral transforms and particle interactions with oscillatory kernels*. Computer Physics Communications, 1991.
- <sup>6</sup>A. Brandt. Unpublished letter to S. Osher, V. Rokhlin, L. Greengard, Ami Harten, 1992.
- <sup>7</sup>A. Brandt and A.A. Lubrecht. *Multilevel Matrix Multiplications and Fast Solution of Integral Equations*. Journal of Computational Physics, 90:348–370, 1990.
- <sup>8</sup>G.H. Cottet and P.D. Koumoutsakos. *Vortex Methods: Theory and Practice*. Cambridge University Press, 2000.
- <sup>9</sup>M. Drela. XFOIL. <http://web.mit.edu/drela/Public/web/xfoil/>.
- <sup>10</sup>M. Drela. *XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils*. In T.J. Mueller, editor, *Low Reynolds Number Aerodynamics*, Vol. 54 of *Lecture Notes in Engineering*, pages 1–12. Springer-Verlag, New York, 1989.
- <sup>11</sup>M. Drela. *Three-Dimensional Integral Boundary Layer Formulation for General Configurations*. AIAA Paper 2013–2437, AIAA, 2013.
- <sup>12</sup>W.F. Durand, editor. *Aerodynamic Theory*, Vol. 1. Dover Publications, 1934.
- <sup>13</sup>A. van Garrel. *Integral Boundary Layer Methods for Wind Turbine Aerodynamics*. ECN-C--04-004, Energy research Centre of the Netherlands, 2004.
- <sup>14</sup>A. van Garrel. *Development of a Wind Turbine Rotor Flow Panel Method*. ECN-E--11-071, Energy research Centre of the Netherlands, 2011.
- <sup>15</sup>A. van Garrel. *Multilevel Panel Method for Wind Turbine Rotor Flow Simulations*. PhD thesis, University of Twente, 2016.
- <sup>16</sup>H.W.M. Hoeijmakers. *Panel Methods for Aerodynamic Analysis and Design*. In Special Course on Engineering Methods in Aerodynamic Analysis and Design of Aircraft, AGARD Report R-783, 1991.
- <sup>17</sup>J. Katz and A. Plotkin. *Low-Speed Aerodynamics: From Wing Theory to Panel Methods*. Cambridge University Press, second edition, 2001.
- <sup>18</sup>J.C. Kok. *Resolving the Dependence on Freestream Values for the  $k - \omega$  Turbulence Model*. AIAA Journal, 38(7):1292–1295, 2000.
- <sup>19</sup>J.C. Kok. *A high-order low-dispersion symmetry-preserving finite-volume method for compressible flow on curvilinear grids*. Journal of Computational Physics, 228:6811–6832, 2009.
- <sup>20</sup>H. Lamb. *Hydrodynamics*. Cambridge University Press, sixth edition, 1932.
- <sup>21</sup>R.C. Lock and B.R. Williams. *Viscous-Inviscid Interactions in External Aerodynamics*. Progress in Aerospace Sciences, 24:51–171, 1987.
- <sup>22</sup>B. Maskew. *Prediction of Subsonic Aerodynamic Characteristics: A Case for Low Order Panel Methods*. Journal of Aircraft, 19(2):157–163, 1982.
- <sup>23</sup>L.M. Milne-Thomson. *Theoretical Hydrodynamics*. MacMillan & Co, fourth edition, 1962.
- <sup>24</sup>L. Morino and C.-C. Kuo. *Subsonic Potential Aerodynamics for Complex Configurations: A General Theory*. AIAA Journal, 12(2):191–197, 1974.
- <sup>25</sup>H. Özdemir. *Development of a discontinuous Galerkin method for the unsteady integral boundary layer equations*. In Euromech Fluid Mechanics Conference - 8, 2010.
- <sup>26</sup>S. ten Pas. *CFD simulations of the MEXICO wind turbine. Validating ENSOLV for wind turbine flows*. Master’s thesis, University of Twente, 2016.
- <sup>27</sup>P.G. Saffman. *Vortex Dynamics*. Cambridge Monographs on Mechanics and Applied Mathematics. Cambridge University Press, 1992.
- <sup>28</sup>B. Sandak. *Multiscale Fast Summation of Long-Range Charge and Dipolar Interactions*. Journal of Computational Chemistry, 22(7):717–731, 2001.
- <sup>29</sup>H. Snel, J.G. Schepers, and N.B. Siccama. *MEXICO Project: The Database and Results of Data Processing and Interpretation*. AIAA Paper 2009–1217, AIAA, January 2009.
- <sup>30</sup>C.H. Venner and A.A. Lubrecht. *Multilevel Methods in Lubrication*, Vol. 37 of *Tribology Series*. Elsevier Science, 2000.