A Fast Panel Code for Complex Actuator Disk Flows

Gamme, S.; de Oliveira Andrade, Gael; Ragni, Daniele; Lau, F.

DOI
10.2514/6.2017-0752

Publication date
2017

Document Version
Accepted author manuscript

Published in
55th AIAA Aerospace Sciences Meeting

Postprint of: Simon Gamme, ; Gael de Oliveira, ; Daniele Ragni, ; Fernando Lau,. "A Fast Panel Code for Complex Actuator Disk Flws", 55th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2017-0752)
http://dx.doi.org/10.2514/6.2017-0752

A Fast Panel Code for Complex Actuator Disk Flows

Simon Gamme *, Gaël de Oliveira †, Daniele Ragni ‡ Fernando Lau §

A fast, linear scaling vortex method is presented to study inviscid incompressible flow problems involving one or more actuator disks. Building upon previous efforts that were limited to axi-symmetric flow cases, the proposed methodology is able to handle arbitrary configurations with no symmetry constraints. Applications include the conceptual study of wake interaction mechanisms in wind farms, and the correction of wind tunnel blockage effects in test sections of arbitrary shape.
Actuator disks represent wind turbines through the shedding of a deformable vortex wake, discretized with a plaid of triangular distributed dipole singularities. An iterative method is adopted to align the wake with the local flow field, which is reconstructed from the vorticity field with a Green function approach.
Interactions are computed with a Fast Multipole Method (FMM), effectively overcoming the quadratic scaling of computational time associated with traditional panel methods. When compared to direct computation, the use of an FMM algorithm reduced solution time by a factor 30 when studying the wake of a single actuator disk with 60000 panels. In the same case, the mass flux of the actuator streamtube was conserved to 0.002%. Finally, the presence of round and square impermeable walls around the actuator is considered to demonstrate the code applicability to wind tunnel wall interference correction problems.

I. Introduction

Wind turbines extract energy from the wind by forming a wake of relatively lower speed flow [1, 2]. Higher values of turbulence are typically found in the decelerated flow of the wake [3], and turbines operated in the wake of each-other display greater fatigue loads coupled with reduced energy yields [4, 5, 6]. The ability to predict wake development and optimize wind farm layout [7, 8] is therefore critical to exploit scale economies when clustering machines [9, 10].

Considerable efforts have been dedicated to the simulation of wind farm flows [11, 12] and calibration of semi-analytical [11, 10] or surrogate models [13]. Numerical simulations adopt diverse representations of wind turbines, ranging from detailed meshes including blade geometry [14, 15] to simpler models with actuator line [16] and disk [17, 18] models.

Actuator disk representations of rotors lie at the heart of semi-analytical works [19, 20, 21, 22] because they strike the right balance between complexity and accuracy for conceptual studies [23, 24].

Flow surrounding actuation surfaces has been studied with a variety of numerical tools: early efforts built upon panel method era practice [25, 26, 27] whereas the last two decades staged significant progress on Reynolds Averaged Navier Stokes (RANS) codes [28, 29], Parabolized Naver-Stokes [30, 31] models and Large Eddy Simulation (LES) approaches [24, 32, 12].

Recent work by van Kuik and Lignarolo [33, 24] revived interest in steady state vortex methods. As the steady equivalent of Lagrangian vortex methods [34], steady state vortex codes discretize the vorticity field explicitly to handle discontinuous velocity fields without introducing diffusion. Vorticity is lumped on singular elements that are deformed to match the flow field [27, 35], instead of being convected as in Lagrangian vortex methods [36, 34].

Most steady state vortex models of the actuator disk represent the wake with a collection of vortex rings, as in references [26, 33, 24]. The use of vortex rings leads to small singularity counts and distinct computational advantages but limits the scope to axi-symmetric flows [24]. Most configurations of practical interest are not axi-symmetric: real flows involve interactions between multiple turbines, terrain and wind tunnel wall interference effects . The ability to analyse realistic flows with a steady state vortex method would serve the scientific community well.

This paper introduces a novel collection of steady state vortex codes applicable to arbitrary actuator disk flows. The methods rely on Fast Multipole [37, 38, 39] and FastBEM [40, 41, 42, 43] algorithms to overcome traditional computational cost issues associated with large singularity count meshes. The article starts (section II) with a review

*Msc Graduate, Aerospace Faculty, Delft University of Technology, The Netherlands gammesimon@gmail.com
†PhD Candidate, Aerospace Faculty, Delft University of Technology, The Netherlands g.l.deoliveiraandrade@tudelft.nl
‡Assistant Professor, Aerospace Faculty, Delft University of Technology, The Netherlands d.ragni@tudelft.nl
§Assistant Professor, CCTAE, IDMEC, Instituto Superior Tcnico, Universidade de Lisboa, Portugal

of the vortex wake interpretation of actuator disks from literature. Insight on wake topology is used to construct a steady state panel method for actuator disk flows in section III. The method is verified against analytical solutions while assessing its robustness and scalability. Section IV adds impermeable bodies to the flow with a FastBEM module. The module is verfified with a unit test consisting of a converging duct in free flow. Finally, section V combines the results of sections III and IV to study the effect of wind tunnel wall interference on actuator performance. Results are compared with published literature and the article concludes with a brief account of future applications and research perspectives.

II. Actuator Disk as Vorticity Generation Device

Froude [19] conceptualized the actuator disk [33] as a surface Ω_a across which fluids experience an instantaneous pressure jump $\phi = \Delta p$, which consequently creates a wake. Early studies relied on explicit mass, momentum and energy balances to predict the performance of actuator disks, as in the efforts of Rankine and Betz [20, 21].

Joukowski pioneered the interpretation of the actuator disk as a vorticity generation device in the beginning of the twentieth century [22] , but accurate potential flow solutions only appeared in the computational era [25, 44]. In this approach, the actuation surface appears as a non-conservative force term [44] in the vorticity equation .

Both approaches, pressure and vorticity based, focus on the steady flow of an inviscid incompressible fluid. The flow field $\boldsymbol{U} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is governed by a simplified form of the Euler equations [1]:

$$(\boldsymbol{U} \cdot \nabla)\, \boldsymbol{U} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\boldsymbol{f} \qquad , \qquad \nabla \cdot \boldsymbol{U} = 0$$

This particular form of the Euler equations is formally equivalent with the steady state version of the incompressible vorticity equations [1, 44]:

$$(\boldsymbol{U} \cdot \nabla)\, \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\, \boldsymbol{U} + \nabla \times \boldsymbol{f} \qquad , \qquad \nabla \cdot \boldsymbol{U} = 0 \qquad , \qquad \boldsymbol{\omega} = \nabla \times \boldsymbol{U}$$

The force field of a flat actuator disk centered on the origin with radius r_a and constant normal loading $\phi \boldsymbol{e}_z$ aligned with the free-stream $\boldsymbol{U}_o = U_o \boldsymbol{e}_z$ can be described with a simple combination of Dirac delta and Heavyside step functions:

$$\boldsymbol{f} = \phi \delta_{(z-z_a)} H_{(r-r_a)} \boldsymbol{e}_z$$

The curl of the force field, appearing in the vorticity equation, is written in a cylindrical coordinate system using the azimuthal unit vector $\boldsymbol{e}_\theta$:

$$\nabla \times \boldsymbol{f} = -\frac{\partial f_z}{\partial r}\boldsymbol{e}_\theta = -\phi \delta_{(z-z_a)} \delta_{(r-r_a)} \boldsymbol{e}_\theta$$

The actuator disk induces an axi-symmetric forcing on the flow, but the presence of other bodies or actuators can break flow axi-symmetry. The latter would only affect wake development and therefore actuator performance, whereas the strength of vorticity generation remains unchanged. Since this vorticity strength is essentially decoupled from external perturbations, the flow will be treated axi-symmetrically for the rest of this section, in view of clarity.

In pure axi-symmetric flow there is no swirl , $U_\theta = 0$, and the vorticity vector is aligned with the azimuthal unit vector $\boldsymbol{e}_\theta$. A single scalar field $\omega : (r, z) \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ suffices to describe the vorticity field $\boldsymbol{\omega} = \omega \boldsymbol{e}_\theta$, the vortex stretching term $(\boldsymbol{\omega} \cdot \nabla)\, \boldsymbol{U}$ vanishes and the vorticity equation collapses into a scalar transport equation:

$$(\boldsymbol{U} \cdot \nabla)\, \omega = \nabla \times \boldsymbol{f} \qquad , \qquad \nabla \cdot \boldsymbol{U} = 0 \qquad , \qquad \omega = (\nabla \times \boldsymbol{U}) \cdot \boldsymbol{e}_\theta$$

As in all scalar transport problems, the convective derivative can be reworked into the projection of the scalar gradient $\nabla\omega$ on the flow vector:

$$(\boldsymbol{U} \cdot \nabla)\, \omega = \left(\frac{U_r}{r}\frac{\partial}{\partial r} r + \frac{U_\theta}{r}\frac{\partial}{\partial \theta} + U_z \frac{\partial}{\partial z} \right) \omega = \begin{bmatrix} U_r & U_\theta & U_z \end{bmatrix} \begin{bmatrix} \frac{1}{r}\frac{\partial (r\omega)}{\partial r} \\ \frac{1}{r}\frac{\partial \omega}{\partial \theta} \\ \frac{\partial \omega}{\partial z} \end{bmatrix} = (\nabla \omega) \cdot \boldsymbol{U}$$

Through which the vorticity transport equation can be restated as:

$$(\nabla \omega) \cdot \frac{\boldsymbol{U}}{|\boldsymbol{U}|} = -\frac{1}{\rho\,|\boldsymbol{U}|}\frac{\partial f_z}{\partial r} = -\frac{\phi}{\rho\,|\boldsymbol{U}|}\delta_{(z-z_a)}\delta_{(r-r_a)}$$

To see where vorticity is generated, a streamline segment $\sigma \subset \mathbb{R}^3$ is considered which crosses the actuator boundary, at some point $\boldsymbol{x}_a$:

$$\sigma = \left\{ \boldsymbol{x} \in \mathbb{R}^3 \ : \ \boldsymbol{x}_{(\gamma)} = \boldsymbol{x}_a + \int \left(\frac{\boldsymbol{U}}{|\boldsymbol{U}|} \right) \Big|_{\boldsymbol{x}_{(\bar{\gamma})}} d\bar{\gamma} \right\}$$

Finally, integrate the vorticity transport equation along σ. Use the fundamental theorem of multivariate calculus by noticing that $\frac{\boldsymbol{U}}{|\boldsymbol{U}|}$ is aligned with the tangent unit vector of σ:

$$\int (\nabla \omega) \cdot \frac{\boldsymbol{U}}{|\boldsymbol{U}|} d\sigma = \int -\frac{1}{\rho \, |\boldsymbol{U}|} \frac{\partial f_z}{\partial r} d\sigma = \int -\frac{\phi}{\rho \, |\boldsymbol{U}|} \delta_{(z-z_a)} \delta_{(r-r_a)} d\sigma$$

$$\Rightarrow \qquad (\omega)|_{\partial\sigma} = \frac{\phi}{\rho \left| \boldsymbol{U}_{(\boldsymbol{x}_a)} \right|} \qquad , \quad \boldsymbol{x}_a = (\theta, r_a, z_a) \quad \forall \, \theta \in [0, 2\pi]$$

Showing that vorticity is generated at the actuator edges, with a strenght that is directly proportional to the actuator loading and inversely proportional to the velocity magnitude on the actuator edge. The vorticity is then transported along the wake, following the flow streamlines.

In order to follow the development of the wake, it is necessary to determine the flow field, by reconstructing it from the vorticity field.

III. Numerical solution of Actuator Disk Flow

Wakes of lightly loaded actuators [2, 45] share a universal topology consisting of a tubular sheet of vorticity aligned with flow streamlines (section II). The strength of the wake is known at the actuator edges, but there are no general analytical solutions for wake shape and vorticity distribution when multiple objects are present in the flow.

Knowledge about wake topology enables the construction of a non-diffusive numerical method for solving inviscid actuator disk flows. The general approach consists in parametrizing a manifold meant to describe the wake shape, formulating a first guess for its location and then deforming it until it matches the flow streamlines.

A. Vorticity field discretization

Inviscid actuator disk flows are non-diffusive. The vorticity field $\boldsymbol{\omega} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is therefore null everywhere but on a restricted set of points that form a 2-manifold $P^\omega \subset \mathbb{R}^3$ covering the wake(s). It is therefore possible to represent the vorticity field completely in terms of a function $f^{P\omega}$ defined over the wakes manifold P^ω .

$$\boldsymbol{\omega}_{(\boldsymbol{x})} = \begin{cases} f^{P\omega}_{(\boldsymbol{x})} & \boldsymbol{x} \in P^\omega \\ 0 & \boldsymbol{x} \notin P^\omega \end{cases} \qquad with \quad f^{P\omega} : \psi \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

The support of the vorticity field P^ω is discretized with a structured mesh of quadrilateral panels P_i like the ones shown on figure 1 for an expanding wake. The first guess consists of a straight tubular wake extending behind the actuator for several diameters, before connecting to a straight quasi-infinite vortex tube. Panels are clustered in areas where change occurs to capture streamlines accurately and a consistent normal direction is defined for all panels, taken outward in this case. Vorticity is lumped over constant strength vortex rings fitted to panel edges ∂P_i, as in classic

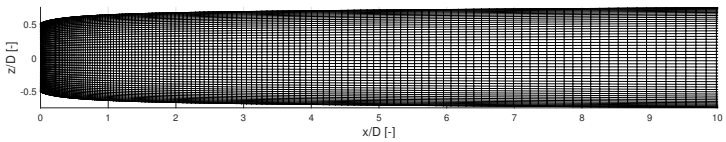


Figure 1. Vorticity panel distribution on an expanding wake

low order panel methods [27, 46]. This approximation strategy leads to discontinuous velocity and vorticity fields, but yields distinct computational advantages [47] related with the equivalence between vortex rings and continuous dipole patches [27].

The problem of finding the strength and position of each panel is non-linear. The correct orientation of each panel depends on the velocity field which itself depends on the position and strength of all panels. This means an iterative method is needed to find the shape of the wake, as discussed in section D. However this first requires a procedure to relate the hypothetical vorticity distributions to a velocity field.

B. Reconstruction of flow field

Several approaches for reconstructing the velocity field from the vorticity distribution exist. Most modern vortex methods [34] build upon the Helmholtz decomposition of the velocity field to formulate a set of Poisson equations for the scalar ϕ and vector $\boldsymbol{\psi}$ potentials. Approaches differ in the way the Poisson equations are solved, with FFT [48, 49] and Green function [25, 26, 27, 35] strategies enjoying the greatest popularity.

We opt for a Green function approach to ease the handling of discontinuous velocity fields with explicit free-space boundary conditions. The formulation of the Poisson equations is briefly restated to shed light on the present method's applicability and computational complexity. The exposition starts by noting that, as any vector field in $\mathbb{R}^3$, the velocity field $\boldsymbol{U} : \mathbb{R}^3 \to \mathbb{R}^3$ can be represented with a Helmholtz decomposition:

$$\exists \begin{array}{l} \phi : \mathbb{R}^3 \to \mathbb{R} \\ \boldsymbol{\psi} : \mathbb{R}^3 \to \mathbb{R}^3 \end{array} \quad : \quad \boldsymbol{U} = \nabla\phi + \nabla \times \boldsymbol{\psi} \tag{1}$$

Building upon a well known identity for the divergence of a curl, the decomposed velocity field (1) transforms the continuity equation into a Poisson equation for the scalar potential:

$$\begin{aligned} \nabla \cdot \boldsymbol{U} = \nabla \cdot (\nabla\phi + \nabla \times \boldsymbol{\psi}) = \nabla \cdot (\nabla\phi) + \cancel{\nabla \cdot (\nabla \times \boldsymbol{\psi})} = \Delta\phi \\ \Rightarrow \qquad \nabla \cdot \boldsymbol{U} = q_{(\boldsymbol{x})} \quad \Leftrightarrow \quad \Delta\phi = q_{(\boldsymbol{x})} \end{aligned} \tag{2}$$

The Laplace operator is linear and its Green function for tri-dimensional free-space is well known [34, 50]. A standard representation of the solution of equation 2 consists in the convolution of the Green function $G^{3d} : \mathbb{R}^{3\times3} \to \mathbb{R}$ with the inhomogeneous forcing $q : \mathbb{R}^3 \to \mathbb{R}$:

$$\phi_{(\boldsymbol{x})} = G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} \star q_{(\tilde{\boldsymbol{x}})} = \int_{\mathbb{R}^3} G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} q_{(\tilde{\boldsymbol{x}})} d\tilde{\boldsymbol{x}} \qquad with \quad G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} = \frac{1}{4\pi} \frac{1}{|\boldsymbol{x} - \tilde{\boldsymbol{x}}|}$$

The forcing field $q_{(\boldsymbol{x})}$ represents the distribution of sources or sinks over the entire space. It vanishes everywhere but on specific 2-manifolds like body surfaces. Like wakes, these manifolds are plaided with panels $P_i \subset \mathbb{R}^3$ whose constant source strength q_i may or may not be null. Panel contributions to the scalar potential can be highlighted with an interaction function $K^\phi : \left(\mathbb{R}^3, \mathcal{P}\left(\mathbb{R}^3\right)\right) \to \mathbb{R}$, as suggested in reference [50]:

$$\phi_{(\boldsymbol{x})} = \sum_i K^\phi_{(\boldsymbol{x},P_i)} q_i \qquad with \quad K^\phi_{(\boldsymbol{x},P_i)} = \int_{P_i} G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} q_{(\tilde{\boldsymbol{x}})} d\tilde{\boldsymbol{x}} = \int_{P_i} G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} q_i d\tilde{\boldsymbol{x}} \tag{3}$$

The scalar potential depends solely on the source distribution and its contribution to the velocity has no effect on flow vorticity. The vorticity definition can be reworked with the Helmholtz decomposition (1) to obtain a different set of Poisson equations:

$$\begin{aligned} \boldsymbol{\omega} = \nabla \times \boldsymbol{U} \quad &= \cancel{\nabla \times (\nabla\phi)} + \nabla \times (\nabla \times \boldsymbol{\psi}) \qquad && \forall \phi^\psi : \mathbb{R}^3 \to \mathbb{R} \\ &= \nabla (\nabla \cdot \boldsymbol{\psi}) - \Delta\boldsymbol{\psi} && \phi^\psi \in C^2 \;\Rightarrow\; \nabla \times \boldsymbol{\psi} = \nabla \times \left(\boldsymbol{\psi} + \nabla\phi^\psi\right) \\ \Rightarrow \quad \Delta\boldsymbol{\psi} = \boldsymbol{\omega}_{(\boldsymbol{x})} \end{aligned} \tag{4}$$

In the above manipulation, the vector potential was chosen to be divergence free without loss of generality regarding the variety of velocity fields that can be represented. Solutions of equation 4 are written by convoluting the vorticity field with the Green function:

$$\boldsymbol{\psi}_{(\boldsymbol{x})} = G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} \star \boldsymbol{\omega}_{(\tilde{\boldsymbol{x}})} = \int_{\mathbb{R}^3} G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} \boldsymbol{\omega}_{(\tilde{\boldsymbol{x}})} d\tilde{\boldsymbol{x}}$$

The convolution integral for the streamfunction can be parted to highlight contributions from vorticity associated with individual panels. As explained in the previous section (A) , each panel P_i supports a ring of vorticity lumped along its border ∂P_i with strength γ_i aligned with the tangent unit vector $\boldsymbol{r}_{\partial P_i}$ of the panel edges. An interaction function $\boldsymbol{K}^\psi : \left(\mathbb{R}^3, \mathcal{P}\left(\mathbb{R}^3\right)\right) \to \mathbb{R}^3$ is defined for clarity:

$$\boldsymbol{\psi}_{(\boldsymbol{x})} = \sum_i \boldsymbol{K}^\psi_{(\boldsymbol{x},P_i)} \gamma_i \qquad with \quad \boldsymbol{K}^\psi_{(\boldsymbol{x},P_i)} = \int_{P_i} G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} \frac{\boldsymbol{\omega}}{\gamma_i} d\tilde{\boldsymbol{x}} = \int_{\partial P_i} G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} \boldsymbol{r}_{\partial P_i} d\tilde{\boldsymbol{x}} \tag{5}$$

The solutions of the Poisson equations (3 and 5) serve as inputs for reconstructing the decomposed velocity field (1) from the source strength q and vorticity $\boldsymbol{\omega}$ fields.

$$\boldsymbol{U}_{(\boldsymbol{x})} = \nabla \left(G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} \star q_{(\tilde{\boldsymbol{x}})} \right) + \nabla \times \left(G^{3d}_{(\boldsymbol{x},\tilde{\boldsymbol{x}})} \star \boldsymbol{\omega}_{(\tilde{\boldsymbol{x}})} \right)$$

$$= \sum_i \left(q_i \nabla K^{\phi}_{(\boldsymbol{x},P_i)} \right) + \sum_i \left(\gamma_i \nabla \times \boldsymbol{K}^{\psi}_{(\boldsymbol{x},P_i)} \right)$$

Velocity contributions from individual singularities are best interpreted in terms of two additional interaction functions, $K^{\phi U} : \left(\mathbb{R}^3, \mathcal{P}\left(\mathbb{R}^3\right)\right) \to \mathbb{R}^3$ and $\boldsymbol{K}^{\psi U} : \left(\mathbb{R}^3, \mathcal{P}\left(\mathbb{R}^3\right)\right) \to \mathbb{R}^3$, representing the induction from source sheets or vortex rings fitted to the input set.

$$\boldsymbol{U}_{(\boldsymbol{x})} = \sum_i \left(q_i \boldsymbol{K}^{\phi U}_{(\boldsymbol{x},P_i)} \right) + \sum_i \left(\gamma_i \boldsymbol{K}^{\psi U}_{(\boldsymbol{x},P_i)} \right) \qquad with \quad \begin{cases} K^{\phi}_{(\boldsymbol{x},P_i)} = \nabla K^{\phi}_{(\boldsymbol{x},P_i)} \\ \boldsymbol{K}^{\psi U}_{(\boldsymbol{x},P_i)} = \nabla \times \boldsymbol{K}^{\psi}_{(\boldsymbol{x},P_i)} \end{cases}$$

C. Fast Evaluation of Velocity Field

The iterative procedure for approximating the shape of the vorticity support manifold requires that the velocity $\boldsymbol{U}_j$ be evaluated at the center of each panel $\boldsymbol{x}_j$:

$$\boldsymbol{U}_j = \boldsymbol{U}_{(\boldsymbol{x}_j)} = \sum_i \left(q_i \boldsymbol{K}^{\phi U}_{(\boldsymbol{x}_j,P_i)} \right) + \sum_i \left(\gamma_i \boldsymbol{K}^{\psi U}_{(\boldsymbol{x}_j,P_i)} \right) \qquad with \quad \boldsymbol{x}_j = \frac{\int_{P_j} \boldsymbol{x}\, d\boldsymbol{x}}{\int_{P_j} d\boldsymbol{x}}$$

The velocity ought to be computed on each of the N panels, and each computation requires that the influence of all N panels be taken into account. The velocity reconstruction operation scales quadratically $\mathcal{O}(N^2)$ with the number of panels, acting as the computational bottleneck of direct Green function based vortex methods [34].

Greengard and Rokhlin [37] proposed the Fast Multipole Method (FMM) algorithm to rapidly evaluate potential and force fields involving a large number of particles that interact with each other. The FMM algorithm relies on the hierarchic subdivision of 3D space in cubical boxes. Particles within these boxes are clustered to construct multipole expansions. Finally, the potential field and its derivatives (gradient, hessian) are evaluated from the expansions, which are truncated to obtain a certain precision.

Asymptotic CPU time estimates can be reduced from order $\mathcal{O}(N)$ to $\mathcal{O}(N^2)$ using FMM algorithms. The method became a well established tool in plasma physics, molecular dynamics, fluid dynamics and celestial mechanics. Still, to the best of the authors knowledge, no steady state FMM-powered codes for actuator disk flows were published before the present contribution.

The current implementation relies on the libFMM library developed by Greengard and Gimbutas [39]. The library handles diverse singularity support geometries (particles and triangles) for both the Laplace and Helmholtz equations. For the Laplace equation, triangle singularities can be of either sheet source or sheet dipole types.

In the present method, libFMM is used by constructing triangle pairs to represent quadrilateral panels P_i. Source interactions $\boldsymbol{K}^{\phi U}_{(\boldsymbol{x},P_i)}$ are obtained directly from the gradient of the scalar potential, whereas vortex interactions $\boldsymbol{K}^{\psi U}_{(\boldsymbol{x},P_i)}$ are computed by exploiting the equivalence between quadrilateral dipole sheets and fitted vortex rings [27].

D. Adaptation of Vorticity Field

Wakes cannot sustain forces, and must therefore be aligned with the local flow field iteratively. The algorithm for determining the wake shape is illustrated on figure 2: the normal velocity on each panel is evaluated at every iteration, and panels are moved to align with flow streamlines with an artificial time step. The procedure is simple from a

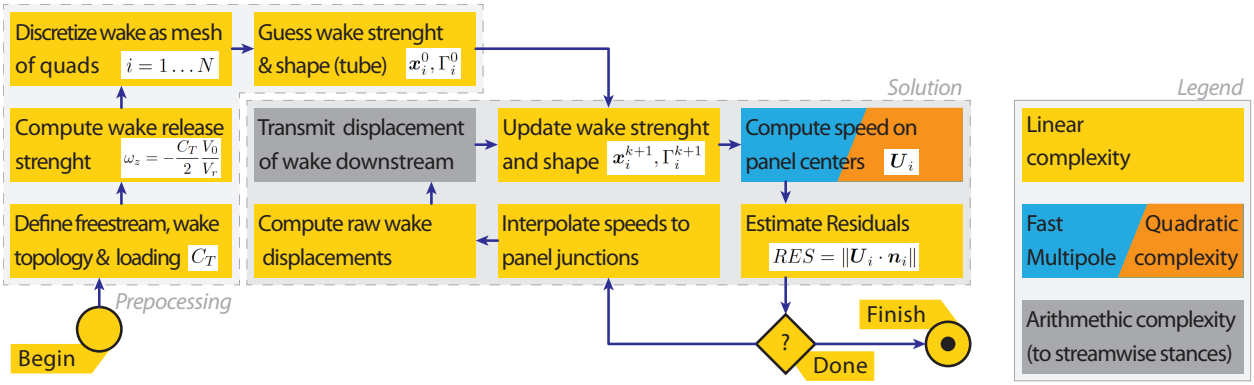


Figure 2. Algorithm for adaptive wake method (FMM vs Direct)

conceptual standpoint but the robustness of the method depends on the careful selection of evaluation points and re-interpolation strategies. This is specially true since the present method does not resort to any form of desingularization [34], unlike most recent steady state vortex methods for actuator disks [33, 24]. Avoiding desingularization enables the representation of strictly un-diffused velocity fields.

The details of the wake adaptation steps conducted on each iteration are illustrated with two arbitrary adjacent wake panels like those shown in figure 3. The center velocities V_1 and V_2; and axis systems (t_1, m_1, n_1) and (t_2, m_2, n_2)

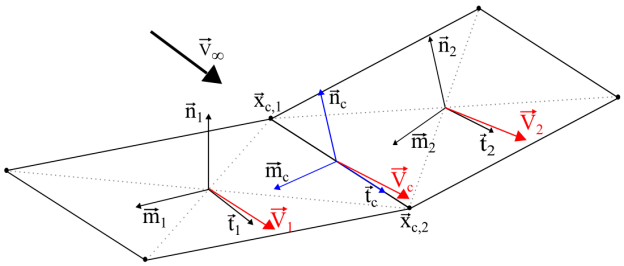


Figure 3. Wake alignment nomenclature

are interpolated to the center of the joining edge, yielding V_c and (t_c, m_c, n_c). Then the displacement is calculated by:

$$\vec{\delta} = \vec{x_{c,1}} + \|\vec{x_{c,2}} - \vec{x_{c,1}}\| \cdot \frac{\vec{V_c}}{\|\vec{V_c}\|} - \vec{x_{c,2}} \tag{6}$$

This displacement $\vec{\delta}$ can be projected on the plane spanned by $\vec{m}_c$ and $\vec{n}_c$ to get $\vec{\delta}_d$. Finally $\vec{x}_{c,2}$ and all the points downstream of that spanwise instance are displaced with $\beta\vec{\delta}_d$, with β being a preset constant for the magnitude of artificial time steps. The resulting panel code structure is shown in figure 2. Important to note is that the FMM can be introduced at the velocity calculation on the panel centers, which reduces the original quadratic scaling.

E. Results and Validation

The method is assessed by studying an actuator disk operated in wind turbine mode with a C_T of $8/9$, as in the Betz limit case. The study starts with a review of convergence trends and proceeds to compare results with analytical solutions from momentum theory. A discussion on the effectiveness of the FMM for reducing computational times closes method validation.

1. Convergence

Wake convergence is monitored with two residuals, the wake residual, related to the normal velocity trough the center of each wake panel, and the solver residual, based on the wake displacement at each iteration step.

$$RES^{wake} = \sqrt{\sum \left(\boldsymbol{U}_i \cdot \boldsymbol{n}_i\right)^2}$$

For the coarse discretization case, the wake residual converges to $O(10^{-7})$ for a wide range of artificial time step values. In the long run, residuals can increase again as numerical noise [51] excites unstable modes on the vortex sheet. This phenomenon is consistent with Rosenhead's considerations on the growth of instabilities on surfaces of discontinuity [52] and Caflish's discussion of the ill-posedness of the vortex-sheet problem [53]. Still, the growth of unstable modes is not problematic unless very small residuals (of no practical importance) are sought.

The value of the smallest reached residual was found to be primarily determined by the spatial discretization of the wake, as shown in figure 4. The use of variable density discretizations further improves the residual, as different regions of the wake react to different wavelenghts in this case, and no single simple mode can destabilize the solution. The adoption of a variable density mesh is a meaningful choice from every perspective, as the region of rapid expansion near the disk requires a fine panelling to allow wake alignment with local streamlines.

The FMM precision of 10^{-9} imposes a lower bound on the solver residual in the $O(10^{-10})$ range. At this point, induced velocities and their according displacements are random since they fall below FMM precision. Another source of numerical noise comes from the floating point accuracy, as large sums can generate numerical noise with larger order of magnitude than floating point accuracy [51].

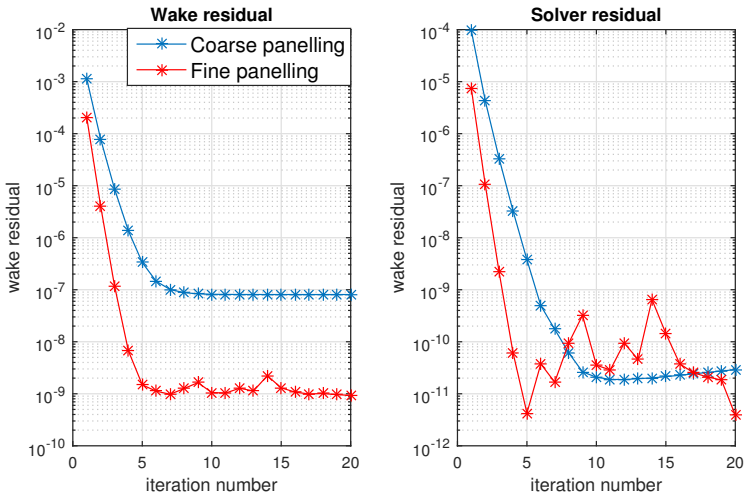


Figure 4. Wake and solver residual for coarse and fine panelling (FMM precision 10^{-9})

A fine panelling will capture higher frequency noise, which are instabilities with small wavelengths. This results in a higher fluctuation of the solver residual for finer panellings, as seen in figure 4.

Still, it is particularly encouraging to observe that the number of iterations needed to reach a converged residual seems to be rather independent from the number of panels in the wake. The convergence rate even seems to increase for finer panellings, meaning that an increase in the number of panels only affects computational times by increasing the time spent on each artificial time step, without affecting the number of required iterations.

This feature is one of the present method's greatest advantages, as it means that the computational time needed for reaching a solution will scale with the time spent on each artificial time step, which essentially depends on the performance of the FMM algorithm.

2. *Validation against Momentum Theory*

The velocity field with the converged wake can be compared to 1-D momentum theory. The cross-sectional velocities for an ideal rotor with a C_T of $8/9$ is shown in figure 5 From the in plane velocity of figure 5 it can be observed that

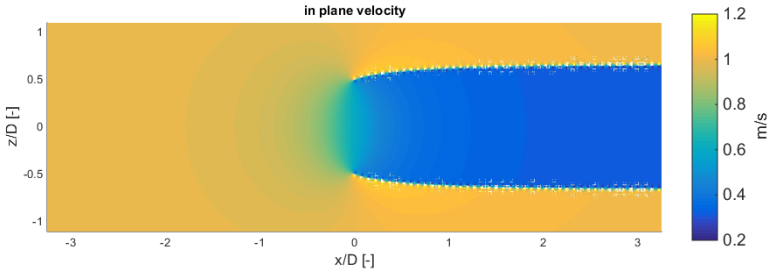


Figure 5. In plane velocities for a 3D actuator disk with $C_T = 8/9$

the velocity downstream stabilizes approximately two diameters behind the disk. The velocity along the centerline is plotted in figure 6, showing a converging end velocity. Refining the mesh to $100 \cdot 10^3$ did influence the velocity field by less than one percent, hence the apparent single line. An induction of $1/3$ corresponds to a theoretical non-dimensional end velocity of 1/3, indicated by the dashed line. Momentum theory predicts the average velocity across the cross-section while figure 6 shows the velocity along the centerline. That is why the end velocity appears higher than the theoretical value on figure 6. Velocity profiles are plotted on figure 7 and exhibit good agreement with the theoretical solution: integrated velocities over the disk lead to a power coefficient prediction of $C_P = 0.5911$, which is 0.25% lower than the Betz limit ($C_P = 0.5926$). The wake expands to conserve mass while the flow is slowed down. This mass flux fluctuation in percent with respect to the mass flux at the disk is plotted in figure 8. The coarse mesh leads to to large fluctuations in mass flux since the expansion is hard to capture. Refining the mesh reduces the

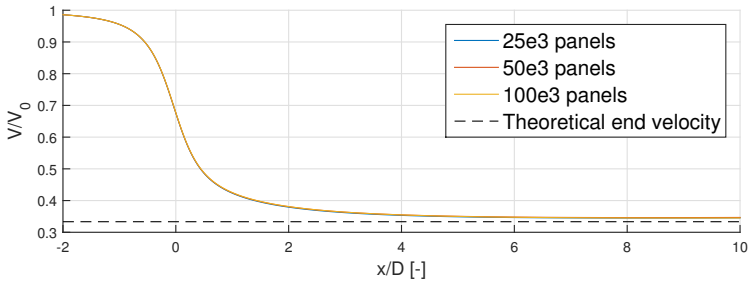
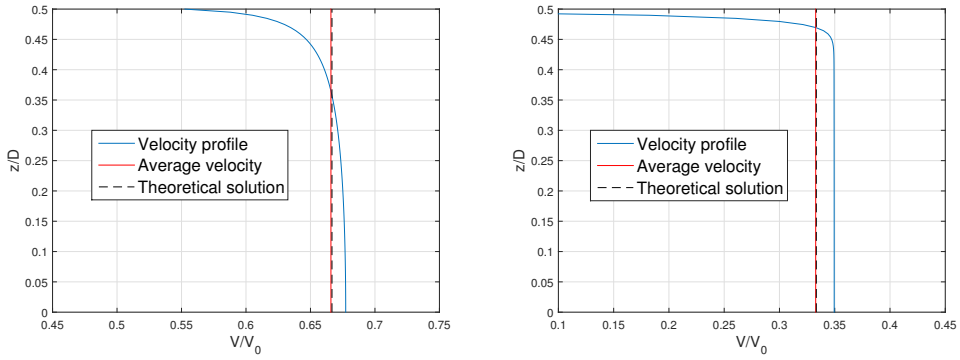


Figure 6. Variation of velocity along centerline



a) Disk **b) 5 diameters downstream**

Figure 7. Velocity profiles for an ideal rotor

maximum difference in mass flux from 2% to 0.2%. For the refined case the average mass flux fluctuation is 0.002%, this corresponds to a wake residual of $\mathcal{O}(10^{-11})$. Refining more does not lower these quantities any further due to numerical noise. However this is considered as an adequate mass convergence given a mass flow through the wake boundary of 0.4% by a reference solver [33].

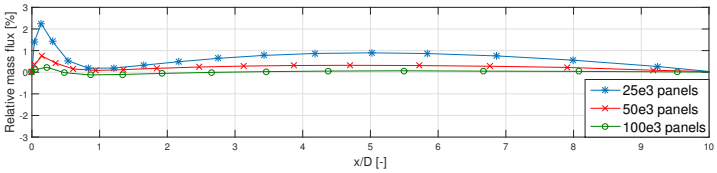


Figure 8. Mass conservation through the expanding wake

3. Fast Multipole Method Timings

The present method is meant to demonstrate the use of FMM algorithms for speeding up the iterative wake adaptation procedure. As shown in section 1 the FMM did not affect the convergence process unless very low wake residuals were sought. Only then could FMM inacurracies cause the wake residual to stagnate. In general, the ability to accelerate calculations at each iteration reduces the time required for reaching a converged wake. Iteration timings are shown for various panel counts on figure 9.

There is no advantage of using the FMM for low panel numbers due to the methods initialization time. But the slope of the iteration time curve for the direct method is significantly larger than for the FMM cases. For 60000 panels for example, the direct method is 30 times slower than the FMM with a precision of 10^{-9}. Extrapolating the direct method to a panel count of 100000 shows that the Fast panel code would be roughly 100 times faster for this FMM precision.

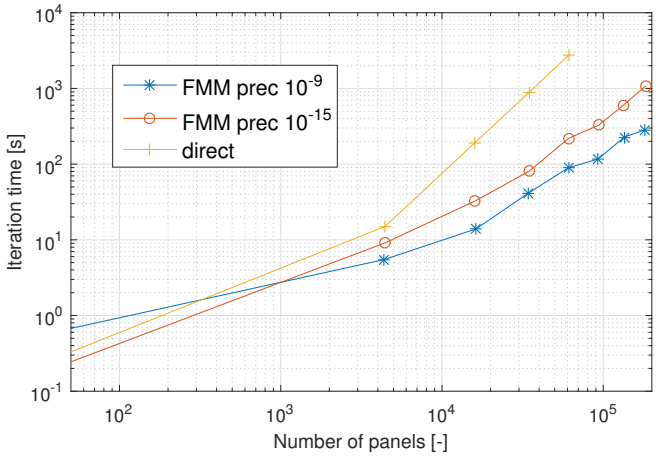


Figure 9. Calculation time for each iteration step (direct vs. FMM)

IV. Numerical Solution of Flow with Bodies

The force-free boundary condition governing actuator disk wakes can be satisfied implicitly satisfied by displacing the wake. However, many flows also require the enforcement of explicit impermeability conditions on fixed geometries. Examples include the study of actuators in the presence of nearby bodies like ducts, wind tunnels or wings.

A. Body discretization and boundary conditions

Different types of bodies call for different types of discretization and singularity distributions. The proposed method focuses on infinitesimally thin, duct-like bodies but the general ideas apply to arbitrary types of bodies. Infinitesimally thin bodies can be discretized in a way that is very similar to the wake of an actuator disk. A vorticity sheet is laid over the surface by means of quadrilateral dipole panels whose role is equivalent to vortex rings. A collocation point is defined at the center of each panel $\boldsymbol{x}_i$, where impermeability is enforced. The velocity at this point is composed of the self influence of the panel, the induced velocity due to other panels in the flow and any externally enforced flow velocity.

B. Linear Problem for Lift and Impermeability

The sum of aforementioned velocities should equal zero in the direction of the panel normal. This is only true for a specific set of vortex strengths, which are initially unknown. For a body approximated with N panels, the unknown dipole strengths can be found by solving a system of N linear equations with N unknowns:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ a_{31} & a_{32} & \cdots & a_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \cdots \\ \mu_m \end{bmatrix} = \begin{bmatrix} \mathrm{RHS}_1 \\ \mathrm{RHS}_2 \\ \mathrm{RHS}_3 \\ \cdots \\ \mathrm{RHS}_m \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{A} \cdot \boldsymbol{\mu} = \mathbf{RHS} \tag{7}$$

The left side of figure 7 contains the induced velocity of each dipole panel with unknown strength at each collocation point. The right hand side contains influence of the known flow strengths such as actuator wakes and free streams. Setting up the influence matrix $\boldsymbol{A}$ requires N influence function evaluations for N collocation points, leading to a computational complexity that scales quadratically with the number of panels.

Figure 10 shows the archetypal structure of classical panel method, highlighting that all steps before the matrix calculating scale linearly. The last step, solving the linear system, exhibits a higher asymptotic computational complexity that depends on the selected linear solver. The variable of interest is not the matrix itself but rather the vorticity distribution. The vorticity distribution can be obtained by solving the system for a specific forcing vector or finding the inverse matrix $\boldsymbol{A}$ with a variety of forcing vectors ($\mathbf{RHS}$) of interest. During postprocessing, the vorticity distribution can be used to evaluate the flow field at any point of interest.

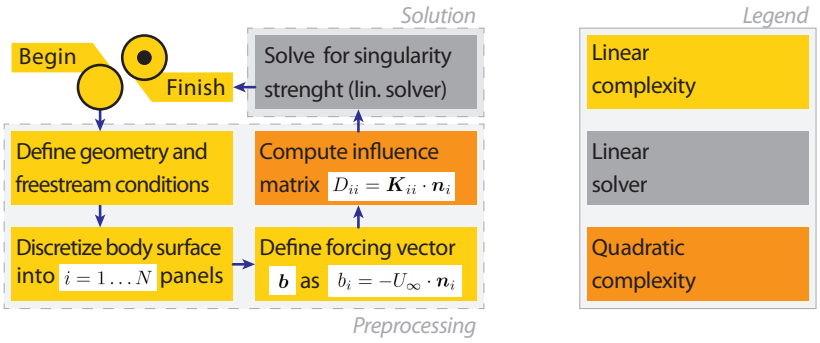


Figure 10. Structure of a conventional panel method

C. Fast Solution of Linear Problem

The explicit generation of the influence matrix exhibits quadractic computional complexity, often leading to prohibitive computation times when large panel counts are considered [54, 55, 46]. The FMM cannot be used to set-up the influence matrix $\boldsymbol{A}$ explicitly since individual contributions from each panel are needed while FMM combines the influence of distanced panels. The FMM can however be weaved into an iterative linear solver to construct a FastBEM method [41, 42, 43]. The Fast Boundary Element Method (FastBEM) gained popularity in electromagnetic applications and recently received increased interest from the aerodynamic panel code community [56, 35].

FastBEM algorithms avoid the N^2 cost of setting-up the A matrix and some of the higher order costs associated with its solution. The method approximates solutions of the linear problem without generating the influence matrix $\boldsymbol{A}$ explicitly.

The principle is best demonstrated by weaving the FMM with a stationary linear solver like the Jacobi algorithm [57]. The Jacobi algorithm solves diagonally dominant sets of linear equations iteratively by departing from a first guess. For our system of equations (7), the k-th iterative step would be defined according to equation 8.

$$\boldsymbol{\mu}^{k+1} = D^{-1}(\mathbf{RHS} - R\boldsymbol{\mu}^k) \tag{8}$$

D denotes the diagonal matrix of A, and R is the remainder such that $A = D + R$. Equation 8 can be rewritten in terms of the diagonal matrix D and a matrix product $A\boldsymbol{\mu}$:

$$\boldsymbol{\mu}^{k+1} = D^{-1}(\mathbf{RHS} - A\boldsymbol{\mu}^k + D\boldsymbol{\mu}^k) \tag{9}$$

The $A\boldsymbol{\mu}$ product corresponds to the evaluation of the flow field for some prescribed μ, and it can be calculated with linear computational complexity using the FMM. The generation of the diagonal matrix (D) is also a linear complexity operation: diagonal entries represent the self influence of panels on their own collocation point.

Figure 11 describes an hypothetical FastBEM panel method based on a stationary solver like the Jacobi algorithm. No operations with explicit quadratic complexity are required. In principle, the system would be solved iteratively without defining the A matrix by departing from a first guess $\boldsymbol{\mu}^{k+1}$.

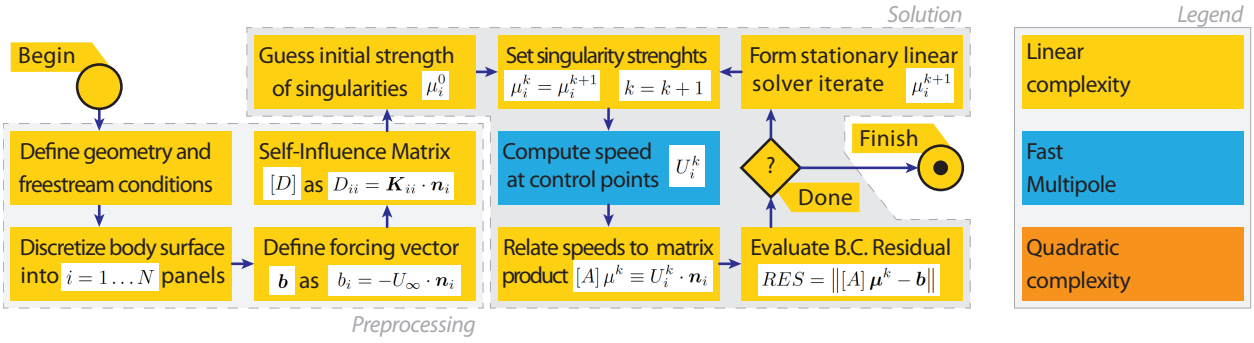


Figure 11. Structure of a stationary FastBEM panel method

The effectiveness of the FastBEM method depends on the linear solver's ability to converge. Unfortunately the Jacobi algorithm is only guaranteed to converge for diagonally dominant systems. Since many flow problems do not

exhibit these features in the absence of very careful panel ordering strategies [35], extra care should be given to less restrictive iterative linear solvers for successful practical applications.

The Generalised Minimised Residual (GMRES) linear solver [58, 57] exhibits robust convergence for very diverse systems, waiving most restrictions associated with stationary solvers. A variety of successful applications based on the GMRES solver have been reported [41, 42, 43].

As an instationary method, GMRES is far more complex than the Jacobi algorithm, and its weaving with the FMM is summarized in the Appendix. The implementation was verified with a 2D demonstrator combining the FMM with the GMRES algorithm. In figure 12 it can be seen that for 1000 panels, the linear solver residual reached 10^{-9} after 6 seconds while the direct calculation took about 17 times longer.

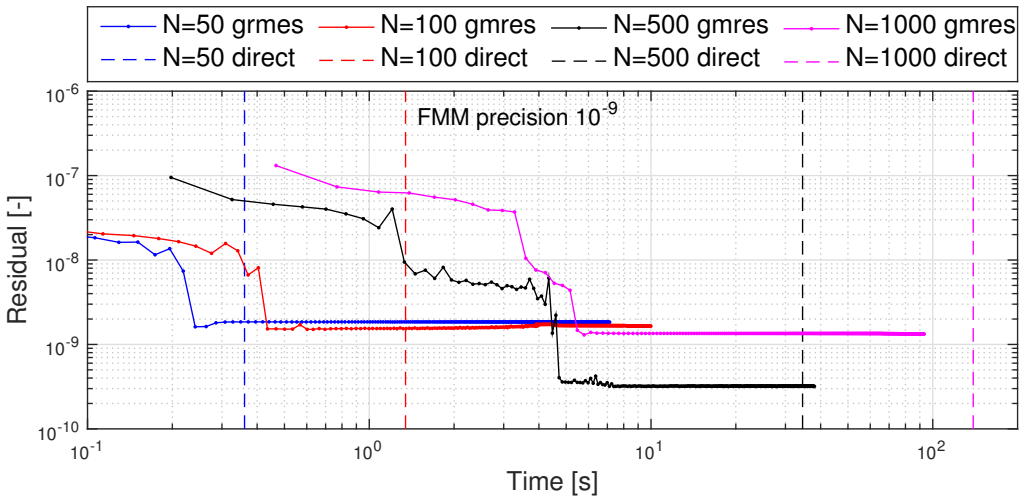


Figure 12. Calculation time for the FastBEM algorithm w.r.t. direct evaluation

FMM precision and preconditioning influence the convergence behaviour to a large extent, as noted in previous literature [41, 56, 35]. A thorough mathematical analysis would be needed to determine the optimal preconditioning strategy.

D. Results and validation

The ability to enforce explicit impermeability conditions is validated by modelling a converging duct: The duct inlet radius is twice the exit diameter, and the radius follows a quadratic curvature. The in plane velocities are shown in figure 13.

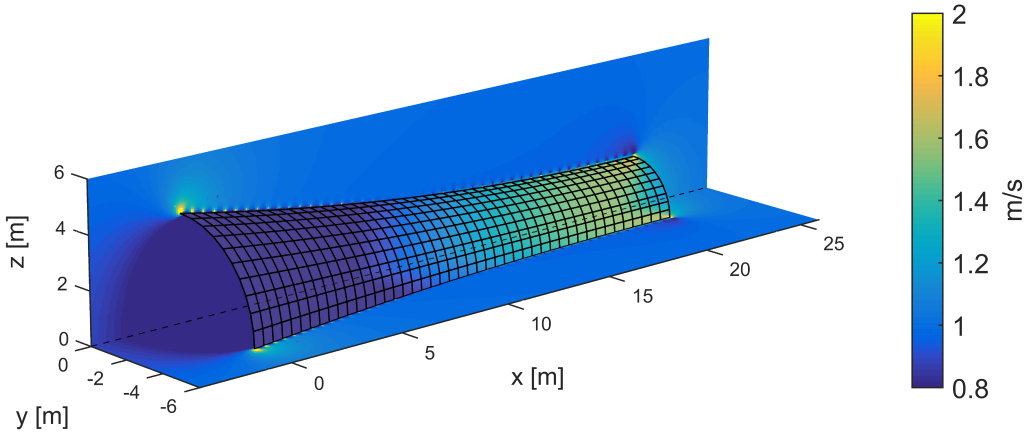


Figure 13. In plane velocity of the nozzle

The normal velocities at each collocation point is $\mathcal{O}(10^{-10})$ when calculated with the FMM. Direct evaluation gives normal velocities of $\mathcal{O}(10^{-15})$, which is machine precision as expected. This does however not imply that no flow is going through the walls at all. In the vicinity of panel edges for example, this boundary condition is not defined explicitly and cross flow might occur. Near the panel edges high velocities can be observed which follow from the discontinuous vorticity sheet.

Furthermore one can observe high velocities near the inlet and outlet edge of the duct since they are at an incidence w.r.t. the free stream. Similar to the leading edge of a flat plate, the flow circumvents the duct's infinitesimal edge with infinite curvature compensated by locally singular velocity gradients.

Theoretically, the integrated velocities scale with the square of the radius ratio to conserve mass. These are represented by the red dashed line in figure 14. Important to note is the converging behaviour for increased panel counts since since this decreases the gradient strength, thus the large velocities at the edge of each panel.

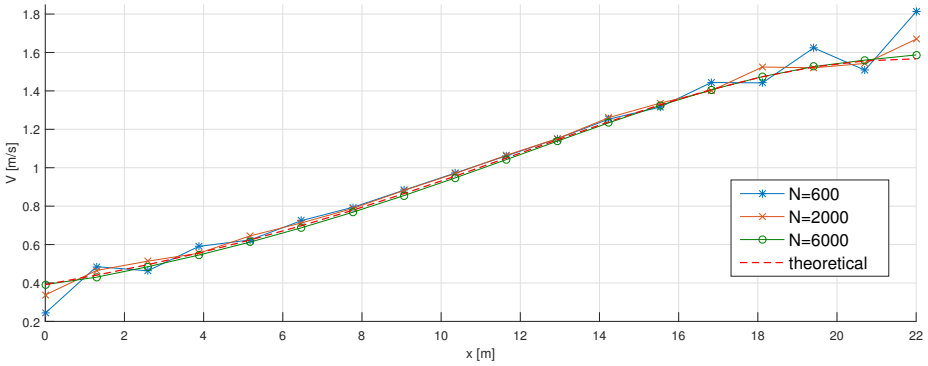


Figure 14. Integrated mean velocity along the nozzle

V. Actuator in a wind tunnel

The study of configurations combining actuator disks with impermeable surfaces has useful practical applications. Cases of interest include the analysis of tidal farm layouts with complex seabed topography [59, 60], the use of the method of images [27] to represent multiple actuators and the study of wind tunnel interference effects in rotor flows.

The historic correction for rotor blockage was proposed by Glauert [61], by combining 1D momentum theory with experiments using propellers in a closed test section wind tunnel. In 2002, Mikkelsen and Sørensen [62] developed a method to describe the induced wind speed through the rotor in a closed wind tunnel using one equation instead of a complete set. Later they generalized their method to apply it to open test sections [63].

This section demonstrates the use of a Fast Panel code for the study of wind tunnel interference effects on rotors. The Fast Panel code used in this section consists in the combination of the methods presented in section III and section II. Note that more complex configurations with asymmetric test sections, converging sections, yawed inflow, etc. can also be analyzed with this code.

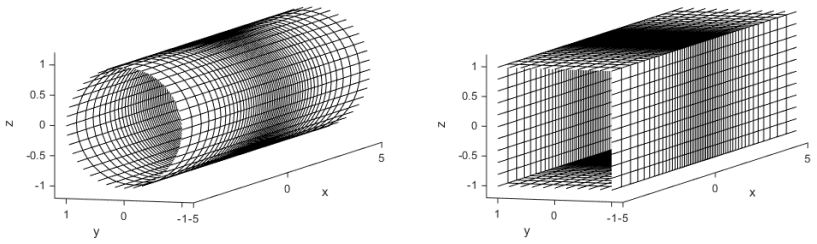
A. Method

From the actuator analysis it was found that very fine meshing is needed to conserve mass throughout the actuator. Also Holt and Hunt [64] established that very dense panelling is needed to avoid leakage through the wind tunnel walls. Therefore the wind tunnel panel mesh was made finer near the actuator disk, as shown in figure 15.

It was found that the directly calculating the A matrix was more efficient than using the FastBEM algorithm. This is because the self influence of the tunnel does not change and its calculation can therefore be excluded from the next wake iterations. Due to the dense panelling, the first iteration will be numerically expensive. Furthermore once this tunnel A matrix is calculated, new wake geometries can be added to the tunnel without significant added computational cost. Since no boundary condition is to be met on the actuator itself, its induced velocity can be calculated by the FMM and added to the right hand side of equation 7 to ensure a zero normal velocity at the collocation point.

B. Results

For positive C_T values, the expansion of the wake is constrained due to the wind tunnel walls. This means the induction will be underestimated, thus a higher disk and end-velocity will be obtained. For negative C_T values the acceleration



a) Cylindrical wind tunnel **b) Square wind tunnel**

Figure 15. Lay out of dipole panels on wind tunnel walls

will be limited due to the walls, leading to underestimated velocities at the disk and downstream. This behaviour can be seen in figure 16.

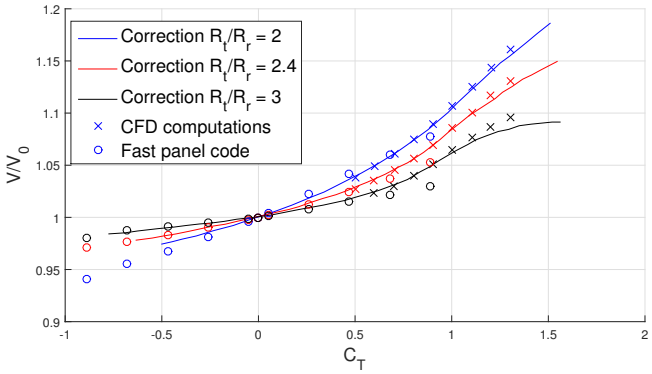


Figure 16. Influence of a cylindrical wind tunnel on the disk velocity

This leads to an overestimation of the C_P compared to the undisturbed case. Table 1 shows relative C_P values scaled by the free-space case for a constant C_T of $8/9$ and different relative square tunnel sizes.

	C_P Circular tunnel [%]	C_P Square tunnel [%]
R_t/R_r=3	+1.99	+5.18
R_t/R_r=2.4	+4.25	+6.48
R_t/R_r=2	+6.75	+9.08

Table 1. Relative power coefficient C_P w.r.t. the Betz limit for an actuator disk with thrust coefficient $C_T = 8/9$

Larger interference factors for square wind tunnels were also observed in the experiments conducted by Theodorsen and Silverstein [65]. Results from the vortex lattice method of Joppa [66] were also consistent with this observation.

Conclusion

A steady state vortex method was presented which is able to model actuator disks without the axi-symmetry requirement. The traditional quadratic scaling was reduced by employing an FMM algorithm.

Quadrilateral source and dipole panels were constructed with coplanar triangular panels to enable the use of a well established FMM library. This decreased the traditional $O(N^2)$ cost associated with the iterative wake procedure. For the actuator disk case, this translated in a time benefit of factor 30 per iteration step, for 60000 panels with an FMM accuracy of 10^{-9}. For 10^4 panels, the mass was conserved within the converged actuator wake to 0.002% and the C_P value was underestimated by 0.25%. Both metrics improve when increasing the number of panels, and so does the convergence rate.

Then the actuator was surrounded by 4 impermeable walls to simulate a wind tunnel environment. The walls consisted of dipole sheets with unknown strength, which can be computed by setting up a linear set of equations. The

A matrix in this set is to be calculated only the first iteration step since the self-influence of the tunnel does not change. For low C_T values results lined up with reference CFD results while for highly loaded disks, flow might leave the tunnel through the dipole panel edges, leading to unreliable results.

The scalability of the code and the non-symmetric problems it can handle make it a useful tool to further analyse actuator problems. Impermeable walls can be added to simulate the wind tunnel environment, ground effects or even multiple rotors by using the method of images. Multiple rotors can also be defined explicitly to analyse wind farms. Cases with yawed or varying inflow can also be studied.

References

[1] Batchelor, G., *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967.

[2] Burton, T., Sharpe, D., Jenkins, N., and Bossanyi, E., *Wind Energy Handbook*, Wiley, 2002.

[3] Rethore, P., "Modelling issues with wind turbine wake and atmospheric turbulence," *Torque 2010*, 2009, pp. 349–57.

[4] Lissaman, P., "Energy effectiveness of arbitrary arrays of wind turbines," *AIAA Journal*, Vol. 74, No. 0114, 1979.

[5] Jensen, N., "A note on wind generator interaction," Tech. rep., Risoe, DTU National Laboratory for Sustainable Energy, 1983.

[6] Ainslie, F., "Calculating the field in the wake of wind turbines," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 27, 1988, pp. 213–24.

[7] Corten, G. and Schaak, P., "More power and less loads in wind farms: "heat and flux"," *EWEC Proceedings*, 2004.

[8] Meyers, J. and Meneveau, C., "Optimal Turbine Spacing in Fully Developed Wind Farm Boundary Layers," *Wind Energy*, 2011.

[9] Diamond, K. and Crivella, E., "Wind turbine wakes, wake effect impacts and wind leases," *Duke Environmental Law and Policy Forum*, Vol. 22, No. 95, 2011.

[10] Schepers, J., *Engineering Models in Wind Energy Aerodynamics*, Ph.D. thesis, Delft University of Technology, 2012.

[11] Sanderse, B., "Aerodynamics of wind turbine wakes," Tech. Rep. ECN-E-09-016, Energy Center of the Netherlands (ECN), 2009.

[12] Pierella, F., Krogstad, P., and Saetran, L., "Blind Test 2 calculations for two inline model wind turbines where the dowstream turbine operates at various rotational speeds," *Renewable Energy*, Vol. 70, 2014, pp. 62–77.

[13] Kamaludeen, S., van Zuijle, A., and Bijl, H., "Surrogate based wind farm layout optimization using manifold mapping," *J. of Phys.: Conf. Series, Torque 2016*, Vol. 753, No. 092005, 2016.

[14] Kalvig, S., Manger, E., and Hjertager, B., "Comparing different CFD wind turbine modelling approaches with wind tunnel measurements," *J. of Phys.: Conf. Series, Torque 2012*, Vol. 555, 2014, pp. 012056.

[15] Sitaraman, J., Mavriplis, D., and EP, D., "Wind Farm simulations using a Full Rotor Model for Wind Turbines," *32nd ASME Wind Energy Symposium*, 2014, pp. 2014–1086).

[16] Troldborg, N., Sorensen, J., and RF, M., *Actuator line modeling of wind turbine wakes*, Ph.D. thesis, Technical University of Denmark, 2009.

[17] Sorensen, J., Shen, W., and Munduate, X., "Analysis of wake states by a full-field actuator actuator disk model," *Wind Energy*, Vol. 1, 1998, pp. 73–88.

[18] Mikkelsen, R., *Actuator disk methods applied to wind turbines*, Ph.D. thesis, Departmenty of Mechanical Engineering, Technical University of Denmark, 2003.

[19] Froude, R., "On the part played in propulsion by differences of fluid pressure," *Transactions of the Institute of Naval Architects*, Vol. 30, 1889, pp. 390–405.

[20] Rankine, W., "On the mechanical principles of the action of propellers," *Transactions of the Institution of Naval Architects*, Vol. 6, No. 1330, 1865.

[21] Betz, A., "Das Maximum der theoretisch Ausnuetzung des Windes durch Windmotoren," *Zeitschrift fuer das gesamte Turbinenwesen*, Vol. 26, 1920, pp. 307–309.

[22] Joukowski, N., *Theorie tourbillonaire de l'hélice propulsive*, Gauthier-Villars, Paris, 1929.

[23] Tossas, L. and S, L., "Wind Turbine Modeling for Computational Fluid Dynamics," Tech. rep., NREL/SR-5000-55054, 2014.

[24] Lignarolo, L., *On the turbulent mixing in horizonthal axis wind turbine wakes*, Ph.D. thesis, Delft University of Technology, 2016.

[25] Falcao de Campos, J., *On the calculation of ducted propeller performance in axisymmetric flows*, Ph.D. thesis, Delft University of Technology, 1983.

[26] Øye, S., "A simple vortex model," *Proc. of the third IEA Symposium on the Aerodynamics of Wind Turbines, ETSU, Harwell*, 1990, pp. 4–1.

[27] Katz, J. and Plotkin, A., *Low-Speed Aerodynamics*, Cambridge University Press, 2001.

[28] Sanderse, B., Nool, M., van Haren, S., and Koren, B., "ECNS: a new dedicated Navier-Stokes code for wind-turbine wake simulations," *J. of Phys.: Conf. Series, Torque 2012*, Vol. 555, 2012.

[29] Batten, W., Harrison, M., and Bahaj, A., "Accuracy of the actuator disc-RANS approach for predicting the performance and wake of tidal turbines," *Philosophical Transactions A*, Vol. 371, No. 1985, 2013.

[30] Crespo, A. and Hernandez, J., "A numerical model of wind turbine wake and wind farms," *Proceedings of EWEC86, vol 2, Rome, Italy, 1986: 111-5*, 1986, pp. 111–115.

[31] Schepers, J. and van der Pijl, S., "Improved modelling of wake aerodynamics and assessment of new farm control strategies," *J. of Phys.: Conf. Series, Torque 2007*, Vol. 75, 2007, pp. 012039.

[32] Calaf, M., Meneveau, C., and J., M., "Large eddy simulation study of fully developed wind-turbine array boundary layers," *Physics of Fluids*, Vol. 22, No. 015, 2011, pp. 110–16.

[33] Kuik, G. and Lignarolo, L., "Potential flow solutions for energy extracting actuator disc flows," *Wind Energy*, 2015.

[34] Cottet, G. and P.D., K., *Vortex Methods Theory and Practice*, Cambridge University Press, 2000.

[35] Moore, J., *An arbitrarily high order, unstructured, free-wake panel code*, Ph.D. thesis, Massachussets Institute of Technology, 2013.

[36] Chorin, A., "Numerical study of slightly viscous flow," *J. Fluid Mech.*, Vol. 57, No. 4, 1973, pp. 785–796.

[37] Greengard, L. and Rokhlin, V., "A fast algorithm for particle simulations," *Journal of Computational Physics*, Vol. 73, No. 2, 1987, pp. 325–348.

[38] Cipra, B., "The Best of the 20th Century: Editors Name Top 10 Algorithms," *SIAM News*, Vol. 33, No. 4, 2000, pp. 1–2.

[39] Greengard, L. and Gimbutas, Z., *FMMLIB3D Users Guide, version 1.2*, 2012.

[40] Liu, Y., "Fast multipole accelerated boundary integral equation methods," *Appl. Mech. Rev*, Vol. 55, No. 4, 2002, pp. 299–324.

[41] Liu, Y. and Nishimura, N., "The fast multipole boundary element method for potential problems: a tutorial," *Engineering Analysis with Boundary Elements*, Vol. 30, No. 5, 2006, pp. 371–381.

[42] Liu, Y., "A new fast multipole boundary element method for solving large-scale two-dimensional elastostatic problems," *International Journal for Numerical Methods in Engineering*, Vol. 65, No. 6, 2006, pp. 863–881.

[43] Liu, Y., *Fast multipole boundary element method: theory and applications in engineering*, Cambridge university press, 2009.

[44] van Kuik, G. A., "The relationship between loads and power of a rotor and an actuator disc," *Journal of Physics: Conference Series*, Vol. 555, IOP Publishing, 2014, p. 012101.

[45] Leishman, G., *Principles of Helicopter Aerodynamics*, Cambridge University Press, 2000.

[46] Maskew, B., "Program VSAERO theory Document: a computer program for calculating nonlinear aerodynamic characteristics of arbitrary configurations," Tech. rep., NASA-CR-4023, 1987.

[47] Ashby, D., Dudley, M., and Iguchi, S., "Development and validation of an advanced low-order panel method," Tech. rep., NASA TM-101024, 1988.

[48] Sbalzarini, I., Walther, J., Bergdorf, M., Hieber, S., Kotsalis, E., and Koumotsakos, P., "PPM - A highly efficient parallel particle-mesh library for the simulation of continuum systems," *Journal of Computational Physics*, Vol. 215, 2006, pp. 566–588.

[49] Molholm, M. and Walther, J., "A multiresolution method for solving the Poisson equation using high order regularization," *Journal of Computational Physics*, Vol. 326, 2016, pp. 188–196.

[50] Speck, R., "Generalized Algebraic Kernels and Multipole Expansions for Massively Parallel Vortex Particle Methods," Tech. rep., Schriften des Forschungszentrums Juelich, IAS Series Vol. 7, 2011.

[51] Kahan, W., "Further remarks on reducing truncation errors," *Communications of the ACM*, Vol. 8, No. 1, 1965, pp. 40.

[52] Rosenhead, L., "The formation of vortices from a surface of discontinuity," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 1931, pp. 170–192.

[53] Caflisch, R. and Orellana, O., "Singular solutions and ill-posedness for the evolution of vortex sheets," *SIAM Journal on Mathematical Analysis*, Vol. 20, No. 2, 1989, pp. 293–307.

[54] Erickson, L. L., "Panel methods: An introduction," Tech. rep., NASA-TP-2995, 1990.

[55] Louie, A., "Evaluation of VSAERO in prediction of aerodynamic characteristics of helicopter hub fairings," Tech. rep., NASA-TM-101048, 1989.

[56] Scalia, A., Sumbatyan, M. A., and Popuzin, V., "A fast BIE iteration method for an arbitrary body in a flow of incompressible inviscid fluid," *Journal of Computational and Applied Mathematics*, Vol. 237, No. 1, 2013, pp. 508–519.

[57] Barrett et al, R., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Society for Industrial and Applied Mathematics, 1994.

[58] Saad, Y. and Schultz, M. H., "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[59] Kolekar, N. and Banerjee, A., "Performance characterization and placement of a marine hydrokinetic turbine in a tidal channel under boundary proximity and blockage effects," *Applied Energy*, Vol. 148, 2015, pp. 121–133.

[60] Lo Brutto, O., Nguyen, V. T., Guillou, S., Thibot, J., and Gualous, H., "Performance characterization and placement of a marine hydrokinetic turbine in a tidal channel under boundary proximity and blockage effects," *Renewable Energy*, Vol. 99, 2016, pp. 347–359.

[61] Glauert, H., "Aerodynamic Theory: A General Review of Progress, volume IV, chapter Division L, Airplane Propellers," 1963.

[62] Mikkelsen, R. and Sørensen, J., "Modelling of Wind Tunnel Blockage," *15th IEA Symposium on the Aerodynamics of Wind Turbines*, 2002.

[63] Sorensen, J., Shen, W., and Mikkelsen, R., "Wall correction model for wind tunnels with open test section," *AIAA journal*, Vol. 44, No. 8, 2006, pp. 1890–1894.

[64] Holt, D. and Hunt, B., "The use of panel methods for the evaluation of subsonic wall interference," *AGARD Wall Interference in Wind Tunnels*, 1982.

[65] Theodorsen, T. and Silverstein, A., *Experimental Verification of the Theory of Wind-Tunnel Boundary Interference*, US Government Printing Office, 1934.

[66] Joppa, R., "A Method of Calculating Wind Tunnel Interference Factors for Tunnels of Arbitrary Cross-Section," Tech. rep., NASA-CR-845, 1967.

[67] W.E., A., "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quarterly of Applied Mathematics*, Vol. 9, No. 1, 1951, pp. 17–29.

[68] Trefethen, L. and Bau, D., *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, 1997.

Appendix

GMRES algorithm for FastBIE applications

Since the Jacobi method is not guaranteed to converge for the considered problems, one should look at other solution methods. The Generalised Minimised Residual (GMRES) method is applicable on any nonsymmetric system of linear equations. This is a nonstationary iterative method, meaning that its computations involve information that changes each iteration. Constants can for example be calculated using the inner product of residuals. Below the main structure of the algorithm is shown, for detailed information one should consider Ref. [58].

In the Jacobi method it was seen that the final iterative solution is a linear combination of $A^{k-1}\mathbf{b}$, with k is ranging from 1 to the maximum number of iterations. This span is defined as a Krylov subspace, the n-th Krylov can thus be written as:

$$\mathcal{K}_n(A, \mathbf{b}) = \text{span}\left\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \cdots, A^{n-1}\mathbf{b}\right\} \tag{10}$$

This basis is not linearly independent due to the way they are defined. Therefore it is useful to convert this basis to a set of orthonormal vectors $\mathcal{Q}$. The vectors $q_1, q_2, ..., q_n$ can be formed explicitly using the Arnoldi method [67], of which the pseudocode is shown below:

```
% Arnoldi iteration
q1 = b/‖b‖
for n = 1,2,3,...
    v = A q_n
    for j=1:n
        h_jn = q_j v
        v = v − h_jn q_j
    end
    h_{n+1,n} = ‖v‖_2
    q_{n+1} = v/h_{n+1,n}
end
```

The most expensive step in the algorithm is the matrix-vector multiplication $A\mathbf{q}_n$, which is in the order of $\mathcal{O}(Nn)$, with N the number of unknowns and n the number of orthonormal basis vectors. It can be seen that the algorithm does not need A explicitly, meaning that the product $A\mathbf{q}_n$ can be calculated through the FMM, thus initiating a FastBEM method.

Since the solution $x_n \epsilon \mathcal{K}_n$, it can be written as a linear combination of basis vectors $q_1, ..., q_n$. If matrix Q has those vectors as columns, then $x_n = Q_n y_n$ with $y_n \epsilon \mathbf{R}$. Now the goal is to find y_n such that the residual norm is minimized:

$$\|Ax_n - \mathbf{b}\| = \|AQ_n y_n - \mathbf{b}\| \rightarrow min \tag{11}$$

This expression can be simplified. Next to the matrix Q_n, also a matrix $\widetilde{H}_n$ is an output of the Arnoldi iteration. This is an $(n+1)$ by n upper Hessenberg matrix, meaning that entries are zero below the first subdiagonal. This matrix is related to A by:

$$AQ_n = Q_{n+1}\widetilde{H}_n \tag{12}$$

This means equation 11 can be written as:

$$\|Q_{n+1}\widetilde{H}_n y_n - \mathbf{b}\| \rightarrow min \tag{13}$$

Since the norm does not change due to the multiplication with an orthogonal matrix, one can multiply the left term with Q_{n+1}^{-1}, yielding:

$$\|\widetilde{H}_n y_n - Q_{n+1}^{-1}\mathbf{b}\| \rightarrow min \tag{14}$$

Then the product $Q_{n+1}^{-1}\mathbf{b}$ can be rewritten, starting from equation 15:

$$Q_{n+1}^{-1}\mathbf{b} = \begin{bmatrix} \mathbf{q}_1^{-1}\mathbf{b} \\ \mathbf{q}_2^{-1}\mathbf{b} \\ \vdots \\ \mathbf{q}_{n+1}^{-1}\mathbf{b} \end{bmatrix} \tag{15}$$

Since the the columns $\mathbf{q}_j$ of matrix Q_n are an orthonormal basis for Krylov space $\mathcal{K}_n$, $\mathbf{q}_1$ can be written as $\mathbf{q}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$. Due to the orthonormal properties, ${\mathbf{q_j}}^{-1}\mathbf{b} = 0$ for any $j > 1$. Which means the product $Q_{n+1}^{-1}\mathbf{b}$ can be written as $\|\mathbf{b}\|\mathbf{e}_1$. This leads to the final least squares formulation of equation 16. This least squares problem can for example be solved using a QR factorisation [68]

$$\left\|\widetilde{H}_n y_n - \|\mathbf{b}\|\mathbf{e_1}\right\| \rightarrow min. \tag{16}$$

Now the main structure of a GMRES iteration can be destilled:

```
% GMRES
q_1 = b/||b||
for n = 1,2,3,...
    Perform n-th step of Arnoldi iteration and calculate H̃_n and Q_n

    Find y that minimizes ||H̃_n y_n − ||b|| e_1||

    Check residual norm

    x_n = Q_n y.
end
```

The major drawback of GMRES is that the amount of work and storage for each iteration scales linearly with the iteration count. So unless really fast convergence is reached, this cost may become prohibitive [57]. This can be solved by restarting the iteration, after a preset n number of Arnoldi iterations, the data is cleared and set as initial data for the next n iterations. The numerical efficiency of the GMRES was proven to depend primarily on the value of n, as shown by [58]. If n is too small, GMRES might be slow to converge or fail to converge entirely. However, if it is larger than the required m, work and memory requirements increase. Determining n is a matter of experience since there are no definite rules to predict it.

For the FastBIE application of GMRES, this means the FMM is executed n times for every GMRES iteration. This can also be seen in the overview of a FastBEM implementation of the GMRES iterative solver below.

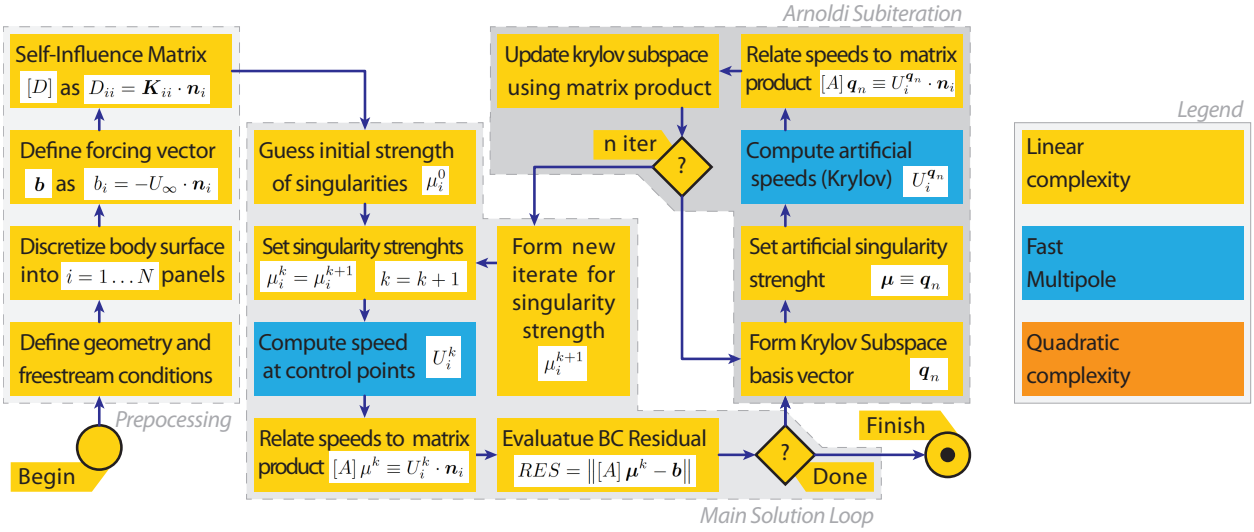


Figure 17. Structure of a instationary FastBEM panel method