Numerical study of 2D vertical axis wind and tidal turbines with a degree-adaptive hybridizable discontinuous Galerkin method

Adeline Montlaur and Giorgio Giorgiani

Abstract. This work presents a 2D study of vertical axis turbines with application to wind or tidal energy production. On one hand, a degree-adaptive Hybridizable Discontinuous Galerkin (HDG) method is used to solve this incompressible Navier-Stokes problem. The HDG method allows to drastically reduce the coupled degrees of freedom of the computation, seeking for an approximation of the solution that is defined only on the edges of the mesh. The discontinuous character of the solution provides an optimal framework for a degree-adaptive technique. Degree-adaptivity further reduces the number of degrees of freedom in the HDG computation by means of degree-refining only where more precision is needed. On the other hand, the finite volume method of ANSYS® is used to validate and compare the obtained results.

1 Introduction

In the last two decades, driven by the efforts to develop energy alternatives in order to reduce fossil fuel consumption and carbon emissions, wind power has been growing fast as an energy source, leading to cover 8% of the European Union electricity consumption in 2013 [5]. Wind energy applications have first focused on locations where enhanced speed could be utilized to increase energy output, which explains why coastal and exposed hill tops have been preferred development sites, taking advantage of speed-up effects, [2, 12]. More recently, especially in the last five years, offshore renewable energy, including tidal and offshore wind energy, has been undergoing rapid

Adeline Montlaur
Laboratori de Càlcul Numèric (LaCàN), Universitat Politècnica de Catalunya, e-mail: adeline.de.montlaur@upc.edu

Giorgio Giorgiani
CNRS, Maison de la simulation, CEA Saclay, e-mail: giorgio.giorgiani@cea.fr
development globally. 2013 was indeed a record year for offshore installations, with 1.567MW of new capacity grid connected, and with offshore wind power installations representing over 14% of the annual EU wind energy market [5].

Since the technology is similar for onshore and offshore applications, offshore wind turbines are the most developed of the marine-based renewable energy resources. Lately, the prospect of increasing its share in electricity production has given rise to significant interest in modeling tidal turbines, since tidal current energy has the distinct advantage of being reliable and highly predictable. Though horizontal-axis turbines are the most classically turbines used onshore and offshore, vertical axis wind or tidal turbines are regarded as an alternative because of two attractive basic properties: they are cheaper in terms of installation and maintenance, since the rotor and the generator are installed at the base of the mast, and they are less sensitive to turbulence and to wind or water direction. There is also an ongoing offshore project on capitalizing on the energy potential available both in the winds above the ocean, and in the currents flowing beneath the waves. A prototype has been designed, by Mitsui Ocean Development Engineering Company [11], which uses an omnidirectional Darrieus wind turbine above the sea, on one end of a vertical shaft, along with a different type of omnidirectional turbine design, a Savonius one, spinning at the other end under the water surface.

In this context, special attention is needed on the analysis of new designs of vertical axis turbines with increased performance. For such a demanding technological application, numerical methods with higher accuracy than the standard first or second-order methods are an asset in order to allow a high-fidelity computation of the aerodynamic loads. Previous work (e.g. [6]) has considered high order DG to compute vertical axis turbines showing the necessity of high accuracy for this type of application. In this work, 2D vertical axis turbines are studied. On the one hand, a degree-adaptive Hybridizable Discontinuous Galerkin (HDG) method is used to solve this incompressible Navier-Stokes problem [7, 8, 15]. The HDG method allows to drastically reduce the coupled degrees of freedom (DOF) of the computation, seeking for an approximation of the solution that is defined only on the edges of the mesh. The discontinuous character of the solution provides an optimal framework for a degree-adaptive (or p adaptation) technique, which further reduces the number of degrees of freedom in the HDG computation by means of degree-refining only where more precision is needed. This method is used along with a coordinate transformation that attaches the coordinate system to the rotating blades [10]. On the other hand, the second-order finite volume method of ANSYS® Academic Research CFD, Release 15.0 [1], is used to validate and complete the obtained results.
2 Variable degree HDG for incompressible Navier-Stokes

2.1 Navier-Stokes over a broken domain

Let \( \Omega \subset \mathbb{R}^d \) be an open bounded domain with boundary \( \partial \Omega \) split in the Dirichlet, \( \partial \Omega_D \), and Neumann, \( \partial \Omega_N \), boundaries, and let \( T \) be the final instant of interest. Recall the unsteady Navier-Stokes equations

\[
\begin{cases}
\frac{\partial u}{\partial t} + \nabla \cdot (u \otimes u) + \nabla p - \nu \Delta u = f & \text{in } \Omega \times ]0, T[,

\nabla \cdot u = 0 & \text{in } \Omega \times ]0, T[, \\
\nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_D \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
u \nabla u \cdot \mathbf{n} = \mathbf{t} & \text{on } \partial \Omega_N \times ]0, T[, \\
abla \cdot \mathbf{u}_0 = 0 & \text{in } \Omega,
\end{cases}
\]

where \( u \) and \( p \) are the velocity and the kinematic pressure in the fluid, \( \nu \) is the kinematic viscosity, \( f \) is a body force, \( n \) is the unitary outward normal vector, \( I \) is the identity matrix, \( g \) is the prescribed velocity on the Dirichlet boundary \( \partial \Omega_D \), \( t \) are the prescribed pseudo-tractions imposed on the Neumann boundary \( \partial \Omega_N \), and \( u_0 \) is the initial velocity field (assumed solenoidal: \( \nabla \cdot u_0 = 0 \)).

Note that these Neumann boundary conditions do not correspond to regular stresses but to pseudo-stresses, as it is standard in velocity–pressure formulation, see [4]. However, more physical stress boundary conditions can also be implemented, see [7] for a detailed description.

For discontinuous Galerkin approaches the domain \( \Omega \) is partitioned in \( n_{el} \) disjoint elements \( \Omega_i \) with boundaries \( \partial \Omega_i \), such that

\[
\overline{\Omega} = \bigcup_{i=1}^{n_{el}} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \text{ for } i \neq j,
\]

and the union of all \( n_{fc} \) faces (sides for 2D) is denoted as

\[
\Gamma := \bigcup_{i=1}^{n_{el}} \partial \Omega_i.
\]

The discontinuous setting induces a new problem equivalent to (1). It is written as a system of first order partial differential equations (mixed form) with some element-by-element equations and some global ones, namely, for \( i = 1, \ldots, n_{el} \).
Adeline Montlaur and Giorgio Giorgiani

\[
\begin{aligned}
L - \nabla u &= 0 \\
\partial_t u + \nabla \cdot (u \otimes u + pI - \nu L) &= f \\
\nabla \cdot u &= 0
\end{aligned}
\quad \text{in } \Omega_i \times ]0, T[ , \tag{2a}
\]

\[
\begin{aligned}
u
\end{aligned}
\quad \text{in } \Omega_i , \tag{2b}
\]

and

\[
\begin{aligned}
\llbracket u \otimes n \rrbracket &= 0 \quad \text{on } \Gamma \backslash \partial \Omega \times ]0, T[, \tag{2c}

\llbracket (-pI + \nu L) \cdot n \rrbracket &= 0 \quad \text{on } \Gamma \backslash \partial \Omega \times ]0, T[, \tag{2d}

u &= g \quad \text{on } \partial \Omega_D \times ]0, T[, \tag{2e}

(-pI + \nu L) \cdot n &= t \quad \text{on } \partial \Omega_N \times ]0, T[, \tag{2f}
\]

where a new variable \( L \) for the velocity gradient tensor is introduced after splitting the second order momentum conservation equation in two first order equations. The \textit{jump} \( \llbracket \cdot \rrbracket \) operator is defined at each internal face of \( \Gamma \), \textit{i.e.} on \( \Gamma \backslash \partial \Omega \), using values from the elements to the left and right of the face (say, \( \Omega_i \) and \( \Omega_j \)), namely

\[
\llbracket \otimes \rrbracket = \otimes_i + \otimes_j ,
\]

and always involving the normal vector \( n \), see [14] for details. Thus, equation (2c) imposes the continuity of velocity and equation (2d) imposes the continuity of the normal component of the pseudo-stress across interior faces.

\textbf{2.2 The HDG local problem}

A major feature of HDG is that, in general, unknowns are restricted to the skeleton of the mesh, that is, the union of all faces denoted by \( \Gamma \). Here, the velocity field, \( \tilde{u}(x,t) \), on the mesh skeleton \( \Gamma \) is this unknown. However, as described next, satisfying the incompressibility equation requires the introduction of one scalar unknown per element, irrespective of the polynomial degree used in the approximation. There are many choices for this scalar unknown; here, the mean pressure on the element boundary is chosen. The introduction of the new variable \( \tilde{u}(x,t) \) on the mesh skeleton \( \Gamma \) is crucial to define two types of problems: a local problem for each element and a global one for all faces.

In fact, the local element-by-element problem corresponds to the Navier-Stokes equations on each element, see equations (2a), with imposed Dirichlet boundary conditions. The imposed Dirichlet conditions on the element boundary are precisely the velocities \( \tilde{u}(x,t) \) for \( x \in \Gamma \).

It is well known that an incompressible Navier-Stokes problem on a bounded domain with non-homogeneous velocity prescribed everywhere on the boundary requires a solvability condition and implies that pressure is
Numerical study of 2D VAWTT with a HDG method

Known up to a constant. More precisely, the continuity equation implies a zero flux compatibility condition. Thus, the newly introduced variable $\hat{u}(x,t)$ on the mesh skeleton $\Gamma$ must verify

$$\langle \hat{u} \cdot n, 1 \rangle_{\partial \Omega_i} = 0, \quad \text{for } i = 1, \ldots, n_{el}. \quad (3)$$

Now the local element-by-element Navier-Stokes equations can be solved to determine $(u, L, p)$ in terms of the imposed $\hat{u}(x,t)$ on the mesh skeleton $\Gamma$. Thus, for $i = 1, \ldots, n_{el}$ the local HDG problem is solved, namely

\[
\begin{align*}
L - \nabla u &= 0 \\
\partial_t u + \nabla \cdot (u \otimes u + pI - \nu L) &= f \\
\nabla \cdot u &= 0
\end{align*} \quad \text{in } \Omega_i \times ]0,T[, \quad (4a)
\]

\[
\begin{align*}
u(x,0) &= u_0 \quad \text{in } \Omega_i, \quad (4b) \\
\hat{u} &= u \quad \text{on } \partial \Omega_i \times ]0,T[, \quad (4c)
\end{align*}
\]

\[
\langle p, 1 \rangle_{\partial \Omega_i} = \rho_i. \quad (4d)
\]

As noted earlier, (4) is a Dirichlet problem and, consequently, pressure is determined up to a constant. This constant is determined by prescribing some value for the pressure. Typical choices are to impose pressure at one point or to prescribe the mean value in the domain. In HDG, as noted earlier and since the unknowns are restricted to the mesh skeleton, the usual choice is to prescribe the mean pressure on the element boundary, see (4d), where $\langle \cdot, \cdot \rangle_B$ denotes the $L^2$ scalar product of the traces over any $B \subset \Gamma$.

At this point it is important to notice that given the values of the velocities on $\Gamma$, $\hat{u}(\cdot, t) \in [L^2(\Gamma)]^d$ for any instant $t \in [0, T]$, the same Dirichlet boundary condition is imposed to the left and right element of a given face. Consequently, the velocity continuity, recall equation (2c), is ensured by (4c). Obviously, on the Dirichlet boundary $\hat{u} = g$, replicating (2e).

Observe that in each element the original unknowns, $(L, u, p)$, can be determined in terms of the two extra unknowns; the velocity on $\Gamma$, $\hat{u}$, and the vector of average pressures on the boundaries for each element, $\{\rho\}_{i=1}^{n_{el}} \in \mathbb{R}^{n_{el}}$.

An approximation is obtained after the corresponding discretization, see [16]. Two types of finite dimensional spaces must be defined one for functions in the elements interior and another for trace functions, namely

\[
\begin{align*}
\mathcal{V}^h_i &:= \{ v : v(\cdot, t) \in \mathcal{V}^h \text{ for any } t \in [0,T] \}, \quad \text{with} \\
\mathcal{V}^h &:= \{ v \in L^2(\Omega) : v|_{\Omega_i} \in \mathcal{P}^{k_{\Omega_i}}(\Omega_i) \text{ for } i = 1, \ldots, n_{el} \}, \quad \text{and} \\
\mathcal{A}^h_i &:= \{ \hat{v} : \hat{v}(\cdot, t) \in \mathcal{A}^h \text{ for any } t \in [0,T] \}, \quad \text{with} \\
\mathcal{A}^h &:= \{ \hat{v} \in L^2(\Gamma) : \hat{v}|_{\Gamma_i} \in \mathcal{P}^{k_{\Gamma_i}}(\Gamma_i) \text{ for } i = 1, \ldots, n_{fc} \},
\end{align*}
\]

where $\mathcal{P}^k$ denotes the space of polynomials of degree less or equal to $k$, while $k_{\Omega_i}$ and $k_{\Gamma_i}$ are the polynomial degrees in element $\Omega_i$ and face $\Gamma_i$ respectively.
To simplify the presentation, in an abuse of notation, the same notation is used for the numerical approximation, belonging to the finite dimensional spaces, and the exact solution, that is \((u, L, p)\).

The weak problem for each element corresponding to (4) becomes: given \(\{\rho_i\}_{i=1}^{n_{el}} \in \mathbb{R}^{n_{el}}\) and \(\tilde{u} \in [A^h]^d\) satisfying (3), find an approximation \((L, u, p) \in [V^h]^{d \times d} \times [V^h]^d \times V^h\) such that

\[
\begin{align*}
(G, L)_{\Omega_i} + (\nabla \cdot G, u)_{\Omega_i} - \langle G \cdot n, \tilde{u}\rangle_{\partial \Omega_i} & = 0, \\
(v, \partial_t u)_{\Omega_i} + (\nu L - p I - u \otimes u, \nabla v)_{\Omega_i} & + \left( -\nu L + p I + \tilde{u} \otimes \tilde{u} \right) \cdot n + \tau (u - \tilde{u}) \cdot v \rangle_{\partial \Omega_i} = (v, f)_{\Omega_i}, \quad (5) \\
- (\nabla q, u)_{\Omega_i} + \langle q, \tilde{u} \cdot n\rangle_{\partial \Omega_i} & = 0, \\
\langle p, 1\rangle_{\partial \Omega_i} & = \rho_i,
\end{align*}
\]

for all \((G, v, q) \in [V^h]^d \times [V^h]^d \times V^h\), for \(i = 1, \ldots, n_{el}\), and with the initial condition defined in (4b). As usual, \(\langle \cdot, \cdot \rangle_{\Omega_i}\) denotes the \(L^2\) scalar product in the element \(\Omega_i\).

In this weak problem it is important to note two details. First, the Dirichlet boundary conditions, (4c), are imposed weakly; and second, the trace of the normal stress has been replaced in all boundary integrals by the following numerical trace

\[
\begin{align*}
\langle -p I + \nu L \rangle \cdot n := \langle -p I + \nu L \rangle \cdot n + \tau (\tilde{u} - u),
\end{align*}
\]

where \(\tau\) is a stability parameter. A discussion on the choice of the stabilization parameter for a degree-adaptive HDG method applied to the Navier-Stokes equations can be found in [7]. A constant \(\tau\) in the whole domain is suggested, with value \(\tau \approx \|u\|\), where \(\|u\|\) is a characteristic velocity. Here, the problem involves a moving reference frame, which is rotating with the turbine. Thus, the body forces appearing in the computational domain generate a rotating velocity whose magnitude increases with the distance from the rotation center. For this reason, particular care must be taken with the choice of the stabilization parameter \(\tau\). In fact, numerical evidence shows that if a constant \(\tau\) is used, either loss of stability or loss of superconvergence may deteriorate the numerical solution. Hence, a variable \(\tau\) is used in this case, with the following relation

\[
\tau = \begin{cases} 
\tau_0 \omega \|x\| & \text{if } \omega \|x\| \geq 1, \\
\tau_0 & \text{if } \omega \|x\| < 1,
\end{cases}
\]

where \(x\) is the position vector pointing at the barycenter of the face on which \(\tau\) is defined, \(\omega\) is the angular velocity of the turbine (assumed positive) and \(\tau_0\) is a constant value for the whole mesh.
2.3 The HDG global problem

The local problems (4), or (5), allow to compute the solution in the whole domain (that is, the variables \(L, u\) and \(p\)) in terms of the trace of the velocity on the mesh skeleton, \(\hat{u}\), and the mean pressure for each element, \(\{\rho_i\}_{i=1}^{n_{el}}\).

Thus, \(\hat{u}\) and \(\rho_i\) can now be understood as the actual unknowns of the problem. They are determined using the global equations in (2), in particular (2d) and (2f), and the solvability condition (3). In fact, as already discussed, equations (2c) and (2e) are already imposed. Thus (2d) and (2f) are the remaining global conditions which must be imposed. These two equations (in weak form) and the solvability condition (3) determine the HDG global problem. Namely, find approximations \(\hat{u} \in [A^h(\Omega)]^d\) and \(\{\rho_i\}_{i=1}^{n_{el}} \in \mathbb{R}^{n_{el}}\) such that

\[
\sum_{i=1}^{n_{el}} \left\langle \hat{v}, (-p I + \nu L) \cdot n + \tau(\hat{u} - u) \right\rangle_{\partial \Omega_i} = \left\langle \hat{v}, t \right\rangle_{\partial \Omega_N}, \tag{7a}
\]

\[
\left\langle \hat{u} \cdot n, 1 \right\rangle_{\partial \Omega_i} = 0, \text{ for } i = 1, \ldots, n_{el}, \tag{7b}
\]

for all \(\hat{v} \in [A^h(0)]^d\). Here \((L, u, p) \in [V^h_0]^d \times [V^h_0]^d \times V^h_0\) are solution of the local problems (5) and the trace spaces associated to the Dirichlet boundary are defined by \([A^h(\square)]^d = \{\hat{v} \in [A_h]^d : \hat{v} = \mathbb{P}_\Theta \square \text{ on } \partial \Omega_D\}\), with \(\mathbb{P}_\Theta\) the \(L^2\) projection on \(\partial \Omega_D\).

Note that equation (2d) imposes continuity of the of the normal component of the pseudo-traction on each element face, which induces (7a) after using (6). Thus equation (7a) weakly imposes the continuity of the normal pseudo-stress.

Remark 1 (Nonlinear DAE system). The HDG discrete problem is defined by (5) and (7). It is a system of Differential Algebraic Equations (DAE) of index 1, that can be efficiently discretized in time with an implicit time integrator, such as backward Euler, a Backward Differentiation Formula, or a diagonally implicit Runge-Kutta method, see [16]. Time discretization of (5) and (7) leads to a non-linear system of equations that can be solved with an iterative scheme. Here, the non-linear system has been linearized using the Newton-Raphson method. That is, every non-linear convective term in (5), which can be expressed as a trilinear form \(c(u, u; v)\), is linearized using the first-order approximation \(c(u^r, u^r; v) \approx c(u^{r-1}, u^r; v) + c(u^r, u^{r-1}; v) - c(u^{r-1}, u^{r-1}; v)\), where here \(r\) is the iteration count. Obviously, this implies that the global solution, equation (7), must be iterated.

In any case, a linear system of equations is solved for each iteration of the non-linear solver. In this linear system, the equations corresponding to (5) can be solved element-by-element to express the solution at each element \(\Omega_i\) in terms of the trace variable, \(\hat{u}\), and the mean of the pressure in the element boundary, \(\rho_i\). Then, these expressions are replaced in (7) yielding a global system of equations that only involves \(\hat{u}\) and \(\{\rho_i\}_{i=1}^{n_{el}}\), with an important
reduction in number of DOF. Further details on the efficient solution of the non-linear DAE can be found in [16].

Remark 2 (HDG post-processed solution). The solution of the HDG problem, given by equations (5) and (7), provides a numerical solution \((L, u) \in [V^h_d]^{d \times d} \times [V^h_d]^{d}\) with optimal numerical convergence in both variables. Then, a new problem can be solved element-by-element to compute a superconvergent approximation of the velocity \(u^*\), namely, for \(i = 1, \ldots, n_{el}\), solve

\[
\begin{align*}
\nabla \cdot \nabla u^* &= \nabla \cdot L & \text{in } \Omega_i, \\
n \cdot \nabla u^* &= n \cdot L & \text{on } \partial \Omega_i, \\
(u^*, 1)_{\Omega_i} &= (u, 1)_{\Omega_i};
\end{align*}
\]

This induces a weak problem in a richer finite dimensional space, that is, find \(u^* \in [V^h_*]^{d}\) such that

\[
(\nabla v, \nabla u^*)_{\Omega_i} = (\nabla v, L)_{\Omega_i} \quad \text{and} \quad (u^*, 1)_{\Omega_i} = (u, 1)_{\Omega_i},
\]

for all \(v \in [V^h_*]^{d}\) and \(i = 1, \ldots, n_{el}\), where \(V^h_*\) must be a bigger space than \(V^h\). In fact, with one degree more in the element-by-element polynomial approximation, i.e. \(V^h_* = \{v \in L^2(\Omega) : v|_{\Omega_i} \in P^{k+2}_{\Omega_i} \text{, for } i = 1, \ldots, n_{el}\}\), \(u^*\) converges asymptotically at a rate \(k + 2\) in the \(L^2\) norm. More details on the superconvergence properties of the post-processed solution, in particular in the case of non-uniform degree distribution, can be found in [7]. Note that the post-processing solution is not required to be computed at each time step, but only when an improved solution is needed. Moreover, the computational overhead is small and decreases with the approximation degree, see [9].

In the following section, the superconvergent solution \(u^*\) is used to compute a reliable and inexpensive error estimator for the HDG velocity approximation \(u\).

### 3 Error estimation and degree-adaptive algorithm

In the next section, the adaptive process introduced in [7] is briefly described and used in the degree-adaptive HDG simulation of the vertical axis turbine. It is based on an inexpensive, reliable and computable error estimator for the velocity field, derived using the superconvergent HDG post-processed solution.

More precisely, the \(L^2\) error in the velocity field \(u\) is estimated in an element \(\Omega_i\) as

\[
E_i^2 = \frac{1}{|\Omega_i|} \int_{\Omega_i} (u^* - u)^2 \, d\Omega,
\]

(8)
where \( \mathbf{u} \) is the solution of the HDG problem, see equations (5) and (7), and \( \mathbf{u}^* \) is the improved velocity, see Remark 2. The elemental error is normalized by the element measure: this is crucial for non-uniform meshes, see [7].

Given the error estimate (8), an automatic degree-adaptive process is proposed. A tolerance \( \varepsilon \) for the velocity elemental error is assumed. The adaptive process aims at getting a map of elemental degrees \( \{k_{\Omega_i}\}_{i=1}^{\text{n}\Omega} \) such that the error estimate of the HDG solution satisfies

\[
E_i \leq \varepsilon \quad \forall \Omega_i \subset \Omega.
\]

Hence using the error distribution, a map of degree increments \( \Delta k_{\Omega_i} \) is evaluated for each element in the whole computational domain \( \Omega \). The degree variation in each element is computed as

\[
\Delta k_{\Omega_i} = \lceil \log_b \left( \frac{E_i}{\varepsilon} \right) \rceil, \quad \text{for } i = 1, \ldots, \text{n}\Omega.
\]

where \( \lceil \cdot \rceil \) denotes the ceiling function and \( b \) is the so called adaptation aggressiveness, see [7].

The adaption process starts with a uniform degree mesh. The error estimation and consequent degree adaptation is performed after a fixed number of time steps \( N \). That is, every \( N \) time steps, the superconvergent velocity \( \mathbf{u}^* \) is computed, the error estimate (8) is evaluated, the degree map is updated using (10), the solution is projected onto the new computational mesh, and the time integration is continued.

The flow chart of the adaptive strategy is shown in Figure 1.

\[\text{Initial mesh}\]

\[\text{Calculate } N\text{ time steps}\]

\[\text{Compute new degree map; Project solution}\]

\[\text{Error estimation}\]

\[\text{Final time reached}\]

\[\begin{cases}
\text{no} & \\
\text{yes} & \text{stop}
\end{cases}\]

\textbf{Fig. 1} Flow chart of the adaptive strategy.
Numerical experiments in [7] show that, even though the adaptive process is based on an error estimate for the velocity field, accurate approximations of the fluid-dynamic forces are also obtained.

4 Numerical simulation of a vertical axis turbine

4.1 Moving reference frame

For the application of a vertical axis turbine, which is a body moving with a rotational angle $\theta = \theta(t) = \omega t$, where $\omega$ is the angular velocity of the turbine, in a absolute frame of reference $x' = (x', y')$, a corresponding moving frame of reference can be attached on the body with the transformation

\begin{align}
    x' &= x \cos \theta + y \sin \theta \\
    y' &= -x \sin \theta + y \cos \theta
\end{align}

the transformation of velocity is thus

\[ u' = A(u - \dot{\theta} I_0 x) \quad (13) \]

with

\[ A = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \quad \text{and} \quad I_0 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \quad (14) \]

System (1) is thus modified as follows for the momentum equation

\[ \partial_t u + \nabla \cdot (u \otimes u) + \nabla p - \nu \Delta u = f + 2 \dot{\theta} I_0 u + (\dot{\theta}^2 + \ddot{\theta} I_0) x \quad \text{in } \Omega, \quad (15) \]

and equation (13) is used to transform Dirichlet boundary condition. Finally, for the Neumann boundary conditions results

\[ (-p I + \nu \nabla u)n = \dot{\theta} I_0 n \quad \text{on } \partial \Omega_N. \]

More details can be found in [10].

4.2 Hypothesis

A dimensionless problem consisting of a 2D 4-blades vertical axis turbine is considered. The blades consist of NACA 0012 airfoils with a chord of dimension 1.2, and the radius of the turbine is 2.5. An angular velocity $\omega = 1$ for the turbine and an upstream velocity $V_0 = 1$ are set. The Reynolds number is $\text{Re} = 1/\nu = 1000$. The statement of the problem is shown in Figure 2.
Numerical simulations were carried out using the degree-adaptive HDG method described in Section 2 and with the commercial CFD solver, ANSYS® Academic Research CFD, Release 15.0 [1], using the sliding mesh model from the Fluent solver. An unstructured triangular mesh composed by 4474 elements was used for the HDG method. The curved boundaries of the NACA profiles were represented by piecewise second order polynomials. The adaptive HDG procedure was set up using a parameter \( b = 10 \) and a tolerance \( \varepsilon = 10^{-4} \), and the error estimation and adaptive procedure was performed every 10 time steps.

In the Fluent solver, all velocity calculations use a second-order bounded differencing scheme, and a second order scheme is used for all diffusive terms. The SIMPLE algorithm, using a relationship between velocity and pressure corrections to enforce mass conservation and to obtain the pressure field, is applied. To compare adequately the results, no turbulence model is set in the Fluent solver. The ANSYS® computational mesh is composed of 24786 quadrilateral elements. A body sizing of 0.03 is used in the region of interest, that is, in the turbine vicinity. Within the computational mesh, the coarsest element size was set to 0.5. Both methods use a first-order implicit time integration, with a time step of \( \Delta t = 0.01 \).

4.3 Results

Figure 3 illustrates how adaptive computation is particularly suited for moving geometries and reference frames. Figure 3(a) shows the vorticity in the area surrounding the turbine blades, whereas Figure 3(b) shows how the adaptive technique correctly places high-order elements around the blades and in the generated wake, where more precision is needed to properly capture the flow properties.
Vortices formation around the vertical axis turbine

Mesh and degree map around the turbine.

Fig. 3 Vorticity (a) and degree map (b) for $t = 4.3$

Figure 4 shows the time evolutions of the moment coefficient $C_M$ of the turbine and of the number of DOF. As expected, after the initial transient where the degree-adaptive algorithm increases the degree of interpolation when needed, the number of DOF reaches an almost constant value when the periodic state is achieved.

Figure 5 shows a frequency analysis of the lift and drag coefficients of the turbine. A good agreement between the HDG solution and the one computed with the commercial code ANSYS® is found, and the main harmonics are well captured by both methods. Note that the good correlation of the results observed is obtained at similar cost as Table 1 shows. HDG allows using a coarser mesh by increasing the degree of interpolation up to 9 in the region where more precision is needed, see also Figure 3(b), whereas ANSYS® needs
Fig. 5  Frequency analysis of the drag (a) and lift (b) coefficient of the vertical turbine obtained with the HDG method and with ANSYS®.

a finer mesh since the interpolation degree is the same everywhere. This leads to a similar cost, in term of number of degrees of freedom, for both methods.

<table>
<thead>
<tr>
<th></th>
<th>HDG</th>
<th>Fluent</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of elements</td>
<td>4474</td>
<td>24.786</td>
</tr>
<tr>
<td>number of DOF</td>
<td>60,000</td>
<td>75,000</td>
</tr>
<tr>
<td>degree of interpolation</td>
<td>1 – 9</td>
<td>0</td>
</tr>
</tbody>
</table>

5 Conclusions

This study shows the advantage of a degree-adaptive technique for moving reference frame applications. The HDG method is particularly suited for implanting this technique due to, on the one hand, the discontinuous character of the solution that allows to easily handle variable degree meshes. On the other hand, the superconvergence properties of the HDG solution allow to derive a simple, reliable and computationally cheap error estimator to drive the automatic adaptive process. With a simple strategy that does not involve modification of the mesh, the proposed degree-adaptive technique relaxes the need of a priori designing finite element discretization to correctly resolve the solution.

An example of a 2D cross-section of a 4-blade vertical axis turbine is considered in order to validate the method and compare the results with the commercial software ANSYS®. Similar results are obtained at a very
competitive cost for the HDG method. Future work should incorporate a turbulence model, which is indispensable in this type of studies.

References

1. ANSYS 15.0 Product Documentation (2013)