Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 80 (2015) 294 – 301

Energy Procedia

12th Deep Sea Offshore Wind R&D Conference, EERA DeepWind'2015

Spline based mesh generator for high fidelity simulation of flow around turbine blades

E. Fonn[a,*], A. Rasheed[a], A. M. Kvarving[a], T. Kvamsdal[a,b]

[a]*Applied Mathematics, SINTEF ICT, Trondheim 7035, Norway*
[b]*Mathematical Sciences, Alfred Getz vei 1, Trondheim 7091, Norway*

Abstract

Mesh generation involving complex geometries, such as wind turbine blades, is highly complicated. The problem is generally addressed using tetrahedral meshes, or hybrid meshes with hexahedral elements close to the body and tetrahedral elements elsewhere. The popularity of such mesh generators can be attributed to their associated ease of use and relative eutomation, which comes at the cost of numerical accuracy in the subsequent analysis. Added to this, such meshes can generally not represent the true geometry. Isogeometric analysis (IGA), offering an integration of analsis and CAD geometry through use of the same basis functions, has been catching up since 2005. The method offers demonstrably better accuracy, as well as an exact geometric representation. Since its inception, the method has been applied to problems from both fluid and structural mechanics. The availability of NURBS-based surface modeling software such as Rhinoceros has made it possible to create complex geometries with relative ease, but the lack of a volumetric spline-based mesh generator proves a bottleneck. In this paper we describe a mesh generator that has been developed at the Applied Mathematics Department of SINTEF ICT which can generate spline based block structured meshes of high quality for subsequent fluid or structural simulations.

Keywords: Mesh generation, Wind turbine blades, Computational fluid dynamics, Variable turbulent intensity

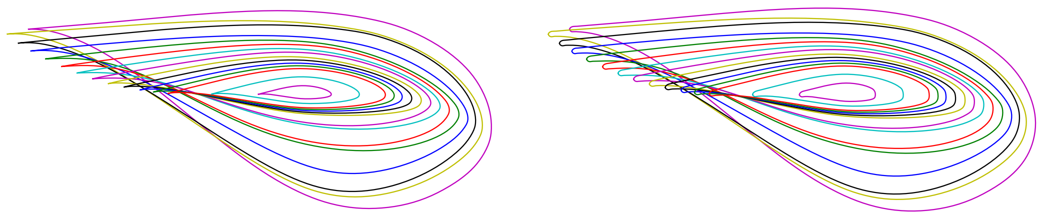
1. Introduction

This work presents an automatic block-structured spline-based mesh generator for wind turbine blades being developed to streamline the workflow from CAD modeling to simulation and analysis. It was developed initially for the NREL 5MW reference blade, but with a modular approach that will allow it to handle other geometries as well.

The whole procedure can be subdivided into two steps: solid modeling or blade construction and volumetric mesh generation. The mesh generation is performed using cubic splines everywhere, and the spline order is then adjusted in the final step before output, to also allow the creation of linear or quadratic models.

* Corresponding author. Tel.: +47-41449889
E-mail address: eivind.fonn@sintef.no, adil.rasheed@sintef.no, arne.morten.kvarving@sintef.no, trond.kvamsdal@sintef.no

doi:10.1016/j.egypro.2015.11.433



(a) Sharp trailing edge (b) Trailing edge modification (exaggerated)

Fig. 1: Airfoils for the NREL 5MW wind turbine blade.

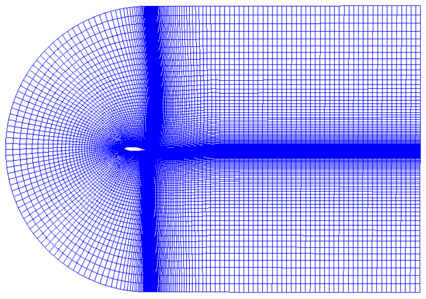


Fig. 2: A "C-mesh."

2. Methodology

2.1. Trailing edge modification of airfoils

The NREL 5MW reference blade [1] is defined in terms of cross-sectional data at 19 points along the blade axis from 2 m to 62.9 m. At each point is defined the airfoil shape (in terms of an ordered point cloud), the chord length (an isotropic scaling factor), the aerodynamic center and the twist angle. The innermost airfoils (until about 10 m) are cylindrical, the middle (until about 40 m) are Delft University airfoils of various kinds, and the remaining cross sections are NACA64 airfoils. See Figure 1a.

Each airfoil is defined as a sequence of points

$$\boldsymbol{x}_i = (x_i, f(x_i)) \pm \boldsymbol{n}_i t_i,$$

where $0 \leq x_i \leq 1$ is the chordlength parametrization, and f is a function defining the shape of the central line, on which the vector $\boldsymbol{n}$ is always normal. The numbers t_i then give half the thickness of the airfoil at each point.

The defining characteristic of the airfoils is a sharp trailing edge. Typically, this forces the construction of a "C-mesh" (Figure 2) with a large waste of degrees of freedom in regions where they are not required (above and below the trailing edge). In order to create a more efficient "O-mesh", we introduce a modification to produce rounded trailing edges, as seen in Figure 1b. This is realized as a modification to the thickness values, $\tilde{t}_i = t_i + \delta x_i$ will produce a trailing edge gap of width δ. A semicircle is then fitted in this gap with the required continuity to complete the modified airfoil.

Typically the trailing edge is chosen to be of uniform physical size, independent of the chord length, which means in practice that, for some airfoil with chord length c, the unscaled gap width δ must be chosen as $\delta(c) = \delta_0/c$. We have found small (around 1 cm) modifications to have negligible effect on observables (drag and lift). This corresponds to a modification that is between 0.22 % and 1.4 % of chord length. Figure 1b shows the effect exaggerated by a factor five.

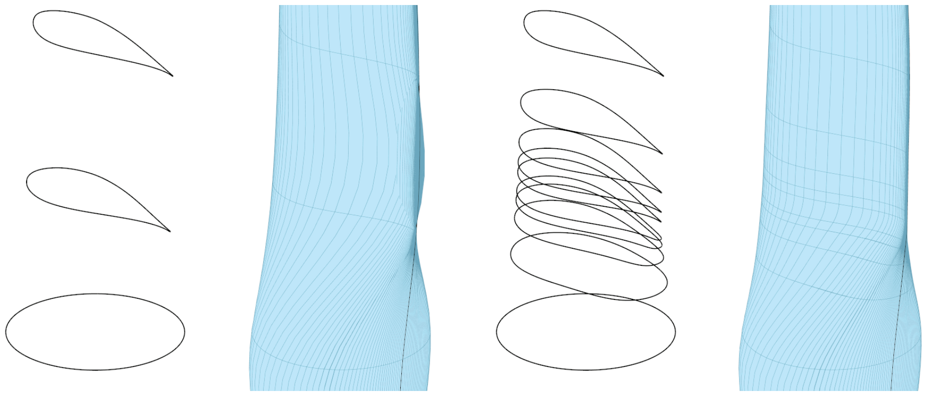


Fig. 3: An intermediate interpolation step of lower order produces a non-self-intersecting mesh.

2.2. *Lengthwise interpolation*

It is desirable to use a higher resolution in the length direction than that provided by the NREL 5MW specification. Cubic interpolation with suitable boundary conditions can be used for this, but some care will have to be taken in the intersection between DU airfoils and cylindrical cross sections. Because of the sharp transition, a high order interpolator will cause the geometry to self-intersect. This is fixed by an intermediate linear interpolation step centered at the offending airfoil. See Figure 3.

2.3. *Transfinite interpolation*

The basic premise of this section can be summarized in Figure 4.

1. A circle is drawn around the aerodynamic center of each airfoil at a suitable radius.
2. The “O-mesh” is generated and then split into eight equal patches.
3. The boundaries for eight additional patches is drawn, which creates a square mesh enveloping the “O-mesh”.
4. The enclosed area is then meshed.
5. Additional patches can then be fit to the outside as required.

This method relies on a technique to mesh an area enclosed by four curves, as in steps 2 and 4. This method is *transfinite interpolation* (TFI), also known as the Gordon Hall algorithm [2]. The remainder of this section will deal with this technique, its limitations, and how they are overcome.

2.3.1. *Linear TFI*

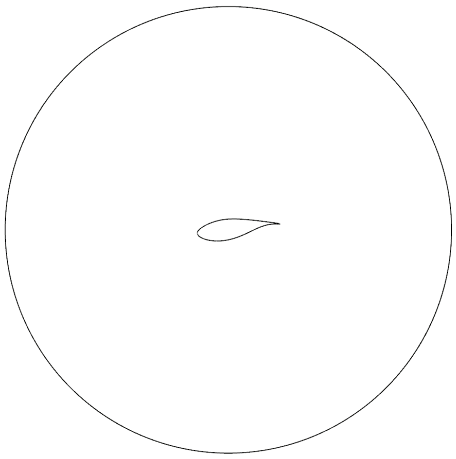
Given four curves parametrized as shown in Figure 5a, the enveloped area can be parametrized as follows, assuming without loss of generality that the parameter domains of all curves are [0, 1].

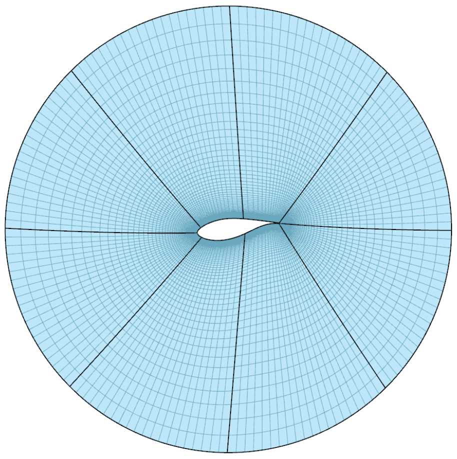
$$\begin{aligned} S(u,v) &= (1-v)c_1(u) + vc_2(u) + (1-u)c_3(v) + uc_4(v) \\ &\quad - [(1-u)(1-v)c_1(0) + u(1-v)c_1(1) + (1-u)vc_2(0) + uvc_2(1)] \end{aligned} \tag{1}$$

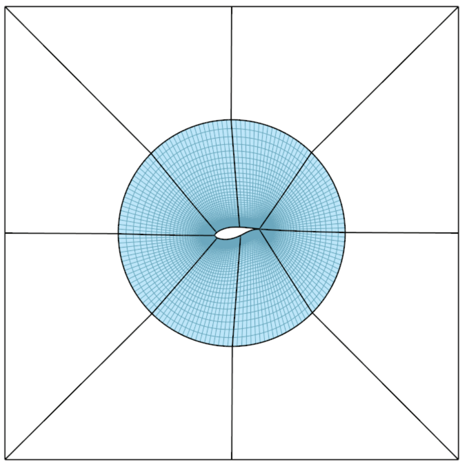
This is essentially the sum of two linear interpolations (one in each direction) minus the linear interpolation in both directions.

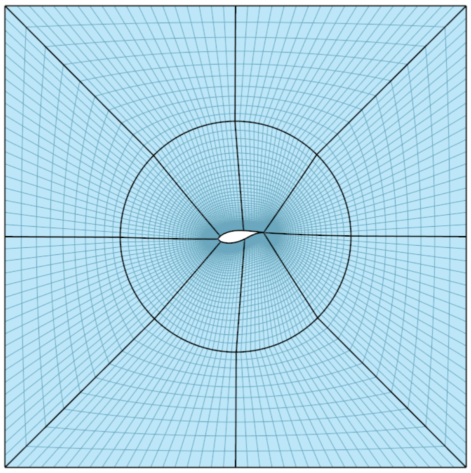
This simple method produces generally excellent and reliable results, with a few exceptions.

- It is highly sensitive to the parametrization (not just the geometry) of the four curves.
- It cannot guarantee orthogonal meshlines near the body, which is of interest in high Reynolds number CFD applications.


(a) Airfoil and circle.


(b) “O-mesh” filled in and split.


(c) “O-mesh” and square.


(d) Square filled in.

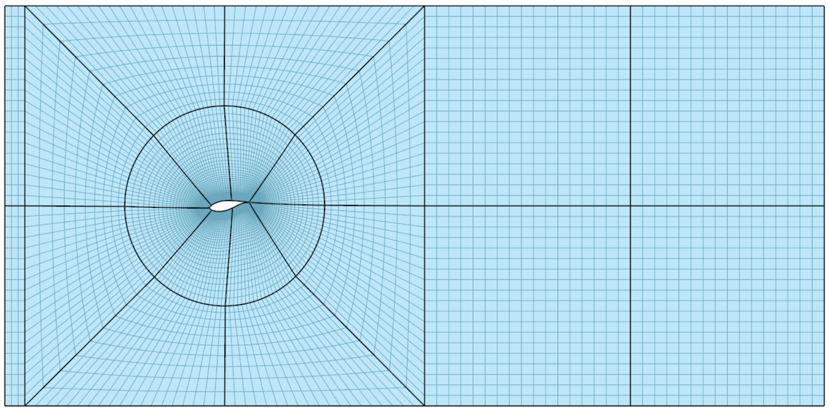
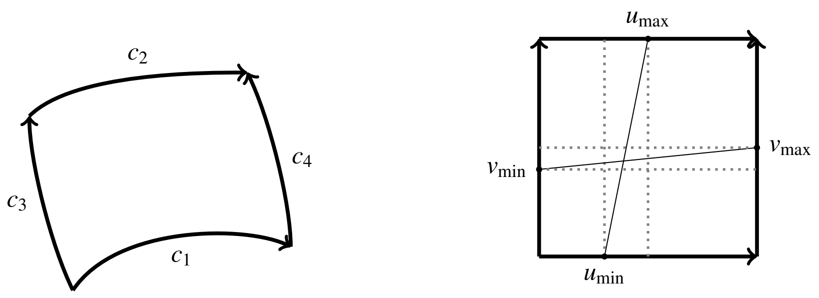

(e) Example of additional patches.

Fig. 4: Strategy for two-dimensional mesh generation. Compare with Figure 2.



(a) Curve-to-surface TFI. (b) Normalized TFI.

- The Jacobian is not guaranteed to be everywhere positive, which means gridlines might intersect. This happens in rare cases of nonconvex domains.

In the following, we offer solutions to these problems.

2.3.2. *Normalized TFI*

It is sometimes useful to be able to produce distinct parametrizations of the interior domain from distinct parametrizations of the surrounding curves, but in our work we have found it useful to implement a parameter-normalized form of TFI.

If all curves c_i are parametrized by arclength, (1) produces resonable results. However, since two opposing curves must have the same parametrization, it is not generally possible for both to be simultaneously parametrized by arclength.

Given parameters $u_{\min}, u_{\max}, v_{\min}, v_{\max}$ of $c_1, \ldots, c_4$ respectively, corresponding for example to knots, the normalized parameters of the interior should satisfy the following condition (see Figure 5b),

$$u = u_{\min} + v\underbrace{(u_{\max} - u_{\min})}_{\Delta u}, \qquad v = v_{\min} + u\underbrace{(v_{\max} - v_{\min})}_{\Delta v}$$

Whence we find the values u, v for use in (1),

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{1 - \Delta u \Delta v} \begin{pmatrix} 1 & \Delta u \\ \Delta v & 1 \end{pmatrix} \begin{pmatrix} u_{\min} \\ v_{\min} \end{pmatrix}.$$

2.3.3. *Orthogonalized TFI*

To obtain orthogonal meshlines near the airfoil (Figure 6), we employ an algorithm that grows the mesh layer by layer from the body outwards. At each step j, the gridpoints on the next layer are given by a weighted mean,

$$\boldsymbol{p}_i^j = \alpha_j \boldsymbol{q}_i^j + (1 - \alpha_j)\boldsymbol{t}_i^j. \tag{2}$$

Here, $\boldsymbol{t}$ denotes the points produced by (1), and $\boldsymbol{q}$ denotes the orthogonally projected points from the previous layer, i.e.

$$\boldsymbol{q}_i^j = \boldsymbol{p}_i^{j-1} + \boldsymbol{n}_i^{j-1}\Delta_i^j$$

where $\boldsymbol{n}$ denotes the normal to each layer curve and Δ_i^j is a suitable distance function.

The coefficient α_j controls the orthogonality of the mesh at each step, and should vanish as j grows. A suitable choice is

$$\alpha_j = \begin{cases} 1, & j \leq k, \\ 0, & j = N, \\ r_\alpha^{j-k}, & \text{otherwise.} \end{cases}$$

where k is the number of elements in the boundary layer, N is the total number of elements and $r_\alpha < 1$.

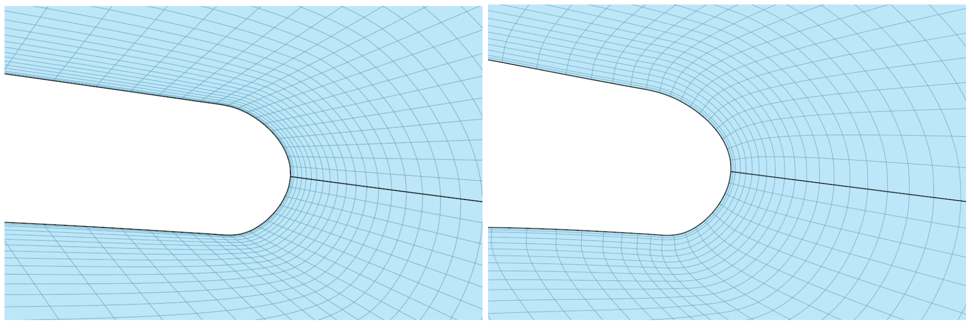


Fig. 6: Orthogonal meshlines near the trailing edge.

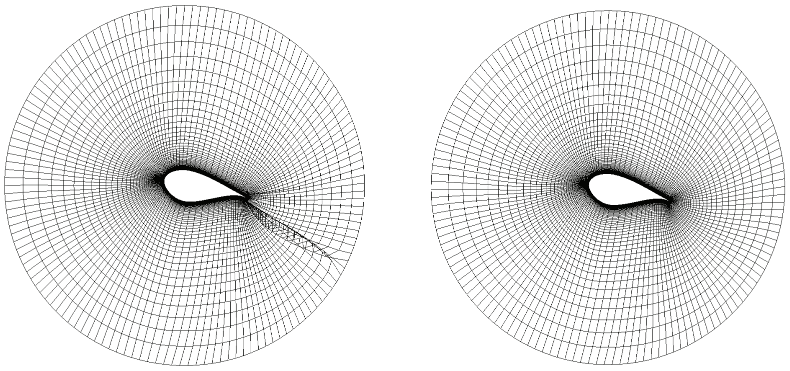


Fig. 7: Laplacian smoothing.

2.4. Smoothed TFI

The given algorithm may produce negative-Jacobian meshes in some cases, particularly for concave geometries and, in connection with orthogonality, large boundary layers. To combat this, one may perform a Laplacian smoothing on the orthogonally projected points $\boldsymbol{q}$ prior to (2). If the precise distribution of the final layer points is not critical, it might be more effective to perform the smoothing *after* (2), on the points $\boldsymbol{p}$. Note that this smoothing operator should act on the parameter space of the curve traced out by the points rather than on the points directly. The smoothing factor should behave much like α, possibly with a larger decay factor. See Figure 7.

2.5. Tip geometry and closure

We here describe a heuristic algorithm for generating a closed tip geometry, which is fitted on the final airfoil in a C^1-continuous manner. To form the basic geometry of the tip, we construct the midpoint of the final airfoil, and raise it a suitable distance (here, about 20 cm) to produce a center point. We then interpolate between the two endpoints of the airfoil and the center point to produce a raised midline. At each pair of corresponding points on the airfoil, we can now interpolate to form a cross-curve orthogonal to the raised midline. This produces a parametrization of the tip geometry, with singularities. This geometry can then be partitioned into twelve non-singular patches as shown in Figure 8.

For the eight patches not touching the singular points of the original para-metrization, this is most easily done by constructing a mesh on the parameter domain and mapping it accordingly. For the other four patches, we have performed ordinary TFI in physical space as already described. The resulting geometry can then be projected by least distance onto the wingtip.

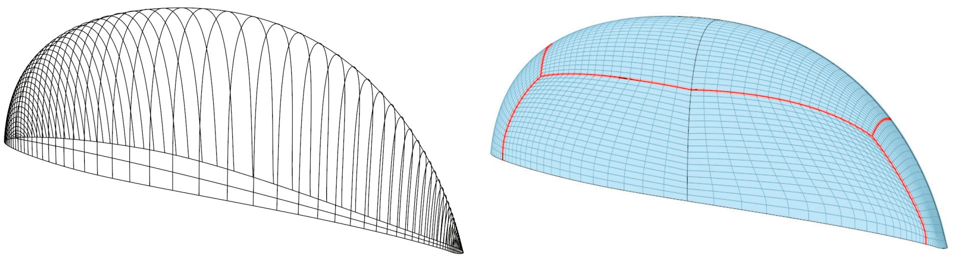


Fig. 8: Constructing the tip geometry.

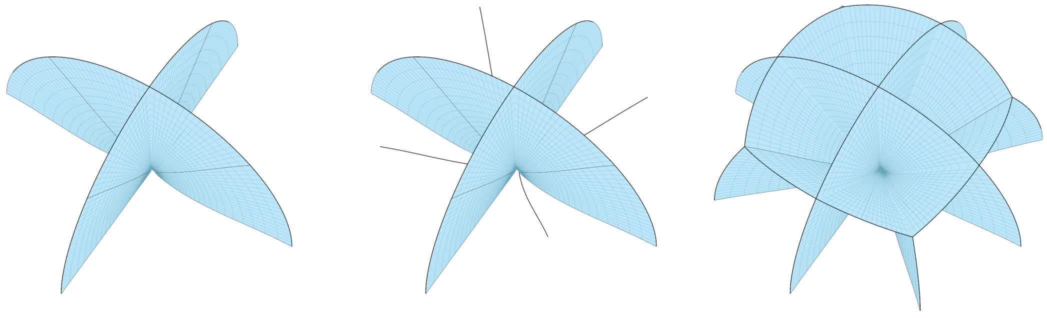


Fig. 9: Generating the surfaces of the surrounding mesh.

2.6. *Surrounding mesh*

To produce the surrounding volumetric wingtip mesh, we rely on volumetric TFI, which can be used to mesh the volume enclosed in six surfaces, and to which all the modifications described in subsection 2.3 carry over in more or less straightforward manner.

The strategy is then:

1. Form the curves describing the patch structure of the surrounding mesh.
2. Form the surfaces between these curves using two-dimensional TFI.
3. Form the volumes using volumetric TFI.

Most of the surfaces required in step 2 can be generated immediately using the same algorithm as already described. The remainder require the formation of four central curves emanating from the center point of each quadrant. This can be done by ordinary interpolation, but a better method is to combine two patch curves emanating from the center point (marked with red in Figure 8) as a spline curve with a C^0 discontinuity, perform TFI with the corresponding outer curve, and extracting the necessary curve from the resulting surface. This allows the remaining surfaces to be generated. See Figure 9.

The block structure of the completed mesh, with four radial patches, can be seen in Figure 10.

3. Conclusion and future work

A 2D version of the mesh generator has been developed to generate spline based meshes for NACA0015 [3] and NACA0012 [4] air foils for computation fluid dynamics studies and cylinder with an attached bar [5] for fluid structure interaction simulations. All these papers give detailed analyses of the mesh quality using different criteria.

A more sophisticated 3D version of the code has been used to study thermal heat storage in concrete. In the work thermal induced expansion and resulting stresses were analyzed in addition to the heat storage characteristics for

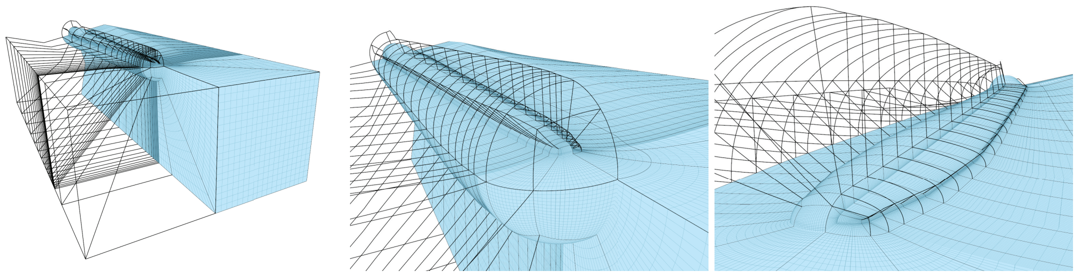


Fig. 10: The full mesh block structure.

different geometries. Currently a new concept based on strip theory to simulate fluid structure interaction of a full 3D blade is being tested.

A highlight of the mesh generator is the ease of control on the mesh quality allowed by changing key parameters like geometry, local resolution and load balancing. The tool presented in this paper bridges a gap that existed in the comprehensive design methodology that is being developed to conduct fluid structure interaction simulation of a full offshore wind turbine [6]. Although the generator has been developed for our indigenous tool IFEM it is fully compatible with other CFD tools like OpenFOAM. The methodology ensures exact geometric representation and hexahedral elements everywhere. The block structured meshing approach offers more flexibility in optimal load balancing for high performance computing.

Currently, work is ongoing to handle even more complex geometries like terrain, groups of buildings and heat exchangers with high performance computing in mind.

Acknowledgements

The authors acknowledge the financial support from the Norwegian Research Council and the industrial partners of the FSI-WT (216465/E20) (http://www.fsi-wt.no) projects.

References

[1] Jonkman, J., Butterfield, S., Musial, W., Scott, G.. Definition of a 5-MW reference wind turbine for offshore system development. Tech. Rep. NREL/TP-500-36060; National Renewable Energy Laboratory; Golden, Colorado, USA; 2009. URL: http://www.nrel.gov/docs/fy09osti/38060.pdf.

[2] Gordon, W.J., Hall, C.A.. Construction of curvilinear co-ordinate systems and applications to mesh generation. International Journal for Numerical Methods in Engineering 1973;7(4):461–477.

[3] Nordanger, K., Holdahl, R., Kvarving, A.M., Kvamsdal, T., Rasheed, A.. Simulation of air past a NACA0015 airfoil using an isogeometric incompressible Navier-Stokes solver with the Spalart-Allmaras turbulence model. Conditionally accepted in Computer Methods in Applied Mechanics and Engineering 2015;.

[4] Nordanger, K., Holdahl, R., Kvarving, A.M., Rasheed, A., Kvamsdal, T.. Implementation and comparison of three isogeometric Navier-Stokes solvers applied to simulation of flow past a fixed 2D NACA0012 airfoil at high Reynolds number. Computer Methods in Applied Mechanics and Engineering 2015;284:664–688. URL: http://www.sciencedirect.com/science/article/pii/S0045782514004034.

[5] Nordanger, K., et al. Numerical benchmarking of fluid-structure interaction: an isogeometric finite element approach. Submitted to the Journal of Fluids and Structures 2015;.

[6] Rasheed, A., Holdahl, R., Kvamsdal, T., kervik, E.. A comprehensive simulation methodology for fluid-structure intraction of offshore wind turbines. Energy Procedia 2014;53:135–145. URL: http://www.sciencedirect.com/science/article/pii/S1876610214010996.