Copy No. _____

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

# A procedure to create CFD meshes around propellers

*David Hally*

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2013-180
November 2013

Canada

# A procedure to create CFD meshes around propellers

David Hally

**Defence Research and Development Canada – Atlantic**
Technical Memorandum
DRDC Atlantic TM 2013-180
November 2013

# Abstract

A procedure has been developed for generating computational grids that can be used for calculating the flow around propellers using Computational Fluid Dynamics (CFD) flow solvers. The grid generation program Pointwise from Pointwise, Inc. is used. The procedures are encapsulated in six scripts in the Pointwise scripting language Pointwise Glyph2. Each script is described in detail and instructions for their use are also provided.

# Résumé

Nous avons conçu une procédure de production de grilles de calcul en mécanique des fluides numérique (MFN), utilisable avec des solutionneurs d'écoulements pour déterminer l'écoulement autour des hélices. Ces grilles sont produites à l'aide du programme Pointwise de la société Pointwise, inc. Les procédures sont encapsulées dans six scripts écrits en Glyph2, le langage de script de cette entreprise. Nous décrivons en détail chaque script et donnons des instructions pour leur utilisation.

This page intentionally left blank.

# Executive summary

## A procedure to create CFD meshes around propellers

David Hally; DRDC Atlantic TM 2013-180; Defence Research and Development Canada – Atlantic; November 2013.

**Background:** At DRDC Atlantic, panel methods have been used to calculate flows around propellers for many years. However, panel methods are unable to give good predictions when viscous flow becomes important: e.g. predictions of pressures in tip, leading edge and hub vortices to allow the prediction of cavitation inception and the consequent radiated noise; and predictions of propeller performance at off-design conditions when flow separation may be important. To address these issues viscous flow solvers must be used. To that end, DRDC Atlantic is investigating the use of Reynolds-averaged Navier-Stokes (RANS) solvers for predicting the flows past propellers.

The accuracy of a RANS computation is strongly dependent on the quality of the grid which it uses. This document describes procedures for generating high quality grids on propellers.

**Principal results:** Procedures have been developed for generating high quality grids on propellers using the grid generation program Pointwise from Pointwise Inc. The procedures are encapsulated in six scripts in the Pointwise scripting language Pointwise Glyph2.

**Significance:** The scripts allow rapid generation of grids to allow the flow around propellers to be computed and used in the prediction of propeller performance (thrust and torque) as well as the prediction of cavitation inception and the levels of radiated noise resulting from it.

**Future work:** The scripts could be extended in several ways: to allow fillets between the blades and the hub; to include the propeller shaft extending to the inflow plane; and to modify the block topology to increase accuracy in the prediction of tip and hub vortices, and to provide more flexibility in the concentration of grid nodes.

# Sommaire

## A procedure to create CFD meshes around propellers

David Hally ; DRDC Atlantic TM 2013-180 ; Recherche et développement pour la défense Canada – Atlantique ; novembre 2013.

**Contexte :** RDDC Atlantique utilise depuis plusieurs années des méthodes par facettes pour calculer l'écoulement autour des hélices. Toutefois, de telles méthodes ne peuvent faire de bonnes prédictions en présence d'un important écoulement visqueux : par exemple, les prédictions des pressions des tourbillons au sommet et au bord d'attaque des pales ainsi qu'au moyeu pour prédire l'apparition de la cavitation et du bruit rayonné qu'elle produit ; et les prédictions du rendement de l'hélice dans des conditions non prévues à la conception lorsque la séparation d'écoulement peut être importante. On utilise des solutionneurs d'écoulement visqueux pour résoudre ces difficultés. À cette fin, RDDC Atlantique fait des recherches sur l'utilisation de solutionneurs d'équations de Navier-Stokes en moyenne de Reynolds (Reynolds-Averaged Navier-Stokes - RANS) pour prévoir l'écoulement autour des hélices.

La précision de ces calculs dépend fortement de la qualité du maillage utilisé. Dans le présent document, nous décrivons des procédures de production de grilles de haute qualité pour le calcul des hélices.

**Résultats principaux :** Nous avons conçu des procédures pour produire des grilles de calcul de haute qualité pour les hélices à l'aide du programme Pointwise de la société Pointwise. Les procédures sont encapsulées dans six scripts écrits en Glyph2, le langage de script de cette entreprise.

**Importance :** Les scripts permettent de produire rapidement des grilles permettant le calcul de l'écoulement autour des hélices lequel est utilisé pour prédire le rendement de l'hélice (poussée et couple) et l'apparition de la cavitation et l'intensité du bruit rayonné qui en résulte.

**Travaux à venir :** On peut étendre ces scripts de plusieurs façons : permettre le calcul des pieds de pales ; inclure le prolongement de l'arbre porte-hélice dans le plan d'écoulement ; modifier la topologie des blocs pour accroître la précision de la prédiction des tourbillons au sommet des pales et au moyeu, et pour offrir plus de souplesse lors de la définition de la concentration des nœuds de la grille.

# List of figures

This page intentionally left blank.

# 1 Introduction

At DRDC Atlantic, panel methods have been used to calculate flows around propellers for many years. However, panel methods are unable to give good predictions when viscous flow becomes important: e.g. predictions of pressures in tip, leading edge and hub vortices; and predictions of propeller performance at off-design conditions when flow separation may be important. To address these issues viscous flow solvers must be used. To that end, DRDC Atlantic is investigating the use of Reynolds-averaged Navier-Stokes (RANS) solvers for predicting the flows past propellers.

The accuracy of a RANS computation is strongly dependent on the quality of the grid which it uses. This document describes procedures for generating high quality grids on propellers using the grid generation program Pointwise from Pointwise, Inc. [1]. The procedures are encapsulated in six scripts in the Pointwise scripting language Pointwise Glyph2 [2], based on the programming language Tcl [3].

The scripts save the grid in both Pointwise format and the format used by ANSYS CFX, the flow solver used at DRDC Atlantic for propeller calculations. However, it would be simple to include output for other flow solvers as well.

# 2 Coordinate systems

The scripts use a Cartesian coordinate system attached to the propeller in which the origin is the point at which the blade generator line meets the propeller axis. The $z$ coordinate increases from upstream to downstream along the propeller axis. The $y$ coordinate increases outward from the propeller axis along the propeller reference line. The $x$ coordinate is perpendicular to $y$ and $z$ such that the coordinate system is right-handed; it increases to starboard when the tip of the reference blade is at top dead centre. The Cartesian coordinate system is illustrated in Figure 1.

The reference propeller blade is parameterized using $\xi$ and $\eta$; a point on the blade will be denoted by $\mathbf{b}(\xi,\eta)$. On all blades $\eta$ increases from 0.0 at the propeller axis to 1.0 at the blade tip. Typically there is a minimum value of $\eta$ below which the reference blade geometry is undefined.

On a right-handed blade $\xi$ increases from 0.0 at the trailing edge, increases along the pressure side of the blade to 0.5 at the leading edge, the increases along the suction side of the blade to 1.0 at the trailing edge. On a left-handed blade $\xi$ increases first along the suction side, then along the pressure side. This ensures that a normal to the blade defined by

$$\mathbf{n} = \frac{\partial \mathbf{b}}{\partial \xi} \times \frac{\partial \mathbf{b}}{\partial \eta} \tag{1}$$

is always outward pointing.

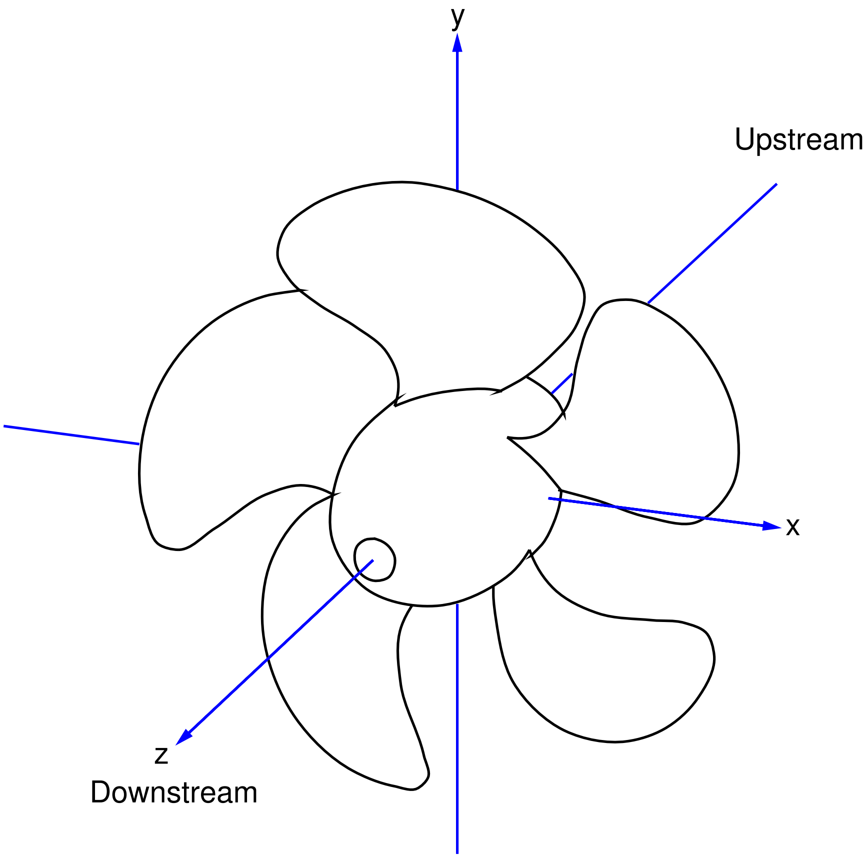

***Figure 1:*** *The Cartesian coordinate system.*

# 3 Propeller geometry

The geometry of propeller blades is usually specified by providing sections from the hub to the tip of a reference blade. Each section is specified by its shape (usually given as a series of offsets from the chord line joining the leading and trailing edges), chord length, pitch, skew angle and rake. These data are splined to generate a surface representation of the whole blade.

The DRDC propeller geometry classes [4] can be used to generate the reference blade by splining the sections to generate a smooth surface. However, the surface obtained is not suitable for the grid generation procedures for two reasons:

1. it contains a coordinate singularity at the tip where the chord length reduces to zero; and
2. it will often have extremely high curvature in the region near the tip which can cause the grid extrusion methods to fail.

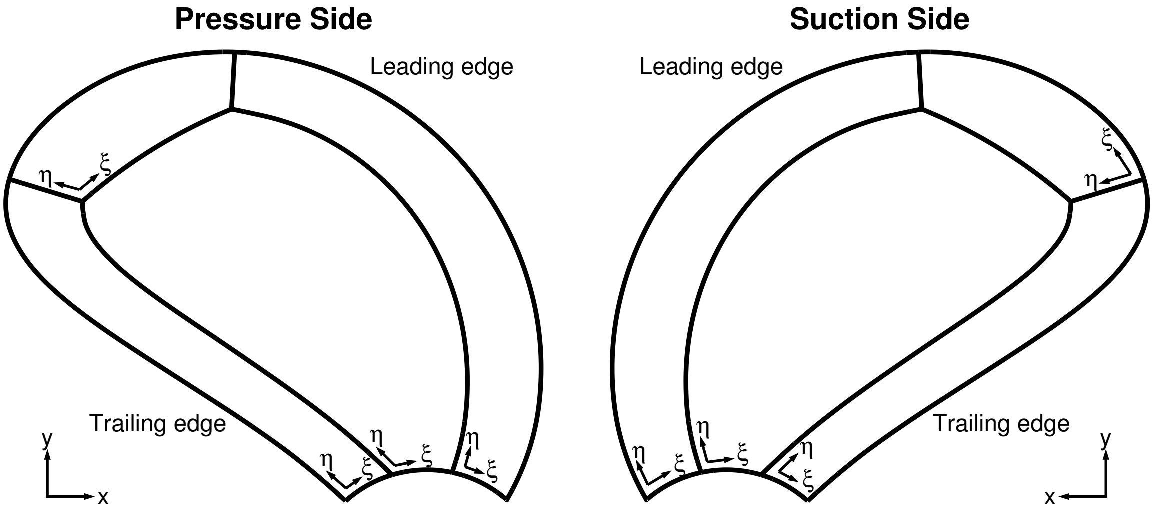

***Figure 2:*** *The parameters* $(\xi,\eta)$ *used to define the blade surfaces.*

The program `smooth-prop` [5,6] resolves both these problems. It splits the surface of the reference blade into five adjoining surfaces to avoid the coordinate singularity and also smooths the surfaces along the leading and trailing edges near the tip.

The hub is an axisymmetric surface specified by a curve which is rotated about the propeller axis. The hub is split into sectors, one for each blade. The edge of each sector runs roughly parallel to the line on the hub joining the leading and trailing edges of its blade, tracing a helical shape as it extends to the ends of the hub. The intersection of the hub and the reference blade is calculated and the hub sectors are trimmed by removing the footprints of the blades. The blades are also trimmed so that only the portion above the hub is retained.

No filleting is used between the blades and hub surface as fillets are usually not defined in standard descriptions of propeller geometry.

Each surface is defined in terms of two parameters, $(\xi,\eta)$, oriented so that $\hat{\xi}\times\hat{\eta}$ is an outward pointing normal. Figure 2 shows the directions of these parameters for the blade surfaces. On the hub surface the parameter $\xi$ increases from the upstream end of the hub to the downstream end; the parameter $\eta$ increases around the circumference of the hub.

The propeller geometry is saved in a file in IGES format [7] which is used as an input file for the scripts. Only the five surfaces on the reference blade, its associated hub sector and the intersection line between the blade and hub are saved as separate entities. The other blades and the full hub are generated by copying and rotating these entities. Figure 3 shows a propeller along with the decomposition of the blade and hub surface into the six reference surfaces. A full specification of the contents of the IGES file is given in Annex B.

The reference surfaces written to the IGES file are as follows:

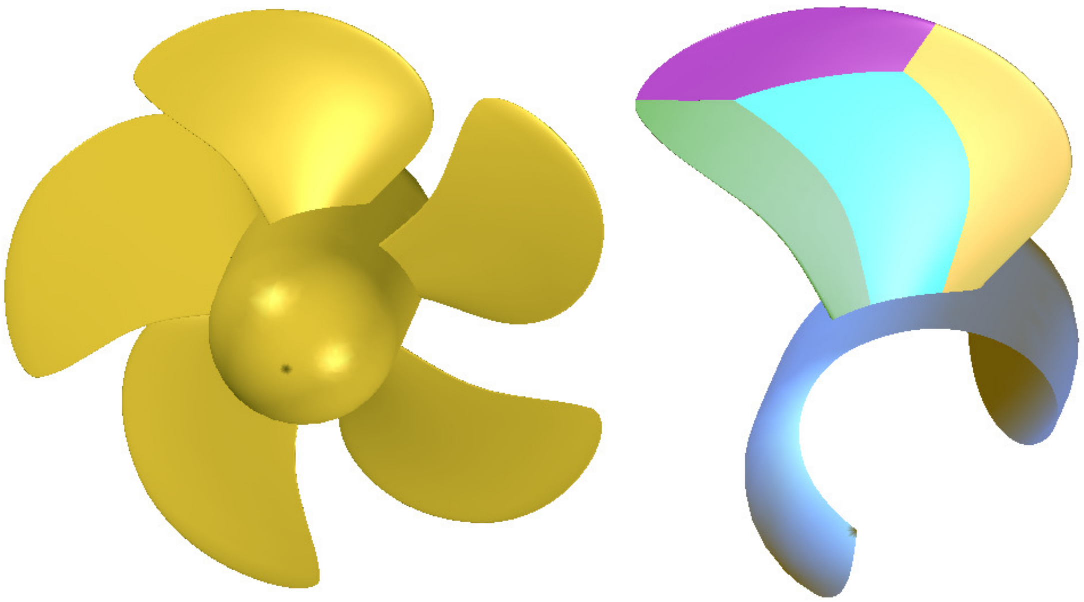

***Figure 3:*** *The full propeller (left) and the reference surfaces saved in the IGES file (right). Four of the five blade surfaces can be seen. The fifth is in the centre of the other side of the blade.*

**central surface pressure side**

The surface in the center of the pressure (aft/downstream) side of the blade. The parameter $\xi$ increases as one proceeds from the edge near the trailing edge toward the edge near the leading edge. The parameter $\eta$ increases as one proceeds from the hub toward the tip.

**central surface suction side**

The surface in the center of the suction (forward/upstream) side of the blade. The parameter $\xi$ increases as one proceeds from the edge near the leading edge toward the edge near the trailing edge. The parameter $\eta$ increases as one proceeds from the hub toward the tip.

**tip surface**

The surface containing the tip wrapping around the upper portions of both the leading and trailing edges. Its parameter $\xi$ increases along the top of the central region from the leading edge toward the trailing edge. The parameter $\eta$ increases as one proceeds across the leading/trailing edge from the pressure side to the suction side. The leading/trailing edge is the line with $\eta = 0.5$.

**leading edge surface**

The surface wrapping around the leading edge. The parameter $\xi$ increases as one passes through the leading edge from the pressure side to the suction side. The parameter $\eta$ increases as one proceeds from the hub toward the tip. The leading edge is the line with $\xi = 0.5$.

**trailing edge surface**

The surface wrapping around the leading edge. The parameter $\xi$ increases as one passes through the trailing edge from the suction side to the pressure side. The parameter $\eta$ increases as one proceeds from the hub toward the tip. The trailing edge is the line with $\xi = 0.5$.

# 4 The grid generation process

The propeller grid generation is done using the program Pointwise from Pointwise, Inc. Pointwise provides a scripting language, PointWise Glyph2, based on the programming language Tcl [3], which has been used to automate the propeller grid generation. The propeller scripts use the IGES file generated by `smooth-prop` as input.

## 4.1 Terminology

In Pointwise parlance a connector is a curve containing nodes. Its dimension is the number of nodes that it contains. The placement of the nodes is determined by the shape of the connector, its dimension, the spacing of the nodes at each end, and a distribution function. The term relative spacing will also be used to mean the ratio of the node spacing at the end of a connector to the arclength of the connector.

A domain is a surface containing nodes. It can be structured or unstructured: structured domains consist of a topologically rectangular mesh of quadrilateral cells; unstructured domains consist of nodes connected into a mesh of triangles. The boundaries of a domain are called edges; each edge consists of a sequence of connectors. Structured domains always have four edges, while unstructured domains have a single edge defining the outer boundary and any number of additional internal edges defining holes in the domain.

A block is a region of space containing nodes. It, too, may be structured or unstructured: structured blocks consist of an ordered mesh of hexahedral cells; unstructured blocks consist of a mesh of modes connected into tetrahedra, triangular prisms, pyramids, or hexahedra. The boundaries of a block are called faces; each face consists of a collection of domains. Structured blocks always have six faces, while unstructured blocks have a single face defining the outer boundary and any number of additional internal faces defining holes in the block.

Database entities are geometric objects used to constrain the grid. They include points, curves, surfaces, etc.

A layer is a group of connectors, domains, blocks or database entities. The whole layer can easily be made visible or invisible, so they provide a convenient mechanism for decluttering the display and allowing the user to show only what is currently of interest.

## 4.2 The attribute files

The user of the scripts can control them by defining a number of attributes to set the number of nodes on connectors, spacing at the ends of the connectors, etc. All but three of the attributes are given default values in the file `default-attributes.glf` which is read by each of the scripts. The three undefined attributes, as well as any of the attributes whose values are to be changed from the default, must be defined in the input file `attributes.glf`.

A full list of the attributes is given in Annex A. The use of each attribute by each script is also described in the following sections.

## 4.3 Overview

The grid generation is performed using the following steps.

1. The geometry of the propeller is read from the IGES file.
2. Two nested cylindrical shells are created, the inner one enclosing the propeller blades. The outer cylinder has radius $R_o$ and extends from $z = z_{omin}$ to $z = z_{omax}$; the inner cylinder has radius $R_i$ and extends from $z = z_{imin}$ to $z = z_{imax}$. If the plane $z = z_{imin}$ intersects the hub, then the hub surface is trimmed at $z = z_{imin}$ and extended as a cylinder to $z = z_{omin}$. This makes it straightforward to model the propeller shaft even if it is not included in the IGES file. Similarly, if the plane $z = z_{imax}$ intersects the hub, it will be trimmed and extended downstream to $z = z_{omax}$ making it easy to model a typical open water set-up in which the shaft extends downstream of the propeller.
3. Structured domains (two-dimensional grids) are made on each of the five blade surfaces then smoothed to minimize adverse effects of skewness.
4. The domains on the blades are extruded normal to the blade surfaces to generate an inflation layer for the boundary layers on the blades. The domain edges at the hub are required to follow the hub surface during the extrusion.
5. An unstructured domain is generated on the sector of the hub associated with the blade (the portion of the hub stored in the IGES file: see Figure 3).
6. The hub domain and the blocks on the blades are copied and rotated so that the full hub and all propeller blades are covered.
7. An inflation layer for the hub boundary layer is generated by normal extrusion of the unstructured hub domains. The edges of the extruded block are required to match the lower portions of the blade blocks.

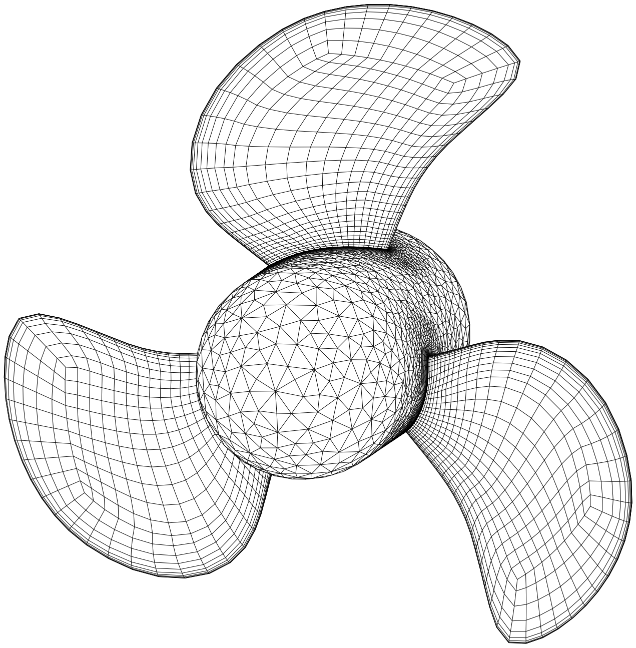

*Figure 4:* The domains on the surfaces of the blades and hub.

8. The region between the nested cylinders is meshed with a combination of structured and unstructured blocks.
9. The space between the inner cylindrical shell and the blocks on the blades and hub is filled with an unstructured block of pyramid and tetrahedral elements.

An example of the domains on the surface of the propeller is shown in Figure 4. The grid on the surrounding cylindrical shells are shown in Figure 5. In Figure 5, the hub is contained within the inner cylinder. Figure 6 shows the grid on the blades and hub when the hub is extended to the upstream and downstream boundaries. The hub grid is unstructured inside the inner cylinder, structured outside.

Because all blocks meeting boundaries are created by normal extrusion, each cell on a boundary is oriented very nearly perpendicularly to that boundary. This allows the flow solver, and in particular ANSYS CFX, to generate an accurate implementation of the boundary conditions when it calculates the flow.

In order to resolve the high curvature at the leading and trailing edges, the node spacing will typically need to be about $0.0001R$ or smaller, where $R$ is the propeller radius. If unstructured domains were used on the surfaces wrapping around the leading and trailing edges, 10,000 nodes or more would be required along the leading and trailing edges resulting in very large domains. Structured domains are much more tolerant of high aspect ratio cells,

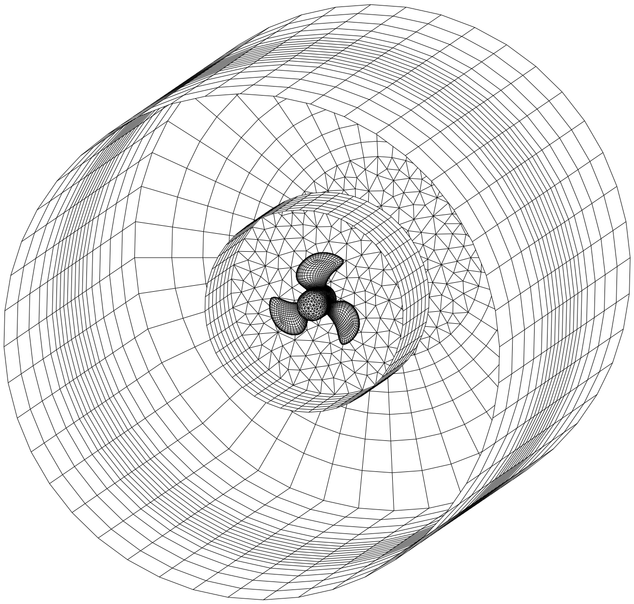

***Figure 5:*** *The grid on the propeller and the inner and outer cylinders. The upstream ends of the cylinders have been removed.*

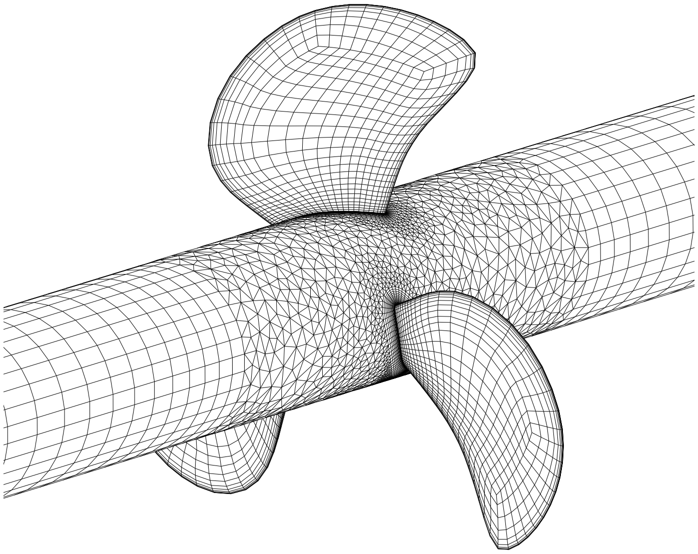

***Figure 6:*** *The grid on the blades and hub when the hub is extended to the upstream and downstream boundaries.*

provided that they are oriented to match the directions of the principal radii of curvature. Therefore structured domains were used for these surfaces. They have very high aspect ratio cells on the blade surface near the leading and trailing edges, but that diminishes naturally during extrusion so that the cells in the outer domains of the blade blocks have much smaller aspect ratios. Since the outer domains of the blade blocks form the surface of the unstructured block between the propeller and the shell, the extrusion should be carried far enough that the cell aspect ratios in the outer domains are no more than about 5.0; otherwise the pyramid cells used to match the structured and unstructured blocks will have very small internal angles which can lead to loss of accuracy and convergence problems in the flow solver.

The use of structured blocks on the surfaces in the centre of the blades constrains the number of nodes on their left edges to be the same as their right edges, and the number of nodes on their top edges to be the same as on their bottom edges. When a large number of nodes is needed to resolve the flow near the tip, one also gets a large number of nodes near the root. In addition, the number of nodes along the leading and trailing edges must be the same. It would be convenient if the numbers of nodes along the edges of these domains could be decoupled. This could be done if the central blade domains were unstructured. This has been tried, but led to problems during the extrusion of the blade blocks: structured and unstructured blocks cannot be extruded together, but when extruded separately they are not well enough constrained and result in deformities in the blocks where they join. This problem could probably be surmounted by creating small surfaces extruding normal to the blade from the edges of the central surface; they would be used to constrain the extrusion. However, that has not yet been tried.

Because Glyph2 does not allow the grid to be examined as it is being generated, the grid generation process has been split into six separate scripts. After each script has been run, the grid should be examined for quality before proceeding. The subsequent script will continue from where the preceding script stopped. The basic tasks performed by the six scripts are as follows:

**`prop1.glf`**
Reads the geometry of the propeller from an IGES file and makes domains on the reference blade.

**`prop2.glf`**
Obtains the domains on the reference blade created by `prop1.glf`, then extrudes them to make structured blocks covering the reference blade.

**`prop3.glf`**
Obtains the structured blocks on the reference blade created by `prop2.glf`; if necessary, trims the hub at the upstream and downstream ends of the inner cylinder; makes an unstructured domain on the portion of the hub not covered by the blade

blocks; and rotates all domains and blocks so that the whole hub and all blades are covered.

**prop4.glf**
: Obtains the structured blade blocks and hub domains created by `prop3.glf`, then extrudes the hub domains to make a block of prism elements covering the hub. The block created by the extrusion is called the hub inflation layer.

**prop5.glf**
: Makes blocks in the region between the inner and outer cylinders.

**prop6.glf**
: Obtains all the blocks created by the earlier scripts, then makes an unstructured block filling the space between the inner cylindrical shell and the blocks on the blades and hub. Boundary conditions are applied and the grid is saved in Pointwise and ANSYS CFX formats.

Each of the scripts is described in more detail in Sections 4.4–4.9. Instructions on how to use the scripts are given in Section 5.

## 4.4 Making the blade domains: script prop1.glf

The first propeller script, `prop1.glf`, creates domains covering the reference blade. It begins by reading the IGES file defining the propeller geometry, then creates connectors along the edges of each of the five surfaces on the reference blade. New connectors along the leading edges are created and used to bisect the surfaces around the leading edge, trailing edge and tip; this ensures that a concentration of nodes can be maintained near the regions of high curvature along the leading and trailing edge.

To aid in the extrusion of the hub domains by script `prop4.glf`, connectors are also included at the top of the hub inflation layer. For one side of the blade this results in 18 connectors which divide the blade into seven regions: see Figure 7 which also assigns numbers to the connectors for easy reference. Over the whole blade there are 14 regions and 31 connectors in total (connectors 4, 7, 11, 14, and 18 are shared by the front and the back of the blade; each of the other connectors has a counterpart on the other side of the blade).

Each of the fourteen regions will be covered by a structured domain. Since opposite edges of a structured block must have the same number of nodes, the dimensions of the connectors are fully specified by four numbers:

$N_{tip}$ the dimension of the connector passing through the tip: connector 18. It is also the dimension of connectors 2, 9, 16 and the corresponding connectors on the other side of the blade.

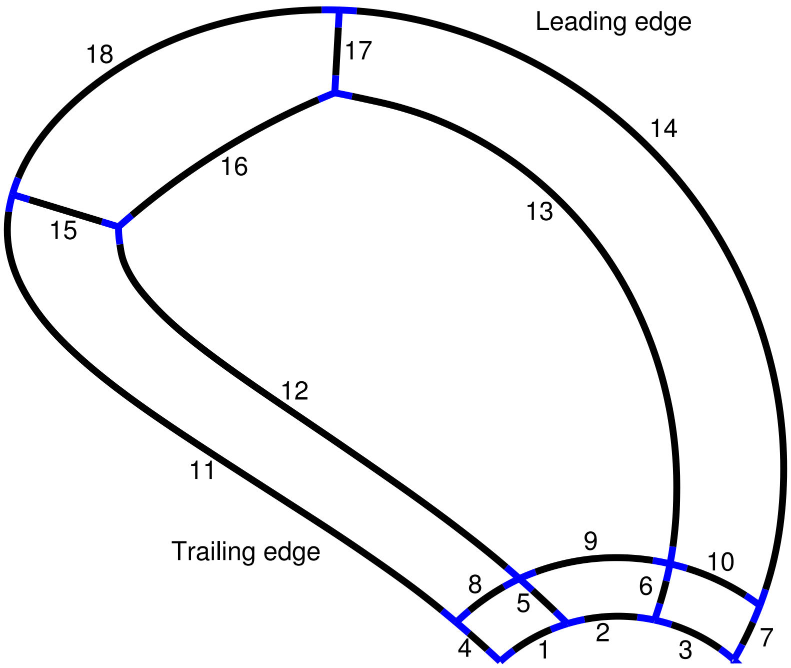

***Figure 7:*** *Connectors on the pressure side of the reference blade.*

$N_{vert}$ the dimension of connectors 11, 12, 13, 14 and their corresponding connectors on the other side of the blade.

$N_h$ the number of nodes across the hub inflation layer. It is the dimension of connectors 4, 5, 6, 7 and the corresponding connectors on the other side of the blade.

$N_e$ the dimension of edges 1, 3, 8, 10, 15, 17 and the corresponding connectors on the other side of the blade.

The values of $N_{tip}$, $N_{vert}$ and $N_e$ are set explicitly. Ideally the nodes on the connectors rising vertically through the hub inflation layer would be distributed using a geometric distribution with spacing of the first two nodes equal to the hub wall distance, $s_h$, and each pair of nodes having spacing increased by the hub growth factor, $g_h$ (the ratio of sizes of neighbouring cells as one proceeds upwards through the hub inflation layer). The height of the hub inflation layer would then be

$$h_h = s_h \frac{g_h^{N_h-1} - 1}{g_h - 1} \tag{2}$$

and the spacing at the upper end of the connector would be

$$s_{hu} = \frac{h_h(g_h - 1) + s_h}{g_h} \tag{3}$$

Since it is usually more useful to specify $h_h$ rather than $N_h$, we invert Equation (2) to provide an expression for $N_h$:

$$N_h = 1 + \left\{ \frac{\log\left(1 + \frac{h_h(g_h-1)}{s_h}\right)}{\log(g_h)} \right\} \tag{4}$$

with the curly brackets denoting "the closest integer to". Because of the rounding of $N_h$, the distribution of nodes will no longer be strictly geometric. Instead a tanh distribution is used with spacing at the hub equal to $s_h$ and spacing at the upper edge of the hub inflation layer equal to $s_{hu}$ as given by Equation (3).

However, if the nodes on these connectors are distributed in this way, then during the extrusion of the blade domains the smoothing used to keep the extrusion robust will tend to increase the spacing at the hub as the extrusion proceeds; typically the outer extruded domain will have wall spacing much larger than $s_h$. This can be avoided by setting the normal volume smoothing during the extrusion to a very low value, but that makes the extrusion less robust and inhibits the decrease of the aspect ratios of the cells as the extrusion proceeds.

There is also another problem caused by the blade extrusion: after extrusion, the spacing near the hub on the vertical connectors of the outer blade domains will often differ significantly. This will certainly be true if there is high rake. The rake causes the blade to be tipped forward or aft so that the normals on one side of the blade point into the hub while on the other side they point out of the hub. After extrusion the node spacing on the side where the normals point into the hub will be smaller than the spacing on the other side. The difference in spacing can cause problems for the hub extrusion (see below) which must match the nodes on the blade.

Both these problems are avoided by choosing the initial distribution of nodes on these connectors so that the cell spacing is nearly constant and equal to $s_{hu}$. After the extrusion, in script `prop2.glf`, the node distribution is adjusted so that the wall spacing and growth rate are correct.

Since there are 36 connectors, there are 72 end-spacings to be set. The number of independent spacings is reduced using the following rules:

1. When two connectors both lie on the same smooth curve, the spacings where their ends meet will be the same. Thus, for example, the spacings where connectors 8 and

9 meet will be equal. One end of connectors 5 and 12 also meet at the same point, but their spacings need not equal that of connectors 8 and 9 because they do not lie on the same smooth curve. However, the spacings of connectors 5 and 12 at this point must be the same.

2. The spacings where connectors 12, 15 and 16 meet are all equal. Likewise for the spacings where connectors 13, 16 and 17 meet and for the corresponding points on the other side of the blade.

3. Where a connector meets the hub vertically, the spacing will equal $s_h$, the hub wall spacing. The other end of the connector, at the edge of the hub inflation layer, will have spacing $s_{hu}$. This applies to connectors 4, 5, 6, 7 and the corresponding connectors on the other side of the blade.

4. The relative spacings at the ends of connector 16 are the same as the relative spacings at the ends of connector 18: i.e. if $L_{16}$ and $L_{18}$ are the arclengths of connector 16 and 18, and if $s_{r16}$ and $s_{r18}$ are their spacings at their right ends, then $s_{r16} = s_{r18}L_{16}/L_{18}$. Similarly for the spacing at the left end.

5. The spacings at the ends of connectors 1 and 8 lying on the trailing edge are the same.

6. The spacings at the ends of connectors 3 and 10 lying on the leading edge are the same.

7. The relative spacings at the right ends of connectors 1, 8 and 15 are all the same.

8. The relative spacings at the left ends of connectors 3, 10 and 17 are all the same.

Using these rules, all the spacings can be determined from the connector dimensions, $s_h$, $g_h$, $h$ and the following parameters:

| | |
|---|---|
| $s_{le}$ | The spacing at the end of connector 18 lying on the leading edge. |
| $s_{te}$ | The spacing at the end of connector 18 lying on the trailing edge. |
| $s_{le-tip}$ | The spacing at the end of connector 17 lying on the leading edge. |
| $s_{te-tip}$ | The spacing at the end of connector 15 lying on the trailing edge. |
| $s_{le-root}$ | The spacing at the end of connector 3 lying on the leading edge. |
| $s_{te-root}$ | The spacing at the end of connector 1 lying on the trailing edge. |

The script `prop1.glf` performs the following steps.

1. The values of the attributes are obtained from the files `attributes.glf` and `default-attributes.glf`. An error is written if `attributes.glf` does not exist in the directory in which Pointwise is being run.

2. The propeller geometry is imported by reading the IGES input file. All the propeller surfaces are rendered as shaded surfaces. The five blade surfaces are added to a separate layer (`Blade Surfaces`) so that they can easily be made invisible. The hub surface is also added to a separate layer (`Hub Surface`).

   **Requirements:**

   (a) Attribute `prop_iges_file`: The name of the IGES file containing the propeller geometry.

3. The node and connector tolerances are set to match the propeller accuracy used by `smooth-prop`. This ensures that connectors on the edges of the blade surfaces and along the blade/hub intersection are properly merged.

   **Requirements:**

   (a) Attribute `prop_accuracy`: The propeller accuracy.

4. A connector is made along the leading/trailing edge through the tip; this is the curve in the tip surface with $\eta = 0.5$.

   **Requirements:**

   (a) Attribute `num_nodes_tip`: The number of nodes, $N_{tip}$.

   (b) Attribute `le_spacing`: The spacing at the end on the leading edge, $s_{le}$. If the value is zero, the spacing is set to the average spacing: i.e. its arclength divided by $N_{tip} - 1$.

   (c) Attribute `te_spacing`: The spacing at the end on the trailing edge, $s_{te}$. If the value is zero, the spacing is set to the average spacing of the connector.

5. Connectors are created along the tops of the central surfaces on the pressure and suction sides of the reference blade. The number of nodes on these connectors is equal to $N_{tip}$. The relative spacings at the ends are the same as the relative spacings for connector 18.

6. Connectors are created along the vertical edges of the central, leading edge and trailing edge surfaces. Each connector has the same number of nodes.

   **Requirements:**

   (a) Attribute `num_nodes_vert`: The number of nodes, $N_{vert}$.

7. The number of nodes across the hub inflation layer, $N_h$, is set to

$$N_h = 1 + \left\{ \frac{h_h}{s_{hu}} \right\} = 1 + \left\{ \frac{h_h g_h}{h_h(g_h - 1) + s_h} \right\} \qquad (5)$$

**Requirements:**

(a) Attribute `hub_wall_spacing`: The hub wall spacing, $s_h$.

(b) Attribute `hub_layer_height`: The height of the hub inflation layer, $h_h$.

(c) Attribute `hub_growth_rate`: The growth rate of cells in the hub inflation layer, $g_h$.

8. Each vertical connector is split into two at the top of the hub inflation layer. The value of the spacing at the top of the inflation layer, $s_{hu}$, is determined using Equation (3). The lower portion of each connector is assigned $N_h$ nodes.

   The spacing at the lower end of the upper portion of the connector is also set to $s_{hu}$. The spacing at the upper end of the upper portion of the connector is set to be equal to the spacing at the end of the connector across the top of the central region. This means that the cells in the upper corners of the central domains will be nearly square.

9. Connectors 1, 3, 8, 10, 15 and 17 are made, as well as the corresponding connectors on the other side of the blade, each having $N_e$ nodes. Their end spacings are determined as described above.

   **Requirements:**

   (a) Attribute `num_nodes_around_le`: The number of nodes, $N_e$.

   (b) Attribute `le_spacing_root`: The node spacing on connector 3 at the end lying on the leading edge, $s_{le-root}$.

   (c) Attribute `le_spacing_tip`: The node spacing on connector 17 at the end lying on the leading edge, $s_{le-tip}$.

   (d) Attribute `te_spacing_root`: The node spacing on connector 1 at the end lying on the trailing edge, $s_{te-root}$.

   (e) Attribute `te_spacing_tip`: The node spacing on connector 15 at the end lying on the trailing edge, $s_{te-tip}$.

10. Connectors 2 and 9 and their counterparts on the other side of the blade are made.

11. The fourteen structured blade domains are created using the connectors.

12. The eight domains which do not include the hub inflation layer are smoothed. If this is not done, the spacing along the leading or trailing edge can be poor. This is because the node spacing at the leading edge is roughly proportional to chord length; therefore, if the chord length increases significantly, the node spacing at the leading edge will also increase significantly causing poor resolution at the leading edge. Smoothing decreases the dependency of the leading edge spacing on chord length.

The smoothing also improves the gridding near the upper corners of the central domains.

**Requirements:**

(a) Attribute `nsmooth_blade`: The number of smoothing iterations.

(b) Attribute `blade_smoothing_edge_control`: The method of edge control for the smoothing of the blade domains.

13. Domains on the pressure and suction sides which contain a common portion of the leading and trailing edge are joined to create a single domain: for example, the domain bounded by connectors 8, 11, 12 and 15 is joined to its counterpart on the other side of the blade. The fourteen domains are reduced to nine and connectors 4, 7, 11, 14 and 18 are no longer required.

14. Each of the connectors 1, 3, 8, 10, 15 and 17 is joined with its counterpart on the other side of the blade to form a single connector. If this is not done, when the face is extruded there can be kinks where the connectors join.

15. The domains are smoothed again to even out any discrepancies in cell size at the leading and trailing edges:

    **Requirements:**

    (a) Attribute `nsmooth_blade_joined`: The number of smoothing iterations.

16. All the connectors and domains are placed in a separate layer (`Reference Blade Domains`)

17. A report on the quality of the blade domains is written to the Pointwise Messages window.

18. The current state is saved in the file `prop1.pw`.

19. Variable values needed by subsequent scripts are saved in the files `radius.dat` and `blade-domains.dat`.

Figure 8 shows an example of the display when `prop1.glf` returns.

The grid can be quite sensitive to the amount of smoothing on the blade domains. If it is too small, the node spacing along the leading and trailing edges can be too large. If it is too large, the smoothing will sometimes cause the grid to deteriorate, presumably because the changes in the propeller surface are very rapid near the leading and trailing edges. These problems seem to be exacerbated by highly skewed blades.

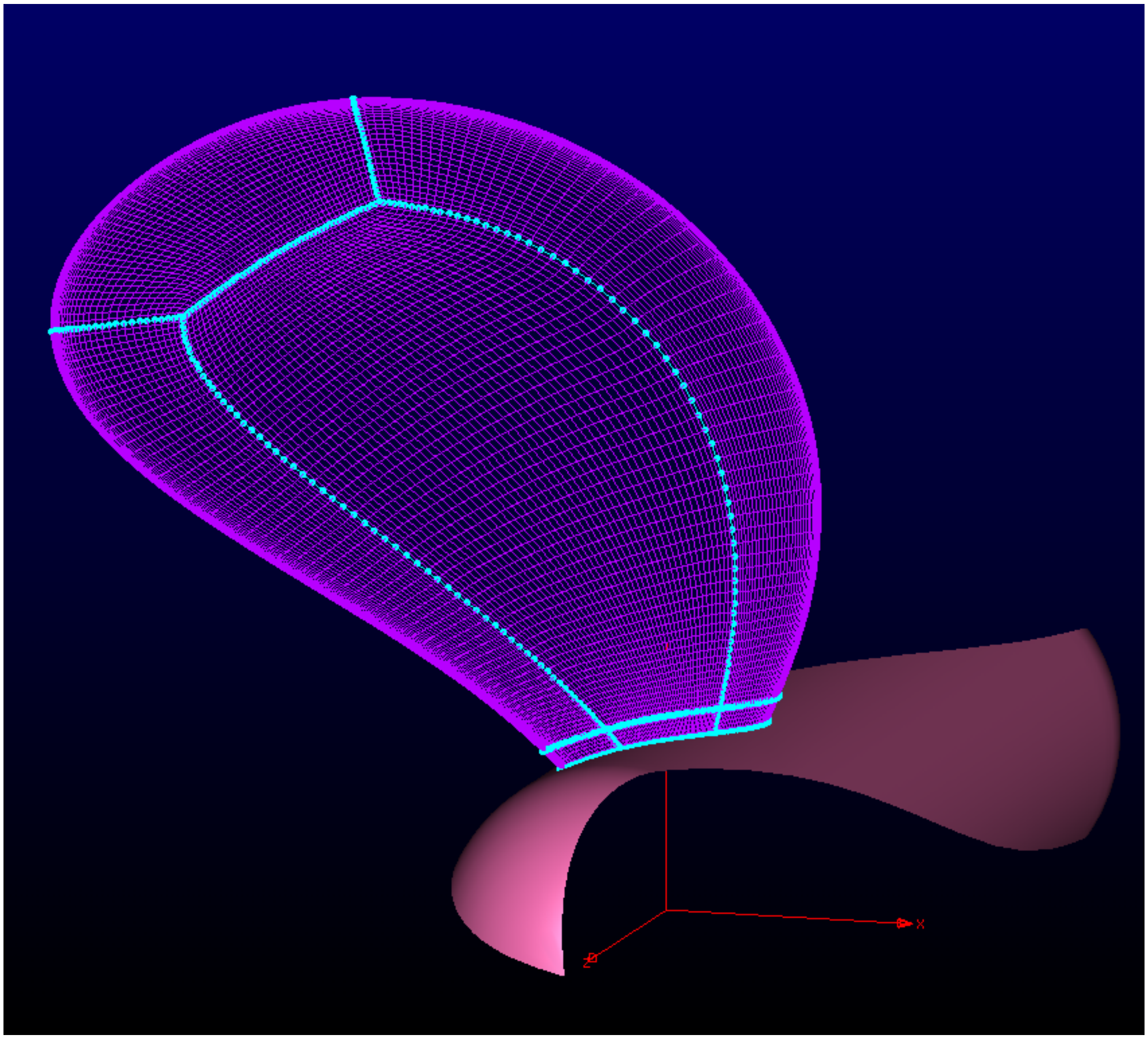

***Figure 8:*** *An example of the display after script `prop1.glf` is finished.*

## 4.5 Making the blade blocks: script prop2.glf

The script `prop2.glf` performs the following steps.

1. The data saved by `prop1.glf` are read.

2. The number of nodes in the blade inflation layer, $N_b$, is determined from its height, $h_b$, blade wall spacing, $s_b$, and the growth rate, $g_b$:

$$N_b = 1 + \left\{ \frac{\log\left(1 + \frac{h_b(g_b - 1)}{s_b}\right)}{\log(g_b)} \right\} \tag{6}$$

**Requirements:**

(a) Attribute `blade_wall_spacing`: The blade wall spacing, $s_b$ (used as the initial step size).

(b) Attribute `blade_growth_rate`: The factor by which the step size is increased in each extrusion step, $g_b$.

(c) Attribute `blade_layer_height`: The height of the blade inflation layer, $h_b$.

3. The nine blade domains are extruded hyperbolically for $N_b - 1$ steps; the extrusion from the connectors on the blade/hub intersection is required to follow the hub surface.

   **Requirements:**

   (a) Attribute `extrusion_smoothing`: A normal explicit smoothing factor to be used during the extrusion.

   (b) Attribute `extrusion_KB_smoothing`: The Kinsey-Barth smoothing factor to be used during the extrusion.

4. The value of $N_h$ is changed so that it is given by Equation (4), then each of the connectors rising vertically through the hub inflation layer is adjusted so that it has $N_h$ nodes with spacing $s_h$ at the hub and $s_{hu}$ at its upper end. The interior nodes of the blocks in the hub inflation layer are adjusted automatically.

5. The maximum aspect ratios of the outermost faces of the extruded blocks are reported to the Messages window. After the extrusion, the cells on the outer extruded domains can be quite elongated (high aspect ratio) due to the small node spacing across the leading and trailing edges relative to the node spacing along the leading and trailing edges. High aspect ratios will cause small angles between edges in the tetrahedra of the block between the blade blocks and the inner cylinder. The aspect ratio of the cells can be improved by increasing the height of the inflation layer or by increasing the number of nodes along the leading and trailing edges, $N_{vert}$.

6. The blocks are assigned to two new layers called `Next to Hub Blocks` and `Reference Blade Blocks`.

7. The current state is saved in the file `prop2.pw`.

8. Variables needed by subsequent scripts are saved in `blade-blocks.dat`.

Figure 9 shows an example of the display when `prop2.glf` returns.

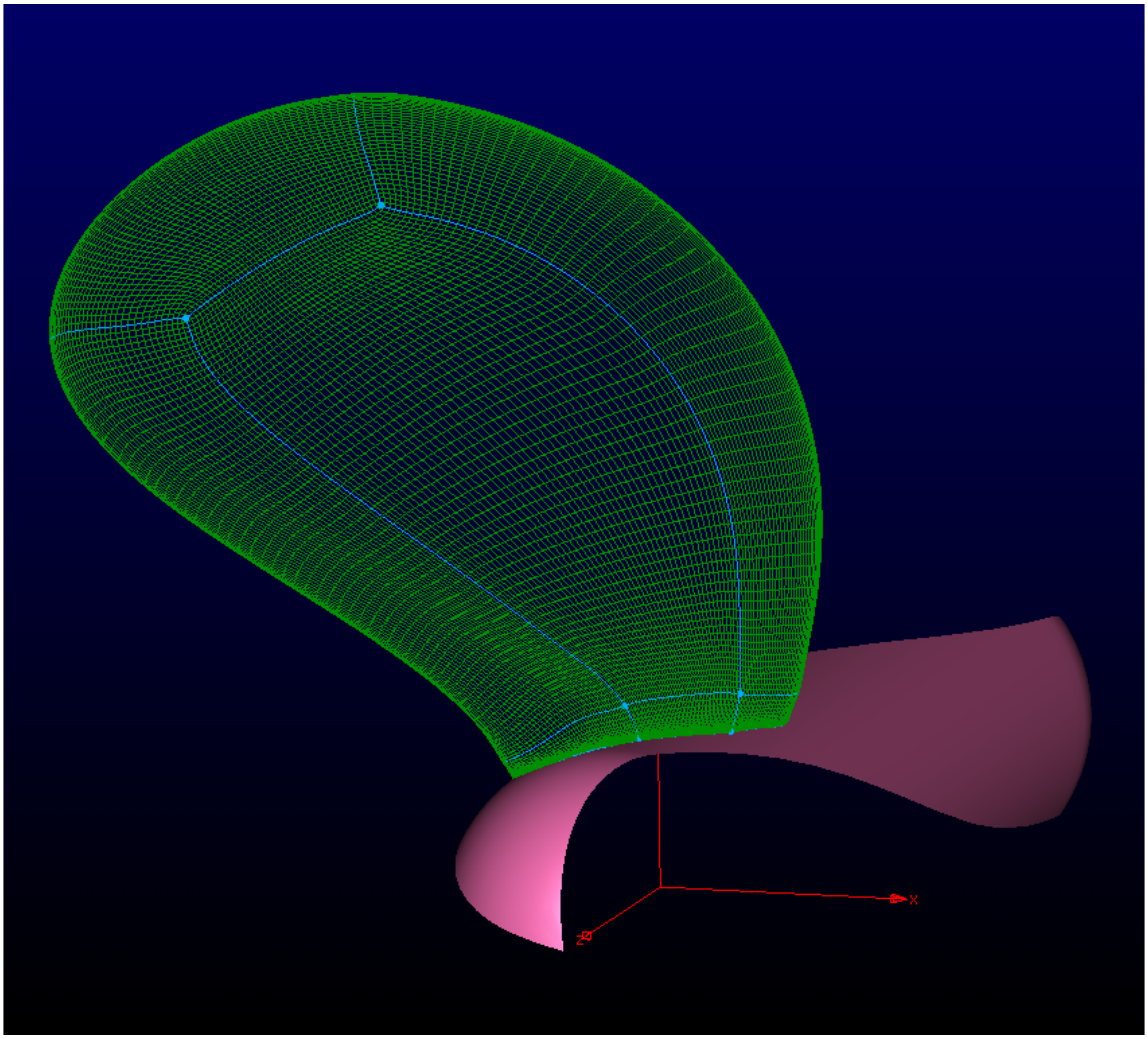

***Figure 9:*** *An example of the display after script* `prop2.glf` *is finished.*

## 4.6 Making the hub domain: script prop3.glf

The script `prop3.glf` performs the following steps.

1. The data saved by `prop2.glf` are read.
2. If the upstream end of the hub is cut by the plane $z = z_{imin}$, the hub is trimmed by the plane; similarly, the hub is trimmed at $z = z_{imax}$ if necessary.

   **Requirements:**

   (a) Attribute `zmin_inner`: The value of $z_{imin}/R$.

   (b) Attribute `zmax_inner`: The value of $z_{imax}/R$.
3. An unstructured grid is made on the hub surface.

**Requirements:**

   (a) Attribute `hub_edge_spacing`: The node spacing on the hub edges.
   (b) Attribute `hub_boundary_decay`: The boundary decay factor for the hub domains. Values closer to 1.0 cause the sizes of the cells on the boundary to be respected further into the interior of the block.

4. $Z-1$ rotated copies of all the connectors, domains and blocks are made so that the full hub surface and all the blades are covered ($Z$ is the number of blades).

5. The domains on the propeller surfaces as well as the domains joining the hub and blade blocks are assigned to new layers (`Blade Domains`, `Hub Domains` and `Lower Outer Domains`).

6. The current state is saved in the file `prop3.pw`.

7. Variables needed by subsequent scripts are saved in the file `hub-domains.dat` and `hub-trimmed.dat`.

Figure 10 shows an example of the display when `prop3.glf` returns.

## 4.7 Making the hub block: script prop4.glf

The script `prop4.glf` performs the following steps.

1. The data saved by `prop3.glf` are read.

2. The hub domains are extruded hyperbolically to generate unstructured prismatic blocks covering the hub inflation layer. The extrusion is required to match the faces in the blade blocks which are adjacent to the hub. If the hub was trimmed at $z = z_{imin}$, the upstream faces of the extruded blocks are also required to lie in the plane $z = z_{imin}$; similarly, the downstream faces are constrained to lie in $z = z_{imax}$ if the hub was trimmed there.

   Because the extrusion tries to proceed in the direction normal to the hub, if the outermost faces of these blocks meet the hub at a fairly large angle, the extruded nodes will begin to bunch up near these faces. Large numbers of smoothing iterations are sometimes necessary to combat the bunching and make the extrusion stable.

   **Requirements:**

   (a) Attribute `nsmooth_hub`: The number of smoothing iterations.

3. The quality of the hub block is reported. The quality of each domain to be used as the inner surface of the unstructured block between the blades and hub and the inner cylinder is also reported. (These domains are the outermost domains of the hub block and of the blade blocks not adjacent to the hub.)

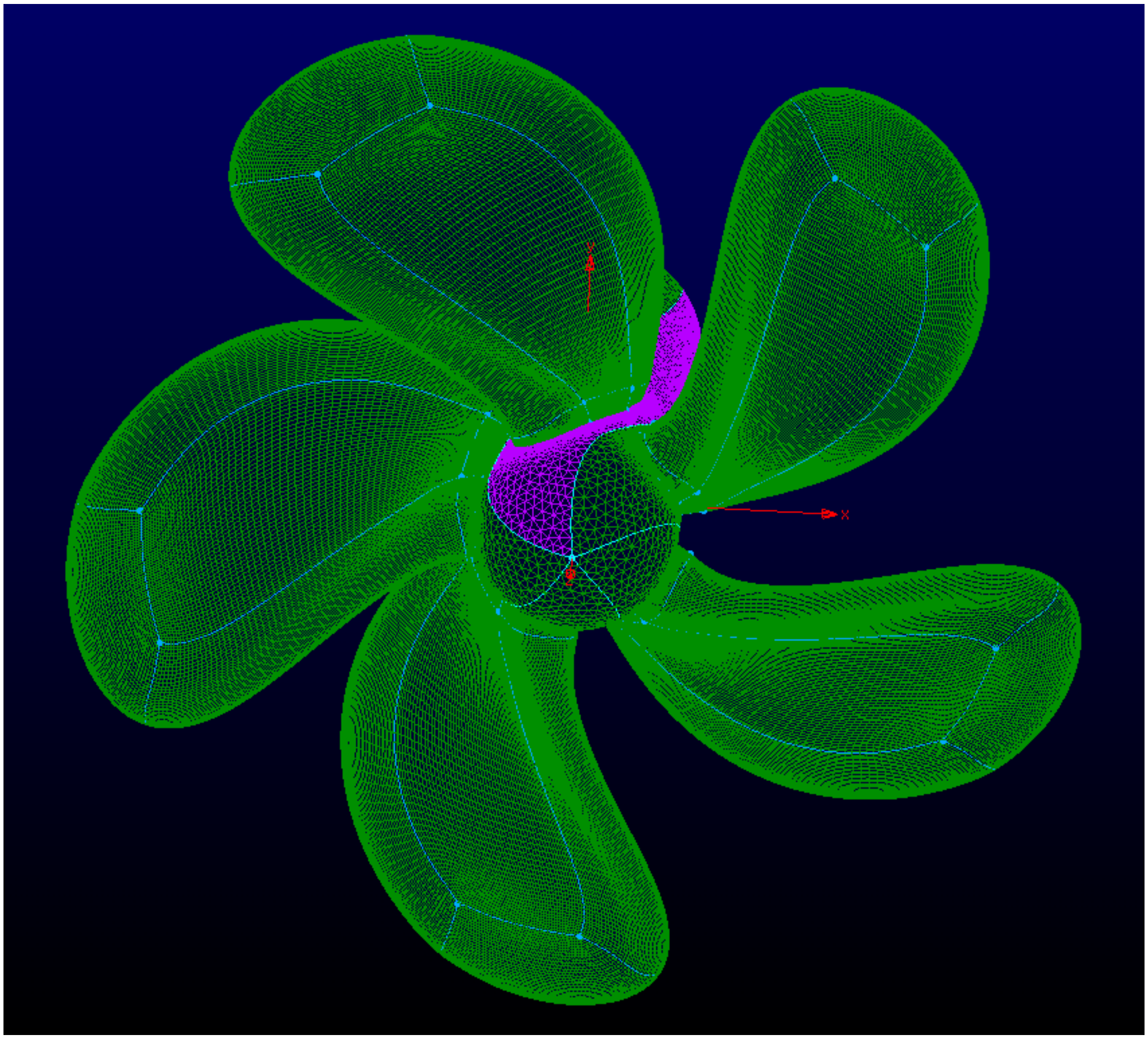

***Figure 10:*** *An example of the display after script* `prop3.glf` *is finished.*

4. The hub block and the outer domains of the hub and blade blocks are assigned to new layers (`Hub Block` and `Interior Domains`).
5. The current state is saved in the file `prop4.pw`.
6. Variables needed by subsequent scripts are saved in the file `hub-blocks.dat`.

Figure 11 shows an example of the display when `prop4.glf` returns.

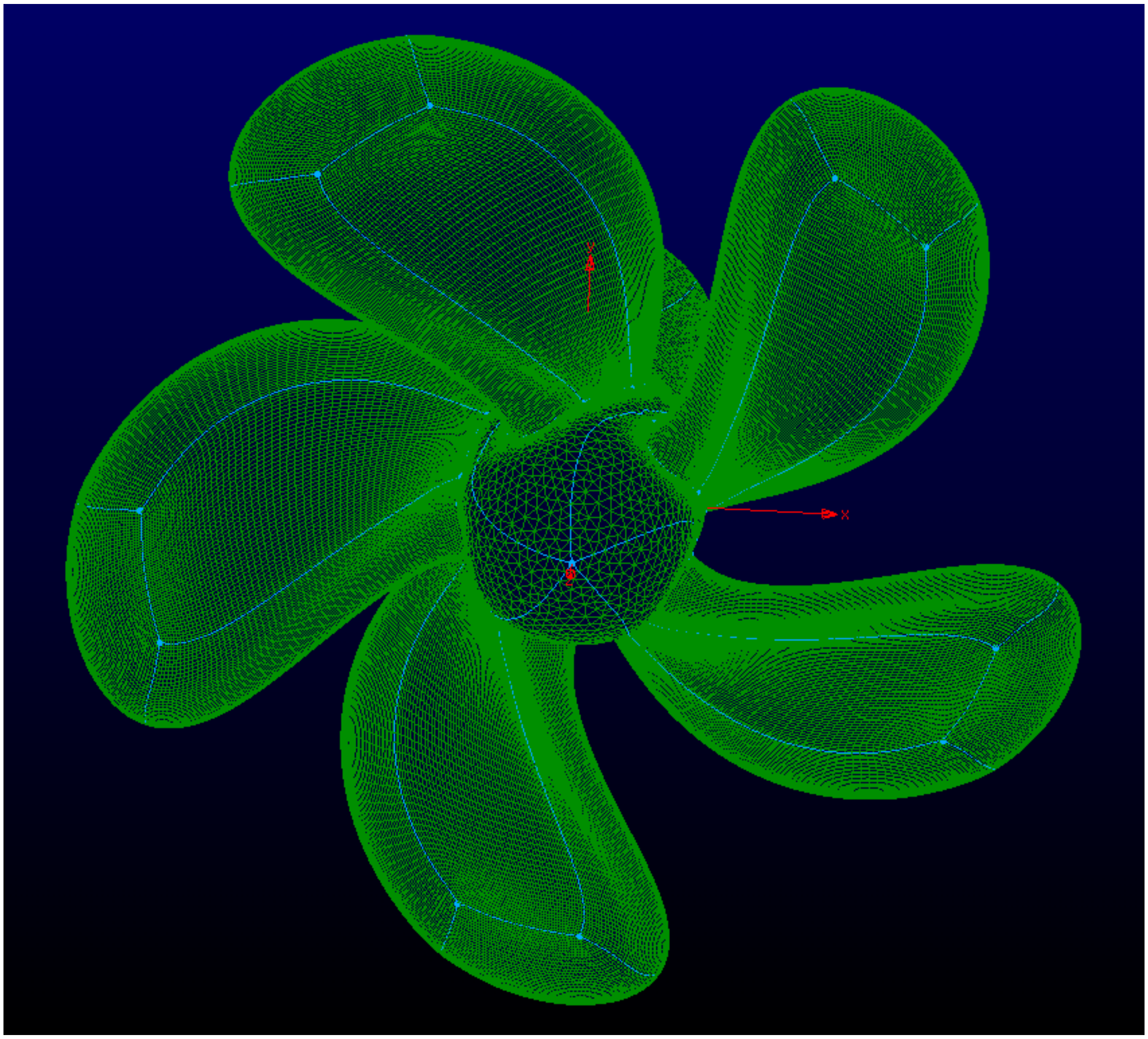

***Figure 11:*** *An example of the display after script `prop4.glf` is finished.*

## 4.8 Making the outer cylindrical blocks: script prop5.glf

The script `prop5.glf` creates blocks between two concentric cylinders surrounding the propeller. The outer cylinder has radius $R_o$ and extends from $z = z_{omin}$ to $z = z_{omax}$; the inner cylinder has radius $R_i$ and extends from $z = z_{imin}$ to $z = z_{imax}$. The outer cylinder marks the outer boundary of the region of flow.

The number and type of the blocks depends on whether the hub extends past the planes $z = z_{imin}$ and/or $z = z_{imax}$. As explained in Section 4.3, if the plane $z = z_{imin}$ intersects the hub, the hub is extended all the way to the upstream boundary; similarly, it is extended to the downstream boundary if it intersects $z = z_{imax}$. There is always a structured block in the region $\{z_{imin} \leq z \leq z_{imax};\ R_i \leq r \leq R_o\}$. If the hub extends to the upstream boundary, its radius is constant and equal to $R_{hu}$ for $z \leq z_{imin}$; there will then be three blocks in the region $z \leq z_{imin}$:

1. a structured block in the inflation layer of the hub: $R_{hu} \leq r \leq R_{hu} + h_h$;
2. an unstructured block in the region $R_{hu} + h_h \leq r \leq R_i$; and
3. a structured block in the region $R_i \leq r \leq R_o$.

If the upstream end of the hub is enclosed within the inner cylinder, there will only be two blocks:

1. an unstructured block in the region $0 \leq r \leq R_i$ and
2. a structured block in the region $R_i \leq r \leq R_o$.

Each of the blocks is made by extrusion parallel to the propeller axis; the cells of the two unstructured blocks are all triangular prisms.

Similarly, there will be either two or three blocks in the region $z \geq z_{imax}$ depending on whether the hub extends to the downstream boundary.

The ANSYS CFX boundary conditions are most accurate if cells adjacent to a boundary are aligned perpendicular to it; these blocks ensure that this is true for all the outer boundaries. The inner cylinder also allows an extra measure of control over the size of the elements close to the propeller.

The script `prop5.glf` performs the following steps.

1. The data saved by `prop1.glf` and `prop3.glf` are read.
2. A connector is created between $(r,\theta,z) = (R_i, 0, z_{imin})$ and $(R_o, 0, z_{imin})$. Its dimension is $N_c$ and end-spacings are $s_i$ and $s_o$ where $s_i$ is given and

   $$N_c = \left\{ \frac{\ln(R_o/R_i)}{\ln(\alpha)} \right\} + 1; \qquad \alpha = \frac{2R_i + s_i}{2R_i - s_i}; \qquad s_o = \frac{s_i R_o}{R_i} \tag{7}$$

   These choices make the radial node spacing increase with radius roughly in proportion to the circumferential node spacing.

   **Requirements:**

   (a) Attribute `outer_radius`: The value of $R_o/R$ where $R$ is the propeller radius.
   (b) Attribute `inner_radius`: The value of $R_i/R$.
   (c) Attribute `zmin_inner`: The value of $z_{imin}/R$.

   (d) Attribute `cell_size`: The value of $s_i/R$ where $s_i$ is the size of cells on the inner cylinder.

3. A structured annular domain is made by rotational extrusion of the connector through 360 degrees. The number of extrusion steps, $N_\theta - 1$ (one less than the number of nodes), is chosen so that the cell size at $R_i$ is approximately $s_i$:

$$N_\theta = \left\{ \frac{\pi R_i}{s_i} \right\} + 1 \tag{8}$$

4. An unstructured domain is made in the plane $z_{imin}$ for $r \leq R_i$ using the inner connector of the annular domain as its outer boundary. If the hub was trimmed at $z_{imin}$, it will extend to the top of the hub inflation layer ($r \geq R_{hu} + h_h$); otherwise it will extend to $r = 0$.

5. A connector is made between $(r, \theta, z) = (R_o, 0, z_{omin})$ and $(R_o, 0, z_{imin})$. Its end-spacing at $z = z_{imin}$ is $s_i$. If the value of the spacing at $z_{omin}$ is zero, then it is set to

$$s_{zmin} = \frac{s_o(z_{imin} - z_{omin})}{R_o - R_i} \tag{9}$$

   which will cause the sizes of the cells to increase at roughly the same rate in the axial direction as in the radial direction. The dimension of the connector is set to

$$N_{zmin} = \left\{ \frac{\ln(s_{zmin}/s_i)}{\ln\left(\frac{z_{imin} - z_{omin} - s_i}{z_{imin} - z_{omin} - s_{zmin}}\right)} \right\} + 1 \tag{10}$$

   which ensures that the node spacings increase nearly geometrically.

   **Requirements:**

   (a) Attribute `zmin_outer`: The value of $z_{omin}/R$.

   (b) Attribute `zmin_spacing`: The value of $s_{zmin}/R$.

6. The domains in the plane $z = z_{imin}$ are extruded parallel to the $z$ axis using the node spacing of the connector created in step 5 to determine the size of each extrusion step. The domains include those just created and, if the hub extends to the upstream boundary, the domains in the hub blocks created by `prop4.glf` which lie in $z = z_{imin}$ (i.e. the upstream faces of the blocks in the hub inflation layer).

7. The domain created in step 3 is extruded from $z_{imin}$ to $z_{imax}$ parallel to the propeller axis. The number of extrusion steps is $\{(z_{imax} - z_{imin})/s_i\}$ so that the cell size will be approximately $s_i$. This creates the annular structured block in the region $\{z_{imin} \leq z \leq z_{imax};\ R_i \leq r \leq R_o\}$.

8. An unstructured domain is made in the plane $z_{imax}$ for $r \leq R_i$ using the inner connector of the upstream domain of the block created in step 7 as its outer boundary. If the hub was trimmed at $z_{imax}$, the unstructured domain will extend to to the top of the hub inflation layer ($r \geq R_{hd} + h_h$); otherwise it will extend $r = 0$.

9. A connector is made between $(r, \theta, z) = (R_o, 0, z_{imax})$ and $(R_o, 0, z_{omax})$. Its end-spacing at $z = z_{imax}$ is $s_i$. If the value of the spacing at $z_{omax}$ is zero, then it is set to
$$s_{zmax} = \frac{s_o(z_{omax} - z_{imax})}{R_o - R_i} \tag{11}$$
which will cause the sizes of the cells to increase at roughly the same rate in the axial direction as in the radial direction. The dimension of the connector is set to
$$N_{zmax} = \left\{ \frac{\ln(s_{zmax}/s_i)}{\ln\left(\frac{z_{omax} - z_{imax} - s_i}{z_{omax} - z_{imax} - s_{zmax}}\right)} \right\} + 1 \tag{12}$$
which ensures that the node spacings increase nearly geometrically.

   **Requirements:**

   (a) Attribute `zmax_outer`: The value of $z_{omax}/R$.

   (b) Attribute `zmax_spacing`: The value of $s_{zmax}/R$.

10. The domains in the plane $z = z_{imax}$ are extruded parallel to the $z$ axis using the node spacing of the connector created in step 9 to determine the size of each extrusion step. The domains include the upstream domain of the block created in step 7, the unstructured domain created in step 8 and, if the hub extends to the downstream boundary, the domains in the hub blocks created by `prop4.glf` which lie in $z = z_{imax}$ (i.e. the downstream faces of the blocks in the hub inflation layer).

11. The current state is saved in the file `prop5.pw`.

12. The newly created domains and blocks are assigned to new layers called `Inflow`, `Outflow`, `Inner Shell`, `Outer Shell` and `Shell Blocks`.

13. Variables needed by subsequent scripts are saved in `shell-blocks.dat`.

Figure 12 shows an example of the display when `prop5.glf` returns. In this case the hub does not extend to the near boundary.

## 4.9 Making the unstructured block: script prop6.glf

The script `prop6.glf` performs the following steps.

1. The data saved by `prop4.glf` and `prop5.glf` are read.

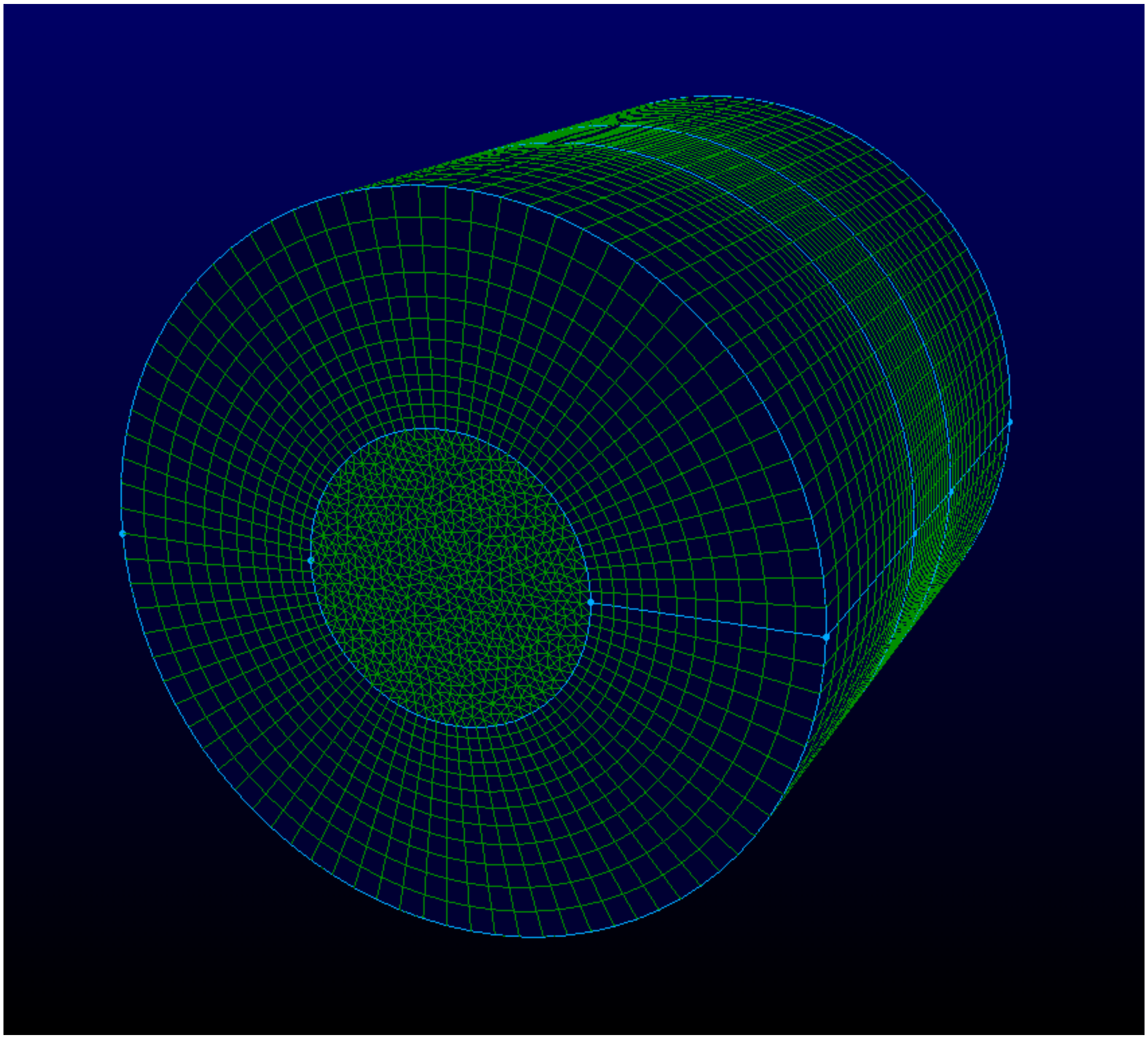

***Figure 12:*** *An example of the display after script `prop5.glf` is finished.*

2. The main unstructured block is made; it fills the space between the blade and hub blocks and the domains covering the inner cylinder.

   **Requirements:**

   (a) Attribute `main_boundary_decay`: The decay factor for the main block. Values closer to 1.0 cause the sizes of the cells on the boundary to be respected further into the interior of the block.

   (b) Attribute `pyramid_aspect_ratio`: The aspect ratio for the height of pyramids. A value of `Default` (equivalent to 0.5) is allowed. Smaller values will sometimes increase minimum interior angles in the main grid. The quality of the main grid can be quite sensitive to the pyramid aspect ratio, especially when the cells on the extruded faces of the structured blade blocks are highly skewed. For highly skewed blades the pyramid aspect ratio may need to be set as low as 0.2

to prevent very small internal angles in the tetrahedra which abut the pyramids.

    (c) Attribute `num_block_iter`: The number of iterations used when making the main block. Using more than one iteration takes significantly more time, usually with minimal improvements in the quality of the grid.

3. The quality of the main block is reported.
4. Boundary conditions are assigned. When ANSYS CFX is used, these do not actually cause specific boundary conditions to be used by the flow solver; however, they do serve to identify the regions on which similar boundary conditions should be applied.
5. The unstructured block is assigned to a new layer (`Main Block`).
6. The grid is saved in both ANSYS CFX and in Pointwise formats.

   **Requirements:**

    (a) Attribute `CFX_file`: The name of the output file containing the propeller grid in ANSYS CFX format.

    (b) Attribute `PW_file`: The name of the output file containing the propeller grid in Pointwise format.

The display is not changed by `prop6.glf`.

# 5 Running the scripts

This section provides detailed instructions on how to run the Glyph2 scripts starting with the IGES representation of the propeller. Reference 6 describes how the IGES file can be created using the program `smooth-prop`.

Before running the scripts you must first create the file `attributes.glf` in the directory in which you are running Pointwise. It must define the following three attributes:

1. `prop_iges_file`, the name of the IGES file defining the propeller geometry;
2. `CFX_file`, the output file for the grid in ANSYS CFX format; and
3. `PW_file`, an output file for the grid in Pointwise format.

In addition, `attributes.glf` can be used to redefine the values of any of the attributes given default values by `default-attributes.glf` (see Annex A).

The file `attributes.glf` is Tcl source code that sets the values of the attributes using lines with the following format:

```
set attribute value
```

For example, here is a minimal `attributes.glf` file:

```
set prop_iges_file P4119.igs; # IGES input file
set CFX_file P4119.grd;       # Grid file for ANSYS CFX
set PW_file P4119.pw;         # Grid file for Pointwise
```

The # symbol denotes the start of a comment. However, if it occurs on the same line as the `set` command, there must be a semi-colon after the attribute value.

## 5.1 Running prop1.glf

When `attributes.glf` has been created, run Pointwise.

```
pointwise
```

Note that Pointwise must be run from a Unix/Linux/DOS shell and not by clicking on the Pointwise icon; otherwise it will be unable to find `attributes.glf`.

When the Pointwise window has appeared, choose `Execute` from the `Script` menu. Navigate to the location of the scripts and select `prop1.glf`. Pointwise will now execute the script writing reports similar to the following to the Messages window:

```
Script: Reading P4382.igs
Script: propeller diameter = 0.304790791365
Script: Making connectors on the reference blade.
Script: Initial number of nodes in hub inflation layer: 7
Script: Making domains on the reference blade.
Script: Smoothing domains on the blade
Script: Extra smoothing on the blade domains
Script: Length ratios in each each domain during smoothing:
Script: Iter.   TE Domain        LE Domain        Tip Domain
Script:           I       J        I       J        I       J
Script:  0   1.8111  1.1166  1.4086  1.1458  1.2348  1.7913
Script:  1   1.6412  1.1164  1.3400  1.1304  1.1984  1.6406
Script:  2   1.5326  1.1161  1.2913  1.1215  1.1711  1.5297
Script:  3   1.4580  1.1160  1.2613  1.1169  1.1527  1.4480
Script:  4   1.4039  1.1159  1.2611  1.1142  1.1384  1.3869
Script:  5   1.3629  1.1159  1.2612  1.1126  1.1272  1.3589
Script: Quality of Blade Domains
Script:   Maximum length ratio: 1.3628791109 at 37 3 in domain dom-3
Script:   Minimum skew: 35.7020608896 at dom-10 node 45 59
Script:   Maximum equi-angle skew: 0.672136802867 at dom-12 node 1 64
Script: Saving current state in ./prop1.pw
Script: Writing blade-domains.dat
Script: Writing radius.dat
Script:
Script: To continue execute the script prop2.glf
```

Use the F2 key to make the display of the grid expand to the full screen. The Pointwise display window will now look similar to Figure 8. The blade domains are displayed as wireframes with hidden line removal. The hub is also shown as a smooth solid surface but the blade surface is invisible so that it will not obscure the blade domains.

Three layers are defined to make it easier to examine the display:

**`Reference Blade Surface`** (hidden)
Contains the five blade surfaces read from the IGES file.

**`Hub Surface`** (visible)
Contains the hub surface read from the IGES file.

**`Reference Blade Domains`** (visible)
Contains all connectors and domains created so far.

Rotate and zoom in on the grid to examine it closely for quality. Pay particular attention to the spacing of the nodes at the leading and trailing edges: is it small enough?

Some overall attributes of the quality of the mesh have already been written to the Messages window. During the extra smoothing of the blade domains (the smoothing done after the domains on the front and back of the blades have been joined to make domains which wrap around the leading and trailing edge: see Section 4.4), the maximum length ratios for each of the three domains was reported. The length ratio is the ratio of the size of a cell relative to its immediate neighbours. The closer the value is to 1, the better. In the output above, it can be seen that the smoothing reduced the length ratios on all domains.

The Messages window also reports on the minimum skew over all the cells; this is the smallest angle, in degrees, between adjacent sides of a cell. It also reports on the "Maximum equi-angle skew". This is a measure of the skew angle relative to the angle expected for an equilateral cell (for the actual formula used, see the Pointwise documentation [8]). For good grids its value will not exceed 0.8; values up to 0.9 can be tolerated.

You can use the Pointwise Examine menu to determine other aspects of the grid quality.

If you wish to try to improve the quality of the domains further, you may apply more smoothing to any of the blade domains before proceeding to the next script. However, you must not do anything that might cause the domain names to change: e.g. split a domain and rejoin it.

If necessary, adjust parameters by setting their values in the `attributes.glf` file, then rerun the script.

It is possible that `prop1.glf` will stop with a message similar to

```
Could not make domain dom-4 from four connectors. There is probably a mismatch in
the connectors.
```

The most likely cause is that the edges of the surfaces in the IGES file don't match to within the accuracy that you have specified with attribute `prop_accuracy`. If you created the IGES file with `smooth-prop`, check that the value of `prop_accuracy` matches the accuracy specified when running `smooth-prop`. The latter is listed in the description section at the top of the IGES file.

## 5.2 Running prop2.glf

When you are satisfied with the domains on the reference blade, execute the next script, `prop2.glf`. It will generate output in the Messages window similar to the following:

```
Script: Reading ./blade-domains.dat
Script: propeller diameter = 0.304790791365
Script: Making the blade blocks
Script: Number of nodes in blade inflation layer: 29
Script: Extruding structured blade blocks
Script:   Iteration 1 completed
              . . .
Script:   Iteration 28 completed
Script: Quality of extruded domains
Script:   Maximum aspect ratio: 6.76848280706 at 30 39 in domain dom-54
Script: Quality of Blade Blocks
Script:   Minimum skew: 35.7020608896 at blk-4 node 45 59 1
Script:   Maximum skew: 150.492312258 at blk-8 node 1 64 1
Script:   Maximum equi-angle skew: 0.672136802867 at blk-8 node 1 64 1
Script: Number of nodes in hub inflation layer: 24
Script: 710529 nodes in the blades blocks. Script: Saving current state in ./prop2.pw
Script: Writing blade-blocks.dat
Script:
Script: To continue execute the script prop3.glf
```

The Pointwise display window will now look similar to Figure 9. The blade domains are displayed as wireframes with hidden line removal so only the outermost domains can be seen; they are coloured green. The hub is also shown as a smooth solid surface.

Two new layers are defined by `prop2.glf`:

**`Next to Hub Blocks`** (visible)
    Contains all the blocks adjacent to the hub.

**Reference Blade Blocks** (visible)

Contains all connectors, domains and blocks not already assigned to a layer.

Examine the grid closely for quality, paying special attention to the cells near the leading and trailing edges. Some aspects of the quality of the blade blocks have already been written in the Messages window: the minimum and maximum skew angles in degrees, and the maximum equi-angle skew. Additional information on the quality of the blocks can be obtained using the Pointwise Examine menu.

If necessary, adjust parameters by setting their values in the `attributes.glf` file, use the Undo button to revert to the grid before `prop2.glf` was run, then rerun the script. If you change any of the attributes required by `prop1.glf`, you must start again from the beginning.

## 5.3 Running prop3.glf

When you are satisfied with the blocks on the reference blade, execute the next script, `prop3.glf`. It will generate output in the Messages window similar to the following:

```
Script: Reading ./blade-blocks.dat
Script: Reading ./radius.dat
Script: Making the domain on the hub
Script: Quality of Hub Domains
Script:   Minimum skew: 26.851183134 at dom-15 node 128
Script:   Maximum equi-angle skew: 0.5524802811 at dom-15 node 128
Script: Making rotated copies of grid entities
Script:   Rotation 1 completed
Script:   Rotation 2 completed
Script:   Rotation 3 completed
Script:   Rotation 4 completed
Script: Saving current state in ./prop3.pw
Script: Writing hub-domains.dat
Script:
Script: To continue execute the script prop4.glf
```

The Pointwise display window will now look similar to Figure 10. the outermost blade domains and the hub domains are displayed as wireframes with hidden line removal. The hub surface is made invisible so that it will not obscure the hub domains. The purple domain is the unstructured domain on the hub surface around the reference blade.

Two new layers are defined by `prop3.glf`:

**Blade Domains** (visible)

Contains all connectors and domains on the surfaces of the blades. The domains and connectors on the surface of the reference blade are moved from layer `Reference Blade Domains` to this layer; the layer `Reference Blade Domains` is removed as it no longer contains anything.

**`Hub Domains`** (visible)
Contains all connectors and domains on the hub.

**`Lower Outer Domains`** (hidden)
Contains the domains on the outer faces of the blade blocks adjacent to the hub (the domains which constrain the extrusion of the hub block).

Examine the reference domain on the hub surface closely for quality; it is the one shown in purple (the other hub domains are rotated copies of this one). Some aspects of the quality of the reference hub domain have already been written in the Messages window: the minimum skew angles in degrees and the maximum equi-angle skew. On a propeller with many blades, the distance between the blades can be quite small. That may require reducing the height of the blade inflation layer (attribute `blade_layer_height`) or the distance between nodes on the outer edges of the hub domain (attribute `hub_edge_spacing`).

If necessary, adjust parameters by setting their values in the `attributes.glf` file, use the Undo button to revert to the grid before `prop3.glf` was run, then rerun the script. If you change any of the attributes required by `prop2.glf` (e.g. `blade_layer_height`), you must revert to the state before `prop2.glf` was run. Do this by pressing the Undo button at least twice, or by clearing the current grid and loading the file `prop1.pw` from your current working directory. After each script is run, the state is saved in a file of the form `prop`*n*`.glf` to make it easy to back up and rerun any of the scripts.

## 5.4 Running prop4.glf

When you are satisfied with the domains on the hub, execute the next script, `prop4.glf`. It will generate output in the Messages window similar to the following:

```
Script: Reading ./hub-domains.dat
Script: Creating the hub blocks by extrusion
Script:   Connector generating spacing: con-41
Script:   23 extrusion steps
Script:   Iteration 1 completed
             . . .
Script:   Iteration 23 completed
Script: Quality of Hub Block
Script:   Minimum skew: 16.6890718414 at blk-46 node 15681 1 23
Script:   Maximum skew: 135.181371437 at blk-46 node 113 1 1
Script:   Maximum equi-angle skew: 0.721848802643 at blk-46 node 15681 1 23
Script: Quality of Interior Domains
```

```
Script:   Maximum length ratio: 1.34562744161 at 51 64 in domain dom-137
Script:   Minimum skew: 16.6890718414 at dom-220 node 121
Script:   Maximum equi-angle skew: 0.721848802643 at dom-220 node 121
Script: Saving current state in ./prop4.pw
Script: Writing hub-blocks.dat
Script:
Script: To continue execute the script prop5.glf
```

The Pointwise display window will now look similar to Figure 11. The outermost blade and hub domains are displayed as wireframes with hidden line removal. The hub surface is made invisible so that it will not obscure the hub domains. The green hub domains are the outer domains of the extruded hub block.

Three new layers are defined by `prop4.glf`:

**`Hub Block`** (hidden)
Contains the hub block and any connectors and domains that it uses which have not already been assigned to another layer.

**`Interior Domains`** (visible)
Contains the outermost blade and hub domains. These will be used as the inner surface of the main unstructured block by `prop6.glf`.

Examine the grid closely for quality, paying special attention to the regions near the roots of the blades. If the outer domain of the extruded hub block contains highly distorted cells it may be necessary to increase the smoothing during the hub extrusion (attribute `nsmooth_hub`). Also consider the aspect ratios of of the cells in the blade blocks along the leading and trailing edges. These should be no more than about 5.0; otherwise the pyramid elements built on these cells in the main unstructured block will have small minimum skew angles. If the aspect ratios are too large, it may be necessary to increase the number of nodes along the leading and trailing edges (attributes `num_nodes_tip` and `num_nodes_vert`).

If necessary, adjust parameters by setting their values in the `attributes.glf` file, revert to the state before the script where they the changed parameters were first used, then rerun.

## 5.5 Running prop5.glf

When you are satisfied with the blocks on the hub, execute the next script, `prop5.glf`. It will generate output in the Messages window similar to the following:

```
Script: Reading ./radius.dat
Script: Make the outer shell blocks.
Script: Saving current state in ./prop5.pw
```

```
Script: Writing shell-blocks.dat
Script:
Script: To continue execute the script prop6.glf
```

The Pointwise display window will now look similar to Figure 12. The visible domains are those on the outer surface of the cylindrical flow region.

Five new layers are defined by `prop5.glf`:

**`Inflow`** (visible)

Contains the connectors and domains on the upstream end-cap of the outer cylindrical shell.

**`Outflow`** (visible)

Contains the connectors and domains on the downstream end-cap of the outer cylindrical shell.

**`Outer Shell`** (visible)

Contains the connectors and domains on the circumferential portion of the outer cylindrical shell.

**`Inner Shell`** (visible)

Contains the connectors and domains on the surface of the inner cylindrical shell. These domains are used as the outer face of the main unstructured block.

**`Shell Blocks`** (hidden)

Contains the connectors, domains and blocks in the blocks between the shells which have not already been assigned to another layer.

The outer domains can be made invisible to reveal the propeller by deselecting the layers Inflow, Outflow and Outer Shell.

If either end of the hub lies within the inner cylindrcal shell, make the outer domains invisible and turn on the visibility of the layer Inner Shell. Check the size of the gap between the upstream and downstream ends of the cylinder and the ends of the hub; it should be at least about 10 times the cell size on the domains on the inner cylinder (attribute `cell_size`).

Examine the grid closely for quality, paying special attention to the regions near the roots of the blades. If necessary, adjust parameters by setting their values in `attributes.glf`, revert to the state before the script where they the changed parameters were first used, then rerun.

## 5.6 Running prop6.glf

When you are satisfied with the blocks in the outer shell, execute `prop6.glf`. It will generate output in the Messages window similar to the following:

```
Script: Reading ./hub-blocks.dat
Script: Reading ./shell-blocks.dat
Script: Making the main grid
Script:   Assembling faces
Script:   Initializing the block
Script:   1689452 nodes
Script: Quality of Main Block
Script:   Minimum skew: 9.34426314276 at blk-52 node 7126859
Script:   Maximum skew: 164.756230343 at blk-52 node 7126859
Script:   Maximum equi-angle skew: 0.867511344634 at blk-52 node 7126859
Script: Saving grid in ./P4382.grd
Script: Saving grid in ./P4382.pw
```

The display does not change after `prop6.glf` runs.

Sometimes the script `prop6.glf` will fail with the message

```
Script:   ERROR: There was an error running the solver
```

This indicates that the Pointwise algorithm for generating the main unstructured block has failed. The most likely cause is that the aspect ratios of the cells in its interior domains are too large or the skew angles are too small. Examine these domains very closely and try to make adjustments to improve their quality.

One new layer is defined by `prop6.glf`:

**`Main Block`** (visible)
Contains the main unstructured block.

# 6 Concluding remarks

A procedure has been developed for generating high quality grids on propellers using the program Pointwise and its scripting language Pointwise Glyph2. The grid is generated by running each of six scripts in turn. This report has described what each script does and provided instructions on how to run them.

The scripts could be extended in several ways:

- The scripts currently assume that there is no fillet between the blades and the hub; fillets could be included when an appropriate definition of their geometry is provided.

- The scripts generate an unstructured grid in the main portion of the flow. Since unstructured grids are known to increase the dissipation in the flow calculations, it may prove beneficial to replace the unstructured block with structured blocks to increase the accuracy of the flow calculations, particularly in the regions of the tip and hub vortices.

- Using the method described in Section 4.3, the scripts could be modified so that the domains on the central blade surface are unstructured, thus decoupling the numbers of nodes along the leading and trailing edges and allowing more flexibility in the concentration of nodes. Since the flow near the centre of the blade is typically very regular, it is unlikely that the extra dissipation from changing from structured to unstructured domains would be significant, but that would need to be checked.

# References

[1] (2009), Pointwise: Reliable CFD Meshing You Trust (online), Pointwise, Inc., Fort Worth, Texas, `http://www.pointwise.com` (Access Date: November 2009).

[2] (2010), Glyph2: version 2.3 (online), Pointwise, Inc., Fort Worth, Texas, `http://www.pointwise.com/glyph2` (Access Date: December 2010).

[3] Ousterhout, J. K. and Jones, K. (2009), Tcl and the Tk Toolkit, 2$^{\text{nd}}$ ed, Addison-Wesley Professional.

[4] Hally, D. (2013), C++ classes for representing propeller geometry, (DRDC Atlantic TM 2013-177) Defence Research and Development Canada – Atlantic.

[5] Hally, D. (2013), Smoothing propeller tip geometry for use in a RANS solver, (DRDC Atlantic TM 2013-178) Defence Research and Development Canada – Atlantic.

[6] Hally, D. (2013), User's guide for smooth-prop: a program to smooth propeller tip geometry, (DRDC Atlantic TM 2013-179) Defence Research and Development Canada – Atlantic.

[7] (1988), Initial Graphics Exchange Specification (IGES) Version 4.0, US Dept. of Commerce, National Bureau of Standards. Document No. NBSIR 88-3813.

[8] (2008), Pointwise User's Manual, Pointwise, Inc. Included with the Pointwise distribution in the directory `doc`.

This page intentionally left blank.

# Annex A: List of attributes

The following attributes are used to control the grid generation process. All but three (`prop_iges_file`, `CFX_file` and `PW_file`) are assigned a default value in by the script `default-attributes.glf`. The user sets the values of the three undefined attributes and overrides default values using the script `attributes.glf`.

**`prop_iges_file`**

Default value: None
First used in `prop1.glf`
The name of the IGES file containing the propeller geometry: i.e. the file created by the program `smooth-prop`.

**`prop_accuracy`**

Default value: `8.0e-05`
First used in `prop1.glf`
The propeller accuracy with respect to the propeller diameter. This value should not exceed the value used by `smooth-prop` when generating the IGES file.

**`num_nodes_tip`**

Default value: `51`
First used in `prop1.glf`
$N_{tip}$: the dimension of the connector 18 in Figure 7.

**`num_nodes_around_le`**

Default value: `71`
First used in `prop1.glf`
The number of nodes on connectors passing around the leading and trailing edges. The dimension $N_e$, used for connectors 1, 3, 8, 10, 15, 17 in Figure 7, has the value `(num_nodes_around_le+1)/2`.

**`num_nodes_vert`**

Default value: `71`
First used in `prop1.glf`
$N_{vert}$: the dimension of connectors 11, 12, 13, 14 in Figure 7.

**`le_spacing_root`**

Default value: `0.0005`
First used in `prop1.glf`
$s_{le-root}$: The node spacing on connector 3 in Figure 7 at the end lying on the leading edge.

**`le_spacing_tip`**

Default value: `0.0001`

First used in `prop1.glf`

$s_{le-tip}$: The node spacing on connector 17 in Figure 7 at the end lying on the leading edge.

**`te_spacing_root`**

Default value: `0.0002`

First used in `prop1.glf`

$s_{te-root}$: The node spacing on connector 1 in Figure 7 at the end lying on the trailing edge.

**`te_spacing_tip`**

Default value: `0.0001`

First used in `prop1.glf`

$s_{te-tip}$: The node spacing on connector 15 in Figure 7 at the end lying on the trailing edge.

**`le_spacing`**

Default value: `0.0`

First used in `prop1.glf`

$s_{le}$: the node spacing on connector 18 in Figure 7 at the end lying on the leading edge. If this value is zero, then the spacing will be set to the average node spacing for the connector: i.e. the arclength of connector 18 divided by $N_{tip} - 1$.

**`te_spacing`**

Default value: `0.0`

First used in `prop1.glf`

$s_{te}$: The node spacing on connector 18 in Figure 7 at the end lying on the trailing edge. If this value is zero, then the spacing will be set to the average node spacing for the connector: i.e. the arclength of connector 18 divided by $N_{tip} - 1$.

**`hub_layer_height`**

Default value: `0.03`

First used in `prop1.glf`

Height of the hub inflation layer with respect to the propeller radius.

**`hub_wall_spacing`**

Default value: `5.0e-05`

First used in `prop1.glf`

$s_h$: the spacing at the bottom of the hub inflation layer with respect to the propeller diameter.

**`hub_growth_rate`**

Default value: `1.2`

First used in `prop1.glf`

$g_h$: the rate of growth of cells in the hub inflation layer.

**nsmooth_blade**

Default value: `10`
First used in `prop1.glf`
The number of smoothing iterations for the structured domains on the blade surface while they are split at the leading and trailing edges.

**nsmooth_blade_joined**

Default value: `5`
First used in `prop1.glf`
The number of smoothing iterations for the blade domains after they have been rejoined across the leading and trailing edges.

**blade_smoothing_edge_control**

Default value: `Default`
First used in `prop1.glf`
The edge control for the smoothing of the blade domains. Its values can be `Default`, `HilgenstockWhite`, `StegerSorenson` or `None`.

**blade_layer_height**

Default value: `0.05`
First used in `prop2.glf`
Height of the blade inflation layer with respect to radius.

**blade_wall_spacing**

Default value: `2.0e-05`
First used in `prop2.glf`
$s_b$: the spacing at the bottom of the blade inflation layer.

**blade_growth_rate**

Default value: `1.2`
First used in `prop2.glf`
$g_b$: the rate of growth of cells in the blade inflation layer.

**extrusion_smoothing**

Default value: `Default`
First used in `prop2.glf`
The smoothing factor in [0,10] used for the extrusion of the structured blade blocks. A value of `Default` is allowed.

**extrusion_KB_smoothing**

Default value: `Default`
First used in `prop2.glf`
The Kinsey-Barth smoothing factor used during extrusion of the blade blocks. A value of `Default` is allowed.

**hub_edge_spacing**

Default value: `0.01`
First used in `prop3.glf`
Spacing of nodes on the edge of the hub with respect to the propeller diameter.

**hub_boundary_decay**

Default value: `0.9`
First used in `prop3.glf`
Boundary decay factor for the hub domains: 0 = boundary has no influence; 1 = boundary has maximum influence.

**nsmooth_hub**

Default value: `100`
First used in `prop4.glf`
The number of smoothing iterations for the extrusion of the hub block.

**outer_radius**

Default value: `5.0`
First used in `prop5.glf`
The radius of the outer cylinder with respect to the prop radius.

**inner_radius**

Default value: `2.0`
First used in `prop5.glf`
The radius of the inner cylinder with respect to the prop radius.

**zmin_inner**

Default value: `-1.0`
First used in `prop5.glf`
The value of $z$ with respect to propeller radius at the upstream end of the inner cylindrical shell. The propeller plane is at $z = 0$.

**zmax_inner**

Default value: `1.0`
First used in `prop5.glf`
The value of $z$ with respect to propeller radius at the downstream end of the inner cylindrical shell.

**zmin_outer**

Default value: `-5.0`
First used in `prop5.glf`
The value of $z$ with respect to propeller radius at the upstream end of the outer cylindrical shell.

**zmax_outer**

Default value: `5.0`

First used in `prop5.glf`

The value of $z$ with respect to propeller radius at the downstream end of the outer cylindrical shell.

**cell_size**

Default value: `0.15`

First used in `prop5.glf`

The size of cells on the inner cylinder with respect to the propeller radius.

**zmin_spacing**

Default value: `0.0`

First used in `prop5.glf`

Spacing with respect to the propeller radius in the axial direction at the upstream end of the outer cylinder. If zero, will be set so that the cell size increases in proportion to the radial cell size.

**zmax_spacing**

Default value: `0.0`

First used in `prop5.glf`

Spacing with respect to the propeller radius in the axial direction at the downstream end of the outer cylinder. If zero, will be set so that the cell size increases in proportion to the radial cell size.

**main_boundary_decay**

Default value: `0.8`

First used in `prop6.glf`

Decay factor for the main block.

**pyramid_aspect_ratio**

Default value: `Default`

First used in `prop6.glf`

Aspect ratio for the height of pyramids. A value of `Default` is allowed. Smaller values will sometimes increase minimum interior angles in the main grid.

**num_block_iter**

Default value: `1`

First used in `prop6.glf`

Number of iterations used when making the main block.

**CFX_file**

Default value: None

First used in `prop6.glf`

The name of the output file containing the propeller grid in ANSYS CFX format.

**`PW_file`**

Default value: None

First used in `prop6.glf`

The name of the output file containing the propeller grid in Pointwise format.

# Annex B: Specification of the IGES file defining the propeller geometry

The scripts described in this report use an IGES file to define the geometry of the propeller and the hub. This annex gives a complete specification for the file. Refer to the IGES documentation [7] for a complete description of the IGES format.

In the description below, the parameters of a surface will be denoted $(\xi,\eta)$. The range of $\xi$ is $[\xi_{\mathrm{lo}},\xi_{\mathrm{hi}}]$ and the range of $\eta$ is $[\eta_{\mathrm{lo}},\eta_{\mathrm{hi}}]$.

1. The propeller accuracy is the value of the Minimum User-Intended Resolution field in the Global Section of the file. It will be denoted by $\varepsilon$.
2. The file must contain six physically dependent smooth parametric surfaces: five defining the reference propeller blade and one defining the sector of the hub containing the footprint of the reference blade. The surfaces are grouped in a Subfigure Definition Entity which is used in $Z$ Singular Subfigure Instance Entities, the $n^{\mathrm{th}}$ instance representing a copy of the reference blade and its hub sector rotated through $2\pi n/Z$ about the $z$ axis.
3. The union of the resulting $6Z$ surfaces should form a closed surface to within the blade accuracy with the exception of two possible holes: one at each end of the hub.
4. The five surfaces defining the reference blade will all have label `Blade` and their subscript numbers will be 1 through 5. For each surface the parameter range is $[0,1]\times[0,1]$: i.e. $\xi_{\mathrm{lo}}=0$, $\xi_{\mathrm{hi}}=1$, $\eta_{\mathrm{lo}}=0$ and $\eta_{\mathrm{hi}}=1$. In the terminology used earlier, `Blade 1` is the trailing edge surface, `Blade 2` is the central surface pressure side, `Blade 3` is the leading edge surface, `Blade 4` is the central surface suction side and `Blade 5` is the tip surface.
5. The curve $\xi=0.5$ in surface `Blade 1` lies along the trailing edge with $(\xi,\eta)=(0.5,0.0)$ being the point where the trailing edge meets the hub.
6. The curve $\xi=0.5$ in surface `Blade 3` lies along the leading edge with $(\xi,\eta)=(0.5,0.0)$ being the point where the leading edge meets the hub. `Blade 3` is the leading edge surface.
7. The curve $\eta=0.5$ in surface `Blade 5` lies along the union of the leading and trailing edges. The point with $(\xi,\eta)=(0.5,0.0)$ lies on the trailing edge and the point $(\xi,\eta)=(0.5,1.0)$ lies on the leading edge.
8. Let the surface `Blade i` be denoted $\mathrm{s}_i(\xi,\eta)$. Then the edges of the blade surfaces must match according to the following rules:
   (a) $|\mathrm{s}_1(1,\eta)-\mathrm{s}_2(0,\eta)|<\varepsilon$ for $\eta\in[0,1]$;

(b) $|s_2(1,\eta) - s_3(0,\eta)| < \varepsilon$ for $\eta \in [0,1]$;

(c) $|s_3(1,\eta) - s_4(0,\eta)| < \varepsilon$ for $\eta \in [0,1]$;

(d) $|s_4(1,\eta) - s_1(0,\eta)| < \varepsilon$ for $\eta \in [0,1]$;

(e) $|s_5(0,\eta) - s_1(1-\eta,1)| < \varepsilon$ for $\eta \in [0,1]$;

(f) $|s_5(1,\eta) - s_3(\eta,1)| < \varepsilon$ for $\eta \in [0,1]$;

(g) $|s_5(\xi,0) - s_2(\xi,1)| < \varepsilon$ for $\xi \in [0,1]$;

(h) $|s_5(\xi,1) - s_4(1-\xi,1)| < \varepsilon$ for $\xi \in [0,1]$.

9. The hub sector is a Trimmed (Parametric) Surface (IGES entity type 144) with label `Trim Hub`. It has two bounding curves: the first is the footprint of the reference blade; the second is the perimeter of the hub sector.

10. The union of the curves with $\eta = 0$ and $\xi \in [0,1]$ on surfaces `Blade 1`, `Blade 2`, `Blade 3` and `Blade 4` must match the reference blade footprint on trimmed surface `Trim Hub` to within $\varepsilon$.

# List of symbols

| | |
|---|---|
| $\xi$ | A parameter for the reference blade geometry. |
| $\eta$ | A parameter for the reference blade geometry. |
| $\mathrm{b}(\xi,\eta)$ | A point on the reference blade. |
| $g_b$ | The growth rate in the blade inflation layer. |
| $g_h$ | The growth rate in the hub inflation layer. |
| $h_h$ | The height of the hub inflation layer. |
| $N_\theta$ | The number of nodes around the circumference of the cylindrical shell blocks. |
| $N_b$ | The number of nodes across the blade inflation layer. |
| $N_e$ | The dimension of edges 1, 3, 8, 10, 15 and 17: see Figure 7. |
| $N_c$ | The number of nodes between the inner and outer cylinders of the shell. |
| $N_h$ | The number of nodes across the hub inflation layer. |
| $N_{tip}$ | The dimension of connectors 2, 9, 16 and 18: see Figure 7. |
| $N_{vert}$ | The dimension of connectors 11, 12, 13 and 14: see Figure 7. |
| $N_{zmin}$ | The number of nodes between the upstream ends of the inner and outer cylinders of the shell. |
| $N_{zmax}$ | The number of nodes between the downstream ends of the inner and outer cylinders of the shell. |
| $R$ | Propeller radius. |
| $R_o$ | The radius of the outer cylinder of the shell. |
| $R_{hd}$ | The radius of the portion of the hub extended to the downstream boundary. |
| $R_{hu}$ | The radius of the portion of the hub extended to the upstream boundary. |
| $R_i$ | The radius of the inner cylinder of the shell. |
| $s_h$ | The hub wall spacing. |
| $s_i$ | The spacing of nodes, both radially and axially, on the inner cylinder of the shell. |
| $s_o$ | The radial spacing of nodes on the outer cylinder of the shell. |

| | |
|---|---|
| $s_{hu}$ | The spacing at the upper end of connectors rising through the hub inflation layer. |
| $s_{le}$ | The spacing at the end of connector 18 lying on the leading edge. |
| $s_{le-root}$ | The spacing at the end of connector 3 lying on the leading edge. |
| $s_{le-tip}$ | The spacing at the end of connector 17 lying on the leading edge. |
| $s_o$ | The size of cells in the outer shell blocks. |
| $s_{te}$ | The spacing at the end of connector 18 lying on the trailing edge. |
| $s_{te-root}$ | The spacing at the end of connector 1 lying on the trailing edge. |
| $s_{te-tip}$ | The spacing at the end of connector 15 lying on the trailing edge. |
| $s_{zmin}$ | The spacing in the $z$ direction at the inflow plane. |
| $s_{zmax}$ | The spacing in the $z$ direction at the outflow plane. |
| $Z$ | The number of propeller blades. |
| $z_{imax}$ | The value of $z$ at the downstream end of the inner cylindrical shell. |
| $z_{imin}$ | The value of $z$ at the upstream end of the inner cylindrical shell. |
| $z_{omax}$ | The value of $z$ at the downstream end of the outer cylindrical shell. |
| $z_{omin}$ | The value of $z$ at the upstream end of the outer cylindrical shell. |

## DOCUMENT CONTROL DATA

(Security markings for the title, abstract and indexing annotation must be entered when the document is Classified or Designated.)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence Research and Development Canada – Atlantic
PO Box 1012, Dartmouth NS  B2Y 3Z7, Canada

2a. SECURITY MARKING (Overall security marking of the document, including supplemental markings if applicable.)

UNCLASSIFIED

2b. CONTROLLED GOODS

(NON-CONTROLLED GOODS)
DMC A
REVIEW: GCEC APRIL 2011

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

A procedure to create CFD meshes around propellers

4. AUTHORS (Last name, followed by initials – ranks, titles, etc. not to be used.)

Hally, D.

5. DATE OF PUBLICATION (Month and year of publication of document.)

November 2013

6a. NO. OF PAGES (Total containing information. Include Annexes, Appendices, etc.)

60

6b. NO. OF REFS (Total cited in document.)

8

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence Research and Development Canada – Atlantic
PO Box 1012, Dartmouth NS  B2Y 3Z7, Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

Project 01gl08

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Atlantic TM 2013-180

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

( X ) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11)) is possible, a wider announcement audience may be selected.)

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

A procedure has been developed for generating computational grids that can be used for calculating the flow around propellers using Computational Fluid Dynamics (CFD) flow solvers. The grid generation program Pointwise from Pointwise, Inc. is used. The procedures are encapsulated in six scripts in the Pointwise scripting language Pointwise Glyph2. Each script is described in detail and instructions for their use are also provided.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Computational Fluid Dynamics
Grid generation
Propellers
Pointwise
Pointwise Glyph2
IGES
ANSYS CFX

**Defence R&D Canada**

Canada's leader in defence
and National Security
Science and Technology

**R & D pour la défense Canada**

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

**www.drdc-rddc.gc.ca**